Application Serial No. 17/027,481

Filing date: September 21, 2023

Patent No. 11,514,138

Issue date: November 29, 2022

US011514138B1

(12) **United States Patent**
Jakobsson

(10) **Patent No.: US 11,514,138 B1**
(45) **Date of Patent: *Nov. 29, 2022**

(54) **AUTHENTICATION TRANSLATION**

(71) Applicant: **RightQuestion, LLC**, Portola Valley, CA (US)

(72) Inventor: **Bjorn Markus Jakobsson**, Portola Valley, CA (US)

(73) Assignee: **RightQuestion, LLC**, Portola Valley, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **17/027,481**

(22) Filed: **Sep. 21, 2020**

Related U.S. Application Data

(63) Continuation of application No. 16/773,767, filed on Jan. 27, 2020, now Pat. No. 10,929,512, which is a continuation of application No. 16/563,715, filed on Sep. 6, 2019, now Pat. No. 10,824,696, which is a continuation of application No. 16/273,797, filed on Feb. 12, 2019, now Pat. No. 10,521,568, which is a continuation of application No. 15/042,636, filed on

(Continued)

(51) **Int. Cl.**

G06F 21/00	(2013.01)
G06F 21/10	(2013.01)
G06F 21/31	(2013.01)
H04L 9/40	(2022.01)
G06F 21/12	(2013.01)
G06F 21/44	(2013.01)
G06F 21/32	(2013.01)

(52) **U.S. Cl.**

CPC ***G06F 21/10*** (2013.01); ***G06F 21/121*** (2013.01); ***G06F 21/128*** (2013.01); ***G06F 21/31*** (2013.01); ***G06F 21/32*** (2013.01); ***G06F 21/44*** (2013.01); ***H04L 63/083*** (2013.01); ***H04L 63/0861*** (2013.01); ***H04L 63/10*** (2013.01); ***H04L 63/20*** (2013.01)

(58) **Field of Classification Search**

None

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,010,571	A	4/1991	Katznelson
5,499,298	A	3/1996	Narasimhalu

(Continued)

FOREIGN PATENT DOCUMENTS

WO	2004051585	6/2004
WO	2005001751	1/2005

OTHER PUBLICATIONS

Brands et al. Distance-Bounding Protocols. Jan. 28, 1994: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.51.6437&rep=rep1&type=pdf.

(Continued)

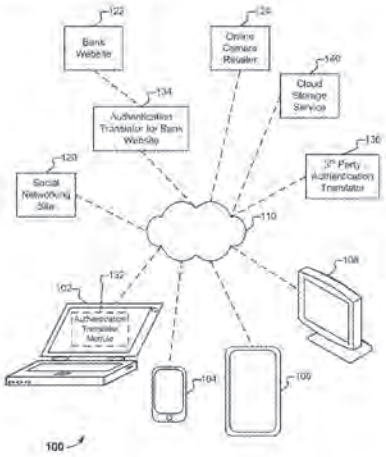
Primary Examiner — Andrew J Steinle

(74) *Attorney, Agent, or Firm* — Van Pelt, Yi & James LLP

(57) **ABSTRACT**

Authentication translation is disclosed. A request to access a resource is received at an authentication translator, as is an authentication input. The authentication input corresponds to at least one stored record. The stored record is associated at least with the resource. In response to the receiving, a previously stored credential associated with the resource is accessed. The credential is provided to the resource.

25 Claims, 8 Drawing Sheets



Related U.S. Application Data

Feb. 12, 2016, now Pat. No. 10,360,351, which is a continuation of application No. 13/706,254, filed on Dec. 5, 2012, now Pat. No. 9,294,452.

(60) Provisional application No. 61/587,387, filed on Jan. 17, 2012, provisional application No. 61/569,112, filed on Dec. 9, 2011.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,016,476	A	1/2000	Maes	
6,691,232	B1 *	2/2004	Wood	H04L 63/0815 726/6
7,512,965	B1	3/2009	Amdur	
7,697,729	B2	4/2010	Howell	
7,780,080	B2 *	8/2010	Owen	G06Q 20/3674 235/382
7,950,051	B1	5/2011	Spitz	
8,145,916	B2	3/2012	Boshra	
8,549,300	B1 *	10/2013	Kumar	H04L 9/3263 713/175
8,577,813	B2	11/2013	Weiss	
8,776,214	B1 *	7/2014	Johansson	H04L 63/08 726/19
8,856,539	B2	10/2014	Weiss	
8,984,596	B2	3/2015	Griffin	
9,100,826	B2	8/2015	Weiss	
10,872,152	B1	12/2020	Martel	
2004/0107170	A1	6/2004	Labrou	
2004/0236632	A1	11/2004	Maritzen	
2005/0198348	A1	9/2005	Yeates	
2006/0085844	A1	4/2006	Buer	
2007/0257104	A1	11/2007	Owen	
2007/0266256	A1	11/2007	Shah	
2008/0059804	A1	3/2008	Shah	
2009/0100269	A1	4/2009	Naccache	
2009/0191846	A1 *	7/2009	Shi	H04L 63/0861 455/411
2010/0242102	A1	9/2010	Cross	
2011/0078771	A1	3/2011	Griffin	
2011/0138450	A1	6/2011	Kesanupalli	
2011/0205016	A1	8/2011	Al-Azem	
2011/0231651	A1	9/2011	Bollay	
2012/0110341	A1 *	5/2012	Beigi	H04W 12/069 713/186
2012/0167193	A1	6/2012	Gargaro	
2014/0250079	A1	9/2014	Gardner	
2017/0230179	A1 *	8/2017	Mannan	H04L 9/3226

OTHER PUBLICATIONS

Jakobsson et al. Proving Without Knowing: On Oblivious, Agnostic and Blindfolded Provers. Jul. 24, 1996: http://markus-jakobsson.com/papers/jakobsson-crypto96.pdf.

Monrose et al. Using Voice to Generate Cryptographic Keys. May 13, 2001: https://www.cs.unc.edu/~fabian/papers/odyssey.pdf.

Seshadri et al. Pioneer: Verifying Code Integrity and Enforcing Untampered Code Execution on Legacy Systems. Oct. 23, 2005: https://netsec.ethz.ch/publications/papers/pioneer.pdf.

"Managing Authorization and Access Control", Author: unknown, Published Nov. 3, 2005, pp. 1-12, URL: http://technet.microsoft.com/en-us/library/bb457115.aspx.

Hammer-Lahav, Ed. "The OAuth 1.0 Protocol", from https://tools.ietf.org/html/rfc5849, Apr. 2010.

IPR2022-00244 Claim Mapping Table for the '696 Patent. Nov. 30, 2021.

IPR2022-00244 Petition for Inter Partes Review of U.S. Pat. No. 10,824,696. Nov. 30, 2021.

IPR2022-00251 Claim Mapping Table for the '512 Patent. Dec. 1, 2021.

IPR2022-00251 Petition for Inter Partes Review of U.S. Pat. No. 10,929,512. Dec. 1, 2021.

* cited by examiner

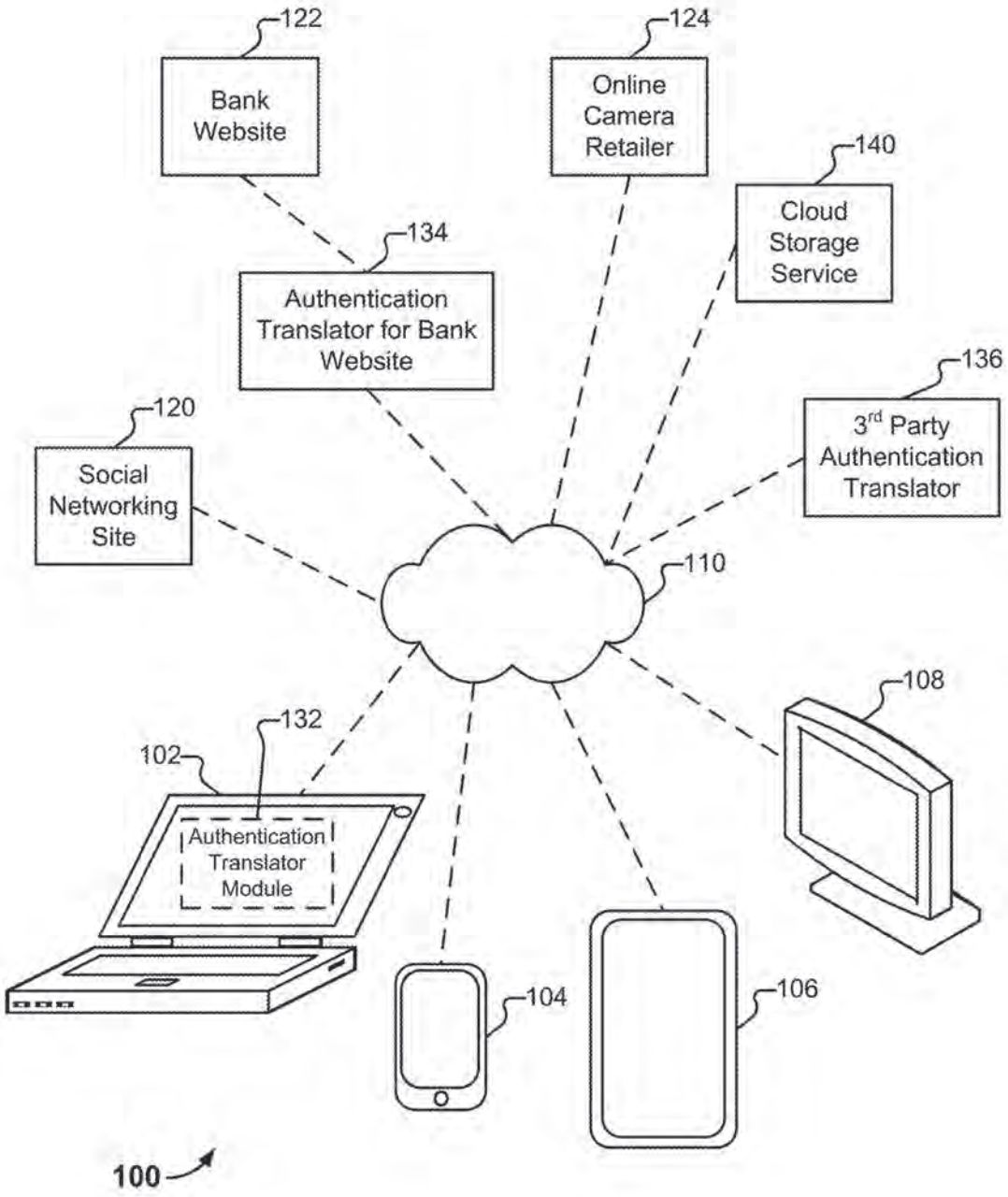

122
Bank Website
124
Online Camera Retailer
140
Cloud Storage Service
134
Authentication Translator for Bank Website
136
3rd Party Authentication Translator
120
Social Networking Site
110
108
132
102
Authentication Translator Module
104
106
100

FIG. 1

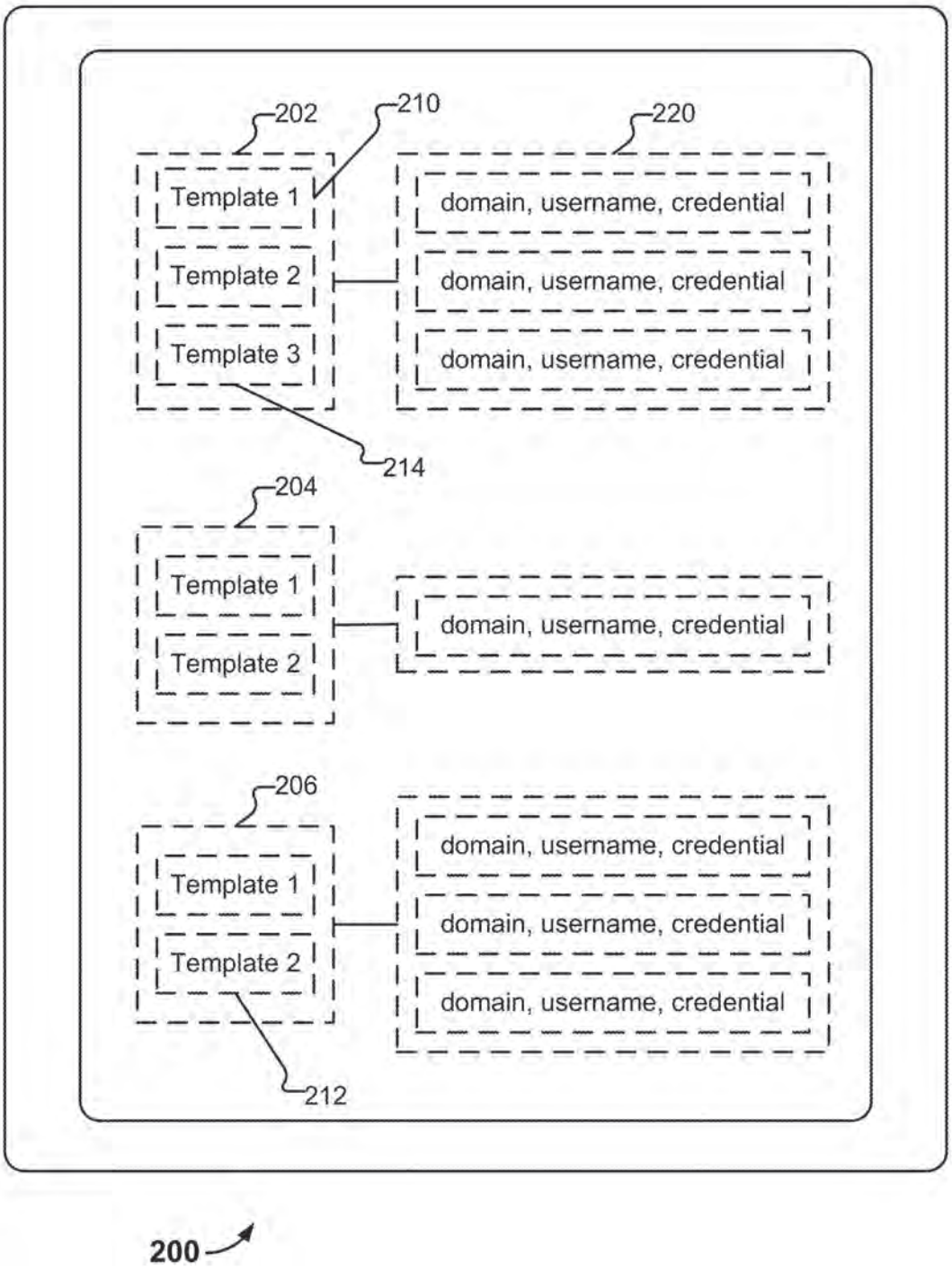
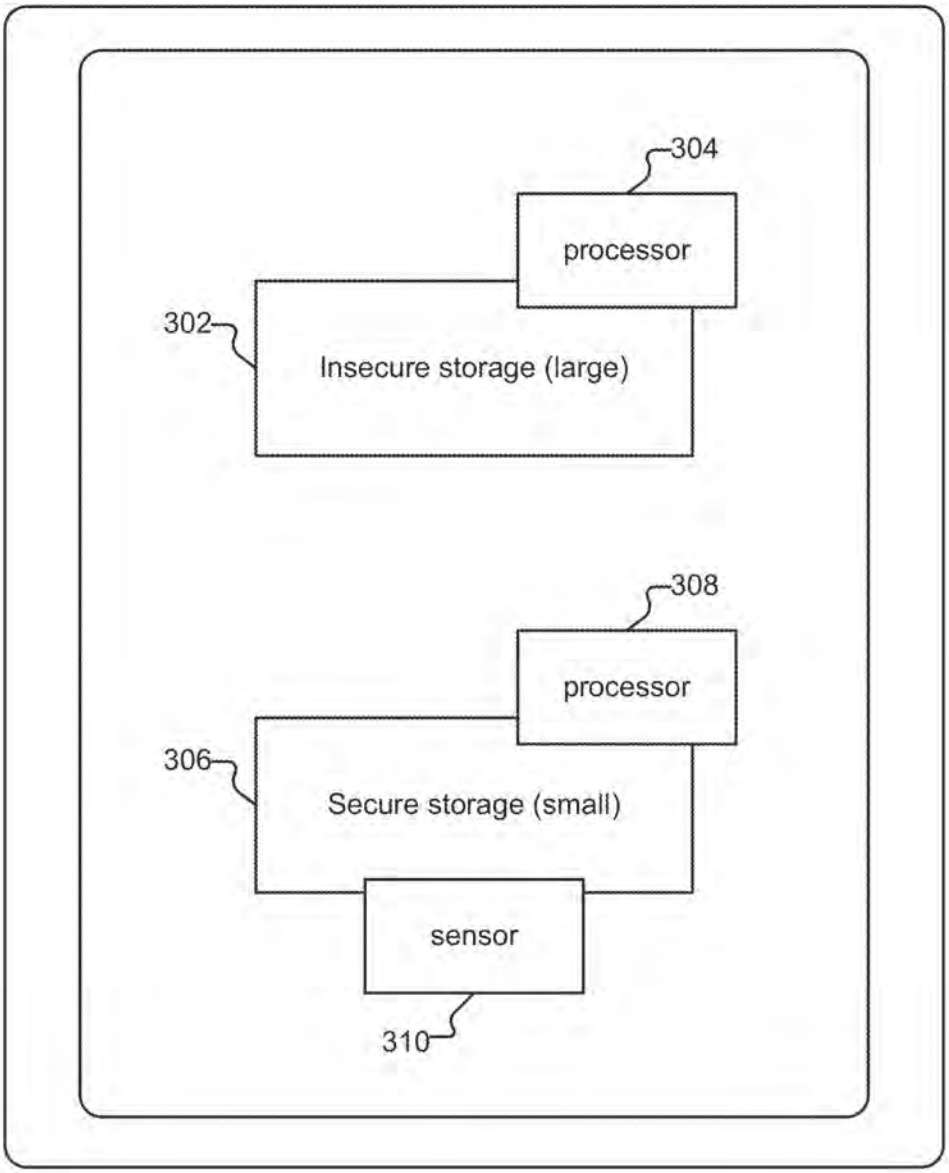

202
210
220
Template 1
Template 2
Template 3
domain, username, credential
domain, username, credential
domain, username, credential
214
204
Template 1
Template 2
domain, username, credential
206
Template 1
Template 2
domain, username, credential
domain, username, credential
domain, username, credential
212
200

FIG. 2


304
processor
302
Insecure storage (large)
308
processor
306
Secure storage (small)
sensor
310

300

FIG. 3

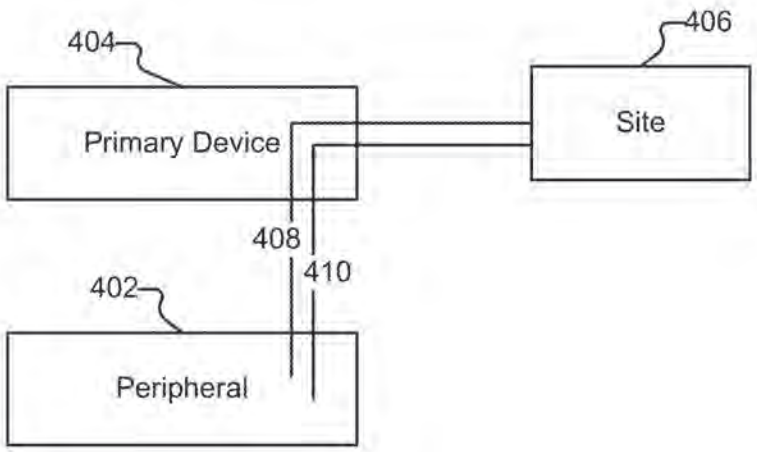


FIG. 4

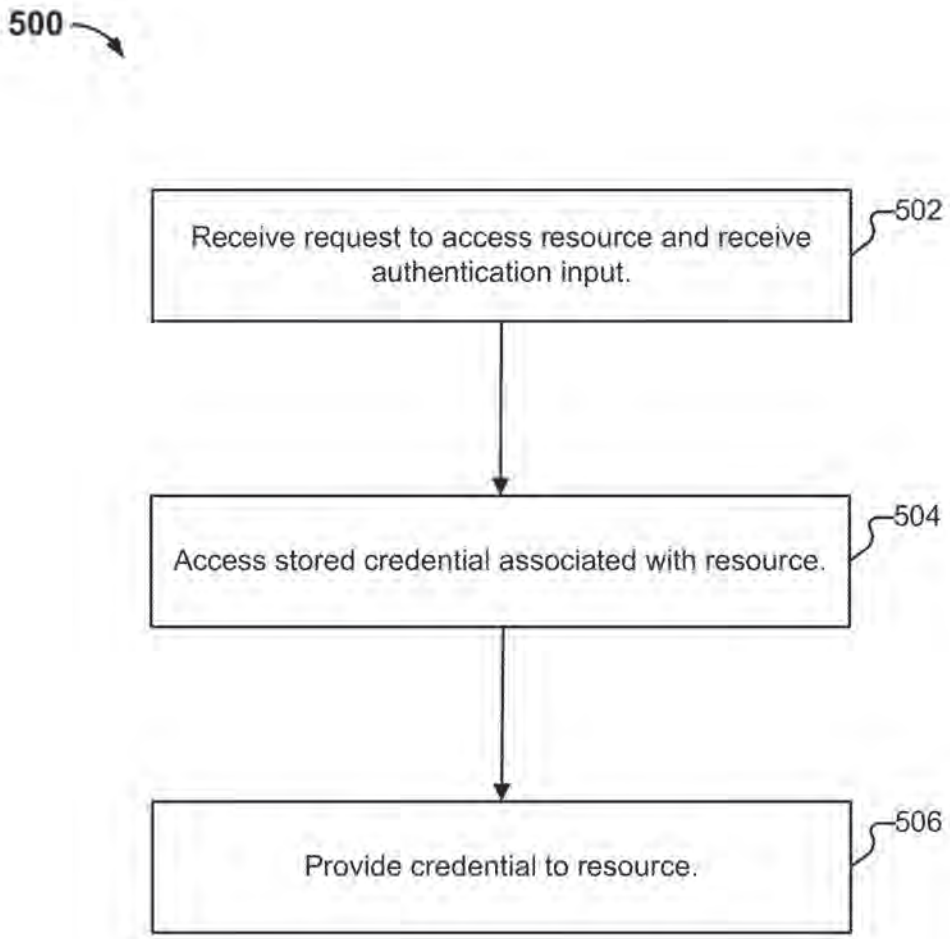

500
Receive request to access resource and receive authentication input.
502
Access stored credential associated with resource.
504
Provide credential to resource.
506

FIG. 5

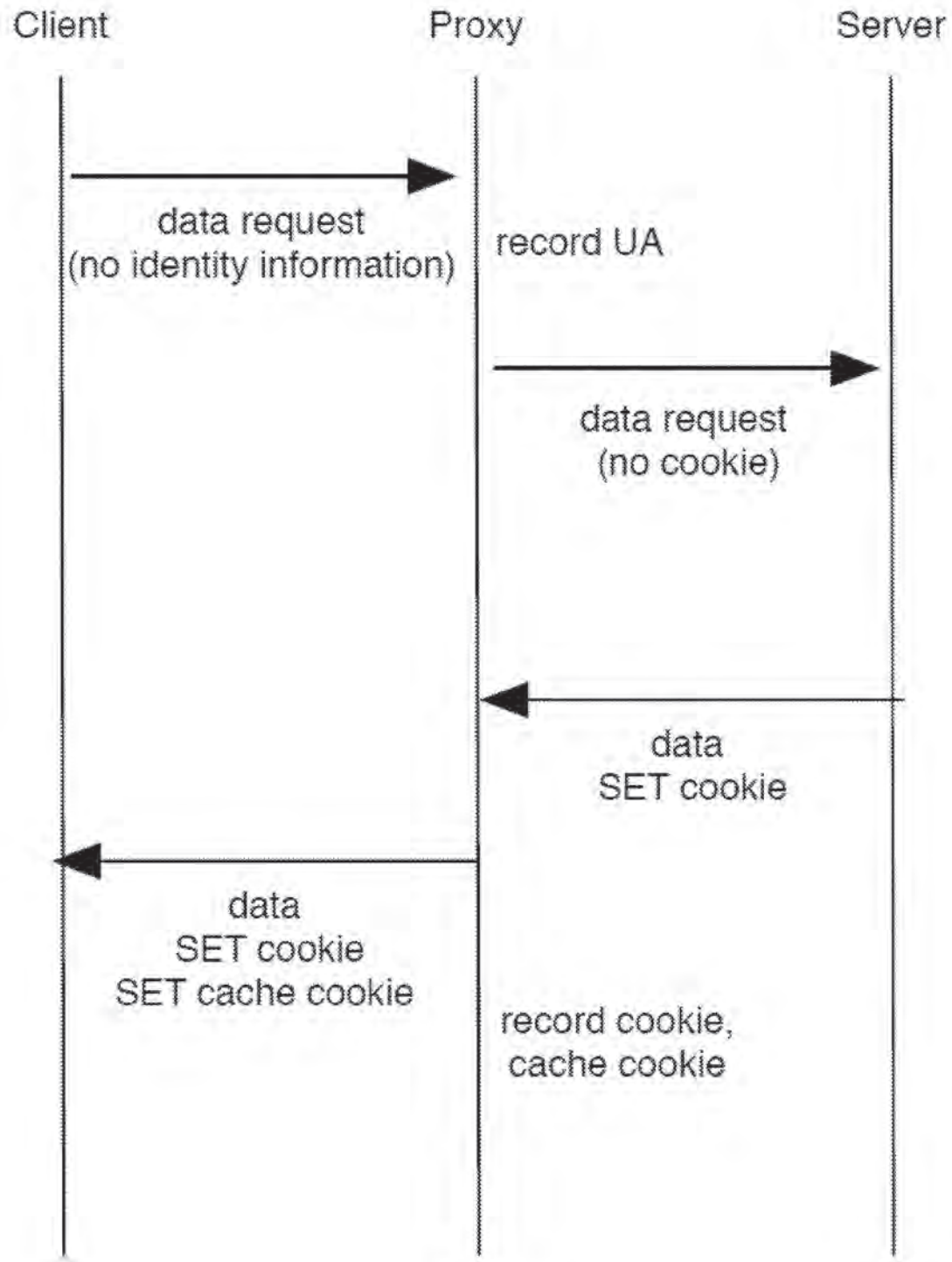

Client
Proxy
Server
data request
(no identity information)
record UA
data request
(no cookie)
data
SET cookie
data
SET cookie
SET cache cookie
record cookie,
cache cookie

FIG. 6

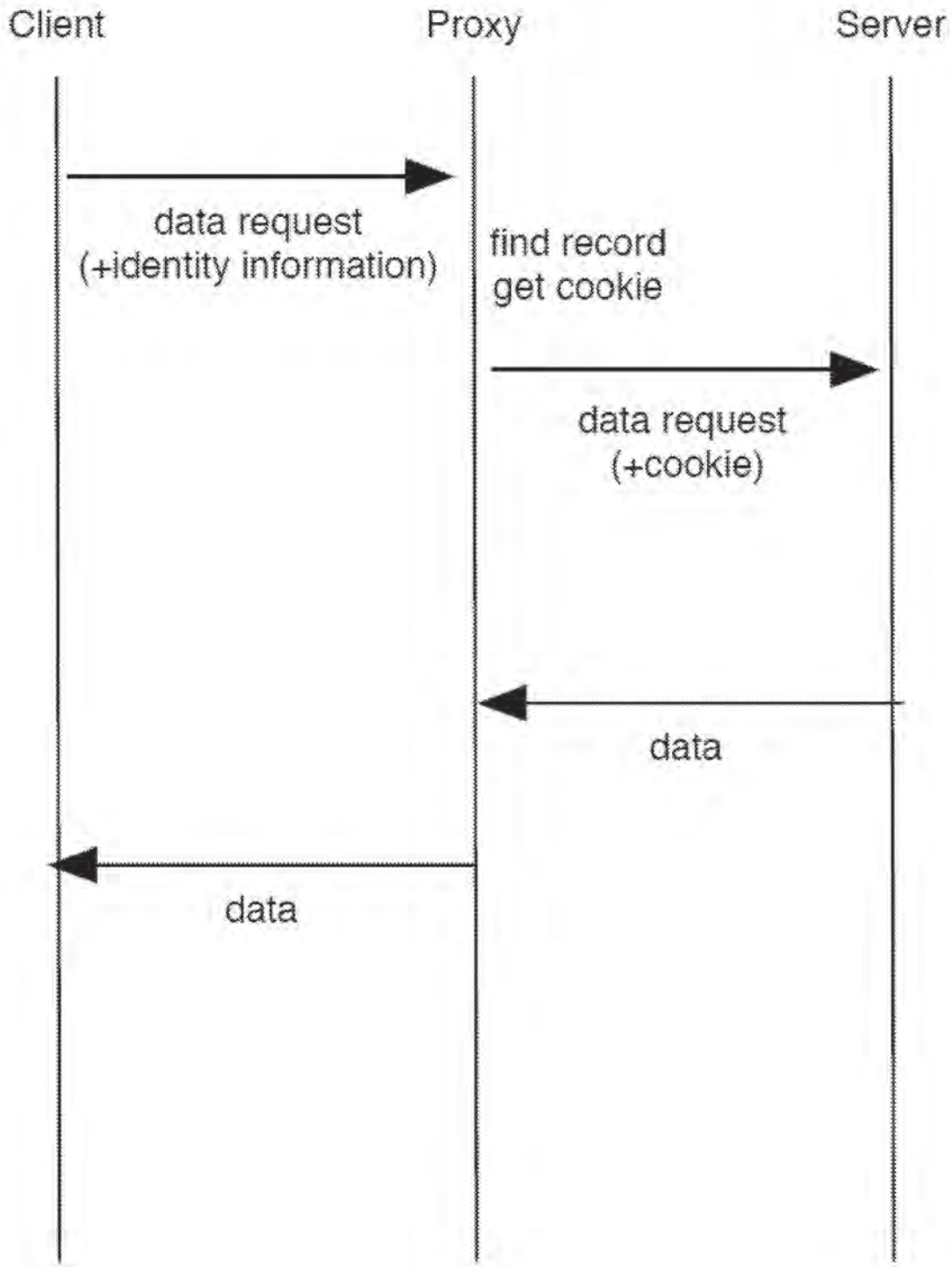

Client
Proxy
Server
data request
(+identity information)
find record
get cookie
data request
(+cookie)
data
data

FIG. 7

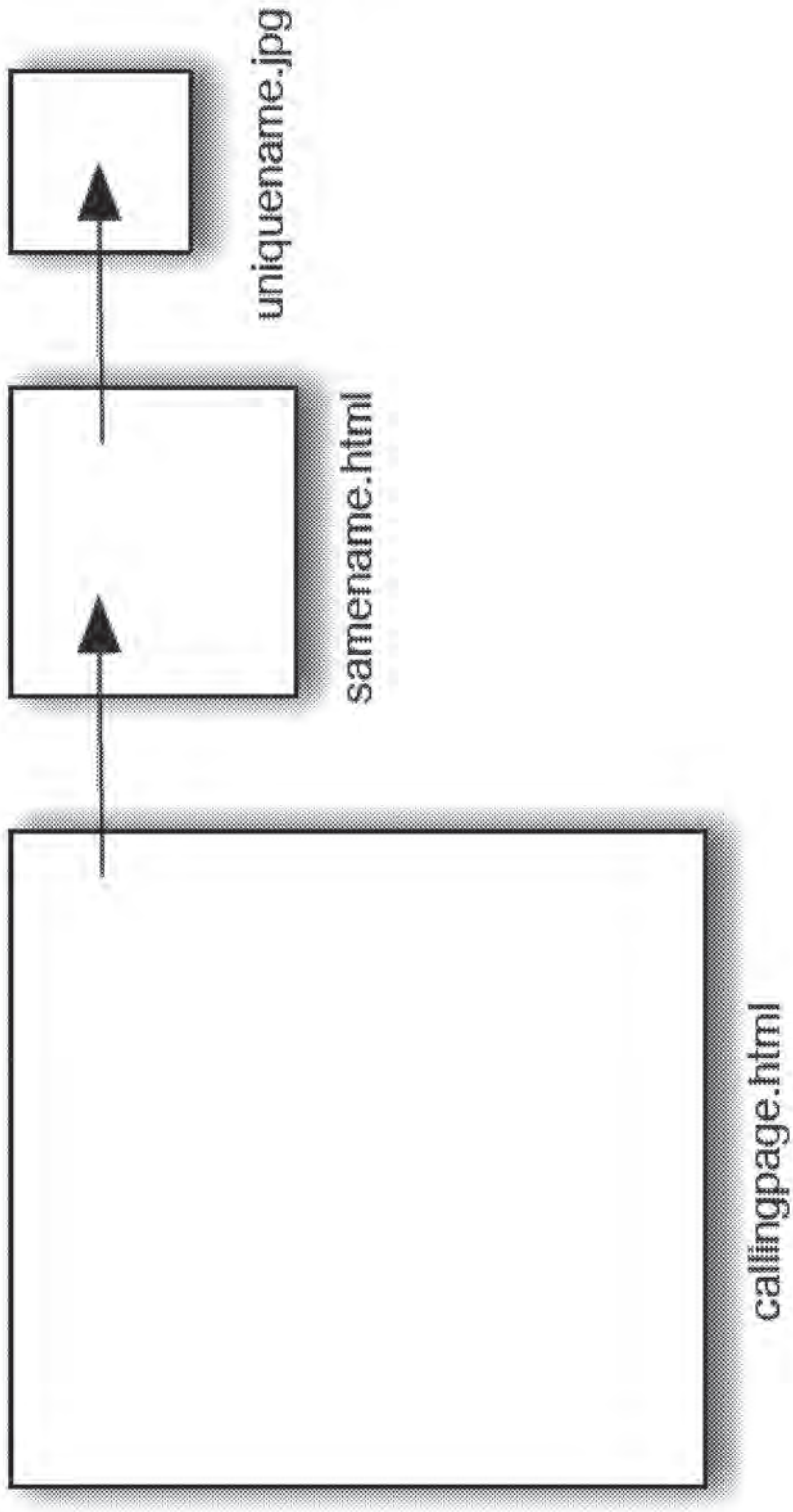


FIG. 8

AUTHENTICATION TRANSLATION

CROSS REFERENCE TO OTHER APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 16/773,767, entitled AUTHENTICATION TRANSLATION filed Jan. 27, 2020 which is incorporated herein by reference for all purposes, which is a continuation of U.S. patent application Ser. No. 16/563,715, entitled AUTHENTICATION TRANSLATION filed Sep. 6, 2019 which is incorporated herein by reference for all purposes, which is a is a continuation of U.S. patent application Ser. No. 16/273,797, entitled AUTHENTICATION TRANSLATION filed Feb. 12, 2019, now U.S. Pat. No. 10,521,568, which is incorporated herein by reference for all purposes, which is a is a continuation of U.S. patent application Ser. No. 15/042,636, entitled AUTHENTICATION TRANSLATION filed Feb. 12, 2016, now U.S. Pat. No. 10,360,351, which is incorporated herein by reference for all purposes, which is a continuation of U.S. patent application Ser. No. 13/706,254, entitled AUTHENTICATION TRANSLATION filed Dec. 5, 2012, now U.S. Pat. No. 9,294,452, which is incorporated herein by reference for all purposes, which claims priority to U.S. Provisional Application No. 61/587,387, entitled BIOMETRICS-SUPPORTED SECURE AUTHENTICATION SYSTEM filed Jan. 17, 2012 which is incorporated herein by reference for all purposes. U.S. patent application Ser. No. 13/706,254 also claims priority to U.S. Provisional Patent Application No. 61/569,112 entitled BACKWARDS COMPATIBLE ROBUST COOKIES filed Dec. 9, 2011, which is incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

Providing credentials to a service, whether via a mobile or other device, is often a tedious experience for a user. Unfortunately, to make authentication easier for themselves, users will often engage in practices such as password re-use, and/or the selection of poor quality passwords, which render their credentials less secure against attacks. Accordingly, improvements in authentication techniques would be desirable. Further, it would be desirable for such improvements to be widely deployable, including on existing/legacy systems.

BRIEF DESCRIPTION OF THE DRAWINGS

Various embodiments of the invention are disclosed in the following detailed description and the accompanying drawings.

FIG. **1** illustrates an embodiment of an environment in which authentication translation is provided.

FIG. **2** illustrates an embodiment of credential information stored on a device.

FIG. **3** illustrates an embodiment of a device with secure storage.

FIG. **4** illustrates an example of a renegotiation.

FIG. **5** illustrates an embodiment of a process for performing authentication translation.

FIG. **6** illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator.

FIG. **7** illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator.

FIG. **8** shows the structure of an example of a cache cookie used in some embodiments.

DETAILED DESCRIPTION

The invention can be implemented in numerous ways, including as a process; an apparatus; a system; a composition of matter; a computer program product embodied on a computer readable storage medium; and/or a processor, such as a processor configured to execute instructions stored on and/or provided by a memory coupled to the processor. In this specification, these implementations, or any other form that the invention may take, may be referred to as techniques. In general, the order of the steps of disclosed processes may be altered within the scope of the invention. Unless stated otherwise, a component such as a processor or a memory described as being configured to perform a task may be implemented as a general component that is temporarily configured to perform the task at a given time or a specific component that is manufactured to perform the task. As used herein, the term 'processor' refers to one or more devices, circuits, and/or processing cores configured to process data, such as computer program instructions.

A detailed description of one or more embodiments of the invention is provided below along with accompanying figures that illustrate the principles of the invention. The invention is described in connection with such embodiments, but the invention is not limited to any embodiment. The scope of the invention is limited only by the claims and the invention encompasses numerous alternatives, modifications and equivalents. Numerous specific details are set forth in the following description in order to provide a thorough understanding of the invention. These details are provided for the purpose of example and the invention may be practiced according to the claims without some or all of these specific details. For the purpose of clarity, technical material that is known in the technical fields related to the invention has not been described in detail so that the invention is not unnecessarily obscured.

FIG. **1** illustrates an embodiment of an environment in which authentication translation is provided. In the example shown, a variety of client devices **102-108** connect, via one or more networks (represented as a single network cloud **110**) to a variety of services **120-124** (also referred to herein as sites **120-124**). In particular, client device **102** is a notebook computer owned by a user hereinafter referred to as Alice. Notebook **102** includes a camera, a microphone, and a fingerprint sensor. Client device **104** is a smartphone, also owned by Alice. Client device **104** includes a camera. Client device **106** is a tablet owned by Bob, and sometimes used by Bob's son Charlie. Client device **106** includes a camera and a fingerprint sensor. Client device **108** is a kiosk located in the lobby of a hotel. Kiosk **108** includes a camera and a microphone. The techniques described herein can be used with or adapted to be used with other devices, as applicable. For example, the techniques can be used in conjunction with gaming systems, with peripheral devices such as mice, and with embedded devices, such as door locks.

Service **120** is a social networking site. Service **122** is a website of a bank. Service **124** is the online store of a boutique camera retailer. Each of services **120-124** requires a username and password (and/or a cookie) from a user prior to giving that user access to protected content and/or other features. As will be described in more detail below, using the techniques described herein, users need not type such usernames and passwords into their devices whenever required

by a service. Instead, users can authenticate themselves to an "authentication translator" via an appropriate technique, and the authentication translator will provide the appropriate credentials to the implicated service on the user's behalf. Also as will be described in more detail below, authentication translators can be located in a variety of places within an environment. For example, notebook computer **102** includes an authentication translator module **132** that provides authentication translation services. The other devices **104-108** can also include (but need not include) their own respective authentication translator modules. The owner of bank website **122** also operates an authentication translator **134** associated with the bank. Finally, authentication translator **136** provides authentication translation services to a variety of businesses, including online camera retailer **124**.

FIG. **2** illustrates an embodiment of credential information stored on a device. In particular, device **200** stores three user profiles **202-206**, each of which contains a username and one or more templates (e.g., template **210**) associated with the user. In various embodiments, a template is a collection of biometric features. Using fingerprints as an example type of biometric, a corresponding template includes a collection of patterns, minutia, and/or other features that can be matched against to determine if a person's fingerprint matches the fingerprint of the registered user (i.e., the owner of a given user profile). A representation of a single fingerprint may be included in multiple templates (e.g., in different resolutions, in accordance with different protocols, as captured during warm vs. cold conditions, and/or by itself or in combination with multiple fingerprints). When other biometrics are employed (e.g., facial recognition, voiceprint, or retina scan technology), features appropriate to those types of biometrics are included in the template. Other types of features can also be included in templates. As one example, a user's typing speed and/or accuracy can be measured by a device, such as device **102**, and used to distinguish between multiple users of a device. For example, suppose Alice types at 100 words per minute and rarely makes mistakes. A representation of this information can be stored in template **212**. Also suppose Alice's niece, who sometimes uses Alice's laptop computer when visiting Alice types at 20 words per minute and makes many mistakes. In some embodiments, the fact that a user was recently (e.g., within the last 5 minutes) typing on laptop **102** at 90 words per minute is evidence of a match against template **212**. In this case, the typing speed of 90 words per minute is similar enough to Alice's typical behavior, it is considered a match. Various policies can be included in a profile that govern how matches are to be performed. For example, policies can specify thresholds/tolerances for what constitutes a match, and can specify that different levels of matches can result in different levels of access to different resources.

A profile is associated with a vault (e.g., vault **220**). The vault, in turn, contains triples specifying a service provider/domain, a username, and a credential. The vault can also contain other sensitive user information, such as account numbers, address/phone number information, and health care data. The credential for a service provider/domain can be a password (e.g., for legacy servers), and can also take alternate forms (e.g., a cryptographic key for service providers supporting stronger authentication methods).

In some embodiments, profiles, templates, and vaults (collectively "authentication information") are stored entirely in an unprotected storage area, and are stored in the clear. In other embodiments, secure storage techniques are used to secure at least a portion of the authentication information.

One example of a device with secure storage is illustrated in FIG. **3**. In the example shown, a mobile phone **300** includes a large and insecure storage **302** attached to a fast processor **304**, and a smaller but secure storage **306** attached to a dedicated processor **308** and a sensor **310** (e.g., a camera or a fingerprint reader). Users (and applications) can read from and write to the insecure storage area. However, users cannot access the secure storage area, and the fast processor can only communicate with the dedicated processor/sensor via a restricted API. As another example, a unique decryption key associated with a given vault can be stored in a profile. The vault is an encrypted and authenticated container that can be stored on insecure storage, e.g., on the device, and also backed up (e.g., to a cloud storage service **140** or to an alternate form of external storage). As needed, authentication information or portions thereof can be loaded into secure storage and decrypted. For example, one can use AES to encrypt the files one by one, using a key stored on the secured storage. A message authentication technique, such as HMAC, can be used for authenticating the encrypted files to provide tamper prevention. Profiles and vaults can be updated while in secure storage; if this occurs, they are encrypted and MACed before being written back to the insecure storage, which may in turn propagate them to external backup storage. In yet other embodiments, profiles and vaults are stored entirely in secure storage, in plaintext, which allows them to be both read and written—and in particular, searched.

Example Transaction Types

A variety of transaction types can take place in the environment shown in FIG. **1**, examples of which are discussed in this section.

Initial Registration

In order to begin using the techniques described herein, users perform some form of initial registration. As one example, suppose Alice launches an enrollment program installed on laptop **102**. She uses the program to capture various biometric information (e.g., fingerprints, photographs of her face, etc.). A user profile is created for Alice, and the biometric information captured about her is encoded into a plurality of templates, such as templates **210** and **214**. In some embodiments, Alice is also explicitly asked to supply credential information for services she would like to use, such as by providing the domain name of social networking site **120**, along with her username and password for site **120**. In other embodiments, domain/username/credential information is at least passively captured on Alice's behalf and included in one or more vaults such as vault **220**. Credential information can also be important from a browser password manager already in use by Alice or other appropriate source. In some embodiments, Alice also registers with cloud storage service **140**, which will allow her to back up her authentication information and to synchronize it across her devices (e.g., **102** and **104**), as described in more detail below.

Other registration approaches can also be used. For example, registration can be integrated into the experience the first time a device is used. Thus, when Bob first turns on tablet **106**, he may be prompted to take a picture of his face (with a profile/templates being created in response). Simi-

larly, the first time Charlie uses tablet **106**, the techniques described herein can be used to determine that Charlie does not yet have a profile (e.g., because none of the templates already present on tablet **106** match his biometrics) and Charlie can be prompted to enroll as a second user of the device.

Authentication

Suppose Alice wishes to authenticate to banking website **122**. Using a fingerprint reader incorporated into her laptop, she performs a fingerprint scan, which causes her biometric features to be extracted and compared to any stored templates residing on her computer. If a match is found, an associated decryption key is selected, and the associated vault is loaded and decrypted. The vault is scanned for an entry that matches the selected service provider (i.e., website **122**). If a matching entry is found, the associated domain, username, and site credential are extracted from the vault. In some embodiments, the validity of the domain name mapping is verified at this point to harden the system against domain name poisoning. Next, a secure connection is established between Alice's computer and the service provider, and Alice is authenticated. For service providers supporting strong user authentication, mutual SSL can be used, for example. A variety of policies can be involved when performing matching. For example, to access certain domains, Alice's print may need only match template **210**. To access other domains, Alice may need to match multiple templates (e.g., both **210** and **214**). As another example, in order to access social networking site **120**, Alice may merely need to be sitting in front of her computer, which has an integrated webcam. Even in relatively low light conditions, a match can be performed against Alice's face and features stored in a template. However, in order to access bank website **122**, Alice may need a high quality photograph (i.e., requiring her to turn on a bright light) and may need to demonstrate liveness (e.g., by blinking or turning her head). As yet another example, other contextual information can be included in policies. For example, if Alice's IP address indicates she is in a country that she is not usually in, she may be required to match multiple templates (or match a template with more/better quality features) in order to access retailer **124**, as distinguished from when her IP address indicates she is at home.

In some embodiments, the biometric sensor used by a user may be a peripheral device (e.g., a mouse with an integrated fingerprint scanner that is connected to the user's primary device via USB). In such scenarios, the peripheral device may be responsible for storing at least a portion of authentication information and may perform at least some of the authentication tasks previously described as having been performed by Alice's computer. For example, instead of processors **304** and **308**, and storages **302** and **306** being collocated on a single device (e.g., laptop **102**), processor **304** and storage **302** may be present on a primary device, and processor **308** and storage **306** may be present on a peripheral device (e.g., that also includes a sensor, such as a fingerprint reader).

In such scenarios, once Alice's login to banking website **122** is successfully completed, the secure session can be handed over from the peripheral device to the primary device, in a way that does not allow the primary device retroactive access to the plaintext data of the transcripts exchanged between the peripheral device and the service provider. One way this can be accomplished is by renegotiating SSL keys between the peripheral device and the website, after which the newly negotiated key can be handed off from the peripheral device to the primary device. This avoids retroactive credential capture in a setting where the device is infected by malware.

An example of renegotiation is depicted in FIG. **4**. Specifically, after a user has successfully authenticated to a fingerprint reader, a login is performed to a service provider. Using the primary device (**404**) as a proxy, the peripheral fingerprint reader **402** negotiates a first SSL connection (**408**) with a service provider **406**, over which credentials are exchanged. The proxy then renegotiates SSL (**410**), which replaces the old key with a new one. The new key is disclosed to the device, which then seamlessly takes over the connection with the service provider and performs the transaction protected by the authentication. The credentials exchanged during the first SSL connection cannot be accessed by device **404**, since the key of the renegotiated session is independent of the key of the original session; this provides protection against malware residing on the device. Renegotiation can be used when the primary device **404** is believed to be in a safe state when performing the negotiation of the SSL connection, but it is not known whether it is in a safe state during the transaction protected by the authentication. Renegotiation can also be used when a secure component of the primary device **404** performs the negotiation of the SSL connection and another and potentially insecure component of the primary device **404** is involved in the transaction protected by the authentication.

FIG. **5** illustrates an embodiment of a process for performing authentication translation. The process begins at **502** when a request to access a resource is received, as is an authentication input. One example of the processing performed at **502** is as follows. Suppose Alice wishes to sign into social networking website **120**. She directs a web browser application installed on client **102** to the social networking website. Authentication translator module **132** recognizes, from the context of Alice's actions (e.g., that she is attempting to access site **120** with her browser) that she would like to access a particular resource. Authentication translator module **132** prompts Alice (e.g., by a popup message or via a sound) to provide biometric information (e.g., to use the integrated fingerprint reader on her laptop). In some embodiments, the translator module does not prompt Alice, for example, because Alice has been trained to provide biometric information automatically when attempting to access certain resources. In yet other embodiments, the translator module only prompts Alice if she fails to provide acceptable biometric information within a timeout period (e.g., 30 seconds).

Module **132** compares Alice's supplied biometric data to the templates stored on her computer. If a suitable match is found, and if an entry for site **120** is present in the applicable vault, at **504**, a previously stored credential associated with the resource is accessed. In particular, the username and password for the website, as stored in a vault, such as vault **220**, are retrieved from the vault.

Finally, at **506**, the credential is provided to the resource. For example, Alice's username and password for site **120** are transmitted to site **120** at **506**. The credential can be transmitted directly (e.g., by the module or by Alice's computer) and can also be supplied indirectly (e.g., through the use of one or more proxies, routers, or other intermediaries, as applicable).

Other devices can also make use of process **500** or portions thereof. For example, when Alice launches a banking application on phone **104**, implicit in her opening that application is her desire to access the resources of website

134. The application can take Alice's picture and compare it to stored templates/vault information. If an appropriate match is found, a credential can be retrieved from the vault on her phone (or, e.g., retrieved from cloud storage service **140**) and provided to website **134**.

As another example, suppose Charlie is using tablet **106** and attempts to visit site **120**, whether via a dedicated application or via a web browser application installed on the tablet. Charlie's photograph is taken, and then compared against the profiles stored on tablet **106** (e.g., both Bob and Charlie's profiles). When a determination is made that Charlie's photograph matches a template stored in his stored profile (and not, e.g., Bob's), Charlie's credentials for site **120** are retrieved from a vault and transmitted by an authentication translator module residing on client **106**.

As yet another example, kiosk **108** can be configured to provide certain local resources (e.g., by displaying a company directory or floor plan on demand) when users speak certain requests into a microphone. Enrolled users (e.g., with stored voiceprint or facial recognition features) can be granted access to additional/otherwise restricted services in accordance with the techniques described herein and process **500**.

New device

In some embodiments, to register a new device, a user provides an identifier, such as a username or an account number to the device. The new device connects to an external storage (such as cloud storage **140**), provides the user identifier and credential, and downloads the user's templates/vaults from the service. In some embodiments, the templates/vaults are encrypted. Once downloaded, the template is decrypted and stored in a secure storage area, while the still encrypted vault can be stored in insecure storage. The decryption key can be generated from information the user has/knows, or from biometric data—such as features extracted from fingerprinting of all ten fingers. In some embodiments, more arduous fingerprinting is required for the setup of a new device than for regular authentication to avoid that a new device gets registered by a user thinking she is merely authenticating—or worse still, simply touching the device. Moreover, it translates into higher entropy of the decryption keys.

Backup Authentication

Backup authentication allows a user, such as Alice, to access resources in the event she is unable to or unwilling to interact with a particular biometric sensor. As one example, instead of having a single template associated with her profile, Alice can have multiple templates associated with it, e.g., where the first template includes fingerprint features and the second template includes voice biometric, facial recognition, or iris detection features. As a second example, where the service Alice is connecting to is a legacy website (i.e., one that users authenticate to using usernames and passwords), such a service would allow the use of passwords and password reset mechanisms by Alice without requiring Alice to use a fingerprint reader.

In various embodiments, environment **100** supports the ability of users (e.g., under duress) to release the contents of their vaults. For example, if Alice was physically threatened with the loss of a finger by a criminal, Alice could instead release the contents of her vault(s)—the ultimate goal of the criminal. As one example, in the event Alice supplies all 10 fingerprints to the sensor, provides a special password, or supplies a fingerprint and a second identifier, a cleartext version of her vault(s) could be made available.

Access Policies

In various embodiments, cloud storage service **140** is configured to accept backups from multiple devices associated with a single account, and synchronize the updates so that all devices get automatically refreshed. For example, Alice's laptop **102** and phone **104** could both communicate with cloud storage service **140** which would keep their authentication information synchronized. Refreshes can also be made in accordance with user-configured restrictions. For example, Alice's employer could prevent privileged employer data from being stored on shared personal devices, or on any device that was not issued by the employer. As another example, arbitrary policies can be defined regarding the access to and synchronization of software and data, and to tie a license or access rights to a person (and her fingerprint) rather than to a device. As yet another example, in some embodiments (e.g., where a device is made publicly available or otherwise shared by many users), no or a reduced amount of authentication information resides on a device, and at least a portion of authentication information is always retrieved from cloud storage service **140**.

Remote wiping

Remote wiping of a user's authentication information (e.g., templates) can be used both to "unshare" previously shared devices (e.g., where Bob and Charlie both have user profiles on their shared tablet **106**), and to avoid that criminals with physical component access to lost devices gain access to templates and vault contents. In some embodiments, polices such as ones where a template self-wipes if it is not matched within a particular duration of time are supported. Since user data can be frequently backed up to the cloud storage, and recovered from this using the new device registration process, inconvenience to the user will be minimized.

Legacy Server Support

New authentication schemes typically require changes to a significant codebase residing with service providers. If the code is well written and documented, such changes may be relatively simple. Commonly, though, this may not be so. The engineers who originally wrote the code of relevance may have long since left the company; the code they left behind may be poorly documented—if documented at all. In severe cases, the legacy code may have been written in an outdated programming language or written in a way that does not follow guidelines for good code. This makes updates to the codebase impractical or virtually impossible in many common cases. Even if none of these challenges complicate the desired modifications, it is commonly a great bureaucratic burden to obtain permission to make modifications (e.g., to store an extra field in a backend database), because every affected part of the organization may need to review the request.

As will be described in the following section, the technologies described herein can be used in conjunction with legacy servers (e.g., existing servers that rely on usernames and passwords to authenticate users), and in particular, can be used without requiring modification to such legacy servers.

Cookies

Cookies are commonly used by legacy servers for user authentication. Unfortunately, cookies have several problems. For one thing, they are sometimes deleted—whether explicitly/intentionally by the end user or by the user's software. In addition, cookies are commonly stolen. Approaches such as cache cookies and identification using user agents can be more resistant to these problems, however, they have their own problems. For example, their use requires new code and new fields in the credential database stored by the server.

In some embodiments, authentication translators, such as translators **134** and **136** (also referred to herein as proxies) provide authentication translation services on behalf of associated services. Translators **134** and **136** are illustrated as single logical devices in FIG. **1**. In some embodiments, the translators comprise standard commercially available server hardware (e.g., a multi-core processor, **4+** Gigabytes of RAM, and one or more Gigabit network interface adapters) and run typical server-class operating systems (e.g., Linux). Translators **134** and **136** can also be implemented using a scalable, elastic architecture and may comprise several distributed components, including components provided by one or more third parties. For example, translators **134** and **136** may store user credential information or may task cloud storage service **140** with storing at least a portion of that information.

In the case of authentication translator **134**, service is provided with respect to bank website **122** only. Authentication translator **134** is positioned between a legacy web server (**122**) and the Internet (**110**)—and therefore between the legacy server and any client devices. Authentication translator **134** is configured to translate traffic between the legacy server and client devices so that the client devices (and respective users) perceive the new authentication mechanism, while the legacy server remains unchanged. Authentication translator **136** works similarly, but it provides authentication translation as a third party service to multiple providers, an example of which is online camera retailer **124**.

Authentication translators **134** and **136** can also perform process **500**. For example, when a device transmits a request to access website **122**, the request is intercepted by translator **134**, as is cookie/user agent information. The received information can be used to determine a username/password associated with the device, and that information can be passed by translator **134** to website **122** on behalf of the device.

FIG. **6** illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator. The translator (referred to in the figure as a "proxy") fails to identify the client, and passes on the request to the legacy server. The legacy server responds to the request and sets a cookie. The proxy passes on the response, including the cookie and also a cache cookie. The proxy stores the information about both these types of identifiers, along with the user agent of the client device. This triplet of information is also referred to herein as an identifier.

FIG. **7** illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator. In this scenario, the device's request is accompanied by some form of identifying information. The proxy uses this information to identify the associated cookie and passes this along to the legacy server, along with the request. In some embodiments, additional processing is also involved. For example, reading a cache cookie may require user interaction. Moreover, if not all of the identifying information is present in the request, the proxy can be configured to set the missing information again by sending a corresponding request to the client device.

The translation of cache cookies and user agent information to cookies involves a two-way translation. First, when the legacy server sets a cookie, the proxy will set the two types of cookies—both an HTML cookie and a cache cookie—and then create a new record in which the two cookies are stored, along with the user agent information of the client device. The user agent information can include quite a bit of data associated with a browser—such as the browser type and version, the clock skew, and the fonts that are installed. While each of these pieces of information only contributes a small amount of entropy, the collection of items can be sufficient to identify the device in most cases. Moreover, while some of these types of data may change over time—in fact, all of them may—they do not typically change, and when one or two of them do, the others typically do not. When the client device is used to visit a site controlled by the legacy server, the cookie, cache cookie and user agent information are read (if available), the record identified, and the request translated and sent to the legacy server. When a legacy server requests that the user password is updated (e.g., as part of an annual or other periodic requirement), the transmission of this request to the user can be suppressed—in which case the database of the proxy is updated to create the illusion of an update. The user can be involved in authentication as needed, e.g., if, in addition to supplying a credential, a user must also solve a CAPTCHA, the CAPTCHA can be displayed to the user (with the user's credentials being handled in accordance with the techniques described herein).

FIG. **8** shows the structure of an example of a cache cookie used in some embodiments. Cache cookies can be associated with a particular webpage, just like an HTML cookie can. In the example shown in FIG. **8**, the proxy wishes to associate a page "calling page.html" with a cache cookie. It embeds a request for a second object, "samename.html" in callingpage.html for every visitor. However, as the cache cookie is set for one visitor, a customized samename.html is served to this visitor. The page samename.html refers to an object with a different name for each user; that object is referred to as "uniquename.jpg." The cache cookie is set by embedding the request for samename.html in callingpage.html. The client browser attempts to render this, causing a request for samename.html from the server. The server configures samename.html to refer to a uniquely named file uniquename.jpg, and serves samename.html to the client. For the client browser to render samename.html, it requests the file uniquename.jpg, which is intentionally not served. That concludes the setting of a cache cookie. As a user returns to the page callingname.html, the browser again attempts to render the entire page, which causes it to load the object samename.html from its cache. As that is rendered, the client browser requests uniquename.jpg, which is not in its cache (since it was not served previously). The server still does not serve it, but takes note of the name of the file being requested, as it identifies the client device. Note that samename.html can be displayed in a zero-sized iframe, which makes the end user unaware of it being rendered.

A cache cookie is an implementation of the typical cookie functionality that uses the client device's browser cache. Unlike user agents, it does not change over time, and like standard HTML cookies, it cannot be read by a party other

than that which set it. However, like HTML cookies, it could be deleted—by the user clearing his or her browser cache. Cache cookies are not automatically transmitted with GET requests, unless the cache elements are embedded in the referring pages. This adds a potential round of communication in some settings. By relying on user agent information, cache cookies, and HTML cookies to identify the client device, it is much more likely that a machine will be recognized than if only HTML cookies are used.

Although the foregoing embodiments have been described in some detail for purposes of clarity of understanding, the invention is not limited to the details provided. There are many alternative ways of implementing the invention. The disclosed embodiments are illustrative and not restrictive.

What is claimed is:

1. A device, comprising:

a processor on the device;

a storage on the device that is accessible by the processor on the device via an interface that facilitates communication between the storage on the device and the processor on the device, wherein the processor on the device is configured to:

based at least in part on a request from a user to access an external resource, facilitate, using the interface, access of at least one record stored at least in part in the storage on the device;

wherein the at least one record is associated at least with the external resource, and wherein the at least one record comprises:

a biometric template; and

a credential comprising a cryptographic key;

in response to determining a match between a biometric input and the biometric template, retrieve, from the at least one record, the credential, wherein the biometric input corresponds to at least one of a fingerprint, a feature usable for facial recognition, a voiceprint, a feature usable for a retina scan, or a typing feature, wherein the biometric input is received subsequent to presenting of a prompt, wherein the prompt comprises a prompt to provide biometric information, and wherein the prompt is visually presented;

establish a connection with the external resource;

facilitate a login of the user to the external resource at least in part by transmitting, via the established connection, output based at least in part on the credential retrieved from the at least one record, and wherein the user is logged in to the external resource based at least in part on the output; and

facilitate wiping of both the biometric template and the credential comprising the cryptographic key of the at least one record stored at least in part in the storage on the device; and

a memory coupled to the processor and configured to provide the processor with instructions.

2. The device recited in claim **1** wherein the storage is connected to a sensor.

3. The device recited in claim **2** wherein the sensor comprises at least one of a camera and a fingerprint reader.

4. The device recited in claim **1** wherein at least some of the at least one record is stored in plaintext in the storage.

5. The device recited in claim **1** wherein the prompt is aurally presented.

6. The device recited in claim **1** wherein the prompt is presented in response to a user failing to provide acceptable biometric information within a timeout period.

7. The device recited in claim **1** wherein facilitating wiping of the at least portion of the at least one record comprises facilitating remote wiping of the at least portion of the at least one record.

8. The device recited in claim **1** wherein the at least portion of the at least one record is automatically wiped based at least in part on a policy.

9. The device recited in claim **1** wherein the biometric template is wiped in response to determining that the biometric template has not been matched within a duration of time.

10. The device recited in claim **1** wherein the at least portion of the at least one record is backed up to a remote entity.

11. The device recited in claim **1**, wherein the interface comprises a restricted interface.

12. The device recited in claim **1**, wherein the storage comprises a secure storage.

13. A method, comprising:

based at least in part on a request from a user to access an external resource, using a processor on a device to facilitate access of at least one record stored at least in part in a storage on the device, wherein the storage on the device is accessible by the processor on the device via an interface that facilitates communication between the storage on the device and the processor on the device;

wherein the at least one record is associated at least with the external resource, and wherein the at least one record comprises:

a biometric template; and

a credential comprising a cryptographic key;

in response to determining a match between a biometric input and the biometric template, retrieving, from the at least one record, the credential, wherein the biometric input corresponds to at least one of a fingerprint, a feature usable for facial recognition, a voiceprint, a feature usable for a retina scan, or a typing feature, wherein the biometric input is received subsequent to presenting of a prompt, wherein the prompt comprises a prompt to provide biometric information, and wherein the prompt is visually presented;

establishing a connection with the external resource;

facilitating a login of the user to the external resource at least in part by transmitting, via the established connection, output based at least in part on the credential retrieved from the at least one record, and wherein the user is logged in to the external resource based at least in part on the output; and

facilitating, by the processor on the device, wiping of both the biometric template and the credential comprising the cryptographic key of the at least one record stored at least in part in the storage on the device.

14. The method of claim **13** wherein the storage is connected to a sensor.

15. The method of claim **14** wherein the sensor comprises at least one of a camera and a fingerprint reader.

16. The method of claim **13** wherein at least some of the at least one record is stored in plaintext in the storage.

17. The method of claim **13** wherein the prompt is aurally presented.

18. The method of claim **13** wherein the prompt is presented in response to a user failing to provide acceptable biometric information within a timeout period.

19. The method of claim **12** wherein facilitating wiping of the at least portion of the at least one record comprises facilitating remote wiping of the at least portion of the at least one record.

20. The method of claim **13** wherein the at least portion of the record is automatically wiped based at least in part on a policy.

21. The method of claim **13** wherein the biometric template is wiped in response to determining that the biometric template has not been matched within a duration of time.

22. The method of claim **13** wherein the at least portion of the at least one record is backed up to a remote entity.

23. The method of claim **13**, wherein the interface comprises a restricted interface.

24. The method of claim **13**, wherein the storage comprises a secure storage.

25. A computer program product embodied in a non-transitory computer readable storage medium and comprising computer instructions for:

based at least in part on a request from a user to access an external resource, using a processor on a device to facilitate access of at least one record stored at least in part in a storage on the device, wherein the storage on the device is accessible by the processor on the device via an interface that facilitates communication between the storage on the device and the processor on the device;

wherein the at least one record is associated at least with the external resource, and wherein the at least one record comprises:

a biometric template; and

a credential comprising a cryptographic key;

in response to determining a match between a biometric input and the biometric template, retrieving, from the at least one record, the credential, wherein the biometric input corresponds to at least one of a fingerprint, a feature usable for facial recognition, a voiceprint, a feature usable for a retina scan, or a typing feature, wherein the biometric input is received subsequent to presenting of a prompt, wherein the prompt comprises a prompt to provide biometric information, and wherein the prompt is visually presented;

establishing a connection with the external resource;

facilitating a login of the user to the external resource at least in part by transmitting, via the established connection, output based at least in part on the credential retrieved from the at least one record, and wherein the user is logged in to the external resource based at least in part on the output; and

facilitating, by the processor on the device, wiping of both the biometric template and the credential comprising the cryptographic key of the at least one record stored at least in part in the storage on the device.

* * * * *

Electronic Acknowledgement Receipt	
EFS ID:	40614418
Application Number:	17027481
International Application Number:	
Confirmation Number:	7336
Title of Invention:	AUTHENTICATION TRANSLATION
First Named Inventor/Applicant Name:	Bjorn Markus Jakobsson
Customer Number:	21912
Filer:	Yeu-Ting George Cheng/Elaine Nguyen
Filer Authorized By:	Yeu-Ting George Cheng
Attorney Docket Number:	MJAKP008C5
Receipt Date:	21-SEP-2020
Filing Date:	
Time Stamp:	18:57:40
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	DA
Payment was successfully received in RAM	$935
RAM confirmation Number	E20209KI58153757
Deposit Account	500685
Authorized User	Elaine Nguyen
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: 37 CFR 1.16 (National application filing, search, and examination fees) 37 CFR 1.17 (Patent application and reexamination processing fees)	

37 CFR 1.19 (Document supply fees)

37 CFR 1.20 (Post Issuance fees)

37 CFR 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Application Data Sheet	MJAKP008C5_ADS.pdf	1793984 72fdc6f2ed3c0n7899ad591d55587e8(23f)c2ec	no	9
Warnings:					
Information:					
2	Oath or Declaration filed	MJAKP008C5_Executed_Dec.pdf	93450 8aab681ee91b3a52fd972ce59278630ed97ed0e	no	1
Warnings:					
Information:					
3	Power of Attorney	MJAKP008C5_POA_AIA82A.pdf	228276 7ecd8e76e2c2130705bd1337f5d1e67ab18d9e66	no	1
Warnings:					
Information:					
4	Power of Attorney	MJAK_Executed_POA_AIA82B_RightQuestionLLC.pdf	387899 a756b6df61aa99188ddcc71473ee8bc19a8[illegible]	no	1
Warnings:					
Information:					
5	Transmittal Letter	MJAKP008C5_IDS_01_Transmittal.pdf	77904 d01b71fb1f80d28e6645f172c34e84ce761d8b2c	no	2
Warnings:					
Information:					
6	Information Disclosure Statement (IDS) Form (SB08)	MJAKP008C5_IDS_01_SB08.pdf	1054194 d002b3fcff328e5d0d3f42ee3f250d67562dae0f	no	6
Warnings:					
Information:					

7	Specification	MJAKP008C5_APP.pdf	163211 [illegible]	no	22
Warnings:					
Information:					
8	Drawings-only black and white line drawings	MJAKP008C5_APP_Figures.pdf	112090 [illegible]	no	8
Warnings:					
Information:					
9	Fee Worksheet (SB06)	fee-info.pdf	36745 [illegible]	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	3947753		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.
National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.
New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Attorney Docket No. MJAKP008C5

APPLICATION FOR UNITED STATES PATENT

AUTHENTICATION TRANSLATION

By Inventor:

Bjorn Markus Jakobsson
Portola Valley, CA
A Citizen of Sweden

Assignee: RightQuestion, LLC

VAN PELT, YI & JAMES LLP
10050 N. Foothill Blvd., Suite 200
Cupertino, CA 95014
Telephone (408) 973-2585

AUTHENTICATION TRANSLATION

CROSS REFERENCE TO OTHER APPLICATIONS

[0001] This application is a continuation of U.S. Patent Application No. 16/773,767, entitled AUTHENTICATION TRANSLATION filed January 27, 2020 which is incorporated herein by reference for all purposes, which is a continuation of U.S. Patent Application No. 16/563,715, entitled AUTHENTICATION TRANSLATION filed September 06, 2019 which is incorporated herein by reference for all purposes, which is a is a continuation of U.S. Patent Application No. 16/273,797, entitled AUTHENTICATION TRANSLATION filed February 12, 2019, now U.S. Patent No. 10,521,568, which is incorporated herein by reference for all purposes, which is a is a continuation of U.S. Patent Application No. 15/042,636, entitled AUTHENTICATION TRANSLATION filed February 12, 2016, now U.S. Patent No. 10,360,351, which is incorporated herein by reference for all purposes, which is a continuation of U.S. Patent Application No. 13/706,254, entitled AUTHENTICATION TRANSLATION filed December 5, 2012, now U.S. Patent No. 9,294,452, which is incorporated herein by reference for all purposes, which claims priority to U.S. Provisional Application No. 61/587,387, entitled BIOMETRICS-SUPPORTED SECURE AUTHENTICATION SYSTEM filed January 17, 2012 which is incorporated herein by reference for all purposes. U.S. Patent Application No. 13/706,254 also claims priority to U.S. Provisional Patent Application No. 61/569,112 entitled BACKWARDS COMPATIBLE ROBUST COOKIES filed December 9, 2011, which is incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

[0002] Providing credentials to a service, whether via a mobile or other device, is often a tedious experience for a user. Unfortunately, to make authentication easier for themselves, users will often engage in practices such as password re-use, and/or the selection of poor quality passwords, which render their credentials less secure against attacks. Accordingly, improvements in authentication techniques would be desirable. Further, it would be desirable for such improvements to be widely deployable, including on existing/legacy systems.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] Various embodiments of the invention are disclosed in the following detailed description and the accompanying drawings.

[0004] Figure 1 illustrates an embodiment of an environment in which authentication translation is provided.

[0005] Figure 2 illustrates an embodiment of credential information stored on a device.

[0006] Figure 3 illustrates an embodiment of a device with secure storage.

[0007] Figure 4 illustrates an example of a renegotiation.

[0008] Figure 5 illustrates an embodiment of a process for performing authentication translation.

[0009] Figure 6 illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator.

[0010] Figure 7 illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator.

[0011] Figure 8 shows the structure of an example of a cache cookie used in some embodiments.

DETAILED DESCRIPTION

[0012] The invention can be implemented in numerous ways, including as a process; an apparatus; a system; a composition of matter; a computer program product embodied on a computer readable storage medium; and/or a processor, such as a processor configured to execute instructions stored on and/or provided by a memory coupled to the processor. In this specification, these implementations, or any other form that the invention may take, may be referred to as techniques. In general, the order of the steps of disclosed processes may be altered within the scope of the invention. Unless stated otherwise, a component such as a processor or a memory described as being configured to perform a task may be implemented as a general component that is temporarily configured to perform the task at a given time or a specific component that is manufactured to perform the task. As used herein, the term 'processor' refers to one or more devices, circuits, and/or processing cores configured to process data, such as computer program instructions.

[0013] A detailed description of one or more embodiments of the invention is provided below along with accompanying figures that illustrate the principles of the invention. The invention is described in connection with such embodiments, but the invention is not limited to any embodiment. The scope of the invention is limited only by the claims and the invention encompasses numerous alternatives, modifications and equivalents. Numerous specific details are set forth in the following description in order to provide a thorough understanding of the invention. These details are provided for the purpose of example and the invention may be practiced according to the claims without some or all of these specific details. For the purpose of clarity, technical material that is known in the technical fields related to the invention has not been described in detail so that the invention is not unnecessarily obscured.

[0014] Figure 1 illustrates an embodiment of an environment in which authentication translation is provided. In the example shown, a variety of client devices 102-108 connect, via one or more networks (represented as a single network cloud 110) to a variety of services 120-124 (also referred to herein as sites 120-124). In particular, client device 102 is a notebook computer owned by a user hereinafter referred to as Alice. Notebook 102 includes a camera, a microphone, and a fingerprint sensor. Client device 104 is a smartphone, also owned by Alice.

Client device 104 includes a camera. Client device 106 is a tablet owned by Bob, and sometimes used by Bob's son Charlie. Client device 106 includes a camera and a fingerprint sensor. Client device 108 is a kiosk located in the lobby of a hotel. Kiosk 108 includes a camera and a microphone. The techniques described herein can be used with or adapted to be used with other devices, as applicable. For example, the techniques can be used in conjunction with gaming systems, with peripheral devices such as mice, and with embedded devices, such as door locks.

[0015] Service 120 is a social networking site. Service 122 is a website of a bank. Service 124 is the online store of a boutique camera retailer. Each of services 120-124 requires a username and password (and/or a cookie) from a user prior to giving that user access to protected content and/or other features. As will be described in more detail below, using the techniques described herein, users need not type such usernames and passwords into their devices whenever required by a service. Instead, users can authenticate themselves to an "authentication translator" via an appropriate technique, and the authentication translator will provide the appropriate credentials to the implicated service on the user's behalf. Also as will be described in more detail below, authentication translators can be located in a variety of places within an environment. For example, notebook computer 102 includes an authentication translator module 132 that provides authentication translation services. The other devices 104-108 can also include (but need not include) their own respective authentication translator modules. The owner of bank website 122 also operates an authentication translator 134 associated with the bank. Finally, authentication translator 136 provides authentication translation services to a variety of businesses, including online camera retailer 124.

[0016] Figure 2 illustrates an embodiment of credential information stored on a device. In particular, device 200 stores three user profiles 202-206, each of which contains a username and one or more templates (e.g., template 210) associated with the user. In various embodiments, a template is a collection of biometric features. Using fingerprints as an example type of biometric, a corresponding template includes a collection of patterns, minutia, and/or other features that can be matched against to determine if a person's fingerprint matches the fingerprint of the registered user (i.e., the owner of a given user profile). A representation of a single fingerprint may be included in multiple templates (e.g., in different resolutions, in accordance with different protocols, as captured during warm vs. cold conditions, and/or by itself

or in combination with multiple fingerprints). When other biometrics are employed (e.g., facial recognition, voiceprint, or retina scan technology), features appropriate to those types of biometrics are included in the template. Other types of features can also be included in templates. As one example, a user's typing speed and/or accuracy can be measured by a device, such as device 102, and used to distinguish between multiple users of a device. For example, suppose Alice types at 100 words per minute and rarely makes mistakes. A representation of this information can be stored in template 212. Also suppose Alice's niece, who sometimes uses Alice's laptop computer when visiting Alice types at 20 words per minute and makes many mistakes. In some embodiments, the fact that a user was recently (e.g., within the last 5 minutes) typing on laptop 102 at 90 words per minute is evidence of a match against template 212. In this case, the typing speed of 90 words per minute is similar enough to Alice's typical behavior, it is considered a match. Various policies can be included in a profile that govern how matches are to be performed. For example, policies can specify thresholds/tolerances for what constitutes a match, and can specify that different levels of matches can result in different levels of access to different resources.

[0017] A profile is associated with a vault (e.g., vault 220). The vault, in turn, contains triples specifying a service provider/domain, a username, and a credential. The vault can also contain other sensitive user information, such as account numbers, address/phone number information, and health care data. The credential for a service provider/domain can be a password (e.g., for legacy servers), and can also take alternate forms (e.g., a cryptographic key for service providers supporting stronger authentication methods).

[0018] In some embodiments, profiles, templates, and vaults (collectively "authentication information") are stored entirely in an unprotected storage area, and are stored in the clear. In other embodiments, secure storage techniques are used to secure at least a portion of the authentication information.

[0019] One example of a device with secure storage is illustrated in Figure 3. In the example shown, a mobile phone 300 includes a large and insecure storage 302 attached to a fast processor 304, and a smaller but secure storage 306 attached to a dedicated processor 308 and a sensor 310 (e.g., a camera or a fingerprint reader). Users (and applications) can read from and

write to the insecure storage area. However, users cannot access the secure storage area, and the fast processor can only communicate with the dedicated processor/sensor via a restricted API. As another example, a unique decryption key associated with a given vault can be stored in a profile. The vault is an encrypted and authenticated container that can be stored on insecure storage, e.g., on the device, and also backed up (e.g., to a cloud storage service 140 or to an alternate form of external storage). As needed, authentication information or portions thereof can be loaded into secure storage and decrypted. For example, one can use AES to encrypt the files one by one, using a key stored on the secured storage. A message authentication technique, such as HMAC, can be used for authenticating the encrypted files to provide tamper prevention. Profiles and vaults can be updated while in secure storage; if this occurs, they are encrypted and MACed before being written back to the insecure storage, which may in turn propagate them to external backup storage. In yet other embodiments, profiles and vaults are stored entirely in secure storage, in plaintext, which allows them to be both read and written -- and in particular, searched.

[0020] Example Transaction Types

[0021] A variety of transaction types can take place in the environment shown in Figure 1, examples of which are discussed in this section.

[0022] Initial Registration

[0023] In order to begin using the techniques described herein, users perform some form of initial registration. As one example, suppose Alice launches an enrollment program installed on laptop 102. She uses the program to capture various biometric information (e.g., fingerprints, photographs of her face, etc.). A user profile is created for Alice, and the biometric information captured about her is encoded into a plurality of templates, such as templates 210 and 214. In some embodiments, Alice is also explicitly asked to supply credential information for services she would like to use, such as by providing the domain name of social networking site 120, along with her username and password for site 120. In other embodiments, domain/username/credential information is at least passively captured on Alice's behalf and included in one or more vaults such as vault 220. Credential information can also be important from a browser password manager already in use by Alice or other appropriate source. In some

embodiments, Alice also registers with cloud storage service 140, which will allow her to back up her authentication information and to synchronize it across her devices (e.g., 102 and 104), as described in more detail below.

[0024] Other registration approaches can also be used. For example, registration can be integrated into the experience the first time a device is used. Thus, when Bob first turns on tablet 106, he may be prompted to take a picture of his face (with a profile/templates being created in response). Similarly, the first time Charlie uses tablet 106, the techniques described herein can be used to determine that Charlie does not yet have a profile (e.g., because none of the templates already present on tablet 106 match his biometrics) and Charlie can be prompted to enroll as a second user of the device.

[0025] Authentication

[0026] Suppose Alice wishes to authenticate to banking website 122. Using a fingerprint reader incorporated into her laptop, she performs a fingerprint scan, which causes her biometric features to be extracted and compared to any stored templates residing on her computer. If a match is found, an associated decryption key is selected, and the associated vault is loaded and decrypted. The vault is scanned for an entry that matches the selected service provider (i.e., website 122). If a matching entry is found, the associated domain, username, and site credential are extracted from the vault. In some embodiments, the validity of the domain name mapping is verified at this point to harden the system against domain name poisoning. Next, a secure connection is established between Alice's computer and the service provider, and Alice is authenticated. For service providers supporting strong user authentication, mutual SSL can be used, for example. A variety of policies can be involved when performing matching. For example, to access certain domains, Alice's print may need only match template 210. To access other domains, Alice may need to match multiple templates (e.g., both 210 and 214). As another example, in order to access social networking site 120, Alice may merely need to be sitting in front of her computer, which has an integrated webcam. Even in relatively low light conditions, a match can be performed against Alice's face and features stored in a template. However, in order to access bank website 122, Alice may need a high quality photograph (i.e., requiring her to turn on a bright light) and may need to demonstrate liveness (e.g., by blinking or turning her

head). As yet another example, other contextual information can be included in policies. For example, if Alice's IP address indicates she is in a country that she is not usually in, she may be required to match multiple templates (or match a template with more/better quality features) in order to access retailer 124, as distinguished from when her IP address indicates she is at home.

[0027] In some embodiments, the biometric sensor used by a user may be a peripheral device (e.g., a mouse with an integrated fingerprint scanner that is connected to the user's primary device via USB). In such scenarios, the peripheral device may be responsible for storing at least a portion of authentication information and may perform at least some of the authentication tasks previously described as having been performed by Alice's computer. For example, instead of processors 304 and 308, and storages 302 and 306 being collocated on a single device (e.g., laptop 102), processor 304 and storage 302 may be present on a primary device, and processor 308 and storage 306 may be present on a peripheral device (e.g., that also includes a sensor, such as a fingerprint reader).

[0028] In such scenarios, once Alice's login to banking website 122 is successfully completed, the secure session can be handed over from the peripheral device to the primary device, in a way that does not allow the primary device retroactive access to the plaintext data of the transcripts exchanged between the peripheral device and the service provider. One way this can be accomplished is by renegotiating SSL keys between the peripheral device and the website, after which the newly negotiated key can be handed off from the peripheral device to the primary device. This avoids retroactive credential capture in a setting where the device is infected by malware.

[0029] An example of renegotiation is depicted in Figure 4. Specifically, after a user has successfully authenticated to a fingerprint reader, a login is performed to a service provider. Using the primary device (404) as a proxy, the peripheral fingerprint reader 402 negotiates a first SSL connection (408) with a service provider 406, over which credentials are exchanged. The proxy then renegotiates SSL (410), which replaces the old key with a new one. The new key is disclosed to the device, which then seamlessly takes over the connection with the service provider and performs the transaction protected by the authentication. The credentials exchanged during the first SSL connection cannot be accessed by device 404, since the key of the

renegotiated session is independent of the key of the original session; this provides protection against malware residing on the device. Renegotiation can be used when the primary device 404 is believed to be in a safe state when performing the negotiation of the SSL connection, but it is not known whether it is in a safe state during the transaction protected by the authentication. Renegotiation can also be used when a secure component of the primary device 404 performs the negotiation of the SSL connection and another and potentially insecure component of the primary device 404 is involved in the transaction protected by the authentication.

[0030] Figure 5 illustrates an embodiment of a process for performing authentication translation. The process begins at 502 when a request to access a resource is received, as is an authentication input. One example of the processing performed at 502 is as follows. Suppose Alice wishes to sign into social networking website 120. She directs a web browser application installed on client 102 to the social networking website. Authentication translator module 132 recognizes, from the context of Alice's actions (e.g., that she is attempting to access site 120 with her browser) that she would like to access a particular resource. Authentication translator module 132 prompts Alice (e.g., by a popup message or via a sound) to provide biometric information (e.g., to use the integrated fingerprint reader on her laptop). In some embodiments, the translator module does not prompt Alice, for example, because Alice has been trained to provide biometric information automatically when attempting to access certain resources. In yet other embodiments, the translator module only prompts Alice if she fails to provide acceptable biometric information within a timeout period (e.g., 30 seconds).

[0031] Module 132 compares Alice's supplied biometric data to the templates stored on her computer. If a suitable match is found, and if an entry for site 120 is present in the applicable vault, at 504, a previously stored credential associated with the resource is accessed. In particular, the username and password for the website, as stored in a vault, such as vault 220, are retrieved from the vault.

[0032] Finally, at 506, the credential is provided to the resource. For example, Alice's username and password for site 120 are transmitted to site 120 at 506. The credential can be transmitted directly (e.g., by the module or by Alice's computer) and can also be supplied

indirectly (e.g., through the use of one or more proxies, routers, or other intermediaries, as applicable).

[0033] Other devices can also make use of process 500 or portions thereof. For example, when Alice launches a banking application on phone 104, implicit in her opening that application is her desire to access the resources of website 134. The application can take Alice's picture and compare it to stored templates/vault information. If an appropriate match is found, a credential can be retrieved from the vault on her phone (or, e.g., retrieved from cloud storage service 140) and provided to website 134.

[0034] As another example, suppose Charlie is using tablet 106 and attempts to visit site 120, whether via a dedicated application or via a web browser application installed on the tablet. Charlie's photograph is taken, and then compared against the profiles stored on tablet 106 (e.g., both Bob and Charlie's profiles). When a determination is made that Charlie's photograph matches a template stored in his stored profile (and not, e.g., Bob's), Charlie's credentials for site 120 are retrieved from a vault and transmitted by an authentication translator module residing on client 106.

[0035] As yet another example, kiosk 108 can be configured to provide certain local resources (e.g., by displaying a company directory or floor plan on demand) when users speak certain requests into a microphone. Enrolled users (e.g., with stored voiceprint or facial recognition features) can be granted access to additional/otherwise restricted services in accordance with the techniques described herein and process 500.

[0036] New device

[0037] In some embodiments, to register a new device, a user provides an identifier, such as a username or an account number to the device. The new device connects to an external storage (such as cloud storage 140), provides the user identifier and credential, and downloads the user's templates/vaults from the service. In some embodiments, the templates/vaults are encrypted. Once downloaded, the template is decrypted and stored in a secure storage area, while the still encrypted vault can be stored in insecure storage. The decryption key can be generated from information the user has/knows, or from biometric data -- such as features

extracted from fingerprinting of all ten fingers. In some embodiments, more arduous fingerprinting is required for the setup of a new device than for regular authentication to avoid that a new device gets registered by a user thinking she is merely authenticating -- or worse still, simply touching the device. Moreover, it translates into higher entropy of the decryption keys.

[0038] Backup Authentication

[0039] Backup authentication allows a user, such as Alice, to access resources in the event she is unable to or unwilling to interact with a particular biometric sensor. As one example, instead of having a single template associated with her profile, Alice can have multiple templates associated with it, e.g., where the first template includes fingerprint features and the second template includes voice biometric, facial recognition, or iris detection features. As a second example, where the service Alice is connecting to is a legacy website (i.e., one that users authenticate to using usernames and passwords), such a service would allow the use of passwords and password reset mechanisms by Alice without requiring Alice to use a fingerprint reader.

[0040] In various embodiments, environment 100 supports the ability of users (e.g., under duress) to release the contents of their vaults. For example, if Alice was physically threatened with the loss of a finger by a criminal, Alice could instead release the contents of her vault(s) – the ultimate goal of the criminal. As one example, in the event Alice supplies all 10 fingerprints to the sensor, provides a special password, or supplies a fingerprint and a second identifier, a cleartext version of her vault(s) could be made available.

[0041] Access Policies

[0042] In various embodiments, cloud storage service 140 is configured to accept backups from multiple devices associated with a single account, and synchronize the updates so that all devices get automatically refreshed. For example, Alice's laptop 102 and phone 104 could both communicate with cloud storage service 140 which would keep their authentication information synchronized. Refreshes can also be made in accordance with user-configured restrictions. For example, Alice's employer could prevent privileged employer data from being stored on shared personal devices, or on any device that was not issued by the employer. As

another example, arbitrary policies can be defined regarding the access to and synchronization of software and data, and to tie a license or access rights to a person (and her fingerprint) rather than to a device. As yet another example, in some embodiments (e.g., where a device is made publicly available or otherwise shared by many users), no or a reduced amount of authentication information resides on a device, and at least a portion of authentication information is always retrieved from cloud storage service 140.

[0043] Remote wiping

[0044] Remote wiping of a user's authentication information (e.g., templates) can be used both to "unshare" previously shared devices (e.g., where Bob and Charlie both have user profiles on their shared tablet 106), and to avoid that criminals with physical component access to lost devices gain access to templates and vault contents. In some embodiments, polices such as ones where a template self-wipes if it is not matched within a particular duration of time are supported. Since user data can be frequently backed up to the cloud storage, and recovered from this using the new device registration process, inconvenience to the user will be minimized.

[0045] **Legacy Server Support**

[0046] New authentication schemes typically require changes to a significant codebase residing with service providers. If the code is well written and documented, such changes may be relatively simple. Commonly, though, this may not be so. The engineers who originally wrote the code of relevance may have long since left the company; the code they left behind may be poorly documented -- if documented at all. In severe cases, the legacy code may have been written in an outdated programming language or written in a way that does not follow guidelines for good code. This makes updates to the codebase impractical or virtually impossible in many common cases. Even if none of these challenges complicate the desired modifications, it is commonly a great bureaucratic burden to obtain permission to make modifications (e.g., to store an extra field in a backend database), because every affected part of the organization may need to review the request.

[0047] As will be described in the following section, the technologies described herein can be used in conjunction with legacy servers (e.g., existing servers that rely on usernames and

passwords to authenticate users), and in particular, can be used without requiring modification to such legacy servers.

[0048] Cookies

[0049] Cookies are commonly used by legacy servers for user authentication. Unfortunately, cookies have several problems. For one thing, they are sometimes deleted -- whether explicitly/intentionally by the end user or by the user's software. In addition, cookies are commonly stolen. Approaches such as cache cookies and identification using user agents can be more resistant to these problems, however, they have their own problems. For example, their use requires new code and new fields in the credential database stored by the server.

[0050] In some embodiments, authentication translators, such as translators 134 and 136 (also referred to herein as proxies) provide authentication translation services on behalf of associated services. Translators 134 and 136 are illustrated as single logical devices in Figure 1. In some embodiments, the translators comprise standard commercially available server hardware (e.g., a multi-core processor, 4+ Gigabytes of RAM, and one or more Gigabit network interface adapters) and run typical server-class operating systems (e.g., Linux). Translators 134 and 136 can also be implemented using a scalable, elastic architecture and may comprise several distributed components, including components provided by one or more third parties. For example, translators 134 and 136 may store user credential information or may task cloud storage service 140 with storing at least a portion of that information.

[0051] In the case of authentication translator 134, service is provided with respect to bank website 122 only. Authentication translator 134 is positioned between a legacy web server (122) and the Internet (110) -- and therefore between the legacy server and any client devices. Authentication translator 134 is configured to translate traffic between the legacy server and client devices so that the client devices (and respective users) perceive the new authentication mechanism, while the legacy server remains unchanged. Authentication translator 136 works similarly, but it provides authentication translation as a third party service to multiple providers, an example of which is online camera retailer 124.

[0052] Authentication translators 134 and 136 can also perform process 500. For example, when a device transmits a request to access website 122, the request is intercepted by translator 134, as is cookie/user agent information. The received information can be used to determine a username/password associated with the device, and that information can be passed by translator 134 to website 122 on behalf of the device.

[0053] Figure 6 illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator. The translator (referred to in the figure as a "proxy") fails to identify the client, and passes on the request to the legacy server. The legacy server responds to the request and sets a cookie. The proxy passes on the response, including the cookie and also a cache cookie. The proxy stores the information about both these types of identifiers, along with the user agent of the client device. This triplet of information is also referred to herein as an identifier.

[0054] Figure 7 illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator. In this scenario, the device's request is accompanied by some form of identifying information. The proxy uses this information to identify the associated cookie and passes this along to the legacy server, along with the request. In some embodiments, additional processing is also involved. For example, reading a cache cookie may require user interaction. Moreover, if not all of the identifying information is present in the request, the proxy can be configured to set the missing information again by sending a corresponding request to the client device.

[0055] The translation of cache cookies and user agent information to cookies involves a two-way translation. First, when the legacy server sets a cookie, the proxy will set the two types of cookies -- both an HTML cookie and a cache cookie -- and then create a new record in which the two cookies are stored, along with the user agent information of the client device. The user agent information can include quite a bit of data associated with a browser -- such as the browser type and version, the clock skew, and the fonts that are installed. While each of these pieces of information only contributes a small amount of entropy, the collection of items can be sufficient to identify the device in most cases. Moreover, while some of these types of data may change over time -- in fact, all of them may -- they do not typically change, and when one or two of

them do, the others typically do not. When the client device is used to visit a site controlled by the legacy server, the cookie, cache cookie and user agent information are read (if available), the record identified, and the request translated and sent to the legacy server. When a legacy server requests that the user password is updated (e.g., as part of an annual or other periodic requirement), the transmission of this request to the user can be suppressed -- in which case the database of the proxy is updated to create the illusion of an update. The user can be involved in authentication as needed, e.g., if, in addition to supplying a credential, a user must also solve a CAPTCHA, the CAPTCHA can be displayed to the user (with the user's credentials being handled in accordance with the techniques described herein).

[0056] Figure 8 shows the structure of an example of a cache cookie used in some embodiments. Cache cookies can be associated with a particular webpage, just like an HTML cookie can. In the example shown in Figure 8, the proxy wishes to associate a page "calling page.html" with a cache cookie. It embeds a request for a second object, "samename.html" in callingpage.html for every visitor. However, as the cache cookie is set for one visitor, a customized samename.html is served to this visitor. The page samename.html refers to an object with a different name for each user; that object is referred to as "uniquename.jpg." The cache cookie is set by embedding the request for samename.html in callingpage.html. The client browser attempts to render this, causing a request for samename.html from the server. The server configures samename.html to refer to a uniquely named file uniquename.jpg, and serves samename.html to the client. For the client browser to render samename.html, it requests the file uniquename.jpg, which is intentionally not served. That concludes the setting of a cache cookie. As a user returns to the page callingname.html, the browser again attempts to render the entire page, which causes it to load the object samename.html from its cache. As that is rendered, the client browser requests uniquename.jpg, which is not in its cache (since it was not served previously). The server still does not serve it, but takes note of the name of the file being requested, as it identifies the client device. Note that samename.html can be displayed in a zero-sized iframe, which makes the end user unaware of it being rendered.

[0057] A cache cookie is an implementation of the typical cookie functionality that uses the client device's browser cache. Unlike user agents, it does not change over time, and like standard HTML cookies, it cannot be read by a party other than that which set it. However, like

HTML cookies, it could be deleted -- by the user clearing his or her browser cache. Cache cookies are not automatically transmitted with GET requests, unless the cache elements are embedded in the referring pages. This adds a potential round of communication in some settings. By relying on user agent information, cache cookies, and HTML cookies to identify the client device, it is much more likely that a machine will be recognized than if only HTML cookies are used.

[0058] Although the foregoing embodiments have been described in some detail for purposes of clarity of understanding, the invention is not limited to the details provided. There are many alternative ways of implementing the invention. The disclosed embodiments are illustrative and not restrictive.

[0059] WHAT IS CLAIMED IS:

CLAIMS

1. A system, comprising:

a processor configured to:

based at least in part on a request from a user to access an external resource, facilitate, using a restricted interface, access of at least one record stored at least in part in a secure storage;

wherein the at least one record is associated at least with the external resource, and wherein the at least one record comprises:

a biometric template; and

a credential comprising at least one of a password, a cookie, or a cryptographic key;

in response to determining a match between a biometric input and the biometric template, retrieve, from the at least one record, the credential, wherein the biometric input corresponds to at least one of a fingerprint, a feature usable for facial recognition, a voiceprint, a feature usable for a retina scan, or a typing feature, and wherein the biometric input is received subsequent to presenting of a prompt;

establish a connection with the external resource;

facilitate a login of the user to the external resource at least in part by transmitting, via the established connection, output based at least in part on the credential retrieved from the at least one record, and wherein the user is logged in to the external resource based at least in part on the output; and

facilitate wiping of at least a portion of the at least one record; and

a memory coupled to the processor and configured to provide the processor with instructions.

2. The system recited in claim 1 wherein the secure storage is connected to a sensor.

3. The system recited in claim 2 wherein the sensor comprises at least one of a camera and a fingerprint reader.

4. The system recited in claim 1 wherein at least some of the at least one record is stored in plaintext in the secure storage.

5. The system recited in claim 1 wherein the prompt comprises a prompt to provide biometric information, and wherein the prompt is visually presented.

6. The system recited in claim 1 wherein the prompt comprises a prompt to provide biometric information, and wherein the prompt is aurally presented.

7. The system recited in claim 1 wherein the prompt is presented in response to a user failing to provide acceptable biometric information within a timeout period.

8. The system recited in claim 1 wherein facilitating wiping of the at least portion of the at least one record comprises facilitating remote wiping of the at least portion of the at least one record.

9. The system recited in claim 1 wherein the at least portion of the at least one record is automatically wiped based at least in part on a policy.

10. The system recited in claim 1 wherein the biometric template is wiped in response to determining that the biometric template has not been matched within a duration of time.

11. The system recited in claim 1 wherein the at least portion of the at least one record is backed up to a remote entity.

12. A method, comprising:

based at least in part on a request from a user to access an external resource, using a processor to facilitate, using a restricted interface, access of at least one record stored at least in part in a secure storage;

wherein the at least one record is associated at least with the external resource, and wherein the at least one record comprises:

a biometric template; and

at least one of a password, a cookie, or a cryptographic key;

in response to determining a match between a biometric input and the biometric template, retrieving, from the at least one record, the credential, wherein the biometric input corresponds to at least one of a fingerprint, a feature usable for facial recognition, a voiceprint, a feature usable

for a retina scan, or a typing feature, and wherein the biometric input is received subsequent to presenting of a prompt;

establishing a connection with the external resource;

facilitating a login of the user to the external resource at least in part by transmitting, via the established connection, output based at least in part on the credential retrieved from the at least one record, and wherein the user is logged in to the external resource based at least in part on the output; and

facilitating wiping of at least a portion of the at least one record.

13. The method of claim 12 wherein the secure storage is connected to a sensor.

14. The method of claim 13 wherein the sensor comprises at least one of a camera and a fingerprint reader.

15. The method of claim 12 wherein at least some of the at least one record is stored in plaintext in the secure storage.

16. The method of claim 12 wherein the prompt comprises a prompt to provide biometric information, and wherein the prompt is visually presented.

17. The method of claim 12 wherein the prompt comprises a prompt to provide biometric information, and wherein the prompt is aurally presented.

18. The method of claim 12 wherein the prompt is presented in response to a user failing to provide acceptable biometric information within a timeout period.

19. The method of claim 12 wherein facilitating wiping of the at least portion of the at least one record comprises facilitating remote wiping of the at least portion of the at least one record.

20. The method of claim 12 wherein the at least portion of the record is automatically wiped based at least in part on a policy.

21. The method of claim 12 wherein the biometric template is wiped in response to determining that the biometric template has not been matched within a duration of time.

22. The method of claim 12 wherein the at least portion of the at least one record is backed up to a remote entity.

23. A computer program product embodied in a non-transitory computer readable storage medium and comprising computer instructions for:

based at least in part on a request from a user to access an external resource, using a processor to facilitate, using a restricted interface, access of at least one record stored at least in part in a secure storage;

wherein the at least one record is associated at least with the external resource, and wherein the at least one record comprises:

a biometric template; and

a credential comprising at least one of a password, a cookie, or a cryptographic key;

in response to determining a match between a biometric input and the biometric template, retrieving, from the at least one record, the credential, wherein the biometric input corresponds to at least one of a fingerprint, a feature usable for facial recognition, a voiceprint, a feature usable for a retina scan, or a typing feature, and wherein the biometric input is received subsequent to presenting of a prompt;

establishing a connection with the external resource;

facilitating a login of the user to the external resource at least in part by transmitting, via the established connection, output based at least in part on the credential retrieved from the at least one record, and wherein the user is logged in to the external resource based at least in part on the output; and

facilitating wiping of at least a portion of the at least one record.

ABSTRACT OF THE DISCLOSURE

Authentication translation is disclosed. A request to access a resource is received at an authentication translator, as is an authentication input. The authentication input corresponds to at least one stored record. The stored record is associated at least with the resource. In response to the receiving, a previously stored credential associated with the resource is accessed. The credential is provided to the resource.

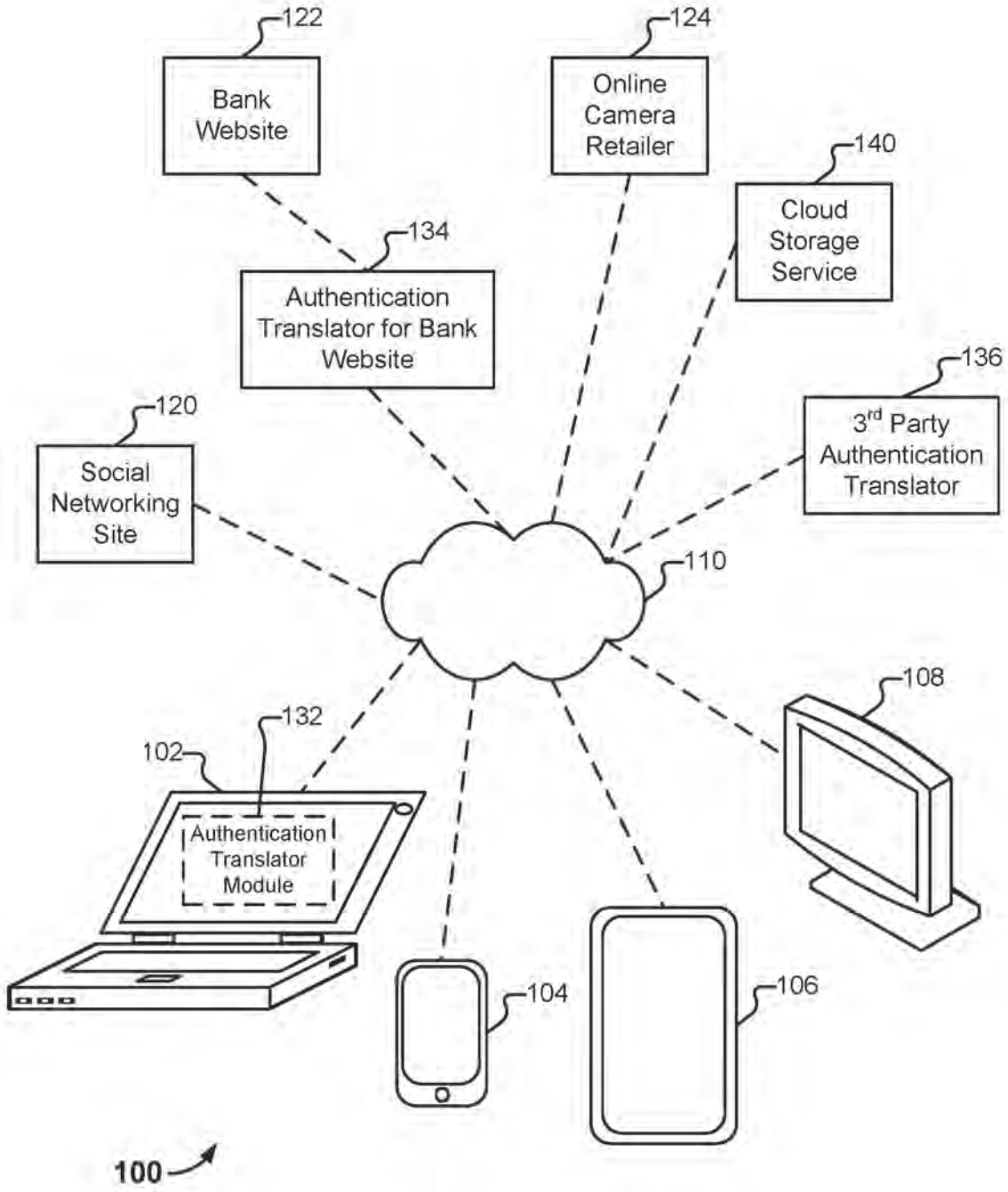

122
Bank Website
124
Online Camera Retailer
140
Cloud Storage Service
134
Authentication Translator for Bank Website
136
3rd Party Authentication Translator
120
Social Networking Site
110
108
132
102
Authentication Translator Module
104
106
100

FIG. 1

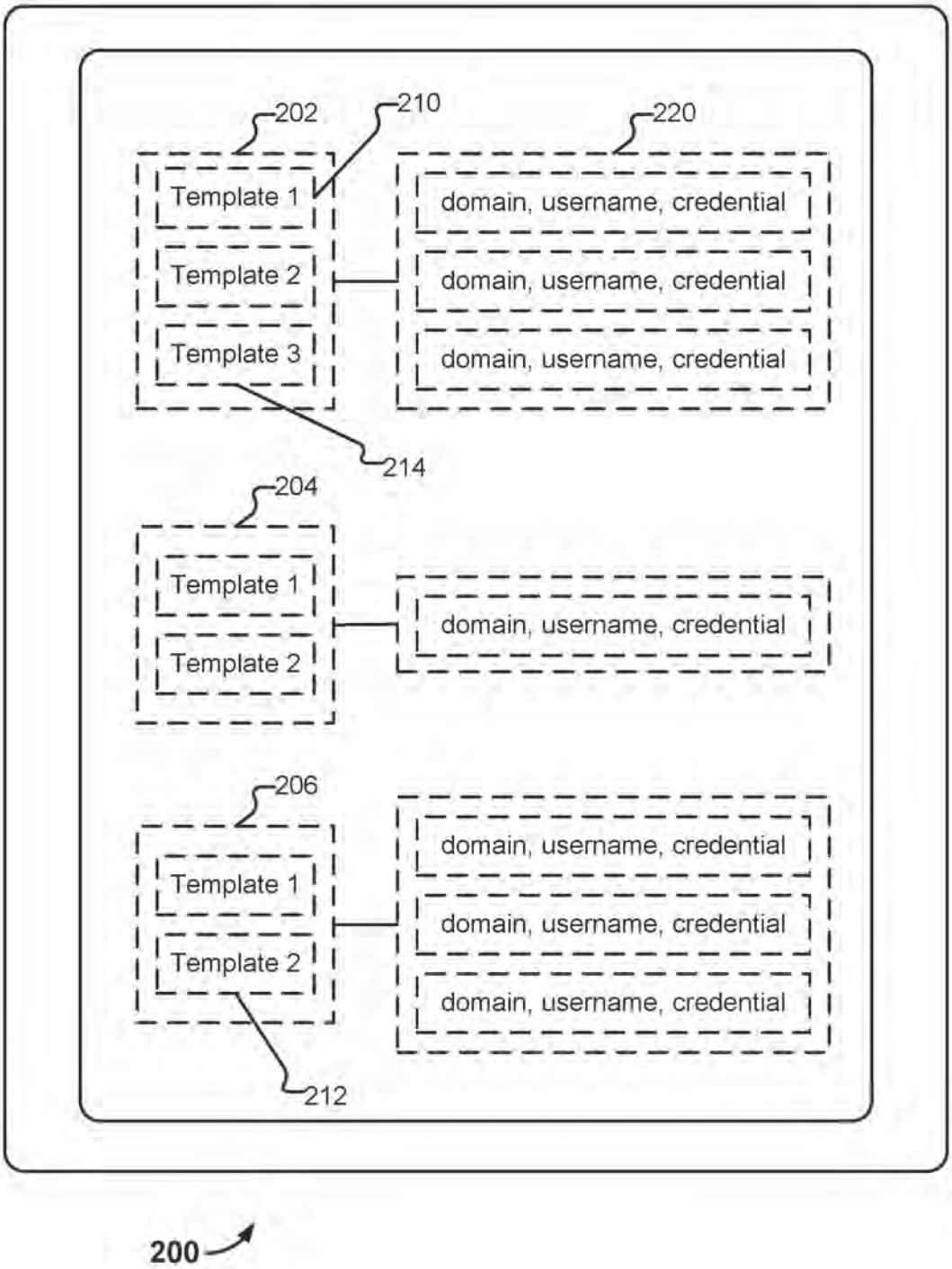
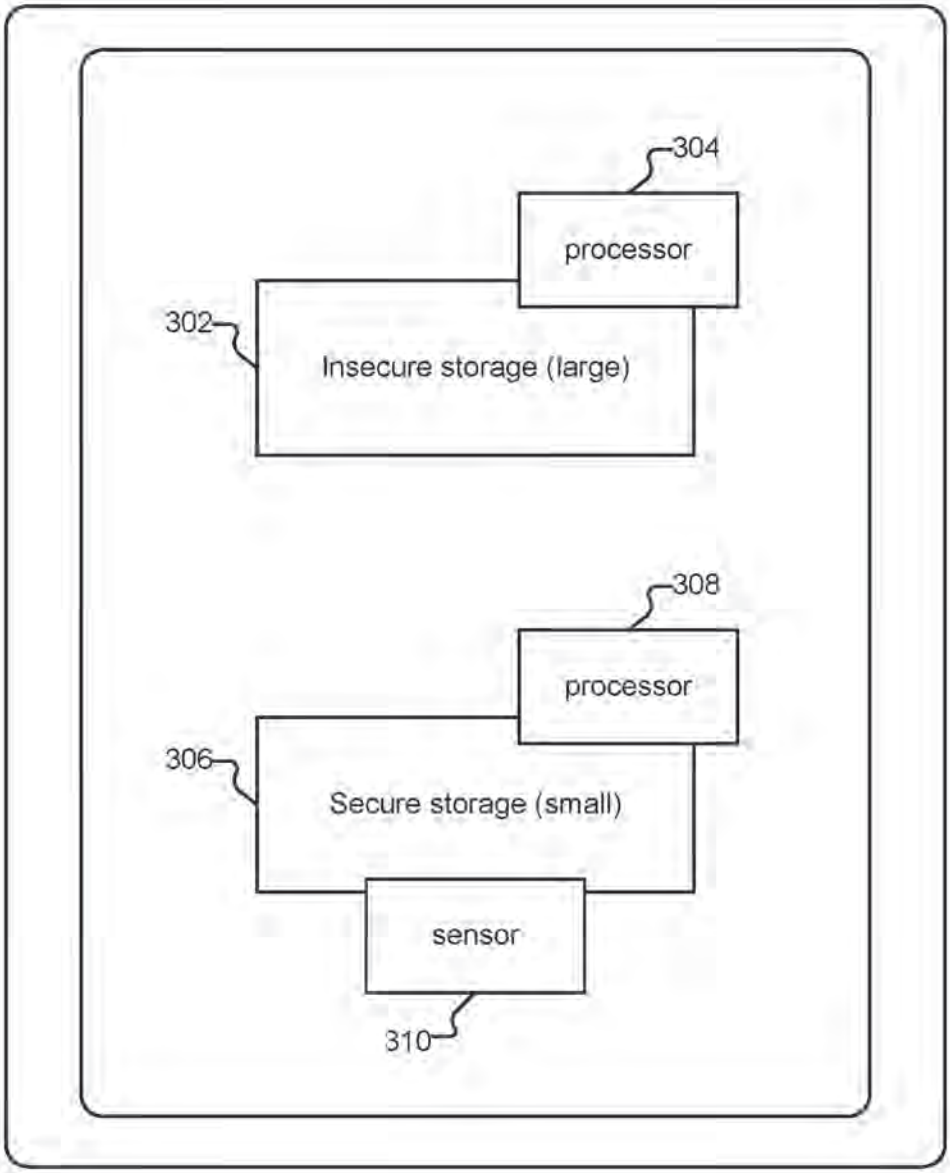

202
210
220
Template 1
Template 2
Template 3
domain, username, credential
domain, username, credential
domain, username, credential
214
204
Template 1
Template 2
domain, username, credential
206
Template 1
Template 2
domain, username, credential
domain, username, credential
domain, username, credential
212
200

FIG. 2


304
processor
302
Insecure storage (large)
308
processor
306
Secure storage (small)
sensor
310

300

FIG. 3

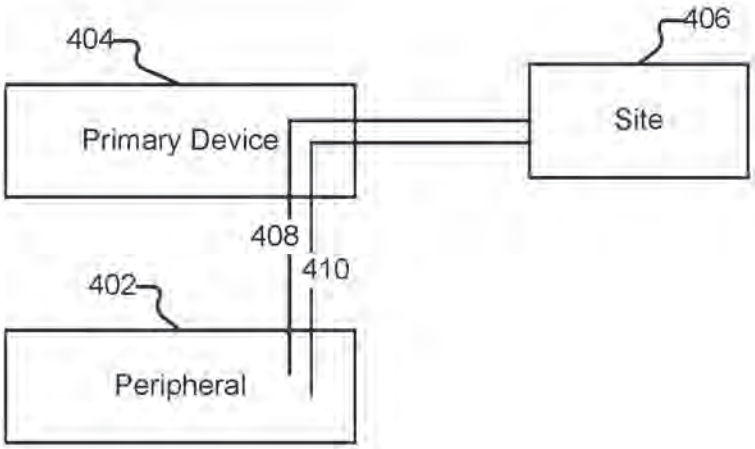


FIG. 4

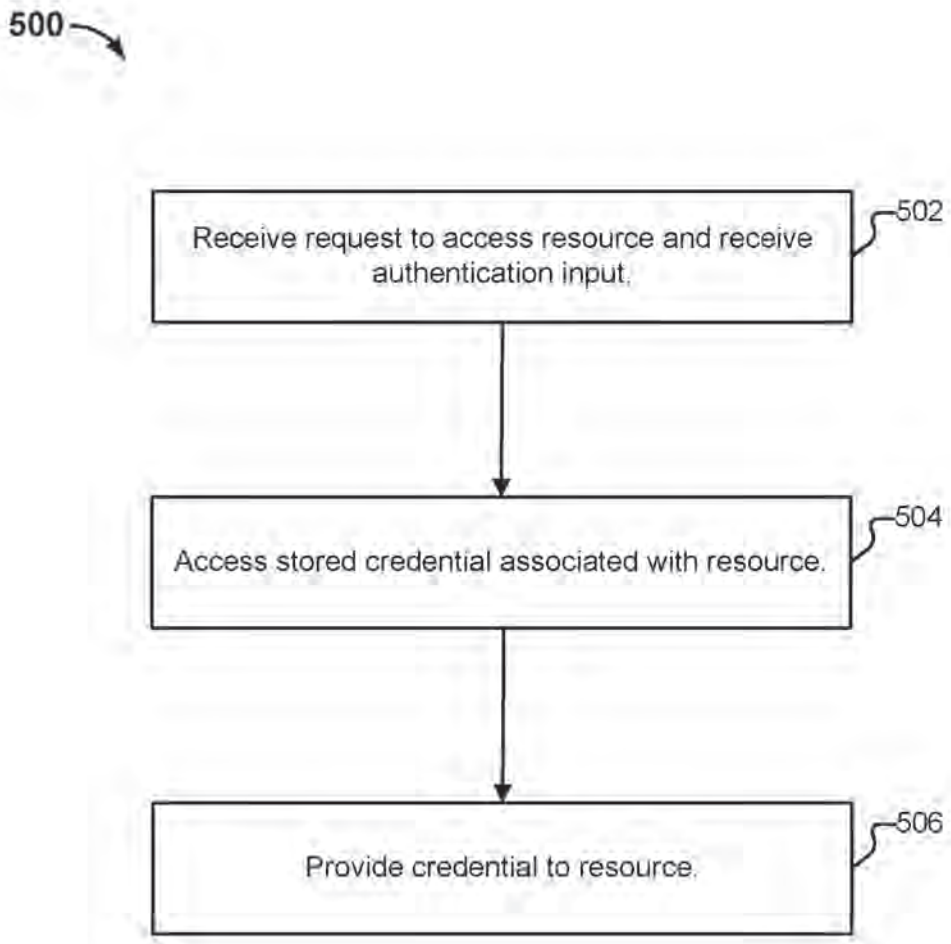

500
Receive request to access resource and receive authentication input.
502
Access stored credential associated with resource.
504
Provide credential to resource.
506

FIG. 5

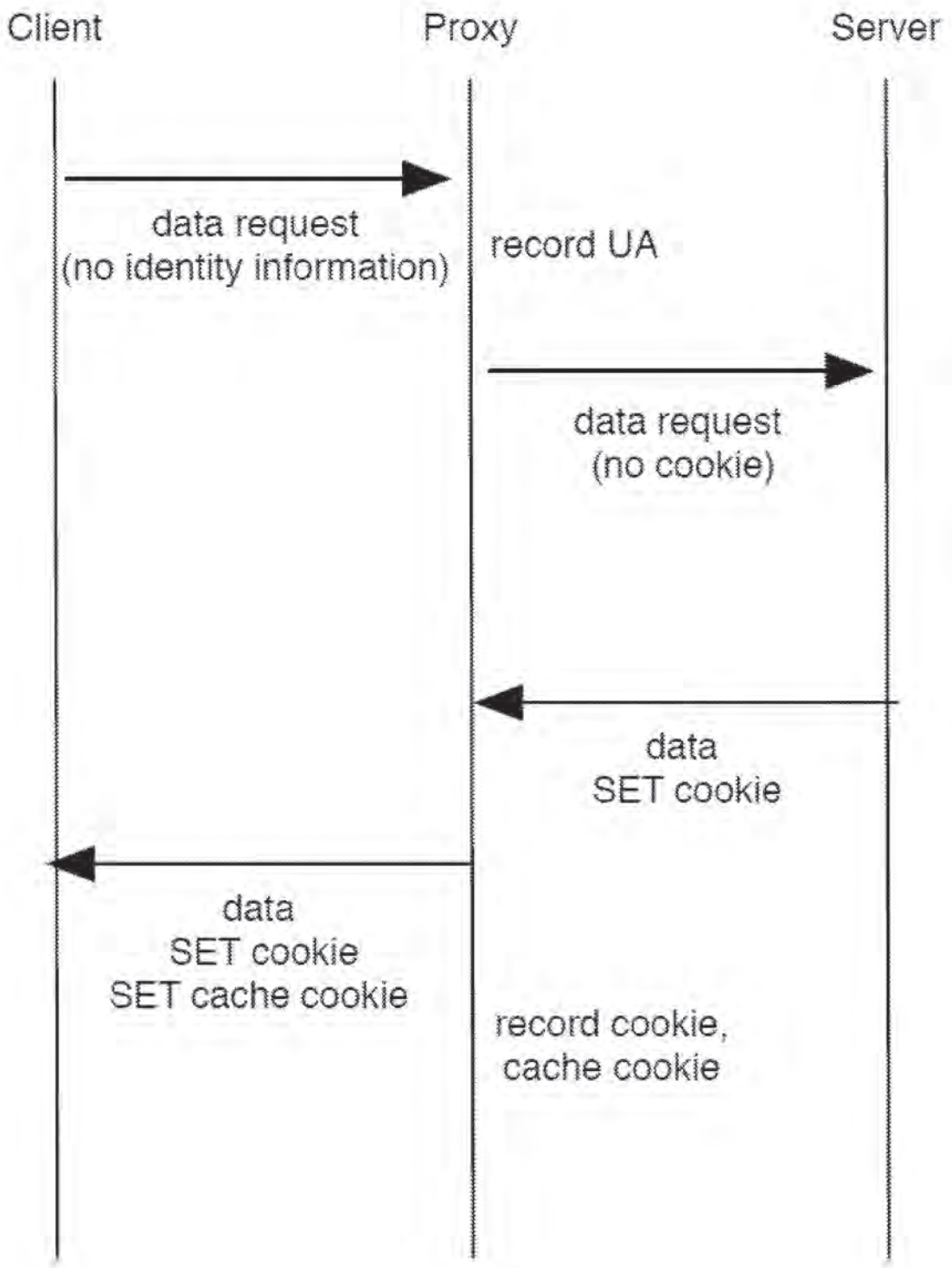

Client
Proxy
Server
data request
(no identity information)
record UA
data request
(no cookie)
data
SET cookie
data
SET cookie
SET cache cookie
record cookie,
cache cookie

FIG. 6

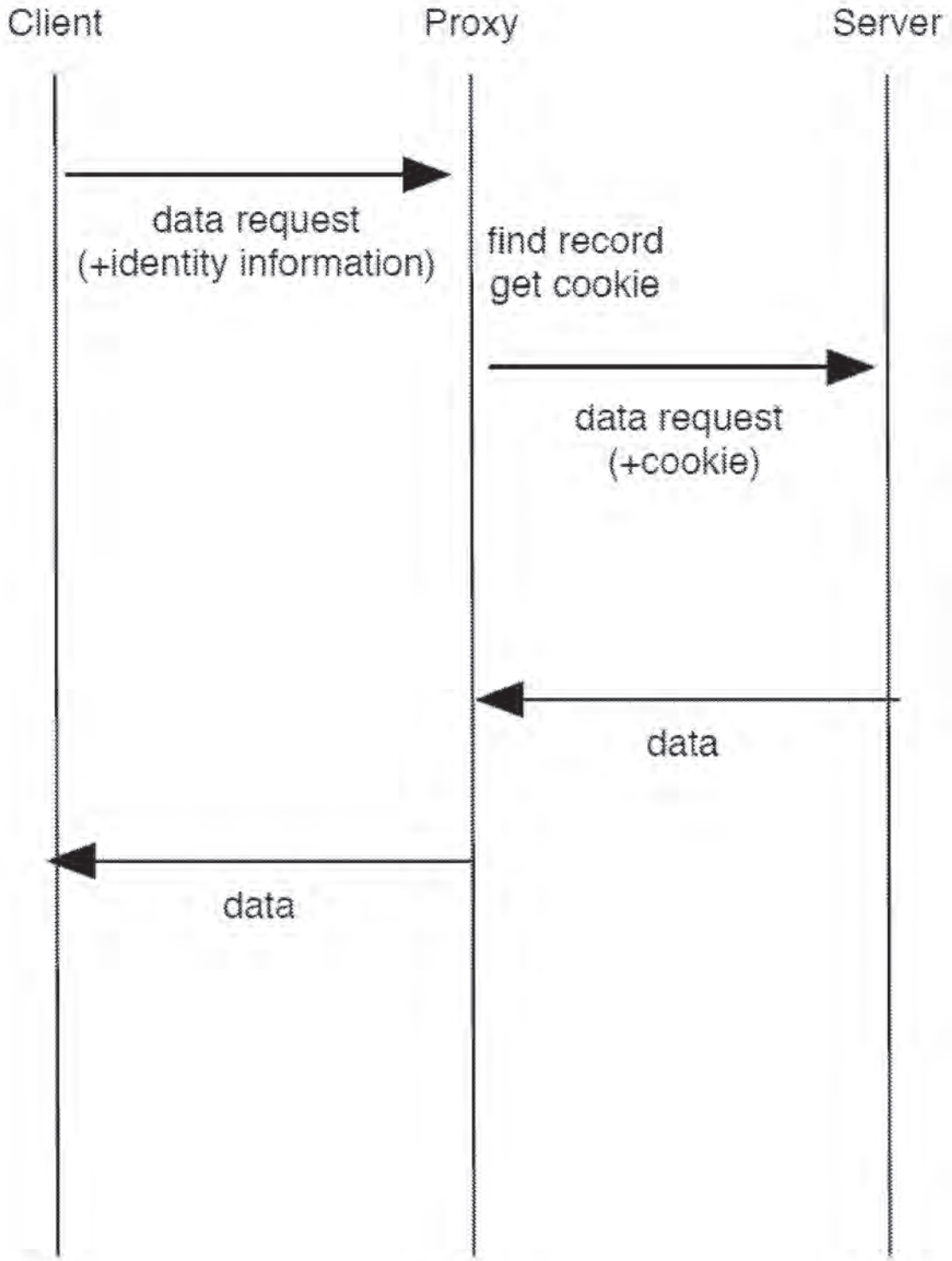

Client
Proxy
Server
data request
(+identity information)
find record
get cookie
data request
(+cookie)
data
data

FIG. 7

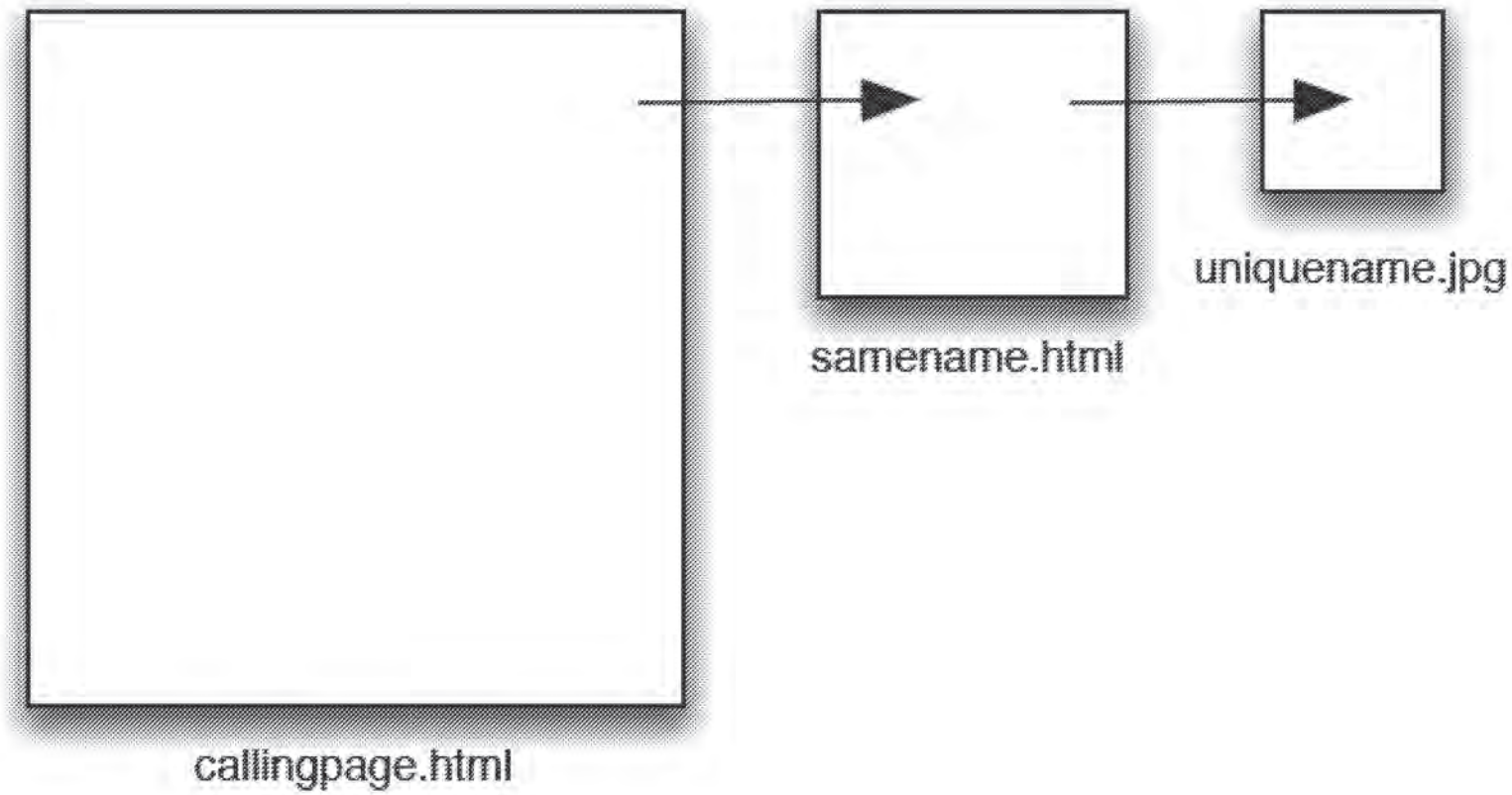


FIG. 8

Application Serial No. 16/773,767

Filing date: January 27, 2020

Patent No. 10,929,512

Issue date: February 23, 2021

US010929512B1

(12) **United States Patent**
Jakobsson

(10) **Patent No.: US 10,929,512 B1**
(45) **Date of Patent: *Feb. 23, 2021**

(54) **AUTHENTICATION TRANSLATION**

(71) Applicant: **RightQuestion, LLC**, Portola Valley, CA (US)

(72) Inventor: **Bjorn Markus Jakobsson**, Portola Valley, CA (US)

(73) Assignee: **RightQuestion, LLC**, Portola Valley, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **16/773,767**

(22) Filed: **Jan. 27, 2020**

Related U.S. Application Data

(63) Continuation of application No. 16/563,715, filed on Sep. 6, 2019, which is a continuation of application No. 16/273,797, filed on Feb. 12, 2019, now Pat. No. 10,521,568, which is a continuation of application No. 15/042,636, filed on Feb. 12, 2016, now Pat. No. 10,360,351, which is a continuation of application

(Continued)

(51) **Int. Cl.**

G06F 21/00	(2013.01)
G06F 21/10	(2013.01)
H04L 29/06	(2006.01)
G06F 21/32	(2013.01)
G06F 21/12	(2013.01)
G06F 21/31	(2013.01)
G06F 21/44	(2013.01)

(52) **U.S. Cl.**

CPC ***G06F 21/10*** (2013.01); ***G06F 21/121*** (2013.01); ***G06F 21/128*** (2013.01); ***G06F 21/31*** (2013.01); ***G06F 21/32*** (2013.01); ***G06F 21/44*** (2013.01); ***H04L 63/083*** (2013.01); ***H04L 63/0861*** (2013.01); ***H04L 63/10*** (2013.01); ***H04L 63/20*** (2013.01)

(58) **Field of Classification Search**

None

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,010,571 A	4/1991	Katznelson
5,499,298 A	3/1996	Narasimhalu

(Continued)

FOREIGN PATENT DOCUMENTS

WO	2004051585	6/2004
WO	2005001751	1/2005

OTHER PUBLICATIONS

"Managing Authorization and Access Control", Author: unknown, Published Nov. 3, 2005, pp. 1-12, URL: http://technet.microsoft.com/en-us/library/bb457115.aspx.

(Continued)

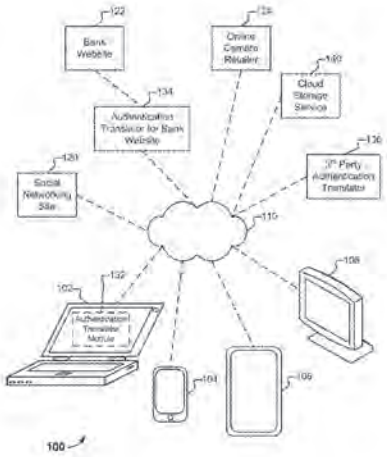
Primary Examiner — Andrew J Steinle

(74) *Attorney, Agent, or Firm* — Van Pelt, Yi & James LLP

(57) **ABSTRACT**

Authentication translation is disclosed. A request to access a resource is received at an authentication translator, as is an authentication input. The authentication input corresponds to at least one stored record. The stored record is associated at least with the resource. In response to the receiving, a previously stored credential associated with the resource is accessed. The credential is provided to the resource.

21 Claims, 8 Drawing Sheets



Related U.S. Application Data

No. 13/706,254, filed on Dec. 5, 2012, now Pat. No. 9,294,452.

(60) Provisional application No. 61/587,387, filed on Jan. 17, 2012, provisional application No. 61/569,112, filed on Dec. 9, 2011.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,016,476	A	1/2000	Maes et al.	
6,691,232	B1 *	2/2004	Wood	H04L 63/0815 726/6
7,512,965	B1	3/2009	Amdur	
7,697,729	B2	4/2010	Howell	
7,950,051	B1	5/2011	Spitz	
8,145,916	B2	3/2012	Boshra	
8,549,300	B1 *	10/2013	Kumar	H04L 9/3263 713/175
8,577,813	B2	11/2013	Weiss	
8,856,539	B2	10/2014	Weiss	
8,984,596	B2	3/2015	Griffin	
9,100,826	B2	8/2015	Weiss	
10,872,152	B1	12/2020	Martel	
2004/0107170	A1	6/2004	Labrou	
2004/0236632	A1	11/2004	Maritzen	
2005/0198348	A1	9/2005	Yeates	
2009/0100269	A1	4/2009	Naccache	
2010/0242102	A1	9/2010	Cross	
2011/0078771	A1	3/2011	Griffin	
2011/0205016	A1	8/2011	Al-Azem	
2011/0231651	A1	9/2011	Bollay	
2012/0110341	A1 *	5/2012	Beigi	H04L 9/3268 713/186
2012/0167193	A1	6/2012	Gargaro	

OTHER PUBLICATIONS

Hammer-Lahav, Ed. "The OAuth 1.0 Protocol", from https://tools.ieff.org/html/rfc5849, Apr. 2010.

* cited by examiner

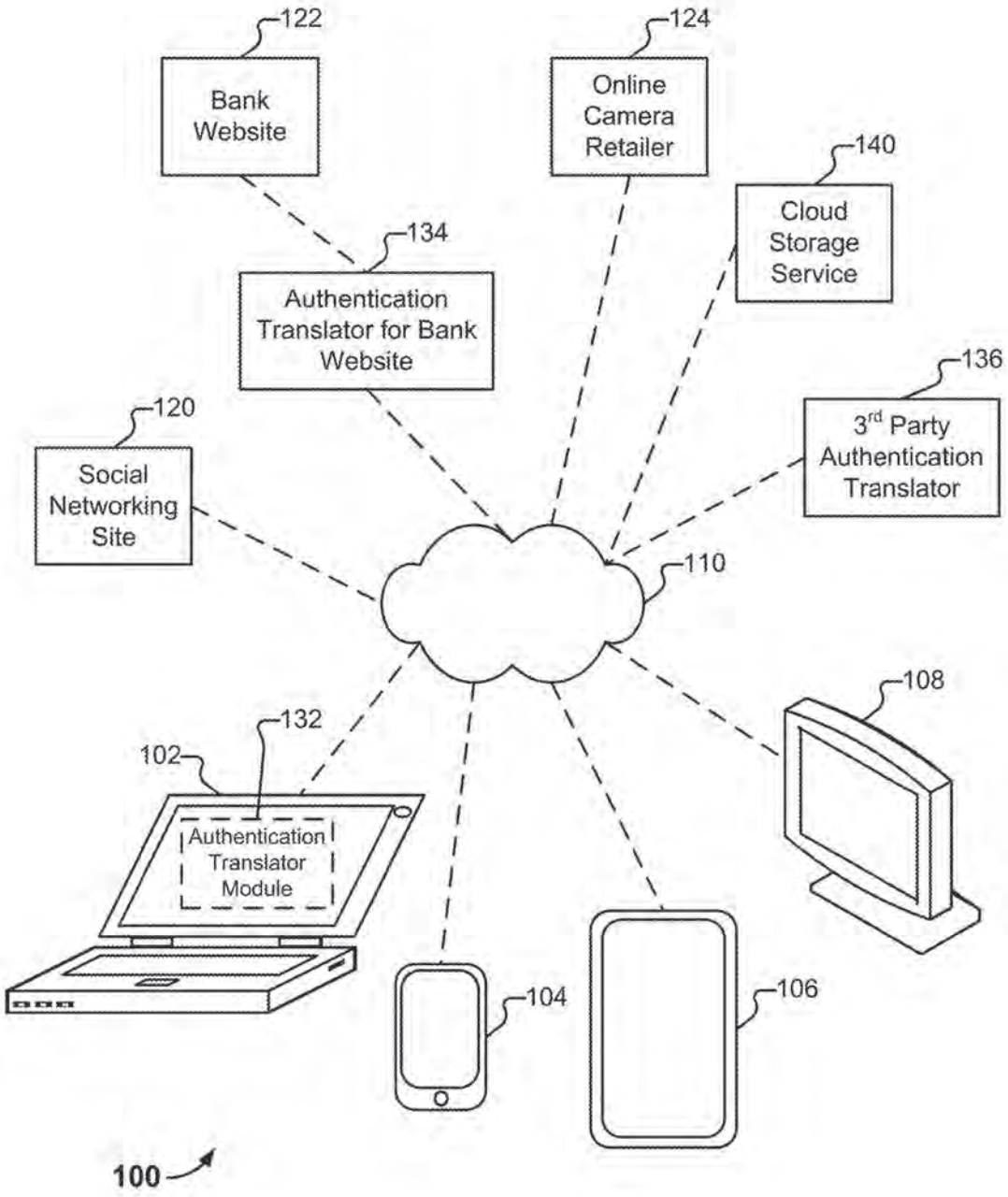

122
Bank Website
124
Online Camera Retailer
140
Cloud Storage Service
134
Authentication Translator for Bank Website
136
3rd Party Authentication Translator
120
Social Networking Site
110
108
132
102
Authentication Translator Module
104
106
100

FIG. 1

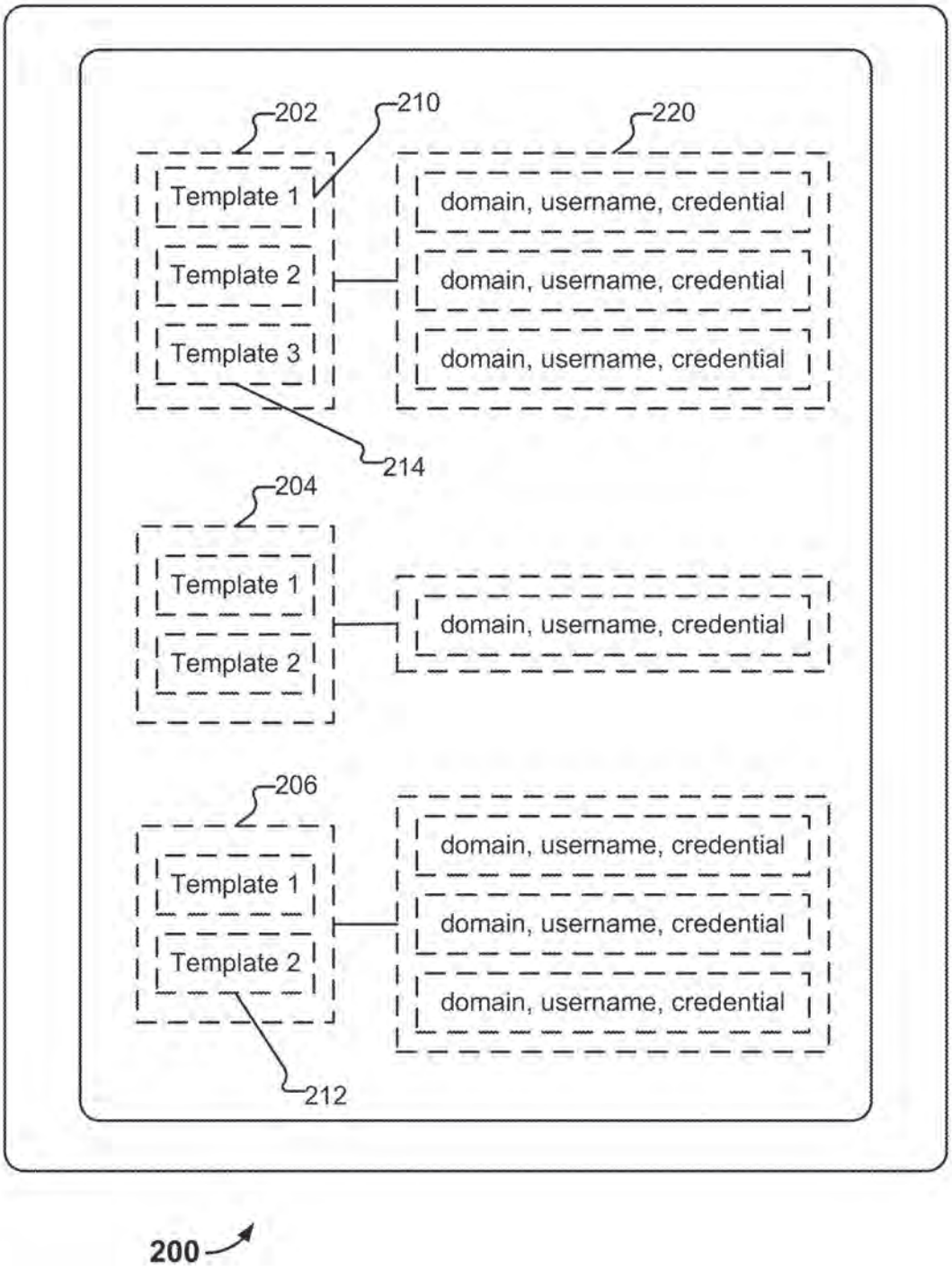
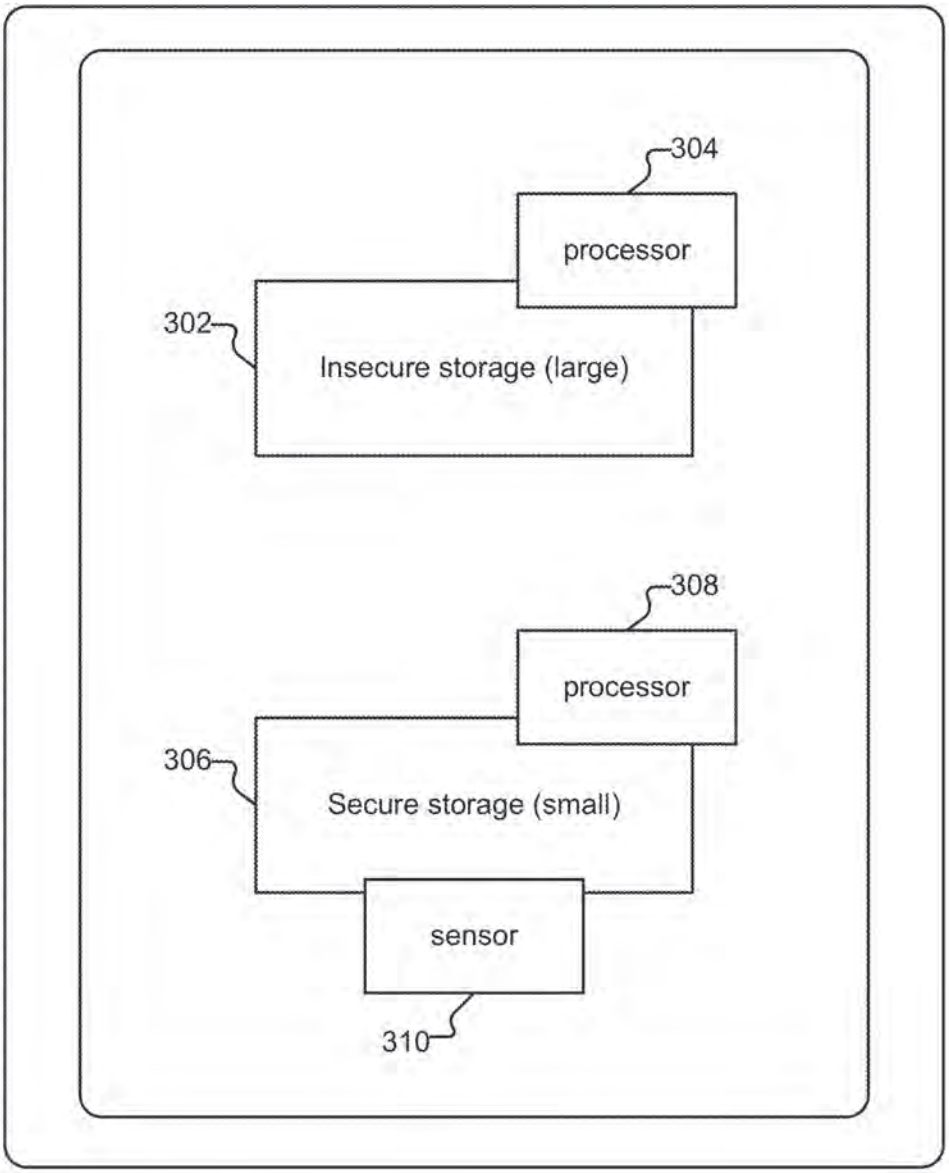

202
210
220
Template 1
Template 2
Template 3
domain, username, credential
domain, username, credential
domain, username, credential
214
204
Template 1
Template 2
domain, username, credential
206
Template 1
Template 2
domain, username, credential
domain, username, credential
domain, username, credential
212
200

FIG. 2


304
processor
302
Insecure storage (large)
308
processor
306
Secure storage (small)
sensor
310

300

FIG. 3

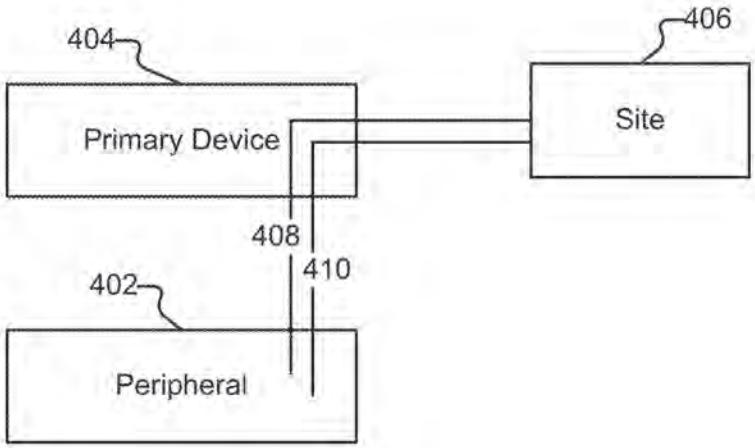


FIG. 4

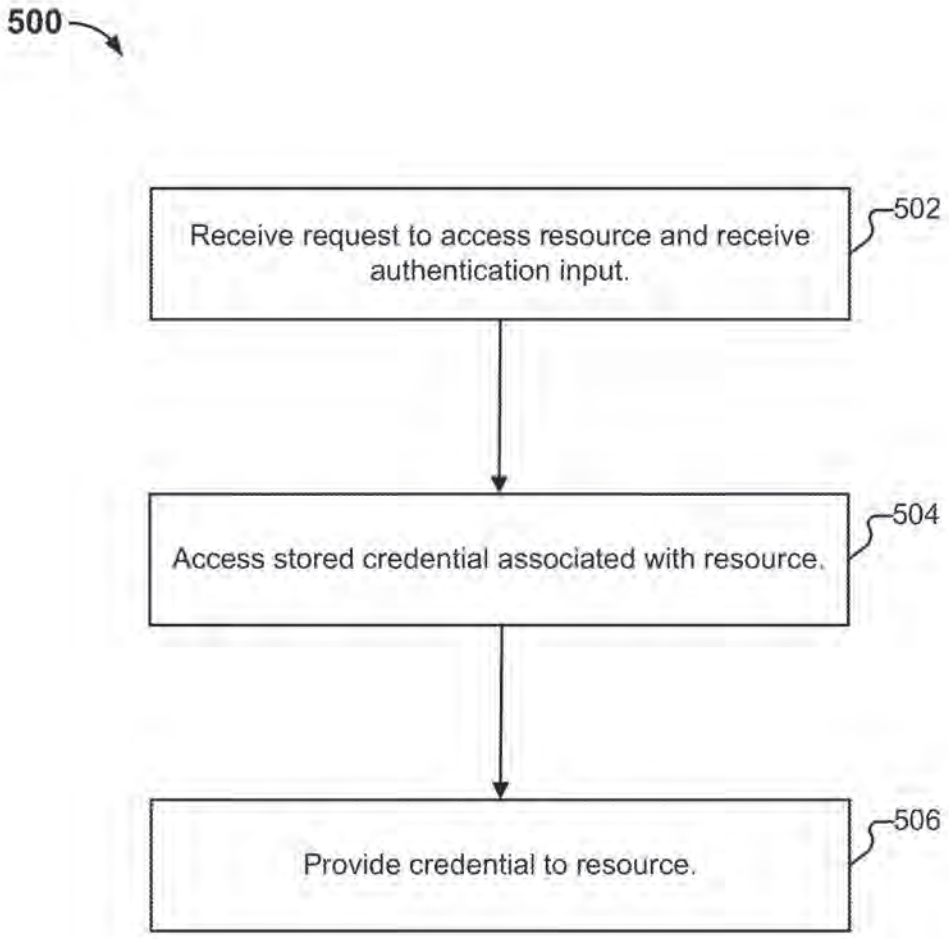

500
Receive request to access resource and receive authentication input.
502
Access stored credential associated with resource.
504
Provide credential to resource.
506

FIG. 5

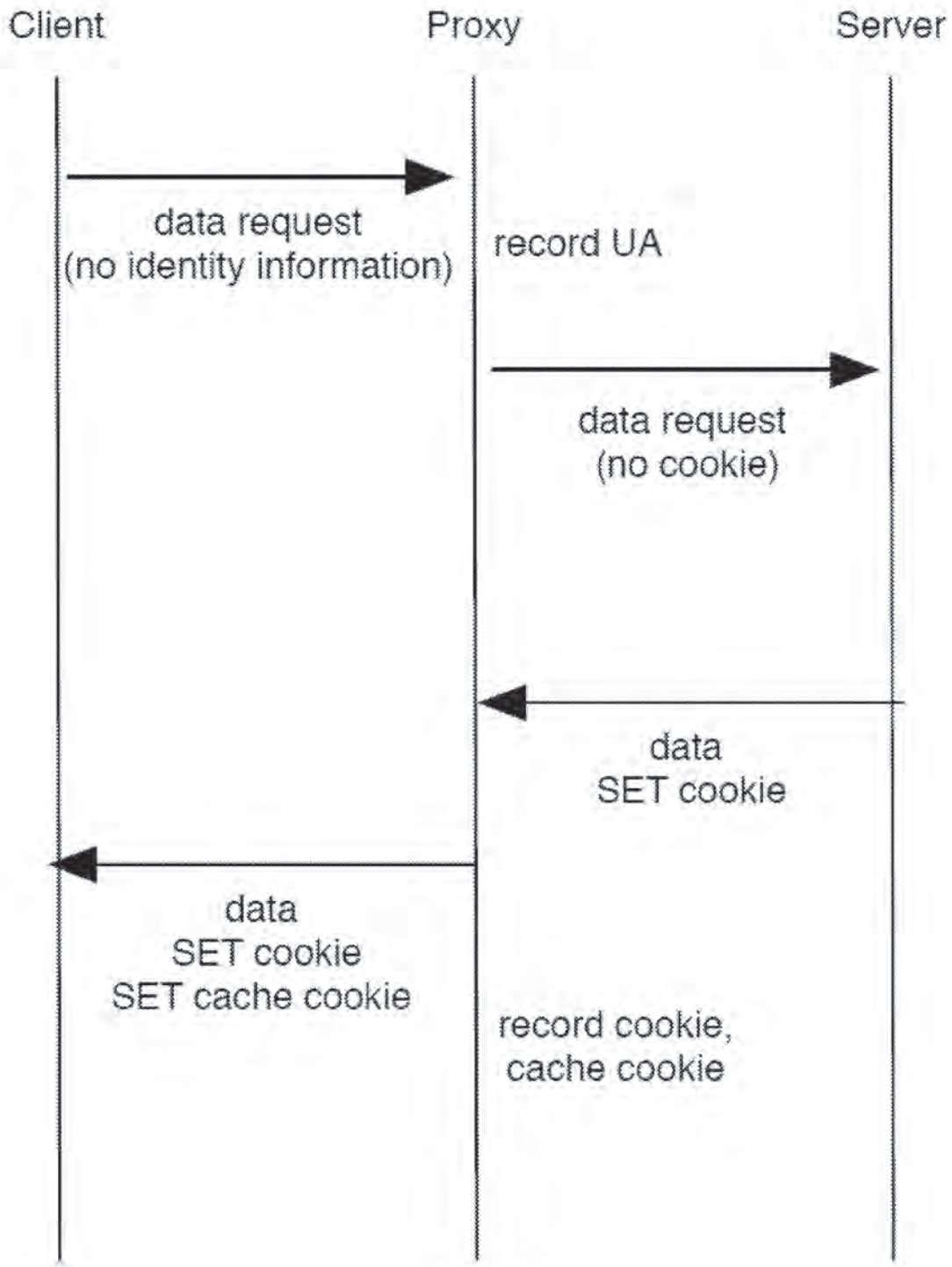

Client
Proxy
Server
data request
(no identity information)
record UA
data request
(no cookie)
data
SET cookie
data
SET cookie
SET cache cookie
record cookie,
cache cookie

FIG. 6

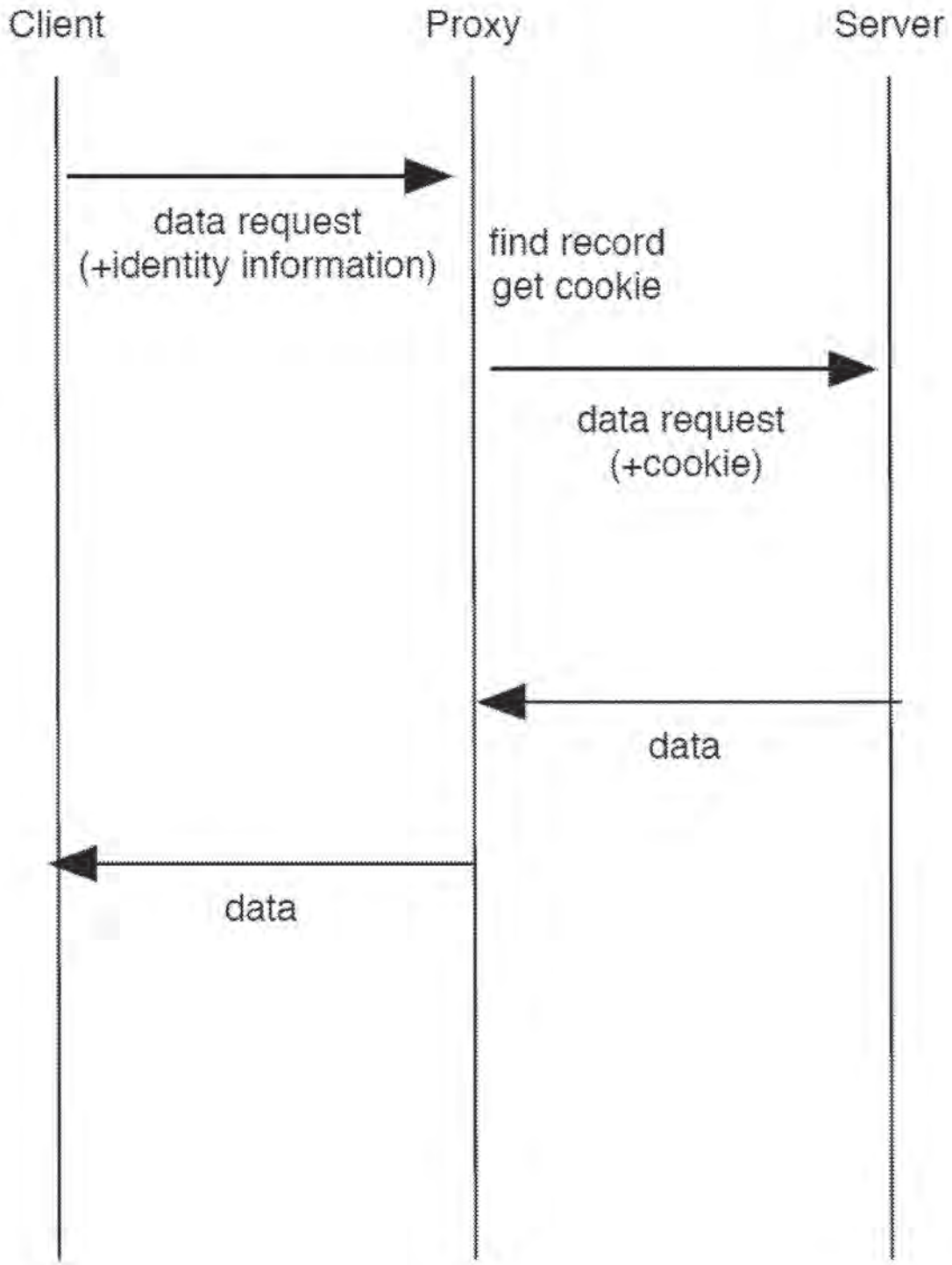

Client
Proxy
Server
data request
(+identity information)
find record
get cookie
data request
(+cookie)
data
data

FIG. 7

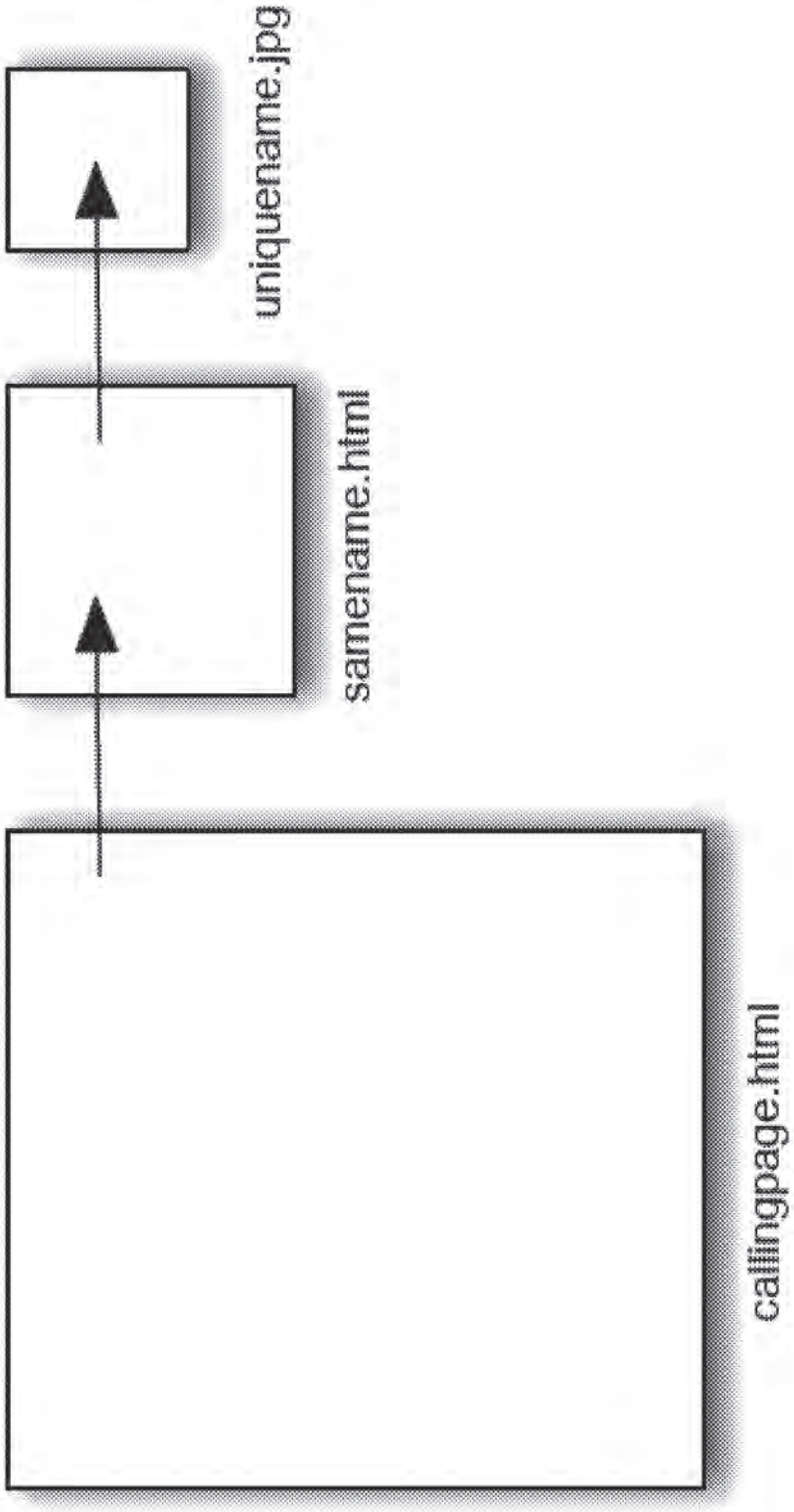


FIG. 8

AUTHENTICATION TRANSLATION

CROSS REFERENCE TO OTHER APPLICATIONS

This application is a continuation of co-pending U.S. patent application Ser. No. 16/563,715, entitled AUTHENTICATION TRANSLATION filed Sep. 6, 2019 which is incorporated herein by reference for all purposes, which is a is a continuation of U.S. patent application Ser. No. 16/273,797, entitled AUTHENTICATION TRANSLATION filed Feb. 12, 2019, now U.S. Pat. No. 10,521,568, which is incorporated herein by reference for all purposes, which is a is a continuation of U.S. patent application Ser. No. 15/042,636, entitled AUTHENTICATION TRANSLATION filed Feb. 12, 2016, now U.S. Pat. No. 10,360,351, which is incorporated herein by reference for all purposes, which is a continuation of U.S. patent application Ser. No. 13/706,254, entitled AUTHENTICATION TRANSLATION filed Dec. 5, 2012, now U.S. Pat. No. 9,294,452, which is incorporated herein by reference for all purposes, which claims priority to U.S. Provisional Application No. 61/587,387, entitled BIOMETRICS-SUPPORTED SECURE AUTHENTICATION SYSTEM filed Jan. 17, 2012 which is incorporated herein by reference for all purposes. U.S. patent application Ser. No. 13/706,254 also claims priority to U.S. Provisional Patent Application No. 61/569,112 entitled BACKWARDS COMPATIBLE ROBUST COOKIES filed Dec. 9, 2011, which is incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

Providing credentials to a service, whether via a mobile or other device, is often a tedious experience for a user. Unfortunately, to make authentication easier for themselves, users will often engage in practices such as password re-use, and/or the selection of poor quality passwords, which render their credentials less secure against attacks. Accordingly, improvements in authentication techniques would be desirable. Further, it would be desirable for such improvements to be widely deployable, including on existing/legacy systems.

BRIEF DESCRIPTION OF THE DRAWINGS

Various embodiments of the invention are disclosed in the following detailed description and the accompanying drawings.

FIG. **1** illustrates an embodiment of an environment in which authentication translation is provided.

FIG. **2** illustrates an embodiment of credential information stored on a device.

FIG. **3** illustrates an embodiment of a device with secure storage.

FIG. **4** illustrates an example of a renegotiation.

FIG. **5** illustrates an embodiment of a process for performing authentication translation.

FIG. **6** illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator.

FIG. **7** illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator.

FIG. **8** shows the structure of an example of a cache cookie used in some embodiments.

DETAILED DESCRIPTION

The invention can be implemented in numerous ways, including as a process; an apparatus; a system; a composition of matter; a computer program product embodied on a computer readable storage medium; and/or a processor, such as a processor configured to execute instructions stored on and/or provided by a memory coupled to the processor. In this specification, these implementations, or any other form that the invention may take, may be referred to as techniques. In general, the order of the steps of disclosed processes may be altered within the scope of the invention. Unless stated otherwise, a component such as a processor or a memory described as being configured to perform a task may be implemented as a general component that is temporarily configured to perform the task at a given time or a specific component that is manufactured to perform the task. As used herein, the term 'processor' refers to one or more devices, circuits, and/or processing cores configured to process data, such as computer program instructions.

A detailed description of one or more embodiments of the invention is provided below along with accompanying figures that illustrate the principles of the invention. The invention is described in connection with such embodiments, but the invention is not limited to any embodiment. The scope of the invention is limited only by the claims and the invention encompasses numerous alternatives, modifications and equivalents. Numerous specific details are set forth in the following description in order to provide a thorough understanding of the invention. These details are provided for the purpose of example and the invention may be practiced according to the claims without some or all of these specific details. For the purpose of clarity, technical material that is known in the technical fields related to the invention has not been described in detail so that the invention is not unnecessarily obscured.

FIG. **1** illustrates an embodiment of an environment in which authentication translation is provided. In the example shown, a variety of client devices **102-108** connect, via one or more networks (represented as a single network cloud **110**) to a variety of services **120-124** (also referred to herein as sites **120-124**). In particular, client device **102** is a notebook computer owned by a user hereinafter referred to as Alice. Notebook **102** includes a camera, a microphone, and a fingerprint sensor. Client device **104** is a smartphone, also owned by Alice. Client device **104** includes a camera. Client device **106** is a tablet owned by Bob, and sometimes used by Bob's son Charlie. Client device **106** includes a camera and a fingerprint sensor. Client device **108** is a kiosk located in the lobby of a hotel. Kiosk **108** includes a camera and a microphone. The techniques described herein can be used with or adapted to be used with other devices, as applicable. For example, the techniques can be used in conjunction with gaming systems, with peripheral devices such as mice, and with embedded devices, such as door locks.

Service **120** is a social networking site. Service **122** is a website of a bank. Service **124** is the online store of a boutique camera retailer. Each of services **120-124** requires a username and password (and/or a cookie) from a user prior to giving that user access to protected content and/or other features. As will be described in more detail below, using the techniques described herein, users need not type such usernames and passwords into their devices whenever required by a service. Instead, users can authenticate themselves to an "authentication translator" via an appropriate technique, and the authentication translator will provide the appropriate

credentials to the implicated service on the user's behalf. Also as will be described in more detail below, authentication translators can be located in a variety of places within an environment. For example, notebook computer **102** includes an authentication translator module **132** that provides authentication translation services. The other devices **104-108** can also include (but need not include) their own respective authentication translator modules. The owner of bank website **122** also operates an authentication translator **134** associated with the bank. Finally, authentication translator **136** provides authentication translation services to a variety of businesses, including online camera retailer **124**.

FIG. **2** illustrates an embodiment of credential information stored on a device. In particular, device **200** stores three user profiles **202-206**, each of which contains a username and one or more templates (e.g., template **210**) associated with the user. In various embodiments, a template is a collection of biometric features. Using fingerprints as an example type of biometric, a corresponding template includes a collection of patterns, minutia, and/or other features that can be matched against to determine if a person's fingerprint matches the fingerprint of the registered user (i.e., the owner of a given user profile). A representation of a single fingerprint may be included in multiple templates (e.g., in different resolutions, in accordance with different protocols, as captured during warm vs. cold conditions, and/or by itself or in combination with multiple fingerprints). When other biometrics are employed (e.g., facial recognition, voiceprint, or retina scan technology), features appropriate to those types of biometrics are included in the template. Other types of features can also be included in templates. As one example, a user's typing speed and/or accuracy can be measured by a device, such as device **102**, and used to distinguish between multiple users of a device. For example, suppose Alice types at 100 words per minute and rarely makes mistakes. A representation of this information can be stored in template **212**. Also suppose Alice's niece, who sometimes uses Alice's laptop computer when visiting Alice types at 20 words per minute and makes many mistakes. In some embodiments, the fact that a user was recently (e.g., within the last 5 minutes) typing on laptop **102** at 90 words per minute is evidence of a match against template **212**. In this case, the typing speed of 90 words per minute is similar enough to Alice's typical behavior, it is considered a match. Various policies can be included in a profile that govern how matches are to be performed. For example, policies can specify thresholds/tolerances for what constitutes a match, and can specify that different levels of matches can result in different levels of access to different resources.

A profile is associated with a vault (e.g., vault **220**). The vault, in turn, contains triples specifying a service provider/domain, a username, and a credential. The vault can also contain other sensitive user information, such as account numbers, address/phone number information, and health care data. The credential for a service provider/domain can be a password (e.g., for legacy servers), and can also take alternate forms (e.g., a cryptographic key for service providers supporting stronger authentication methods).

In some embodiments, profiles, templates, and vaults (collectively "authentication information") are stored entirely in an unprotected storage area, and are stored in the clear. In other embodiments, secure storage techniques are used to secure at least a portion of the authentication information.

One example of a device with secure storage is illustrated in FIG. **3**. In the example shown, a mobile phone **300** includes a large and insecure storage **302** attached to a fast processor **304**, and a smaller but secure storage **306** attached to a dedicated processor **308** and a sensor **310** (e.g., a camera or a fingerprint reader). Users (and applications) can read from and write to the insecure storage area. However, users cannot access the secure storage area, and the fast processor can only communicate with the dedicated processor/sensor via a restricted API. As another example, a unique decryption key associated with a given vault can be stored in a profile. The vault is an encrypted and authenticated container that can be stored on insecure storage, e.g., on the device, and also backed up (e.g., to a cloud storage service **140** or to an alternate form of external storage). As needed, authentication information or portions thereof can be loaded into secure storage and decrypted. For example, one can use AES to encrypt the files one by one, using a key stored on the secured storage. A message authentication technique, such as HMAC, can be used for authenticating the encrypted files to provide tamper prevention. Profiles and vaults can be updated while in secure storage; if this occurs, they are encrypted and MACed before being written back to the insecure storage, which may in turn propagate them to external backup storage. In yet other embodiments, profiles and vaults are stored entirely in secure storage, in plaintext, which allows them to be both read and written—and in particular, searched.

Example Transaction Types

A variety of transaction types can take place in the environment shown in FIG. **1**, examples of which are discussed in this section.

Initial Registration

In order to begin using the techniques described herein, users perform some form of initial registration. As one example, suppose Alice launches an enrollment program installed on laptop **102**. She uses the program to capture various biometric information (e.g., fingerprints, photographs of her face, etc.). A user profile is created for Alice, and the biometric information captured about her is encoded into a plurality of templates, such as templates **210** and **214**. In some embodiments, Alice is also explicitly asked to supply credential information for services she would like to use, such as by providing the domain name of social networking site **120**, along with her username and password for site **120**. In other embodiments, domain/username/credential information is at least passively captured on Alice's behalf and included in one or more vaults such as vault **220**. Credential information can also be important from a browser password manager already in use by Alice or other appropriate source. In some embodiments, Alice also registers with cloud storage service **140**, which will allow her to back up her authentication information and to synchronize it across her devices (e.g., **102** and **104**), as described in more detail below.

Other registration approaches can also be used. For example, registration can be integrated into the experience the first time a device is used. Thus, when Bob first turns on tablet **106**, he may be prompted to take a picture of his face (with a profile/templates being created in response). Similarly, the first time Charlie uses tablet **106**, the techniques described herein can be used to determine that Charlie does not yet have a profile (e.g., because none of the templates already present on tablet **106** match his biometrics) and Charlie can be prompted to enroll as a second user of the device.

Authentication

Suppose Alice wishes to authenticate to banking website **122**. Using a fingerprint reader incorporated into her laptop,

she performs a fingerprint scan, which causes her biometric features to be extracted and compared to any stored templates residing on her computer. If a match is found, an associated decryption key is selected, and the associated vault is loaded and decrypted. The vault is scanned for an entry that matches the selected service provider (i.e., website **122**). If a matching entry is found, the associated domain, username, and site credential are extracted from the vault. In some embodiments, the validity of the domain name mapping is verified at this point to harden the system against domain name poisoning. Next, a secure connection is established between Alice's computer and the service provider, and Alice is authenticated. For service providers supporting strong user authentication, mutual SSL can be used, for example. A variety of policies can be involved when performing matching. For example, to access certain domains, Alice's print may need only match template **210**. To access other domains, Alice may need to match multiple templates (e.g., both **210** and **214**). As another example, in order to access social networking site **120**, Alice may merely need to be sitting in front of her computer, which has an integrated webcam. Even in relatively low light conditions, a match can be performed against Alice's face and features stored in a template. However, in order to access bank website **122**, Alice may need a high quality photograph (i.e., requiring her to turn on a bright light) and may need to demonstrate liveness (e.g., by blinking or turning her head). As yet another example, other contextual information can be included in policies. For example, if Alice's IP address indicates she is in a country that she is not usually in, she may be required to match multiple templates (or match a template with more/better quality features) in order to access retailer **124**, as distinguished from when her IP address indicates she is at home.

In some embodiments, the biometric sensor used by a user may be a peripheral device (e.g., a mouse with an integrated fingerprint scanner that is connected to the user's primary device via USB). In such scenarios, the peripheral device may be responsible for storing at least a portion of authentication information and may perform at least some of the authentication tasks previously described as having been performed by Alice's computer. For example, instead of processors **304** and **308**, and storages **302** and **306** being collocated on a single device (e.g., laptop **102**), processor **304** and storage **302** may be present on a primary device, and processor **308** and storage **306** may be present on a peripheral device (e.g., that also includes a sensor, such as a fingerprint reader).

In such scenarios, once Alice's login to banking website **122** is successfully completed, the secure session can be handed over from the peripheral device to the primary device, in a way that does not allow the primary device retroactive access to the plaintext data of the transcripts exchanged between the peripheral device and the service provider. One way this can be accomplished is by renegotiating SSL keys between the peripheral device and the website, after which the newly negotiated key can be handed off from the peripheral device to the primary device. This avoids retroactive credential capture in a setting where the device is infected by malware.

An example of renegotiation is depicted in FIG. **4**. Specifically, after a user has successfully authenticated to a fingerprint reader, a login is performed to a service provider. Using the primary device (**404**) as a proxy, the peripheral fingerprint reader **402** negotiates a first SSL connection (**408**) with a service provider **406**, over which credentials are exchanged. The proxy then renegotiates SSL (**410**), which replaces the old key with a new one. The new key is disclosed to the device, which then seamlessly takes over the connection with the service provider and performs the transaction protected by the authentication. The credentials exchanged during the first SSL connection cannot be accessed by device **404**, since the key of the renegotiated session is independent of the key of the original session; this provides protection against malware residing on the device. Renegotiation can be used when the primary device **404** is believed to be in a safe state when performing the negotiation of the SSL connection, but it is not known whether it is in a safe state during the transaction protected by the authentication. Renegotiation can also be used when a secure component of the primary device **404** performs the negotiation of the SSL connection and another and potentially insecure component of the primary device **404** is involved in the transaction protected by the authentication.

FIG. **5** illustrates an embodiment of a process for performing authentication translation. The process begins at **502** when a request to access a resource is received, as is an authentication input. One example of the processing performed at **502** is as follows. Suppose Alice wishes to sign into social networking website **120**. She directs a web browser application installed on client **102** to the social networking website. Authentication translator module **132** recognizes, from the context of Alice's actions (e.g., that she is attempting to access site **120** with her browser) that she would like to access a particular resource. Authentication translator module **132** prompts Alice (e.g., by a popup message or via a sound) to provide biometric information (e.g., to use the integrated fingerprint reader on her laptop). In some embodiments, the translator module does not prompt Alice, for example, because Alice has been trained to provide biometric information automatically when attempting to access certain resources. In yet other embodiments, the translator module only prompts Alice if she fails to provide acceptable biometric information within a timeout period (e.g., 30 seconds).

Module **132** compares Alice's supplied biometric data to the templates stored on her computer. If a suitable match is found, and if an entry for site **120** is present in the applicable vault, at **504**, a previously stored credential associated with the resource is accessed. In particular, the username and password for the website, as stored in a vault, such as vault **220**, are retrieved from the vault.

Finally, at **506**, the credential is provided to the resource. For example, Alice's username and password for site **120** are transmitted to site **120** at **506**. The credential can be transmitted directly (e.g., by the module or by Alice's computer) and can also be supplied indirectly (e.g., through the use of one or more proxies, routers, or other intermediaries, as applicable).

Other devices can also make use of process **500** or portions thereof. For example, when Alice launches a banking application on phone **104**, implicit in her opening that application is her desire to access the resources of website **134**. The application can take Alice's picture and compare it to stored templates/vault information. If an appropriate match is found, a credential can be retrieved from the vault on her phone (or, e.g., retrieved from cloud storage service **140**) and provided to website **134**.

As another example, suppose Charlie is using tablet **106** and attempts to visit site **120**, whether via a dedicated application or via a web browser application installed on the tablet. Charlie's photograph is taken, and then compared against the profiles stored on tablet **106** (e.g., both Bob and Charlie's profiles). When a determination is made that

Charlie's photograph matches a template stored in his stored profile (and not, e.g., Bob's), Charlie's credentials for site **120** are retrieved from a vault and transmitted by an authentication translator module residing on client **106**.

As yet another example, kiosk **108** can be configured to provide certain local resources (e.g., by displaying a company directory or floor plan on demand) when users speak certain requests into a microphone. Enrolled users (e.g., with stored voiceprint or facial recognition features) can be granted access to additional/otherwise restricted services in accordance with the techniques described herein and process **500**.

New Device

In some embodiments, to register a new device, a user provides an identifier, such as a username or an account number to the device. The new device connects to an external storage (such as cloud storage **140**), provides the user identifier and credential, and downloads the user's templates/vaults from the service. In some embodiments, the templates/vaults are encrypted. Once downloaded, the template is decrypted and stored in a secure storage area, while the still encrypted vault can be stored in insecure storage. The decryption key can be generated from information the user has/knows, or from biometric data—such as features extracted from fingerprinting of all ten fingers. In some embodiments, more arduous fingerprinting is required for the setup of a new device than for regular authentication to avoid that a new device gets registered by a user thinking she is merely authenticating—or worse still, simply touching the device. Moreover, it translates into higher entropy of the decryption keys.

Backup Authentication

Backup authentication allows a user, such as Alice, to access resources in the event she is unable to or unwilling to interact with a particular biometric sensor. As one example, instead of having a single template associated with her profile, Alice can have multiple templates associated with it, e.g., where the first template includes fingerprint features and the second template includes voice biometric, facial recognition, or iris detection features. As a second example, where the service Alice is connecting to is a legacy website (i.e., one that users authenticate to using usernames and passwords), such a service would allow the use of passwords and password reset mechanisms by Alice without requiring Alice to use a fingerprint reader.

In various embodiments, environment **100** supports the ability of users (e.g., under duress) to release the contents of their vaults. For example, if Alice was physically threatened with the loss of a finger by a criminal, Alice could instead release the contents of her vault(s)—the ultimate goal of the criminal. As one example, in the event Alice supplies all 10 fingerprints to the sensor, provides a special password, or supplies a fingerprint and a second identifier, a cleartext version of her vault(s) could be made available.

Access Policies

In various embodiments, cloud storage service **140** is configured to accept backups from multiple devices associated with a single account, and synchronize the updates so that all devices get automatically refreshed. For example, Alice's laptop **102** and phone **104** could both communicate with cloud storage service **140** which would keep their authentication information synchronized. Refreshes can also be made in accordance with user-configured restrictions. For example, Alice's employer could prevent privileged employer data from being stored on shared personal devices, or on any device that was not issued by the employer. As another example, arbitrary policies can be defined regarding the access to and synchronization of software and data, and to tie a license or access rights to a person (and her fingerprint) rather than to a device. As yet another example, in some embodiments (e.g., where a device is made publicly available or otherwise shared by many users), no or a reduced amount of authentication information resides on a device, and at least a portion of authentication information is always retrieved from cloud storage service **140**.

Remote Wiping

Remote wiping of a user's authentication information (e.g., templates) can be used both to "unshare" previously shared devices (e.g., where Bob and Charlie both have user profiles on their shared tablet **106**), and to avoid that criminals with physical component access to lost devices gain access to templates and vault contents. In some embodiments, polices such as ones where a template self-wipes if it is not matched within a particular duration of time are supported. Since user data can be frequently backed up to the cloud storage, and recovered from this using the new device registration process, inconvenience to the user will be minimized.

Legacy Server Support

New authentication schemes typically require changes to a significant codebase residing with service providers. If the code is well written and documented, such changes may be relatively simple. Commonly, though, this may not be so. The engineers who originally wrote the code of relevance may have long since left the company; the code they left behind may be poorly documented—if documented at all. In severe cases, the legacy code may have been written in an outdated programming language or written in a way that does not follow guidelines for good code. This makes updates to the codebase impractical or virtually impossible in many common cases. Even if none of these challenges complicate the desired modifications, it is commonly a great bureaucratic burden to obtain permission to make modifications (e.g., to store an extra field in a backend database), because every affected part of the organization may need to review the request.

As will be described in the following section, the technologies described herein can be used in conjunction with legacy servers (e.g., existing servers that rely on usernames and passwords to authenticate users), and in particular, can be used without requiring modification to such legacy servers.

Cookies

Cookies are commonly used by legacy servers for user authentication. Unfortunately, cookies have several problems. For one thing, they are sometimes deleted—whether explicitly/intentionally by the end user or by the user's software. In addition, cookies are commonly stolen. Approaches such as cache cookies and identification using user agents can be more resistant to these problems, however, they have their own problems. For example, their use requires new code and new fields in the credential database stored by the server.

In some embodiments, authentication translators, such as translators **134** and **136** (also referred to herein as proxies) provide authentication translation services on behalf of associated services. Translators **134** and **136** are illustrated as single logical devices in FIG. **1**. In some embodiments, the translators comprise standard commercially available server hardware (e.g., a multi-core processor, 4+ Gigabytes of RAM, and one or more Gigabit network interface adapters) and run typical server-class operating systems (e.g., Linux). Translators **134** and **136** can also be implemented using a scalable, elastic architecture and may comprise

several distributed components, including components provided by one or more third parties. For example, translators **134** and **136** may store user credential information or may task cloud storage service **140** with storing at least a portion of that information.

In the case of authentication translator **134**, service is provided with respect to bank website **122** only. Authentication translator **134** is positioned between a legacy web server (**122**) and the Internet (**110**)—and therefore between the legacy server and any client devices. Authentication translator **134** is configured to translate traffic between the legacy server and client devices so that the client devices (and respective users) perceive the new authentication mechanism, while the legacy server remains unchanged. Authentication translator **136** works similarly, but it provides authentication translation as a third party service to multiple providers, an example of which is online camera retailer **124**.

Authentication translators **134** and **136** can also perform process **500**. For example, when a device transmits a request to access website **122**, the request is intercepted by translator **134**, as is cookie/user agent information. The received information can be used to determine a username/password associated with the device, and that information can be passed by translator **134** to website **122** on behalf of the device.

FIG. **6** illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator. The translator (referred to in the figure as a "proxy") fails to identify the client, and passes on the request to the legacy server. The legacy server responds to the request and sets a cookie. The proxy passes on the response, including the cookie and also a cache cookie. The proxy stores the information about both these types of identifiers, along with the user agent of the client device. This triplet of information is also referred to herein as an identifier.

FIG. **7** illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator. In this scenario, the device's request is accompanied by some form of identifying information. The proxy uses this information to identify the associated cookie and passes this along to the legacy server, along with the request. In some embodiments, additional processing is also involved. For example, reading a cache cookie may require user interaction. Moreover, if not all of the identifying information is present in the request, the proxy can be configured to set the missing information again by sending a corresponding request to the client device.

The translation of cache cookies and user agent information to cookies involves a two-way translation. First, when the legacy server sets a cookie, the proxy will set the two types of cookies—both an HTML cookie and a cache cookie—and then create a new record in which the two cookies are stored, along with the user agent information of the client device. The user agent information can include quite a bit of data associated with a browser—such as the browser type and version, the clock skew, and the fonts that are installed. While each of these pieces of information only contributes a small amount of entropy, the collection of items can be sufficient to identify the device in most cases. Moreover, while some of these types of data may change over time—in fact, all of them may—they do not typically change, and when one or two of them do, the others typically do not. When the client device is used to visit a site controlled by the legacy server, the cookie, cache cookie and user agent information are read (if available), the record identified, and the request translated and sent to the legacy server. When a legacy server requests that the user password is updated (e.g., as part of an annual or other periodic requirement), the transmission of this request to the user can be suppressed—in which case the database of the proxy is updated to create the illusion of an update. The user can be involved in authentication as needed, e.g., if, in addition to supplying a credential, a user must also solve a CAPTCHA, the CAPTCHA can be displayed to the user (with the user's credentials being handled in accordance with the techniques described herein).

FIG. **8** shows the structure of an example of a cache cookie used in some embodiments. Cache cookies can be associated with a particular webpage, just like an HTML cookie can. In the example shown in FIG. **8**, the proxy wishes to associate a page "calling page.html" with a cache cookie. It embeds a request for a second object, "samename.html" in callingpage.html for every visitor. However, as the cache cookie is set for one visitor, a customized samename.html is served to this visitor. The page samename.html refers to an object with a different name for each user; that object is referred to as "uniquename.jpg." The cache cookie is set by embedding the request for samename.html in callingpage.html. The client browser attempts to render this, causing a request for samename.html from the server. The server configures samename.html to refer to a uniquely named file uniquename.jpg, and serves samename.html to the client. For the client browser to render samename.html, it requests the file uniquename.jpg, which is intentionally not served. That concludes the setting of a cache cookie. As a user returns to the page callingname.html, the browser again attempts to render the entire page, which causes it to load the object samename.html from its cache. As that is rendered, the client browser requests uniquename.jpg, which is not in its cache (since it was not served previously). The server still does not serve it, but takes note of the name of the file being requested, as it identifies the client device. Note that samename.html can be displayed in a zero-sized iframe, which makes the end user unaware of it being rendered.

A cache cookie is an implementation of the typical cookie functionality that uses the client device's browser cache. Unlike user agents, it does not change over time, and like standard HTML cookies, it cannot be read by a party other than that which set it. However, like HTML cookies, it could be deleted—by the user clearing his or her browser cache. Cache cookies are not automatically transmitted with GET requests, unless the cache elements are embedded in the referring pages. This adds a potential round of communication in some settings. By relying on user agent information, cache cookies, and HTML cookies to identify the client device, it is much more likely that a machine will be recognized than if only HTML cookies are used.

Although the foregoing embodiments have been described in some detail for purposes of clarity of understanding, the invention is not limited to the details provided. There are many alternative ways of implementing the invention. The disclosed embodiments are illustrative and not restrictive.

What is claimed is:

1. A system, comprising:

a first processor of a first device, wherein the first processor is configured to:

based at least in part on a request associated with a user to access an external resource, establish a secure connection with the external resource; and

communicate with a second processor using a restricted interface, wherein the second processor is configured to:

receive a biometric input from a sensor, wherein the biometric input corresponds to at least one of a fingerprint, a feature usable for facial recognition, a voiceprint, a feature usable for a retina scan, or a typing feature; and

access a record stored in a secure storage, wherein the record is associated at least with the external resource;

retrieve, from the record, at least one of a password, a cookie, or a cryptographic key;

perform a cryptographic operation;

in response to determining that there is a match between the biometric input and a stored biometric template accessed by the second processor, facilitate a login of the user to the external resource at least in part by transmitting, via the established connection, output based at least in part on the at least one of the password, the cryptographic key, or the cookie retrieved from the record, and wherein the user is logged in to the external resource based at least in part on the output; and

perform a secure backup of the record to a storage service, wherein a second device associated with the user is registered with the storage service, and wherein the record is downloaded from the storage service by the second device; and

a memory coupled to the first processor and configured to provide the first processor with instructions.

2. The system recited in claim **1**, wherein the storage service comprises a cloud storage service.

3. The system recited in claim **1** wherein the second device is registered with the storage service at least in part by:

connecting to the storage service; and

providing at least one of a user identifier or a credential to the storage service, wherein the user identifier comprises at least one of a username or an account number.

4. The system recited in claim **1**, wherein the record downloaded by the second device is encrypted.

5. The system recited in claim **4**, wherein the encrypted record is stored to an insecure storage associated with the second device.

6. The system recited in claim **4**, wherein the second device is configured to decrypt the downloaded record and store the decrypted record to a secure storage associated with the second device.

7. The system recited in claim **6**, wherein the downloaded record is decrypted by the second device using a decryption key that is generated based at least in part on information associated with the user.

8. The system recited in claim **6**, wherein the downloaded record is decrypted by the second device using a decryption key that is generated based at least in part on biometric data.

9. The system recited in claim **6**, wherein the downloaded record is decrypted by the second device using a decryption key generated using one or more features extracted from fingerprinting.

10. The system recited in claim **1**, wherein performing the cryptographic operation comprises decrypting the record.

11. A method, comprising:

based at least in part on a request associated with a user to access an external resource, establishing, by a first processor of a first device, a secure connection with the external resource; and

communicating with a second processor using a restricted interface, wherein the second processor is configured to:

receive a biometric input from a sensor, wherein the biometric input corresponds to at least one of a fingerprint, a feature usable for facial recognition, a voiceprint, a feature usable for a retina scan, or a typing feature; and

access a record stored in a secure storage, wherein the record is associated at least with the external resource;

retrieve, from the record, at least one of a password, a cookie, or a cryptographic key;

perform a cryptographic operation;

in response to determining that there is a match between the biometric input and a stored biometric template accessed by the second processor, facilitate a login of the user to the external resource at least in part by transmitting, via the established connection, output based at least in part on the at least one of the password, the cryptographic key, or the cookie retrieved from the record, and wherein the user is logged in to the external resource based at least in part on the output; and

perform a secure backup of the record to a storage service, wherein a second device associated with the user is registered with the storage service, and wherein the record is downloaded from the storage service by the second device.

12. The method of claim **11**, wherein the storage service comprises a cloud storage service.

13. The method of claim **11**, wherein the second device is registered with the storage service at least in part by:

connecting to the storage service; and

providing at least one of a user identifier or a credential to the storage service, wherein the user identifier comprises at least one of a username or an account number.

14. The method of claim **11**, wherein the record downloaded by the second device is encrypted.

15. The method of claim **14**, wherein the second device is configured to decrypt the downloaded record and store the decrypted record to a secure storage associated with the second device.

16. The method of claim **14**, wherein the encrypted record is stored to an insecure storage associated with the second device.

17. The method of claim **15**, wherein the downloaded record is decrypted by the second device using a decryption key that is generated based at least in part on information associated with the user.

18. The method of claim **15**, wherein the downloaded record is decrypted by the second device using a decryption key that is generated based at least in part on biometric data.

19. The method of claim **15**, wherein the downloaded record is decrypted by the second device using a decryption key generated using one or more features extracted from fingerprinting.

20. The method of claim **11**, wherein performing the cryptographic operation comprises decrypting the record.

21. A computer program product embodied in a non-transitory computer readable storage medium and comprising computer instructions for:

based at least in part on a request associated with a user to access an external resource, establishing, by a first processor of a first device, a secure connection with the external resource; and

communicating with a second processor using a restricted interface, wherein the second processor is configured to:

receive a biometric input from a sensor, wherein the biometric input corresponds to at least one of a fingerprint, a feature usable for facial recognition, a voiceprint, a feature usable for a retina scan, or a typing feature; and

access a record stored in a secure storage, wherein the record is associated at least with the external resource;

retrieve, from the record, at least one of a password, a cookie, or a cryptographic key;

perform a cryptographic operation;

in response to determining that there is a match between the biometric input and a stored biometric template accessed by the second processor, facilitate a login of the user to the external resource at least in part by transmitting, via the established connection, output based at least in part on the at least one of the password, the cryptographic key, or the cookie retrieved from the record, and wherein the user is logged in to the external resource based at least in part on the output; and

perform a secure backup of the record to a storage service, wherein a second device associated with the user is registered with the storage service, and wherein the record is downloaded from the storage service by the second device.

* * * * *

Electronic Acknowledgement Receipt

EFS ID:	38412004
Application Number:	16773767
International Application Number:	
Confirmation Number:	4416
Title of Invention:	AUTHENTICATION TRANSLATION
First Named Inventor/Applicant Name:	Bjorn Markus Jakobsson
Customer Number:	21912
Filer:	Yeu-Ting George Cheng/Elaine Nguyen
Filer Authorized By:	Yeu-Ting George Cheng
Attorney Docket Number:	MJAKP008C4
Receipt Date:	27-JAN-2020
Filing Date:	
Time Stamp:	19:01:56
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	DA
Payment was successfully received in RAM	$835
RAM confirmation Number	E20201QJ02184805
Deposit Account	500685
Authorized User	Elaine Nguyen

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

37 CFR 1.16 (National application filing, search, and examination fees)

37 CFR 1.17 (Patent application and reexamination processing fees)

37 CFR 1.19 (Document supply fees)

37 CFR 1.20 (Post Issuance fees)

37 CFR 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Application Data Sheet	MJAKP008C4_ADS.pdf	1793883 3bb36ff2e19b955f4183ccb18bb662c95022 47d2	no	9
Warnings:					
Information:					
2	Oath or Declaration filed	MJAKP008C4_Executed_Dec.pdf	93450 8bbb681ee91b3a5261d972ce592786304 97ed0e	no	1
Warnings:					
Information:					
3	Power of Attorney	MJAKP008C4_POA_AIA82A.pdf	217717 d55c78ac140e3c523e742fff4560936fe79b7 96f	no	1
Warnings:					
Information:					
4	Power of Attorney	MJAK_Executed_POA_AIA82B_RightQuestionLLC.pdf	387899 a756b6df61aa99186ddcc71473ee0bc19a8 [illegible]	no	1
Warnings:					
Information:					
5	Transmittal Letter	MJAKP008C4_IDS_01_Transmittal.pdf	79378 5a54559654f9a6344f4189ae74c49d427002 f5c1	no	2
Warnings:					
Information:					
6	Information Disclosure Statement (IDS) Form (SB08)	MJAKP008C4_IDS_01_SB08.pdf	1054170 3b0364fe93d3e3840a69cca0ceaa1274a89a d624	no	6
Warnings:					
Information:					

7	Specification	MJAKP008C4_APP.pdf	173682 5890c166c7d367b14320cce40b0336e997b266e0	no	23
Warnings:					
Information:					
8	Drawings-only black and white line drawings	MJAKP008C4_APP_Figures.pdf	112090 8348797bd1fc2e79014265a447a54f43bdcc227	no	8
Warnings:					
Information:					
9	Fee Worksheet (SB06)	fee-info.pdf	36758 f25c2d9b1f6f13e3befccfa128aeaa7306bf48d	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	3949027		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.
National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.
New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Attorney Docket No. MJAKP008C4

APPLICATION FOR UNITED STATES PATENT

AUTHENTICATION TRANSLATION

By Inventor:

Bjorn Markus Jakobsson
Portola Valley, CA
A Citizen of Sweden

Assignee: RightQuestion, LLC

VAN PELT, YI & JAMES LLP
10050 N. Foothill Blvd., Suite 200
Cupertino, CA 95014
Telephone (408) 973-2585

AUTHENTICATION TRANSLATION

CROSS REFERENCE TO OTHER APPLICATIONS

[0001] This application is a continuation of co-pending U.S. Patent Application No. 16/563,715, entitled AUTHENTICATION TRANSLATION filed September 06, 2019 which is incorporated herein by reference for all purposes, which is a is a continuation of U.S. Patent Application No. 16/273,797, entitled AUTHENTICATION TRANSLATION filed February 12, 2019, now U.S. Patent No. 10,521,568, which is incorporated herein by reference for all purposes, which is a is a continuation of U.S. Patent Application No. 15/042,636, entitled AUTHENTICATION TRANSLATION filed February 12, 2016, now U.S. Patent No. 10,360,351, which is incorporated herein by reference for all purposes, which is a continuation of U.S. Patent Application No. 13/706,254, entitled AUTHENTICATION TRANSLATION filed December 5, 2012, now U.S. Patent No. 9,294,452, which is incorporated herein by reference for all purposes, which claims priority to U.S. Provisional Application No. 61/587,387, entitled BIOMETRICS-SUPPORTED SECURE AUTHENTICATION SYSTEM filed January 17, 2012 which is incorporated herein by reference for all purposes. U.S. Patent Application No. 13/706,254 also claims priority to U.S. Provisional Patent Application No. 61/569,112 entitled BACKWARDS COMPATIBLE ROBUST COOKIES filed December 9, 2011, which is incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

[0002] Providing credentials to a service, whether via a mobile or other device, is often a tedious experience for a user. Unfortunately, to make authentication easier for themselves, users will often engage in practices such as password re-use, and/or the selection of poor quality passwords, which render their credentials less secure against attacks. Accordingly, improvements in authentication techniques would be desirable. Further, it would be desirable for such improvements to be widely deployable, including on existing/legacy systems.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] Various embodiments of the invention are disclosed in the following detailed description and the accompanying drawings.

[0004] Figure 1 illustrates an embodiment of an environment in which authentication translation is provided.

[0005] Figure 2 illustrates an embodiment of credential information stored on a device.

[0006] Figure 3 illustrates an embodiment of a device with secure storage.

[0007] Figure 4 illustrates an example of a renegotiation.

[0008] Figure 5 illustrates an embodiment of a process for performing authentication translation.

[0009] Figure 6 illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator.

[0010] Figure 7 illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator.

[0011] Figure 8 shows the structure of an example of a cache cookie used in some embodiments.

DETAILED DESCRIPTION

[0012] The invention can be implemented in numerous ways, including as a process; an apparatus; a system; a composition of matter; a computer program product embodied on a computer readable storage medium; and/or a processor, such as a processor configured to execute instructions stored on and/or provided by a memory coupled to the processor. In this specification, these implementations, or any other form that the invention may take, may be referred to as techniques. In general, the order of the steps of disclosed processes may be altered within the scope of the invention. Unless stated otherwise, a component such as a processor or a memory described as being configured to perform a task may be implemented as a general component that is temporarily configured to perform the task at a given time or a specific component that is manufactured to perform the task. As used herein, the term 'processor' refers to one or more devices, circuits, and/or processing cores configured to process data, such as computer program instructions.

[0013] A detailed description of one or more embodiments of the invention is provided below along with accompanying figures that illustrate the principles of the invention. The invention is described in connection with such embodiments, but the invention is not limited to any embodiment. The scope of the invention is limited only by the claims and the invention encompasses numerous alternatives, modifications and equivalents. Numerous specific details are set forth in the following description in order to provide a thorough understanding of the invention. These details are provided for the purpose of example and the invention may be practiced according to the claims without some or all of these specific details. For the purpose of clarity, technical material that is known in the technical fields related to the invention has not been described in detail so that the invention is not unnecessarily obscured.

[0014] Figure 1 illustrates an embodiment of an environment in which authentication translation is provided. In the example shown, a variety of client devices 102-108 connect, via one or more networks (represented as a single network cloud 110) to a variety of services 120-124 (also referred to herein as sites 120-124). In particular, client device 102 is a notebook computer owned by a user hereinafter referred to as Alice. Notebook 102 includes a camera, a microphone, and a fingerprint sensor. Client device 104 is a smartphone, also owned by Alice.

Client device 104 includes a camera. Client device 106 is a tablet owned by Bob, and sometimes used by Bob's son Charlie. Client device 106 includes a camera and a fingerprint sensor. Client device 108 is a kiosk located in the lobby of a hotel. Kiosk 108 includes a camera and a microphone. The techniques described herein can be used with or adapted to be used with other devices, as applicable. For example, the techniques can be used in conjunction with gaming systems, with peripheral devices such as mice, and with embedded devices, such as door locks.

[0015] Service 120 is a social networking site. Service 122 is a website of a bank. Service 124 is the online store of a boutique camera retailer. Each of services 120-124 requires a username and password (and/or a cookie) from a user prior to giving that user access to protected content and/or other features. As will be described in more detail below, using the techniques described herein, users need not type such usernames and passwords into their devices whenever required by a service. Instead, users can authenticate themselves to an "authentication translator" via an appropriate technique, and the authentication translator will provide the appropriate credentials to the implicated service on the user's behalf. Also as will be described in more detail below, authentication translators can be located in a variety of places within an environment. For example, notebook computer 102 includes an authentication translator module 132 that provides authentication translation services. The other devices 104-108 can also include (but need not include) their own respective authentication translator modules. The owner of bank website 122 also operates an authentication translator 134 associated with the bank. Finally, authentication translator 136 provides authentication translation services to a variety of businesses, including online camera retailer 124.

[0016] Figure 2 illustrates an embodiment of credential information stored on a device. In particular, device 200 stores three user profiles 202-206, each of which contains a username and one or more templates (e.g., template 210) associated with the user. In various embodiments, a template is a collection of biometric features. Using fingerprints as an example type of biometric, a corresponding template includes a collection of patterns, minutia, and/or other features that can be matched against to determine if a person's fingerprint matches the fingerprint of the registered user (i.e., the owner of a given user profile). A representation of a single fingerprint may be included in multiple templates (e.g., in different resolutions, in accordance with different protocols, as captured during warm vs. cold conditions, and/or by itself

or in combination with multiple fingerprints). When other biometrics are employed (e.g., facial recognition, voiceprint, or retina scan technology), features appropriate to those types of biometrics are included in the template. Other types of features can also be included in templates. As one example, a user's typing speed and/or accuracy can be measured by a device, such as device 102, and used to distinguish between multiple users of a device. For example, suppose Alice types at 100 words per minute and rarely makes mistakes. A representation of this information can be stored in template 212. Also suppose Alice's niece, who sometimes uses Alice's laptop computer when visiting Alice types at 20 words per minute and makes many mistakes. In some embodiments, the fact that a user was recently (e.g., within the last 5 minutes) typing on laptop 102 at 90 words per minute is evidence of a match against template 212. In this case, the typing speed of 90 words per minute is similar enough to Alice's typical behavior, it is considered a match. Various policies can be included in a profile that govern how matches are to be performed. For example, policies can specify thresholds/tolerances for what constitutes a match, and can specify that different levels of matches can result in different levels of access to different resources.

[0017] A profile is associated with a vault (e.g., vault 220). The vault, in turn, contains triples specifying a service provider/domain, a username, and a credential. The vault can also contain other sensitive user information, such as account numbers, address/phone number information, and health care data. The credential for a service provider/domain can be a password (e.g., for legacy servers), and can also take alternate forms (e.g., a cryptographic key for service providers supporting stronger authentication methods).

[0018] In some embodiments, profiles, templates, and vaults (collectively "authentication information") are stored entirely in an unprotected storage area, and are stored in the clear. In other embodiments, secure storage techniques are used to secure at least a portion of the authentication information.

[0019] One example of a device with secure storage is illustrated in Figure 3. In the example shown, a mobile phone 300 includes a large and insecure storage 302 attached to a fast processor 304, and a smaller but secure storage 306 attached to a dedicated processor 308 and a sensor 310 (e.g., a camera or a fingerprint reader). Users (and applications) can read from and

write to the insecure storage area. However, users cannot access the secure storage area, and the fast processor can only communicate with the dedicated processor/sensor via a restricted API. As another example, a unique decryption key associated with a given vault can be stored in a profile. The vault is an encrypted and authenticated container that can be stored on insecure storage, e.g., on the device, and also backed up (e.g., to a cloud storage service 140 or to an alternate form of external storage). As needed, authentication information or portions thereof can be loaded into secure storage and decrypted. For example, one can use AES to encrypt the files one by one, using a key stored on the secured storage. A message authentication technique, such as HMAC, can be used for authenticating the encrypted files to provide tamper prevention. Profiles and vaults can be updated while in secure storage; if this occurs, they are encrypted and MACed before being written back to the insecure storage, which may in turn propagate them to external backup storage. In yet other embodiments, profiles and vaults are stored entirely in secure storage, in plaintext, which allows them to be both read and written -- and in particular, searched.

[0020] Example Transaction Types

[0021] A variety of transaction types can take place in the environment shown in Figure 1, examples of which are discussed in this section.

[0022] Initial Registration

[0023] In order to begin using the techniques described herein, users perform some form of initial registration. As one example, suppose Alice launches an enrollment program installed on laptop 102. She uses the program to capture various biometric information (e.g., fingerprints, photographs of her face, etc.). A user profile is created for Alice, and the biometric information captured about her is encoded into a plurality of templates, such as templates 210 and 214. In some embodiments, Alice is also explicitly asked to supply credential information for services she would like to use, such as by providing the domain name of social networking site 120, along with her username and password for site 120. In other embodiments, domain/username/credential information is at least passively captured on Alice's behalf and included in one or more vaults such as vault 220. Credential information can also be important from a browser password manager already in use by Alice or other appropriate source. In some

embodiments, Alice also registers with cloud storage service 140, which will allow her to back up her authentication information and to synchronize it across her devices (e.g., 102 and 104), as described in more detail below.

[0024] Other registration approaches can also be used. For example, registration can be integrated into the experience the first time a device is used. Thus, when Bob first turns on tablet 106, he may be prompted to take a picture of his face (with a profile/templates being created in response). Similarly, the first time Charlie uses tablet 106, the techniques described herein can be used to determine that Charlie does not yet have a profile (e.g., because none of the templates already present on tablet 106 match his biometrics) and Charlie can be prompted to enroll as a second user of the device.

[0025] Authentication

[0026] Suppose Alice wishes to authenticate to banking website 122. Using a fingerprint reader incorporated into her laptop, she performs a fingerprint scan, which causes her biometric features to be extracted and compared to any stored templates residing on her computer. If a match is found, an associated decryption key is selected, and the associated vault is loaded and decrypted. The vault is scanned for an entry that matches the selected service provider (i.e., website 122). If a matching entry is found, the associated domain, username, and site credential are extracted from the vault. In some embodiments, the validity of the domain name mapping is verified at this point to harden the system against domain name poisoning. Next, a secure connection is established between Alice's computer and the service provider, and Alice is authenticated. For service providers supporting strong user authentication, mutual SSL can be used, for example. A variety of policies can be involved when performing matching. For example, to access certain domains, Alice's print may need only match template 210. To access other domains, Alice may need to match multiple templates (e.g., both 210 and 214). As another example, in order to access social networking site 120, Alice may merely need to be sitting in front of her computer, which has an integrated webcam. Even in relatively low light conditions, a match can be performed against Alice's face and features stored in a template. However, in order to access bank website 122, Alice may need a high quality photograph (i.e., requiring her to turn on a bright light) and may need to demonstrate liveness (e.g., by blinking or turning her

head). As yet another example, other contextual information can be included in policies. For example, if Alice's IP address indicates she is in a country that she is not usually in, she may be required to match multiple templates (or match a template with more/better quality features) in order to access retailer 124, as distinguished from when her IP address indicates she is at home.

[0027] In some embodiments, the biometric sensor used by a user may be a peripheral device (e.g., a mouse with an integrated fingerprint scanner that is connected to the user's primary device via USB). In such scenarios, the peripheral device may be responsible for storing at least a portion of authentication information and may perform at least some of the authentication tasks previously described as having been performed by Alice's computer. For example, instead of processors 304 and 308, and storages 302 and 306 being collocated on a single device (e.g., laptop 102), processor 304 and storage 302 may be present on a primary device, and processor 308 and storage 306 may be present on a peripheral device (e.g., that also includes a sensor, such as a fingerprint reader).

[0028] In such scenarios, once Alice's login to banking website 122 is successfully completed, the secure session can be handed over from the peripheral device to the primary device, in a way that does not allow the primary device retroactive access to the plaintext data of the transcripts exchanged between the peripheral device and the service provider. One way this can be accomplished is by renegotiating SSL keys between the peripheral device and the website, after which the newly negotiated key can be handed off from the peripheral device to the primary device. This avoids retroactive credential capture in a setting where the device is infected by malware.

[0029] An example of renegotiation is depicted in Figure 4. Specifically, after a user has successfully authenticated to a fingerprint reader, a login is performed to a service provider. Using the primary device (404) as a proxy, the peripheral fingerprint reader 402 negotiates a first SSL connection (408) with a service provider 406, over which credentials are exchanged. The proxy then renegotiates SSL (410), which replaces the old key with a new one. The new key is disclosed to the device, which then seamlessly takes over the connection with the service provider and performs the transaction protected by the authentication. The credentials exchanged during the first SSL connection cannot be accessed by device 404, since the key of the

renegotiated session is independent of the key of the original session; this provides protection against malware residing on the device. Renegotiation can be used when the primary device 404 is believed to be in a safe state when performing the negotiation of the SSL connection, but it is not known whether it is in a safe state during the transaction protected by the authentication. Renegotiation can also be used when a secure component of the primary device 404 performs the negotiation of the SSL connection and another and potentially insecure component of the primary device 404 is involved in the transaction protected by the authentication.

[0030] Figure 5 illustrates an embodiment of a process for performing authentication translation. The process begins at 502 when a request to access a resource is received, as is an authentication input. One example of the processing performed at 502 is as follows. Suppose Alice wishes to sign into social networking website 120. She directs a web browser application installed on client 102 to the social networking website. Authentication translator module 132 recognizes, from the context of Alice's actions (e.g., that she is attempting to access site 120 with her browser) that she would like to access a particular resource. Authentication translator module 132 prompts Alice (e.g., by a popup message or via a sound) to provide biometric information (e.g., to use the integrated fingerprint reader on her laptop). In some embodiments, the translator module does not prompt Alice, for example, because Alice has been trained to provide biometric information automatically when attempting to access certain resources. In yet other embodiments, the translator module only prompts Alice if she fails to provide acceptable biometric information within a timeout period (e.g., 30 seconds).

[0031] Module 132 compares Alice's supplied biometric data to the templates stored on her computer. If a suitable match is found, and if an entry for site 120 is present in the applicable vault, at 504, a previously stored credential associated with the resource is accessed. In particular, the username and password for the website, as stored in a vault, such as vault 220, are retrieved from the vault.

[0032] Finally, at 506, the credential is provided to the resource. For example, Alice's username and password for site 120 are transmitted to site 120 at 506. The credential can be transmitted directly (e.g., by the module or by Alice's computer) and can also be supplied

indirectly (e.g., through the use of one or more proxies, routers, or other intermediaries, as applicable).

[0033] Other devices can also make use of process 500 or portions thereof. For example, when Alice launches a banking application on phone 104, implicit in her opening that application is her desire to access the resources of website 134. The application can take Alice's picture and compare it to stored templates/vault information. If an appropriate match is found, a credential can be retrieved from the vault on her phone (or, e.g., retrieved from cloud storage service 140) and provided to website 134.

[0034] As another example, suppose Charlie is using tablet 106 and attempts to visit site 120, whether via a dedicated application or via a web browser application installed on the tablet. Charlie's photograph is taken, and then compared against the profiles stored on tablet 106 (e.g., both Bob and Charlie's profiles). When a determination is made that Charlie's photograph matches a template stored in his stored profile (and not, e.g., Bob's), Charlie's credentials for site 120 are retrieved from a vault and transmitted by an authentication translator module residing on client 106.

[0035] As yet another example, kiosk 108 can be configured to provide certain local resources (e.g., by displaying a company directory or floor plan on demand) when users speak certain requests into a microphone. Enrolled users (e.g., with stored voiceprint or facial recognition features) can be granted access to additional/otherwise restricted services in accordance with the techniques described herein and process 500.

[0036] New device

[0037] In some embodiments, to register a new device, a user provides an identifier, such as a username or an account number to the device. The new device connects to an external storage (such as cloud storage 140), provides the user identifier and credential, and downloads the user's templates/vaults from the service. In some embodiments, the templates/vaults are encrypted. Once downloaded, the template is decrypted and stored in a secure storage area, while the still encrypted vault can be stored in insecure storage. The decryption key can be generated from information the user has/knows, or from biometric data -- such as features

extracted from fingerprinting of all ten fingers. In some embodiments, more arduous fingerprinting is required for the setup of a new device than for regular authentication to avoid that a new device gets registered by a user thinking she is merely authenticating -- or worse still, simply touching the device. Moreover, it translates into higher entropy of the decryption keys.

[0038] Backup Authentication

[0039] Backup authentication allows a user, such as Alice, to access resources in the event she is unable to or unwilling to interact with a particular biometric sensor. As one example, instead of having a single template associated with her profile, Alice can have multiple templates associated with it, e.g., where the first template includes fingerprint features and the second template includes voice biometric, facial recognition, or iris detection features. As a second example, where the service Alice is connecting to is a legacy website (i.e., one that users authenticate to using usernames and passwords), such a service would allow the use of passwords and password reset mechanisms by Alice without requiring Alice to use a fingerprint reader.

[0040] In various embodiments, environment 100 supports the ability of users (e.g., under duress) to release the contents of their vaults. For example, if Alice was physically threatened with the loss of a finger by a criminal, Alice could instead release the contents of her vault(s) – the ultimate goal of the criminal. As one example, in the event Alice supplies all 10 fingerprints to the sensor, provides a special password, or supplies a fingerprint and a second identifier, a cleartext version of her vault(s) could be made available.

[0041] Access Policies

[0042] In various embodiments, cloud storage service 140 is configured to accept backups from multiple devices associated with a single account, and synchronize the updates so that all devices get automatically refreshed. For example, Alice's laptop 102 and phone 104 could both communicate with cloud storage service 140 which would keep their authentication information synchronized. Refreshes can also be made in accordance with user-configured restrictions. For example, Alice's employer could prevent privileged employer data from being stored on shared personal devices, or on any device that was not issued by the employer. As

another example, arbitrary policies can be defined regarding the access to and synchronization of software and data, and to tie a license or access rights to a person (and her fingerprint) rather than to a device. As yet another example, in some embodiments (e.g., where a device is made publicly available or otherwise shared by many users), no or a reduced amount of authentication information resides on a device, and at least a portion of authentication information is always retrieved from cloud storage service 140.

[0043] Remote wiping

[0044] Remote wiping of a user's authentication information (e.g., templates) can be used both to "unshare" previously shared devices (e.g., where Bob and Charlie both have user profiles on their shared tablet 106), and to avoid that criminals with physical component access to lost devices gain access to templates and vault contents. In some embodiments, polices such as ones where a template self-wipes if it is not matched within a particular duration of time are supported. Since user data can be frequently backed up to the cloud storage, and recovered from this using the new device registration process, inconvenience to the user will be minimized.

[0045] **Legacy Server Support**

[0046] New authentication schemes typically require changes to a significant codebase residing with service providers. If the code is well written and documented, such changes may be relatively simple. Commonly, though, this may not be so. The engineers who originally wrote the code of relevance may have long since left the company; the code they left behind may be poorly documented -- if documented at all. In severe cases, the legacy code may have been written in an outdated programming language or written in a way that does not follow guidelines for good code. This makes updates to the codebase impractical or virtually impossible in many common cases. Even if none of these challenges complicate the desired modifications, it is commonly a great bureaucratic burden to obtain permission to make modifications (e.g., to store an extra field in a backend database), because every affected part of the organization may need to review the request.

[0047] As will be described in the following section, the technologies described herein can be used in conjunction with legacy servers (e.g., existing servers that rely on usernames and

passwords to authenticate users), and in particular, can be used without requiring modification to such legacy servers.

[0048] Cookies

[0049] Cookies are commonly used by legacy servers for user authentication. Unfortunately, cookies have several problems. For one thing, they are sometimes deleted -- whether explicitly/intentionally by the end user or by the user's software. In addition, cookies are commonly stolen. Approaches such as cache cookies and identification using user agents can be more resistant to these problems, however, they have their own problems. For example, their use requires new code and new fields in the credential database stored by the server.

[0050] In some embodiments, authentication translators, such as translators 134 and 136 (also referred to herein as proxies) provide authentication translation services on behalf of associated services. Translators 134 and 136 are illustrated as single logical devices in Figure 1. In some embodiments, the translators comprise standard commercially available server hardware (e.g., a multi-core processor, 4+ Gigabytes of RAM, and one or more Gigabit network interface adapters) and run typical server-class operating systems (e.g., Linux). Translators 134 and 136 can also be implemented using a scalable, elastic architecture and may comprise several distributed components, including components provided by one or more third parties. For example, translators 134 and 136 may store user credential information or may task cloud storage service 140 with storing at least a portion of that information.

[0051] In the case of authentication translator 134, service is provided with respect to bank website 122 only. Authentication translator 134 is positioned between a legacy web server (122) and the Internet (110) -- and therefore between the legacy server and any client devices. Authentication translator 134 is configured to translate traffic between the legacy server and client devices so that the client devices (and respective users) perceive the new authentication mechanism, while the legacy server remains unchanged. Authentication translator 136 works similarly, but it provides authentication translation as a third party service to multiple providers, an example of which is online camera retailer 124.

[0052] Authentication translators 134 and 136 can also perform process 500. For example, when a device transmits a request to access website 122, the request is intercepted by translator 134, as is cookie/user agent information. The received information can be used to determine a username/password associated with the device, and that information can be passed by translator 134 to website 122 on behalf of the device.

[0053] Figure 6 illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator. The translator (referred to in the figure as a "proxy") fails to identify the client, and passes on the request to the legacy server. The legacy server responds to the request and sets a cookie. The proxy passes on the response, including the cookie and also a cache cookie. The proxy stores the information about both these types of identifiers, along with the user agent of the client device. This triplet of information is also referred to herein as an identifier.

[0054] Figure 7 illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator. In this scenario, the device's request is accompanied by some form of identifying information. The proxy uses this information to identify the associated cookie and passes this along to the legacy server, along with the request. In some embodiments, additional processing is also involved. For example, reading a cache cookie may require user interaction. Moreover, if not all of the identifying information is present in the request, the proxy can be configured to set the missing information again by sending a corresponding request to the client device.

[0055] The translation of cache cookies and user agent information to cookies involves a two-way translation. First, when the legacy server sets a cookie, the proxy will set the two types of cookies -- both an HTML cookie and a cache cookie -- and then create a new record in which the two cookies are stored, along with the user agent information of the client device. The user agent information can include quite a bit of data associated with a browser -- such as the browser type and version, the clock skew, and the fonts that are installed. While each of these pieces of information only contributes a small amount of entropy, the collection of items can be sufficient to identify the device in most cases. Moreover, while some of these types of data may change over time -- in fact, all of them may -- they do not typically change, and when one or two of

them do, the others typically do not. When the client device is used to visit a site controlled by the legacy server, the cookie, cache cookie and user agent information are read (if available), the record identified, and the request translated and sent to the legacy server. When a legacy server requests that the user password is updated (e.g., as part of an annual or other periodic requirement), the transmission of this request to the user can be suppressed -- in which case the database of the proxy is updated to create the illusion of an update. The user can be involved in authentication as needed, e.g., if, in addition to supplying a credential, a user must also solve a CAPTCHA, the CAPTCHA can be displayed to the user (with the user's credentials being handled in accordance with the techniques described herein).

[0056] Figure 8 shows the structure of an example of a cache cookie used in some embodiments. Cache cookies can be associated with a particular webpage, just like an HTML cookie can. In the example shown in Figure 8, the proxy wishes to associate a page "calling page.html" with a cache cookie. It embeds a request for a second object, "samename.html" in callingpage.html for every visitor. However, as the cache cookie is set for one visitor, a customized samename.html is served to this visitor. The page samename.html refers to an object with a different name for each user; that object is referred to as "uniquename.jpg." The cache cookie is set by embedding the request for samename.html in callingpage.html. The client browser attempts to render this, causing a request for samename.html from the server. The server configures samename.html to refer to a uniquely named file uniquename.jpg, and serves samename.html to the client. For the client browser to render samename.html, it requests the file uniquename.jpg, which is intentionally not served. That concludes the setting of a cache cookie. As a user returns to the page callingname.html, the browser again attempts to render the entire page, which causes it to load the object samename.html from its cache. As that is rendered, the client browser requests uniquename.jpg, which is not in its cache (since it was not served previously). The server still does not serve it, but takes note of the name of the file being requested, as it identifies the client device. Note that samename.html can be displayed in a zero-sized iframe, which makes the end user unaware of it being rendered.

[0057] A cache cookie is an implementation of the typical cookie functionality that uses the client device's browser cache. Unlike user agents, it does not change over time, and like standard HTML cookies, it cannot be read by a party other than that which set it. However, like

HTML cookies, it could be deleted -- by the user clearing his or her browser cache. Cache cookies are not automatically transmitted with GET requests, unless the cache elements are embedded in the referring pages. This adds a potential round of communication in some settings. By relying on user agent information, cache cookies, and HTML cookies to identify the client device, it is much more likely that a machine will be recognized than if only HTML cookies are used.

[0058] Although the foregoing embodiments have been described in some detail for purposes of clarity of understanding, the invention is not limited to the details provided. There are many alternative ways of implementing the invention. The disclosed embodiments are illustrative and not restrictive.

[0059] WHAT IS CLAIMED IS:

CLAIMS

1. A system, comprising:

a first processor of a first device, wherein the first processor is configured to:

based at least in part on a request associated with a user to access an external resource, establish a secure connection with the external resource; and

communicate with a second processor using a restricted interface, wherein the second processor is configured to:

receive a biometric input from a sensor, wherein the biometric input corresponds to at least one of a fingerprint, a feature usable for facial recognition, a voiceprint, a feature usable for a retina scan, and a typing feature; and

access a record stored in a secure storage, wherein the record is associated at least with the external resource;

retrieve, from the record, at least one of a password, a cookie, and a cryptographic key;

perform a cryptographic operation;

in response to determining that there is a match between the biometric input and a stored biometric template accessed by the second processor, facilitate a login of the user to the external resource at least in part by transmitting, via the established connection, output based at least in part on the at least one of the password, the cryptographic key, and the cookie retrieved from the record, and wherein the user is logged in to the external resource based at least in part on the output; and

perform a secure backup of the record to a storage service, wherein a second device associated with the user is registered with the storage service, and wherein the record is downloaded from the storage service by the second device; and

a memory coupled to the first processor and configured to provide the first processor with instructions.

2. The system recited in claim 1, wherein the storage service comprises a cloud storage service.

3. The system recited in claim 1 wherein the second device is registered with the storage service at least in part by:

connecting to the storage service; and

providing at least one of a user identifier and a credential to the storage service, wherein the user identifier comprises at least one of a username and an account number.

4. The system recited in claim 1, wherein the record downloaded by the second device is encrypted.

5. The system recited in claim 4, wherein the second device is configured to decrypt the downloaded record and store the decrypted record to a secure storage associated with the second device.

6. The system recited in claim 5, wherein the downloaded record is decrypted by the second device using a decryption key that is generated based at least in part on information associated with the user.

7. The system recited in claim 5, wherein the downloaded record is decrypted by the second device using a decryption key that is generated based at least in part on biometric data.

8. The system recited in claim 5, wherein the downloaded record is decrypted by the second device using a decryption key generated using one or more features extracted from fingerprinting.

9. The system recited in claim 4, wherein the encrypted record is stored to an insecure storage associated with the second device.

10. The system recited in claim 1, wherein performing the cryptographic operation comprises decrypting the record.

11. A method, comprising:

based at least in part on a request associated with a user to access an external resource, establishing, by a first processor of a first device, a secure connection with the external resource; and

communicating with a second processor using a restricted interface, wherein the second processor is configured to:

receive a biometric input from a sensor, wherein the biometric input corresponds to at least one of a fingerprint, a feature usable for facial recognition, a voiceprint, a feature usable for a retina scan, and a typing feature; and

access a record stored in a secure storage, wherein the record is associated at least with the external resource;

retrieve, from the record, at least one of a password, a cookie, and a cryptographic key;

perform a cryptographic operation;

in response to determining that there is a match between the biometric input and a stored biometric template accessed by the second processor, facilitate a login of the user to the external resource at least in part by transmitting, via the established connection, output based at least in part on the at least one of the password, the cryptographic key, and the cookie retrieved from the record, and wherein the user is logged in to the external resource based at least in part on the output; and

perform a secure backup of the record to a storage service, wherein a second device associated with the user is registered with the storage service, and wherein the record is downloaded from the storage service by the second device.

12. The system recited in claim 1, wherein the storage service comprises a cloud storage service.

13. The system recited in claim 1 wherein the second device is registered with the storage service at least in part by:

connecting to the storage service; and

providing at least one of a user identifier and a credential to the storage service, wherein the user identifier comprises at least one of a username and an account number.

14. The system recited in claim 1, wherein the record downloaded by the second device is encrypted.

15. The system recited in claim 4, wherein the second device is configured to decrypt the downloaded record and store the decrypted record to a secure storage associated with the second device.

16. The system recited in claim 5, wherein the downloaded record is decrypted by the second device using a decryption key that is generated based at least in part on information associated with the user.

17. The system recited in claim 5, wherein the downloaded record is decrypted by the second device using a decryption key that is generated based at least in part on biometric data.

18. The system recited in claim 5, wherein the downloaded record is decrypted by the second device using a decryption key generated using one or more features extracted from fingerprinting.

19. The system recited in claim 4, wherein the encrypted record is stored to an insecure storage associated with the second device.

20. The system recited in claim 1, wherein performing the cryptographic operation comprises decrypting the record.

21. A computer program product embodied in a non-transitory computer readable storage medium and comprising computer instructions for:

based at least in part on a request associated with a user to access an external resource, establishing, by a first processor of a first device, a secure connection with the external resource; and

communicating with a second processor using a restricted interface, wherein the second processor is configured to:

receive a biometric input from a sensor, wherein the biometric input corresponds to at least one of a fingerprint, a feature usable for facial recognition, a voiceprint, a feature usable for a retina scan, and a typing feature; and

access a record stored in a secure storage, wherein the record is associated at least with the external resource;

retrieve, from the record, at least one of a password, a cookie, and a cryptographic key;

perform a cryptographic operation;

in response to determining that there is a match between the biometric input and a stored biometric template accessed by the second processor, facilitate a login of the user to the external resource at least in part by transmitting, via the established connection, output based at least in part on the at least one of the password, the cryptographic key, and the cookie retrieved from the record, and wherein the user is logged in to the external resource based at least in part on the output; and

perform a secure backup of the record to a storage service, wherein a second device associated with the user is registered with the storage service, and wherein the record is downloaded from the storage service by the second device.

ABSTRACT OF THE DISCLOSURE

Authentication translation is disclosed. A request to access a resource is received at an authentication translator, as is an authentication input. The authentication input corresponds to at least one stored record. The stored record is associated at least with the resource. In response to the receiving, a previously stored credential associated with the resource is accessed. The credential is provided to the resource.

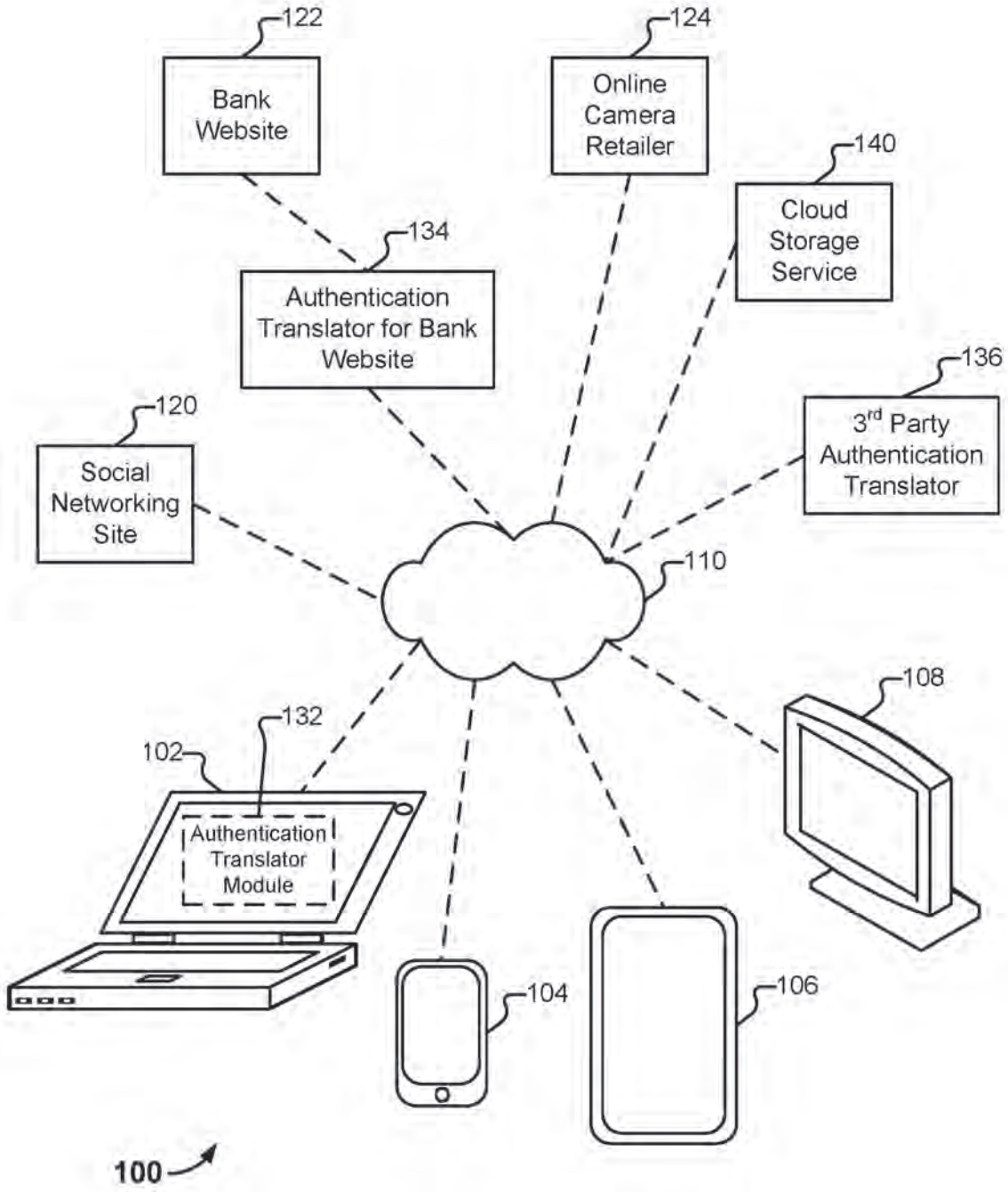

122
Bank Website
124
Online Camera Retailer
140
Cloud Storage Service
134
Authentication Translator for Bank Website
136
3rd Party Authentication Translator
120
Social Networking Site
110
108
132
102
Authentication Translator Module
104
106
100

FIG. 1

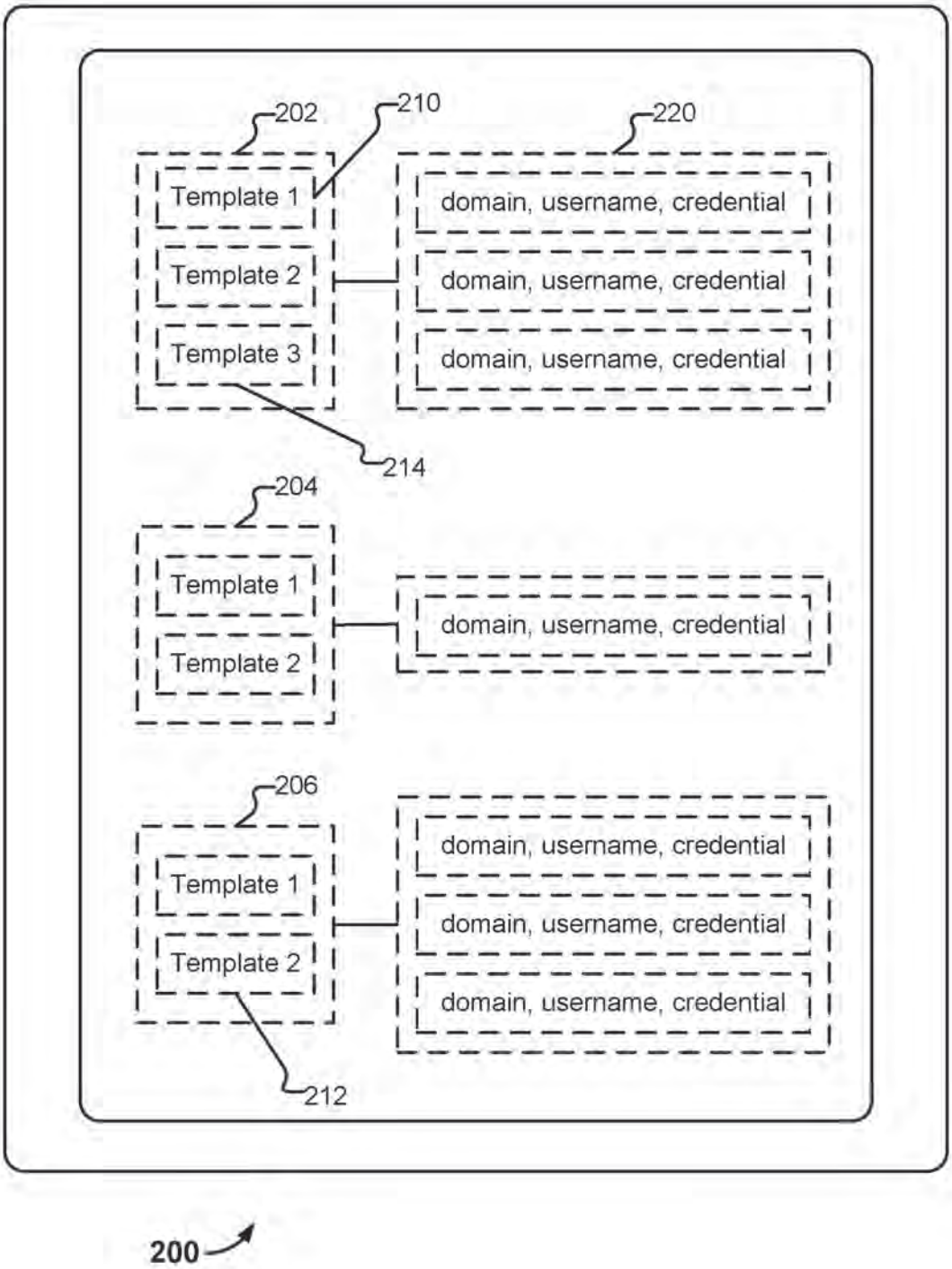
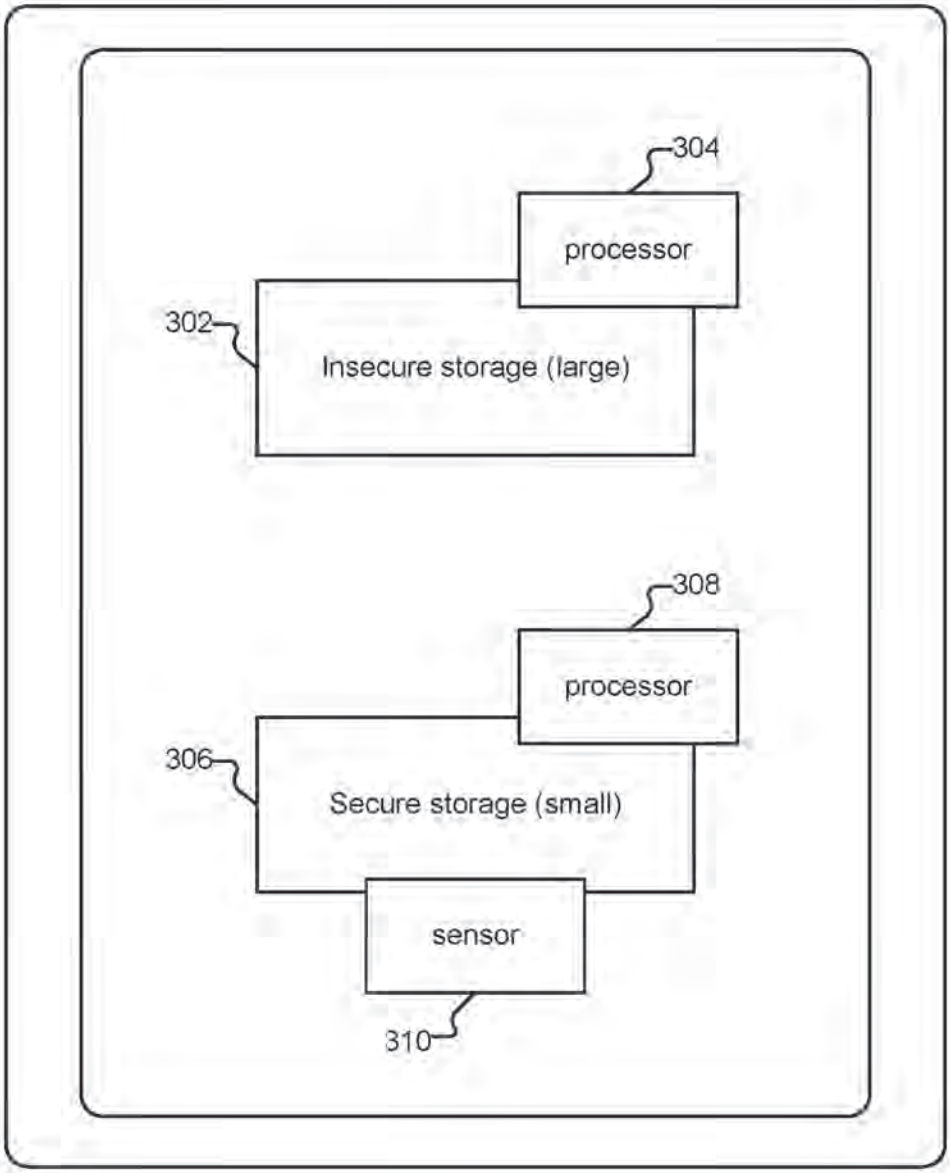

202
210
220
Template 1
Template 2
Template 3
domain, username, credential
domain, username, credential
domain, username, credential
214
204
Template 1
Template 2
domain, username, credential
206
Template 1
Template 2
domain, username, credential
domain, username, credential
domain, username, credential
212
200

FIG. 2


304
processor
302
Insecure storage (large)
308
processor
306
Secure storage (small)
sensor
310

300

FIG. 3

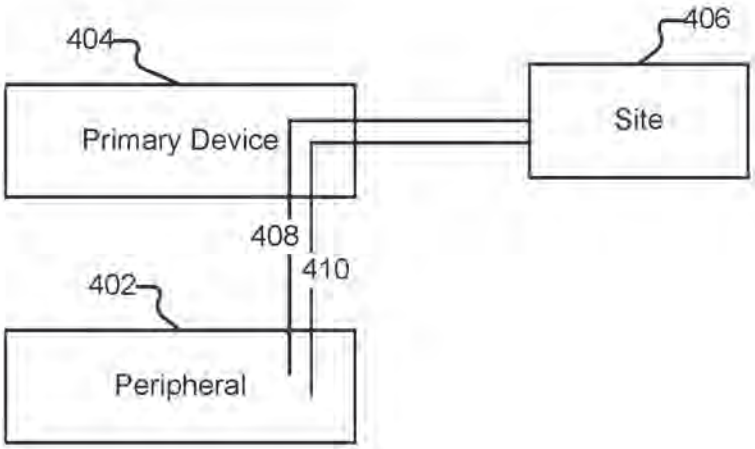


FIG. 4

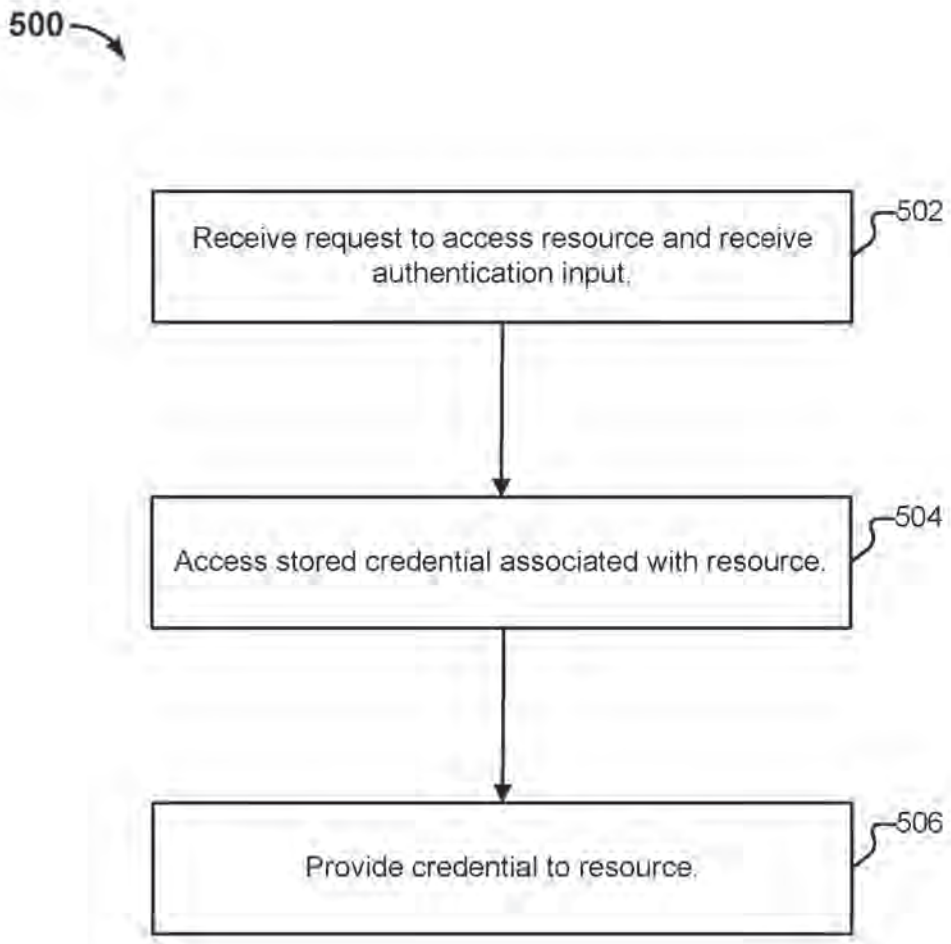

500
Receive request to access resource and receive authentication input.
502
Access stored credential associated with resource.
504
Provide credential to resource.
506

FIG. 5

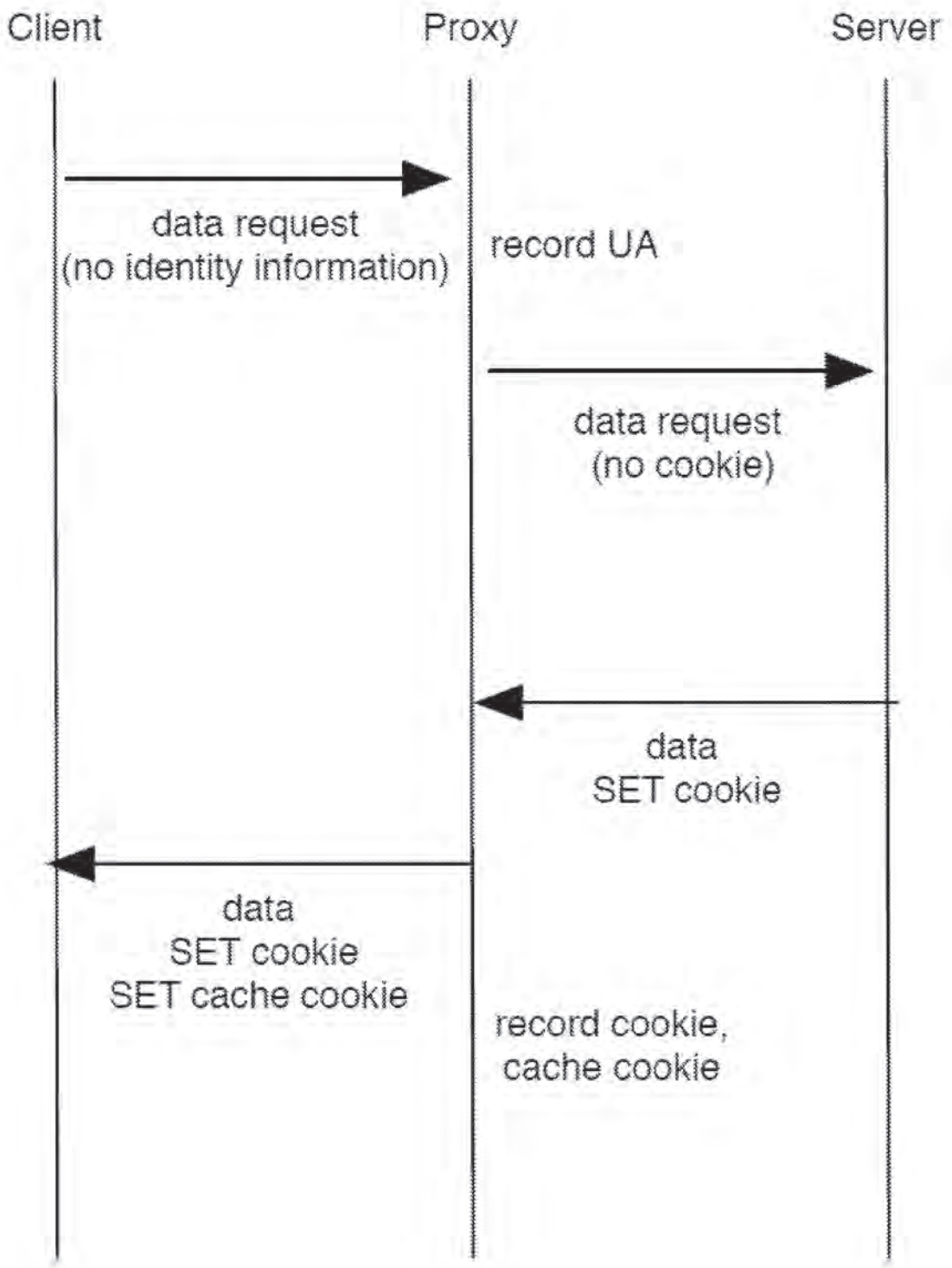

Client
Proxy
Server
data request
(no identity information)
record UA
data request
(no cookie)
data
SET cookie
data
SET cookie
SET cache cookie
record cookie,
cache cookie

FIG. 6

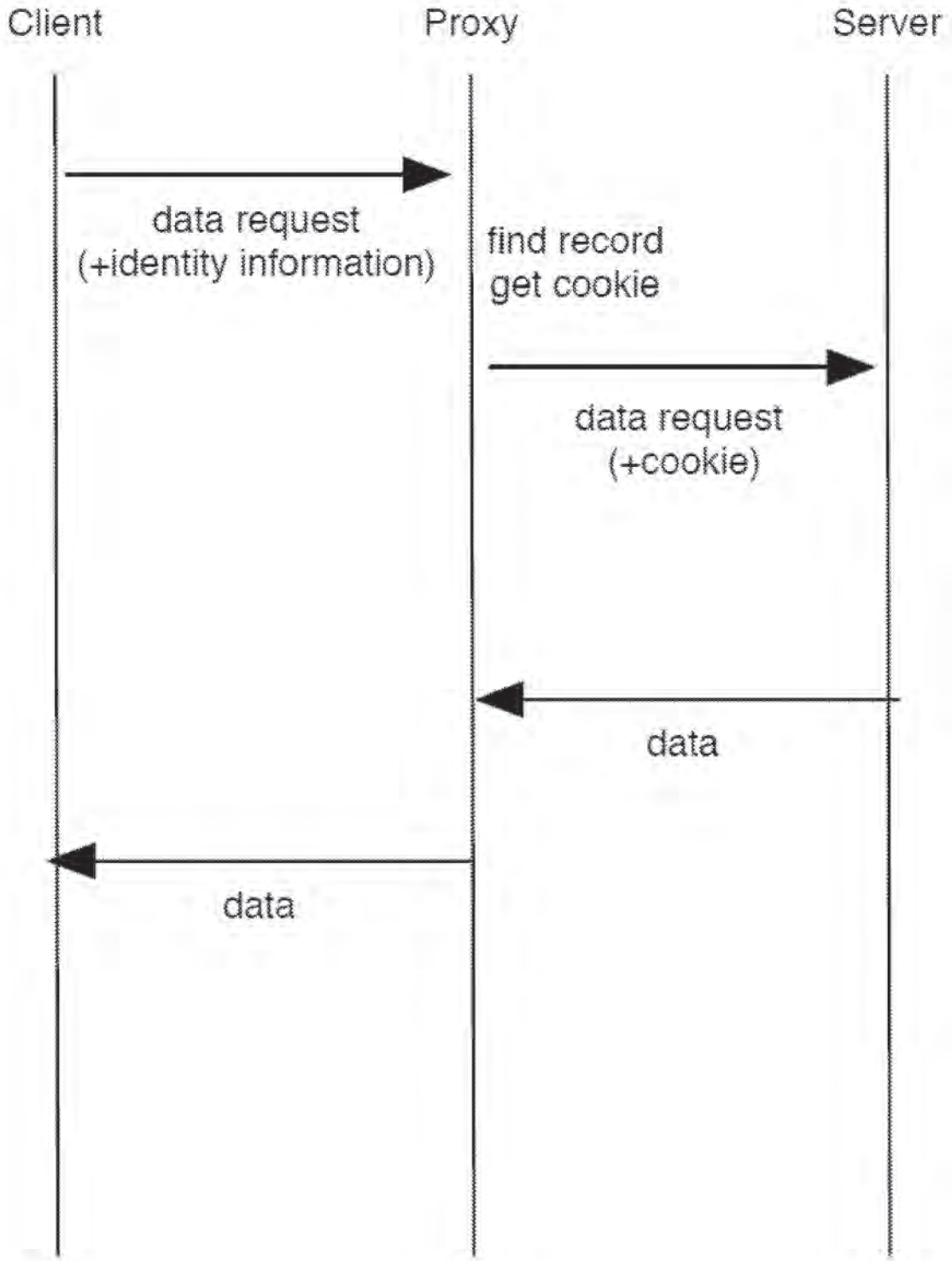

Client
Proxy
Server
data request
(+identity information)
find record
get cookie
data request
(+cookie)
data
data

FIG. 7

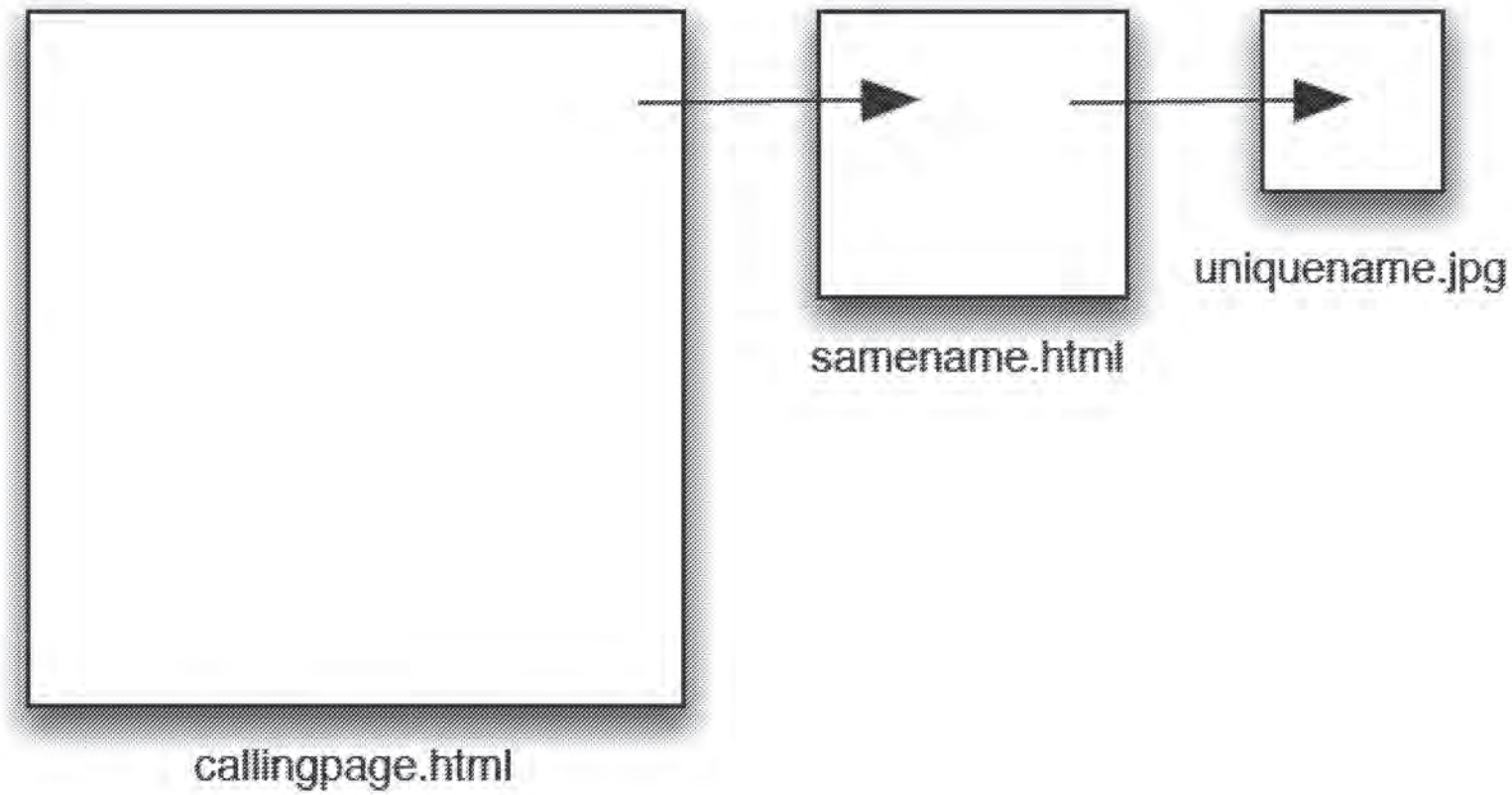


FIG. 8

Application Serial No. 16/563,715

Filing date: September 6, 2019

Patent No. 10,824,696

Issue date: November 3, 2020

US010824696B1

(12) **United States Patent**
Jakobsson

(10) **Patent No.: US 10,824,696 B1**
(45) **Date of Patent: *Nov. 3, 2020**

(54) **AUTHENTICATION TRANSLATION**

(71) Applicant: **RightQuestion, LLC**, Portola Valley, CA (US)

(72) Inventor: **Bjorn Markus Jakobsson**, Portola Valley, CA (US)

(73) Assignee: **RightQuestion, LLC**, Portola Valley, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **16/563,715**

(22) Filed: **Sep. 6, 2019**

Related U.S. Application Data

(63) Continuation of application No. 16/273,797, filed on Feb. 12, 2019, now Pat. No. 10,521,568, which is a continuation of application No. 15/042,636, filed on Feb. 12, 2016, now Pat. No. 10,360,351, which is a

(Continued)

(51) **Int. Cl.**

G06F 21/00	(2013.01)
G06F 21/10	(2013.01)
G06F 21/12	(2013.01)
H04L 29/06	(2006.01)
G06F 21/32	(2013.01)
G06F 21/31	(2013.01)
G06F 21/44	(2013.01)

(52) **U.S. Cl.**

CPC ***G06F 21/10*** (2013.01); ***G06F 21/121*** (2013.01); ***G06F 21/128*** (2013.01); ***G06F 21/31*** (2013.01); ***G06F 21/32*** (2013.01); ***G06F 21/44*** (2013.01); ***H04L 63/083*** (2013.01); ***H04L 63/0861*** (2013.01); ***H04L 63/10*** (2013.01); ***H04L 63/20*** (2013.01)

(58) **Field of Classification Search**

None

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,016,476	A	1/2000	Maes	
6,691,232	B1 *	2/2004	Wood	H04L 63/0815 726/6
7,512,965	B1	3/2009	Amdur	

(Continued)

FOREIGN PATENT DOCUMENTS

WO	2004051585	A2	6/2004
WO	2005001751	A1	1/2005

OTHER PUBLICATIONS

"Managing Authorization and Access Control", Author: unknown, Published Nov. 3, 2005, pp. 1-12, URL: http://technet.microsoft.com/en-us/library/bb457115.aspx.

(Continued)

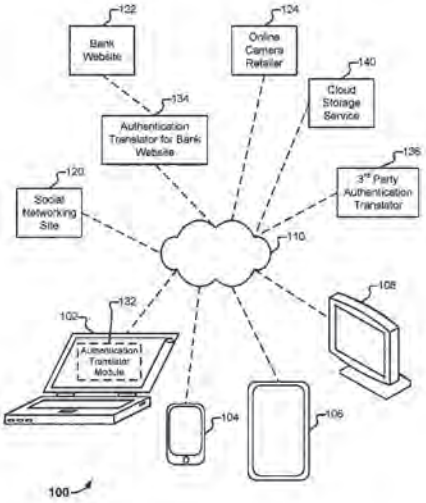
Primary Examiner — Andrew J Steinle

(74) *Attorney, Agent, or Firm* — Van Pelt, Yi & James LLP

(57) **ABSTRACT**

Authentication translation is disclosed. A request to access a resource is received at an authentication translator, as is an authentication input. The authentication input corresponds to at least one stored record. The stored record is associated at least with the resource. In response to the receiving, a previously stored credential associated with the resource is accessed. The credential is provided to the resource.

27 Claims, 8 Drawing Sheets



Related U.S. Application Data

continuation of application No. 13/706,254, filed on Dec. 5, 2012, now Pat. No. 9,294,452.

(60) Provisional application No. 61/587,387, filed on Jan. 17, 2012, provisional application No. 61/569,112, filed on Dec. 9, 2011.

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,697,729	B2	4/2010	Howell	
7,950,051	B1	5/2011	Spitz	
8,145,916	B2	3/2012	Boshra	
8,549,300	B1 *	10/2013	Kumar	H04L 9/3247 713/153
8,577,813	B2	11/2013	Weiss	
8,856,539	B2	10/2014	Weiss	
8,984,596	B2	3/2015	Griffin	
9,100,826	B2	8/2015	Weiss	
2004/0107170	A1	6/2004	Labrou	
2004/0236632	A1	11/2004	Maritzen	
2005/0198348	A1	9/2005	Yeates	
2009/0100269	A1	4/2009	Naccache	
2010/0242102	A1	9/2010	Cross	
2011/0078771	A1	3/2011	Griffin	
2011/0205016	A1	8/2011	Al-Azem	
2011/0231651	A1	9/2011	Bollay	
2012/0110341	A1 *	5/2012	Beigi	H04L 9/3268 713/186
2012/0167193	A1	6/2012	Gargaro	

OTHER PUBLICATIONS

Hammer-Lahav, Ed. "The OAuth 1.0 Protocol", from https://tools.ieff.org/html/rfc5849, Apr. 2010.

* cited by examiner

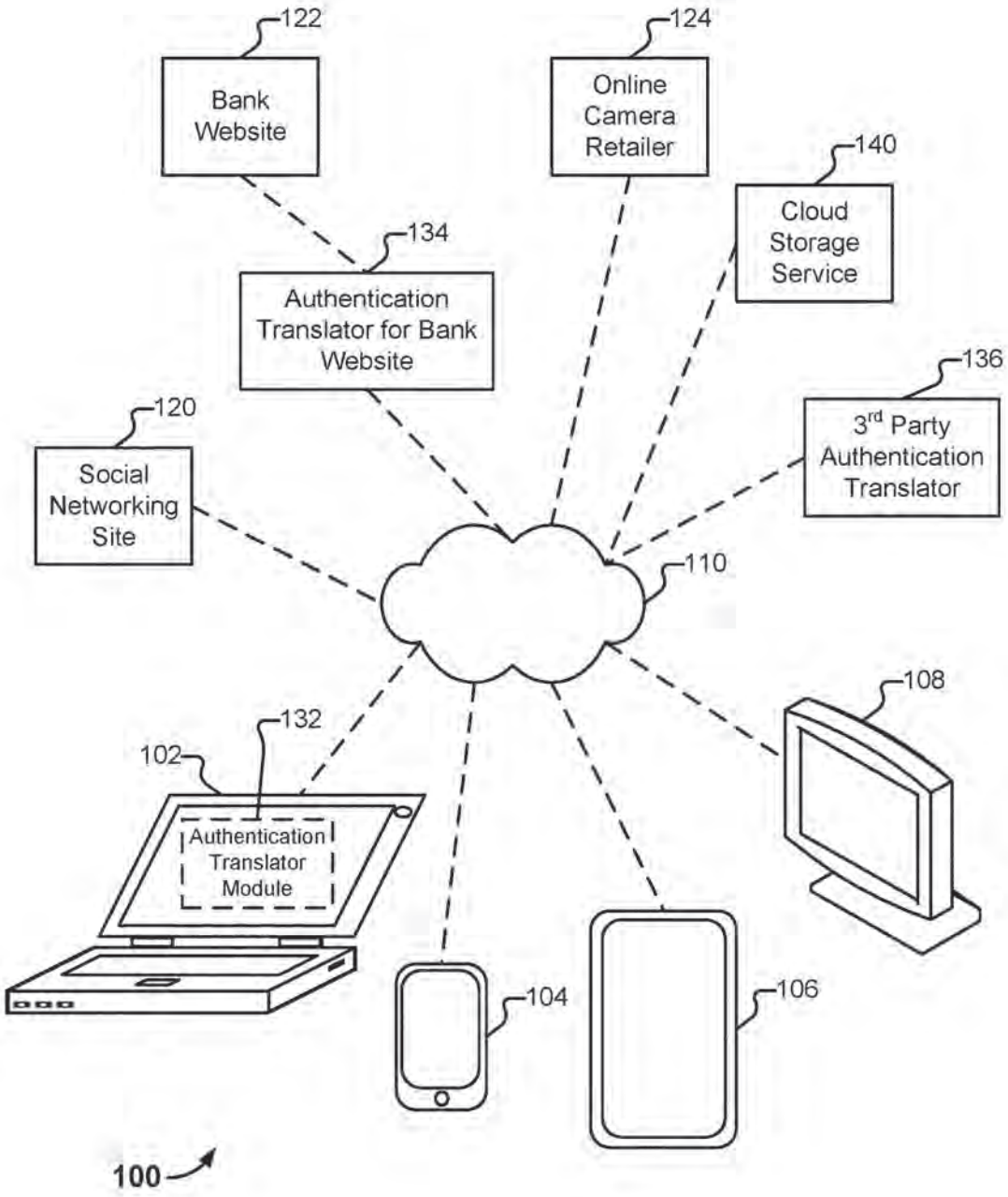

122
Bank Website
124
Online Camera Retailer
140
Cloud Storage Service
134
Authentication Translator for Bank Website
136
3rd Party Authentication Translator
120
Social Networking Site
110
108
132
102
Authentication Translator Module
104
106
100

FIG. 1

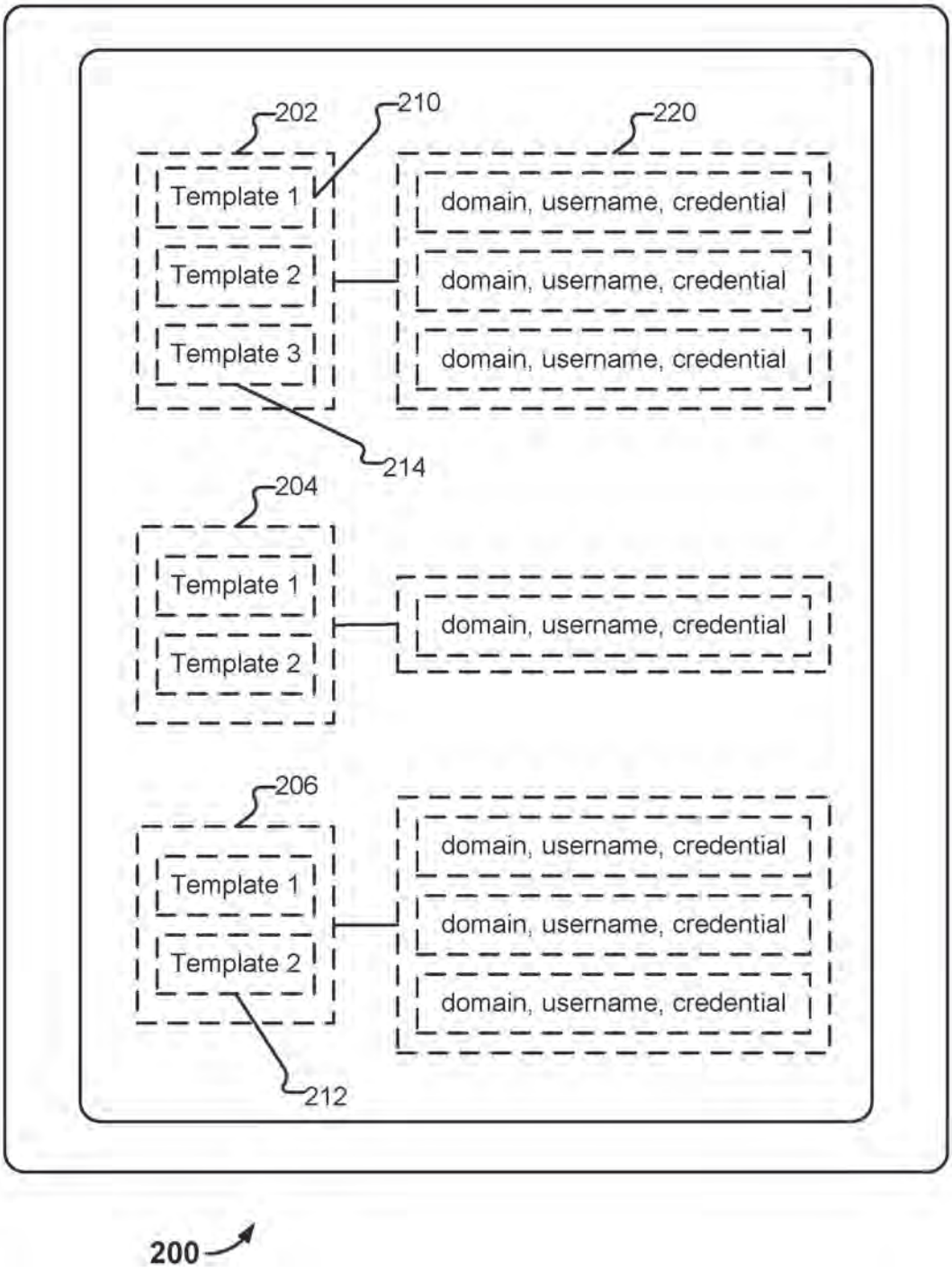
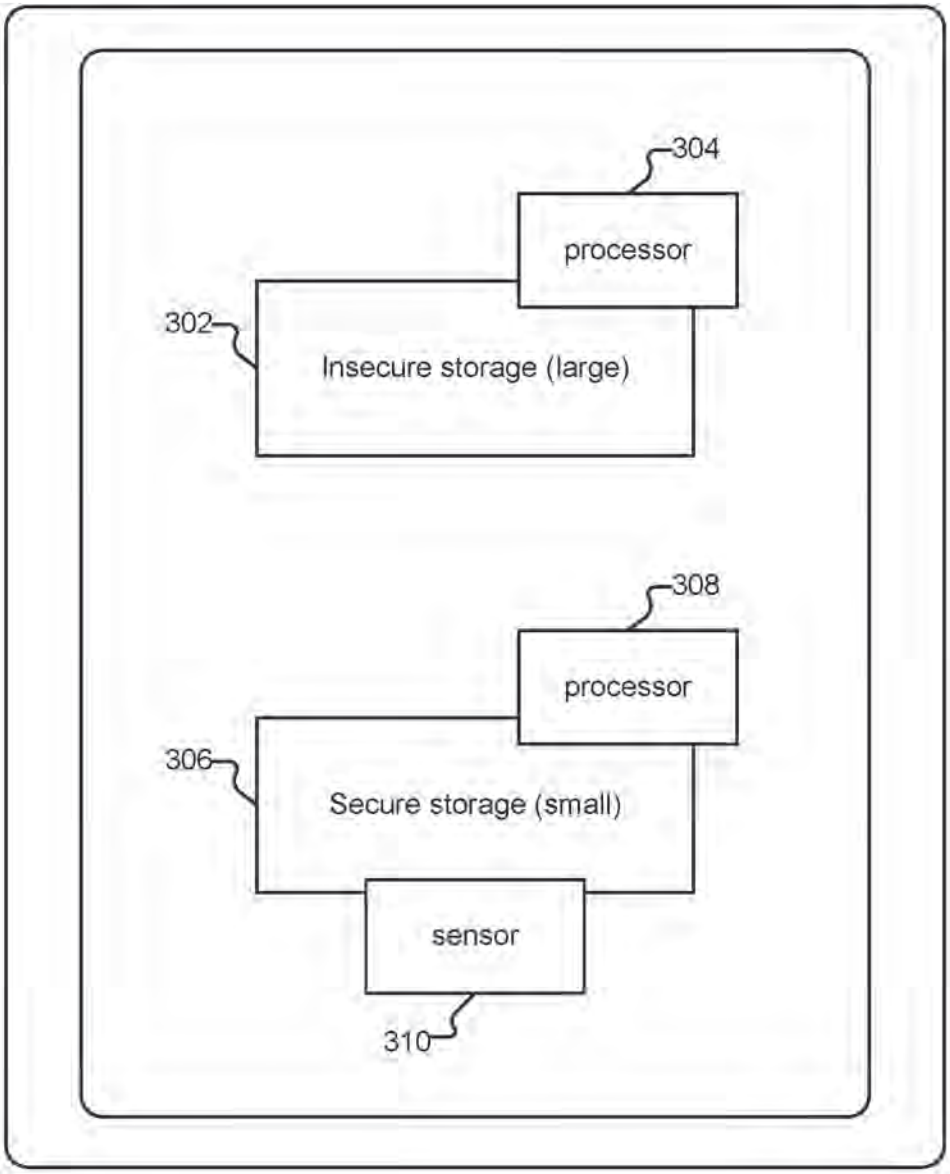

202
210
220
Template 1
Template 2
Template 3
domain, username, credential
domain, username, credential
domain, username, credential
214
204
Template 1
Template 2
domain, username, credential
206
Template 1
Template 2
domain, username, credential
domain, username, credential
domain, username, credential
212
200

FIG. 2


304
processor
302
Insecure storage (large)
308
processor
306
Secure storage (small)
sensor
310

300

FIG. 3

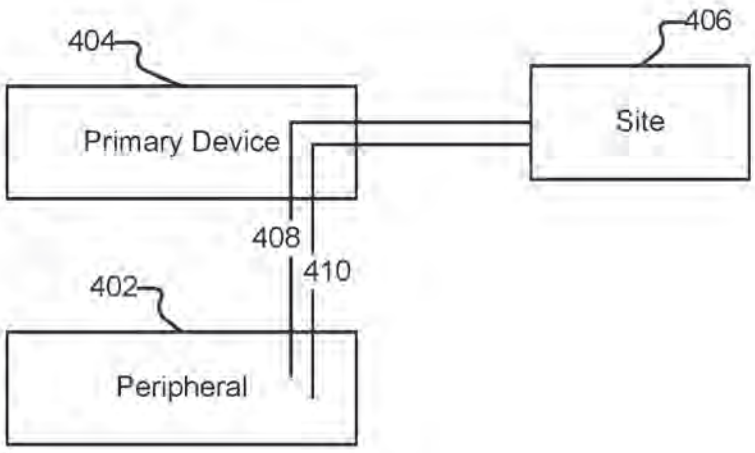


FIG. 4

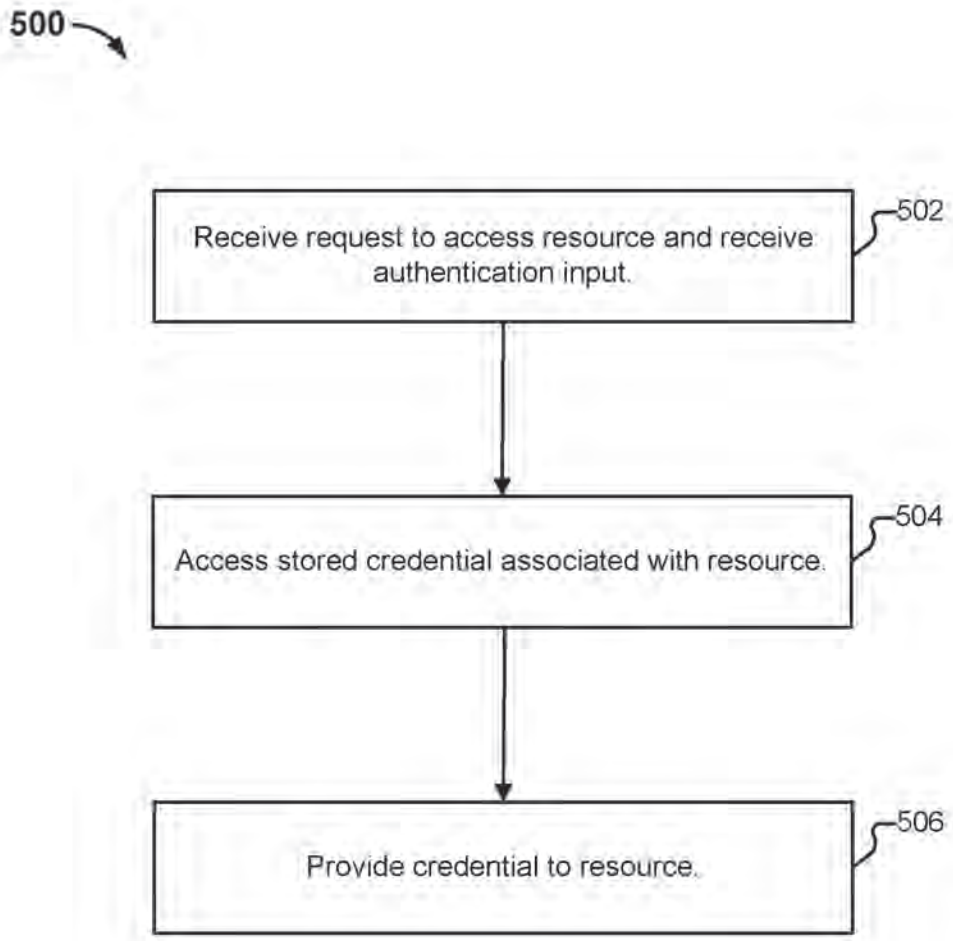

500
Receive request to access resource and receive authentication input.
502
Access stored credential associated with resource.
504
Provide credential to resource.
506

FIG. 5

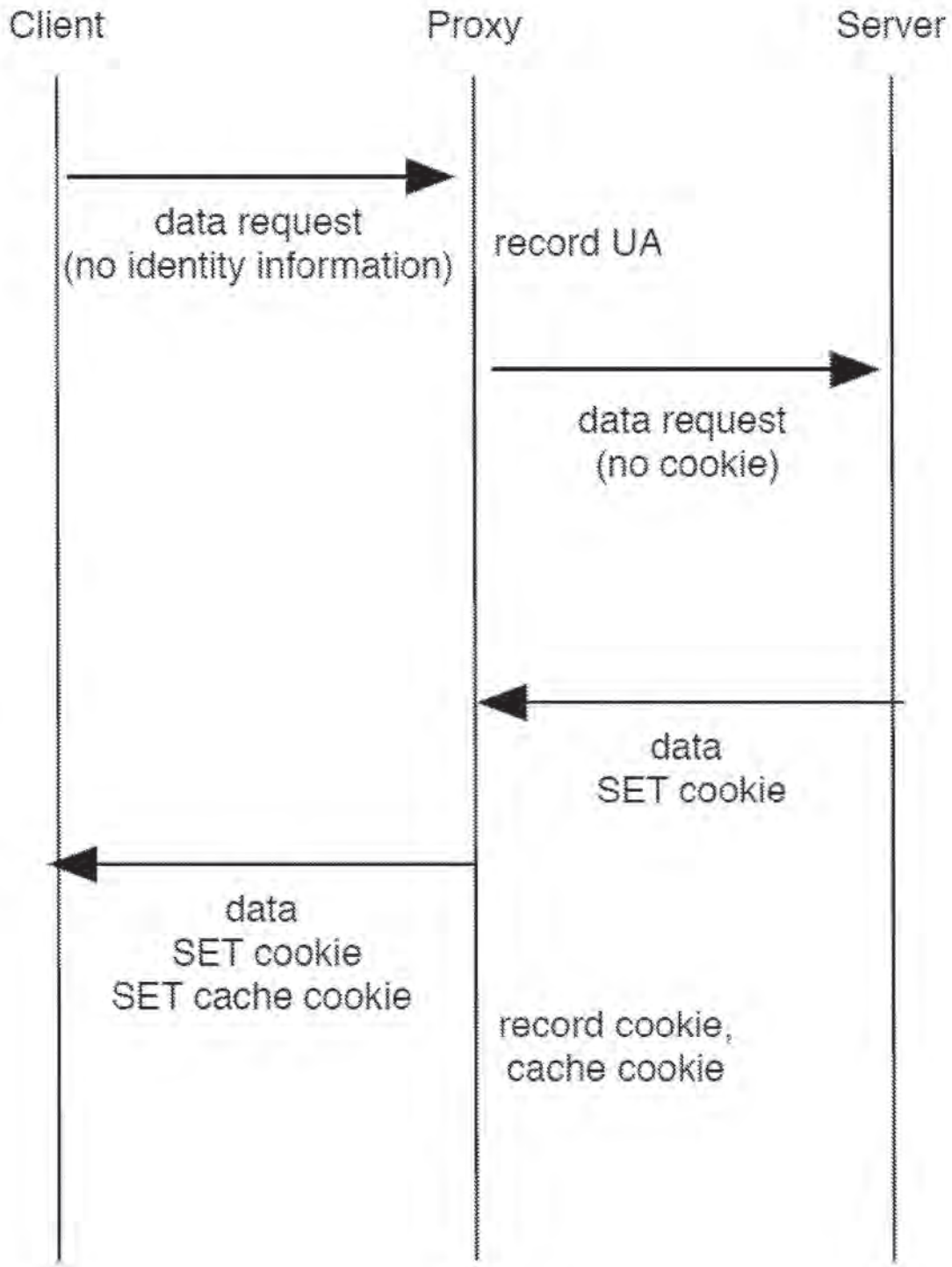

Client
Proxy
Server
data request
(no identity information)
record UA
data request
(no cookie)
data
SET cookie
data
SET cookie
SET cache cookie
record cookie,
cache cookie

FIG. 6

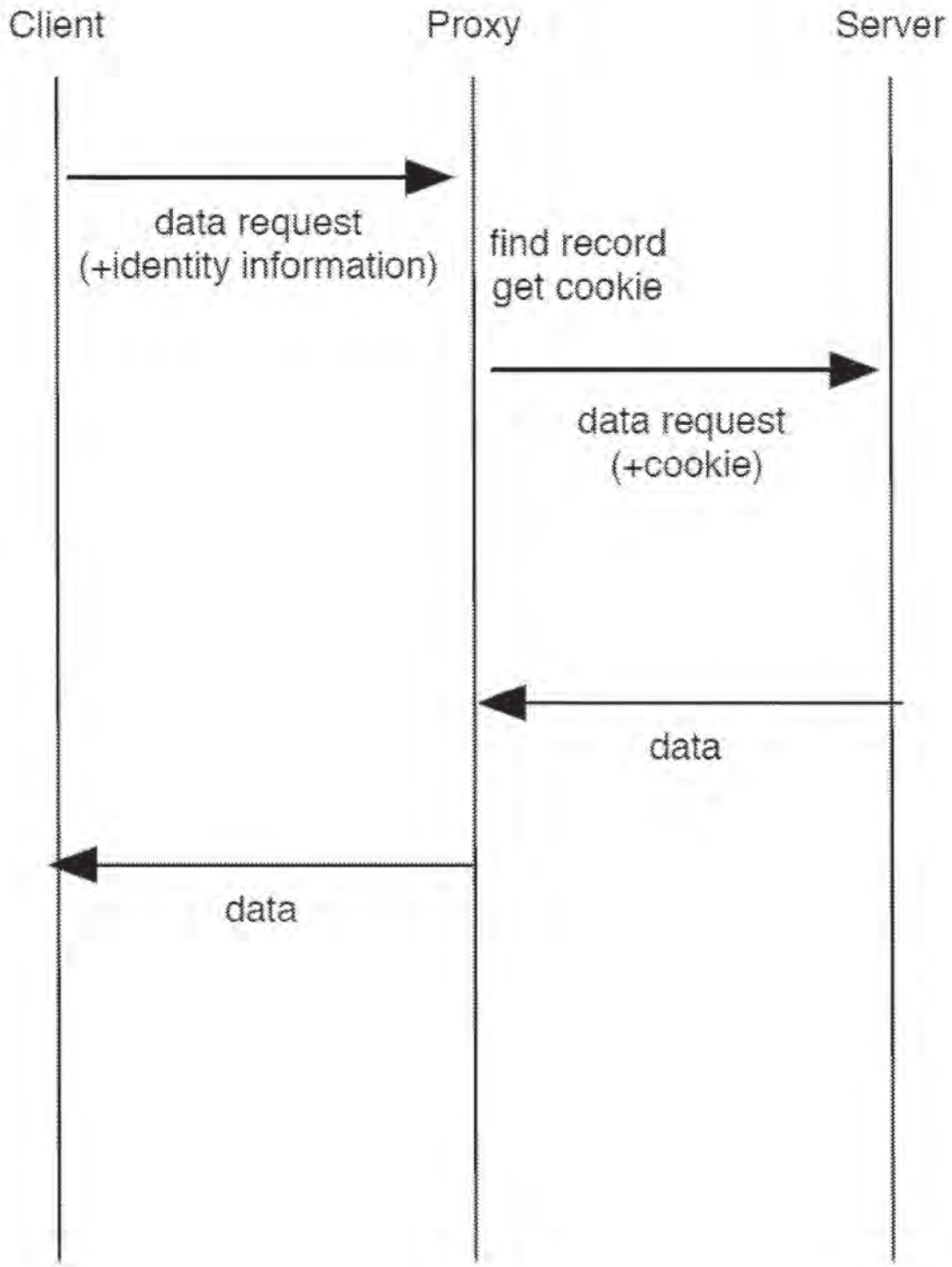

Client
Proxy
Server
data request
(+identity information)
find record
get cookie
data request
(+cookie)
data
data

FIG. 7

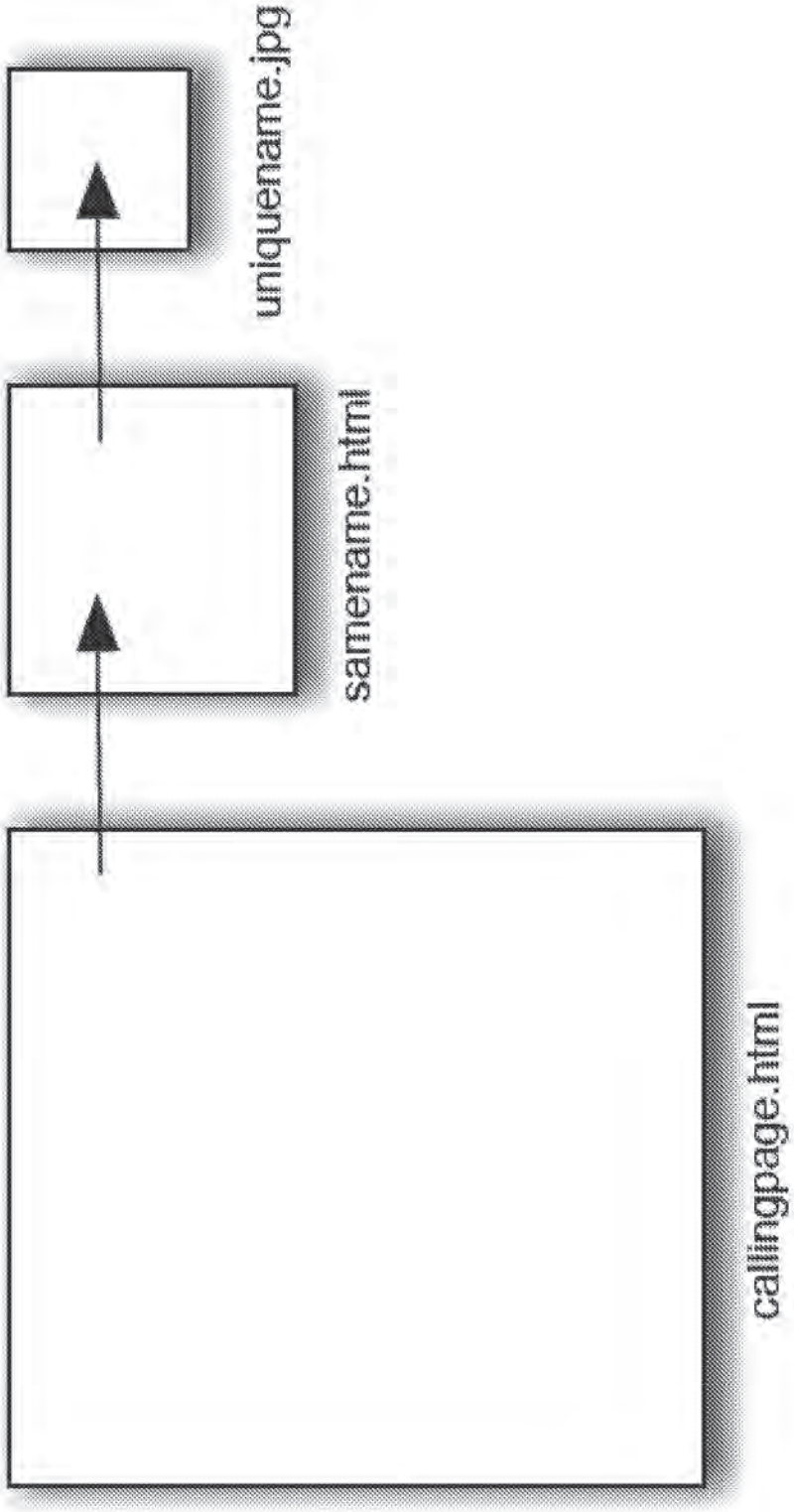


FIG. 8

AUTHENTICATION TRANSLATION

CROSS REFERENCE TO OTHER APPLICATIONS

This application is a continuation of co-pending U.S. patent application Ser. No. 16/273,797, entitled AUTHENTICATION TRANSLATION filed Feb. 12, 2019 which is incorporated herein by reference for all purposes, which is a is a continuation of U.S. patent application Ser. No. 15/042,636, entitled AUTHENTICATION TRANSLATION filed Feb. 12, 2016, now U.S. Pat. No. 10,360,351, which is incorporated herein by reference for all purposes, which is a continuation of U.S. patent application Ser. No. 13/706,254, entitled AUTHENTICATION TRANSLATION filed Dec. 5, 2012, now U.S. Pat. No. 9,294,452, which is incorporated herein by reference for all purposes, which claims priority to U.S. Provisional Application No. 61/587,387, entitled BIOMETRICS-SUPPORTED SECURE AUTHENTICATION SYSTEM filed Jan. 17, 2012 which is incorporated herein by reference for all purposes. U.S. patent application Ser. No. 13/706,254 also claims priority to U.S. Provisional Patent Application No. 61/569,112 entitled BACKWARDS COMPATIBLE ROBUST COOKIES filed Dec. 9, 2011, which is incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

Providing credentials to a service, whether via a mobile or other device, is often a tedious experience for a user. Unfortunately, to make authentication easier for themselves, users will often engage in practices such as password re-use, and/or the selection of poor quality passwords, which render their credentials less secure against attacks. Accordingly, improvements in authentication techniques would be desirable. Further, it would be desirable for such improvements to be widely deployable, including on existing/legacy systems.

BRIEF DESCRIPTION OF THE DRAWINGS

Various embodiments of the invention are disclosed in the following detailed description and the accompanying drawings.

FIG. **1** illustrates an embodiment of an environment in which authentication translation is provided.

FIG. **2** illustrates an embodiment of credential information stored on a device.

FIG. **3** illustrates an embodiment of a device with secure storage.

FIG. **4** illustrates an example of a renegotiation.

FIG. **5** illustrates an embodiment of a process for performing authentication translation.

FIG. **6** illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator.

FIG. **7** illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator.

FIG. **8** shows the structure of an example of a cache cookie used in some embodiments.

DETAILED DESCRIPTION

The invention can be implemented in numerous ways, including as a process; an apparatus; a system; a composition of matter; a computer program product embodied on a computer readable storage medium; and/or a processor, such as a processor configured to execute instructions stored on and/or provided by a memory coupled to the processor. In this specification, these implementations, or any other form that the invention may take, may be referred to as techniques. In general, the order of the steps of disclosed processes may be altered within the scope of the invention. Unless stated otherwise, a component such as a processor or a memory described as being configured to perform a task may be implemented as a general component that is temporarily configured to perform the task at a given time or a specific component that is manufactured to perform the task. As used herein, the term 'processor' refers to one or more devices, circuits, and/or processing cores configured to process data, such as computer program instructions.

A detailed description of one or more embodiments of the invention is provided below along with accompanying figures that illustrate the principles of the invention. The invention is described in connection with such embodiments, but the invention is not limited to any embodiment. The scope of the invention is limited only by the claims and the invention encompasses numerous alternatives, modifications and equivalents. Numerous specific details are set forth in the following description in order to provide a thorough understanding of the invention. These details are provided for the purpose of example and the invention may be practiced according to the claims without some or all of these specific details. For the purpose of clarity, technical material that is known in the technical fields related to the invention has not been described in detail so that the invention is not unnecessarily obscured.

FIG. **1** illustrates an embodiment of an environment in which authentication translation is provided. In the example shown, a variety of client devices **102-108** connect, via one or more networks (represented as a single network cloud **110**) to a variety of services **120-124** (also referred to herein as sites **120-124**). In particular, client device **102** is a notebook computer owned by a user hereinafter referred to as Alice. Notebook **102** includes a camera, a microphone, and a fingerprint sensor. Client device **104** is a smartphone, also owned by Alice. Client device **104** includes a camera. Client device **106** is a tablet owned by Bob, and sometimes used by Bob's son Charlie. Client device **106** includes a camera and a fingerprint sensor. Client device **108** is a kiosk located in the lobby of a hotel. Kiosk **108** includes a camera and a microphone. The techniques described herein can be used with or adapted to be used with other devices, as applicable. For example, the techniques can be used in conjunction with gaming systems, with peripheral devices such as mice, and with embedded devices, such as door locks.

Service **120** is a social networking site. Service **122** is a website of a bank. Service **124** is the online store of a boutique camera retailer. Each of services **120-124** requires a username and password (and/or a cookie) from a user prior to giving that user access to protected content and/or other features. As will be described in more detail below, using the techniques described herein, users need not type such usernames and passwords into their devices whenever required by a service. Instead, users can authenticate themselves to an "authentication translator" via an appropriate technique, and the authentication translator will provide the appropriate credentials to the implicated service on the user's behalf. Also as will be described in more detail below, authentication translators can be located in a variety of places within an environment. For example, notebook computer **102**

includes an authentication translator module **132** that provides authentication translation services. The other devices **104-108** can also include (but need not include) their own respective authentication translator modules. The owner of bank website **122** also operates an authentication translator **134** associated with the bank. Finally, authentication translator **136** provides authentication translation services to a variety of businesses, including online camera retailer **124**.

FIG. **2** illustrates an embodiment of credential information stored on a device. In particular, device **200** stores three user profiles **202-206**, each of which contains a username and one or more templates (e.g., template **210**) associated with the user. In various embodiments, a template is a collection of biometric features. Using fingerprints as an example type of biometric, a corresponding template includes a collection of patterns, minutia, and/or other features that can be matched against to determine if a person's fingerprint matches the fingerprint of the registered user (i.e., the owner of a given user profile). A representation of a single fingerprint may be included in multiple templates (e.g., in different resolutions, in accordance with different protocols, as captured during warm vs. cold conditions, and/or by itself or in combination with multiple fingerprints). When other biometrics are employed (e.g., facial recognition, voiceprint, or retina scan technology), features appropriate to those types of biometrics are included in the template. Other types of features can also be included in templates. As one example, a user's typing speed and/or accuracy can be measured by a device, such as device **102**, and used to distinguish between multiple users of a device. For example, suppose Alice types at 100 words per minute and rarely makes mistakes. A representation of this information can be stored in template **212**. Also suppose Alice's niece, who sometimes uses Alice's laptop computer when visiting Alice types at 20 words per minute and makes many mistakes. In some embodiments, the fact that a user was recently (e.g., within the last 5 minutes) typing on laptop **102** at 90 words per minute is evidence of a match against template **212**. In this case, the typing speed of 90 words per minute is similar enough to Alice's typical behavior, it is considered a match. Various policies can be included in a profile that govern how matches are to be performed. For example, policies can specify thresholds/tolerances for what constitutes a match, and can specify that different levels of matches can result in different levels of access to different resources.

A profile is associated with a vault (e.g., vault **220**). The vault, in turn, contains triples specifying a service provider/domain, a username, and a credential. The vault can also contain other sensitive user information, such as account numbers, address/phone number information, and health care data. The credential for a service provider/domain can be a password (e.g., for legacy servers), and can also take alternate forms (e.g., a cryptographic key for service providers supporting stronger authentication methods).

In some embodiments, profiles, templates, and vaults (collectively "authentication information") are stored entirely in an unprotected storage area, and are stored in the clear. In other embodiments, secure storage techniques are used to secure at least a portion of the authentication information.

One example of a device with secure storage is illustrated in FIG. **3**. In the example shown, a mobile phone **300** includes a large and insecure storage **302** attached to a fast processor **304**, and a smaller but secure storage **306** attached to a dedicated processor **308** and a sensor **310** (e.g., a camera or a fingerprint reader). Users (and applications) can read from and write to the insecure storage area. However, users cannot access the secure storage area, and the fast processor can only communicate with the dedicated processor/sensor via a restricted API. As another example, a unique decryption key associated with a given vault can be stored in a profile. The vault is an encrypted and authenticated container that can be stored on insecure storage, e.g., on the device, and also backed up (e.g., to a cloud storage service **140** or to an alternate form of external storage). As needed, authentication information or portions thereof can be loaded into secure storage and decrypted. For example, one can use AES to encrypt the files one by one, using a key stored on the secured storage. A message authentication technique, such as HMAC, can be used for authenticating the encrypted files to provide tamper prevention. Profiles and vaults can be updated while in secure storage; if this occurs, they are encrypted and MACed before being written back to the insecure storage, which may in turn propagate them to external backup storage. In yet other embodiments, profiles and vaults are stored entirely in secure storage, in plaintext, which allows them to be both read and written—and in particular, searched.

Example Transaction Types

A variety of transaction types can take place in the environment shown in FIG. **1**, examples of which are discussed in this section.

Initial Registration

In order to begin using the techniques described herein, users perform some form of initial registration. As one example, suppose Alice launches an enrollment program installed on laptop **102**. She uses the program to capture various biometric information (e.g., fingerprints, photographs of her face, etc.). A user profile is created for Alice, and the biometric information captured about her is encoded into a plurality of templates, such as templates **210** and **214**. In some embodiments, Alice is also explicitly asked to supply credential information for services she would like to use, such as by providing the domain name of social networking site **120**, along with her username and password for site **120**. In other embodiments, domain/username/credential information is at least passively captured on Alice's behalf and included in one or more vaults such as vault **220**. Credential information can also be important from a browser password manager already in use by Alice or other appropriate source. In some embodiments, Alice also registers with cloud storage service **140**, which will allow her to back up her authentication information and to synchronize it across her devices (e.g., **102** and **104**), as described in more detail below.

Other registration approaches can also be used. For example, registration can be integrated into the experience the first time a device is used. Thus, when Bob first turns on tablet **106**, he may be prompted to take a picture of his face (with a profile/templates being created in response). Similarly, the first time Charlie uses tablet **106**, the techniques described herein can be used to determine that Charlie does not yet have a profile (e.g., because none of the templates already present on tablet **106** match his biometrics) and Charlie can be prompted to enroll as a second user of the device.

Authentication

Suppose Alice wishes to authenticate to banking website **122**. Using a fingerprint reader incorporated into her laptop, she performs a fingerprint scan, which causes her biometric features to be extracted and compared to any stored templates residing on her computer. If a match is found, an associated decryption key is selected, and the associated

vault is loaded and decrypted. The vault is scanned for an entry that matches the selected service provider (i.e., website **122**). If a matching entry is found, the associated domain, username, and site credential are extracted from the vault. In some embodiments, the validity of the domain name mapping is verified at this point to harden the system against domain name poisoning. Next, a secure connection is established between Alice's computer and the service provider, and Alice is authenticated. For service providers supporting strong user authentication, mutual SSL can be used, for example. A variety of policies can be involved when performing matching. For example, to access certain domains, Alice's print may need only match template **210**. To access other domains, Alice may need to match multiple templates (e.g., both **210** and **214**). As another example, in order to access social networking site **120**, Alice may merely need to be sitting in front of her computer, which has an integrated webcam. Even in relatively low light conditions, a match can be performed against Alice's face and features stored in a template. However, in order to access bank website **122**, Alice may need a high quality photograph (i.e., requiring her to turn on a bright light) and may need to demonstrate liveness (e.g., by blinking or turning her head). As yet another example, other contextual information can be included in policies. For example, if Alice's IP address indicates she is in a country that she is not usually in, she may be required to match multiple templates (or match a template with more/better quality features) in order to access retailer **124**, as distinguished from when her IP address indicates she is at home.

In some embodiments, the biometric sensor used by a user may be a peripheral device (e.g., a mouse with an integrated fingerprint scanner that is connected to the user's primary device via USB). In such scenarios, the peripheral device may be responsible for storing at least a portion of authentication information and may perform at least some of the authentication tasks previously described as having been performed by Alice's computer. For example, instead of processors **304** and **308**, and storages **302** and **306** being collocated on a single device (e.g., laptop **102**), processor **304** and storage **302** may be present on a primary device, and processor **308** and storage **306** may be present on a peripheral device (e.g., that also includes a sensor, such as a fingerprint reader).

In such scenarios, once Alice's login to banking website **122** is successfully completed, the secure session can be handed over from the peripheral device to the primary device, in a way that does not allow the primary device retroactive access to the plaintext data of the transcripts exchanged between the peripheral device and the service provider. One way this can be accomplished is by renegotiating SSL keys between the peripheral device and the website, after which the newly negotiated key can be handed off from the peripheral device to the primary device. This avoids retroactive credential capture in a setting where the device is infected by malware.

An example of renegotiation is depicted in FIG. **4**. Specifically, after a user has successfully authenticated to a fingerprint reader, a login is performed to a service provider. Using the primary device (**404**) as a proxy, the peripheral fingerprint reader **402** negotiates a first SSL connection (**408**) with a service provider **406**, over which credentials are exchanged. The proxy then renegotiates SSL (**410**), which replaces the old key with a new one. The new key is disclosed to the device, which then seamlessly takes over the connection with the service provider and performs the transaction protected by the authentication. The credentials exchanged during the first SSL connection cannot be accessed by device **404**, since the key of the renegotiated session is independent of the key of the original session; this provides protection against malware residing on the device. Renegotiation can be used when the primary device **404** is believed to be in a safe state when performing the negotiation of the SSL connection, but it is not known whether it is in a safe state during the transaction protected by the authentication. Renegotiation can also be used when a secure component of the primary device **404** performs the negotiation of the SSL connection and another and potentially insecure component of the primary device **404** is involved in the transaction protected by the authentication.

FIG. **5** illustrates an embodiment of a process for performing authentication translation. The process begins at **502** when a request to access a resource is received, as is an authentication input. One example of the processing performed at **502** is as follows. Suppose Alice wishes to sign into social networking website **120**. She directs a web browser application installed on client **102** to the social networking website. Authentication translator module **132** recognizes, from the context of Alice's actions (e.g., that she is attempting to access site **120** with her browser) that she would like to access a particular resource. Authentication translator module **132** prompts Alice (e.g., by a popup message or via a sound) to provide biometric information (e.g., to use the integrated fingerprint reader on her laptop). In some embodiments, the translator module does not prompt Alice, for example, because Alice has been trained to provide biometric information automatically when attempting to access certain resources. In yet other embodiments, the translator module only prompts Alice if she fails to provide acceptable biometric information within a timeout period (e.g., 30 seconds).

Module **132** compares Alice's supplied biometric data to the templates stored on her computer. If a suitable match is found, and if an entry for site **120** is present in the applicable vault, at **504**, a previously stored credential associated with the resource is accessed. In particular, the username and password for the website, as stored in a vault, such as vault **220**, are retrieved from the vault.

Finally, at **506**, the credential is provided to the resource. For example, Alice's username and password for site **120** are transmitted to site **120** at **506**. The credential can be transmitted directly (e.g., by the module or by Alice's computer) and can also be supplied indirectly (e.g., through the use of one or more proxies, routers, or other intermediaries, as applicable).

Other devices can also make use of process **500** or portions thereof. For example, when Alice launches a banking application on phone **104**, implicit in her opening that application is her desire to access the resources of website **134**. The application can take Alice's picture and compare it to stored templates/vault information. If an appropriate match is found, a credential can be retrieved from the vault on her phone (or, e.g., retrieved from cloud storage service **140**) and provided to website **134**.

As another example, suppose Charlie is using tablet **106** and attempts to visit site **120**, whether via a dedicated application or via a web browser application installed on the tablet. Charlie's photograph is taken, and then compared against the profiles stored on tablet **106** (e.g., both Bob and Charlie's profiles). When a determination is made that Charlie's photograph matches a template stored in his stored profile (and not, e.g., Bob's), Charlie's credentials for site **120** are retrieved from a vault and transmitted by an authentication translator module residing on client **106**.

As yet another example, kiosk **108** can be configured to provide certain local resources (e.g., by displaying a company directory or floor plan on demand) when users speak certain requests into a microphone. Enrolled users (e.g., with stored voiceprint or facial recognition features) can be granted access to additional/otherwise restricted services in accordance with the techniques described herein and process **500**.

New Device

In some embodiments, to register a new device, a user provides an identifier, such as a username or an account number to the device. The new device connects to an external storage (such as cloud storage **140**), provides the user identifier and credential, and downloads the user's templates/vaults from the service. In some embodiments, the templates/vaults are encrypted. Once downloaded, the template is decrypted and stored in a secure storage area, while the still encrypted vault can be stored in insecure storage. The decryption key can be generated from information the user has/knows, or from biometric data—such as features extracted from fingerprinting of all ten fingers. In some embodiments, more arduous fingerprinting is required for the setup of a new device than for regular authentication to avoid that a new device gets registered by a user thinking she is merely authenticating—or worse still, simply touching the device. Moreover, it translates into higher entropy of the decryption keys.

Backup Authentication

Backup authentication allows a user, such as Alice, to access resources in the event she is unable to or unwilling to interact with a particular biometric sensor. As one example, instead of having a single template associated with her profile, Alice can have multiple templates associated with it, e.g., where the first template includes fingerprint features and the second template includes voice biometric, facial recognition, or iris detection features. As a second example, where the service Alice is connecting to is a legacy website (i.e., one that users authenticate to using usernames and passwords), such a service would allow the use of passwords and password reset mechanisms by Alice without requiring Alice to use a fingerprint reader.

In various embodiments, environment **100** supports the ability of users (e.g., under duress) to release the contents of their vaults. For example, if Alice was physically threatened with the loss of a finger by a criminal, Alice could instead release the contents of her vault(s)—the ultimate goal of the criminal. As one example, in the event Alice supplies all 10 fingerprints to the sensor, provides a special password, or supplies a fingerprint and a second identifier, a cleartext version of her vault(s) could be made available.

Access Policies

In various embodiments, cloud storage service **140** is configured to accept backups from multiple devices associated with a single account, and synchronize the updates so that all devices get automatically refreshed. For example, Alice's laptop **102** and phone **104** could both communicate with cloud storage service **140** which would keep their authentication information synchronized. Refreshes can also be made in accordance with user-configured restrictions. For example, Alice's employer could prevent privileged employer data from being stored on shared personal devices, or on any device that was not issued by the employer. As another example, arbitrary policies can be defined regarding the access to and synchronization of software and data, and to tie a license or access rights to a person (and her fingerprint) rather than to a device. As yet another example, in some embodiments (e.g., where a device is made publicly available or otherwise shared by many users), no or a reduced amount of authentication information resides on a device, and at least a portion of authentication information is always retrieved from cloud storage service **140**.

Remote Wiping

Remote wiping of a user's authentication information (e.g., templates) can be used both to "unshare" previously shared devices (e.g., where Bob and Charlie both have user profiles on their shared tablet **106**), and to avoid that criminals with physical component access to lost devices gain access to templates and vault contents. In some embodiments, polices such as ones where a template self-wipes if it is not matched within a particular duration of time are supported. Since user data can be frequently backed up to the cloud storage, and recovered from this using the new device registration process, inconvenience to the user will be minimized.

Legacy Server Support

New authentication schemes typically require changes to a significant codebase residing with service providers. If the code is well written and documented, such changes may be relatively simple. Commonly, though, this may not be so. The engineers who originally wrote the code of relevance may have long since left the company; the code they left behind may be poorly documented—if documented at all. In severe cases, the legacy code may have been written in an outdated programming language or written in a way that does not follow guidelines for good code. This makes updates to the codebase impractical or virtually impossible in many common cases. Even if none of these challenges complicate the desired modifications, it is commonly a great bureaucratic burden to obtain permission to make modifications (e.g., to store an extra field in a backend database), because every affected part of the organization may need to review the request.

As will be described in the following section, the technologies described herein can be used in conjunction with legacy servers (e.g., existing servers that rely on usernames and passwords to authenticate users), and in particular, can be used without requiring modification to such legacy servers.

Cookies

Cookies are commonly used by legacy servers for user authentication. Unfortunately, cookies have several problems. For one thing, they are sometimes deleted—whether explicitly/intentionally by the end user or by the user's software. In addition, cookies are commonly stolen. Approaches such as cache cookies and identification using user agents can be more resistant to these problems, however, they have their own problems. For example, their use requires new code and new fields in the credential database stored by the server.

In some embodiments, authentication translators, such as translators **134** and **136** (also referred to herein as proxies) provide authentication translation services on behalf of associated services. Translators **134** and **136** are illustrated as single logical devices in FIG. **1**. In some embodiments, the translators comprise standard commercially available server hardware (e.g., a multi-core processor, 4+ Gigabytes of RAM, and one or more Gigabit network interface adapters) and run typical server-class operating systems (e.g., Linux). Translators **134** and **136** can also be implemented using a scalable, elastic architecture and may comprise several distributed components, including components provided by one or more third parties. For example, translators

134 and 136 may store user credential information or may task cloud storage service 140 with storing at least a portion of that information.

In the case of authentication translator 134, service is provided with respect to bank website 122 only. Authentication translator 134 is positioned between a legacy web server (122) and the Internet (110)—and therefore between the legacy server and any client devices. Authentication translator 134 is configured to translate traffic between the legacy server and client devices so that the client devices (and respective users) perceive the new authentication mechanism, while the legacy server remains unchanged. Authentication translator 136 works similarly, but it provides authentication translation as a third party service to multiple providers, an example of which is online camera retailer 124.

Authentication translators 134 and 136 can also perform process 500. For example, when a device transmits a request to access website 122, the request is intercepted by translator 134, as is cookie/user agent information. The received information can be used to determine a username/password associated with the device, and that information can be passed by translator 134 to website 122 on behalf of the device.

FIG. 6 illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator. The translator (referred to in the figure as a "proxy") fails to identify the client, and passes on the request to the legacy server. The legacy server responds to the request and sets a cookie. The proxy passes on the response, including the cookie and also a cache cookie. The proxy stores the information about both these types of identifiers, along with the user agent of the client device. This triplet of information is also referred to herein as an identifier.

FIG. 7 illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator. In this scenario, the device's request is accompanied by some form of identifying information. The proxy uses this information to identify the associated cookie and passes this along to the legacy server, along with the request. In some embodiments, additional processing is also involved. For example, reading a cache cookie may require user interaction. Moreover, if not all of the identifying information is present in the request, the proxy can be configured to set the missing information again by sending a corresponding request to the client device.

The translation of cache cookies and user agent information to cookies involves a two-way translation. First, when the legacy server sets a cookie, the proxy will set the two types of cookies—both an HTML cookie and a cache cookie—and then create a new record in which the two cookies are stored, along with the user agent information of the client device. The user agent information can include quite a bit of data associated with a browser—such as the browser type and version, the clock skew, and the fonts that are installed. While each of these pieces of information only contributes a small amount of entropy, the collection of items can be sufficient to identify the device in most cases. Moreover, while some of these types of data may change over time—in fact, all of them may—they do not typically change, and when one or two of them do, the others typically do not. When the client device is used to visit a site controlled by the legacy server, the cookie, cache cookie and user agent information are read (if available), the record identified, and the request translated and sent to the legacy server. When a legacy server requests that the user password is updated (e.g., as part of an annual or other periodic requirement), the transmission of this request to the user can be suppressed—in which case the database of the proxy is updated to create the illusion of an update. The user can be involved in authentication as needed, e.g., if, in addition to supplying a credential, a user must also solve a CAPTCHA, the CAPTCHA can be displayed to the user (with the user's credentials being handled in accordance with the techniques described herein).

FIG. 8 shows the structure of an example of a cache cookie used in some embodiments. Cache cookies can be associated with a particular webpage, just like an HTML cookie can. In the example shown in FIG. 8, the proxy wishes to associate a page "calling page.html" with a cache cookie. It embeds a request for a second object, "samename.html" in callingpage.html for every visitor. However, as the cache cookie is set for one visitor, a customized samename.html is served to this visitor. The page samename.html refers to an object with a different name for each user; that object is referred to as "uniquename.jpg." The cache cookie is set by embedding the request for samename.html in callingpage.html. The client browser attempts to render this, causing a request for samename.html from the server. The server configures samename.html to refer to a uniquely named file uniquename.jpg, and serves samename.html to the client. For the client browser to render samename.html, it requests the file uniquename.jpg, which is intentionally not served. That concludes the setting of a cache cookie. As a user returns to the page callingname.html, the browser again attempts to render the entire page, which causes it to load the object samename.html from its cache. As that is rendered, the client browser requests uniquename.jpg, which is not in its cache (since it was not served previously). The server still does not serve it, but takes note of the name of the file being requested, as it identifies the client device. Note that samename.html can be displayed in a zero-sized iframe, which makes the end user unaware of it being rendered.

A cache cookie is an implementation of the typical cookie functionality that uses the client device's browser cache. Unlike user agents, it does not change over time, and like standard HTML cookies, it cannot be read by a party other than that which set it. However, like HTML cookies, it could be deleted—by the user clearing his or her browser cache. Cache cookies are not automatically transmitted with GET requests, unless the cache elements are embedded in the referring pages. This adds a potential round of communication in some settings. By relying on user agent information, cache cookies, and HTML cookies to identify the client device, it is much more likely that a machine will be recognized than if only HTML cookies are used.

Although the foregoing embodiments have been described in some detail for purposes of clarity of understanding, the invention is not limited to the details provided. There are many alternative ways of implementing the invention. The disclosed embodiments are illustrative and not restrictive.

What is claimed is:

1. A system, comprising:

a first processor configured to:

based at least in part on a request from a user to access an external resource, communicate with a second processor using a restricted interface, wherein the second processor is configured to access at least one record stored at least in part in a secure storage;

wherein the at least one record is associated at least with the external resource, and wherein the at least one record comprises:

a biometric template; and

a credential comprising at least one of a password, a cookie, or a cryptographic key;

in response to determining a match between a biometric input and the biometric template, retrieve, from the at least one record, the credential, wherein the biometric input corresponds to at least one of a fingerprint, a feature usable for facial recognition, a voiceprint, a feature usable for a retina scan, or a typing feature, and wherein the biometric input is received subsequent to presenting of a prompt;

establish a connection with the external resource;

facilitate a login of the user to the external resource at least in part by transmitting, via the established connection, output based at least in part on the credential retrieved from the at least one record, and wherein the user is logged in to the external resource based at least in part on the output; and

facilitate wiping of at least a portion of the at least one record; and

a memory coupled to the first processor and configured to provide the first processor with instructions.

2. The system recited in claim **1** wherein the first processor is unable to directly access the secure storage.

3. The system recited in claim **1** wherein the second processor comprises a dedicated processor, and wherein the secure storage is connected to the dedicated processor.

4. The system recited in claim **1** wherein the secure storage is connected to a sensor.

5. The system recited in claim **4** wherein the sensor comprises at least one of a camera and a fingerprint reader.

6. The system recited in claim **1** wherein at least some of the at least one record is stored in plaintext in the secure storage.

7. The system recited in claim **1** wherein the prompt comprises a prompt to provide biometric information, and wherein the prompt is visually presented.

8. The system recited in claim **1** wherein the prompt comprises a prompt to provide biometric information, and wherein the prompt is aurally presented.

9. The system recited in claim **1** wherein the prompt is presented in response to a user failing to provide acceptable biometric information within a timeout period.

10. The system recited in claim **1** wherein facilitating wiping of the at least portion of the at least one record comprises facilitating remote wiping of the at least portion of the at least one record.

11. The system recited in claim **1** wherein the at least portion of the at least one record is automatically wiped based at least in part on a policy.

12. The system recited in claim **1** wherein the biometric template is wiped in response to determining that the biometric template has not been matched within a duration of time.

13. The system recited in claim **1** wherein the at least portion of the at least one record is backed up to a remote entity.

14. A method, comprising:

based at least in part on a request from a user to access an external resource, using a first processor to communicate with a second processor using a restricted interface, wherein the second processor is configured to access at least one record stored at least in part in a secure storage;

wherein the at least one record is associated at least with the external resource, and wherein the at least one record comprises:

a biometric template; and

a credential comprising at least one of a password, a cookie, or a cryptographic key;

in response to determining a match between a biometric input and the biometric template, retrieving, from the at least one record, the credential, wherein the biometric input corresponds to at least one of a fingerprint, a feature usable for facial recognition, a voiceprint, a feature usable for a retina scan, or a typing feature, and wherein the biometric input is received subsequent to presenting of a prompt;

establishing a connection with the external resource;

facilitating a login of the user to the external resource at least in part by transmitting, via the established connection, output based at least in part on the credential retrieved from the at least one record, and wherein the user is logged in to the external resource based at least in part on the output; and

facilitating wiping of at least a portion of the at least one record.

15. The method of claim **14** wherein the first processor is unable to directly access the secure storage.

16. The method of claim **14** wherein the second processor comprises a dedicated processor, and wherein the secure storage is connected to the dedicated processor.

17. The method of claim **14** wherein the secure storage is connected to a sensor.

18. The method of claim **17** wherein the sensor comprises at least one of a camera and a fingerprint reader.

19. The method of claim **14** wherein at least some of the at least one record is stored in plaintext in the secure storage.

20. The method of claim **14** wherein the prompt comprises a prompt to provide biometric information, and wherein the prompt is visually presented.

21. The method of claim **14** wherein the prompt comprises a prompt to provide biometric information, and wherein the prompt is aurally presented.

22. The method of claim **14** wherein the prompt is presented in response to a user failing to provide acceptable biometric information within a timeout period.

23. The method of claim **14** wherein facilitating wiping of the at least portion of the at least one record comprises facilitating remote wiping of the at least portion of the at least one record.

24. The method of claim **14** wherein the at least portion of the record is automatically wiped based at least in part on a policy.

25. The method of claim **14** wherein the biometric template is wiped in response to determining that the biometric template has not been matched within a duration of time.

26. The method of claim **14** wherein the at least portion of the at least one record is backed up to a remote entity.

27. A computer program product embodied in a non-transitory computer readable storage medium and comprising computer instructions for:

based at least in part on a request from a user to access an external resource, using a first processor to communicate with a second processor using a restricted interface, wherein the second processor is configured to access at least one record stored at least in part in a secure storage;

wherein the at least one record is associated at least with the external resource, and wherein the at least one record comprises:

a biometric template; and

a credential comprising at least one of a password, a cookie, or a cryptographic key;

in response to determining a match between a biometric input and the biometric template, retrieving, from the at least one record, the credential wherein the biometric input corresponds to at least one of a fingerprint, a feature usable for facial recognition, a voiceprint, a feature usable for a retina scan, or a typing feature, and wherein the biometric input is received subsequent to presenting of a prompt;

establishing a connection with the external resource;

facilitating a login of the user to the external resource at least in part by transmitting, via the established connection, output based at least in part on the credential retrieved from the at least one record, and wherein the user is logged in to the external resource based at least in part on the output; and

facilitating wiping of at least a portion of the at least one record.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 10,824,696 B1
APPLICATION NO. : 16/563715
DATED : November 3, 2020
INVENTOR(S) : Bjorn Markus Jakobsson

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

On the Title Page

In page 2, item (56), other publications, Column 2, cite no. 1, delete "https://tools.ieff.org/html/rfc5849" and insert --https://tools.ietf.org/html/rfc5849--, therefor.

Signed and Sealed this
Eighth Day of November, 2022

Katherine Kelly Vidal

Katherine Kelly Vidal
Director of the United States Patent and Trademark Office

Electronic Acknowledgement Receipt	
EFS ID:	37095024
Application Number:	16563715
International Application Number:	
Confirmation Number:	5312
Title of Invention:	AUTHENTICATION TRANSLATION
First Named Inventor/Applicant Name:	Bjorn Markus Jakobsson
Customer Number:	21912
Filer:	Yeu-Ting George Cheng/Elaine Nguyen
Filer Authorized By:	Yeu-Ting George Cheng
Attorney Docket Number:	MJAKP008C3
Receipt Date:	06-SEP-2019
Filing Date:	
Time Stamp:	19:36:29
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	DA
Payment was successfully received in RAM	$1135
RAM confirmation Number	E201996J36526693
Deposit Account	500685
Authorized User	Elaine Nguyen
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: 37 CFR 1.16 (National application filing, search, and examination fees) 37 CFR 1.17 (Patent application and reexamination processing fees)	

37 CFR 1.19 (Document supply fees)

37 CFR 1.20 (Post Issuance fees)

37 CFR 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Application Data Sheet	MJAKP008C3_ADS.pdf	1793735 504[illegible]fe6f47c21279108e64fdee5e122de2fe	no	9
Warnings:					
Information:					
2	Oath or Declaration filed	MJAKP008C3_Executed_Dec.pdf	93450 [illegible]97ed0e	no	1
Warnings:					
Information:					
3	Power of Attorney	MJAKP008C3_POA_AIA82A.pdf	198186 e07ca52c8a2da3a2180e412d7eb1b1c5c1[illegible]745b0	no	1
Warnings:					
Information:					
4	Change of Address	MJAK_Executed_POA_AIA82B_RightQuestionLLC.pdf	387899 a756b6df61aa99186ddcc71473[illegible]	no	1
Warnings:					
Information:					
5	Transmittal Letter	MJAKP008C3_IDS_01_Transmittal.pdf	79367 c19ae99a379bac14e822c23caa29027acbc066e1	no	2
Warnings:					
Information:					
6	Information Disclosure Statement (IDS) Form (SB08)	MJAKP008C3_IDS_01_SB08.pdf	1054150 fle75911c1750133ba6ffe3ffc75ffc0a1af5834	no	6
Warnings:					
Information:					

7	Specification	MJAKP008C3_APP.pdf	173379 142244e484e5cce16966fb56eb6612468d77124	no	23
Warnings:					
Information:					
8	Drawings-only black and white line drawings	MJAKP008C3_APP_Figures.pdf	112090 834879f7bd1fc2e79014265a447a54f43bdec227	no	8
Warnings:					
Information:					
9	Fee Worksheet (SB06)	fee-info.pdf	36629 8cb1955ebdc5a1c88019649e8066513c2634dec3	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	3928885		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

<u>New Applications Under 35 U.S.C. 111</u>
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.
<u>National Stage of an International Application under 35 U.S.C. 371</u>
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.
<u>New International Application Filed with the USPTO as a Receiving Office</u>
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Attorney Docket No. MJAKP008C3

APPLICATION FOR UNITED STATES PATENT

AUTHENTICATION TRANSLATION

By Inventor:

Bjorn Markus Jakobsson
Portola Valley, CA
A Citizen of Sweden

Assignee: RightQuestion, LLC

VAN PELT, YI & JAMES LLP
10050 N. Foothill Blvd., Suite 200
Cupertino, CA 95014
Telephone (408) 973-2585

AUTHENTICATION TRANSLATION

CROSS REFERENCE TO OTHER APPLICATIONS

[0001] This application is a continuation of co-pending U.S. Patent Application No. 16/273,797, entitled AUTHENTICATION TRANSLATION filed February 12, 2019 which is incorporated herein by reference for all purposes, which is a is a continuation of U.S. Patent Application No. 15/042,636, entitled AUTHENTICATION TRANSLATION filed February 12, 2016, now U.S. Patent No. 10,360,351, which is incorporated herein by reference for all purposes, which is a continuation of U.S. Patent Application No. 13/706,254, entitled AUTHENTICATION TRANSLATION filed December 5, 2012, now U.S. Patent No. 9,294,452, which is incorporated herein by reference for all purposes, which claims priority to U.S. Provisional Application No. 61/587,387, entitled BIOMETRICS-SUPPORTED SECURE AUTHENTICATION SYSTEM filed January 17, 2012 which is incorporated herein by reference for all purposes. U.S. Patent Application No. 13/706,254 also claims priority to U.S. Provisional Patent Application No. 61/569,112 entitled BACKWARDS COMPATIBLE ROBUST COOKIES filed December 9, 2011, which is incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

[0002] Providing credentials to a service, whether via a mobile or other device, is often a tedious experience for a user. Unfortunately, to make authentication easier for themselves, users will often engage in practices such as password re-use, and/or the selection of poor quality passwords, which render their credentials less secure against attacks. Accordingly, improvements in authentication techniques would be desirable. Further, it would be desirable for such improvements to be widely deployable, including on existing/legacy systems.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] Various embodiments of the invention are disclosed in the following detailed description and the accompanying drawings.

[0004] Figure 1 illustrates an embodiment of an environment in which authentication translation is provided.

[0005] Figure 2 illustrates an embodiment of credential information stored on a device.

[0006] Figure 3 illustrates an embodiment of a device with secure storage.

[0007] Figure 4 illustrates an example of a renegotiation.

[0008] Figure 5 illustrates an embodiment of a process for performing authentication translation.

[0009] Figure 6 illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator.

[0010] Figure 7 illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator.

[0011] Figure 8 shows the structure of an example of a cache cookie used in some embodiments.

DETAILED DESCRIPTION

[0012] The invention can be implemented in numerous ways, including as a process; an apparatus; a system; a composition of matter; a computer program product embodied on a computer readable storage medium; and/or a processor, such as a processor configured to execute instructions stored on and/or provided by a memory coupled to the processor. In this specification, these implementations, or any other form that the invention may take, may be referred to as techniques. In general, the order of the steps of disclosed processes may be altered within the scope of the invention. Unless stated otherwise, a component such as a processor or a memory described as being configured to perform a task may be implemented as a general component that is temporarily configured to perform the task at a given time or a specific component that is manufactured to perform the task. As used herein, the term 'processor' refers to one or more devices, circuits, and/or processing cores configured to process data, such as computer program instructions.

[0013] A detailed description of one or more embodiments of the invention is provided below along with accompanying figures that illustrate the principles of the invention. The invention is described in connection with such embodiments, but the invention is not limited to any embodiment. The scope of the invention is limited only by the claims and the invention encompasses numerous alternatives, modifications and equivalents. Numerous specific details are set forth in the following description in order to provide a thorough understanding of the invention. These details are provided for the purpose of example and the invention may be practiced according to the claims without some or all of these specific details. For the purpose of clarity, technical material that is known in the technical fields related to the invention has not been described in detail so that the invention is not unnecessarily obscured.

[0014] Figure 1 illustrates an embodiment of an environment in which authentication translation is provided. In the example shown, a variety of client devices 102-108 connect, via one or more networks (represented as a single network cloud 110) to a variety of services 120-124 (also referred to herein as sites 120-124). In particular, client device 102 is a notebook computer owned by a user hereinafter referred to as Alice. Notebook 102 includes a camera, a microphone, and a fingerprint sensor. Client device 104 is a smartphone, also owned by Alice.

Client device 104 includes a camera. Client device 106 is a tablet owned by Bob, and sometimes used by Bob's son Charlie. Client device 106 includes a camera and a fingerprint sensor. Client device 108 is a kiosk located in the lobby of a hotel. Kiosk 108 includes a camera and a microphone. The techniques described herein can be used with or adapted to be used with other devices, as applicable. For example, the techniques can be used in conjunction with gaming systems, with peripheral devices such as mice, and with embedded devices, such as door locks.

[0015] Service 120 is a social networking site. Service 122 is a website of a bank. Service 124 is the online store of a boutique camera retailer. Each of services 120-124 requires a username and password (and/or a cookie) from a user prior to giving that user access to protected content and/or other features. As will be described in more detail below, using the techniques described herein, users need not type such usernames and passwords into their devices whenever required by a service. Instead, users can authenticate themselves to an "authentication translator" via an appropriate technique, and the authentication translator will provide the appropriate credentials to the implicated service on the user's behalf. Also as will be described in more detail below, authentication translators can be located in a variety of places within an environment. For example, notebook computer 102 includes an authentication translator module 132 that provides authentication translation services. The other devices 104-108 can also include (but need not include) their own respective authentication translator modules. The owner of bank website 122 also operates an authentication translator 134 associated with the bank. Finally, authentication translator 136 provides authentication translation services to a variety of businesses, including online camera retailer 124.

[0016] Figure 2 illustrates an embodiment of credential information stored on a device. In particular, device 200 stores three user profiles 202-206, each of which contains a username and one or more templates (e.g., template 210) associated with the user. In various embodiments, a template is a collection of biometric features. Using fingerprints as an example type of biometric, a corresponding template includes a collection of patterns, minutia, and/or other features that can be matched against to determine if a person's fingerprint matches the fingerprint of the registered user (i.e., the owner of a given user profile). A representation of a single fingerprint may be included in multiple templates (e.g., in different resolutions, in accordance with different protocols, as captured during warm vs. cold conditions, and/or by itself

or in combination with multiple fingerprints). When other biometrics are employed (e.g., facial recognition, voiceprint, or retina scan technology), features appropriate to those types of biometrics are included in the template. Other types of features can also be included in templates. As one example, a user's typing speed and/or accuracy can be measured by a device, such as device 102, and used to distinguish between multiple users of a device. For example, suppose Alice types at 100 words per minute and rarely makes mistakes. A representation of this information can be stored in template 212. Also suppose Alice's niece, who sometimes uses Alice's laptop computer when visiting Alice types at 20 words per minute and makes many mistakes. In some embodiments, the fact that a user was recently (e.g., within the last 5 minutes) typing on laptop 102 at 90 words per minute is evidence of a match against template 212. In this case, the typing speed of 90 words per minute is similar enough to Alice's typical behavior, it is considered a match. Various policies can be included in a profile that govern how matches are to be performed. For example, policies can specify thresholds/tolerances for what constitutes a match, and can specify that different levels of matches can result in different levels of access to different resources.

[0017] A profile is associated with a vault (e.g., vault 220). The vault, in turn, contains triples specifying a service provider/domain, a username, and a credential. The vault can also contain other sensitive user information, such as account numbers, address/phone number information, and health care data. The credential for a service provider/domain can be a password (e.g., for legacy servers), and can also take alternate forms (e.g., a cryptographic key for service providers supporting stronger authentication methods).

[0018] In some embodiments, profiles, templates, and vaults (collectively "authentication information") are stored entirely in an unprotected storage area, and are stored in the clear. In other embodiments, secure storage techniques are used to secure at least a portion of the authentication information.

[0019] One example of a device with secure storage is illustrated in Figure 3. In the example shown, a mobile phone 300 includes a large and insecure storage 302 attached to a fast processor 304, and a smaller but secure storage 306 attached to a dedicated processor 308 and a sensor 310 (e.g., a camera or a fingerprint reader). Users (and applications) can read from and

write to the insecure storage area. However, users cannot access the secure storage area, and the fast processor can only communicate with the dedicated processor/sensor via a restricted API. As another example, a unique decryption key associated with a given vault can be stored in a profile. The vault is an encrypted and authenticated container that can be stored on insecure storage, e.g., on the device, and also backed up (e.g., to a cloud storage service 140 or to an alternate form of external storage). As needed, authentication information or portions thereof can be loaded into secure storage and decrypted. For example, one can use AES to encrypt the files one by one, using a key stored on the secured storage. A message authentication technique, such as HMAC, can be used for authenticating the encrypted files to provide tamper prevention. Profiles and vaults can be updated while in secure storage; if this occurs, they are encrypted and MACed before being written back to the insecure storage, which may in turn propagate them to external backup storage. In yet other embodiments, profiles and vaults are stored entirely in secure storage, in plaintext, which allows them to be both read and written -- and in particular, searched.

[0020] Example Transaction Types

[0021] A variety of transaction types can take place in the environment shown in Figure 1, examples of which are discussed in this section.

[0022] Initial Registration

[0023] In order to begin using the techniques described herein, users perform some form of initial registration. As one example, suppose Alice launches an enrollment program installed on laptop 102. She uses the program to capture various biometric information (e.g., fingerprints, photographs of her face, etc.). A user profile is created for Alice, and the biometric information captured about her is encoded into a plurality of templates, such as templates 210 and 214. In some embodiments, Alice is also explicitly asked to supply credential information for services she would like to use, such as by providing the domain name of social networking site 120, along with her username and password for site 120. In other embodiments, domain/username/credential information is at least passively captured on Alice's behalf and included in one or more vaults such as vault 220. Credential information can also be important from a browser password manager already in use by Alice or other appropriate source. In some

embodiments, Alice also registers with cloud storage service 140, which will allow her to back up her authentication information and to synchronize it across her devices (e.g., 102 and 104), as described in more detail below.

[0024] Other registration approaches can also be used. For example, registration can be integrated into the experience the first time a device is used. Thus, when Bob first turns on tablet 106, he may be prompted to take a picture of his face (with a profile/templates being created in response). Similarly, the first time Charlie uses tablet 106, the techniques described herein can be used to determine that Charlie does not yet have a profile (e.g., because none of the templates already present on tablet 106 match his biometrics) and Charlie can be prompted to enroll as a second user of the device.

[0025] Authentication

[0026] Suppose Alice wishes to authenticate to banking website 122. Using a fingerprint reader incorporated into her laptop, she performs a fingerprint scan, which causes her biometric features to be extracted and compared to any stored templates residing on her computer. If a match is found, an associated decryption key is selected, and the associated vault is loaded and decrypted. The vault is scanned for an entry that matches the selected service provider (i.e., website 122). If a matching entry is found, the associated domain, username, and site credential are extracted from the vault. In some embodiments, the validity of the domain name mapping is verified at this point to harden the system against domain name poisoning. Next, a secure connection is established between Alice's computer and the service provider, and Alice is authenticated. For service providers supporting strong user authentication, mutual SSL can be used, for example. A variety of policies can be involved when performing matching. For example, to access certain domains, Alice's print may need only match template 210. To access other domains, Alice may need to match multiple templates (e.g., both 210 and 214). As another example, in order to access social networking site 120, Alice may merely need to be sitting in front of her computer, which has an integrated webcam. Even in relatively low light conditions, a match can be performed against Alice's face and features stored in a template. However, in order to access bank website 122, Alice may need a high quality photograph (i.e., requiring her to turn on a bright light) and may need to demonstrate liveness (e.g., by blinking or turning her

head). As yet another example, other contextual information can be included in policies. For example, if Alice's IP address indicates she is in a country that she is not usually in, she may be required to match multiple templates (or match a template with more/better quality features) in order to access retailer 124, as distinguished from when her IP address indicates she is at home.

[0027] In some embodiments, the biometric sensor used by a user may be a peripheral device (e.g., a mouse with an integrated fingerprint scanner that is connected to the user's primary device via USB). In such scenarios, the peripheral device may be responsible for storing at least a portion of authentication information and may perform at least some of the authentication tasks previously described as having been performed by Alice's computer. For example, instead of processors 304 and 308, and storages 302 and 306 being collocated on a single device (e.g., laptop 102), processor 304 and storage 302 may be present on a primary device, and processor 308 and storage 306 may be present on a peripheral device (e.g., that also includes a sensor, such as a fingerprint reader).

[0028] In such scenarios, once Alice's login to banking website 122 is successfully completed, the secure session can be handed over from the peripheral device to the primary device, in a way that does not allow the primary device retroactive access to the plaintext data of the transcripts exchanged between the peripheral device and the service provider. One way this can be accomplished is by renegotiating SSL keys between the peripheral device and the website, after which the newly negotiated key can be handed off from the peripheral device to the primary device. This avoids retroactive credential capture in a setting where the device is infected by malware.

[0029] An example of renegotiation is depicted in Figure 4. Specifically, after a user has successfully authenticated to a fingerprint reader, a login is performed to a service provider. Using the primary device (404) as a proxy, the peripheral fingerprint reader 402 negotiates a first SSL connection (408) with a service provider 406, over which credentials are exchanged. The proxy then renegotiates SSL (410), which replaces the old key with a new one. The new key is disclosed to the device, which then seamlessly takes over the connection with the service provider and performs the transaction protected by the authentication. The credentials exchanged during the first SSL connection cannot be accessed by device 404, since the key of the

renegotiated session is independent of the key of the original session; this provides protection against malware residing on the device. Renegotiation can be used when the primary device 404 is believed to be in a safe state when performing the negotiation of the SSL connection, but it is not known whether it is in a safe state during the transaction protected by the authentication. Renegotiation can also be used when a secure component of the primary device 404 performs the negotiation of the SSL connection and another and potentially insecure component of the primary device 404 is involved in the transaction protected by the authentication.

[0030] Figure 5 illustrates an embodiment of a process for performing authentication translation. The process begins at 502 when a request to access a resource is received, as is an authentication input. One example of the processing performed at 502 is as follows. Suppose Alice wishes to sign into social networking website 120. She directs a web browser application installed on client 102 to the social networking website. Authentication translator module 132 recognizes, from the context of Alice's actions (e.g., that she is attempting to access site 120 with her browser) that she would like to access a particular resource. Authentication translator module 132 prompts Alice (e.g., by a popup message or via a sound) to provide biometric information (e.g., to use the integrated fingerprint reader on her laptop). In some embodiments, the translator module does not prompt Alice, for example, because Alice has been trained to provide biometric information automatically when attempting to access certain resources. In yet other embodiments, the translator module only prompts Alice if she fails to provide acceptable biometric information within a timeout period (e.g., 30 seconds).

[0031] Module 132 compares Alice's supplied biometric data to the templates stored on her computer. If a suitable match is found, and if an entry for site 120 is present in the applicable vault, at 504, a previously stored credential associated with the resource is accessed. In particular, the username and password for the website, as stored in a vault, such as vault 220, are retrieved from the vault.

[0032] Finally, at 506, the credential is provided to the resource. For example, Alice's username and password for site 120 are transmitted to site 120 at 506. The credential can be transmitted directly (e.g., by the module or by Alice's computer) and can also be supplied

indirectly (e.g., through the use of one or more proxies, routers, or other intermediaries, as applicable).

[0033] Other devices can also make use of process 500 or portions thereof. For example, when Alice launches a banking application on phone 104, implicit in her opening that application is her desire to access the resources of website 134. The application can take Alice's picture and compare it to stored templates/vault information. If an appropriate match is found, a credential can be retrieved from the vault on her phone (or, e.g., retrieved from cloud storage service 140) and provided to website 134.

[0034] As another example, suppose Charlie is using tablet 106 and attempts to visit site 120, whether via a dedicated application or via a web browser application installed on the tablet. Charlie's photograph is taken, and then compared against the profiles stored on tablet 106 (e.g., both Bob and Charlie's profiles). When a determination is made that Charlie's photograph matches a template stored in his stored profile (and not, e.g., Bob's), Charlie's credentials for site 120 are retrieved from a vault and transmitted by an authentication translator module residing on client 106.

[0035] As yet another example, kiosk 108 can be configured to provide certain local resources (e.g., by displaying a company directory or floor plan on demand) when users speak certain requests into a microphone. Enrolled users (e.g., with stored voiceprint or facial recognition features) can be granted access to additional/otherwise restricted services in accordance with the techniques described herein and process 500.

[0036] New device

[0037] In some embodiments, to register a new device, a user provides an identifier, such as a username or an account number to the device. The new device connects to an external storage (such as cloud storage 140), provides the user identifier and credential, and downloads the user's templates/vaults from the service. In some embodiments, the templates/vaults are encrypted. Once downloaded, the template is decrypted and stored in a secure storage area, while the still encrypted vault can be stored in insecure storage. The decryption key can be generated from information the user has/knows, or from biometric data -- such as features

extracted from fingerprinting of all ten fingers. In some embodiments, more arduous fingerprinting is required for the setup of a new device than for regular authentication to avoid that a new device gets registered by a user thinking she is merely authenticating -- or worse still, simply touching the device. Moreover, it translates into higher entropy of the decryption keys.

[0038] Backup Authentication

[0039] Backup authentication allows a user, such as Alice, to access resources in the event she is unable to or unwilling to interact with a particular biometric sensor. As one example, instead of having a single template associated with her profile, Alice can have multiple templates associated with it, e.g., where the first template includes fingerprint features and the second template includes voice biometric, facial recognition, or iris detection features. As a second example, where the service Alice is connecting to is a legacy website (i.e., one that users authenticate to using usernames and passwords), such a service would allow the use of passwords and password reset mechanisms by Alice without requiring Alice to use a fingerprint reader.

[0040] In various embodiments, environment 100 supports the ability of users (e.g., under duress) to release the contents of their vaults. For example, if Alice was physically threatened with the loss of a finger by a criminal, Alice could instead release the contents of her vault(s) – the ultimate goal of the criminal. As one example, in the event Alice supplies all 10 fingerprints to the sensor, provides a special password, or supplies a fingerprint and a second identifier, a cleartext version of her vault(s) could be made available.

[0041] Access Policies

[0042] In various embodiments, cloud storage service 140 is configured to accept backups from multiple devices associated with a single account, and synchronize the updates so that all devices get automatically refreshed. For example, Alice's laptop 102 and phone 104 could both communicate with cloud storage service 140 which would keep their authentication information synchronized. Refreshes can also be made in accordance with user-configured restrictions. For example, Alice's employer could prevent privileged employer data from being stored on shared personal devices, or on any device that was not issued by the employer. As

another example, arbitrary policies can be defined regarding the access to and synchronization of software and data, and to tie a license or access rights to a person (and her fingerprint) rather than to a device. As yet another example, in some embodiments (e.g., where a device is made publicly available or otherwise shared by many users), no or a reduced amount of authentication information resides on a device, and at least a portion of authentication information is always retrieved from cloud storage service 140.

[0043] Remote wiping

[0044] Remote wiping of a user's authentication information (e.g., templates) can be used both to "unshare" previously shared devices (e.g., where Bob and Charlie both have user profiles on their shared tablet 106), and to avoid that criminals with physical component access to lost devices gain access to templates and vault contents. In some embodiments, polices such as ones where a template self-wipes if it is not matched within a particular duration of time are supported. Since user data can be frequently backed up to the cloud storage, and recovered from this using the new device registration process, inconvenience to the user will be minimized.

[0045] **Legacy Server Support**

[0046] New authentication schemes typically require changes to a significant codebase residing with service providers. If the code is well written and documented, such changes may be relatively simple. Commonly, though, this may not be so. The engineers who originally wrote the code of relevance may have long since left the company; the code they left behind may be poorly documented -- if documented at all. In severe cases, the legacy code may have been written in an outdated programming language or written in a way that does not follow guidelines for good code. This makes updates to the codebase impractical or virtually impossible in many common cases. Even if none of these challenges complicate the desired modifications, it is commonly a great bureaucratic burden to obtain permission to make modifications (e.g., to store an extra field in a backend database), because every affected part of the organization may need to review the request.

[0047] As will be described in the following section, the technologies described herein can be used in conjunction with legacy servers (e.g., existing servers that rely on usernames and

passwords to authenticate users), and in particular, can be used without requiring modification to such legacy servers.

[0048] Cookies

[0049] Cookies are commonly used by legacy servers for user authentication. Unfortunately, cookies have several problems. For one thing, they are sometimes deleted -- whether explicitly/intentionally by the end user or by the user's software. In addition, cookies are commonly stolen. Approaches such as cache cookies and identification using user agents can be more resistant to these problems, however, they have their own problems. For example, their use requires new code and new fields in the credential database stored by the server.

[0050] In some embodiments, authentication translators, such as translators 134 and 136 (also referred to herein as proxies) provide authentication translation services on behalf of associated services. Translators 134 and 136 are illustrated as single logical devices in Figure 1. In some embodiments, the translators comprise standard commercially available server hardware (e.g., a multi-core processor, 4+ Gigabytes of RAM, and one or more Gigabit network interface adapters) and run typical server-class operating systems (e.g., Linux). Translators 134 and 136 can also be implemented using a scalable, elastic architecture and may comprise several distributed components, including components provided by one or more third parties. For example, translators 134 and 136 may store user credential information or may task cloud storage service 140 with storing at least a portion of that information.

[0051] In the case of authentication translator 134, service is provided with respect to bank website 122 only. Authentication translator 134 is positioned between a legacy web server (122) and the Internet (110) -- and therefore between the legacy server and any client devices. Authentication translator 134 is configured to translate traffic between the legacy server and client devices so that the client devices (and respective users) perceive the new authentication mechanism, while the legacy server remains unchanged. Authentication translator 136 works similarly, but it provides authentication translation as a third party service to multiple providers, an example of which is online camera retailer 124.

[0052] Authentication translators 134 and 136 can also perform process 500. For example, when a device transmits a request to access website 122, the request is intercepted by translator 134, as is cookie/user agent information. The received information can be used to determine a username/password associated with the device, and that information can be passed by translator 134 to website 122 on behalf of the device.

[0053] Figure 6 illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator. The translator (referred to in the figure as a "proxy") fails to identify the client, and passes on the request to the legacy server. The legacy server responds to the request and sets a cookie. The proxy passes on the response, including the cookie and also a cache cookie. The proxy stores the information about both these types of identifiers, along with the user agent of the client device. This triplet of information is also referred to herein as an identifier.

[0054] Figure 7 illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator. In this scenario, the device's request is accompanied by some form of identifying information. The proxy uses this information to identify the associated cookie and passes this along to the legacy server, along with the request. In some embodiments, additional processing is also involved. For example, reading a cache cookie may require user interaction. Moreover, if not all of the identifying information is present in the request, the proxy can be configured to set the missing information again by sending a corresponding request to the client device.

[0055] The translation of cache cookies and user agent information to cookies involves a two-way translation. First, when the legacy server sets a cookie, the proxy will set the two types of cookies -- both an HTML cookie and a cache cookie -- and then create a new record in which the two cookies are stored, along with the user agent information of the client device. The user agent information can include quite a bit of data associated with a browser -- such as the browser type and version, the clock skew, and the fonts that are installed. While each of these pieces of information only contributes a small amount of entropy, the collection of items can be sufficient to identify the device in most cases. Moreover, while some of these types of data may change over time -- in fact, all of them may -- they do not typically change, and when one or two of

them do, the others typically do not. When the client device is used to visit a site controlled by the legacy server, the cookie, cache cookie and user agent information are read (if available), the record identified, and the request translated and sent to the legacy server. When a legacy server requests that the user password is updated (e.g., as part of an annual or other periodic requirement), the transmission of this request to the user can be suppressed -- in which case the database of the proxy is updated to create the illusion of an update. The user can be involved in authentication as needed, e.g., if, in addition to supplying a credential, a user must also solve a CAPTCHA, the CAPTCHA can be displayed to the user (with the user's credentials being handled in accordance with the techniques described herein).

[0056] Figure 8 shows the structure of an example of a cache cookie used in some embodiments. Cache cookies can be associated with a particular webpage, just like an HTML cookie can. In the example shown in Figure 8, the proxy wishes to associate a page "calling page.html" with a cache cookie. It embeds a request for a second object, "samename.html" in callingpage.html for every visitor. However, as the cache cookie is set for one visitor, a customized samename.html is served to this visitor. The page samename.html refers to an object with a different name for each user; that object is referred to as "uniquename.jpg." The cache cookie is set by embedding the request for samename.html in callingpage.html. The client browser attempts to render this, causing a request for samename.html from the server. The server configures samename.html to refer to a uniquely named file uniquename.jpg, and serves samename.html to the client. For the client browser to render samename.html, it requests the file uniquename.jpg, which is intentionally not served. That concludes the setting of a cache cookie. As a user returns to the page callingname.html, the browser again attempts to render the entire page, which causes it to load the object samename.html from its cache. As that is rendered, the client browser requests uniquename.jpg, which is not in its cache (since it was not served previously). The server still does not serve it, but takes note of the name of the file being requested, as it identifies the client device. Note that samename.html can be displayed in a zero-sized iframe, which makes the end user unaware of it being rendered.

[0057] A cache cookie is an implementation of the typical cookie functionality that uses the client device's browser cache. Unlike user agents, it does not change over time, and like standard HTML cookies, it cannot be read by a party other than that which set it. However, like

HTML cookies, it could be deleted -- by the user clearing his or her browser cache. Cache cookies are not automatically transmitted with GET requests, unless the cache elements are embedded in the referring pages. This adds a potential round of communication in some settings. By relying on user agent information, cache cookies, and HTML cookies to identify the client device, it is much more likely that a machine will be recognized than if only HTML cookies are used.

[0058] Although the foregoing embodiments have been described in some detail for purposes of clarity of understanding, the invention is not limited to the details provided. There are many alternative ways of implementing the invention. The disclosed embodiments are illustrative and not restrictive.

[0059] WHAT IS CLAIMED IS:

CLAIMS

1. A system, comprising:

a first processor configured to:

based at least in part on a request from a user to access an external resource, communicate with a second processor using a restricted interface, wherein the second processor is configured to access a record stored in a secure storage;

wherein the record is associated at least with the external resource, and wherein the record comprises:

a biometric template; and

at least one of a password, a cookie, and a cryptographic key;

in response to determining a match between a biometric input and the biometric template, retrieve, from the record, the at least one of the password, the cookie, and the cryptographic key, wherein the biometric input corresponds to at least one of a fingerprint, a feature usable for facial recognition, a voiceprint, a feature usable for a retina scan, and a typing feature, and wherein the biometric input is received subsequent to presenting of a prompt;

establish a connection with the external resource; and

facilitate a login of the user to the external resource at least in part by transmitting, via the established connection, output based at least in part on the at least one of the password, the cryptographic key, and the cookie retrieved from the record, and wherein the user is logged in to the external resource based at least in part on the output; and

facilitate wiping of at least a portion of the record; and

a memory coupled to the first processor and configured to provide the first processor with instructions.

2. The system recited in claim 1 wherein the first processor is unable to directly access the secure storage.

3. The system recited in claim 1 wherein the second processor comprises a dedicated processor, and wherein the secure storage is connected to the dedicated processor.

4. The system recited in claim 1 wherein the secure storage is connected to a sensor.

5. The system recited in claim 4 wherein the sensor comprises at least one of a camera and a fingerprint reader.

6. The system recited in claim 1 wherein the record is stored in plaintext in the secure storage.

7. The system recited in claim 1 wherein the prompt comprises a prompt to provide biometric information, and wherein the prompt is visually presented.

8. The system recited in claim 1 wherein the prompt comprises a prompt to provide biometric information, and wherein the prompt is aurally presented.

9. The system recited in claim 1 wherein the prompt is presented in response to a user failing to provide acceptable biometric information within a timeout period.

10. The system recited in claim 1 wherein facilitating wiping of the at least portion of the record comprises facilitating remote wiping of the at least portion of the record.

11. The system recited in claim 1 wherein the at least portion of the record is automatically wiped based at least in part on a policy.

12. The system recited in claim 1 wherein the biometric template is wiped in response to determining that the biometric template has not been matched within a duration of time.

13. The system recited in claim 1 wherein the at least portion of the record is backed up to a remote entity.

14. A method, comprising:

based at least in part on a request from a user to access an external resource, using a first processor to communicate with a second processor using a restricted interface, wherein the second processor is configured to access a record stored in a secure storage;

wherein the record is associated at least with the external resource, and wherein the record comprises:

a biometric template; and

at least one of a password, a cookie, and a cryptographic key;

in response to determining a match between a biometric input and the biometric template, retrieving, from the record, the at least one of the password, the cookie, and the cryptographic key, wherein the biometric input corresponds to at least one of a fingerprint, a feature usable for facial recognition, a voiceprint, a feature usable for a retina scan, and a typing feature, and wherein the biometric input is received subsequent to presenting of a prompt;

establishing a connection with the external resource; and

facilitating a login of the user to the external resource at least in part by transmitting, via the established connection, output based at least in part on the at least one of the password, the cryptographic key, and the cookie retrieved from the record, and wherein the user is logged in to the external resource based at least in part on the output; and

facilitating wiping of at least a portion of the record.

15. The method of claim 14 wherein the first processor is unable to directly access the secure storage.

16. The method of claim 14 wherein the second processor comprises a dedicated processor, and wherein the secure storage is connected to the dedicated processor.

17. The method of claim 14 wherein the secure storage is connected to a sensor.

18. The method of claim 17 wherein the sensor comprises at least one of a camera and a fingerprint reader.

19. The method of claim 14 wherein the record is stored in plaintext in the secure storage.

20. The method of claim 14 wherein the prompt comprises a prompt to provide biometric information, and wherein the prompt is visually presented.

21. The method of claim 14 wherein the prompt comprises a prompt to provide biometric information, and wherein the prompt is aurally presented.

22. The method of claim 14 wherein the prompt is presented in response to a user failing to provide acceptable biometric information within a timeout period.

23. The method of claim 14 wherein facilitating wiping of the at least portion of the record comprises facilitating remote wiping of the at least portion of the record.

24. The method of claim 14 wherein the at least portion of the record is automatically wiped based at least in part on a policy.

25. The method of claim 14 wherein the biometric template is wiped in response to determining that the biometric template has not been matched within a duration of time.

26. The method of claim 14 wherein the at least portion of the record is backed up to a remote entity.

27. A computer program product embodied in a non-transitory computer readable storage medium and comprising computer instructions for:

based at least in part on a request from a user to access an external resource, using a first processor to communicate with a second processor using a restricted interface, wherein the second processor is configured to access a record stored in a secure storage;

wherein the record is associated at least with the external resource, and wherein the record comprises:

a biometric template; and

at least one of a password, a cookie, and a cryptographic key;

in response to determining a match between a biometric input and the biometric template, retrieving, from the record, the at least one of the password, the cookie, and the cryptographic key, wherein the biometric input corresponds to at least one of a fingerprint, a feature usable for facial recognition, a voiceprint, a feature usable for a retina scan, and a typing feature, and wherein the biometric input is received subsequent to presenting of a prompt;

establishing a connection with the external resource; and

facilitating a login of the user to the external resource at least in part by transmitting, via the established connection, output based at least in part on the at least

one of the password, the cryptographic key, and the cookie retrieved from the record, and wherein the user is logged in to the external resource based at least in part on the output; and

facilitating wiping of at least a portion of the record.

ABSTRACT OF THE DISCLOSURE

Authentication translation is disclosed. A request to access a resource is received at an authentication translator, as is an authentication input. The authentication input corresponds to at least one stored record. The stored record is associated at least with the resource. In response to the receiving, a previously stored credential associated with the resource is accessed. The credential is provided to the resource.

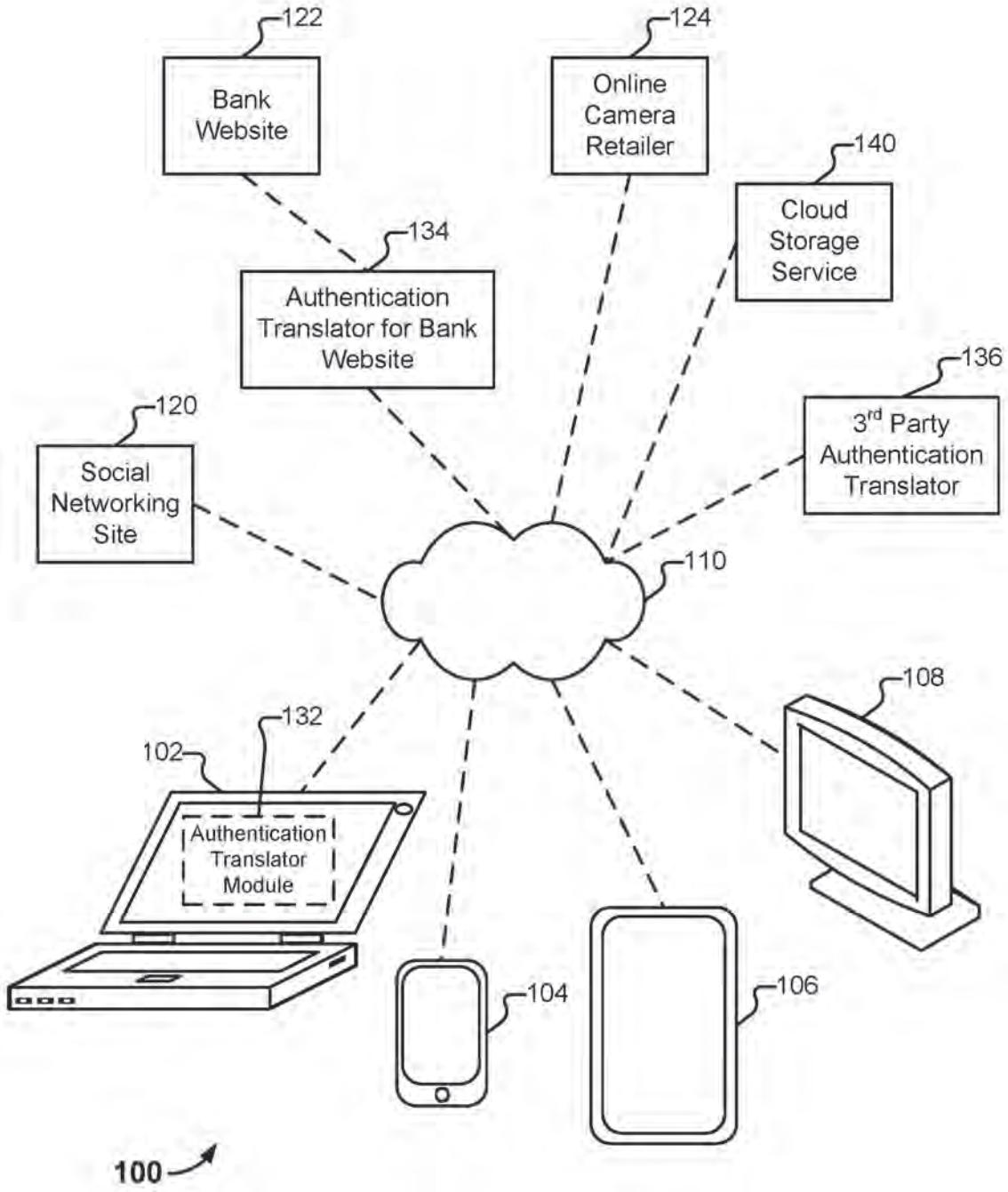

122
Bank Website
124
Online Camera Retailer
140
Cloud Storage Service
134
Authentication Translator for Bank Website
136
3rd Party Authentication Translator
120
Social Networking Site
110
108
132
102
Authentication Translator Module
104
106
100

FIG. 1

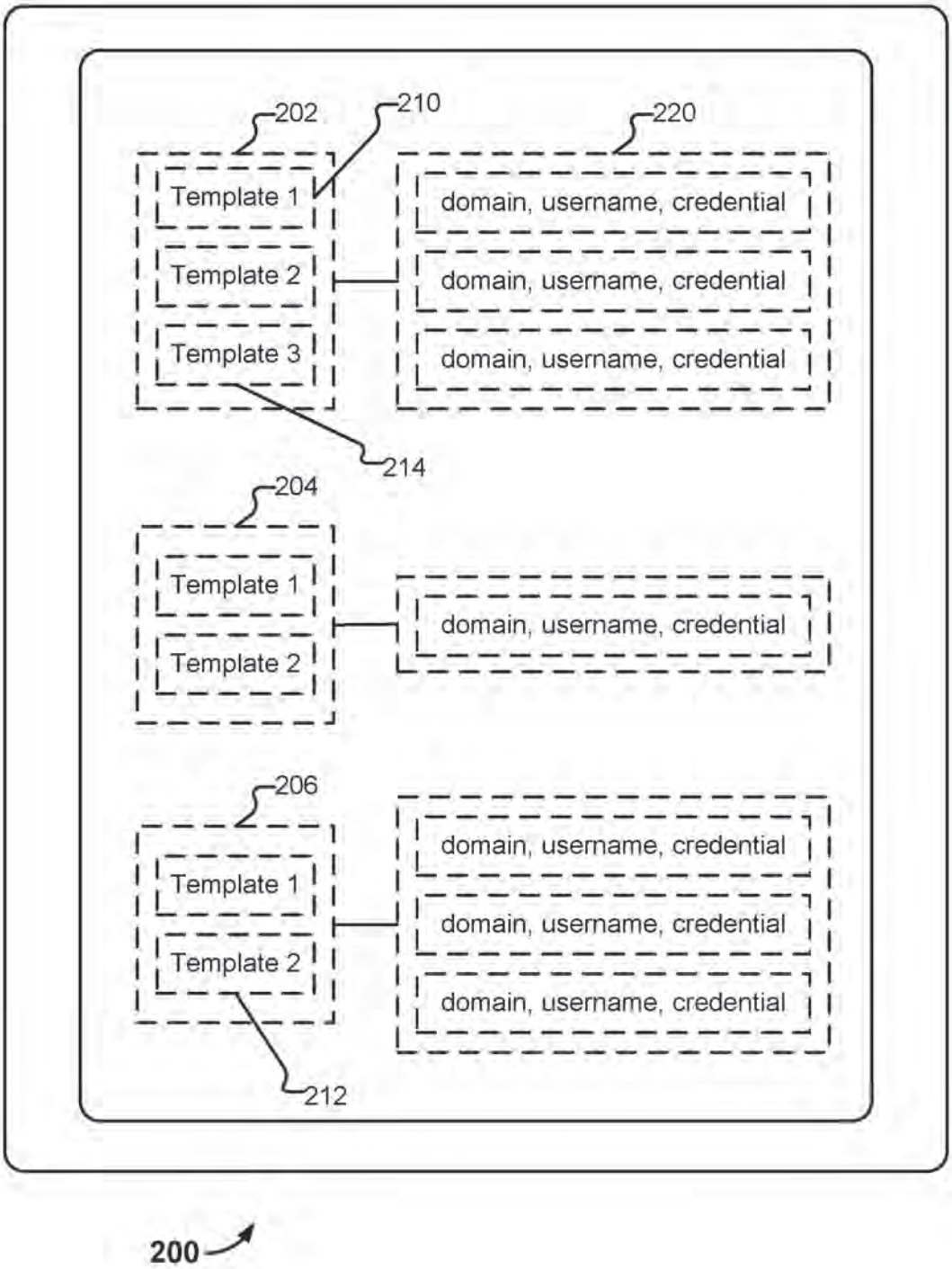
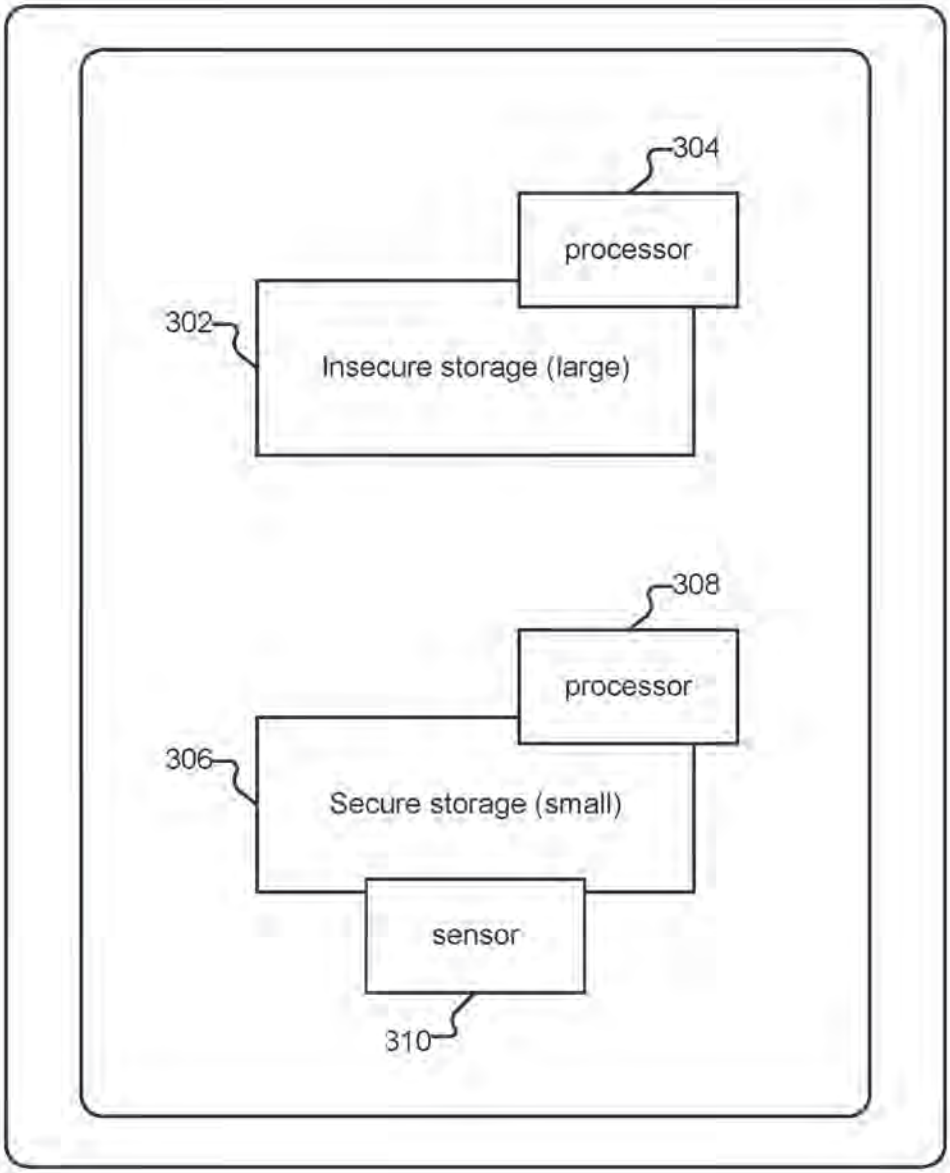

202
210
220
Template 1
Template 2
Template 3
domain, username, credential
domain, username, credential
domain, username, credential
214
204
Template 1
Template 2
domain, username, credential
206
Template 1
Template 2
domain, username, credential
domain, username, credential
domain, username, credential
212
200

FIG. 2


304
processor
302
Insecure storage (large)
308
processor
306
Secure storage (small)
sensor
310

300

FIG. 3

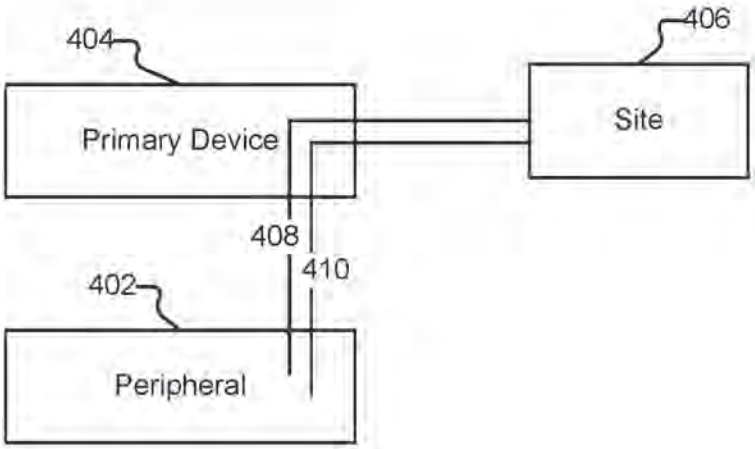


FIG. 4

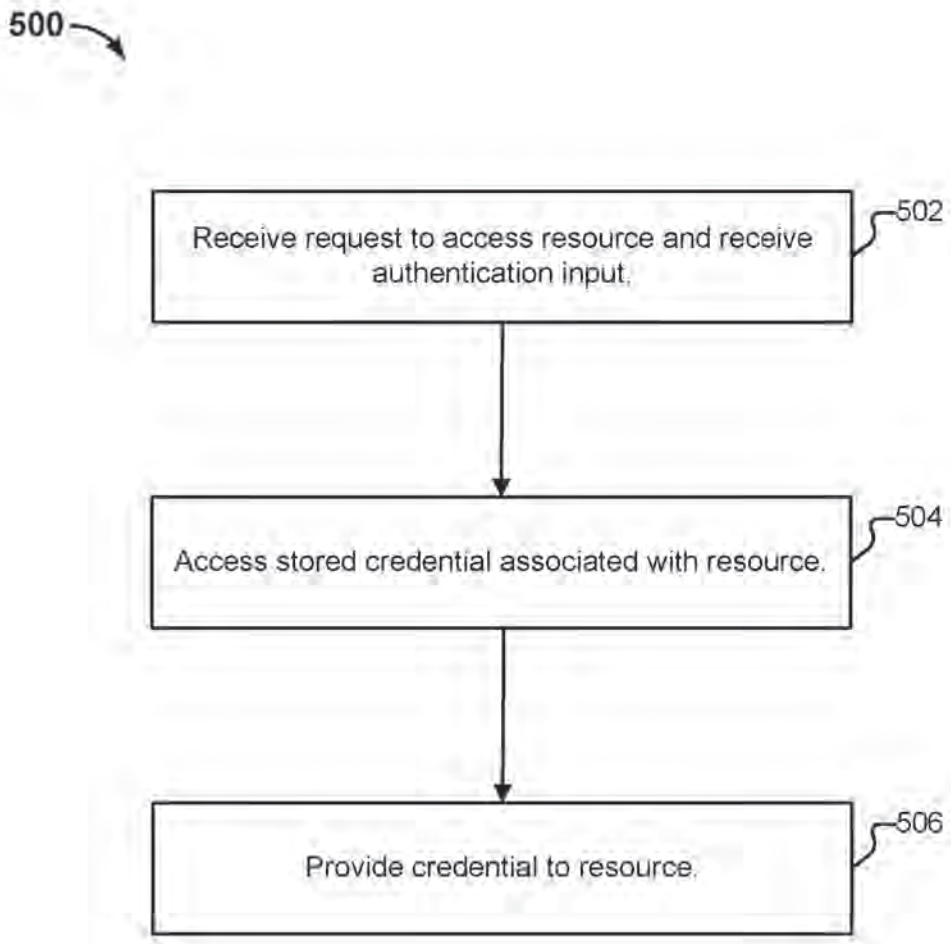

500
Receive request to access resource and receive authentication input.
502
Access stored credential associated with resource.
504
Provide credential to resource.
506

FIG. 5

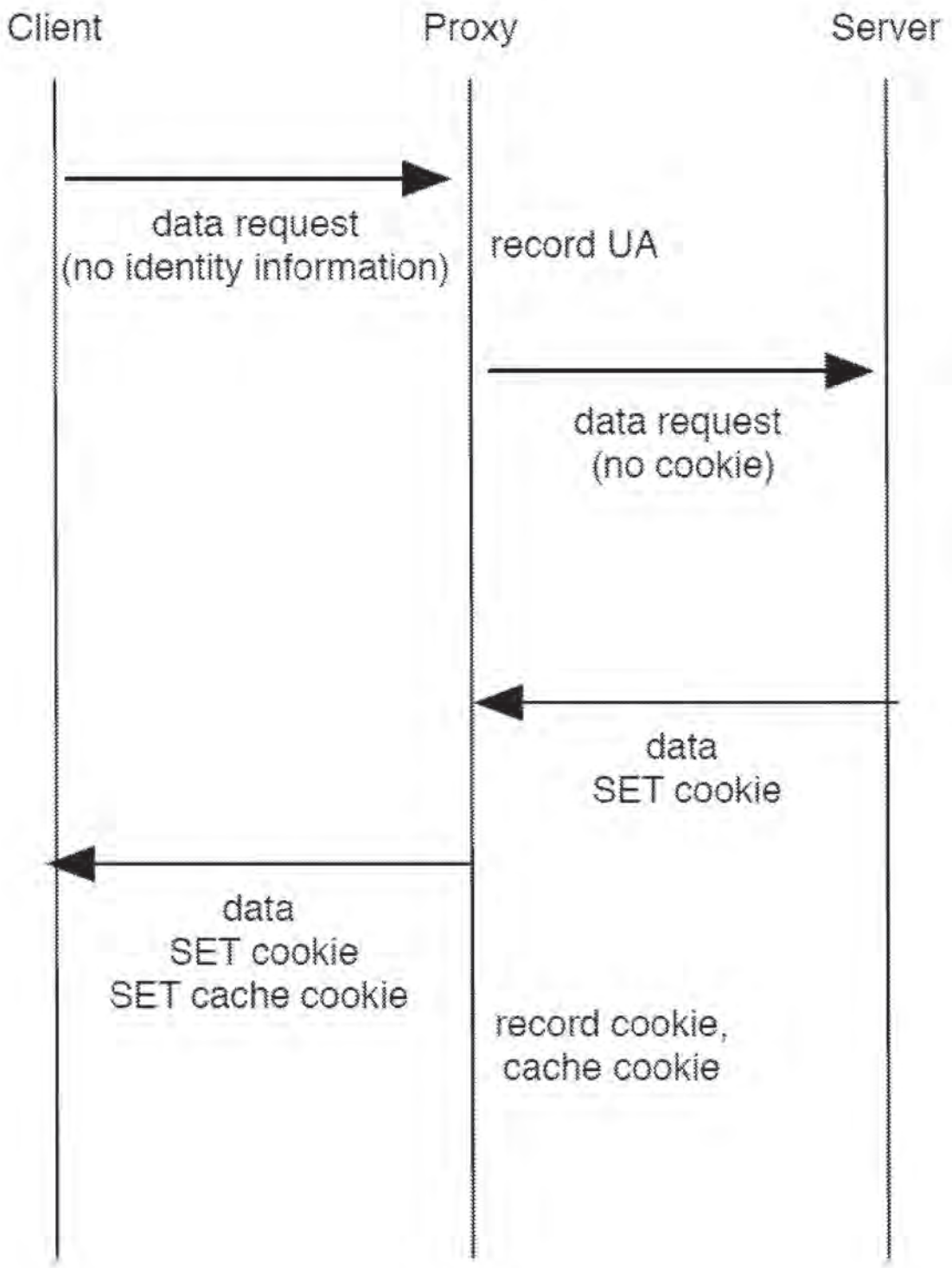

Client
Proxy
Server
data request
(no identity information)
record UA
data request
(no cookie)
data
SET cookie
data
SET cookie
SET cache cookie
record cookie,
cache cookie

FIG. 6

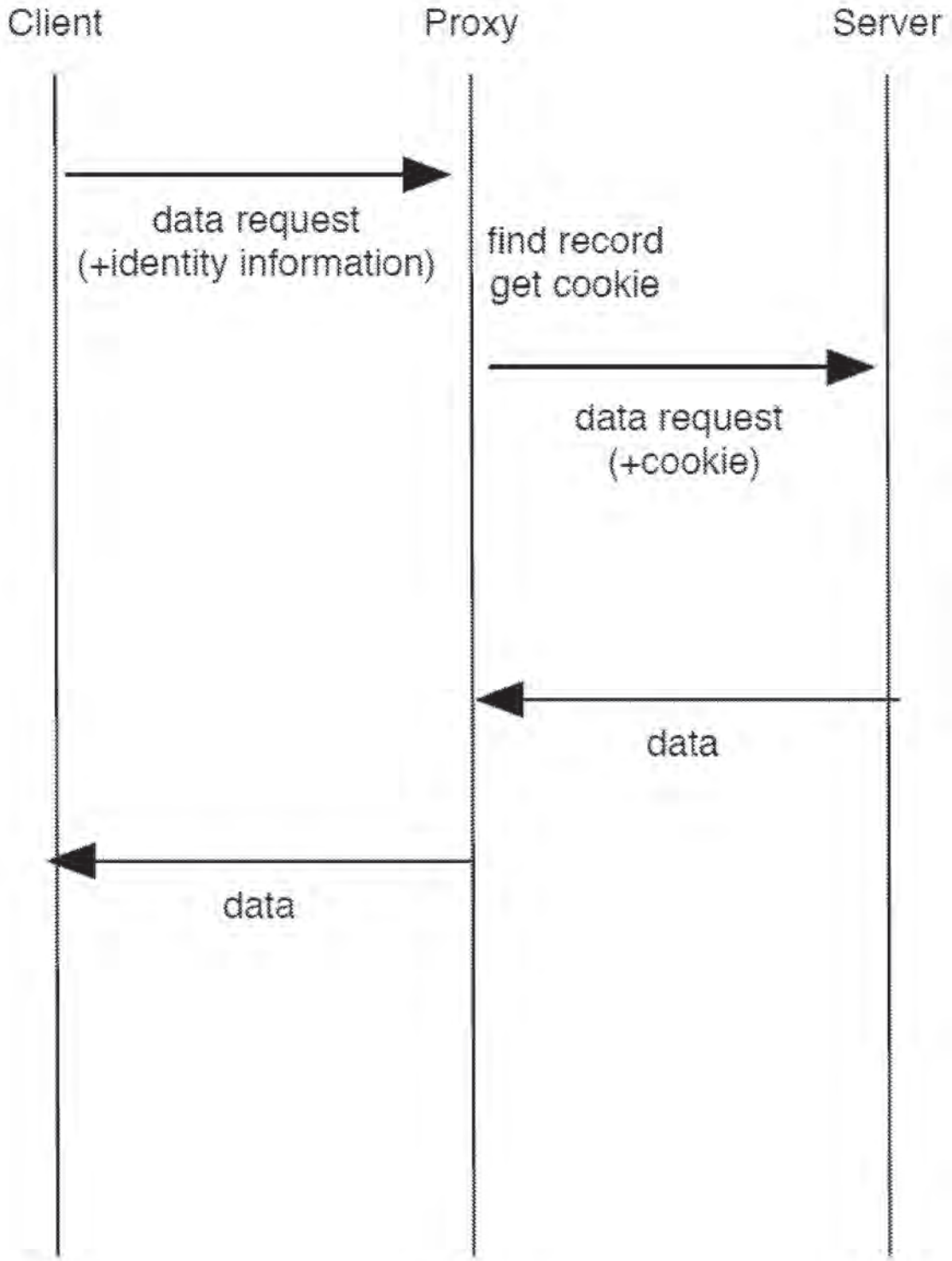

Client
Proxy
Server
data request
(+identity information)
find record
get cookie
data request
(+cookie)
data
data

FIG. 7

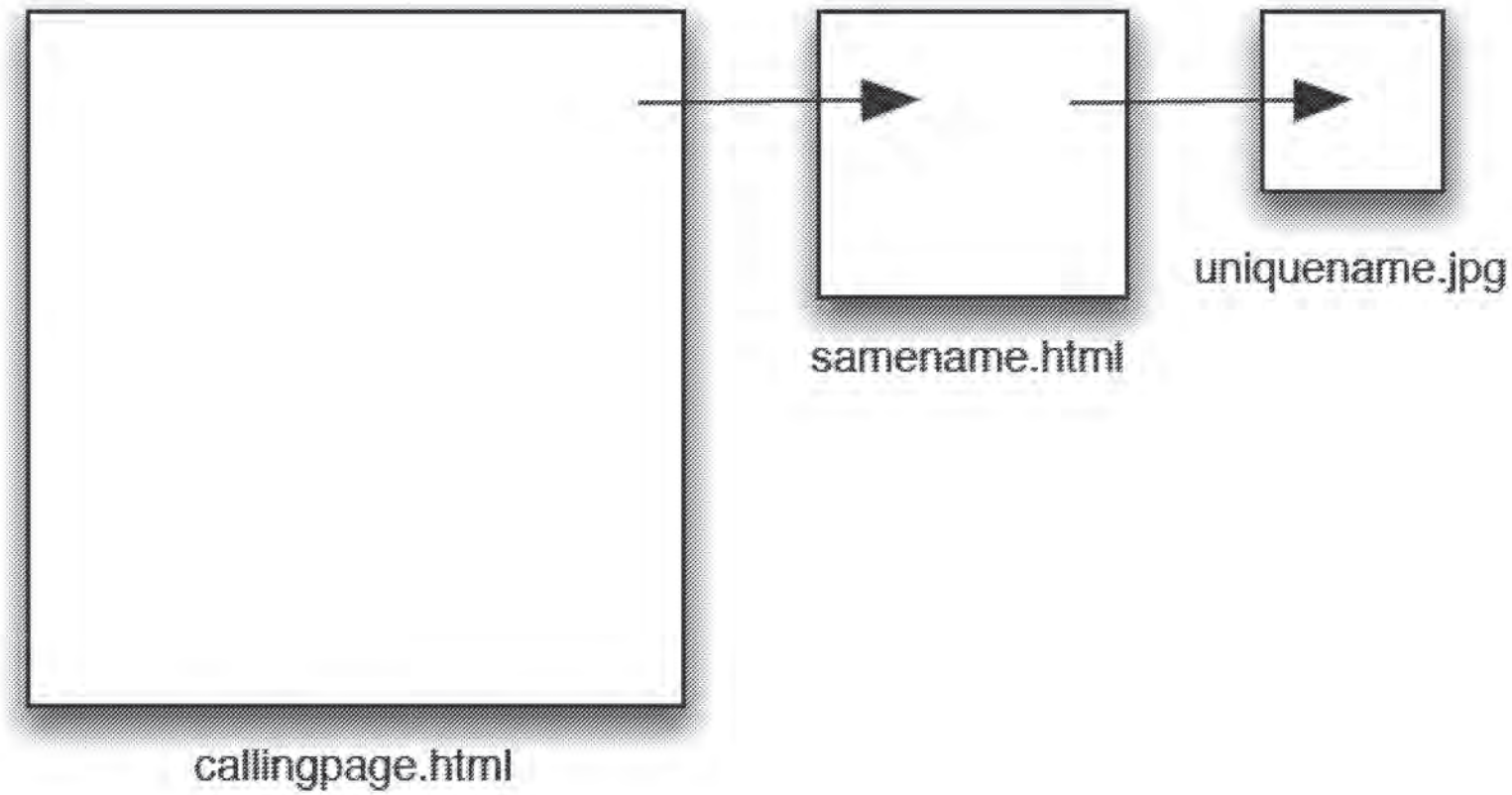


FIG. 8

Application Serial No. 16/273,797

Filing date: February 12, 2019

Patent No. 10,521,568

Issue date: December 31, 2019

US010521568B1

(12) **United States Patent**
Jakobsson

(10) **Patent No.: US 10,521,568 B1**
(45) **Date of Patent: *Dec. 31, 2019**

(54) **AUTHENTICATION TRANSLATION**

(71) Applicant: **RightQuestion, LLC**, Portola Valley, CA (US)

(72) Inventor: **Bjorn Markus Jakobsson**, Portola Valley, CA (US)

(73) Assignee: **RightQuestion, LLC**, Portola Valley, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **16/273,797**

(22) Filed: **Feb. 12, 2019**

Related U.S. Application Data

(63) Continuation of application No. 15/042,636, filed on Feb. 12, 2016, now Pat. No. 10,360,351, which is a continuation of application No. 13/706,254, filed on Dec. 5, 2012, now Pat. No. 9,294,452.

(60) Provisional application No. 61/569,112, filed on Dec. 9, 2011, provisional application No. 61/587,387, filed on Jan. 17, 2012.

(51) **Int. Cl.**

G06F 21/00	(2013.01)
G06F 21/10	(2013.01)
H04L 29/06	(2006.01)
G06F 21/12	(2013.01)
G06F 21/31	(2013.01)
G06F 21/32	(2013.01)
G06F 21/44	(2013.01)

(52) **U.S. Cl.**
CPC ***G06F 21/10*** (2013.01); ***G06F 21/121*** (2013.01); ***G06F 21/128*** (2013.01); ***G06F 21/31*** (2013.01); ***G06F 21/32*** (2013.01); ***G06F 21/44*** (2013.01); ***H04L 63/083*** (2013.01); ***H04L 63/0861*** (2013.01); ***H04L 63/10*** (2013.01); ***H04L 63/20*** (2013.01)

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,016,476	A	1/2000	Maes
7,512,965	B1	3/2009	Amdur
7,697,729	B2	4/2010	Howell

(Continued)

FOREIGN PATENT DOCUMENTS

WO	2004051585	A2	6/2004
WO	2005001751	A1	1/2005

OTHER PUBLICATIONS

"Managing Authorization and Access Control", Author: unknown, Published Nov. 3, 2005, pp. 1-12, URL: http://technet.microsoft.com/en-us/library/bb457115.aspx.

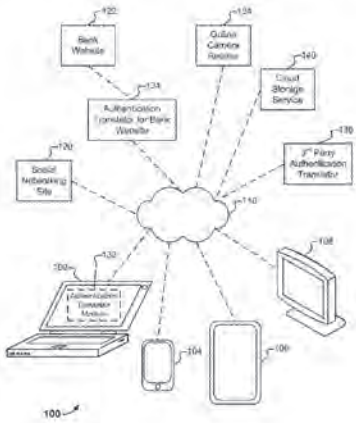
(Continued)

Primary Examiner — Andrew J Steinle
(74) *Attorney, Agent, or Firm* — Van Pelt, Yi & James LLP

(57) **ABSTRACT**

Authentication translation is disclosed. A request to access a resource is received at an authentication translator, as is an authentication input. The authentication input corresponds to at least one stored record. The stored record is associated at least with the resource. In response to the receiving, a previously stored credential associated with the resource is accessed. The credential is provided to the resource.

25 Claims, 8 Drawing Sheets



(56) **References Cited**

U.S. PATENT DOCUMENTS

7,950,051	B1	5/2011	Spitz	
8,145,916	B2	3/2012	Boshra	
8,549,300	B1 *	10/2013	Kumar	H04L 9/3247 713/153
8,577,813	B2	11/2013	Weiss	
8,856,539	B2	10/2014	Weiss	
8,984,596	B2	3/2015	Griffin	
9,100,826	B2	8/2015	Weiss	
2004/0107170	A1	6/2004	Labrou	
2004/0236632	A1	11/2004	Maritzen	
2005/0198348	A1 *	9/2005	Yeates	H04L 12/6418 709/232
2009/0100269	A1	4/2009	Naccache	
2010/0242102	A1	9/2010	Cross	
2011/0078771	A1	3/2011	Griffin	
2011/0205016	A1	8/2011	Al-Azem	
2011/0231651	A1	9/2011	Bollay	
2012/0110341	A1 *	5/2012	Beigi	G06Q 20/3223 713/186
2012/0167193	A1	6/2012	Gargaro	

OTHER PUBLICATIONS

Hammer-Lahav, Ed. "The OAuth 1.0 Protocol", from https://tools.ieff.org/html/rfc5849, Apr. 2010.

* cited by examiner

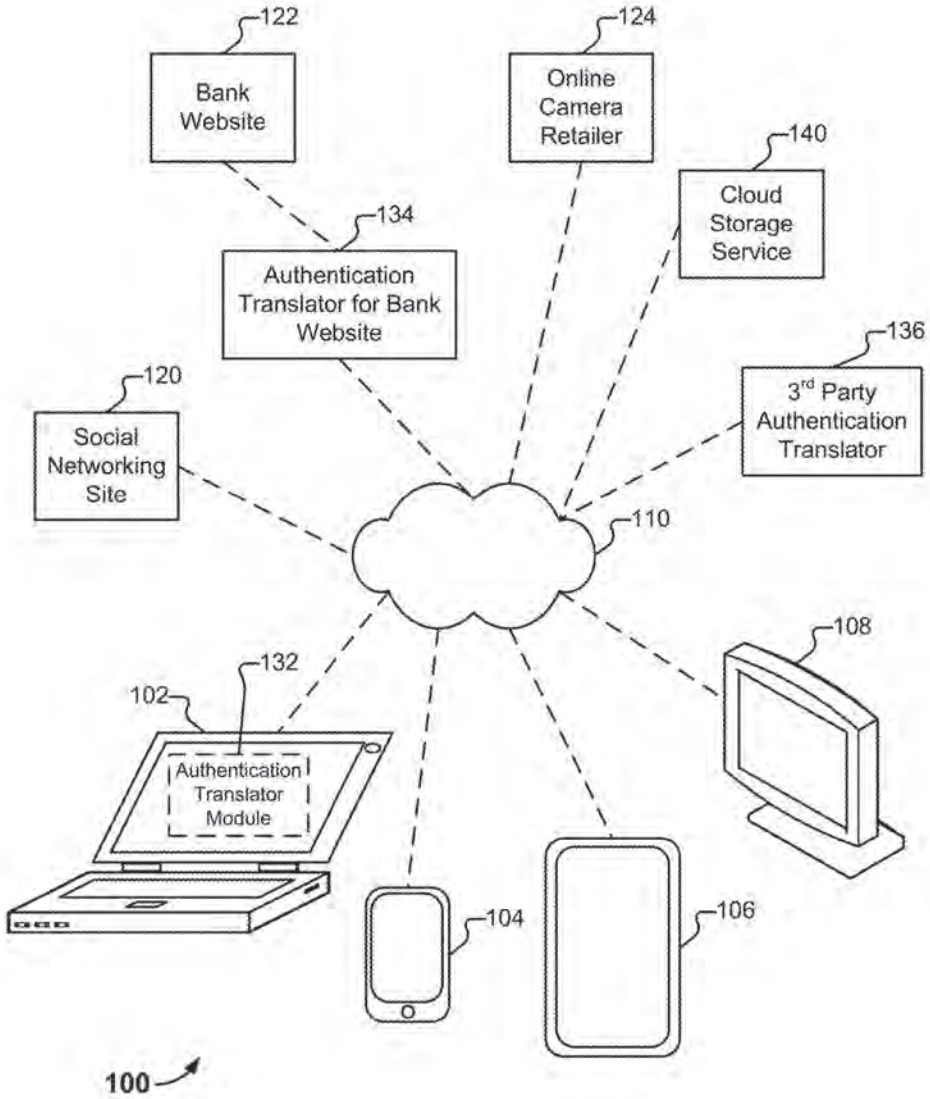

122
Bank Website
124
Online Camera Retailer
140
Cloud Storage Service
134
Authentication Translator for Bank Website
136
3rd Party Authentication Translator
120
Social Networking Site
110
108
132
102
Authentication Translator Module
104
106
100

FIG. 1

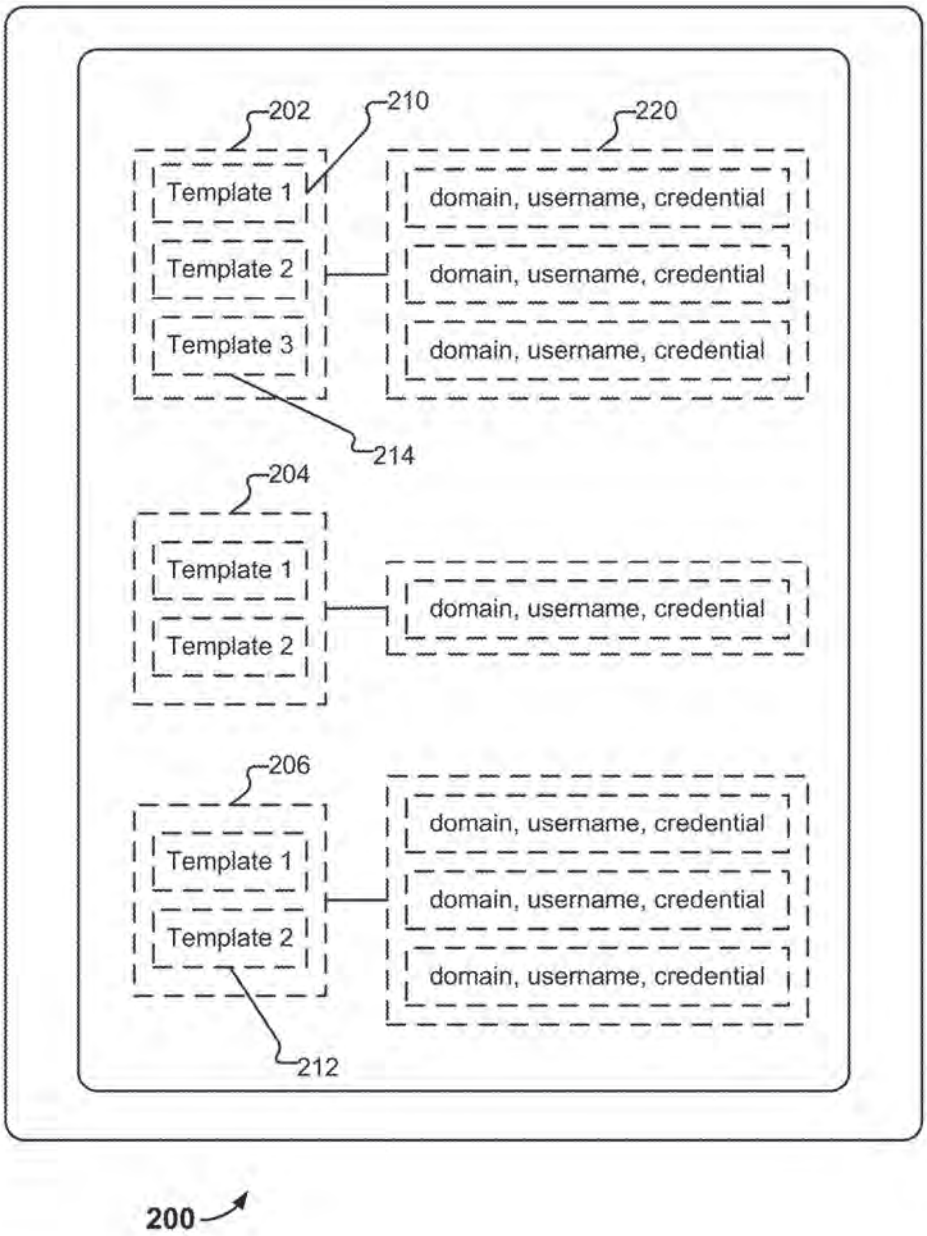
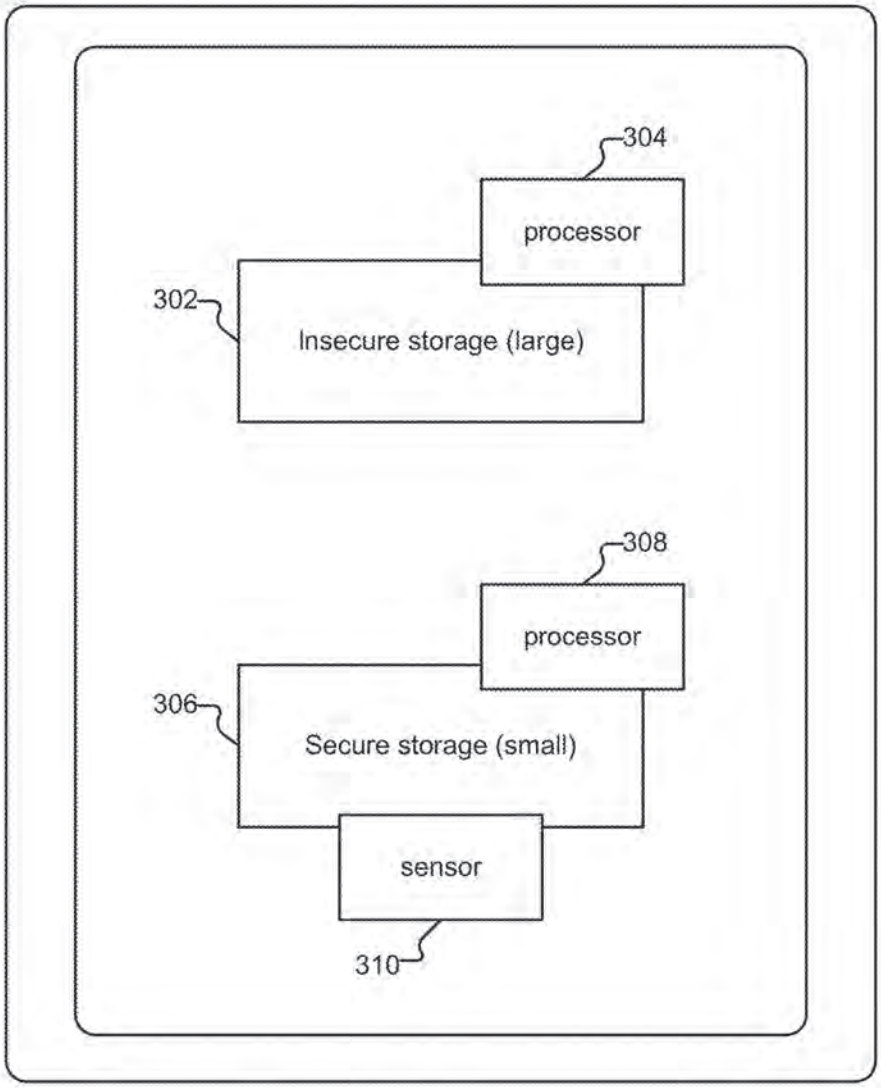

202
210
220
Template 1
Template 2
Template 3
domain, username, credential
domain, username, credential
domain, username, credential
214
204
Template 1
Template 2
domain, username, credential
206
Template 1
Template 2
domain, username, credential
domain, username, credential
domain, username, credential
212
200

FIG. 2


304
processor
302
Insecure storage (large)
308
processor
306
Secure storage (small)
sensor
310

300

FIG. 3

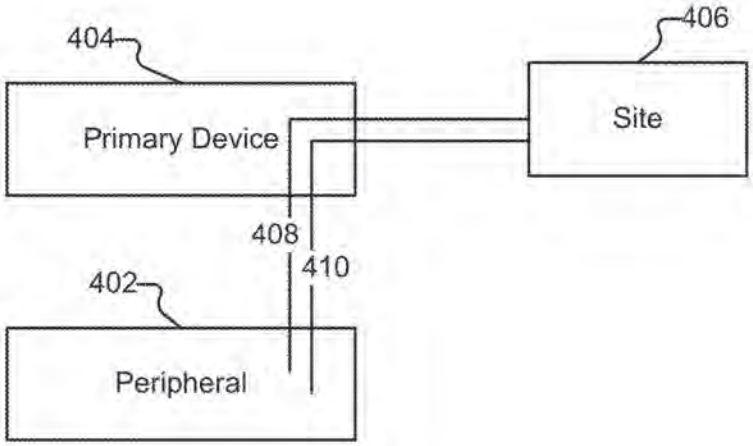


FIG. 4

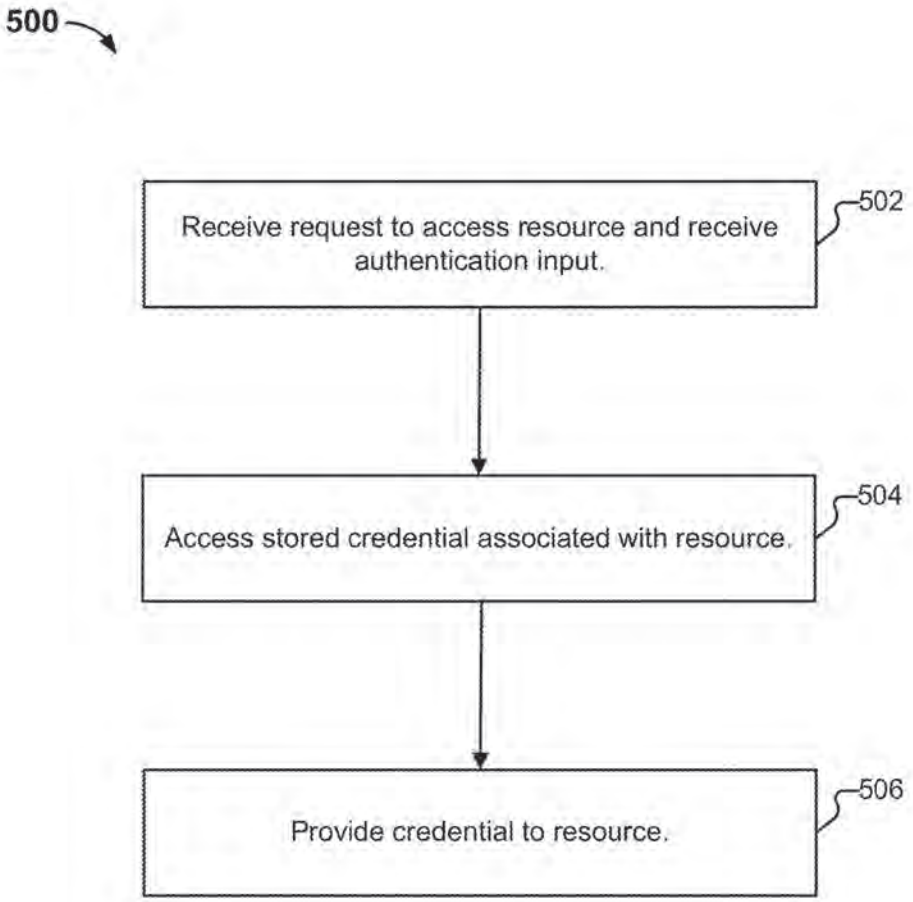

500
Receive request to access resource and receive authentication input.
502
Access stored credential associated with resource.
504
Provide credential to resource.
506

FIG. 5

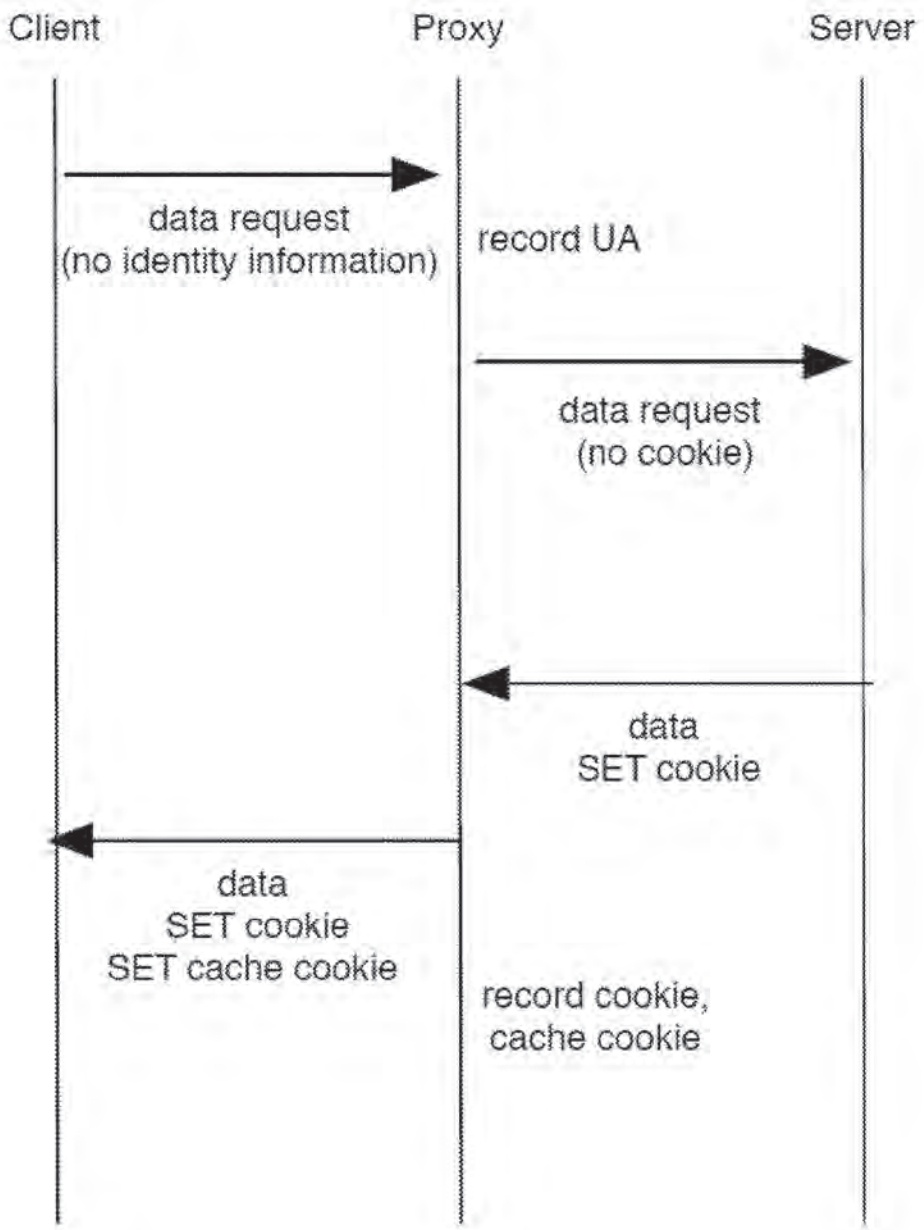

Client
Proxy
Server
data request
(no identity information)
record UA
data request
(no cookie)
data
SET cookie
data
SET cookie
SET cache cookie
record cookie,
cache cookie

FIG. 6

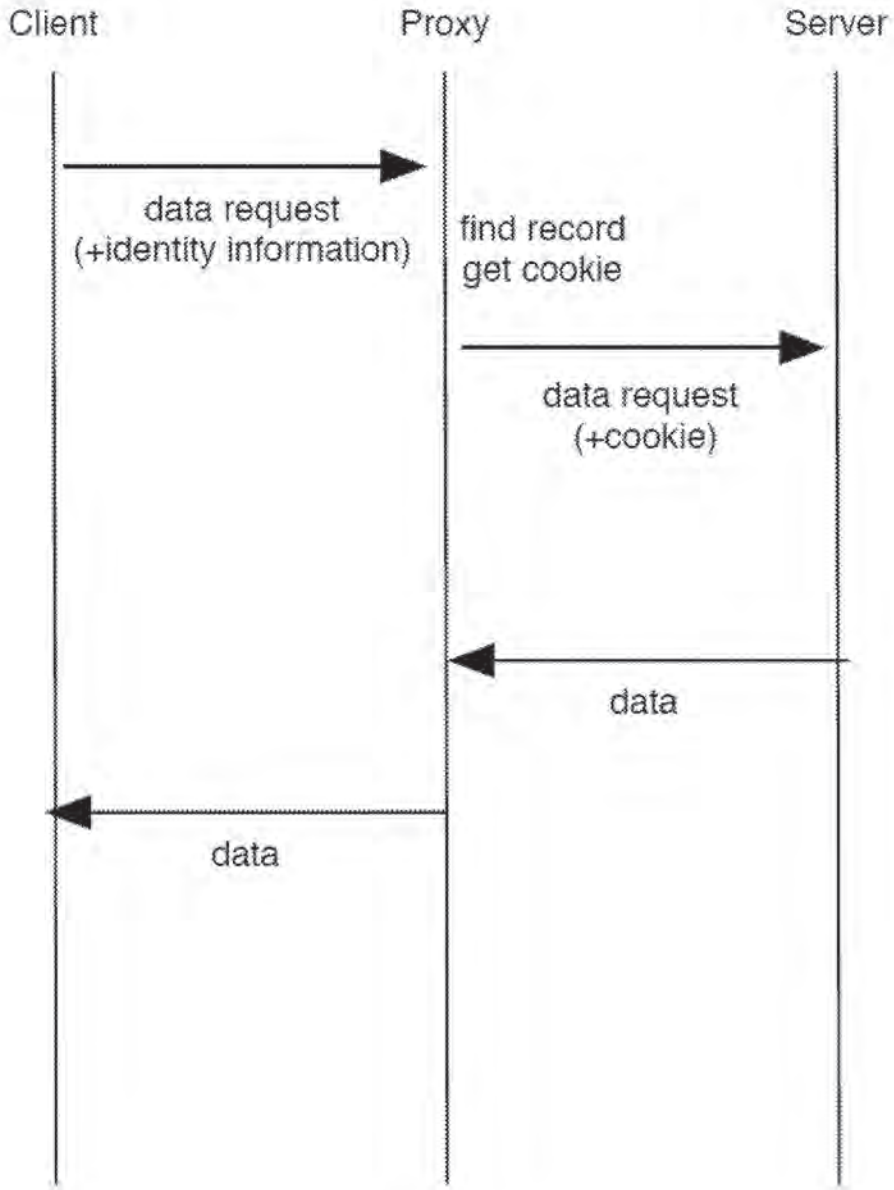

Client
Proxy
Server
data request
(+identity information)
find record
get cookie
data request
(+cookie)
data
data

FIG. 7

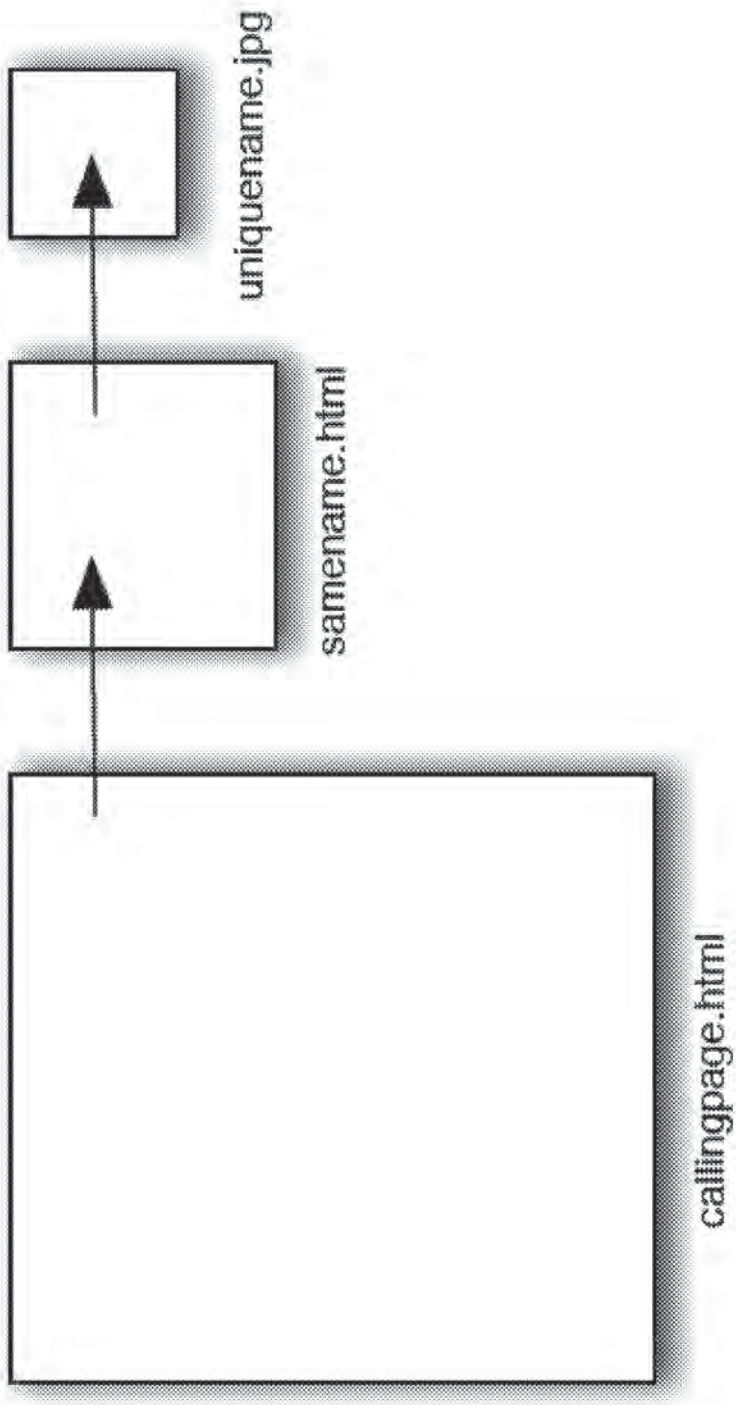


FIG. 8

AUTHENTICATION TRANSLATION

CROSS REFERENCE TO OTHER APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 15/042,636, entitled AUTHENTICATION TRANSLATION filed Feb. 12, 2016 which is incorporated herein by reference for all purposes, which is a continuation of U.S. patent application Ser. No. 13/706,254, entitled AUTHENTICATION TRANSLATION filed Dec. 5, 2012, now U.S. Pat. No. 9,294,452, which is incorporated herein by reference for all purposes. U.S. patent application Ser. No. 13/706,254 claims priority to U.S. Provisional Patent Application No. 61/569,112 entitled BACKWARDS COMPATIBLE ROBUST COOKIES filed Dec. 9, 2011, and also claims priority to U.S. Provisional Patent Application No. 61/587,387 entitled BIOMETRICS-SUPPORTED SECURE AUTHENTICATION SYSTEM filed Jan. 17, 2012, both of which are incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

Providing credentials to a service, whether via a mobile or other device, is often a tedious experience for a user. Unfortunately, to make authentication easier for themselves, users will often engage in practices such as password re-use, and/or the selection of poor quality passwords, which render their credentials less secure against attacks. Accordingly, improvements in authentication techniques would be desirable. Further, it would be desirable for such improvements to be widely deployable, including on existing/legacy systems.

BRIEF DESCRIPTION OF THE DRAWINGS

Various embodiments of the invention are disclosed in the following detailed description and the accompanying drawings.

FIG. **1** illustrates an embodiment of an environment in which authentication translation is provided.

FIG. **2** illustrates an embodiment of credential information stored on a device.

FIG. **3** illustrates an embodiment of a device with secure storage.

FIG. **4** illustrates an example of a renegotiation.

FIG. **5** illustrates an embodiment of a process for performing authentication translation.

FIG. **6** illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator.

FIG. **7** illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator.

FIG. **8** shows the structure of an example of a cache cookie used in some embodiments.

DETAILED DESCRIPTION

The invention can be implemented in numerous ways, including as a process; an apparatus; a system; a composition of matter; a computer program product embodied on a computer readable storage medium; and/or a processor, such as a processor configured to execute instructions stored on and/or provided by a memory coupled to the processor. In this specification, these implementations, or any other form

that the invention may take, may be referred to as techniques. In general, the order of the steps of disclosed processes may be altered within the scope of the invention. Unless stated otherwise, a component such as a processor or a memory described as being configured to perform a task may be implemented as a general component that is temporarily configured to perform the task at a given time or a specific component that is manufactured to perform the task. As used herein, the term 'processor' refers to one or more devices, circuits, and/or processing cores configured to process data, such as computer program instructions.

A detailed description of one or more embodiments of the invention is provided below along with accompanying figures that illustrate the principles of the invention. The invention is described in connection with such embodiments, but the invention is not limited to any embodiment. The scope of the invention is limited only by the claims and the invention encompasses numerous alternatives, modifications and equivalents. Numerous specific details are set forth in the following description in order to provide a thorough understanding of the invention. These details are provided for the purpose of example and the invention may be practiced according to the claims without some or all of these specific details. For the purpose of clarity, technical material that is known in the technical fields related to the invention has not been described in detail so that the invention is not unnecessarily obscured.

FIG. **1** illustrates an embodiment of an environment in which authentication translation is provided. In the example shown, a variety of client devices **102-108** connect, via one or more networks (represented as a single network cloud **110**) to a variety of services **120-124** (also referred to herein as sites **120-124**). In particular, client device **102** is a notebook computer owned by a user hereinafter referred to as Alice. Notebook **102** includes a camera, a microphone, and a fingerprint sensor. Client device **104** is a smartphone, also owned by Alice. Client device **104** includes a camera. Client device **106** is a tablet owned by Bob, and sometimes used by Bob's son Charlie. Client device **106** includes a camera and a fingerprint sensor. Client device **108** is a kiosk located in the lobby of a hotel. Kiosk **108** includes a camera and a microphone. The techniques described herein can be used with or adapted to be used with other devices, as applicable. For example, the techniques can be used in conjunction with gaming systems, with peripheral devices such as mice, and with embedded devices, such as door locks.

Service **120** is a social networking site. Service **122** is a website of a bank. Service **124** is the online store of a boutique camera retailer. Each of services **120-124** requires a username and password (and/or a cookie) from a user prior to giving that user access to protected content and/or other features. As will be described in more detail below, using the techniques described herein, users need not type such usernames and passwords into their devices whenever required by a service. Instead, users can authenticate themselves to an "authentication translator" via an appropriate technique, and the authentication translator will provide the appropriate credentials to the implicated service on the user's behalf. Also as will be described in more detail below, authentication translators can be located in a variety of places within an environment. For example, notebook computer **102** includes an authentication translator module **132** that provides authentication translation services. The other devices **104-108** can also include (but need not include) their own respective authentication translator modules. The owner of bank website **122** also operates an authentication translator

134 associated with the bank. Finally, authentication translator 136 provides authentication translation services to a variety of businesses, including online camera retailer 124.

FIG. 2 illustrates an embodiment of credential information stored on a device. In particular, device 200 stores three user profiles 202-206, each of which contains a username and one or more templates (e.g., template 210) associated with the user. In various embodiments, a template is a collection of biometric features. Using fingerprints as an example type of biometric, a corresponding template includes a collection of patterns, minutia, and/or other features that can be matched against to determine if a person's fingerprint matches the fingerprint of the registered user (i.e., the owner of a given user profile). A representation of a single fingerprint may be included in multiple templates (e.g., in different resolutions, in accordance with different protocols, as captured during warm vs. cold conditions, and/or by itself or in combination with multiple fingerprints). When other biometrics are employed (e.g., facial recognition, voiceprint, or retina scan technology), features appropriate to those types of biometrics are included in the template. Other types of features can also be included in templates. As one example, a user's typing speed and/or accuracy can be measured by a device, such as device 102, and used to distinguish between multiple users of a device. For example, suppose Alice types at 100 words per minute and rarely makes mistakes. A representation of this information can be stored in template 212. Also suppose Alice's niece, who sometimes uses Alice's laptop computer when visiting Alice types at 20 words per minute and makes many mistakes. In some embodiments, the fact that a user was recently (e.g., within the last 5 minutes) typing on laptop 102 at 90 words per minute is evidence of a match against template 212. In this case, the typing speed of 90 words per minute is similar enough to Alice's typical behavior, it is considered a match. Various policies can be included in a profile that govern how matches are to be performed. For example, policies can specify thresholds/tolerances for what constitutes a match, and can specify that different levels of matches can result in different levels of access to different resources.

A profile is associated with a vault (e.g., vault 220). The vault, in turn, contains triples specifying a service provider/domain, a username, and a credential. The vault can also contain other sensitive user information, such as account numbers, address/phone number information, and health care data. The credential for a service provider/domain can be a password (e.g., for legacy servers), and can also take alternate forms (e.g., a cryptographic key for service providers supporting stronger authentication methods).

In some embodiments, profiles, templates, and vaults (collectively "authentication information") are stored entirely in an unprotected storage area, and are stored in the clear. In other embodiments, secure storage techniques are used to secure at least a portion of the authentication information.

One example of a device with secure storage is illustrated in FIG. 3. In the example shown, a mobile phone 300 includes a large and insecure storage 302 attached to a fast processor 304, and a smaller but secure storage 306 attached to a dedicated processor 308 and a sensor 310 (e.g., a camera or a fingerprint reader). Users (and applications) can read from and write to the insecure storage area. However, users cannot access the secure storage area, and the fast processor can only communicate with the dedicated processor/sensor via a restricted API. As another example, a unique decryption key associated with a given vault can be stored in a profile. The vault is an encrypted and authenticated container that can be stored on insecure storage, e.g., on the device, and also backed up (e.g., to a cloud storage service 140 or to an alternate form of external storage). As needed, authentication information or portions thereof can be loaded into secure storage and decrypted. For example, one can use AES to encrypt the files one by one, using a key stored on the secured storage. A message authentication technique, such as HMAC, can be used for authenticating the encrypted files to provide tamper prevention. Profiles and vaults can be updated while in secure storage; if this occurs, they are encrypted and MACed before being written back to the insecure storage, which may in turn propagate them to external backup storage. In yet other embodiments, profiles and vaults are stored entirely in secure storage, in plaintext, which allows them to be both read and written—and in particular, searched.

Example Transaction Types

A variety of transaction types can take place in the environment shown in FIG. 1, examples of which are discussed in this section.

Initial Registration

In order to begin using the techniques described herein, users perform some form of initial registration. As one example, suppose Alice launches an enrollment program installed on laptop 102. She uses the program to capture various biometric information (e.g., fingerprints, photographs of her face, etc.). A user profile is created for Alice, and the biometric information captured about her is encoded into a plurality of templates, such as templates 210 and 214. In some embodiments, Alice is also explicitly asked to supply credential information for services she would like to use, such as by providing the domain name of social networking site 120, along with her username and password for site 120. In other embodiments, domain/username/credential information is at least passively captured on Alice's behalf and included in one or more vaults such as vault 220. Credential information can also be important from a browser password manager already in use by Alice or other appropriate source. In some embodiments, Alice also registers with cloud storage service 140, which will allow her to back up her authentication information and to synchronize it across her devices (e.g., 102 and 104), as described in more detail below.

Other registration approaches can also be used. For example, registration can be integrated into the experience the first time a device is used. Thus, when Bob first turns on tablet 106, he may be prompted to take a picture of his face (with a profile/templates being created in response). Similarly, the first time Charlie uses tablet 106, the techniques described herein can be used to determine that Charlie does not yet have a profile (e.g., because none of the templates already present on tablet 106 match his biometrics) and Charlie can be prompted to enroll as a second user of the device.

Authentication

Suppose Alice wishes to authenticate to banking website 122. Using a fingerprint reader incorporated into her laptop, she performs a fingerprint scan, which causes her biometric features to be extracted and compared to any stored templates residing on her computer. If a match is found, an associated decryption key is selected, and the associated vault is loaded and decrypted. The vault is scanned for an entry that matches the selected service provider (i.e., website 122). If a matching entry is found, the associated domain, username, and site credential are extracted from the vault. In some embodiments, the validity of the domain name map-

ping is verified at this point to harden the system against domain name poisoning. Next, a secure connection is established between Alice's computer and the service provider, and Alice is authenticated. For service providers supporting strong user authentication, mutual SSL can be used, for example. A variety of policies can be involved when performing matching. For example, to access certain domains, Alice's print may need only match template **210**. To access other domains, Alice may need to match multiple templates (e.g., both **210** and **214**). As another example, in order to access social networking site **120**, Alice may merely need to be sitting in front of her computer, which has an integrated webcam. Even in relatively low light conditions, a match can be performed against Alice's face and features stored in a template. However, in order to access bank website **122**, Alice may need a high quality photograph (i.e., requiring her to turn on a bright light) and may need to demonstrate liveness (e.g., by blinking or turning her head). As yet another example, other contextual information can be included in policies. For example, if Alice's IP address indicates she is in a country that she is not usually in, she may be required to match multiple templates (or match a template with more/better quality features) in order to access retailer **124**, as distinguished from when her IP address indicates she is at home.

In some embodiments, the biometric sensor used by a user may be a peripheral device (e.g., a mouse with an integrated fingerprint scanner that is connected to the user's primary device via USB). In such scenarios, the peripheral device may be responsible for storing at least a portion of authentication information and may perform at least some of the authentication tasks previously described as having been performed by Alice's computer. For example, instead of processors **304** and **308**, and storages **302** and **306** being collocated on a single device (e.g., laptop **102**), processor **304** and storage **302** may be present on a primary device, and processor **308** and storage **306** may be present on a peripheral device (e.g., that also includes a sensor, such as a fingerprint reader).

In such scenarios, once Alice's login to banking website **122** is successfully completed, the secure session can be handed over from the peripheral device to the primary device, in a way that does not allow the primary device retroactive access to the plaintext data of the transcripts exchanged between the peripheral device and the service provider. One way this can be accomplished is by renegotiating SSL keys between the peripheral device and the website, after which the newly negotiated key can be handed off from the peripheral device to the primary device. This avoids retroactive credential capture in a setting where the device is infected by malware.

An example of renegotiation is depicted in FIG. **4**. Specifically, after a user has successfully authenticated to a fingerprint reader, a login is performed to a service provider. Using the primary device (**404**) as a proxy, the peripheral fingerprint reader **402** negotiates a first SSL connection (**408**) with a service provider **406**, over which credentials are exchanged. The proxy then renegotiates SSL (**410**), which replaces the old key with a new one. The new key is disclosed to the device, which then seamlessly takes over the connection with the service provider and performs the transaction protected by the authentication. The credentials exchanged during the first SSL connection cannot be accessed by device **404**, since the key of the renegotiated session is independent of the key of the original session; this provides protection against malware residing on the device. Renegotiation can be used when the primary device **404** is believed to be in a safe state when performing the negotiation of the SSL connection, but it is not known whether it is in a safe state during the transaction protected by the authentication. Renegotiation can also be used when a secure component of the primary device **404** performs the negotiation of the SSL connection and another and potentially insecure component of the primary device **404** is involved in the transaction protected by the authentication.

FIG. **5** illustrates an embodiment of a process for performing authentication translation. The process begins at **502** when a request to access a resource is received, as is an authentication input. One example of the processing performed at **502** is as follows. Suppose Alice wishes to sign into social networking website **120**. She directs a web browser application installed on client **102** to the social networking website. Authentication translator module **132** recognizes, from the context of Alice's actions (e.g., that she is attempting to access site **120** with her browser) that she would like to access a particular resource. Authentication translator module **132** prompts Alice (e.g., by a popup message or via a sound) to provide biometric information (e.g., to use the integrated fingerprint reader on her laptop). In some embodiments, the translator module does not prompt Alice, for example, because Alice has been trained to provide biometric information automatically when attempting to access certain resources. In yet other embodiments, the translator module only prompts Alice if she fails to provide acceptable biometric information within a timeout period (e.g., 30 seconds).

Module **132** compares Alice's supplied biometric data to the templates stored on her computer. If a suitable match is found, and if an entry for site **120** is present in the applicable vault, at **504**, a previously stored credential associated with the resource is accessed. In particular, the username and password for the website, as stored in a vault, such as vault **220**, are retrieved from the vault.

Finally, at **506**, the credential is provided to the resource. For example, Alice's username and password for site **120** are transmitted to site **120** at **506**. The credential can be transmitted directly (e.g., by the module or by Alice's computer) and can also be supplied indirectly (e.g., through the use of one or more proxies, routers, or other intermediaries, as applicable).

Other devices can also make use of process **500** or portions thereof. For example, when Alice launches a banking application on phone **104**, implicit in her opening that application is her desire to access the resources of website **134**. The application can take Alice's picture and compare it to stored templates/vault information. If an appropriate match is found, a credential can be retrieved from the vault on her phone (or, e.g., retrieved from cloud storage service **140**) and provided to website **134**.

As another example, suppose Charlie is using tablet **106** and attempts to visit site **120**, whether via a dedicated application or via a web browser application installed on the tablet. Charlie's photograph is taken, and then compared against the profiles stored on tablet **106** (e.g., both Bob and Charlie's profiles). When a determination is made that Charlie's photograph matches a template stored in his stored profile (and not, e.g., Bob's), Charlie's credentials for site **120** are retrieved from a vault and transmitted by an authentication translator module residing on client **106**.

As yet another example, kiosk **108** can be configured to provide certain local resources (e.g., by displaying a company directory or floor plan on demand) when users speak certain requests into a microphone. Enrolled users (e.g., with stored voiceprint or facial recognition features) can be

granted access to additional/otherwise restricted services in accordance with the techniques described herein and process **500**.

New Device

In some embodiments, to register a new device, a user provides an identifier, such as a username or an account number to the device. The new device connects to an external storage (such as cloud storage **140**), provides the user identifier and credential, and downloads the user's templates/vaults from the service. In some embodiments, the templates/vaults are encrypted. Once downloaded, the template is decrypted and stored in a secure storage area, while the still encrypted vault can be stored in insecure storage. The decryption key can be generated from information the user has/knows, or from biometric data—such as features extracted from fingerprinting of all ten fingers. In some embodiments, more arduous fingerprinting is required for the setup of a new device than for regular authentication to avoid that a new device gets registered by a user thinking she is merely authenticating—or worse still, simply touching the device. Moreover, it translates into higher entropy of the decryption keys.

Backup Authentication

Backup authentication allows a user, such as Alice, to access resources in the event she is unable to or unwilling to interact with a particular biometric sensor. As one example, instead of having a single template associated with her profile, Alice can have multiple templates associated with it, e.g., where the first template includes fingerprint features and the second template includes voice biometric, facial recognition, or iris detection features. As a second example, where the service Alice is connecting to is a legacy website (i.e., one that users authenticate to using usernames and passwords), such a service would allow the use of passwords and password reset mechanisms by Alice without requiring Alice to use a fingerprint reader.

In various embodiments, environment **100** supports the ability of users (e.g., under duress) to release the contents of their vaults. For example, if Alice was physically threatened with the loss of a finger by a criminal, Alice could instead release the contents of her vault(s)—the ultimate goal of the criminal. As one example, in the event Alice supplies all 10 fingerprints to the sensor, provides a special password, or supplies a fingerprint and a second identifier, a cleartext version of her vault(s) could be made available.

Access Policies

In various embodiments, cloud storage service **140** is configured to accept backups from multiple devices associated with a single account, and synchronize the updates so that all devices get automatically refreshed. For example, Alice's laptop **102** and phone **104** could both communicate with cloud storage service **140** which would keep their authentication information synchronized. Refreshes can also be made in accordance with user-configured restrictions. For example, Alice's employer could prevent privileged employer data from being stored on shared personal devices, or on any device that was not issued by the employer. As another example, arbitrary policies can be defined regarding the access to and synchronization of software and data, and to tie a license or access rights to a person (and her fingerprint) rather than to a device. As yet another example, in some embodiments (e.g., where a device is made publicly available or otherwise shared by many users), no or a reduced amount of authentication information resides on a device, and at least a portion of authentication information is always retrieved from cloud storage service **140**.

Remote Wiping

Remote wiping of a user's authentication information (e.g., templates) can be used both to "unshare" previously shared devices (e.g., where Bob and Charlie both have user profiles on their shared tablet **106**), and to avoid that criminals with physical component access to lost devices gain access to templates and vault contents. In some embodiments, polices such as ones where a template self-wipes if it is not matched within a particular duration of time are supported. Since user data can be frequently backed up to the cloud storage, and recovered from this using the new device registration process, inconvenience to the user will be minimized.

Legacy Server Support

New authentication schemes typically require changes to a significant codebase residing with service providers. If the code is well written and documented, such changes may be relatively simple. Commonly, though, this may not be so. The engineers who originally wrote the code of relevance may have long since left the company; the code they left behind may be poorly documented—if documented at all. In severe cases, the legacy code may have been written in an outdated programming language or written in a way that does not follow guidelines for good code. This makes updates to the codebase impractical or virtually impossible in many common cases. Even if none of these challenges complicate the desired modifications, it is commonly a great bureaucratic burden to obtain permission to make modifications (e.g., to store an extra field in a backend database), because every affected part of the organization may need to review the request.

As will be described in the following section, the technologies described herein can be used in conjunction with legacy servers (e.g., existing servers that rely on usernames and passwords to authenticate users), and in particular, can be used without requiring modification to such legacy servers.

Cookies

Cookies are commonly used by legacy servers for user authentication. Unfortunately, cookies have several problems. For one thing, they are sometimes deleted—whether explicitly/intentionally by the end user or by the user's software. In addition, cookies are commonly stolen. Approaches such as cache cookies and identification using user agents can be more resistant to these problems, however, they have their own problems. For example, their use requires new code and new fields in the credential database stored by the server.

In some embodiments, authentication translators, such as translators **134** and **136** (also referred to herein as proxies) provide authentication translation services on behalf of associated services. Translators **134** and **136** are illustrated as single logical devices in FIG. **1**. In some embodiments, the translators comprise standard commercially available server hardware (e.g., a multi-core processor, 4+ Gigabytes of RAM, and one or more Gigabit network interface adapters) and run typical server-class operating systems (e.g., Linux). Translators **134** and **136** can also be implemented using a scalable, elastic architecture and may comprise several distributed components, including components provided by one or more third parties. For example, translators **134** and **136** may store user credential information or may task cloud storage service **140** with storing at least a portion of that information.

In the case of authentication translator **134**, service is provided with respect to bank website **122** only. Authentication translator **134** is positioned between a legacy web

server (**122**) and the Internet (**110**)—and therefore between the legacy server and any client devices. Authentication translator **134** is configured to translate traffic between the legacy server and client devices so that the client devices (and respective users) perceive the new authentication mechanism, while the legacy server remains unchanged. Authentication translator **136** works similarly, but it provides authentication translation as a third party service to multiple providers, an example of which is online camera retailer **124**.

Authentication translators **134** and **136** can also perform process **500**. For example, when a device transmits a request to access website **122**, the request is intercepted by translator **134**, as is cookie/user agent information. The received information can be used to determine a username/password associated with the device, and that information can be passed by translator **134** to website **122** on behalf of the device.

FIG. **6** illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator. The translator (referred to in the figure as a "proxy") fails to identify the client, and passes on the request to the legacy server. The legacy server responds to the request and sets a cookie. The proxy passes on the response, including the cookie and also a cache cookie. The proxy stores the information about both these types of identifiers, along with the user agent of the client device. This triplet of information is also referred to herein as an identifier.

FIG. **7** illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator. In this scenario, the device's request is accompanied by some form of identifying information. The proxy uses this information to identify the associated cookie and passes this along to the legacy server, along with the request. In some embodiments, additional processing is also involved. For example, reading a cache cookie may require user interaction. Moreover, if not all of the identifying information is present in the request, the proxy can be configured to set the missing information again by sending a corresponding request to the client device.

The translation of cache cookies and user agent information to cookies involves a two-way translation. First, when the legacy server sets a cookie, the proxy will set the two types of cookies—both an HTML cookie and a cache cookie—and then create a new record in which the two cookies are stored, along with the user agent information of the client device. The user agent information can include quite a bit of data associated with a browser—such as the browser type and version, the clock skew, and the fonts that are installed. While each of these pieces of information only contributes a small amount of entropy, the collection of items can be sufficient to identify the device in most cases. Moreover, while some of these types of data may change over time—in fact, all of them may—they do not typically change, and when one or two of them do, the others typically do not. When the client device is used to visit a site controlled by the legacy server, the cookie, cache cookie and user agent information are read (if available), the record identified, and the request translated and sent to the legacy server. When a legacy server requests that the user password is updated (e.g., as part of an annual or other periodic requirement), the transmission of this request to the user can be suppressed—in which case the database of the proxy is updated to create the illusion of an update. The user can be involved in authentication as needed, e.g., if, in addition to supplying a credential, a user must also solve a CAPTCHA, the CAPTCHA can be displayed to the user (with the user's credentials being handled in accordance with the techniques described herein).

FIG. **8** shows the structure of an example of a cache cookie used in some embodiments. Cache cookies can be associated with a particular webpage, just like an HTML cookie can. In the example shown in FIG. **8**, the proxy wishes to associate a page "calling page.html" with a cache cookie. It embeds a request for a second object, "samename.html" in callingpage.html for every visitor. However, as the cache cookie is set for one visitor, a customized samename.html is served to this visitor. The page samename.html refers to an object with a different name for each user; that object is referred to as "uniquename.jpg." The cache cookie is set by embedding the request for samename.html in callingpage.html. The client browser attempts to render this, causing a request for samename.html from the server. The server configures samename.html to refer to a uniquely named file uniquename.jpg, and serves samename.html to the client. For the client browser to render samename.html, it requests the file uniquename.jpg, which is intentionally not served. That concludes the setting of a cache cookie. As a user returns to the page callingname.html, the browser again attempts to render the entire page, which causes it to load the object samename.html from its cache. As that is rendered, the client browser requests uniquename.jpg, which is not in its cache (since it was not served previously). The server still does not serve it, but takes note of the name of the file being requested, as it identifies the client device. Note that samename.html can be displayed in a zero-sized iframe, which makes the end user unaware of it being rendered.

A cache cookie is an implementation of the typical cookie functionality that uses the client device's browser cache. Unlike user agents, it does not change over time, and like standard HTML cookies, it cannot be read by a party other than that which set it. However, like HTML cookies, it could be deleted—by the user clearing his or her browser cache. Cache cookies are not automatically transmitted with GET requests, unless the cache elements are embedded in the referring pages. This adds a potential round of communication in some settings. By relying on user agent information, cache cookies, and HTML cookies to identify the client device, it is much more likely that a machine will be recognized than if only HTML cookies are used.

Although the foregoing embodiments have been described in some detail for purposes of clarity of understanding, the invention is not limited to the details provided. There are many alternative ways of implementing the invention. The disclosed embodiments are illustrative and not restrictive.

What is claimed is:

1. A system, comprising:

one or more processors configured to:

receive, at a device, a request from a user to access a resource external to the device;

access a record stored on the device, the record comprising credential information associated with the external resource to which the user has requested access;

retrieve at least a portion of the credential information from the record;

establish a connection between the device and the resource external to the device to which the user requested access;

facilitate a login of the user to the external resource at least in part by transmitting, on behalf of the user,

from the device and via the established connection, output based at least in part on the at least portion of the credential information retrieved from the record, wherein the user of the device is logged into the external resource based at least in part on the output transmitted from the device on behalf of the user; and

perform a backup of the record to a remote storage entity, wherein the remote storage entity is configured to synchronize records between at least two devices associated with the user; and

a memory coupled to the one or more processors and configured to provide the one or more processors with instructions.

2. The system recited in claim **1**, wherein the one or more processors are further configured to:

receive an authentication input from the user;

receive at least one of: (1) a template stored on the device, and (2) data stored on the device associated with a password, the template comprising at least one of: (1) fingerprint features, (2) voice biometric features, (3) facial recognition features, (4) iris detection features, and (5) retina features;

compare the authentication input with the at least one of: (1) the template stored on the device, and (2) the data stored on the device associated with the password; and

perform an action based at least in part on the comparison.

3. The system recited in claim **2**, wherein the action comprises at least one of: (1) accessing a vault on the device, (2) accessing a vault on a remote storage entity, (3) performing decryption, (4) establishing a secure connection, (5) accessing the record comprising the credential information, (6) transmitting the output based at least in part on the at least portion of the credential information retrieved from the record, (7) accessing a username, (8) accessing a domain, (9) renegotiating the secure connection, (10) facilitating the login of the user to the external resource, (11) causing information on the device to be wiped, and (12) releasing of data store contents.

4. The system recited in claim **2**, wherein the action comprises accessing a domain, and wherein a first domain and a second domain are associated with different authentication policies.

5. The system recited in claim **2**, wherein the one or more processors are further configured to prompt the user to provide the authentication input.

6. The system recited in claim **2**, wherein the one or more processors are further configured to determine a liveness of the user based at least in part on the authentication input.

7. The system recited in claim **2**, wherein the record is accessed based at least in part on the authentication input.

8. The system recited in claim **1**, wherein authentication information on the device is caused to be wiped.

9. The system recited in claim **1**, wherein at least a portion of information on the device is stored in a secure storage.

10. The system recited in claim **1** wherein the credential information comprises at least one of a password, a cookie, and a cryptographic key.

11. The system recited in claim **1** wherein the record further comprises at least one of a username, an account number, an address, phone number information, and health care data.

12. The system recited in claim **1** wherein the user is associated with at least two of a first authentication template, a second authentication template, and a password.

13. A method, comprising:

receiving, at a device, a request from a user to access a resource external to the device;

accessing a record stored on the device, the record comprising credential information associated with the external resource to which the user has requested access;

retrieving at least a portion of the credential information from the record;

establishing a connection between the device and the resource external to the device to which the user requested access;

facilitating a login of the user to the external resource at least in part by transmitting, on behalf of the user, from the device and via the established connection, output based at least in part on the at least portion of the credential information retrieved from the record, wherein the user of the device is logged into the external resource based at least in part on the output transmitted from the device on behalf of the user; and

performing a backup of the record to a remote storage entity, wherein the remote storage entity is configured to synchronize records between at least two devices associated with the user.

14. The method of claim **13**, further comprising:

receiving an authentication input from the user;

receiving at least one of: (1) a template stored on the device, and (2) data stored on the device associated with a password, the template comprising at least one of: (1) fingerprint features, (2) voice biometric features, (3) facial recognition features, (4) iris detection features, and (5) retina features;

comparing the authentication input with the at least one of: (1) the template stored on the device, and (2) the data stored on the device associated with the password; and

performing an action based at least in part on the comparison.

15. The method of claim **14**, wherein the action comprises at least one of: (1) accessing a vault on the device, (2) accessing a vault on a remote storage entity, (3) performing decryption, (4) establishing a secure connection, (5) accessing the record comprising the credential information, (6) transmitting the output based at least in part on the at least portion of the credential information retrieved from the record, (7) accessing a username, (8) accessing a domain, (9) renegotiating the secure connection, (10) facilitating the login of the user to the external resource, (11) causing information on the device to be wiped, and (12) releasing of data store contents.

16. The method of claim **14**, wherein the action comprises accessing a domain, and wherein a first domain and a second domain are associated with different authentication policies.

17. The method of claim **14**, further comprising prompting the user to provide the authentication input.

18. The method of claim **14**, further comprising determining a liveness of the user based at least in part on the authentication input.

19. The method of claim **14**, wherein the record is accessed based at least in part on the authentication input.

20. The method of claim **13**, wherein authentication information on the device is caused to be wiped.

21. The method of claim **13**, wherein at least a portion of information on the device is stored in a secure storage.

22. The method of claim **13** wherein the credential information comprises at least one of a password, a cookie, and a cryptographic key.

23. The method of claim **13** wherein the record further comprises at least one of a username, an account number, an address, phone number information, and health care data.

24. The method of claim **13** wherein the user is associated with at least two of a first authentication template, a second authentication template, and a password.

25. A computer program product embodied in a non-transitory computer readable storage medium and comprising computer instructions for:

receiving, at a device, a request from a user to access a resource external to the device;

accessing a record stored on the device, the record comprising credential information associated with the external resource to which the user has requested access;

retrieving at least a portion of the credential information from the record;

establishing a connection between the device and the resource external to the device to which the user requested access;

facilitating a login of the user to the external resource at least in part by transmitting, on behalf of the user, from the device and via the established connection, output based at least in part on the at least portion of the credential information retrieved from the record, wherein the user of the device is logged into the external resource based at least in part on the output transmitted from the device on behalf of the user; and

performing a backup of the record to a remote storage entity, wherein the remote storage entity is configured to synchronize records between at least two devices associated with the user.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 10,521,568 B1
APPLICATION NO. : 16/273797
DATED : December 31, 2019
INVENTOR(S) : Bjorn Markus Jakobsson

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In the Claims

Column 11, Line 58, Claim 10, after "claim 1", insert --,--.

Column 11, Line 61, Claim 11, after "claim 1", insert --,--.

Column 11, Line 65, Claim 12, after "claim 1", insert --,--.

Column 12, Line 65, Claim 22, after "claim 13", insert --,--.

Column 13, Line 1, Claim 23, after "claim 13", insert --,--.

Column 13, Line 4, Claim 24, after "claim 13", insert --,--.

Signed and Sealed this
Twenty-ninth Day of March, 2022

Drew Hirshfeld
Performing the Functions and Duties of the
Under Secretary of Commerce for Intellectual Property and
Director of the United States Patent and Trademark Office

Electronic Acknowledgement Receipt

EFS ID:	35124101
Application Number:	16273797
International Application Number:	
Confirmation Number:	3223
Title of Invention:	AUTHENTICATION TRANSLATION
First Named Inventor/Applicant Name:	Bjorn Markus Jakobsson
Customer Number:	21912
Filer:	Yeu-Ting George Cheng/Elaine Nguyen
Filer Authorized By:	Yeu-Ting George Cheng
Attorney Docket Number:	MJAKP008C2
Receipt Date:	12-FEB-2019
Filing Date:	
Time Stamp:	16:14:34
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Application Data Sheet	MJAKP008C2_ADS.pdf	1823391 61a909f6f1f2e07c5e1767798477&12aacfcf2	no	8

Warnings:

Information:					
2	Specification	MJAKP008C2_APP.pdf	162871 379193ii5dcdbcee24ea12b5793a5043111 28330b	no	19
Warnings:					
Information:					
3	Drawings-only black and white line drawings	MJAKP008C2_APP_Figures.pdf	112090 83487947bd1fc2e79014268a447a54f43bd ce227	no	8
Warnings:					
Information:					
		Total Files Size (in bytes):	2098352		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.
National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.
New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Attorney Docket No. MJAKP008C2

APPLICATION FOR UNITED STATES PATENT

AUTHENTICATION TRANSLATION

By Inventor:

Bjorn Markus Jakobsson
Portola Valley, CA
A Citizen of Sweden

Assignee: RightQuestion, LLC

VAN PELT, YI & JAMES LLP
10050 N. Foothill Blvd., Suite 200
Cupertino, CA 95014
Telephone (408) 973-2585

AUTHENTICATION TRANSLATION

CROSS REFERENCE TO OTHER APPLICATIONS

[0001] This application is a continuation of co-pending U.S. Patent Application No. 15/042,636, entitled AUTHENTICATION TRANSLATION filed February 12, 2016 which is incorporated herein by reference for all purposes, which is a continuation of U.S. Patent Application No. 13/706,254, entitled AUTHENTICATION TRANSLATION filed December 5, 2012, now Patent No. 9,294,452, which is incorporated herein by reference for all purposes. U.S. Patent Application No. 13/706,254 claims priority to U.S. Provisional Patent Application No. 61/569,112 entitled BACKWARDS COMPATIBLE ROBUST COOKIES filed December 9, 2011, and also claims priority to U.S. Provisional Patent Application No. 61/587,387 entitled BIOMETRICS-SUPPORTED SECURE AUTHENTICATION SYSTEM filed January 17, 2012, both of which are incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

[0002] Providing credentials to a service, whether via a mobile or other device, is often a tedious experience for a user. Unfortunately, to make authentication easier for themselves, users will often engage in practices such as password re-use, and/or the selection of poor quality passwords, which render their credentials less secure against attacks. Accordingly, improvements in authentication techniques would be desirable. Further, it would be desirable for such improvements to be widely deployable, including on existing/legacy systems.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] Various embodiments of the invention are disclosed in the following detailed description and the accompanying drawings.

[0004] Figure 1 illustrates an embodiment of an environment in which authentication translation is provided.

[0005] Figure 2 illustrates an embodiment of credential information stored on a device.

[0006] Figure 3 illustrates an embodiment of a device with secure storage.

[0007] Figure 4 illustrates an example of a renegotiation.

[0008] Figure 5 illustrates an embodiment of a process for performing authentication translation.

[0009] Figure 6 illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator.

[0010] Figure 7 illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator.

[0011] Figure 8 shows the structure of an example of a cache cookie used in some embodiments.

DETAILED DESCRIPTION

[0012] The invention can be implemented in numerous ways, including as a process; an apparatus; a system; a composition of matter; a computer program product embodied on a computer readable storage medium; and/or a processor, such as a processor configured to execute instructions stored on and/or provided by a memory coupled to the processor. In this specification, these implementations, or any other form that the invention may take, may be referred to as techniques. In general, the order of the steps of disclosed processes may be altered within the scope of the invention. Unless stated otherwise, a component such as a processor or a memory described as being configured to perform a task may be implemented as a general component that is temporarily configured to perform the task at a given time or a specific component that is manufactured to perform the task. As used herein, the term 'processor' refers to one or more devices, circuits, and/or processing cores configured to process data, such as computer program instructions.

[0013] A detailed description of one or more embodiments of the invention is provided below along with accompanying figures that illustrate the principles of the invention. The invention is described in connection with such embodiments, but the invention is not limited to any embodiment. The scope of the invention is limited only by the claims and the invention encompasses numerous alternatives, modifications and equivalents. Numerous specific details are set forth in the following description in order to provide a thorough understanding of the invention. These details are provided for the purpose of example and the invention may be practiced according to the claims without some or all of these specific details. For the purpose of clarity, technical material that is known in the technical fields related to the invention has not been described in detail so that the invention is not unnecessarily obscured.

[0014] Figure 1 illustrates an embodiment of an environment in which authentication translation is provided. In the example shown, a variety of client devices 102-108 connect, via one or more networks (represented as a single network cloud 110) to a variety of services 120-124 (also referred to herein as sites 120-124). In particular, client device 102 is a notebook computer owned by a user hereinafter referred to as Alice. Notebook 102 includes a camera, a microphone, and a fingerprint sensor. Client device 104 is a smartphone, also owned by Alice.

Client device 104 includes a camera. Client device 106 is a tablet owned by Bob, and sometimes used by Bob's son Charlie. Client device 106 includes a camera and a fingerprint sensor. Client device 108 is a kiosk located in the lobby of a hotel. Kiosk 108 includes a camera and a microphone. The techniques described herein can be used with or adapted to be used with other devices, as applicable. For example, the techniques can be used in conjunction with gaming systems, with peripheral devices such as mice, and with embedded devices, such as door locks.

[0015] Service 120 is a social networking site. Service 122 is a website of a bank. Service 124 is the online store of a boutique camera retailer. Each of services 120-124 requires a username and password (and/or a cookie) from a user prior to giving that user access to protected content and/or other features. As will be described in more detail below, using the techniques described herein, users need not type such usernames and passwords into their devices whenever required by a service. Instead, users can authenticate themselves to an "authentication translator" via an appropriate technique, and the authentication translator will provide the appropriate credentials to the implicated service on the user's behalf. Also as will be described in more detail below, authentication translators can be located in a variety of places within an environment. For example, notebook computer 102 includes an authentication translator module 132 that provides authentication translation services. The other devices 104-108 can also include (but need not include) their own respective authentication translator modules. The owner of bank website 122 also operates an authentication translator 134 associated with the bank. Finally, authentication translator 136 provides authentication translation services to a variety of businesses, including online camera retailer 124.

[0016] Figure 2 illustrates an embodiment of credential information stored on a device. In particular, device 200 stores three user profiles 202-206, each of which contains a username and one or more templates (e.g., template 210) associated with the user. In various embodiments, a template is a collection of biometric features. Using fingerprints as an example type of biometric, a corresponding template includes a collection of patterns, minutia, and/or other features that can be matched against to determine if a person's fingerprint matches the fingerprint of the registered user (i.e., the owner of a given user profile). A representation of a single fingerprint may be included in multiple templates (e.g., in different resolutions, in accordance with different protocols, as captured during warm vs. cold conditions, and/or by itself

or in combination with multiple fingerprints). When other biometrics are employed (e.g., facial recognition, voiceprint, or retina scan technology), features appropriate to those types of biometrics are included in the template. Other types of features can also be included in templates. As one example, a user's typing speed and/or accuracy can be measured by a device, such as device 102, and used to distinguish between multiple users of a device. For example, suppose Alice types at 100 words per minute and rarely makes mistakes. A representation of this information can be stored in template 212. Also suppose Alice's niece, who sometimes uses Alice's laptop computer when visiting Alice types at 20 words per minute and makes many mistakes. In some embodiments, the fact that a user was recently (e.g., within the last 5 minutes) typing on laptop 102 at 90 words per minute is evidence of a match against template 212. In this case, the typing speed of 90 words per minute is similar enough to Alice's typical behavior, it is considered a match. Various policies can be included in a profile that govern how matches are to be performed. For example, policies can specify thresholds/tolerances for what constitutes a match, and can specify that different levels of matches can result in different levels of access to different resources.

[0017] A profile is associated with a vault (e.g., vault 220). The vault, in turn, contains triples specifying a service provider/domain, a username, and a credential. The vault can also contain other sensitive user information, such as account numbers, address/phone number information, and health care data. The credential for a service provider/domain can be a password (e.g., for legacy servers), and can also take alternate forms (e.g., a cryptographic key for service providers supporting stronger authentication methods).

[0018] In some embodiments, profiles, templates, and vaults (collectively "authentication information") are stored entirely in an unprotected storage area, and are stored in the clear. In other embodiments, secure storage techniques are used to secure at least a portion of the authentication information.

[0019] One example of a device with secure storage is illustrated in Figure 3. In the example shown, a mobile phone 300 includes a large and insecure storage 302 attached to a fast processor 304, and a smaller but secure storage 306 attached to a dedicated processor 308 and a sensor 310 (e.g., a camera or a fingerprint reader). Users (and applications) can read from and

write to the insecure storage area. However, users cannot access the secure storage area, and the fast processor can only communicate with the dedicated processor/sensor via a restricted API. As another example, a unique decryption key associated with a given vault can be stored in a profile. The vault is an encrypted and authenticated container that can be stored on insecure storage, e.g., on the device, and also backed up (e.g., to a cloud storage service 140 or to an alternate form of external storage). As needed, authentication information or portions thereof can be loaded into secure storage and decrypted. For example, one can use AES to encrypt the files one by one, using a key stored on the secured storage. A message authentication technique, such as HMAC, can be used for authenticating the encrypted files to provide tamper prevention. Profiles and vaults can be updated while in secure storage; if this occurs, they are encrypted and MACed before being written back to the insecure storage, which may in turn propagate them to external backup storage. In yet other embodiments, profiles and vaults are stored entirely in secure storage, in plaintext, which allows them to be both read and written -- and in particular, searched.

[0020] Example Transaction Types

[0021] A variety of transaction types can take place in the environment shown in Figure 1, examples of which are discussed in this section.

[0022] Initial Registration

[0023] In order to begin using the techniques described herein, users perform some form of initial registration. As one example, suppose Alice launches an enrollment program installed on laptop 102. She uses the program to capture various biometric information (e.g., fingerprints, photographs of her face, etc.). A user profile is created for Alice, and the biometric information captured about her is encoded into a plurality of templates, such as templates 210 and 214. In some embodiments, Alice is also explicitly asked to supply credential information for services she would like to use, such as by providing the domain name of social networking site 120, along with her username and password for site 120. In other embodiments, domain/username/credential information is at least passively captured on Alice's behalf and included in one or more vaults such as vault 220. Credential information can also be important from a browser password manager already in use by Alice or other appropriate source. In some

embodiments, Alice also registers with cloud storage service 140, which will allow her to back up her authentication information and to synchronize it across her devices (e.g., 102 and 104), as described in more detail below.

[0024] Other registration approaches can also be used. For example, registration can be integrated into the experience the first time a device is used. Thus, when Bob first turns on tablet 106, he may be prompted to take a picture of his face (with a profile/templates being created in response). Similarly, the first time Charlie uses tablet 106, the techniques described herein can be used to determine that Charlie does not yet have a profile (e.g., because none of the templates already present on tablet 106 match his biometrics) and Charlie can be prompted to enroll as a second user of the device.

[0025] Authentication

[0026] Suppose Alice wishes to authenticate to banking website 122. Using a fingerprint reader incorporated into her laptop, she performs a fingerprint scan, which causes her biometric features to be extracted and compared to any stored templates residing on her computer. If a match is found, an associated decryption key is selected, and the associated vault is loaded and decrypted. The vault is scanned for an entry that matches the selected service provider (i.e., website 122). If a matching entry is found, the associated domain, username, and site credential are extracted from the vault. In some embodiments, the validity of the domain name mapping is verified at this point to harden the system against domain name poisoning. Next, a secure connection is established between Alice's computer and the service provider, and Alice is authenticated. For service providers supporting strong user authentication, mutual SSL can be used, for example. A variety of policies can be involved when performing matching. For example, to access certain domains, Alice's print may need only match template 210. To access other domains, Alice may need to match multiple templates (e.g., both 210 and 214). As another example, in order to access social networking site 120, Alice may merely need to be sitting in front of her computer, which has an integrated webcam. Even in relatively low light conditions, a match can be performed against Alice's face and features stored in a template. However, in order to access bank website 122, Alice may need a high quality photograph (i.e., requiring her to turn on a bright light) and may need to demonstrate liveness (e.g., by blinking or turning her

head). As yet another example, other contextual information can be included in policies. For example, if Alice's IP address indicates she is in a country that she is not usually in, she may be required to match multiple templates (or match a template with more/better quality features) in order to access retailer 124, as distinguished from when her IP address indicates she is at home.

[0027] In some embodiments, the biometric sensor used by a user may be a peripheral device (e.g., a mouse with an integrated fingerprint scanner that is connected to the user's primary device via USB). In such scenarios, the peripheral device may be responsible for storing at least a portion of authentication information and may perform at least some of the authentication tasks previously described as having been performed by Alice's computer. For example, instead of processors 304 and 308, and storages 302 and 306 being collocated on a single device (e.g., laptop 102), processor 304 and storage 302 may be present on a primary device, and processor 308 and storage 306 may be present on a peripheral device (e.g., that also includes a sensor, such as a fingerprint reader).

[0028] In such scenarios, once Alice's login to banking website 122 is successfully completed, the secure session can be handed over from the peripheral device to the primary device, in a way that does not allow the primary device retroactive access to the plaintext data of the transcripts exchanged between the peripheral device and the service provider. One way this can be accomplished is by renegotiating SSL keys between the peripheral device and the website, after which the newly negotiated key can be handed off from the peripheral device to the primary device. This avoids retroactive credential capture in a setting where the device is infected by malware.

[0029] An example of renegotiation is depicted in Figure 4. Specifically, after a user has successfully authenticated to a fingerprint reader, a login is performed to a service provider. Using the primary device (404) as a proxy, the peripheral fingerprint reader 402 negotiates a first SSL connection (408) with a service provider 406, over which credentials are exchanged. The proxy then renegotiates SSL (410), which replaces the old key with a new one. The new key is disclosed to the device, which then seamlessly takes over the connection with the service provider and performs the transaction protected by the authentication. The credentials exchanged during the first SSL connection cannot be accessed by device 404, since the key of the

renegotiated session is independent of the key of the original session; this provides protection against malware residing on the device. Renegotiation can be used when the primary device 404 is believed to be in a safe state when performing the negotiation of the SSL connection, but it is not known whether it is in a safe state during the transaction protected by the authentication. Renegotiation can also be used when a secure component of the primary device 404 performs the negotiation of the SSL connection and another and potentially insecure component of the primary device 404 is involved in the transaction protected by the authentication.

[0030] Figure 5 illustrates an embodiment of a process for performing authentication translation. The process begins at 502 when a request to access a resource is received, as is an authentication input. One example of the processing performed at 502 is as follows. Suppose Alice wishes to sign into social networking website 120. She directs a web browser application installed on client 102 to the social networking website. Authentication translator module 132 recognizes, from the context of Alice's actions (e.g., that she is attempting to access site 120 with her browser) that she would like to access a particular resource. Authentication translator module 132 prompts Alice (e.g., by a popup message or via a sound) to provide biometric information (e.g., to use the integrated fingerprint reader on her laptop). In some embodiments, the translator module does not prompt Alice, for example, because Alice has been trained to provide biometric information automatically when attempting to access certain resources. In yet other embodiments, the translator module only prompts Alice if she fails to provide acceptable biometric information within a timeout period (e.g., 30 seconds).

[0031] Module 132 compares Alice's supplied biometric data to the templates stored on her computer. If a suitable match is found, and if an entry for site 120 is present in the applicable vault, at 504, a previously stored credential associated with the resource is accessed. In particular, the username and password for the website, as stored in a vault, such as vault 220, are retrieved from the vault.

[0032] Finally, at 506, the credential is provided to the resource. For example, Alice's username and password for site 120 are transmitted to site 120 at 506. The credential can be transmitted directly (e.g., by the module or by Alice's computer) and can also be supplied

indirectly (e.g., through the use of one or more proxies, routers, or other intermediaries, as applicable).

[0033] Other devices can also make use of process 500 or portions thereof. For example, when Alice launches a banking application on phone 104, implicit in her opening that application is her desire to access the resources of website 134. The application can take Alice's picture and compare it to stored templates/vault information. If an appropriate match is found, a credential can be retrieved from the vault on her phone (or, e.g., retrieved from cloud storage service 140) and provided to website 134.

[0034] As another example, suppose Charlie is using tablet 106 and attempts to visit site 120, whether via a dedicated application or via a web browser application installed on the tablet. Charlie's photograph is taken, and then compared against the profiles stored on tablet 106 (e.g., both Bob and Charlie's profiles). When a determination is made that Charlie's photograph matches a template stored in his stored profile (and not, e.g., Bob's), Charlie's credentials for site 120 are retrieved from a vault and transmitted by an authentication translator module residing on client 106.

[0035] As yet another example, kiosk 108 can be configured to provide certain local resources (e.g., by displaying a company directory or floor plan on demand) when users speak certain requests into a microphone. Enrolled users (e.g., with stored voiceprint or facial recognition features) can be granted access to additional/otherwise restricted services in accordance with the techniques described herein and process 500.

[0036] New device

[0037] In some embodiments, to register a new device, a user provides an identifier, such as a username or an account number to the device. The new device connects to an external storage (such as cloud storage 140), provides the user identifier and credential, and downloads the user's templates/vaults from the service. In some embodiments, the templates/vaults are encrypted. Once downloaded, the template is decrypted and stored in a secure storage area, while the still encrypted vault can be stored in insecure storage. The decryption key can be generated from information the user has/knows, or from biometric data -- such as features

extracted from fingerprinting of all ten fingers. In some embodiments, more arduous fingerprinting is required for the setup of a new device than for regular authentication to avoid that a new device gets registered by a user thinking she is merely authenticating -- or worse still, simply touching the device. Moreover, it translates into higher entropy of the decryption keys.

[0038] Backup Authentication

[0039] Backup authentication allows a user, such as Alice, to access resources in the event she is unable to or unwilling to interact with a particular biometric sensor. As one example, instead of having a single template associated with her profile, Alice can have multiple templates associated with it, e.g., where the first template includes fingerprint features and the second template includes voice biometric, facial recognition, or iris detection features. As a second example, where the service Alice is connecting to is a legacy website (i.e., one that users authenticate to using usernames and passwords), such a service would allow the use of passwords and password reset mechanisms by Alice without requiring Alice to use a fingerprint reader.

[0040] In various embodiments, environment 100 supports the ability of users (e.g., under duress) to release the contents of their vaults. For example, if Alice was physically threatened with the loss of a finger by a criminal, Alice could instead release the contents of her vault(s) – the ultimate goal of the criminal. As one example, in the event Alice supplies all 10 fingerprints to the sensor, provides a special password, or supplies a fingerprint and a second identifier, a cleartext version of her vault(s) could be made available.

[0041] Access Policies

[0042] In various embodiments, cloud storage service 140 is configured to accept backups from multiple devices associated with a single account, and synchronize the updates so that all devices get automatically refreshed. For example, Alice's laptop 102 and phone 104 could both communicate with cloud storage service 140 which would keep their authentication information synchronized. Refreshes can also be made in accordance with user-configured restrictions. For example, Alice's employer could prevent privileged employer data from being stored on shared personal devices, or on any device that was not issued by the employer. As

another example, arbitrary policies can be defined regarding the access to and synchronization of software and data, and to tie a license or access rights to a person (and her fingerprint) rather than to a device. As yet another example, in some embodiments (e.g., where a device is made publicly available or otherwise shared by many users), no or a reduced amount of authentication information resides on a device, and at least a portion of authentication information is always retrieved from cloud storage service 140.

[0043] Remote wiping

[0044] Remote wiping of a user's authentication information (e.g., templates) can be used both to "unshare" previously shared devices (e.g., where Bob and Charlie both have user profiles on their shared tablet 106), and to avoid that criminals with physical component access to lost devices gain access to templates and vault contents. In some embodiments, polices such as ones where a template self-wipes if it is not matched within a particular duration of time are supported. Since user data can be frequently backed up to the cloud storage, and recovered from this using the new device registration process, inconvenience to the user will be minimized.

[0045] **Legacy Server Support**

[0046] New authentication schemes typically require changes to a significant codebase residing with service providers. If the code is well written and documented, such changes may be relatively simple. Commonly, though, this may not be so. The engineers who originally wrote the code of relevance may have long since left the company; the code they left behind may be poorly documented -- if documented at all. In severe cases, the legacy code may have been written in an outdated programming language or written in a way that does not follow guidelines for good code. This makes updates to the codebase impractical or virtually impossible in many common cases. Even if none of these challenges complicate the desired modifications, it is commonly a great bureaucratic burden to obtain permission to make modifications (e.g., to store an extra field in a backend database), because every affected part of the organization may need to review the request.

[0047] As will be described in the following section, the technologies described herein can be used in conjunction with legacy servers (e.g., existing servers that rely on usernames and

passwords to authenticate users), and in particular, can be used without requiring modification to such legacy servers.

[0048] Cookies

[0049] Cookies are commonly used by legacy servers for user authentication. Unfortunately, cookies have several problems. For one thing, they are sometimes deleted -- whether explicitly/intentionally by the end user or by the user's software. In addition, cookies are commonly stolen. Approaches such as cache cookies and identification using user agents can be more resistant to these problems, however, they have their own problems. For example, their use requires new code and new fields in the credential database stored by the server.

[0050] In some embodiments, authentication translators, such as translators 134 and 136 (also referred to herein as proxies) provide authentication translation services on behalf of associated services. Translators 134 and 136 are illustrated as single logical devices in Figure 1. In some embodiments, the translators comprise standard commercially available server hardware (e.g., a multi-core processor, 4+ Gigabytes of RAM, and one or more Gigabit network interface adapters) and run typical server-class operating systems (e.g., Linux). Translators 134 and 136 can also be implemented using a scalable, elastic architecture and may comprise several distributed components, including components provided by one or more third parties. For example, translators 134 and 136 may store user credential information or may task cloud storage service 140 with storing at least a portion of that information.

[0051] In the case of authentication translator 134, service is provided with respect to bank website 122 only. Authentication translator 134 is positioned between a legacy web server (122) and the Internet (110) -- and therefore between the legacy server and any client devices. Authentication translator 134 is configured to translate traffic between the legacy server and client devices so that the client devices (and respective users) perceive the new authentication mechanism, while the legacy server remains unchanged. Authentication translator 136 works similarly, but it provides authentication translation as a third party service to multiple providers, an example of which is online camera retailer 124.

[0052] Authentication translators 134 and 136 can also perform process 500. For example, when a device transmits a request to access website 122, the request is intercepted by translator 134, as is cookie/user agent information. The received information can be used to determine a username/password associated with the device, and that information can be passed by translator 134 to website 122 on behalf of the device.

[0053] Figure 6 illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator. The translator (referred to in the figure as a "proxy") fails to identify the client, and passes on the request to the legacy server. The legacy server responds to the request and sets a cookie. The proxy passes on the response, including the cookie and also a cache cookie. The proxy stores the information about both these types of identifiers, along with the user agent of the client device. This triplet of information is also referred to herein as an identifier.

[0054] Figure 7 illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator. In this scenario, the device's request is accompanied by some form of identifying information. The proxy uses this information to identify the associated cookie and passes this along to the legacy server, along with the request. In some embodiments, additional processing is also involved. For example, reading a cache cookie may require user interaction. Moreover, if not all of the identifying information is present in the request, the proxy can be configured to set the missing information again by sending a corresponding request to the client device.

[0055] The translation of cache cookies and user agent information to cookies involves a two-way translation. First, when the legacy server sets a cookie, the proxy will set the two types of cookies -- both an HTML cookie and a cache cookie -- and then create a new record in which the two cookies are stored, along with the user agent information of the client device. The user agent information can include quite a bit of data associated with a browser -- such as the browser type and version, the clock skew, and the fonts that are installed. While each of these pieces of information only contributes a small amount of entropy, the collection of items can be sufficient to identify the device in most cases. Moreover, while some of these types of data may change over time -- in fact, all of them may -- they do not typically change, and when one or two of

them do, the others typically do not. When the client device is used to visit a site controlled by the legacy server, the cookie, cache cookie and user agent information are read (if available), the record identified, and the request translated and sent to the legacy server. When a legacy server requests that the user password is updated (e.g., as part of an annual or other periodic requirement), the transmission of this request to the user can be suppressed -- in which case the database of the proxy is updated to create the illusion of an update. The user can be involved in authentication as needed, e.g., if, in addition to supplying a credential, a user must also solve a CAPTCHA, the CAPTCHA can be displayed to the user (with the user's credentials being handled in accordance with the techniques described herein).

[0056] Figure 8 shows the structure of an example of a cache cookie used in some embodiments. Cache cookies can be associated with a particular webpage, just like an HTML cookie can. In the example shown in Figure 8, the proxy wishes to associate a page "calling page.html" with a cache cookie. It embeds a request for a second object, "samename.html" in callingpage.html for every visitor. However, as the cache cookie is set for one visitor, a customized samename.html is served to this visitor. The page samename.html refers to an object with a different name for each user; that object is referred to as "uniquename.jpg." The cache cookie is set by embedding the request for samename.html in callingpage.html. The client browser attempts to render this, causing a request for samename.html from the server. The server configures samename.html to refer to a uniquely named file uniquename.jpg, and serves samename.html to the client. For the client browser to render samename.html, it requests the file uniquename.jpg, which is intentionally not served. That concludes the setting of a cache cookie. As a user returns to the page callingname.html, the browser again attempts to render the entire page, which causes it to load the object samename.html from its cache. As that is rendered, the client browser requests uniquename.jpg, which is not in its cache (since it was not served previously). The server still does not serve it, but takes note of the name of the file being requested, as it identifies the client device. Note that samename.html can be displayed in a zero-sized iframe, which makes the end user unaware of it being rendered.

[0057] A cache cookie is an implementation of the typical cookie functionality that uses the client device's browser cache. Unlike user agents, it does not change over time, and like standard HTML cookies, it cannot be read by a party other than that which set it. However, like

HTML cookies, it could be deleted -- by the user clearing his or her browser cache. Cache cookies are not automatically transmitted with GET requests, unless the cache elements are embedded in the referring pages. This adds a potential round of communication in some settings. By relying on user agent information, cache cookies, and HTML cookies to identify the client device, it is much more likely that a machine will be recognized than if only HTML cookies are used.

[0058] Although the foregoing embodiments have been described in some detail for purposes of clarity of understanding, the invention is not limited to the details provided. There are many alternative ways of implementing the invention. The disclosed embodiments are illustrative and not restrictive.

[0059] WHAT IS CLAIMED IS:

CLAIMS

1. A system, comprising:

a processor configured to:

receive, at an authentication translator, a request to access a resource and an authentication input, wherein the authentication input corresponds to at least one stored record and wherein the stored record is associated at least with the resource;

in response to the receiving, access a previously stored credential associated with the resource; and

cause the credential to be provided to the resource; and

a memory coupled to the processors and configured to provide the processor with instructions.

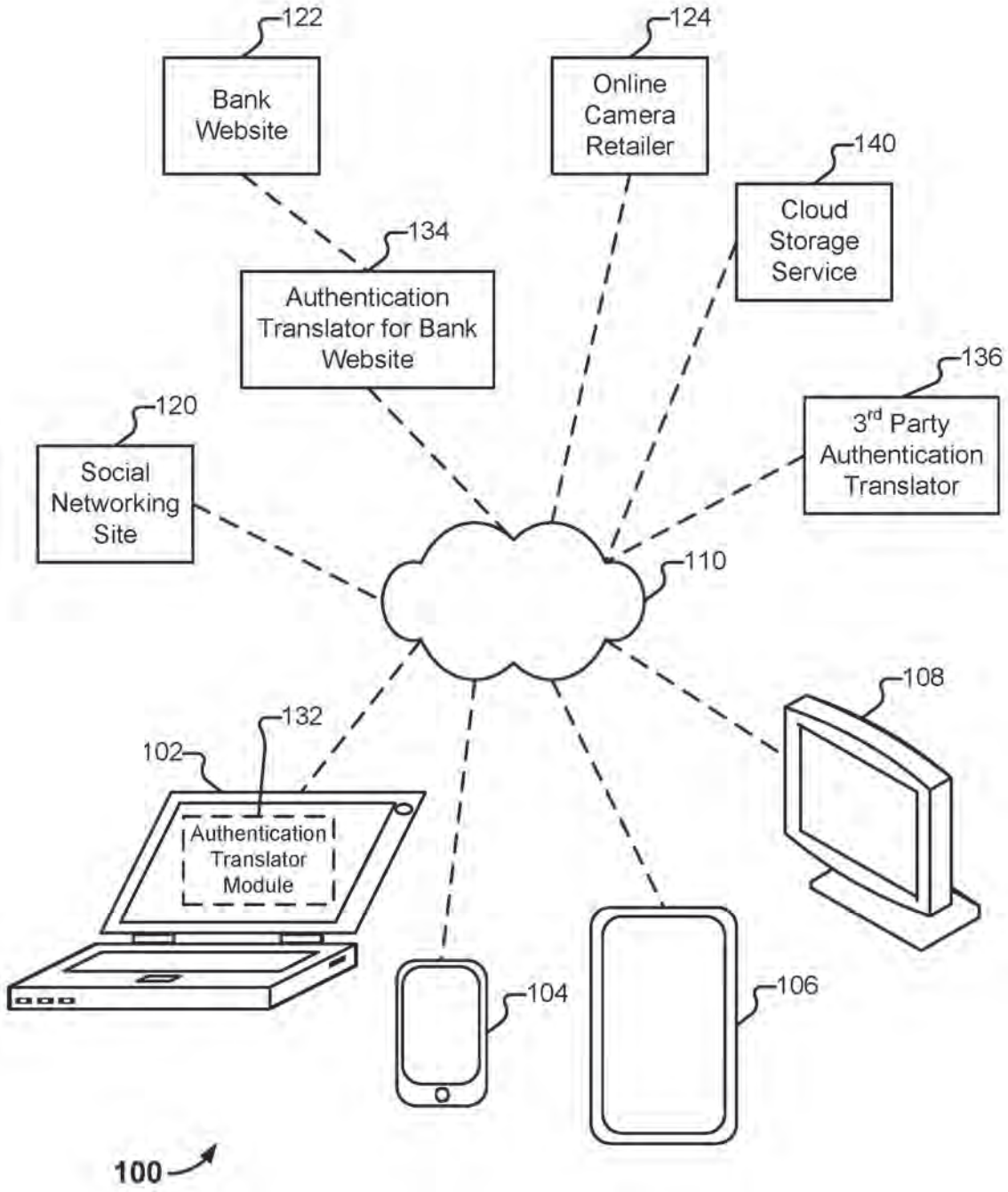

122
Bank Website
124
Online Camera Retailer
140
Cloud Storage Service
134
Authentication Translator for Bank Website
136
3rd Party Authentication Translator
120
Social Networking Site
110
108
132
102
Authentication Translator Module
104
106
100

FIG. 1

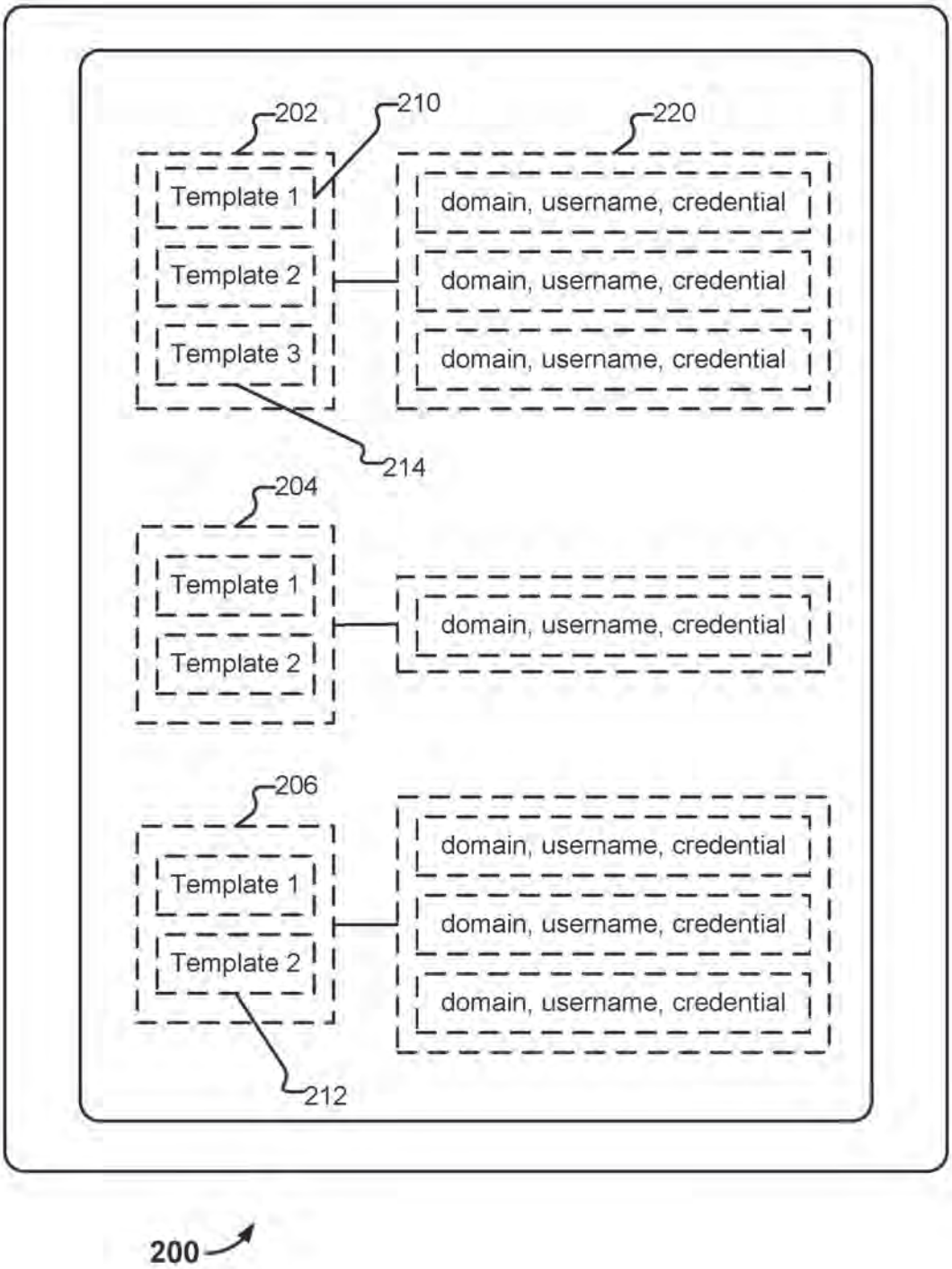
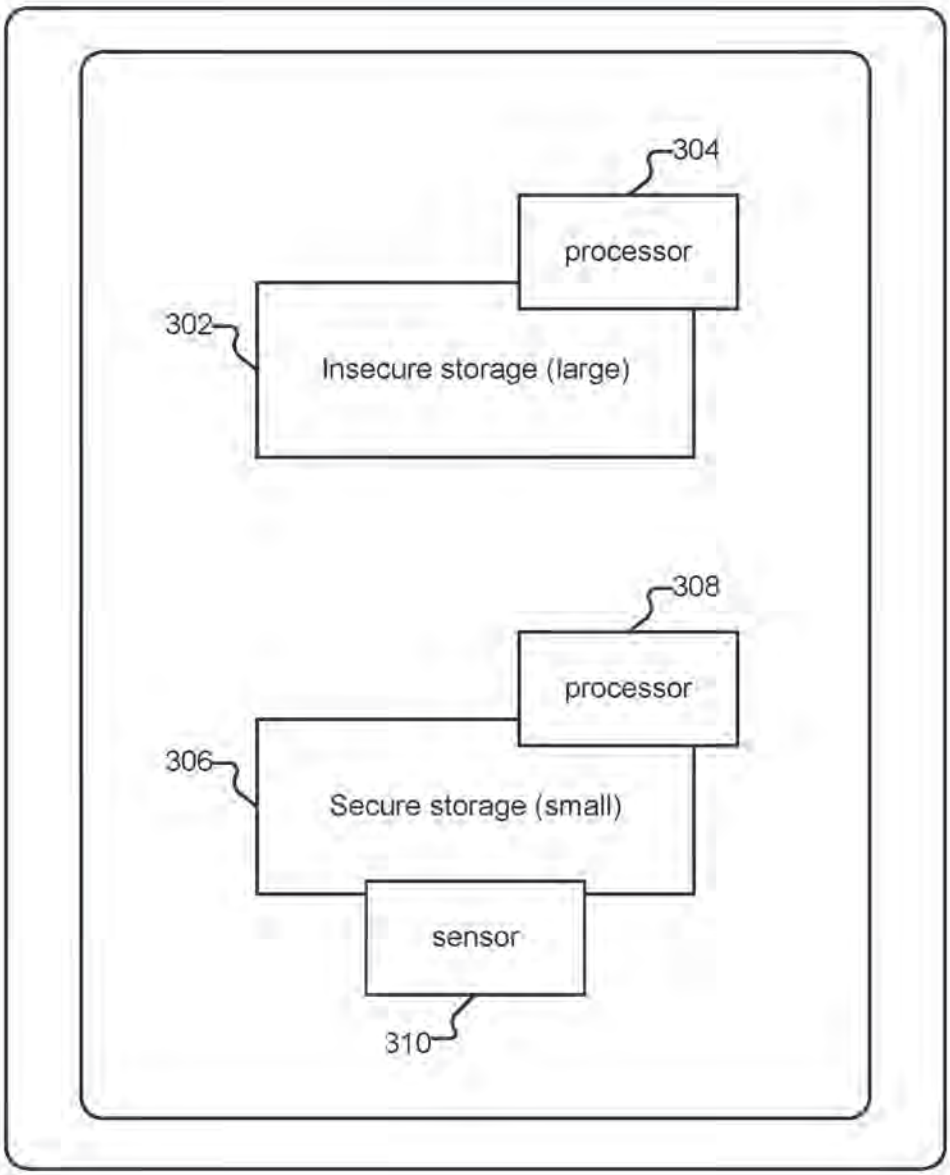

202
210
220
Template 1
Template 2
Template 3
domain, username, credential
domain, username, credential
domain, username, credential
214
204
Template 1
Template 2
domain, username, credential
206
Template 1
Template 2
domain, username, credential
domain, username, credential
domain, username, credential
212
200

FIG. 2


304
processor
302
Insecure storage (large)
308
processor
306
Secure storage (small)
sensor
310

300

FIG. 3

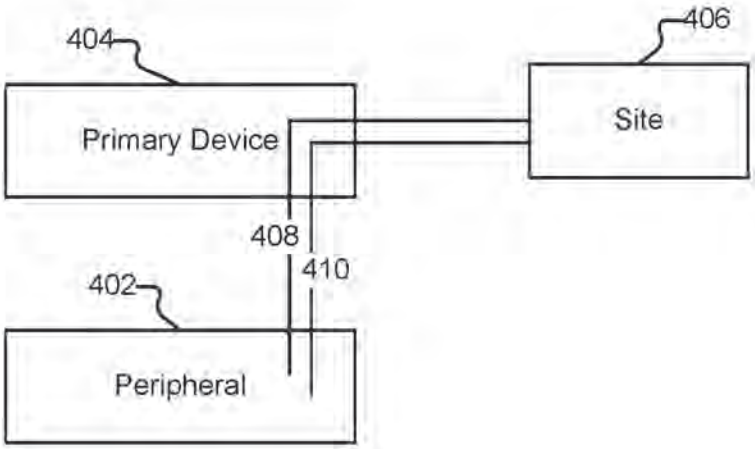


FIG. 4

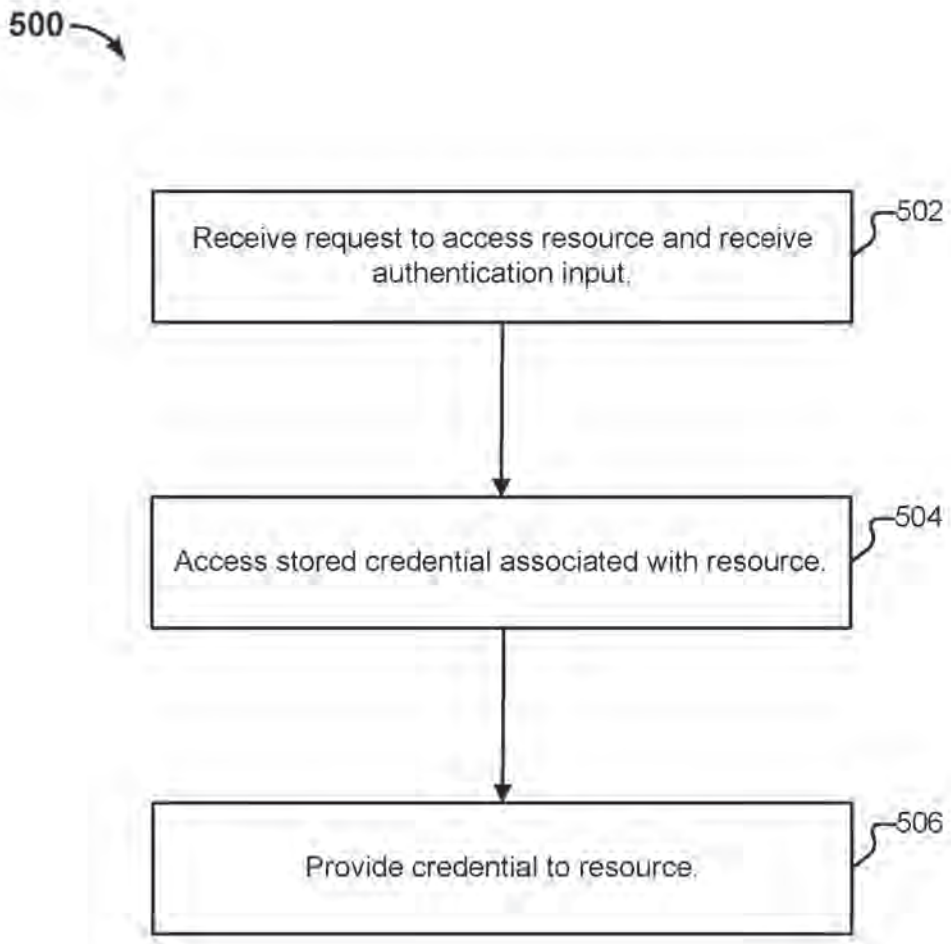

500
Receive request to access resource and receive authentication input.
502
Access stored credential associated with resource.
504
Provide credential to resource.
506

FIG. 5

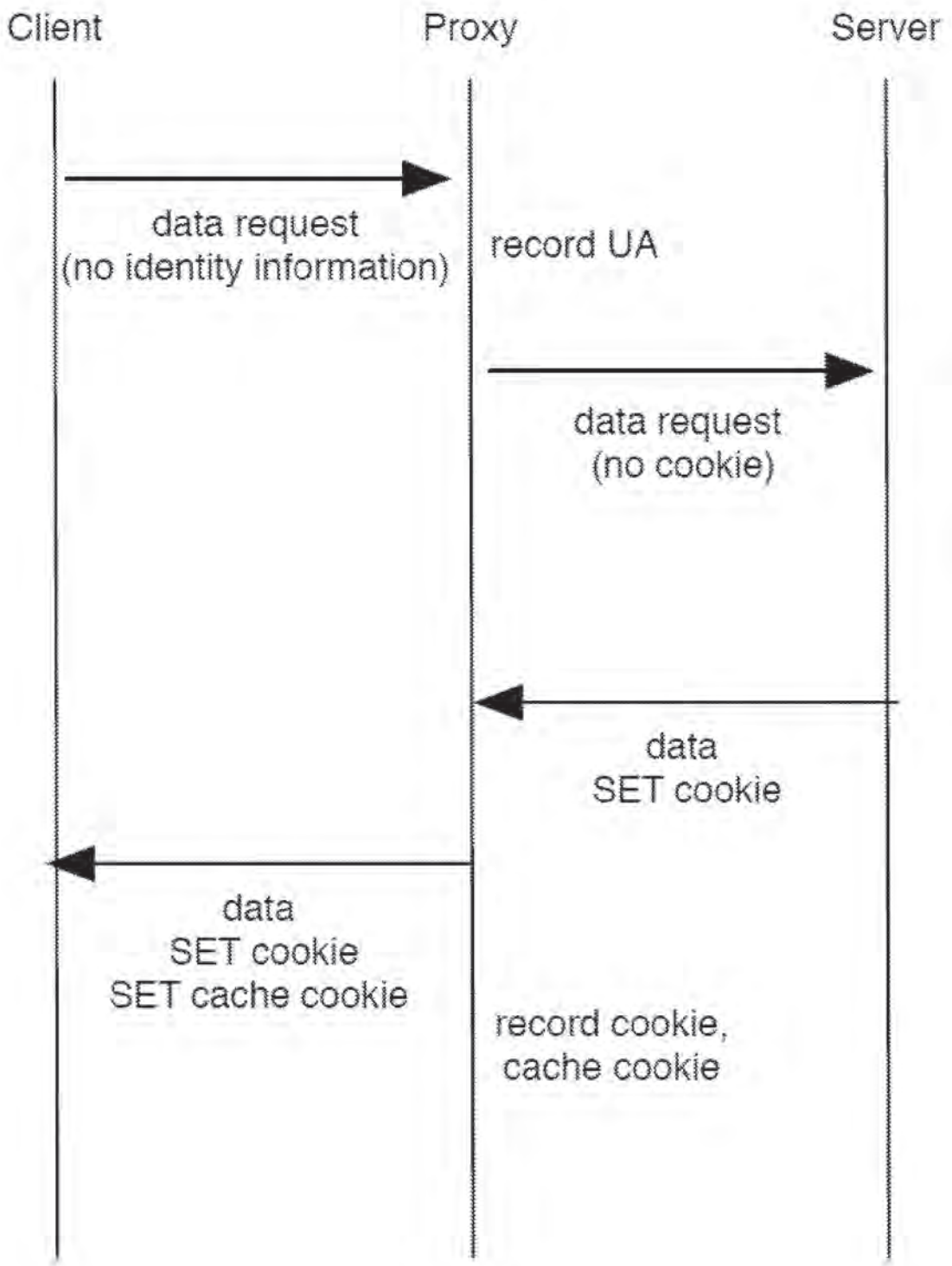

Client
Proxy
Server
data request
(no identity information)
record UA
data request
(no cookie)
data
SET cookie
data
SET cookie
SET cache cookie
record cookie,
cache cookie

FIG. 6

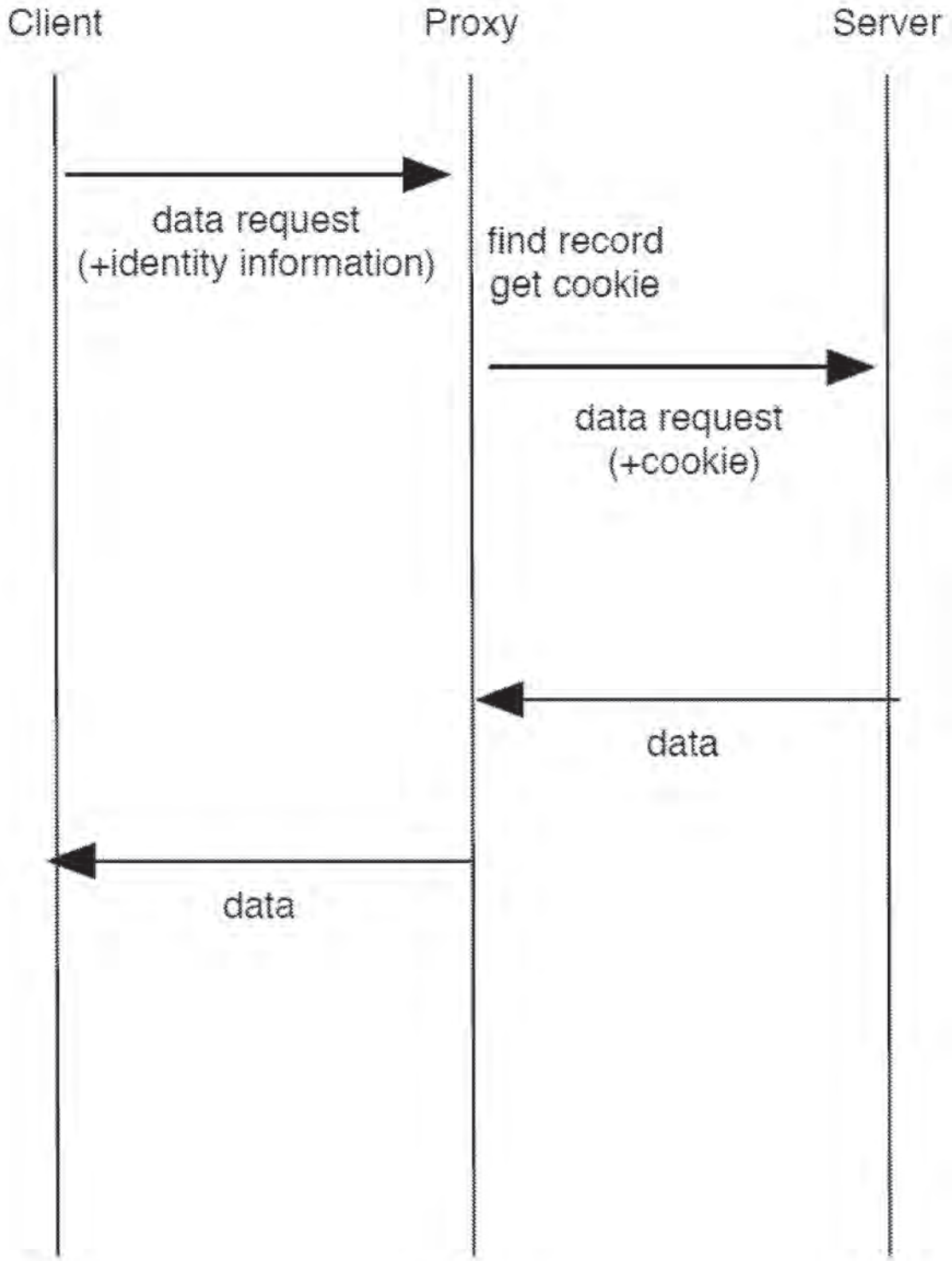

Client
Proxy
Server
data request
(+identity information)
find record
get cookie
data request
(+cookie)
data
data

FIG. 7

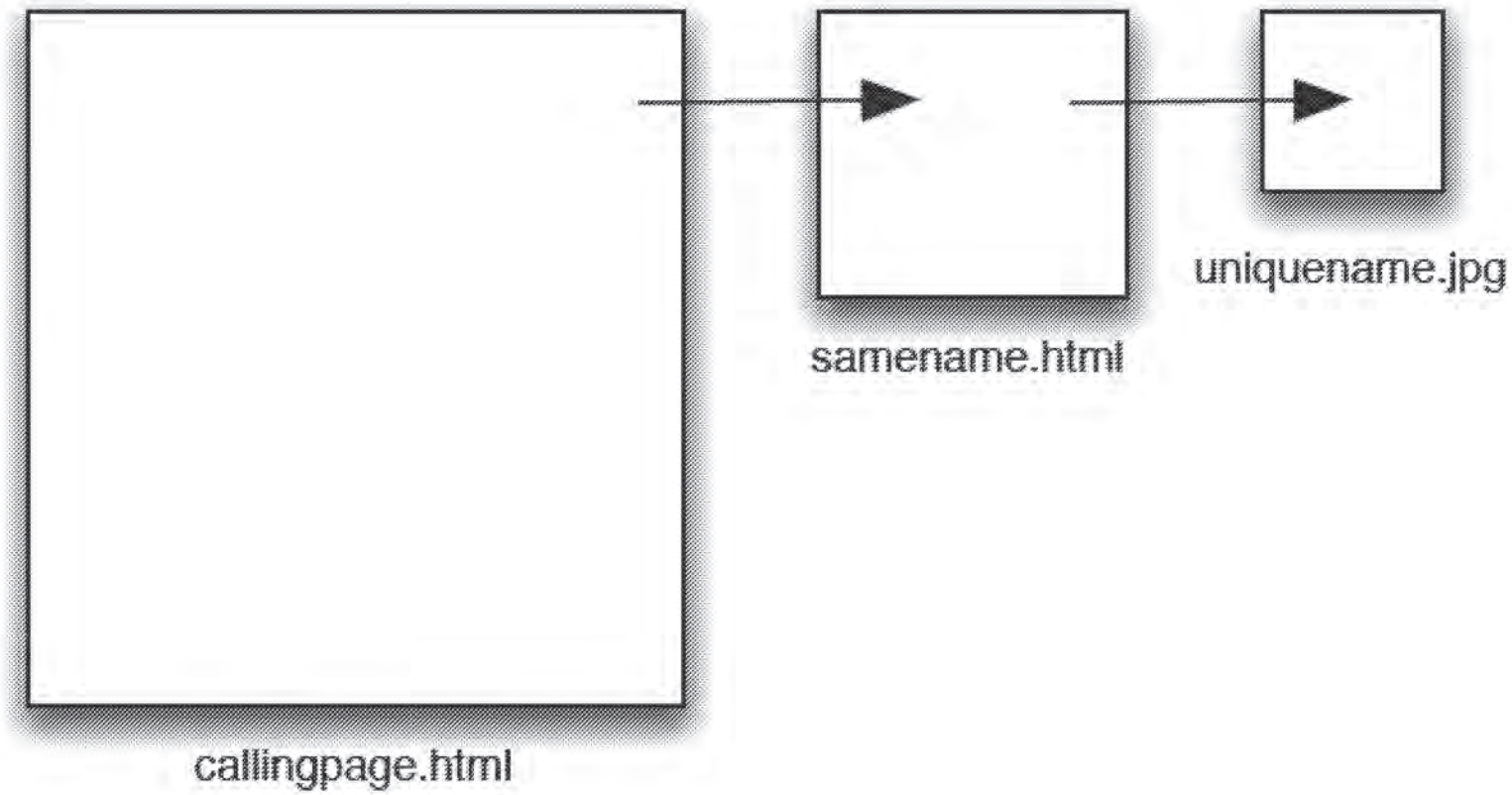


FIG. 8

Application Serial No. 15/042,636

Filing date: February 12, 2016

Patent No. 10,360,351

Issue date: July 23, 2019

US010360351B1

(12) **United States Patent**
Jakobsson

(10) **Patent No.: US 10,360,351 B1**
(45) **Date of Patent: *Jul. 23, 2019**

(54) **AUTHENTICATION TRANSLATION**

(71) Applicant: **RightQuestion, LLC**, Portola Valley, CA (US)

(72) Inventor: **Bjorn Markus Jakobsson**, Portola Valley, CA (US)

(73) Assignee: **RightQuestion, LLC**, Portola Valley, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **15/042,636**

(22) Filed: **Feb. 12, 2016**

Related U.S. Application Data

(63) Continuation of application No. 13/706,254, filed on Dec. 5, 2012, now Pat. No. 9,294,452.

(60) Provisional application No. 61/569,112, filed on Dec. 9, 2011, provisional application No. 61/587,387, filed on Jan. 17, 2012.

(51) **Int. Cl.**

G06F 21/00	(2013.01)
G06F 21/10	(2013.01)
H04L 29/06	(2006.01)
G06F 21/12	(2013.01)

(52) **U.S. Cl.**
CPC ***G06F 21/10*** (2013.01); ***G06F 21/121*** (2013.01); ***G06F 21/128*** (2013.01); ***H04L 63/083*** (2013.01); ***H04L 63/0861*** (2013.01); ***H04L 63/10*** (2013.01); ***H04L 63/20*** (2013.01)

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,016,476	A	1/2000	Maes	
7,512,965	B1 *	3/2009	Amdur	H04L 63/20 726/1
7,950,051	B1 *	5/2011	Spitz	G06F 21/31 380/277
8,549,300	B1 *	10/2013	Kumar	H04L 9/3247 713/153
8,577,813	B2	11/2013	Weiss	
8,856,539	B2	10/2014	Weiss	
9,100,826	B2	8/2015	Weiss	
2004/0107170	A1	6/2004	Labrou	
2004/0236632	A1	11/2004	Maritzen	

(Continued)

FOREIGN PATENT DOCUMENTS

WO	WO-2004051585	A2	6/2004
WO	2005001751	A1	1/2005

OTHER PUBLICATIONS

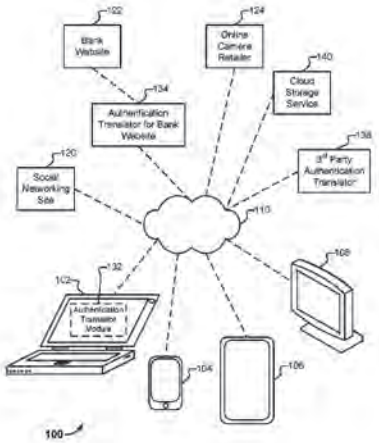
Hammer-Lahav, Ed. "The OAuth 1.0 Protocol", from https://tools.ietf.org/html/rfc5849, Apr. 2010.

Primary Examiner — Andrew J Steinle
(74) *Attorney, Agent, or Firm* — Van Pelt, Yi & James LLP

(57) **ABSTRACT**

Authentication translation is disclosed. A request to access a resource is received at an authentication translator, as is an authentication input. The authentication input corresponds to at least one stored record. The stored record is associated at least with the resource. In response to the receiving, a previously stored credential associated with the resource is accessed. The credential is provided to the resource.

32 Claims, 8 Drawing Sheets



(56) **References Cited**

U.S. PATENT DOCUMENTS

2005/0198348	A1*	9/2005	Yeates	H04L 12/6418 709/232
2009/0100269	A1*	4/2009	Naccache	H04L 9/3271 713/186
2010/0242102	A1*	9/2010	Cross	G06F 21/32 726/7
2011/0205016	A1*	8/2011	Al-Azem	H04L 63/0861 340/5.52
2011/0231651	A1*	9/2011	Bollay	H04L 63/166 713/152
2012/0110341	A1*	5/2012	Beigi	G06Q 20/3223 713/186
2012/0167193	A1*	6/2012	Gargaro	G06F 21/41 726/8

* cited by examiner

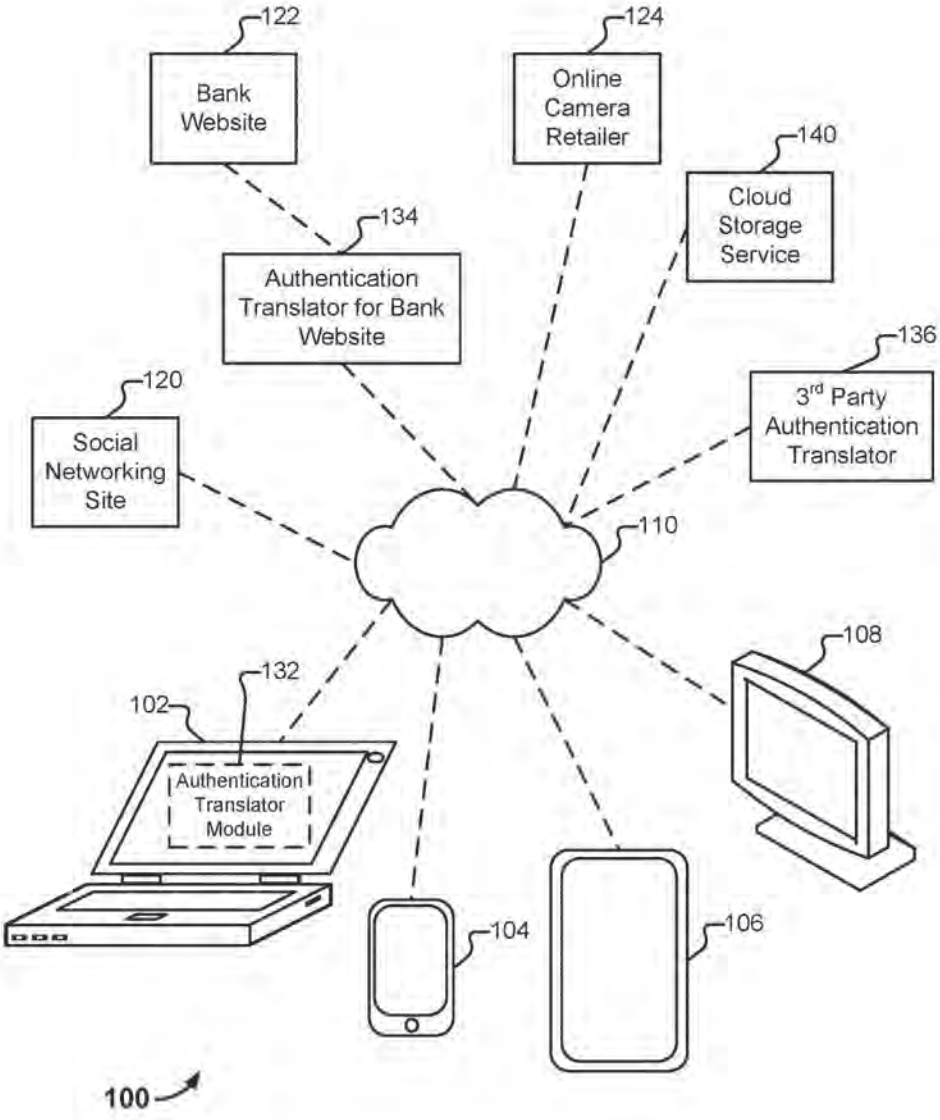

122
Bank Website
124
Online Camera Retailer
140
Cloud Storage Service
134
Authentication Translator for Bank Website
136
3rd Party Authentication Translator
120
Social Networking Site
110
108
132
102
Authentication Translator Module
104
106
100

FIG. 1

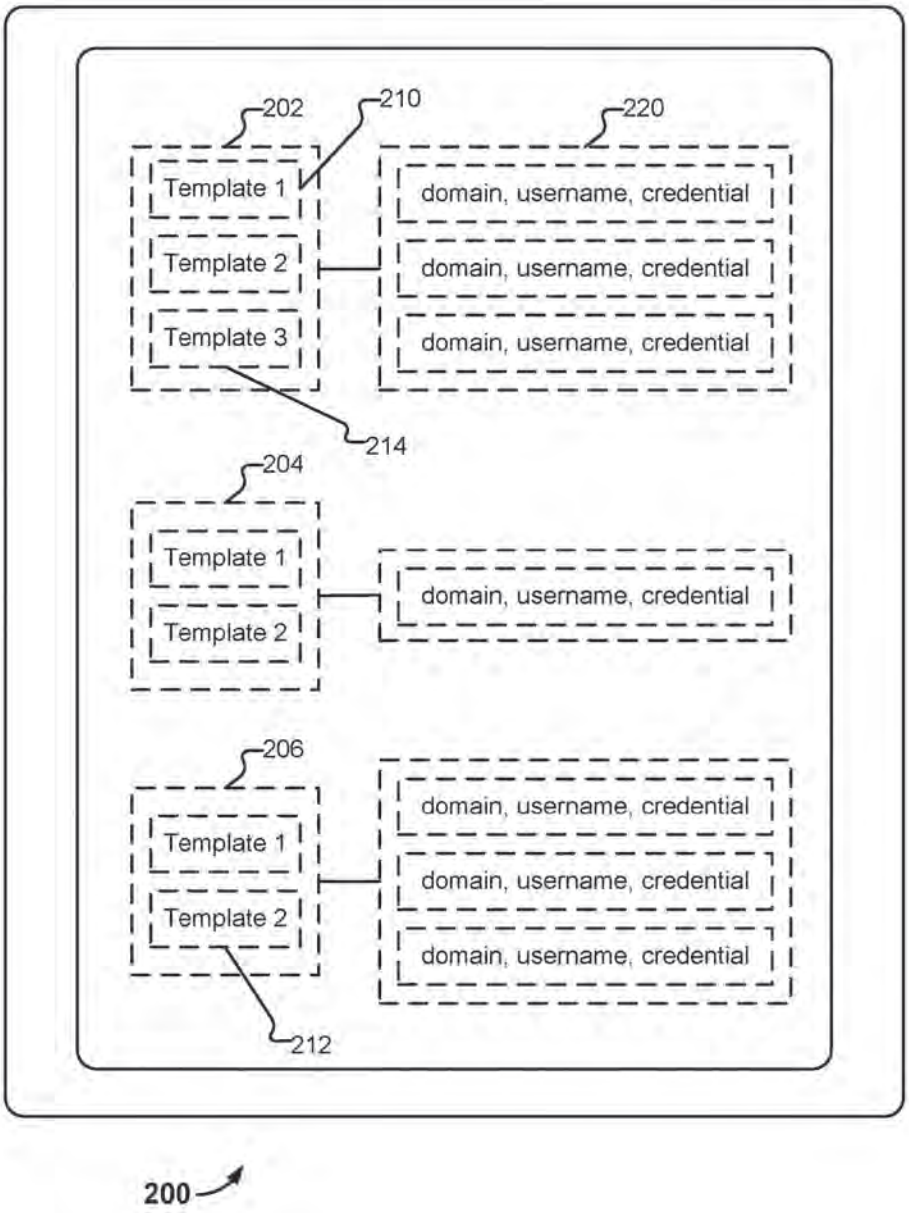
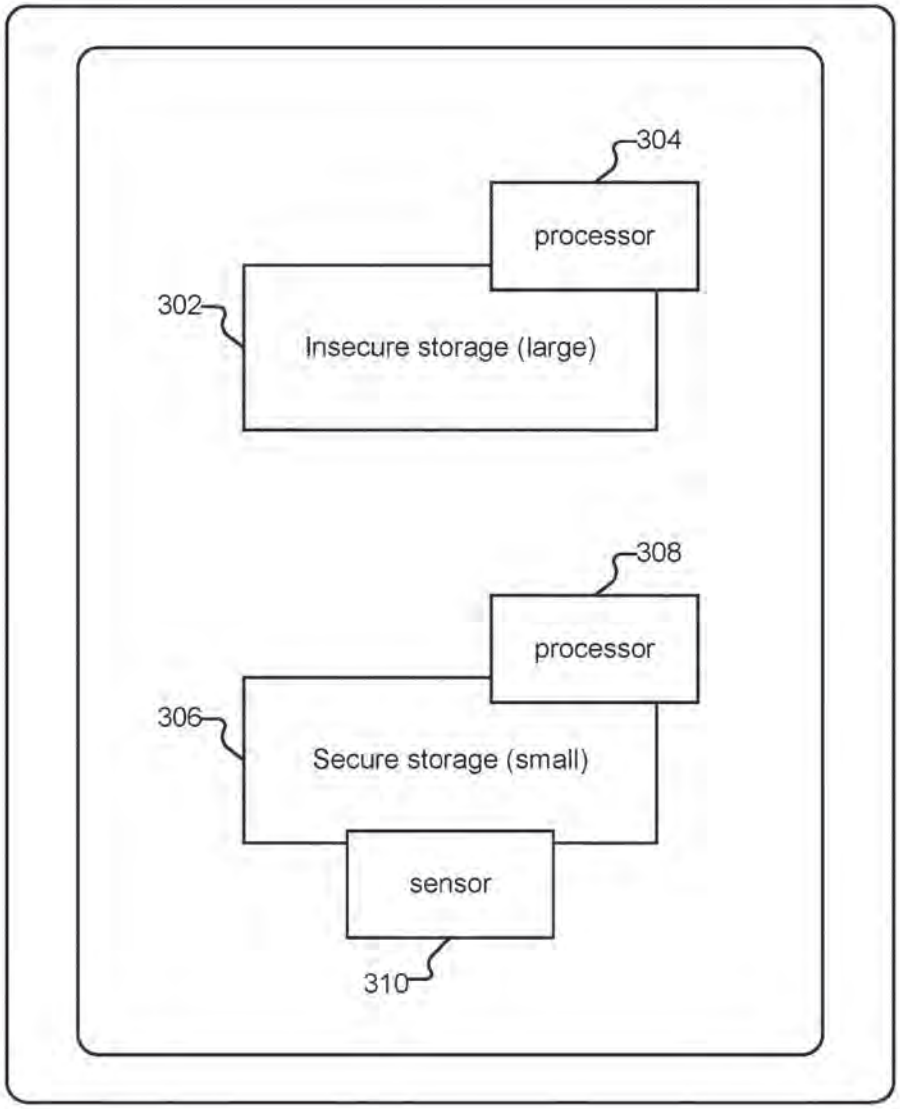

202
210
220
Template 1
Template 2
Template 3
domain, username, credential
domain, username, credential
domain, username, credential
214
204
Template 1
Template 2
domain, username, credential
206
Template 1
Template 2
domain, username, credential
domain, username, credential
domain, username, credential
212
200

FIG. 2


304
processor
302
Insecure storage (large)
308
processor
306
Secure storage (small)
sensor
310

300

FIG. 3

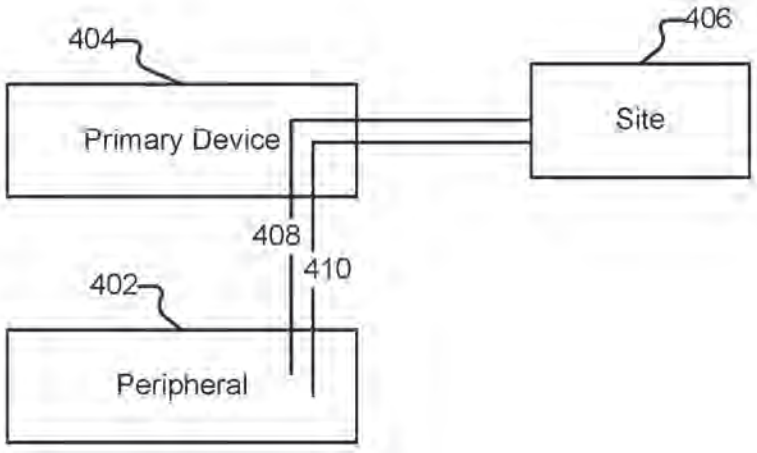


FIG. 4

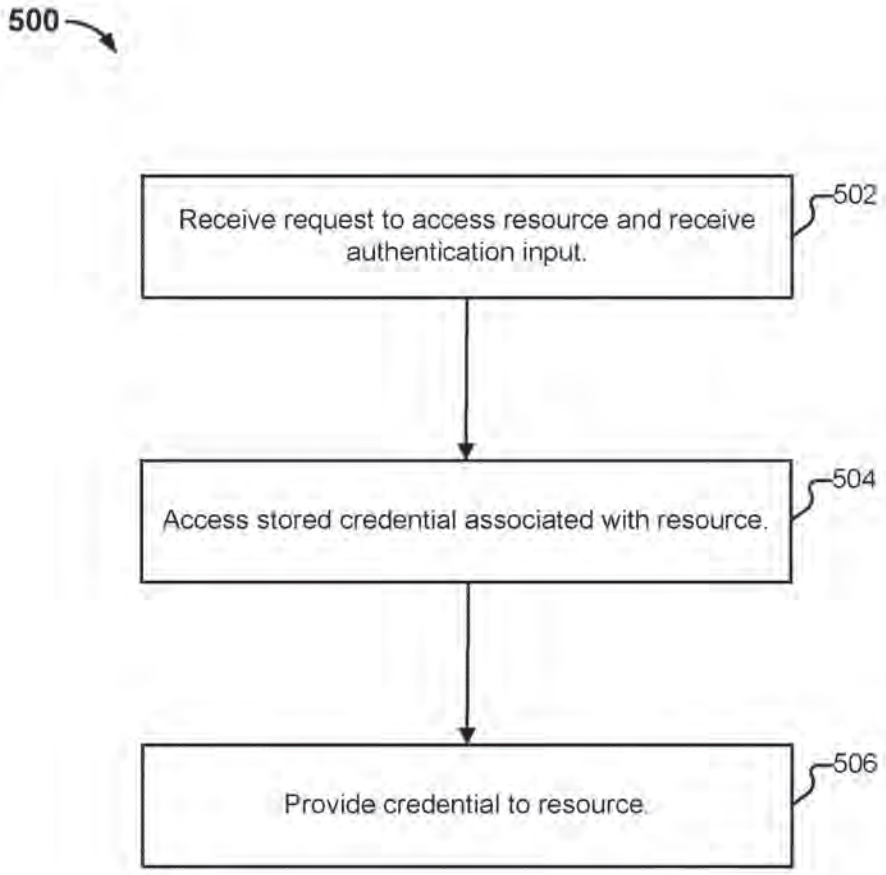

500
Receive request to access resource and receive authentication input.
502
Access stored credential associated with resource.
504
Provide credential to resource.
506

FIG. 5

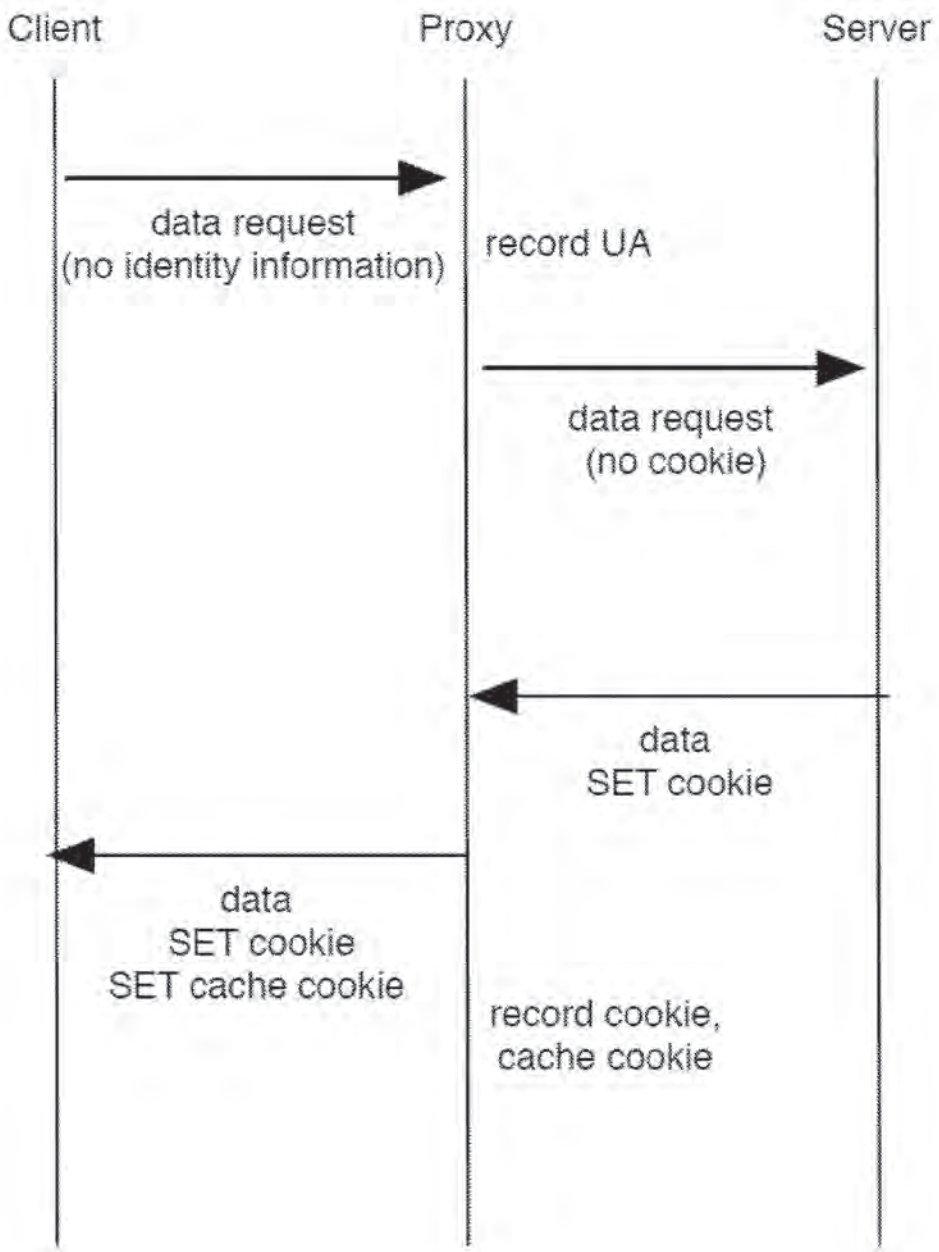

Client
Proxy
Server
data request
(no identity information)
record UA
data request
(no cookie)
data
SET cookie
data
SET cookie
SET cache cookie
record cookie,
cache cookie

FIG. 6

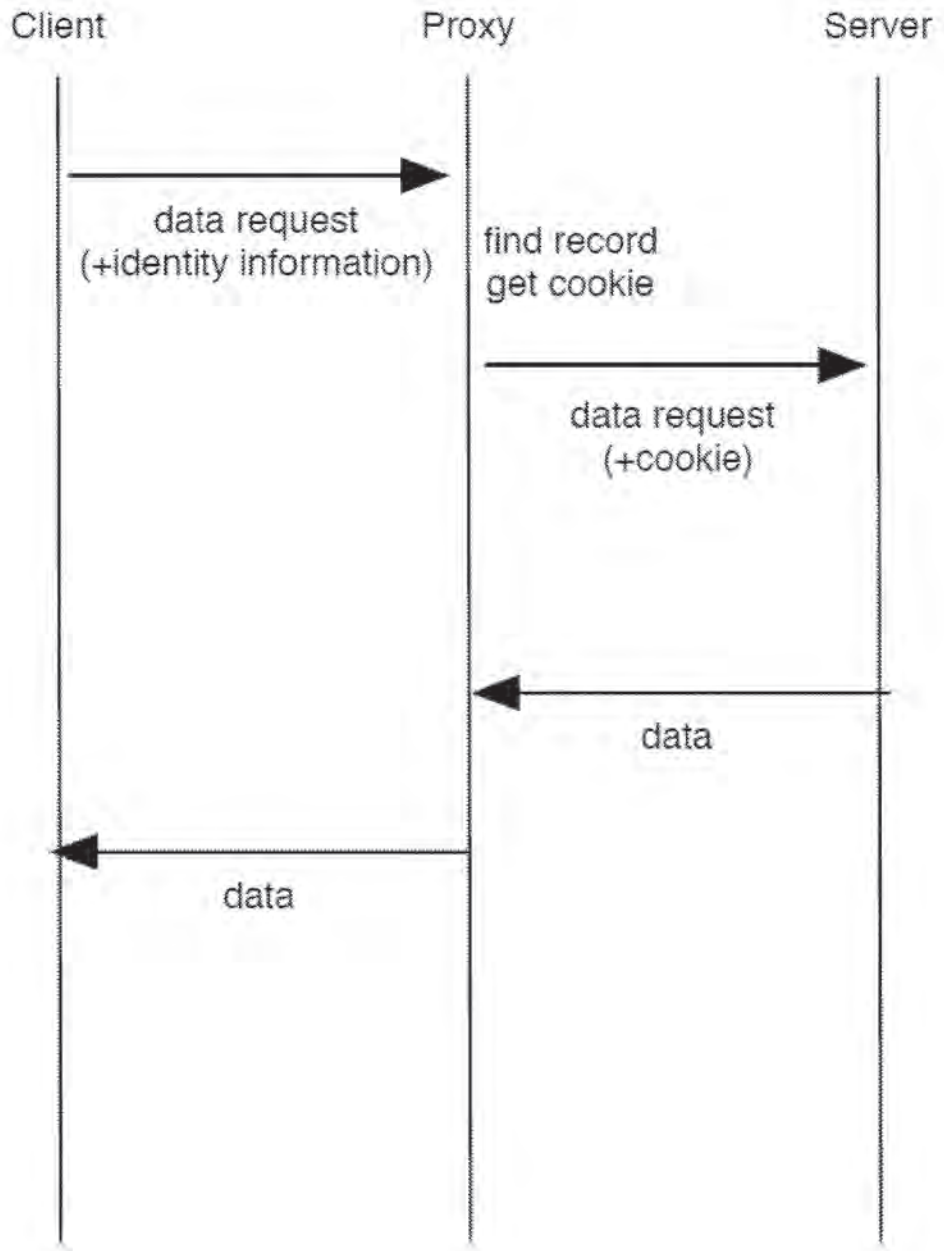

Client
Proxy
Server
data request
(+identity information)
find record
get cookie
data request
(+cookie)
data
data

FIG. 7

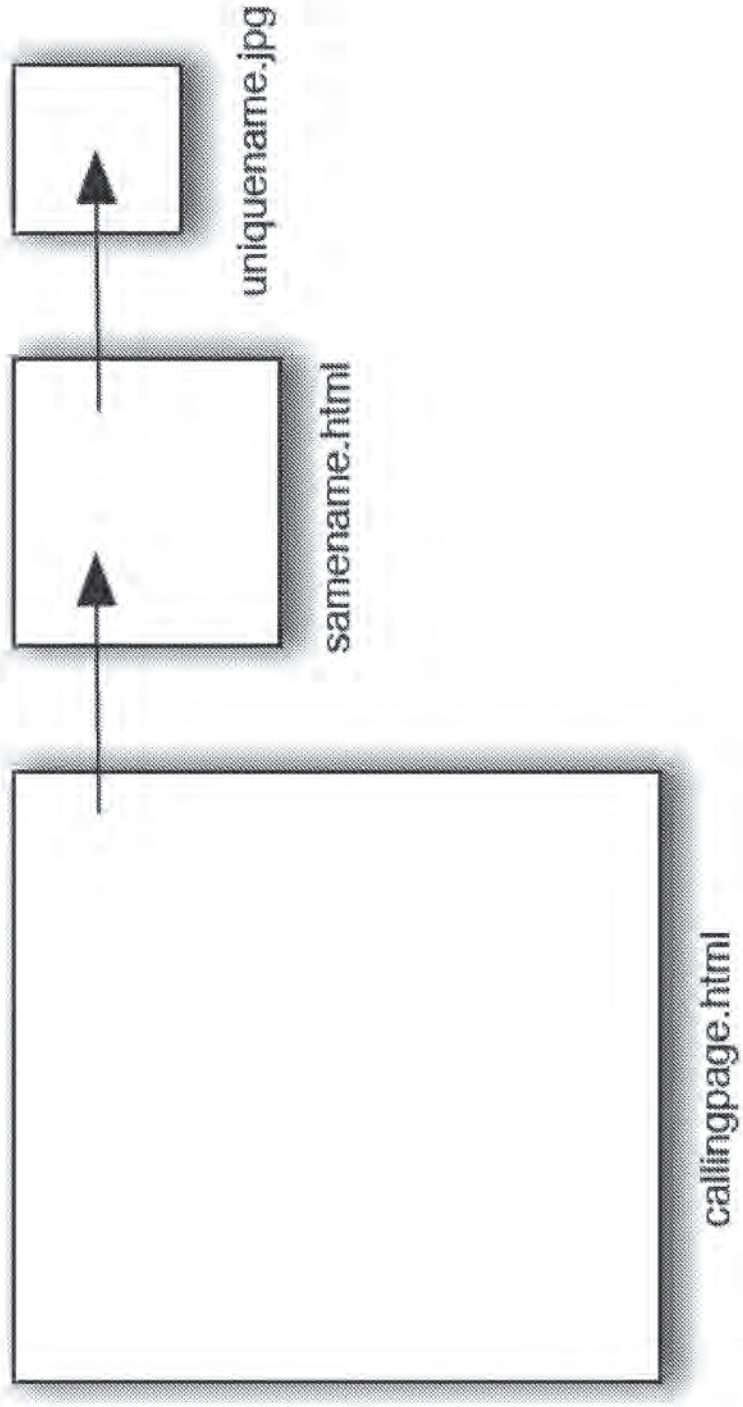


FIG. 8

AUTHENTICATION TRANSLATION

CROSS REFERENCE TO OTHER APPLICATIONS

This application is a continuation of co-pending U.S. patent application Ser. No. 13/706,254, entitled AUTHENTICATION TRANSLATION filed Dec. 5, 2012 which is incorporated herein by reference for all purposes. U.S. patent application Ser. No. 13/706,254 claims priority to U.S. Provisional Patent Application No. 61/569,112 entitled BACKWARDS COMPATIBLE ROBUST COOKIES filed Dec. 9, 2011, and also claims priority to U.S. Provisional Patent Application No. 61/587,387 entitled BIOMETRICS-SUPPORTED SECURE AUTHENTICATION SYSTEM filed Jan. 17, 2012, both of which are incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

Providing credentials to a service, whether via a mobile or other device, is often a tedious experience for a user. Unfortunately, to make authentication easier for themselves, users will often engage in practices such as password re-use, and/or the selection of poor quality passwords, which render their credentials less secure against attacks. Accordingly, improvements in authentication techniques would be desirable. Further, it would be desirable for such improvements to be widely deployable, including on existing/legacy systems.

BRIEF DESCRIPTION OF THE DRAWINGS

Various embodiments of the invention are disclosed in the following detailed description and the accompanying drawings.

FIG. **1** illustrates an embodiment of an environment in which authentication translation is provided.

FIG. **2** illustrates an embodiment of credential information stored on a device.

FIG. **3** illustrates an embodiment of a device with secure storage.

FIG. **4** illustrates an example of a renegotiation.

FIG. **5** illustrates an embodiment of a process for performing authentication translation.

FIG. **6** illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator.

FIG. **7** illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator.

FIG. **8** shows the structure of an example of a cache cookie used in some embodiments.

DETAILED DESCRIPTION

The invention can be implemented in numerous ways, including as a process; an apparatus; a system; a composition of matter; a computer program product embodied on a computer readable storage medium; and/or a processor, such as a processor configured to execute instructions stored on and/or provided by a memory coupled to the processor. In this specification, these implementations, or any other form that the invention may take, may be referred to as techniques. In general, the order of the steps of disclosed processes may be altered within the scope of the invention. Unless stated otherwise, a component such as a processor or

a memory described as being configured to perform a task may be implemented as a general component that is temporarily configured to perform the task at a given time or a specific component that is manufactured to perform the task. As used herein, the term 'processor' refers to one or more devices, circuits, and/or processing cores configured to process data, such as computer program instructions.

A detailed description of one or more embodiments of the invention is provided below along with accompanying figures that illustrate the principles of the invention. The invention is described in connection with such embodiments, but the invention is not limited to any embodiment. The scope of the invention is limited only by the claims and the invention encompasses numerous alternatives, modifications and equivalents. Numerous specific details are set forth in the following description in order to provide a thorough understanding of the invention. These details are provided for the purpose of example and the invention may be practiced according to the claims without some or all of these specific details. For the purpose of clarity, technical material that is known in the technical fields related to the invention has not been described in detail so that the invention is not unnecessarily obscured.

FIG. **1** illustrates an embodiment of an environment in which authentication translation is provided. In the example shown, a variety of client devices **102-108** connect, via one or more networks (represented as a single network cloud **110**) to a variety of services **120-124** (also referred to herein as sites **120-124**). In particular, client device **102** is a notebook computer owned by a user hereinafter referred to as Alice. Notebook **102** includes a camera, a microphone, and a fingerprint sensor. Client device **104** is a smartphone, also owned by Alice. Client device **104** includes a camera. Client device **106** is a tablet owned by Bob, and sometimes used by Bob's son Charlie. Client device **106** includes a camera and a fingerprint sensor. Client device **108** is a kiosk located in the lobby of a hotel. Kiosk **108** includes a camera and a microphone. The techniques described herein can be used with or adapted to be used with other devices, as applicable. For example, the techniques can be used in conjunction with gaming systems, with peripheral devices such as mice, and with embedded devices, such as door locks.

Service **120** is a social networking site. Service **122** is a website of a bank. Service **124** is the online store of a boutique camera retailer. Each of services **120-124** requires a username and password (and/or a cookie) from a user prior to giving that user access to protected content and/or other features. As will be described in more detail below, using the techniques described herein, users need not type such usernames and passwords into their devices whenever required by a service. Instead, users can authenticate themselves to an "authentication translator" via an appropriate technique, and the authentication translator will provide the appropriate credentials to the implicated service on the user's behalf. Also as will be described in more detail below, authentication translators can be located in a variety of places within an environment. For example, notebook computer **102** includes an authentication translator module **132** that provides authentication translation services. The other devices **104-108** can also include (but need not include) their own respective authentication translator modules. The owner of bank website **122** also operates an authentication translator **134** associated with the bank. Finally, authentication translator **136** provides authentication translation services to a variety of businesses, including online camera retailer **124**.

FIG. **2** illustrates an embodiment of credential information stored on a device. In particular, device **200** stores three user profiles **202-206**, each of which contains a username and one or more templates (e.g., template **210**) associated with the user. In various embodiments, a template is a collection of biometric features. Using fingerprints as an example type of biometric, a corresponding template includes a collection of patterns, minutia, and/or other features that can be matched against to determine if a person's fingerprint matches the fingerprint of the registered user (i.e., the owner of a given user profile). A representation of a single fingerprint may be included in multiple templates (e.g., in different resolutions, in accordance with different protocols, as captured during warm vs. cold conditions, and/or by itself or in combination with multiple fingerprints) . When other biometrics are employed (e.g., facial recognition, voiceprint, or retina scan technology), features appropriate to those types of biometrics are included in the template. Other types of features can also be included in templates. As one example, a user's typing speed and/or accuracy can be measured by a device, such as device **102**, and used to distinguish between multiple users of a device. For example, suppose Alice types at 100 words per minute and rarely makes mistakes. A representation of this information can be stored in template **212**. Also suppose Alice's niece, who sometimes uses Alice's laptop computer when visiting Alice types at 20 words per minute and makes many mistakes. In some embodiments, the fact that a user was recently (e.g., within the last 5 minutes) typing on laptop **102** at 90 words per minute is evidence of a match against template **212**. In this case, the typing speed of 90 words per minute is similar enough to Alice's typical behavior, it is considered a match. Various policies can be included in a profile that govern how matches are to be performed. For example, policies can specify thresholds/tolerances for what constitutes a match, and can specify that different levels of matches can result in different levels of access to different resources.

A profile is associated with a vault (e.g., vault **220**). The vault, in turn, contains triples specifying a service provider/domain, a username, and a credential. The vault can also contain other sensitive user information, such as account numbers, address/phone number information, and health care data. The credential for a service provider/domain can be a password (e.g., for legacy servers), and can also take alternate forms (e.g., a cryptographic key for service providers supporting stronger authentication methods).

In some embodiments, profiles, templates, and vaults (collectively "authentication information") are stored entirely in an unprotected storage area, and are stored in the clear. In other embodiments, secure storage techniques are used to secure at least a portion of the authentication information.

One example of a device with secure storage is illustrated in FIG. **3**. In the example shown, a mobile phone **300** includes a large and insecure storage **302** attached to a fast processor **304**, and a smaller but secure storage **306** attached to a dedicated processor **308** and a sensor **310** (e.g., a camera or a fingerprint reader). Users (and applications) can read from and write to the insecure storage area. However, users cannot access the secure storage area, and the fast processor can only communicate with the dedicated processor/sensor via a restricted API. As another example, a unique decryption key associated with a given vault can be stored in a profile. The vault is an encrypted and authenticated container that can be stored on insecure storage, e.g., on the device, and also backed up (e.g., to a cloud storage service **140** or to an alternate form of external storage). As needed, authentication information or portions thereof can be loaded into secure storage and decrypted. For example, one can use AES to encrypt the files one by one, using a key stored on the secured storage. A message authentication technique, such as HMAC, can be used for authenticating the encrypted files to provide tamper prevention. Profiles and vaults can be updated while in secure storage; if this occurs, they are encrypted and MACed before being written back to the insecure storage, which may in turn propagate them to external backup storage. In yet other embodiments, profiles and vaults are stored entirely in secure storage, in plaintext, which allows them to be both read and written—and in particular, searched.

Example Transaction Types

A variety of transaction types can take place in the environment shown in FIG. **1**, examples of which are discussed in this section.

Initial Registration

In order to begin using the techniques described herein, users perform some form of initial registration. As one example, suppose Alice launches an enrollment program installed on laptop **102**. She uses the program to capture various biometric information (e.g., fingerprints, photographs of her face, etc.). A user profile is created for Alice, and the biometric information captured about her is encoded into a plurality of templates, such as templates **210** and **214**. In some embodiments, Alice is also explicitly asked to supply credential information for services she would like to use, such as by providing the domain name of social networking site **120**, along with her username and password for site **120**. In other embodiments, domain/username/credential information is at least passively captured on Alice's behalf and included in one or more vaults such as vault **220**. Credential information can also be important from a browser password manager already in use by Alice or other appropriate source. In some embodiments, Alice also registers with cloud storage service **140**, which will allow her to back up her authentication information and to synchronize it across her devices (e.g., **102** and **104**), as described in more detail below.

Other registration approaches can also be used. For example, registration can be integrated into the experience the first time a device is used. Thus, when Bob first turns on tablet **106**, he may be prompted to take a picture of his face (with a profile/templates being created in response). Similarly, the first time Charlie uses tablet **106**, the techniques described herein can be used to determine that Charlie does not yet have a profile (e.g., because none of the templates already present on tablet **106** match his biometrics) and Charlie can be prompted to enroll as a second user of the device.

Authentication

Suppose Alice wishes to authenticate to banking website **122**. Using a fingerprint reader incorporated into her laptop, she performs a fingerprint scan, which causes her biometric features to be extracted and compared to any stored templates residing on her computer. If a match is found, an associated decryption key is selected, and the associated vault is loaded and decrypted. The vault is scanned for an entry that matches the selected service provider (i.e., website **122**). If a matching entry is found, the associated domain, username, and site credential are extracted from the vault. In some embodiments, the validity of the domain name mapping is verified at this point to harden the system against domain name poisoning. Next, a secure connection is established between Alice's computer and the service provider,

and Alice is authenticated. For service providers supporting strong user authentication, mutual SSL can be used, for example. A variety of policies can be involved when performing matching. For example, to access certain domains, Alice's print may need only match template **210**. To access other domains, Alice may need to match multiple templates (e.g., both **210** and **214**). As another example, in order to access social networking site **120**, Alice may merely need to be sitting in front of her computer, which has an integrated webcam. Even in relatively low light conditions, a match can be performed against Alice's face and features stored in a template. However, in order to access bank website **122**, Alice may need a high quality photograph (i.e., requiring her to turn on a bright light) and may need to demonstrate liveness (e.g., by blinking or turning her head). As yet another example, other contextual information can be included in policies. For example, if Alice's IP address indicates she is in a country that she is not usually in, she may be required to match multiple templates (or match a template with more/better quality features) in order to access retailer **124**, as distinguished from when her IP address indicates she is at home.

In some embodiments, the biometric sensor used by a user may be a peripheral device (e.g., a mouse with an integrated fingerprint scanner that is connected to the user's primary device via USB). In such scenarios, the peripheral device may be responsible for storing at least a portion of authentication information and may perform at least some of the authentication tasks previously described as having been performed by Alice's computer. For example, instead of processors **304** and **308**, and storages **302** and **306** being collocated on a single device (e.g., laptop **102**), processor **304** and storage **302** may be present on a primary device, and processor **308** and storage **306** may be present on a peripheral device (e.g., that also includes a sensor, such as a fingerprint reader).

In such scenarios, once Alice's login to banking website **122** is successfully completed, the secure session can be handed over from the peripheral device to the primary device, in a way that does not allow the primary device retroactive access to the plaintext data of the transcripts exchanged between the peripheral device and the service provider. One way this can be accomplished is by renegotiating SSL keys between the peripheral device and the website, after which the newly negotiated key can be handed off from the peripheral device to the primary device. This avoids retroactive credential capture in a setting where the device is infected by malware.

An example of renegotiation is depicted in FIG. **4**. Specifically, after a user has successfully authenticated to a fingerprint reader, a login is performed to a service provider. Using the primary device (**404**) as a proxy, the peripheral fingerprint reader **402** negotiates a first SSL connection (**408**) with a service provider **406**, over which credentials are exchanged. The proxy then renegotiates SSL (**410**), which replaces the old key with a new one. The new key is disclosed to the device, which then seamlessly takes over the connection with the service provider and performs the transaction protected by the authentication. The credentials exchanged during the first SSL connection cannot be accessed by device **404**, since the key of the renegotiated session is independent of the key of the original session; this provides protection against malware residing on the device. Renegotiation can be used when the primary device **404** is believed to be in a safe state when performing the negotiation of the SSL connection, but it is not known whether it is in a safe state during the transaction protected by the authentication. Renegotiation can also be used when a secure component of the primary device **404** performs the negotiation of the SSL connection and another and potentially insecure component of the primary device **404** is involved in the transaction protected by the authentication.

FIG. **5** illustrates an embodiment of a process for performing authentication translation. The process begins at **502** when a request to access a resource is received, as is an authentication input. One example of the processing performed at **502** is as follows. Suppose Alice wishes to sign into social networking website **120**. She directs a web browser application installed on client **102** to the social networking website. Authentication translator module **132** recognizes, from the context of Alice's actions (e.g., that she is attempting to access site **120** with her browser) that she would like to access a particular resource. Authentication translator module **132** prompts Alice (e.g., by a popup message or via a sound) to provide biometric information (e.g., to use the integrated fingerprint reader on her laptop). In some embodiments, the translator module does not prompt Alice, for example, because Alice has been trained to provide biometric information automatically when attempting to access certain resources. In yet other embodiments, the translator module only prompts Alice if she fails to provide acceptable biometric information within a timeout period (e.g., 30 seconds).

Module **132** compares Alice's supplied biometric data to the templates stored on her computer. If a suitable match is found, and if an entry for site **120** is present in the applicable vault, at **504**, a previously stored credential associated with the resource is accessed. In particular, the username and password for the website, as stored in a vault, such as vault **220**, are retrieved from the vault.

Finally, at **506**, the credential is provided to the resource. For example, Alice's username and password for site **120** are transmitted to site **120** at **506**. The credential can be transmitted directly (e.g., by the module or by Alice's computer) and can also be supplied indirectly (e.g., through the use of one or more proxies, routers, or other intermediaries, as applicable).

Other devices can also make use of process **500** or portions thereof. For example, when Alice launches a banking application on phone **104**, implicit in her opening that application is her desire to access the resources of website **134**. The application can take Alice's picture and compare it to stored templates/vault information. If an appropriate match is found, a credential can be retrieved from the vault on her phone (or, e.g., retrieved from cloud storage service **140**) and provided to website **134**.

As another example, suppose Charlie is using tablet **106** and attempts to visit site **120**, whether via a dedicated application or via a web browser application installed on the tablet. Charlie's photograph is taken, and then compared against the profiles stored on tablet **106** (e.g., both Bob and Charlie's profiles). When a determination is made that Charlie's photograph matches a template stored in his stored profile (and not, e.g., Bob's), Charlie's credentials for site **120** are retrieved from a vault and transmitted by an authentication translator module residing on client **106**.

As yet another example, kiosk **108** can be configured to provide certain local resources (e.g., by displaying a company directory or floor plan on demand) when users speak certain requests into a microphone. Enrolled users (e.g., with stored voiceprint or facial recognition features) can be granted access to additional/otherwise restricted services in accordance with the techniques described herein and process **500**.

New Device

In some embodiments, to register a new device, a user provides an identifier, such as a username or an account number to the device. The new device connects to an external storage (such as cloud storage **140**), provides the user identifier and credential, and downloads the user's templates/vaults from the service. In some embodiments, the templates/vaults are encrypted. Once downloaded, the template is decrypted and stored in a secure storage area, while the still encrypted vault can be stored in insecure storage. The decryption key can be generated from information the user has/knows, or from biometric data—such as features extracted from fingerprinting of all ten fingers. In some embodiments, more arduous fingerprinting is required for the setup of a new device than for regular authentication to avoid that a new device gets registered by a user thinking she is merely authenticating—or worse still, simply touching the device. Moreover, it translates into higher entropy of the decryption keys.

Backup Authentication

Backup authentication allows a user, such as Alice, to access resources in the event she is unable to or unwilling to interact with a particular biometric sensor. As one example, instead of having a single template associated with her profile, Alice can have multiple templates associated with it, e.g., where the first template includes fingerprint features and the second template includes voice biometric, facial recognition, or iris detection features. As a second example, where the service Alice is connecting to is a legacy website (i.e., one that users authenticate to using usernames and passwords), such a service would allow the use of passwords and password reset mechanisms by Alice without requiring Alice to use a fingerprint reader.

In various embodiments, environment **100** supports the ability of users (e.g., under duress) to release the contents of their vaults. For example, if Alice was physically threatened with the loss of a finger by a criminal, Alice could instead release the contents of her vault(s)—the ultimate goal of the criminal. As one example, in the event Alice supplies all 10 fingerprints to the sensor, provides a special password, or supplies a fingerprint and a second identifier, a cleartext version of her vault(s) could be made available.

Access Policies

In various embodiments, cloud storage service **140** is configured to accept backups from multiple devices associated with a single account, and synchronize the updates so that all devices get automatically refreshed. For example, Alice's laptop **102** and phone **104** could both communicate with cloud storage service **140** which would keep their authentication information synchronized. Refreshes can also be made in accordance with user-configured restrictions. For example, Alice's employer could prevent privileged employer data from being stored on shared personal devices, or on any device that was not issued by the employer. As another example, arbitrary policies can be defined regarding the access to and synchronization of software and data, and to tie a license or access rights to a person (and her fingerprint) rather than to a device. As yet another example, in some embodiments (e.g., where a device is made publicly available or otherwise shared by many users), no or a reduced amount of authentication information resides on a device, and at least a portion of authentication information is always retrieved from cloud storage service **140**.

Remote Wiping

Remote wiping of a user's authentication information (e.g., templates) can be used both to "unshare" previously shared devices (e.g., where Bob and Charlie both have user profiles on their shared tablet **106**), and to avoid that criminals with physical component access to lost devices gain access to templates and vault contents. In some embodiments, polices such as ones where a template self-wipes if it is not matched within a particular duration of time are supported. Since user data can be frequently backed up to the cloud storage, and recovered from this using the new device registration process, inconvenience to the user will be minimized.

Legacy Server Support

New authentication schemes typically require changes to a significant codebase residing with service providers. If the code is well written and documented, such changes may be relatively simple. Commonly, though, this may not be so. The engineers who originally wrote the code of relevance may have long since left the company; the code they left behind may be poorly documented—if documented at all. In severe cases, the legacy code may have been written in an outdated programming language or written in a way that does not follow guidelines for good code. This makes updates to the codebase impractical or virtually impossible in many common cases. Even if none of these challenges complicate the desired modifications, it is commonly a great bureaucratic burden to obtain permission to make modifications (e.g., to store an extra field in a backend database), because every affected part of the organization may need to review the request.

As will be described in the following section, the technologies described herein can be used in conjunction with legacy servers (e.g., existing servers that rely on usernames and passwords to authenticate users), and in particular, can be used without requiring modification to such legacy servers.

Cookies

Cookies are commonly used by legacy servers for user authentication. Unfortunately, cookies have several problems. For one thing, they are sometimes deleted—whether explicitly/intentionally by the end user or by the user's software. In addition, cookies are commonly stolen. Approaches such as cache cookies and identification using user agents can be more resistant to these problems, however, they have their own problems. For example, their use requires new code and new fields in the credential database stored by the server.

In some embodiments, authentication translators, such as translators **134** and **136** (also referred to herein as proxies) provide authentication translation services on behalf of associated services. Translators **134** and **136** are illustrated as single logical devices in FIG. **1**. In some embodiments, the translators comprise standard commercially available server hardware (e.g., a multi-core processor, 4+ Gigabytes of RAM, and one or more Gigabit network interface adapters) and run typical server-class operating systems (e.g., Linux). Translators **134** and **136** can also be implemented using a scalable, elastic architecture and may comprise several distributed components, including components provided by one or more third parties. For example, translators **134** and **136** may store user credential information or may task cloud storage service **140** with storing at least a portion of that information.

In the case of authentication translator **134**, service is provided with respect to bank website **122** only. Authentication translator **134** is positioned between a legacy web server (**122**) and the Internet (**110**)—and therefore between the legacy server and any client devices. Authentication translator **134** is configured to translate traffic between the legacy server and client devices so that the client devices

(and respective users) perceive the new authentication mechanism, while the legacy server remains unchanged. Authentication translator **136** works similarly, but it provides authentication translation as a third party service to multiple providers, an example of which is online camera retailer **124**.

Authentication translators **134** and **136** can also perform process **500**. For example, when a device transmits a request to access website **122**, the request is intercepted by translator **134**, as is cookie/user agent information. The received information can be used to determine a username/password associated with the device, and that information can be passed by translator **134** to website **122** on behalf of the device.

FIG. **6** illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator. The translator (referred to in the figure as a "proxy") fails to identify the client, and passes on the request to the legacy server. The legacy server responds to the request and sets a cookie. The proxy passes on the response, including the cookie and also a cache cookie. The proxy stores the information about both these types of identifiers, along with the user agent of the client device. This triplet of information is also referred to herein as an identifier.

FIG. **7** illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator. In this scenario, the device's request is accompanied by some form of identifying information. The proxy uses this information to identify the associated cookie and passes this along to the legacy server, along with the request. In some embodiments, additional processing is also involved. For example, reading a cache cookie may require user interaction. Moreover, if not all of the identifying information is present in the request, the proxy can be configured to set the missing information again by sending a corresponding request to the client device.

The translation of cache cookies and user agent information to cookies involves a two-way translation. First, when the legacy server sets a cookie, the proxy will set the two types of cookies—both an HTML cookie and a cache cookie—and then create a new record in which the two cookies are stored, along with the user agent information of the client device. The user agent information can include quite a bit of data associated with a browser—such as the browser type and version, the clock skew, and the fonts that are installed. While each of these pieces of information only contributes a small amount of entropy, the collection of items can be sufficient to identify the device in most cases. Moreover, while some of these types of data may change over time—in fact, all of them may—they do not typically change, and when one or two of them do, the others typically do not. When the client device is used to visit a site controlled by the legacy server, the cookie, cache cookie and user agent information are read (if available), the record identified, and the request translated and sent to the legacy server. When a legacy server requests that the user password is updated (e.g., as part of an annual or other periodic requirement), the transmission of this request to the user can be suppressed—in which case the database of the proxy is updated to create the illusion of an update. The user can be involved in authentication as needed, e.g., if, in addition to supplying a credential, a user must also solve a CAPTCHA, the CAPTCHA can be displayed to the user (with the user's credentials being handled in accordance with the techniques described herein).

FIG. **8** shows the structure of an example of a cache cookie used in some embodiments. Cache cookies can be associated with a particular webpage, just like an HTML cookie can. In the example shown in FIG. **8**, the proxy wishes to associate a page "calling page.html" with a cache cookie. It embeds a request for a second object, "samename.html" in callingpage.html for every visitor. However, as the cache cookie is set for one visitor, a customized samename.html is served to this visitor. The page samename.html refers to an object with a different name for each user; that object is referred to as "uniquename.jpg." The cache cookie is set by embedding the request for samename.html in callingpage.html. The client browser attempts to render this, causing a request for samename.html from the server. The server configures samename.html to refer to a uniquely named file uniquename.jpg, and serves samename.html to the client. For the client browser to render samename.html, it requests the file uniquename.jpg, which is intentionally not served. That concludes the setting of a cache cookie. As a user returns to the page callingname.html, the browser again attempts to render the entire page, which causes it to load the object samename.html from its cache. As that is rendered, the client browser requests uniquename.jpg, which is not in its cache (since it was not served previously). The server still does not serve it, but takes note of the name of the file being requested, as it identifies the client device. Note that samename.html can be displayed in a zero-sized iframe, which makes the end user unaware of it being rendered.

A cache cookie is an implementation of the typical cookie functionality that uses the client device's browser cache. Unlike user agents, it does not change over time, and like standard HTML cookies, it cannot be read by a party other than that which set it. However, like HTML cookies, it could be deleted—by the user clearing his or her browser cache. Cache cookies are not automatically transmitted with GET requests, unless the cache elements are embedded in the referring pages. This adds a potential round of communication in some settings. By relying on user agent information, cache cookies, and HTML cookies to identify the client device, it is much more likely that a machine will be recognized than if only HTML cookies are used.

Although the foregoing embodiments have been described in some detail for purposes of clarity of understanding, the invention is not limited to the details provided. There are many alternative ways of implementing the invention. The disclosed embodiments are illustrative and not restrictive.

What is claimed is:

1. A system, comprising:

one or more processors configured to:

receive, at an authentication translator at a device, an authentication input and an indication of a request, by a user of the device, to access a resource external to the device;

in response to determining, by the authentication translator at the device, a match using the received authentication input, obtain access to a record stored on the device, wherein the record is associated at least with the external resource;

retrieve, from the stored record accessed at least in part by determining, by the authentication translator at the device, the match using the received authentication input, a previously stored credential associated with the external resource, wherein the previously stored credential is different from the received authentication input, and wherein the previously

stored credential retrieved from the record stored on the device comprises at least one of a cryptographic key and a cookie;

establish a connection between the device and the external resource to which the user requested access; and

facilitate a login of the user to the external resource at least in part by transmitting, on behalf of the user, from the device and via the established connection, output based at least in part on the at least one of the cryptographic key and the cookie retrieved from the record accessed at least in part by determining, by the authentication translator at the device, the matching using the received authentication input, wherein the user of the device is logged into the external resource based at least in part on the output transmitted from the device on behalf of the user; and

a memory coupled to the one or more processors and configured to provide the one or more processors with instructions.

2. The system of claim **1** wherein the authentication input is received in response to a user-supplied biometric input matching a template.

3. The system of claim **1** wherein the device further includes a biometric input component.

4. The system of claim **1** wherein the one or more processors are further configured to facilitate a session renegotiation.

5. The system of claim **1** wherein the one or more processors are further configured to receive, from a remote system, an encrypted container that includes at least one template containing biometric features.

6. The system of claim **1** wherein the authentication input comprises user agent information.

7. The system of claim **1** wherein the one or more processors are further configured to receive a password change request from the external resource and wherein the one or more processors are configured to update the previously stored credential without user input.

8. The system of claim **1** wherein access to another stored record associated with another resource is obtained using a different authentication input.

9. The system of claim **1** wherein the authentication input comprises a biometric input, and wherein access to the record stored on the device including the previously stored credential associated with the external resource is obtained at least in part by determining a match using the biometric input.

10. The system of claim **1** wherein the authentication translator at the device is associated with a plurality of records stored on the device, wherein the plurality of records comprise previously stored credentials associated with a plurality of external resources, and wherein the accessed record is included in the plurality of records.

11. The system of claim **1** further comprising a secure storage and an insecure storage, wherein the record is stored in the insecure storage, and wherein obtaining access to the record comprises loading the record to the secure storage.

12. The system of claim **11** wherein the record stored in the insecure storage is encrypted, and wherein obtaining access to the record further comprises decrypting the record loaded to the secure storage.

13. The system of claim **4** wherein the established connection via which the output was transmitted is associated with a first session, and wherein facilitating the session renegotiation comprises replacing a first key associated with the first session with a second key associated with a renegotiated session.

14. The system of claim **1** wherein the record stored on the device further comprises at least one of a username, a password, an account number, address information, phone information, and health care data.

15. The system of claim **1** wherein the match is determined at least in part by using at least one of a password and a template, the template comprising at least one of a fingerprint feature, a voice biometric feature, a facial recognition feature, an iris detection feature, and a retina scan feature.

16. The system of claim **1**, wherein the one or more processors are further configured to:

encrypt and authenticate at least a portion of the record; and

perform a backup of the encrypted and authenticated record to a remote storage entity, wherein the remote storage entity is configured to synchronize records between at least two devices associated with the user.

17. A method, comprising:

receiving, at an authentication translator at a device, an authentication input and an indication of a request, by a user of the device, to access a resource external to the device;

in response to determining, by the authentication translator at the device, a match using the received authentication input, obtaining access to a record stored on the device, wherein the record is associated at least with the external resource;

retrieving, from the stored record accessed at least in part by determining, by the authentication translator at the device, the match using the received authentication input, a previously stored credential associated with the external resource, wherein the previously stored credential is different from the received authentication input, and wherein the previously stored credential retrieved from the record stored on the device comprises at least one of a cryptographic key and a cookie;

establishing, using one or more processors, a connection between the device and the external resource to which the user requested access; and

facilitating a login of the user to the external resource at least in part by transmitting, on behalf of the user, from the device and via the established connection, output based at least in part on the at least one of the cryptographic key and the cookie retrieved from the record accessed at least in part by determining, by the authentication translator at the device, the match using the received authentication input, wherein the user of the device is logged into the external resource based at least in part on the output transmitted from the device on behalf of the user.

18. The method of claim **17** further comprising facilitating a session renegotiation.

19. The method of claim **17** further comprising receiving, from a remote system, an encrypted container that includes at least one template containing biometric features.

20. The method of claim **17** further comprising receiving a password change request from the external resource and updating the previously stored credential without user input.

21. The method of claim **17** wherein access to another stored record associated with another resource is obtained using a different authentication input.

22. The method of claim **17** wherein the authentication input is received in response to a user-supplied biometric input matching a template.

23. The method of claim **17** wherein the device further includes a biometric input component.

24. The method of claim **17** wherein the authentication input comprises user agent information.

25. The method of claim **17** wherein the authentication input comprises a biometric input, and wherein access to the record stored on the device including the previously stored credential associated with the external resource is obtained at least in part by determining a match using the biometric input.

26. The method of claim **17** wherein the authentication translator at the device is associated with a plurality of records stored on the device, wherein the plurality of records comprise previously stored credentials associated with a plurality of external resources, and wherein the accessed record is included in the plurality of records.

27. The method of claim **17** wherein the record is stored in an insecure storage, and wherein obtaining access to the record comprises loading the record to a secure storage.

28. The method of claim **27** wherein the record stored in the insecure storage is encrypted, and wherein obtaining access to the record further comprises decrypting the record loaded to the secure storage.

29. The method of claim **18** wherein the established connection via which the output was transmitted is associated with a first session, and wherein facilitating the session renegotiation comprises replacing a first key associated with the first session with a second key associated with a renegotiated session.

30. The method of claim **17** wherein the record stored on the device further comprises at least one of a username, a password, an account number, address information, phone information, and health care data.

31. The method of claim **17** wherein the match is determined at least in part by using at least one of a password and a template, the template comprising at least one of a fingerprint feature, a voice biometric feature, a facial recognition feature, an iris detection feature, and a retina scan feature.

32. The method of claim **17**, further comprising:

encrypt and authenticate at least a portion of the record; and

perform a backup of the encrypted and authenticated record to a remote storage entity, wherein the remote storage entity is configured to synchronize records between at least two devices associated with the user.

* * * * *

Electronic Acknowledgement Receipt

EFS ID:	24902342
Application Number:	15042636
International Application Number:	
Confirmation Number:	1040
Title of Invention:	AUTHENTICATION TRANSLATION
First Named Inventor/Applicant Name:	Bjorn Markus Jakobsson
Customer Number:	21912
Filer:	Yeu-Ting George Cheng/Monique Huang
Filer Authorized By:	Yeu-Ting George Cheng
Attorney Docket Number:	MJAKP008C1
Receipt Date:	12-FEB-2016
Filing Date:	
Time Stamp:	14:38:17
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Application Data Sheet	MJAKP008C1_ADS.pdf	1793174 [illegible]	no	8

Warnings:

Information:

2		MJAKP008C1_APP.pdf	148473 d1e1dfb9cb5fa6101341 56db9f90a2397f015a20	yes	19

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Specification	1	17
	Claims	18	18
	Abstract	19	19

Warnings:

Information:

3	Drawings-only black and white line drawings	MJAKP008C1_APP_Figures.pdf	112090 83487f17ad1fc2e79014265a447a54f43bdcc227	no	8

Warnings:

Information:

Total Files Size (in bytes):	2053737

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Attorney Docket No. MJAKP008C1

APPLICATION FOR UNITED STATES PATENT

AUTHENTICATION TRANSLATION

By Inventor:

Bjorn Markus Jakobsson
Portola Valley, CA
A Citizen of Sweden

Assignee: RightQuestion, LLC

VAN PELT, YI & JAMES LLP
10050 N. Foothill Blvd., Suite 200
Cupertino, CA 95014
Telephone (408) 973-2585

AUTHENTICATION TRANSLATION

CROSS REFERENCE TO OTHER APPLICATIONS

[0001] This application is a continuation of co-pending U.S. Patent Application No. 13/706,254, entitled AUTHENTICATION TRANSLATION filed December 5, 2012 which is incorporated herein by reference for all purposes. U.S. Patent Application No. 13/706,254 claims priority to U.S. Provisional Patent Application No. 61/569,112 entitled BACKWARDS COMPATIBLE ROBUST COOKIES filed December 9, 2011, and also claims priority to U.S. Provisional Patent Application No. 61/587,387 entitled BIOMETRICS-SUPPORTED SECURE AUTHENTICATION SYSTEM filed January 17, 2012, both of which are incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

[0002] Providing credentials to a service, whether via a mobile or other device, is often a tedious experience for a user. Unfortunately, to make authentication easier for themselves, users will often engage in practices such as password re-use, and/or the selection of poor quality passwords, which render their credentials less secure against attacks. Accordingly, improvements in authentication techniques would be desirable. Further, it would be desirable for such improvements to be widely deployable, including on existing/legacy systems.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] Various embodiments of the invention are disclosed in the following detailed description and the accompanying drawings.

[0004] Figure 1 illustrates an embodiment of an environment in which authentication translation is provided.

[0005] Figure 2 illustrates an embodiment of credential information stored on a device.

[0006] Figure 3 illustrates an embodiment of a device with secure storage.

[0007] Figure 4 illustrates an example of a renegotiation.

[0008] Figure 5 illustrates an embodiment of a process for performing authentication translation.

[0009] Figure 6 illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator.

[0010] Figure 7 illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator.

[0011] Figure 8 shows the structure of an example of a cache cookie used in some embodiments.

DETAILED DESCRIPTION

[0012] The invention can be implemented in numerous ways, including as a process; an apparatus; a system; a composition of matter; a computer program product embodied on a computer readable storage medium; and/or a processor, such as a processor configured to execute instructions stored on and/or provided by a memory coupled to the processor. In this specification, these implementations, or any other form that the invention may take, may be referred to as techniques. In general, the order of the steps of disclosed processes may be altered within the scope of the invention. Unless stated otherwise, a component such as a processor or a memory described as being configured to perform a task may be implemented as a general component that is temporarily configured to perform the task at a given time or a specific component that is manufactured to perform the task. As used herein, the term 'processor' refers to one or more devices, circuits, and/or processing cores configured to process data, such as computer program instructions.

[0013] A detailed description of one or more embodiments of the invention is provided below along with accompanying figures that illustrate the principles of the invention. The invention is described in connection with such embodiments, but the invention is not limited to any embodiment. The scope of the invention is limited only by the claims and the invention encompasses numerous alternatives, modifications and equivalents. Numerous specific details are set forth in the following description in order to provide a thorough understanding of the invention. These details are provided for the purpose of example and the invention may be practiced according to the claims without some or all of these specific details. For the purpose of clarity, technical material that is known in the technical fields related to the invention has not been described in detail so that the invention is not unnecessarily obscured.

[0014] Figure 1 illustrates an embodiment of an environment in which authentication translation is provided. In the example shown, a variety of client devices 102-108 connect, via one or more networks (represented as a single network cloud 110) to a variety of services 120-124 (also referred to herein as sites 120-124). In particular, client device 102 is a notebook computer owned by a user hereinafter referred to as Alice. Notebook 102 includes a camera, a microphone, and a fingerprint sensor. Client device 104 is a smartphone, also owned by Alice.

Client device 104 includes a camera. Client device 106 is a tablet owned by Bob, and sometimes used by Bob's son Charlie. Client device 106 includes a camera and a fingerprint sensor. Client device 108 is a kiosk located in the lobby of a hotel. Kiosk 108 includes a camera and a microphone. The techniques described herein can be used with or adapted to be used with other devices, as applicable. For example, the techniques can be used in conjunction with gaming systems, with peripheral devices such as mice, and with embedded devices, such as door locks.

[0015] Service 120 is a social networking site. Service 122 is a website of a bank. Service 124 is the online store of a boutique camera retailer. Each of services 120-124 requires a username and password (and/or a cookie) from a user prior to giving that user access to protected content and/or other features. As will be described in more detail below, using the techniques described herein, users need not type such usernames and passwords into their devices whenever required by a service. Instead, users can authenticate themselves to an "authentication translator" via an appropriate technique, and the authentication translator will provide the appropriate credentials to the implicated service on the user's behalf. Also as will be described in more detail below, authentication translators can be located in a variety of places within an environment. For example, notebook computer 102 includes an authentication translator module 132 that provides authentication translation services. The other devices 104-108 can also include (but need not include) their own respective authentication translator modules. The owner of bank website 122 also operates an authentication translator 134 associated with the bank. Finally, authentication translator 136 provides authentication translation services to a variety of businesses, including online camera retailer 124.

[0016] Figure 2 illustrates an embodiment of credential information stored on a device. In particular, device 200 stores three user profiles 202-206, each of which contains a username and one or more templates (e.g., template 210) associated with the user. In various embodiments, a template is a collection of biometric features. Using fingerprints as an example type of biometric, a corresponding template includes a collection of patterns, minutia, and/or other features that can be matched against to determine if a person's fingerprint matches the fingerprint of the registered user (i.e., the owner of a given user profile). A representation of a single fingerprint may be included in multiple templates (e.g., in different resolutions, in accordance with different protocols, as captured during warm vs. cold conditions, and/or by itself

or in combination with multiple fingerprints). When other biometrics are employed (e.g., facial recognition, voiceprint, or retina scan technology), features appropriate to those types of biometrics are included in the template. Other types of features can also be included in templates. As one example, a user's typing speed and/or accuracy can be measured by a device, such as device 102, and used to distinguish between multiple users of a device. For example, suppose Alice types at 100 words per minute and rarely makes mistakes. A representation of this information can be stored in template 212. Also suppose Alice's niece, who sometimes uses Alice's laptop computer when visiting Alice types at 20 words per minute and makes many mistakes. In some embodiments, the fact that a user was recently (e.g., within the last 5 minutes) typing on laptop 102 at 90 words per minute is evidence of a match against template 212. In this case, the typing speed of 90 words per minute is similar enough to Alice's typical behavior, it is considered a match. Various policies can be included in a profile that govern how matches are to be performed. For example, policies can specify thresholds/tolerances for what constitutes a match, and can specify that different levels of matches can result in different levels of access to different resources.

[0017] A profile is associated with a vault (e.g., vault 220). The vault, in turn, contains triples specifying a service provider/domain, a username, and a credential. The vault can also contain other sensitive user information, such as account numbers, address/phone number information, and health care data. The credential for a service provider/domain can be a password (e.g., for legacy servers), and can also take alternate forms (e.g., a cryptographic key for service providers supporting stronger authentication methods).

[0018] In some embodiments, profiles, templates, and vaults (collectively "authentication information") are stored entirely in an unprotected storage area, and are stored in the clear. In other embodiments, secure storage techniques are used to secure at least a portion of the authentication information.

[0019] One example of a device with secure storage is illustrated in Figure 3. In the example shown, a mobile phone 300 includes a large and insecure storage 302 attached to a fast processor 304, and a smaller but secure storage 306 attached to a dedicated processor 308 and a sensor 310 (e.g., a camera or a fingerprint reader). Users (and applications) can read from and

write to the insecure storage area. However, users cannot access the secure storage area, and the fast processor can only communicate with the dedicated processor/sensor via a restricted API. As another example, a unique decryption key associated with a given vault can be stored in a profile. The vault is an encrypted and authenticated container that can be stored on insecure storage, e.g., on the device, and also backed up (e.g., to a cloud storage service 140 or to an alternate form of external storage). As needed, authentication information or portions thereof can be loaded into secure storage and decrypted. For example, one can use AES to encrypt the files one by one, using a key stored on the secured storage. A message authentication technique, such as HMAC, can be used for authenticating the encrypted files to provide tamper prevention. Profiles and vaults can be updated while in secure storage; if this occurs, they are encrypted and MACed before being written back to the insecure storage, which may in turn propagate them to external backup storage. In yet other embodiments, profiles and vaults are stored entirely in secure storage, in plaintext, which allows them to be both read and written -- and in particular, searched.

[0020] Example Transaction Types

[0021] A variety of transaction types can take place in the environment shown in Figure 1, examples of which are discussed in this section.

[0022] Initial Registration

[0023] In order to begin using the techniques described herein, users perform some form of initial registration. As one example, suppose Alice launches an enrollment program installed on laptop 102. She uses the program to capture various biometric information (e.g., fingerprints, photographs of her face, etc.). A user profile is created for Alice, and the biometric information captured about her is encoded into a plurality of templates, such as templates 210 and 214. In some embodiments, Alice is also explicitly asked to supply credential information for services she would like to use, such as by providing the domain name of social networking site 120, along with her username and password for site 120. In other embodiments, domain/username/credential information is at least passively captured on Alice's behalf and included in one or more vaults such as vault 220. Credential information can also be important from a browser password manager already in use by Alice or other appropriate source. In some

embodiments, Alice also registers with cloud storage service 140, which will allow her to back up her authentication information and to synchronize it across her devices (e.g., 102 and 104), as described in more detail below.

[0024] Other registration approaches can also be used. For example, registration can be integrated into the experience the first time a device is used. Thus, when Bob first turns on tablet 106, he may be prompted to take a picture of his face (with a profile/templates being created in response). Similarly, the first time Charlie uses tablet 106, the techniques described herein can be used to determine that Charlie does not yet have a profile (e.g., because none of the templates already present on tablet 106 match his biometrics) and Charlie can be prompted to enroll as a second user of the device.

[0025] Authentication

[0026] Suppose Alice wishes to authenticate to banking website 122. Using a fingerprint reader incorporated into her laptop, she performs a fingerprint scan, which causes her biometric features to be extracted and compared to any stored templates residing on her computer. If a match is found, an associated decryption key is selected, and the associated vault is loaded and decrypted. The vault is scanned for an entry that matches the selected service provider (i.e., website 122). If a matching entry is found, the associated domain, username, and site credential are extracted from the vault. In some embodiments, the validity of the domain name mapping is verified at this point to harden the system against domain name poisoning. Next, a secure connection is established between Alice's computer and the service provider, and Alice is authenticated. For service providers supporting strong user authentication, mutual SSL can be used, for example. A variety of policies can be involved when performing matching. For example, to access certain domains, Alice's print may need only match template 210. To access other domains, Alice may need to match multiple templates (e.g., both 210 and 214). As another example, in order to access social networking site 120, Alice may merely need to be sitting in front of her computer, which has an integrated webcam. Even in relatively low light conditions, a match can be performed against Alice's face and features stored in a template. However, in order to access bank website 122, Alice may need a high quality photograph (i.e., requiring her to turn on a bright light) and may need to demonstrate liveness (e.g., by blinking or turning her

head). As yet another example, other contextual information can be included in policies. For example, if Alice's IP address indicates she is in a country that she is not usually in, she may be required to match multiple templates (or match a template with more/better quality features) in order to access retailer 124, as distinguished from when her IP address indicates she is at home.

[0027] In some embodiments, the biometric sensor used by a user may be a peripheral device (e.g., a mouse with an integrated fingerprint scanner that is connected to the user's primary device via USB). In such scenarios, the peripheral device may be responsible for storing at least a portion of authentication information and may perform at least some of the authentication tasks previously described as having been performed by Alice's computer. For example, instead of processors 304 and 308, and storages 302 and 306 being collocated on a single device (e.g., laptop 102), processor 304 and storage 302 may be present on a primary device, and processor 308 and storage 306 may be present on a peripheral device (e.g., that also includes a sensor, such as a fingerprint reader).

[0028] In such scenarios, once Alice's login to banking website 122 is successfully completed, the secure session can be handed over from the peripheral device to the primary device, in a way that does not allow the primary device retroactive access to the plaintext data of the transcripts exchanged between the peripheral device and the service provider. One way this can be accomplished is by renegotiating SSL keys between the peripheral device and the website, after which the newly negotiated key can be handed off from the peripheral device to the primary device. This avoids retroactive credential capture in a setting where the device is infected by malware.

[0029] An example of renegotiation is depicted in Figure 4. Specifically, after a user has successfully authenticated to a fingerprint reader, a login is performed to a service provider. Using the primary device (404) as a proxy, the peripheral fingerprint reader 402 negotiates a first SSL connection (408) with a service provider 406, over which credentials are exchanged. The proxy then renegotiates SSL (410), which replaces the old key with a new one. The new key is disclosed to the device, which then seamlessly takes over the connection with the service provider and performs the transaction protected by the authentication. The credentials exchanged during the first SSL connection cannot be accessed by device 404, since the key of the

renegotiated session is independent of the key of the original session; this provides protection against malware residing on the device. Renegotiation can be used when the primary device 404 is believed to be in a safe state when performing the negotiation of the SSL connection, but it is not known whether it is in a safe state during the transaction protected by the authentication. Renegotiation can also be used when a secure component of the primary device 404 performs the negotiation of the SSL connection and another and potentially insecure component of the primary device 404 is involved in the transaction protected by the authentication.

[0030] Figure 5 illustrates an embodiment of a process for performing authentication translation. The process begins at 502 when a request to access a resource is received, as is an authentication input. One example of the processing performed at 502 is as follows. Suppose Alice wishes to sign into social networking website 120. She directs a web browser application installed on client 102 to the social networking website. Authentication translator module 132 recognizes, from the context of Alice's actions (e.g., that she is attempting to access site 120 with her browser) that she would like to access a particular resource. Authentication translator module 132 prompts Alice (e.g., by a popup message or via a sound) to provide biometric information (e.g., to use the integrated fingerprint reader on her laptop). In some embodiments, the translator module does not prompt Alice, for example, because Alice has been trained to provide biometric information automatically when attempting to access certain resources. In yet other embodiments, the translator module only prompts Alice if she fails to provide acceptable biometric information within a timeout period (e.g., 30 seconds).

[0031] Module 132 compares Alice's supplied biometric data to the templates stored on her computer. If a suitable match is found, and if an entry for site 120 is present in the applicable vault, at 504, a previously stored credential associated with the resource is accessed. In particular, the username and password for the website, as stored in a vault, such as vault 220, are retrieved from the vault.

[0032] Finally, at 506, the credential is provided to the resource. For example, Alice's username and password for site 120 are transmitted to site 120 at 506. The credential can be transmitted directly (e.g., by the module or by Alice's computer) and can also be supplied

indirectly (e.g., through the use of one or more proxies, routers, or other intermediaries, as applicable).

[0033] Other devices can also make use of process 500 or portions thereof. For example, when Alice launches a banking application on phone 104, implicit in her opening that application is her desire to access the resources of website 134. The application can take Alice's picture and compare it to stored templates/vault information. If an appropriate match is found, a credential can be retrieved from the vault on her phone (or, e.g., retrieved from cloud storage service 140) and provided to website 134.

[0034] As another example, suppose Charlie is using tablet 106 and attempts to visit site 120, whether via a dedicated application or via a web browser application installed on the tablet. Charlie's photograph is taken, and then compared against the profiles stored on tablet 106 (e.g., both Bob and Charlie's profiles). When a determination is made that Charlie's photograph matches a template stored in his stored profile (and not, e.g., Bob's), Charlie's credentials for site 120 are retrieved from a vault and transmitted by an authentication translator module residing on client 106.

[0035] As yet another example, kiosk 108 can be configured to provide certain local resources (e.g., by displaying a company directory or floor plan on demand) when users speak certain requests into a microphone. Enrolled users (e.g., with stored voiceprint or facial recognition features) can be granted access to additional/otherwise restricted services in accordance with the techniques described herein and process 500.

[0036] New device

[0037] In some embodiments, to register a new device, a user provides an identifier, such as a username or an account number to the device. The new device connects to an external storage (such as cloud storage 140), provides the user identifier and credential, and downloads the user's templates/vaults from the service. In some embodiments, the templates/vaults are encrypted. Once downloaded, the template is decrypted and stored in a secure storage area, while the still encrypted vault can be stored in insecure storage. The decryption key can be generated from information the user has/knows, or from biometric data -- such as features

extracted from fingerprinting of all ten fingers. In some embodiments, more arduous fingerprinting is required for the setup of a new device than for regular authentication to avoid that a new device gets registered by a user thinking she is merely authenticating -- or worse still, simply touching the device. Moreover, it translates into higher entropy of the decryption keys.

[0038] Backup Authentication

[0039] Backup authentication allows a user, such as Alice, to access resources in the event she is unable to or unwilling to interact with a particular biometric sensor. As one example, instead of having a single template associated with her profile, Alice can have multiple templates associated with it, e.g., where the first template includes fingerprint features and the second template includes voice biometric, facial recognition, or iris detection features. As a second example, where the service Alice is connecting to is a legacy website (i.e., one that users authenticate to using usernames and passwords), such a service would allow the use of passwords and password reset mechanisms by Alice without requiring Alice to use a fingerprint reader.

[0040] In various embodiments, environment 100 supports the ability of users (e.g., under duress) to release the contents of their vaults. For example, if Alice was physically threatened with the loss of a finger by a criminal, Alice could instead release the contents of her vault(s) – the ultimate goal of the criminal. As one example, in the event Alice supplies all 10 fingerprints to the sensor, provides a special password, or supplies a fingerprint and a second identifier, a cleartext version of her vault(s) could be made available.

[0041] Access Policies

[0042] In various embodiments, cloud storage service 140 is configured to accept backups from multiple devices associated with a single account, and synchronize the updates so that all devices get automatically refreshed. For example, Alice's laptop 102 and phone 104 could both communicate with cloud storage service 140 which would keep their authentication information synchronized. Refreshes can also be made in accordance with user-configured restrictions. For example, Alice's employer could prevent privileged employer data from being stored on shared personal devices, or on any device that was not issued by the employer. As

another example, arbitrary policies can be defined regarding the access to and synchronization of software and data, and to tie a license or access rights to a person (and her fingerprint) rather than to a device. As yet another example, in some embodiments (e.g., where a device is made publicly available or otherwise shared by many users), no or a reduced amount of authentication information resides on a device, and at least a portion of authentication information is always retrieved from cloud storage service 140.

[0043] Remote wiping

[0044] Remote wiping of a user's authentication information (e.g., templates) can be used both to "unshare" previously shared devices (e.g., where Bob and Charlie both have user profiles on their shared tablet 106), and to avoid that criminals with physical component access to lost devices gain access to templates and vault contents. In some embodiments, polices such as ones where a template self-wipes if it is not matched within a particular duration of time are supported. Since user data can be frequently backed up to the cloud storage, and recovered from this using the new device registration process, inconvenience to the user will be minimized.

[0045] **Legacy Server Support**

[0046] New authentication schemes typically require changes to a significant codebase residing with service providers. If the code is well written and documented, such changes may be relatively simple. Commonly, though, this may not be so. The engineers who originally wrote the code of relevance may have long since left the company; the code they left behind may be poorly documented -- if documented at all. In severe cases, the legacy code may have been written in an outdated programming language or written in a way that does not follow guidelines for good code. This makes updates to the codebase impractical or virtually impossible in many common cases. Even if none of these challenges complicate the desired modifications, it is commonly a great bureaucratic burden to obtain permission to make modifications (e.g., to store an extra field in a backend database), because every affected part of the organization may need to review the request.

[0047] As will be described in the following section, the technologies described herein can be used in conjunction with legacy servers (e.g., existing servers that rely on usernames and

passwords to authenticate users), and in particular, can be used without requiring modification to such legacy servers.

[0048] Cookies

[0049] Cookies are commonly used by legacy servers for user authentication. Unfortunately, cookies have several problems. For one thing, they are sometimes deleted -- whether explicitly/intentionally by the end user or by the user's software. In addition, cookies are commonly stolen. Approaches such as cache cookies and identification using user agents can be more resistant to these problems, however, they have their own problems. For example, their use requires new code and new fields in the credential database stored by the server.

[0050] In some embodiments, authentication translators, such as translators 134 and 136 (also referred to herein as proxies) provide authentication translation services on behalf of associated services. Translators 134 and 136 are illustrated as single logical devices in Figure 1. In some embodiments, the translators comprise standard commercially available server hardware (e.g., a multi-core processor, 4+ Gigabytes of RAM, and one or more Gigabit network interface adapters) and run typical server-class operating systems (e.g., Linux). Translators 134 and 136 can also be implemented using a scalable, elastic architecture and may comprise several distributed components, including components provided by one or more third parties. For example, translators 134 and 136 may store user credential information or may task cloud storage service 140 with storing at least a portion of that information.

[0051] In the case of authentication translator 134, service is provided with respect to bank website 122 only. Authentication translator 134 is positioned between a legacy web server (122) and the Internet (110) -- and therefore between the legacy server and any client devices. Authentication translator 134 is configured to translate traffic between the legacy server and client devices so that the client devices (and respective users) perceive the new authentication mechanism, while the legacy server remains unchanged. Authentication translator 136 works similarly, but it provides authentication translation as a third party service to multiple providers, an example of which is online camera retailer 124.

[0052] Authentication translators 134 and 136 can also perform process 500. For example, when a device transmits a request to access website 122, the request is intercepted by translator 134, as is cookie/user agent information. The received information can be used to determine a username/password associated with the device, and that information can be passed by translator 134 to website 122 on behalf of the device.

[0053] Figure 6 illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator. The translator (referred to in the figure as a "proxy") fails to identify the client, and passes on the request to the legacy server. The legacy server responds to the request and sets a cookie. The proxy passes on the response, including the cookie and also a cache cookie. The proxy stores the information about both these types of identifiers, along with the user agent of the client device. This triplet of information is also referred to herein as an identifier.

[0054] Figure 7 illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator. In this scenario, the device's request is accompanied by some form of identifying information. The proxy uses this information to identify the associated cookie and passes this along to the legacy server, along with the request. In some embodiments, additional processing is also involved. For example, reading a cache cookie may require user interaction. Moreover, if not all of the identifying information is present in the request, the proxy can be configured to set the missing information again by sending a corresponding request to the client device.

[0055] The translation of cache cookies and user agent information to cookies involves a two-way translation. First, when the legacy server sets a cookie, the proxy will set the two types of cookies -- both an HTML cookie and a cache cookie -- and then create a new record in which the two cookies are stored, along with the user agent information of the client device. The user agent information can include quite a bit of data associated with a browser -- such as the browser type and version, the clock skew, and the fonts that are installed. While each of these pieces of information only contributes a small amount of entropy, the collection of items can be sufficient to identify the device in most cases. Moreover, while some of these types of data may change over time -- in fact, all of them may -- they do not typically change, and when one or two of

them do, the others typically do not. When the client device is used to visit a site controlled by the legacy server, the cookie, cache cookie and user agent information are read (if available), the record identified, and the request translated and sent to the legacy server. When a legacy server requests that the user password is updated (e.g., as part of an annual or other periodic requirement), the transmission of this request to the user can be suppressed -- in which case the database of the proxy is updated to create the illusion of an update. The user can be involved in authentication as needed, e.g., if, in addition to supplying a credential, a user must also solve a CAPTCHA, the CAPTCHA can be displayed to the user (with the user's credentials being handled in accordance with the techniques described herein).

[0056] Figure 8 shows the structure of an example of a cache cookie used in some embodiments. Cache cookies can be associated with a particular webpage, just like an HTML cookie can. In the example shown in Figure 8, the proxy wishes to associate a page "calling page.html" with a cache cookie. It embeds a request for a second object, "samename.html" in callingpage.html for every visitor. However, as the cache cookie is set for one visitor, a customized samename.html is served to this visitor. The page samename.html refers to an object with a different name for each user; that object is referred to as "uniquename.jpg." The cache cookie is set by embedding the request for samename.html in callingpage.html. The client browser attempts to render this, causing a request for samename.html from the server. The server configures samename.html to refer to a uniquely named file uniquename.jpg, and serves samename.html to the client. For the client browser to render samename.html, it requests the file uniquename.jpg, which is intentionally not served. That concludes the setting of a cache cookie. As a user returns to the page callingname.html, the browser again attempts to render the entire page, which causes it to load the object samename.html from its cache. As that is rendered, the client browser requests uniquename.jpg, which is not in its cache (since it was not served previously). The server still does not serve it, but takes note of the name of the file being requested, as it identifies the client device. Note that samename.html can be displayed in a zero-sized iframe, which makes the end user unaware of it being rendered.

[0057] A cache cookie is an implementation of the typical cookie functionality that uses the client device's browser cache. Unlike user agents, it does not change over time, and like standard HTML cookies, it cannot be read by a party other than that which set it. However, like

HTML cookies, it could be deleted -- by the user clearing his or her browser cache. Cache cookies are not automatically transmitted with GET requests, unless the cache elements are embedded in the referring pages. This adds a potential round of communication in some settings. By relying on user agent information, cache cookies, and HTML cookies to identify the client device, it is much more likely that a machine will be recognized than if only HTML cookies are used.

[0058] Although the foregoing embodiments have been described in some detail for purposes of clarity of understanding, the invention is not limited to the details provided. There are many alternative ways of implementing the invention. The disclosed embodiments are illustrative and not restrictive.

[0059] WHAT IS CLAIMED IS:

CLAIMS

1. A system, comprising:

a processor configured to:

receive, at an authentication translator, a request to access a resource and an authentication input, wherein the authentication input corresponds to at least one stored record and wherein the stored record is associated at least with the resource;

in response to the receiving, access a previously stored credential associated with the resource; and

cause the credential to be provided to the resource; and

a memory coupled to the processors and configured to provide the processor with instructions.

ABSTRACT OF THE DISCLOSURE

Authentication translation is disclosed. A request to access a resource is received at an authentication translator, as is an authentication input. The authentication input corresponds to at least one stored record. The stored record is associated at least with the resource. In response to the receiving, a previously stored credential associated with the resource is accessed. The credential is provided to the resource.

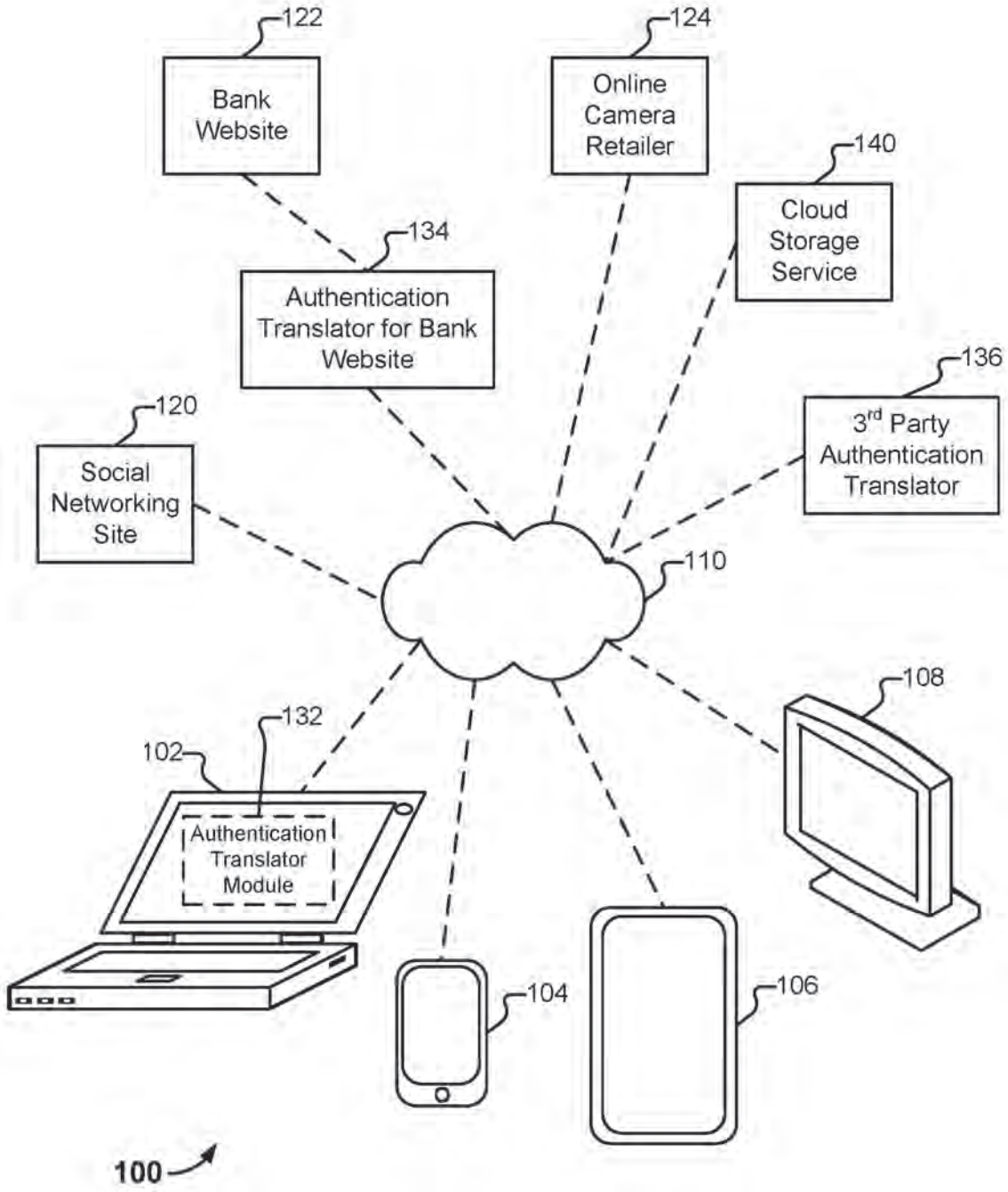

122
Bank Website
124
Online Camera Retailer
140
Cloud Storage Service
134
Authentication Translator for Bank Website
136
3rd Party Authentication Translator
120
Social Networking Site
110
108
132
102
Authentication Translator Module
104
106
100

FIG. 1

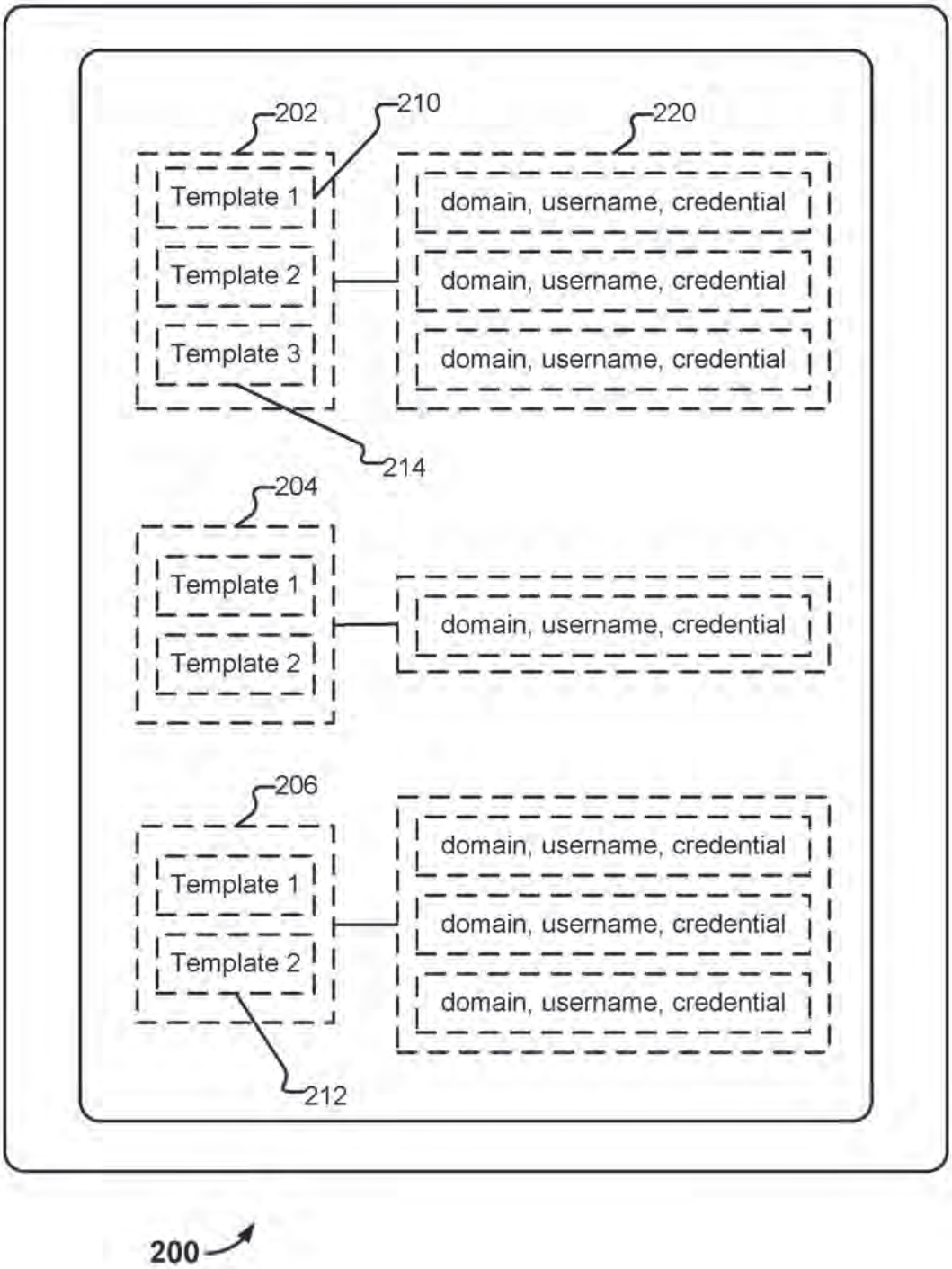
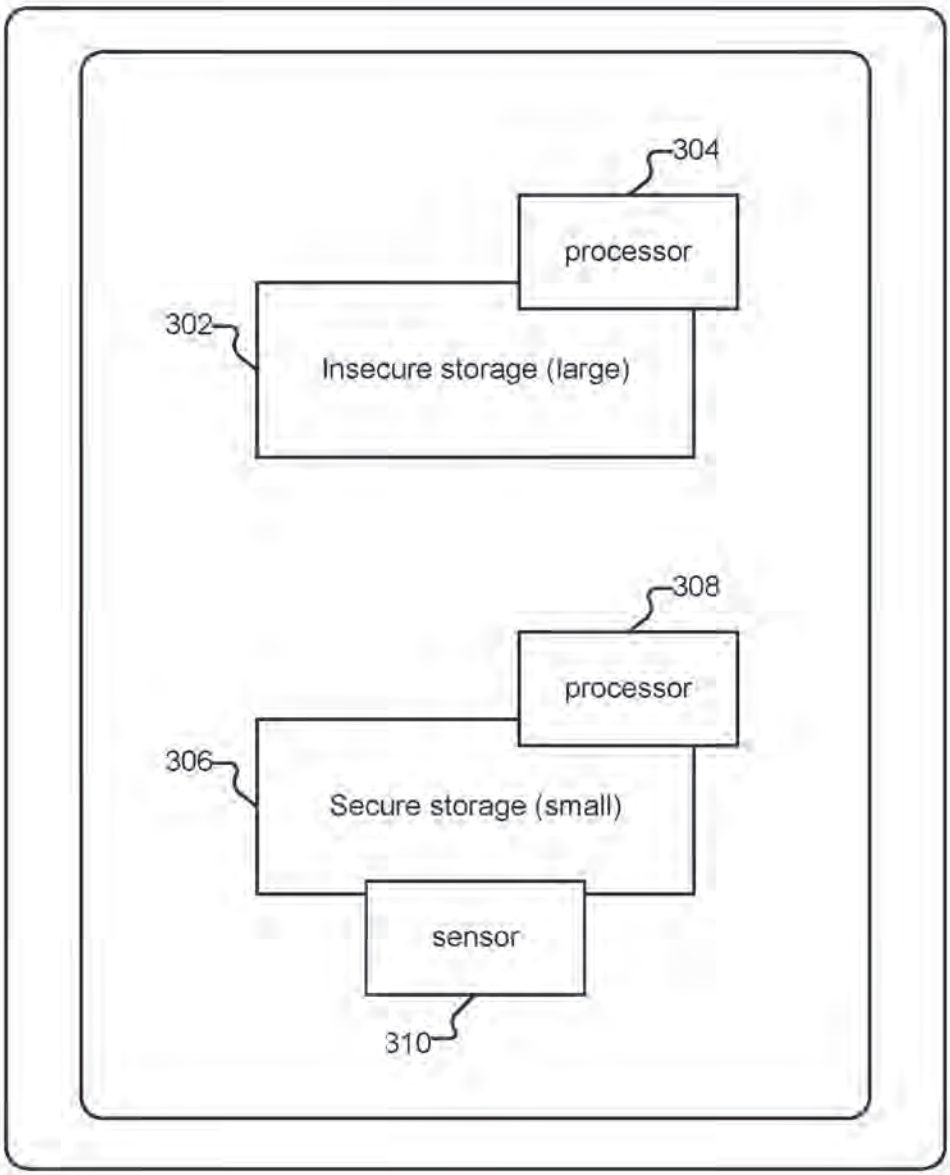

202
210
220
Template 1
Template 2
Template 3
domain, username, credential
domain, username, credential
domain, username, credential
214
204
Template 1
Template 2
domain, username, credential
206
Template 1
Template 2
domain, username, credential
domain, username, credential
domain, username, credential
212
200

FIG. 2


304
processor
302
Insecure storage (large)
308
processor
306
Secure storage (small)
sensor
310

300

FIG. 3

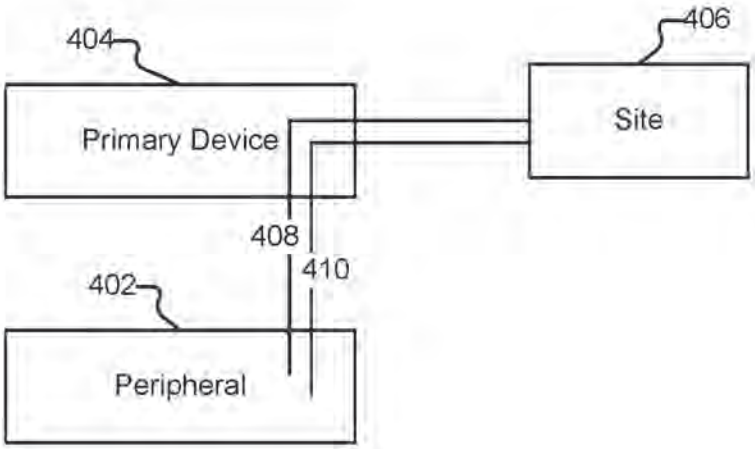


FIG. 4

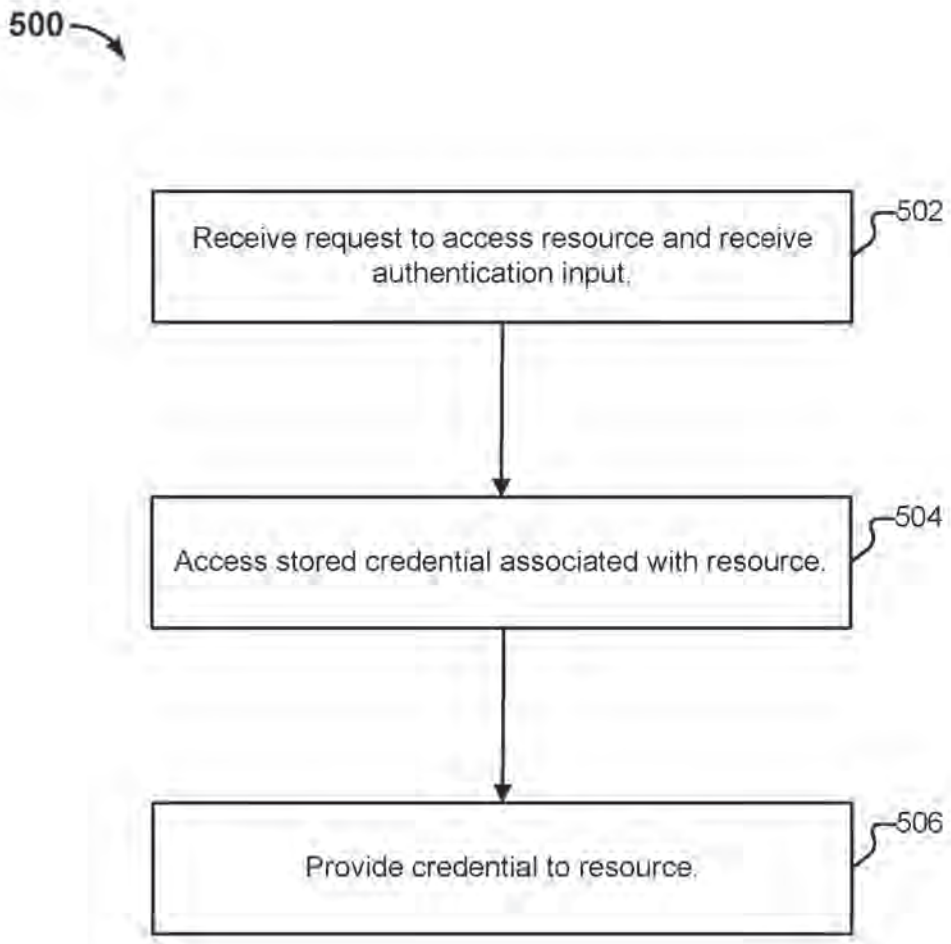

500
Receive request to access resource and receive authentication input.
502
Access stored credential associated with resource.
504
Provide credential to resource.
506

FIG. 5

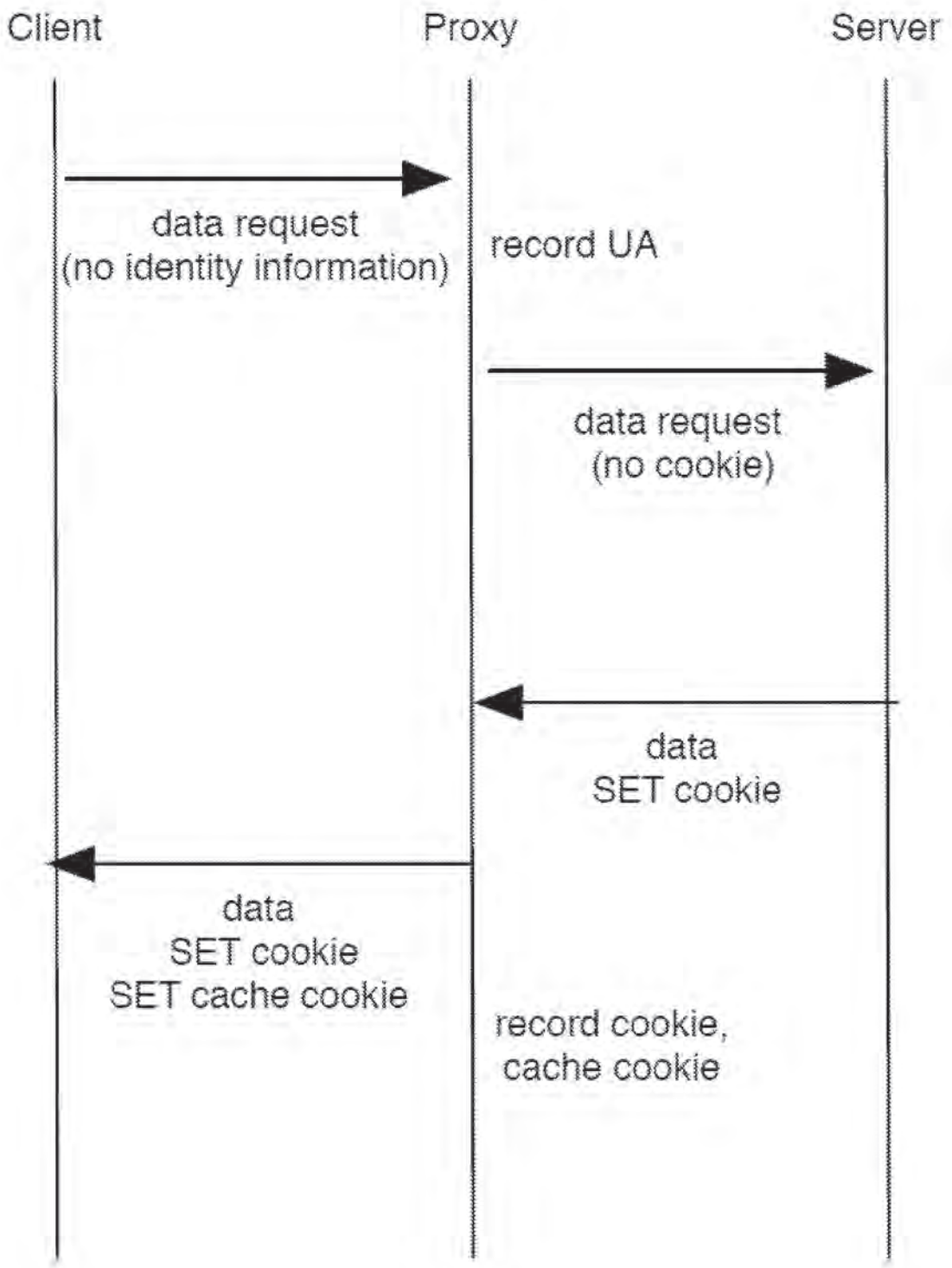

Client
Proxy
Server
data request
(no identity information)
record UA
data request
(no cookie)
data
SET cookie
data
SET cookie
SET cache cookie
record cookie,
cache cookie

FIG. 6

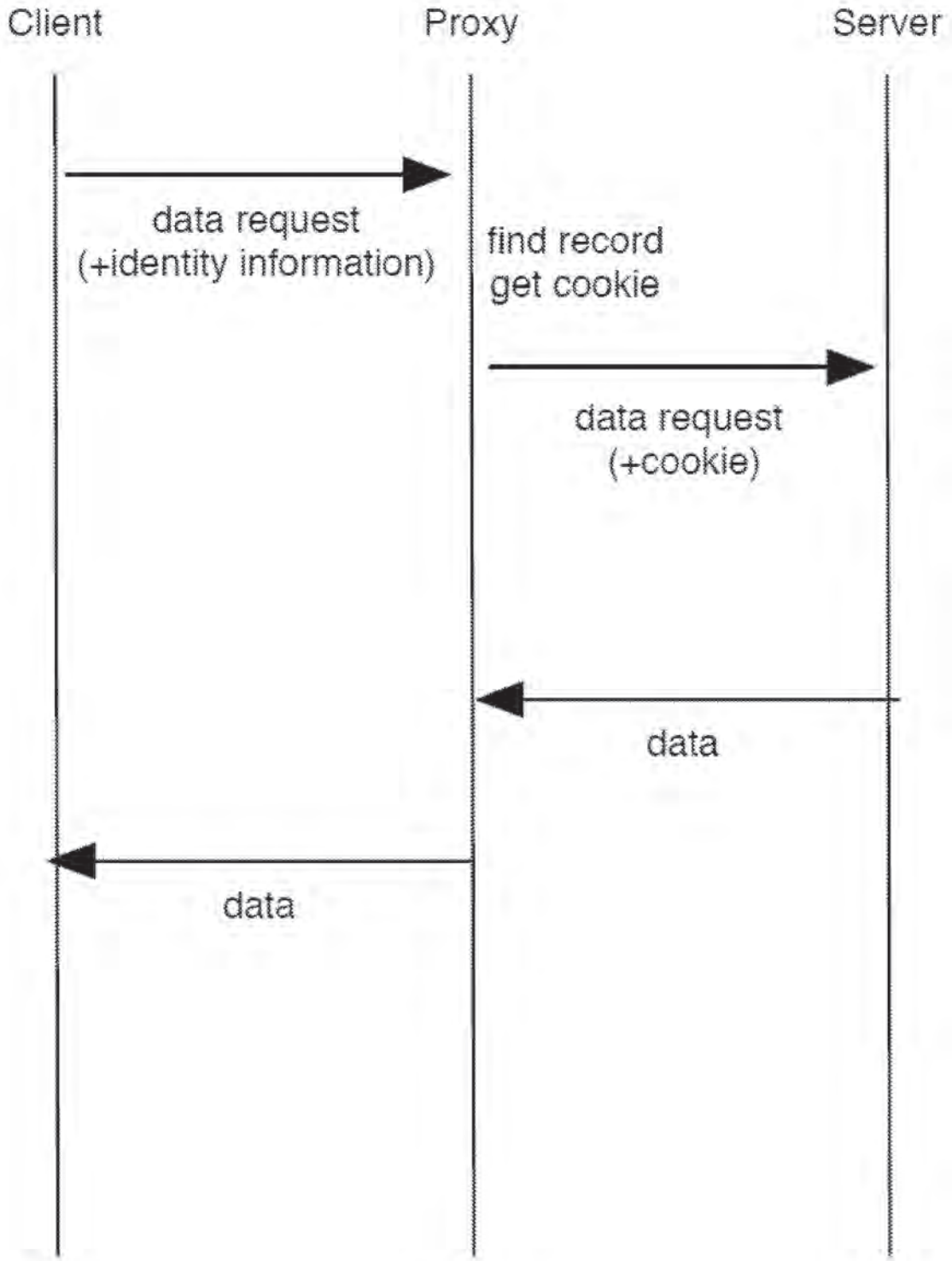

Client
Proxy
Server
data request
(+identity information)
find record
get cookie
data request
(+cookie)
data
data

FIG. 7

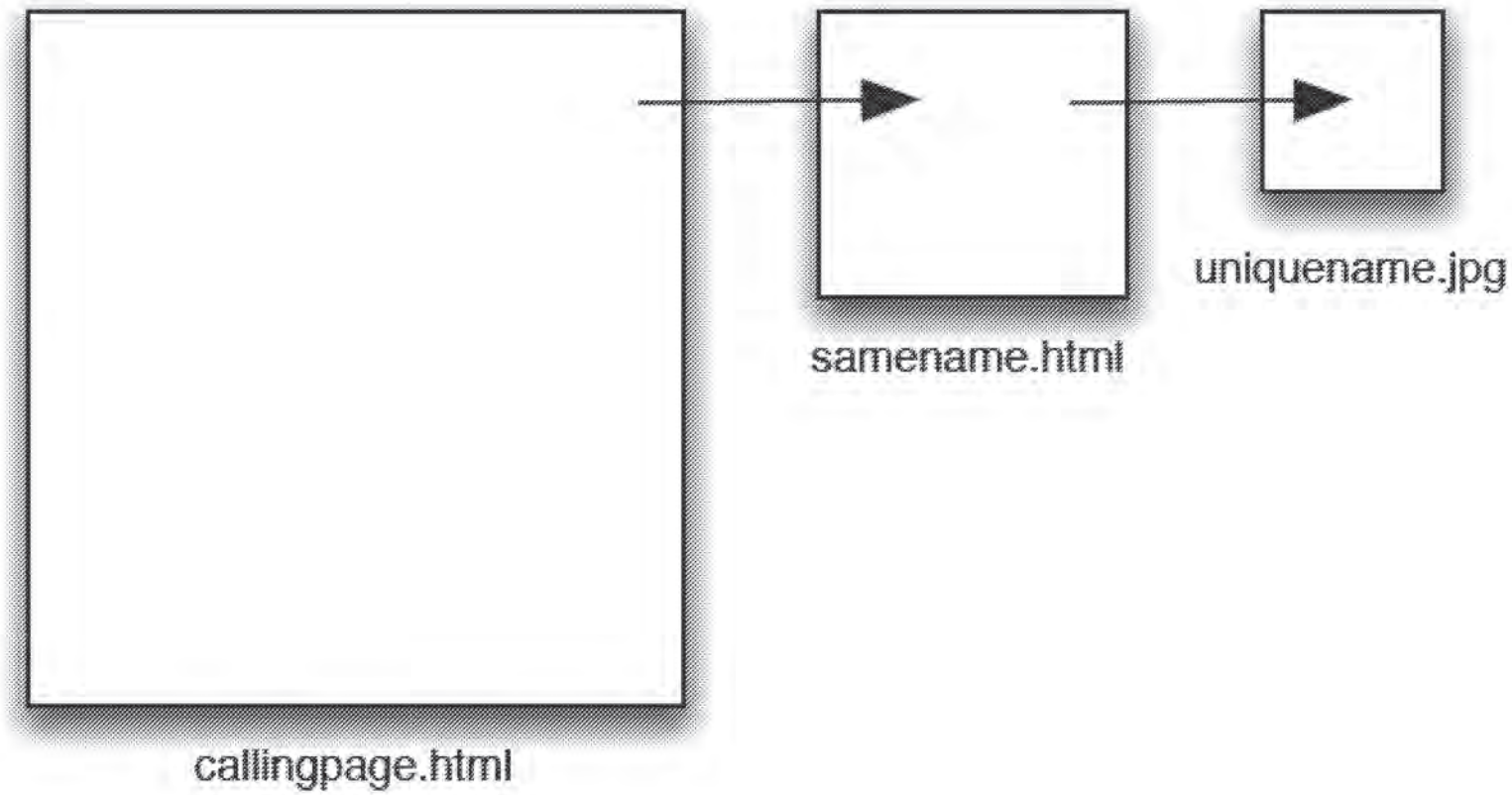


FIG. 8

Application Serial No. 13/706,254

Filing date: December 5, 2012

Patent No. 9,294,452

Issue date: March 22, 2016

US009294452B1

(12) **United States Patent**
Jakobsson

(10) **Patent No.: US 9,294,452 B1**
(45) **Date of Patent: Mar. 22, 2016**

(54) **AUTHENTICATION TRANSLATION**

(71) Applicant: **RightQuestion, LLC**, Portola Valley, CA (US)

(72) Inventor: **Bjorn Markus Jakobsson**, Mountain View, CA (US)

(73) Assignee: **RightQuestion, LLC**, Portola Valley, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **13/706,254**

(22) Filed: **Dec. 5, 2012**

Related U.S. Application Data

(60) Provisional application No. 61/569,112, filed on Dec. 9, 2011, provisional application No. 61/587,387, filed on Jan. 17, 2012.

(51) **Int. Cl.**
H04L 29/06 (2006.01)

(52) **U.S. Cl.**
CPC ***H04L 63/08*** (2013.01); ***H04L 63/0815*** (2013.01); ***H04L 63/0823*** (2013.01); *H04L 63/0861* (2013.01); *H04L 63/0869* (2013.01)

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,512,965	B1 *	3/2009	Amdur et al.	726/1
7,950,051	B1 *	5/2011	Spitz et al.	726/6
8,549,300	B1 *	10/2013	Kumar et al.	713/175
2009/0100269	A1 *	4/2009	Naccache	713/186
2011/0205016	A1 *	8/2011	Al-Azem et al.	340/5.52
2011/0231651	A1 *	9/2011	Bollay	713/152

OTHER PUBLICATIONS

"Managing Authorization and Access Control" Author: unknown Published: Nov. 3, 2005 pp. 1-12 URL: http://technet.microsoft.com/en-us/library/bb457115.aspx.*

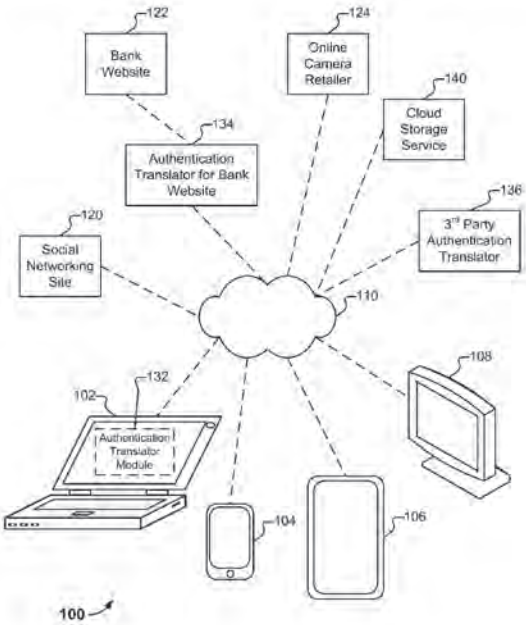
* cited by examiner

Primary Examiner — Harunur Rashid
Assistant Examiner — Andrew Steinle
(74) *Attorney, Agent, or Firm* — Van Pelt, Yi & James LLP

(57) **ABSTRACT**

Authentication translation is disclosed. A request to access a resource is received at an authentication translator, as is an authentication input. The authentication input corresponds to at least one stored record. The stored record is associated at least with the resource. In response to the receiving, a previously stored credential associated with the resource is accessed. The credential is provided to the resource.

14 Claims, 8 Drawing Sheets



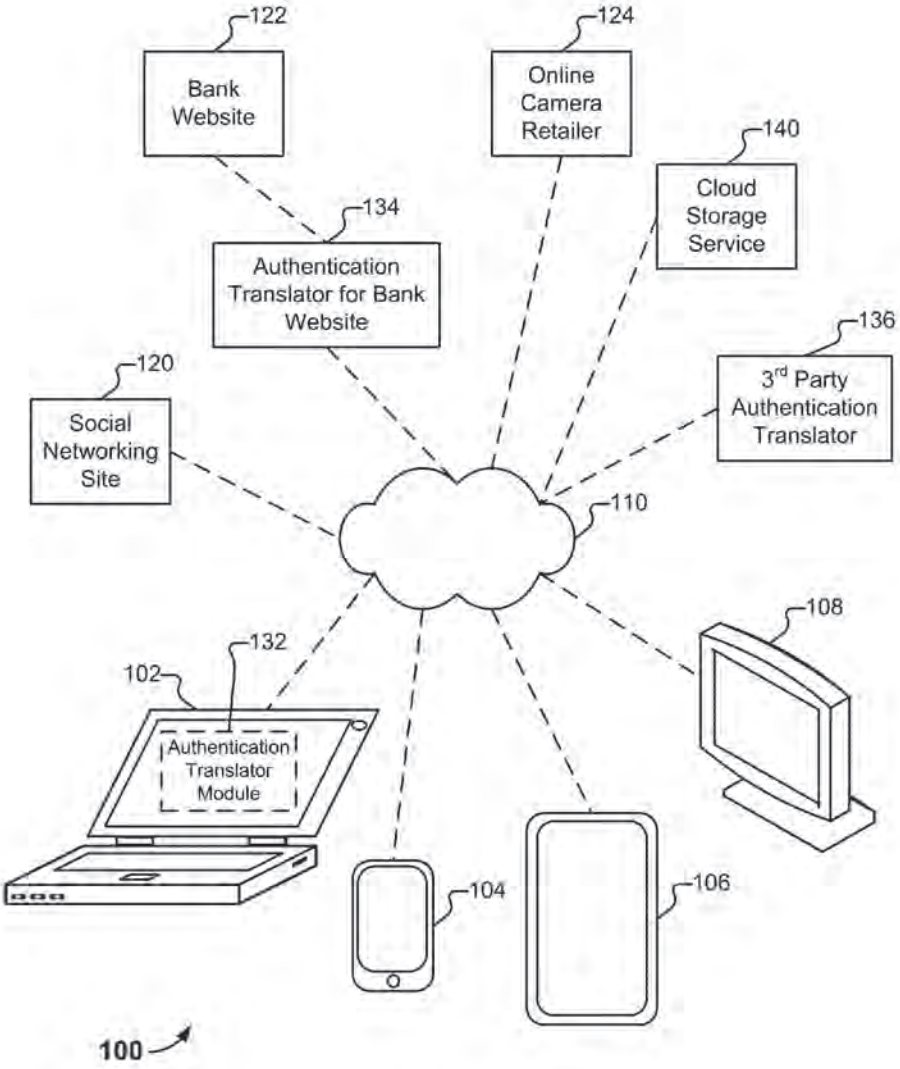

122
Bank Website
124
Online Camera Retailer
140
Cloud Storage Service
134
Authentication Translator for Bank Website
136
3rd Party Authentication Translator
120
Social Networking Site
110
108
132
102
Authentication Translator Module
104
106
100

FIG. 1

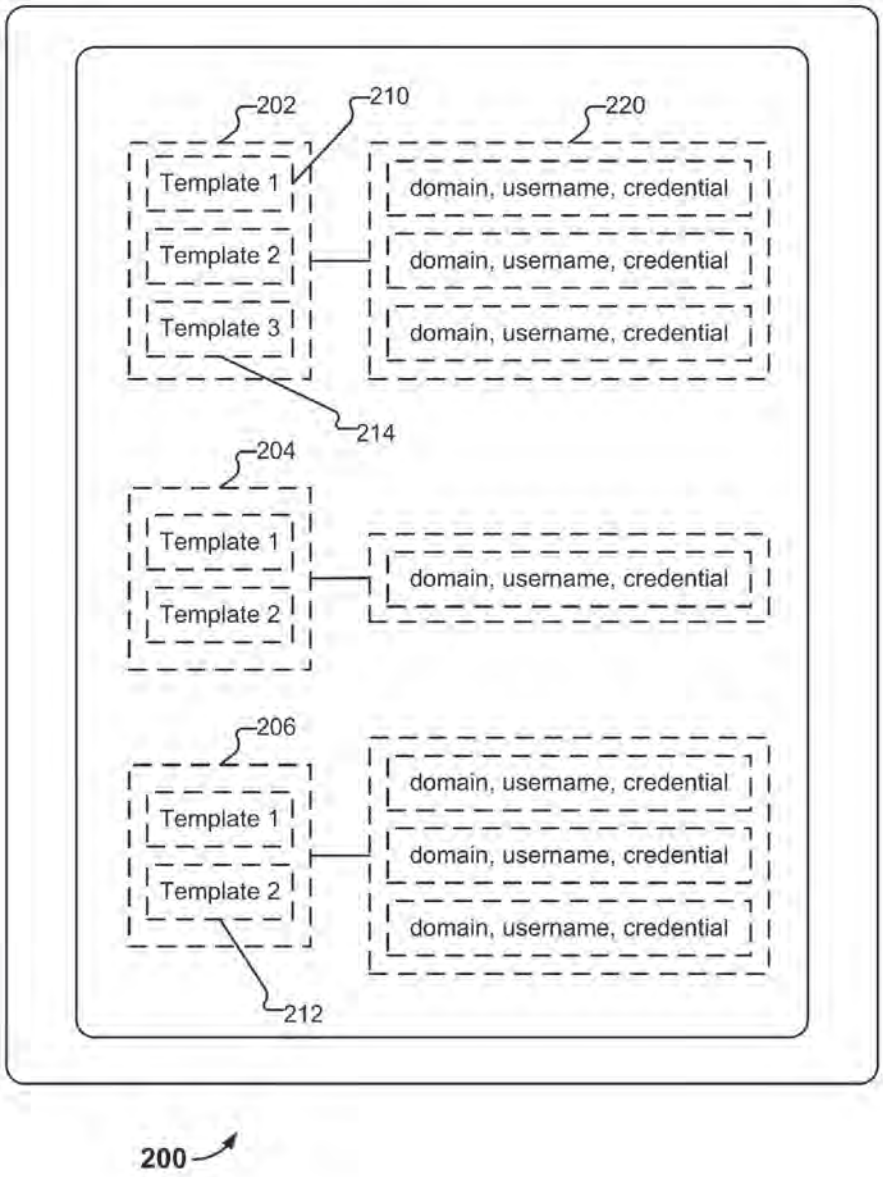
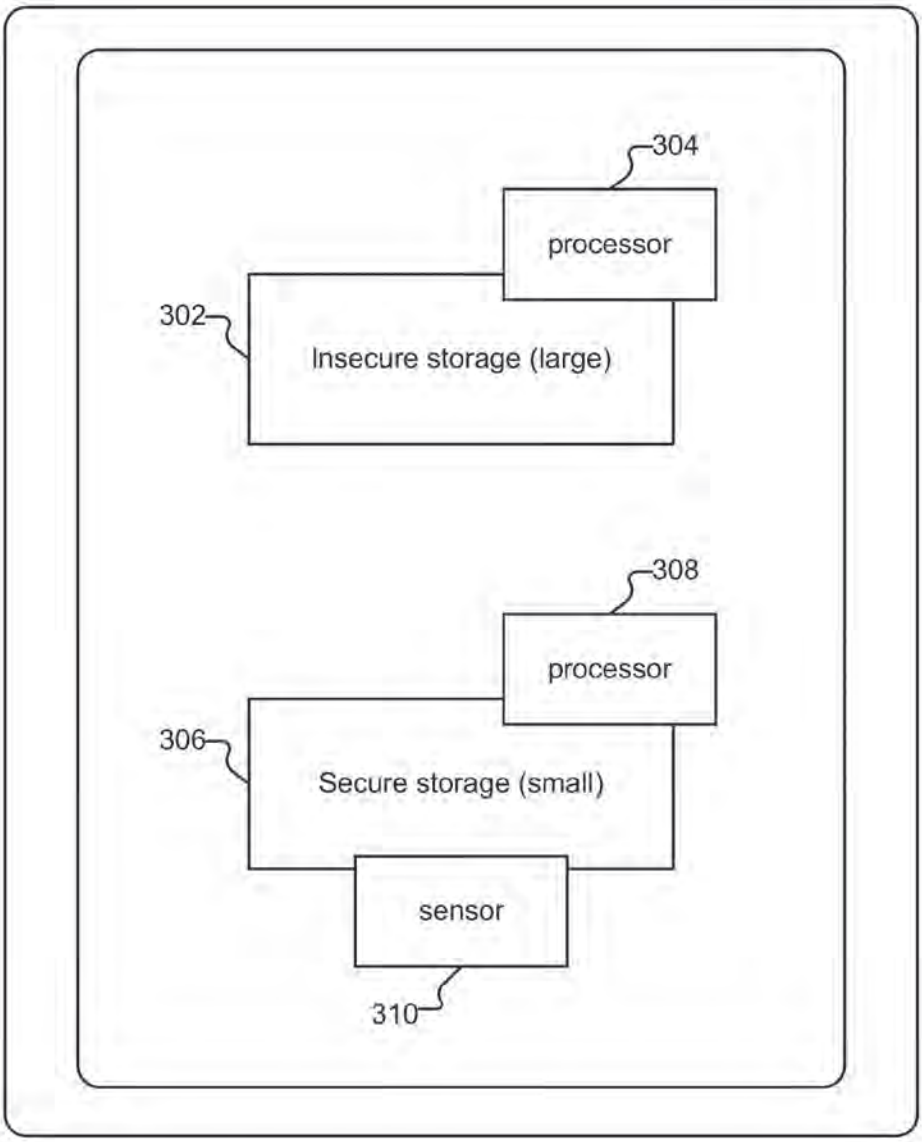

202
210
220
Template 1
Template 2
Template 3
domain, username, credential
domain, username, credential
domain, username, credential
214
204
Template 1
Template 2
domain, username, credential
206
Template 1
Template 2
domain, username, credential
domain, username, credential
domain, username, credential
212
200

FIG. 2


304
processor
302
Insecure storage (large)
308
processor
306
Secure storage (small)
sensor
310

300

FIG. 3

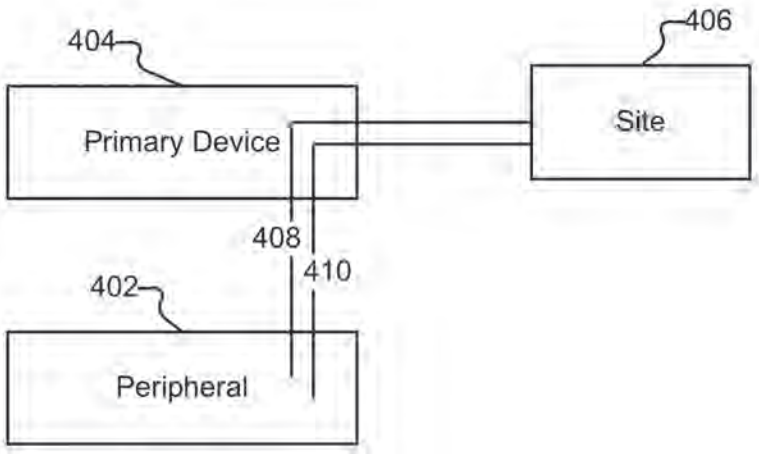


FIG. 4

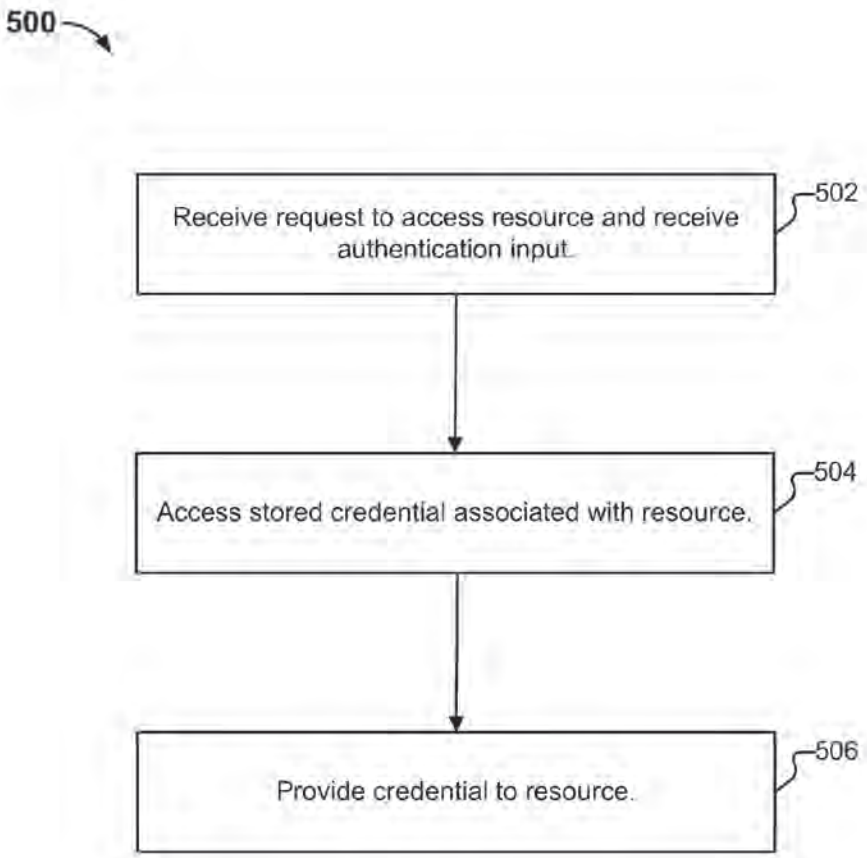

500
Receive request to access resource and receive authentication input.
502
Access stored credential associated with resource.
504
Provide credential to resource.
506

FIG. 5

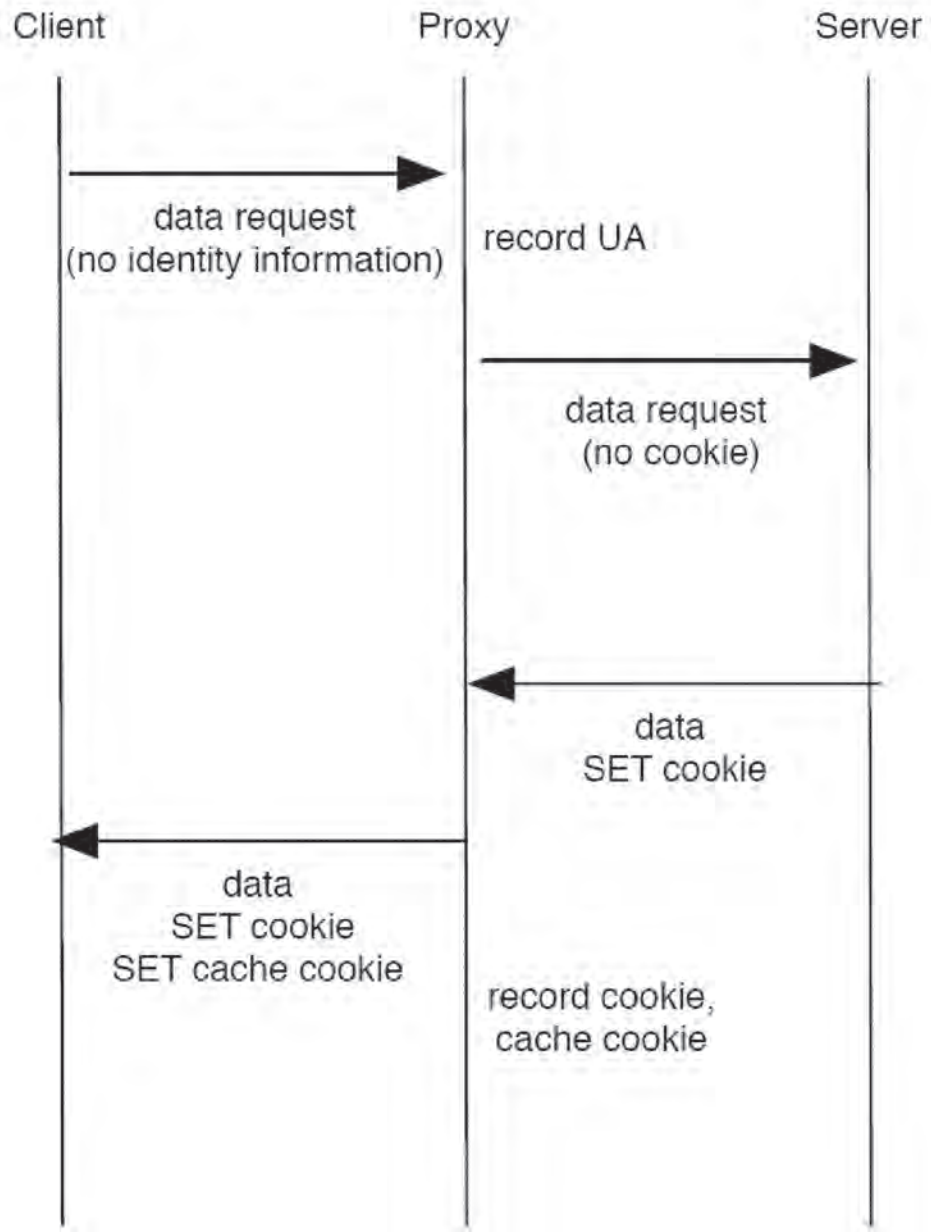

Client
Proxy
Server
data request
(no identity information)
record UA
data request
(no cookie)
data
SET cookie
data
SET cookie
SET cache cookie
record cookie,
cache cookie

FIG. 6

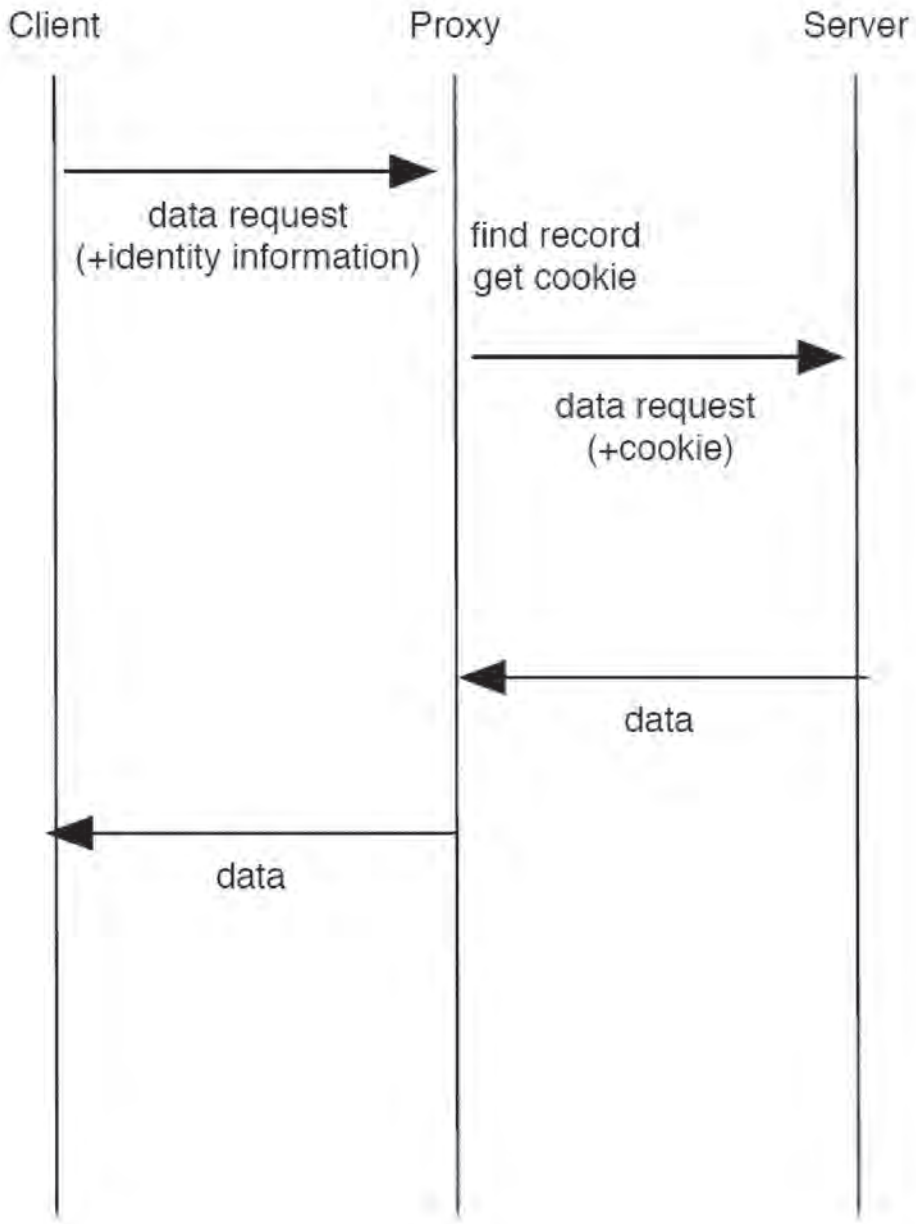

Client
Proxy
Server
data request
(+identity information)
find record
get cookie
data request
(+cookie)
data
data

FIG. 7

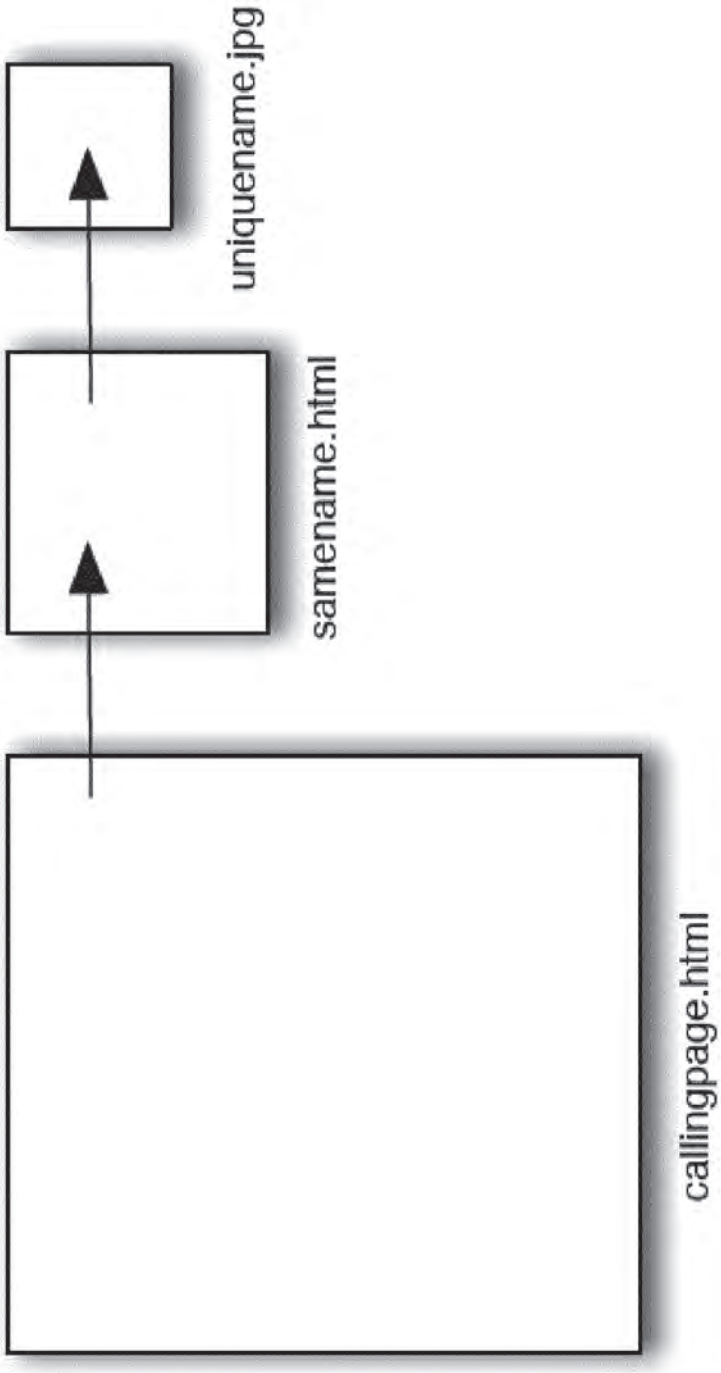


FIG. 8

AUTHENTICATION TRANSLATION

CROSS REFERENCE TO OTHER APPLICATIONS

This application claims priority to U.S. Provisional Patent Application No. 61/569,112 entitled BACKWARDS COMPATIBLE ROBUST COOKIES filed Dec. 9, 2011, and also claims priority to U.S. Provisional Patent Application No. 61/587,387 entitled BIOMETRICS-SUPPORTED SECURE AUTHENTICATION SYSTEM filed Jan. 17, 2012, both of which are incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

Providing credentials to a service, whether via a mobile or other device, is often a tedious experience for a user. Unfortunately, to make authentication easier for themselves, users will often engage in practices such as password re-use, and/or the selection of poor quality passwords, which render their credentials less secure against attacks. Accordingly, improvements in authentication techniques would be desirable. Further, it would be desirable for such improvements to be widely deployable, including on existing/legacy systems.

BRIEF DESCRIPTION OF THE DRAWINGS

Various embodiments of the invention are disclosed in the following detailed description and the accompanying drawings.

FIG. **1** illustrates an embodiment of an environment in which authentication translation is provided.

FIG. **2** illustrates an embodiment of credential information stored on a device.

FIG. **3** illustrates an embodiment of a device with secure storage.

FIG. **4** illustrates an example of a renegotiation.

FIG. **5** illustrates an embodiment of a process for performing authentication translation.

FIG. **6** illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator.

FIG. **7** illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator.

FIG. **8** shows the structure of an example of a cache cookie used in some embodiments.

DETAILED DESCRIPTION

The invention can be implemented in numerous ways, including as a process; an apparatus; a system; a composition of matter; a computer program product embodied on a computer readable storage medium; and/or a processor, such as a processor configured to execute instructions stored on and/or provided by a memory coupled to the processor. In this specification, these implementations, or any other form that the invention may take, may be referred to as techniques. In general, the order of the steps of disclosed processes may be altered within the scope of the invention. Unless stated otherwise, a component such as a processor or a memory described as being configured to perform a task may be implemented as a general component that is temporarily configured to perform the task at a given time or a specific component that is manufactured to perform the task. As used herein, the

term 'processor' refers to one or more devices, circuits, and/or processing cores configured to process data, such as computer program instructions.

A detailed description of one or more embodiments of the invention is provided below along with accompanying figures that illustrate the principles of the invention. The invention is described in connection with such embodiments, but the invention is not limited to any embodiment. The scope of the invention is limited only by the claims and the invention encompasses numerous alternatives, modifications and equivalents. Numerous specific details are set forth in the following description in order to provide a thorough understanding of the invention. These details are provided for the purpose of example and the invention may be practiced according to the claims without some or all of these specific details. For the purpose of clarity, technical material that is known in the technical fields related to the invention has not been described in detail so that the invention is not unnecessarily obscured.

FIG. **1** illustrates an embodiment of an environment in which authentication translation is provided. In the example shown, a variety of client devices **102-108** connect, via one or more networks (represented as a single network cloud **110**) to a variety of services **120-124** (also referred to herein as sites **120-124**). In particular, client device **102** is a notebook computer owned by a user hereinafter referred to as Alice. Notebook **102** includes a camera, a microphone, and a fingerprint sensor. Client device **104** is a smartphone, also owned by Alice. Client device **104** includes a camera. Client device **106** is a tablet owned by Bob, and sometimes used by Bob's son Charlie. Client device **106** includes a camera and a fingerprint sensor. Client device **108** is a kiosk located in the lobby of a hotel. Kiosk **108** includes a camera and a microphone. The techniques described herein can be used with or adapted to be used with other devices, as applicable. For example, the techniques can be used in conjunction with gaming systems, with peripheral devices such as mice, and with embedded devices, such as door locks.

Service **120** is a social networking site. Service **122** is a website of a bank. Service **124** is the online store of a boutique camera retailer. Each of services **120-124** requires a username and password (and/or a cookie) from a user prior to giving that user access to protected content and/or other features. As will be described in more detail below, using the techniques described herein, users need not type such usernames and passwords into their devices whenever required by a service. Instead, users can authenticate themselves to an "authentication translator" via an appropriate technique, and the authentication translator will provide the appropriate credentials to the implicated service on the user's behalf. Also as will be described in more detail below, authentication translators can be located in a variety of places within an environment. For example, notebook computer **102** includes an authentication translator module **132** that provides authentication translation services. The other devices **104-108** can also include (but need not include) their own respective authentication translator modules. The owner of bank website **122** also operates an authentication translator **134** associated with the bank. Finally, authentication translator **136** provides authentication translation services to a variety of businesses, including online camera retailer **124**.

FIG. **2** illustrates an embodiment of credential information stored on a device. In particular, device **200** stores three user profiles **202-206**, each of which contains a username and one or more templates (e.g., template **210**) associated with the user. In various embodiments, a template is a collection of biometric features. Using fingerprints as an example type of

biometric, a corresponding template includes a collection of patterns, minutia, and/or other features that can be matched against to determine if a person's fingerprint matches the fingerprint of the registered user (i.e., the owner of a given user profile). A representation of a single fingerprint may be included in multiple templates (e.g., in different resolutions, in accordance with different protocols, as captured during warm vs. cold conditions, and/or by itself or in combination with multiple fingerprints). When other biometrics are employed (e.g., facial recognition, voiceprint, or retina scan technology), features appropriate to those types of biometrics are included in the template. Other types of features can also be included in templates. As one example, a user's typing speed and/or accuracy can be measured by a device, such as device **102**, and used to distinguish between multiple users of a device. For example, suppose Alice types at 100 words per minute and rarely makes mistakes. A representation of this information can be stored in template **212**. Also suppose Alice's niece, who sometimes uses Alice's laptop computer when visiting Alice types at 20 words per minute and makes many mistakes. In some embodiments, the fact that a user was recently (e.g., within the last 5 minutes) typing on laptop **102** at 90 words per minute is evidence of a match against template **212**. In this case, the typing speed of 90 words per minute is similar enough to Alice's typical behavior, it is considered a match. Various policies can be included in a profile that govern how matches are to be performed. For example, policies can specify thresholds/tolerances for what constitutes a match, and can specify that different levels of matches can result in different levels of access to different resources.

A profile is associated with a vault (e.g., vault **220**). The vault, in turn, contains triples specifying a service provider/domain, a username, and a credential. The vault can also contain other sensitive user information, such as account numbers, address/phone number information, and health care data. The credential for a service provider/domain can be a password (e.g., for legacy servers), and can also take alternate forms (e.g., a cryptographic key for service providers supporting stronger authentication methods).

In some embodiments, profiles, templates, and vaults (collectively "authentication information") are stored entirely in an unprotected storage area, and are stored in the clear. In other embodiments, secure storage techniques are used to secure at least a portion of the authentication information.

One example of a device with secure storage is illustrated in FIG. **3**. In the example shown, a mobile phone **300** includes a large and insecure storage **302** attached to a fast processor **304**, and a smaller but secure storage **306** attached to a dedicated processor **308** and a sensor **310** (e.g., a camera or a fingerprint reader). Users (and applications) can read from and write to the insecure storage area. However, users cannot access the secure storage area, and the fast processor can only communicate with the dedicated processor/sensor via a restricted API. As another example, a unique decryption key associated with a given vault can be stored in a profile. The vault is an encrypted and authenticated container that can be stored on insecure storage, e.g., on the device, and also backed up (e.g., to a cloud storage service **140** or to an alternate form of external storage). As needed, authentication information or portions thereof can be loaded into secure storage and decrypted. For example, one can use AES to encrypt the files one by one, using a key stored on the secured storage. A message authentication technique, such as HMAC, can be used for authenticating the encrypted files to provide tamper prevention. Profiles and vaults can be updated while in secure storage; if this occurs, they are encrypted and MACed before being written back to the insecure storage, which may in turn propagate them to external backup storage. In yet other embodiments, profiles and vaults are stored entirely in secure storage, in plaintext, which allows them to be both read and written—and in particular, searched.

Example Transaction Types

A variety of transaction types can take place in the environment shown in FIG. **1**, examples of which are discussed in this section.

Initial Registration

In order to begin using the techniques described herein, users perform some form of initial registration. As one example, suppose Alice launches an enrollment program installed on laptop **102**. She uses the program to capture various biometric information (e.g., fingerprints, photographs of her face, etc.). A user profile is created for Alice, and the biometric information captured about her is encoded into a plurality of templates, such as templates **210** and **214**. In some embodiments, Alice is also explicitly asked to supply credential information for services she would like to use, such as by providing the domain name of social networking site **120**, along with her username and password for site **120**. In other embodiments, domain/username/credential information is at least passively captured on Alice's behalf and included in one or more vaults such as vault **220**. Credential information can also be important from a browser password manager already in use by Alice or other appropriate source. In some embodiments, Alice also registers with cloud storage service **140**, which will allow her to back up her authentication information and to synchronize it across her devices (e.g., **102** and **104**), as described in more detail below.

Other registration approaches can also be used. For example, registration can be integrated into the experience the first time a device is used. Thus, when Bob first turns on tablet **106**, he may be prompted to take a picture of his face (with a profile/templates being created in response). Similarly, the first time Charlie uses tablet **106**, the techniques described herein can be used to determine that Charlie does not yet have a profile (e.g., because none of the templates already present on tablet **106** match his biometrics) and Charlie can be prompted to enroll as a second user of the device.

Authentication

Suppose Alice wishes to authenticate to banking website **122**. Using a fingerprint reader incorporated into her laptop, she performs a fingerprint scan, which causes her biometric features to be extracted and compared to any stored templates residing on her computer. If a match is found, an associated decryption key is selected, and the associated vault is loaded and decrypted. The vault is scanned for an entry that matches the selected service provider (i.e., website **122**). If a matching entry is found, the associated domain, username, and site credential are extracted from the vault. In some embodiments, the validity of the domain name mapping is verified at this point to harden the system against domain name poisoning. Next, a secure connection is established between Alice's computer and the service provider, and Alice is authenticated. For service providers supporting strong user authentication, mutual SSL can be used, for example. A variety of policies can be involved when performing matching. For example, to access certain domains, Alice's print may need only match template **210**. To access other domains, Alice may need to match multiple templates (e.g., both **210** and **214**). As another example, in order to access social networking site **120**, Alice may merely need to be sitting in front of her computer, which has an integrated webcam. Even in relatively low light conditions, a match can be performed against Alice's face and features stored in a template. However, in order to access bank

website **122**, Alice may need a high quality photograph (i.e., requiring her to turn on a bright light) and may need to demonstrate liveness (e.g., by blinking or turning her head). As yet another example, other contextual information can be included in policies. For example, if Alice's IP address indicates she is in a country that she is not usually in, she may be required to match multiple templates (or match a template with more/better quality features) in order to access retailer **124**, as distinguished from when her IP address indicates she is at home.

In some embodiments, the biometric sensor used by a user may be a peripheral device (e.g., a mouse with an integrated fingerprint scanner that is connected to the user's primary device via USB). In such scenarios, the peripheral device may be responsible for storing at least a portion of authentication information and may perform at least some of the authentication tasks previously described as having been performed by Alice's computer. For example, instead of processors **304** and **308**, and storages **302** and **306** being collocated on a single device (e.g., laptop **102**), processor **304** and storage **302** may be present on a primary device, and processor **308** and storage **306** may be present on a peripheral device (e.g., that also includes a sensor, such as a fingerprint reader).

In such scenarios, once Alice's login to banking website **122** is successfully completed, the secure session can be handed over from the peripheral device to the primary device, in a way that does not allow the primary device retroactive access to the plaintext data of the transcripts exchanged between the peripheral device and the service provider. One way this can be accomplished is by renegotiating SSL keys between the peripheral device and the website, after which the newly negotiated key can be handed off from the peripheral device to the primary device. This avoids retroactive credential capture in a setting where the device is infected by malware.

An example of renegotiation is depicted in FIG. **4**. Specifically, after a user has successfully authenticated to a fingerprint reader, a login is performed to a service provider. Using the primary device (**404**) as a proxy, the peripheral fingerprint reader **402** negotiates a first SSL connection (**408**) with a service provider **406**, over which credentials are exchanged. The proxy then renegotiates SSL (**410**), which replaces the old key with a new one. The new key is disclosed to the device, which then seamlessly takes over the connection with the service provider and performs the transaction protected by the authentication. The credentials exchanged during the first SSL connection cannot be accessed by device **404**, since the key of the renegotiated session is independent of the key of the original session; this provides protection against malware residing on the device. Renegotiation can be used when the primary device **404** is believed to be in a safe state when performing the negotiation of the SSL connection, but it is not known whether it is in a safe state during the transaction protected by the authentication. Renegotiation can also be used when a secure component of the primary device **404** performs the negotiation of the SSL connection and another and potentially insecure component of the primary device **404** is involved in the transaction protected by the authentication.

FIG. **5** illustrates an embodiment of a process for performing authentication translation. The process begins at **502** when a request to access a resource is received, as is an authentication input. One example of the processing performed at **502** is as follows. Suppose Alice wishes to sign into social networking website **120**. She directs a web browser application installed on client **102** to the social networking website. Authentication translator module **132** recognizes, from the context of Alice's actions (e.g., that she is attempting to access site **120** with her browser) that she would like to access a particular resource. Authentication translator module **132** prompts Alice (e.g., by a popup message or via a sound) to provide biometric information (e.g., to use the integrated fingerprint reader on her laptop). In some embodiments, the translator module does not prompt Alice, for example, because Alice has been trained to provide biometric information automatically when attempting to access certain resources. In yet other embodiments, the translator module only prompts Alice if she fails to provide acceptable biometric information within a timeout period (e.g., 30 seconds).

Module **132** compares Alice's supplied biometric data to the templates stored on her computer. If a suitable match is found, and if an entry for site **120** is present in the applicable vault, at **504**, a previously stored credential associated with the resource is accessed. In particular, the username and password for the website, as stored in a vault, such as vault **220**, are retrieved from the vault.

Finally, at **506**, the credential is provided to the resource. For example, Alice's username and password for site **120** are transmitted to site **120** at **506**. The credential can be transmitted directly (e.g., by the module or by Alice's computer) and can also be supplied indirectly (e.g., through the use of one or more proxies, routers, or other intermediaries, as applicable).

Other devices can also make use of process **500** or portions thereof. For example, when Alice launches a banking application on phone **104**, implicit in her opening that application is her desire to access the resources of website **122**. The application can take Alice's picture and compare it to stored templates/vault information. If an appropriate match is found, a credential can be retrieved from the vault on her phone (or, e.g., retrieved from cloud storage service **140**) and provided to website **122**.

As another example, suppose Charlie is using tablet **106** and attempts to visit site **120**, whether via a dedicated application or via a web browser application installed on the tablet. Charlie's photograph is taken, and then compared against the profiles stored on tablet **106** (e.g., both Bob and Charlie's profiles). When a determination is made that Charlie's photograph matches a template stored in his stored profile (and not, e.g., Bob's), Charlie's credentials for site **120** are retrieved from a vault and transmitted by an authentication translator module residing on client **106**.

As yet another example, kiosk **108** can be configured to provide certain local resources (e.g., by displaying a company directory or floor plan on demand) when users speak certain requests into a microphone. Enrolled users (e.g., with stored voiceprint or facial recognition features) can be granted access to additional/otherwise restricted services in accordance with the techniques described herein and process **500**.

New Device

In some embodiments, to register a new device, a user provides an identifier, such as a username or an account number to the device. The new device connects to an external storage (such as cloud storage **140**), provides the user identifier and credential, and downloads the user's templates/vaults from the service. In some embodiments, the templates/vaults are encrypted. Once downloaded, the template is decrypted and stored in a secure storage area, while the still encrypted vault can be stored in insecure storage. The decryption key can be generated from information the user has/knows, or from biometric data—such as features extracted from fingerprinting of all ten fingers. In some embodiments, more arduous fingerprinting is required for the setup of a new device than for regular authentication to avoid that a new device gets registered by a user thinking she is merely authenticating—or

worse still, simply touching the device. Moreover, it translates into higher entropy of the decryption keys.

Backup Authentication

Backup authentication allows a user, such as Alice, to access resources in the event she is unable to or unwilling to interact with a particular biometric sensor. As one example, instead of having a single template associated with her profile, Alice can have multiple templates associated with it, e.g., where the first template includes fingerprint features and the second template includes voice biometric, facial recognition, or iris detection features. As a second example, where the service Alice is connecting to is a legacy website (i.e., one that users authenticate to using usernames and passwords), such a service would allow the use of passwords and password reset mechanisms by Alice without requiring Alice to use a fingerprint reader.

In various embodiments, environment **100** supports the ability of users (e.g., under duress) to release the contents of their vaults. For example, if Alice was physically threatened with the loss of a finger by a criminal, Alice could instead release the contents of her vault(s)—the ultimate goal of the criminal. As one example, in the event Alice supplies all 10 fingerprints to the sensor, provides a special password, or supplies a fingerprint and a second identifier, a cleartext version of her vault(s) could be made available.

Access Policies

In various embodiments, cloud storage service **140** is configured to accept backups from multiple devices associated with a single account, and synchronize the updates so that all devices get automatically refreshed. For example, Alice's laptop **102** and phone **104** could both communicate with cloud storage service **140** which would keep their authentication information synchronized. Refreshes can also be made in accordance with user-configured restrictions. For example, Alice's employer could prevent privileged employer data from being stored on shared personal devices, or on any device that was not issued by the employer. As another example, arbitrary policies can be defined regarding the access to and synchronization of software and data, and to tie a license or access rights to a person (and her fingerprint) rather than to a device. As yet another example, in some embodiments (e.g., where a device is made publicly available or otherwise shared by many users), no or a reduced amount of authentication information resides on a device, and at least a portion of authentication information is always retrieved from cloud storage service **140**.

Remote Wiping

Remote wiping of a user's authentication information (e.g., templates) can be used both to "unshare" previously shared devices (e.g., where Bob and Charlie both have user profiles on their shared tablet **106**), and to avoid that criminals with physical component access to lost devices gain access to templates and vault contents. In some embodiments, polices such as ones where a template self-wipes if it is not matched within a particular duration of time are supported. Since user data can be frequently backed up to the cloud storage, and recovered from this using the new device registration process, inconvenience to the user will be minimized.

Legacy Server Support

New authentication schemes typically require changes to a significant codebase residing with service providers. If the code is well written and documented, such changes may be relatively simple. Commonly, though, this may not be so. The engineers who originally wrote the code of relevance may have long since left the company; the code they left behind may be poorly documented—if documented at all. In severe cases, the legacy code may have been written in an outdated programming language or written in a way that does not follow guidelines for good code. This makes updates to the codebase impractical or virtually impossible in many common cases. Even if none of these challenges complicate the desired modifications, it is commonly a great bureaucratic burden to obtain permission to make modifications (e.g., to store an extra field in a backend database), because every affected part of the organization may need to review the request.

As will be described in the following section, the technologies described herein can be used in conjunction with legacy servers (e.g., existing servers that rely on usernames and passwords to authenticate users), and in particular, can be used without requiring modification to such legacy servers.

Cookies

Cookies are commonly used by legacy servers for user authentication. Unfortunately, cookies have several problems. For one thing, they are sometimes deleted—whether explicitly/intentionally by the end user or by the user's software. In addition, cookies are commonly stolen. Approaches such as cache cookies and identification using user agents can be more resistant to these problems, however, they have their own problems. For example, their use requires new code and new fields in the credential database stored by the server.

In some embodiments, authentication translators, such as translators **134** and **136** (also referred to herein as proxies) provide authentication translation services on behalf of associated services. Translators **134** and **136** are illustrated as single logical devices in FIG. **1**. In some embodiments, the translators comprise standard commercially available server hardware (e.g., a multi-core processor, 4+Gigabytes of RAM, and one or more Gigabit network interface adapters) and run typical server-class operating systems (e.g., Linux). Translators **134** and **136** can also be implemented using a scalable, elastic architecture and may comprise several distributed components, including components provided by one or more third parties. For example, translators **134** and **136** may store user credential information or may task cloud storage service **140** with storing at least a portion of that information.

In the case of authentication translator **134**, service is provided with respect to bank website **122** only. Authentication translator **134** is positioned between a legacy web server (**122**) and the Internet (**110**)—and therefore between the legacy server and any client devices. Authentication translator **134** is configured to translate traffic between the legacy server and client devices so that the client devices (and respective users) perceive the new authentication mechanism, while the legacy server remains unchanged. Authentication translator **136** works similarly, but it provides authentication translation as a third party service to multiple providers, an example of which is online camera retailer **124**.

Authentication translators **134** and **136** can also perform process **500**. For example, when a device transmits a request to access website **122**, the request is intercepted by translator **134**, as is cookie/user agent information. The received information can be used to determine a username/password associated with the device, and that information can be passed by translator **134** to website **122** on behalf of the device.

FIG. **6** illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator. The translator (referred to in the figure as a "proxy") fails to identify the client, and passes on the request to the legacy server. The legacy server responds to the request and sets a cookie. The proxy passes on the response, including the cookie and also a cache cookie. The proxy stores the information about both these types of identifiers, along with

the user agent of the client device. This triplet of information is also referred to herein as an identifier.

FIG. **7** illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator. In this scenario, the device's request is accompanied by some form of identifying information. The proxy uses this information to identify the associated cookie and passes this along to the legacy server, along with the request. In some embodiments, additional processing is also involved. For example, reading a cache cookie may require user interaction. Moreover, if not all of the identifying information is present in the request, the proxy can be configured to set the missing information again by sending a corresponding request to the client device.

The translation of cache cookies and user agent information to cookies involves a two-way translation. First, when the legacy server sets a cookie, the proxy will set the two types of cookies—both an HTML cookie and a cache cookie—and then create a new record in which the two cookies are stored, along with the user agent information of the client device. The user agent information can include quite a bit of data associated with a browser—such as the browser type and version, the clock skew, and the fonts that are installed. While each of these pieces of information only contributes a small amount of entropy, the collection of items can be sufficient to identify the device in most cases. Moreover, while some of these types of data may change over time—in fact, all of them may—they do not typically change, and when one or two of them do, the others typically do not. When the client device is used to visit a site controlled by the legacy server, the cookie, cache cookie and user agent information are read (if available), the record identified, and the request translated and sent to the legacy server. When a legacy server requests that the user password is updated (e.g., as part of an annual or other periodic requirement), the transmission of this request to the user can be suppressed—in which case the database of the proxy is updated to create the illusion of an update. The user can be involved in authentication as needed, e.g., if, in addition to supplying a credential, a user must also solve a CAPTCHA, the CAPTCHA can be displayed to the user (with the user's credentials being handled in accordance with the techniques described herein).

FIG. **8** shows the structure of an example of a cache cookie used in some embodiments. Cache cookies can be associated with a particular webpage, just like an HTML cookie can. In the example shown in FIG. **8**, the proxy wishes to associate a page "calling page.html" with a cache cookie. It embeds a request for a second object, "samename.html" in callingpage-.html for every visitor. However, as the cache cookie is set for one visitor, a customized samename.html is served to this visitor. The page samename.html refers to an object with a different name for each user; that object is referred to as "uniquename.jpg." The cache cookie is set by embedding the request for samename.html in callingpage.html. The client browser attempts to render this, causing a request for samename.html from the server. The server configures samename-.html to refer to a uniquely named file uniquename.jpg, and serves samename.html to the client. For the client browser to render samename.html, it requests the file uniquename.jpg, which is intentionally not served. That concludes the setting of a cache cookie. As a user returns to the page callingname-.html, the browser again attempts to render the entire page, which causes it to load the object samename.html from its cache. As that is rendered, the client browser requests uniquename.jpg, which is not in its cache (since it was not served previously). The server still does not serve it, but takes note of the name of the file being requested, as it identifies the client device. Note that samename.html can be displayed in a zero-sized iframe, which makes the end user unaware of it being rendered.

A cache cookie is an implementation of the typical cookie functionality that uses the client device's browser cache. Unlike user agents, it does not change over time, and like standard HTML cookies, it cannot be read by a party other than that which set it. However, like HTML cookies, it could be deleted—by the user clearing his or her browser cache. Cache cookies are not automatically transmitted with GET requests, unless the cache elements are embedded in the referring pages. This adds a potential round of communication in some settings. By relying on user agent information, cache cookies, and HTML cookies to identify the client device, it is much more likely that a machine will be recognized than if only HTML cookies are used.

Although the foregoing embodiments have been described in some detail for purposes of clarity of understanding, the invention is not limited to the details provided. There are many alternative ways of implementing the invention. The disclosed embodiments are illustrative and not restrictive.

What is claimed is:

1. A system, comprising:

a hardware processor configured to:

receive, at an authentication translator, a request to access a resource and an authentication input;

in response to the receiving, perform authentication using the authentication input;

based at least in part on a result of the performed authentication, determine that access should be granted to at least one stored record corresponding to the authentication input that is associated at least with the requested resource;

in response to the determination, access the stored record that is associated at least with the requested resource to obtain previously stored authentication information associated with the resource, wherein the previously stored authentication information includes a credential;

establish a session between the authentication translator and the resource;

cause the obtained credential to be provided to the resource, wherein a user associated with the obtained credential is authenticated to the requested resource using the obtained credential;

facilitate a session renegotiation;

receive a credential change request from the resource; and

update the credential without user input; and

a memory coupled to the hardware processor and configured to provide the hardware processor with instructions.

2. The system of claim **1** wherein the authentication input comprises a biometric input.

3. The system of claim **1** wherein the authentication input is received in response to a user-supplied biometric input matching a template.

4. The system of claim **1** wherein the system comprises a device and wherein the device further includes a biometric input component.

5. The system of claim **1** wherein the request to access the resource and the authentication input are received from a client device that is different and physically separate from the authentication translator.

6. The system of claim **1** wherein the processor is further configured to receive, from a remote system, an encrypted container that includes the credential.

7. The system of claim 1 wherein the processor is further configured to receive, from a remote system, an encrypted container that includes at least one template containing biometric features.

8. The system of claim 1 wherein the authentication input comprises user agent information.

9. The system of claim 1 wherein the authentication input comprises at least one cookie.

10. A method, comprising:

receiving, at an authentication translator, a request to access a resource and an authentication input;

in response to the receiving, performing authentication using the authentication input;

based at least in part on a result of the performed authentication, determining that access should be granted to at least one stored record corresponding to the authentication input that is associated at least with the requested resource;

in response to the determining, accessing the stored record that is associated at least with the requested resource to obtain previously stored authentication information associated with the resource, wherein the previously stored authentication information includes a credential;

establishing a session between the authentication translator and the resource;

causing the obtained credential to be provided to the resource, wherein a user associated with the obtained credential is authenticated to the requested resource using the obtained credential;

facilitating a session renegotiation;

receiving a credential change request from the resource; and

updating the credential without user input.

11. The method of claim 10 further comprising receiving, from a remote system, an encrypted container that includes the credential.

12. The method of claim 10 further comprising receiving, from a remote system, an encrypted container that includes at least one template containing biometric features.

13. A system, comprising:

a hardware processor configured to:

receive, at an authentication translator, a first authentication information, wherein the first authentication information includes a biometric input;

perform authentication using the biometric input, wherein performing the authentication includes determining whether the biometric input matches to one or more templates in accordance with at least one policy;

based at least in part on a result of the authentication performed using the biometric input, determine whether to provide unlocked access to one or more of a plurality of entries associated with a first user profile;

wherein a first entry in the plurality of entries includes an identifier of a resource and a second authentication information, wherein the second authentication information includes a credential associated with the resource, and wherein the credential is used to authenticate a user associated with the credential to the resource; and

wherein access to the first entry in the plurality of entries associated with the first user profile is allowed based on the determination; and

wherein access to a second entry in the plurality of entries associated with the first user profile is disallowed based on the determination;

establishing a session between the authentication translator and the resource;

facilitating a session renegotiation;

receiving a credential change request from the resource; and

updating the credential without user input; and

a memory coupled to the hardware processor and configured to provide the hardware processor with instructions.

14. The system of claim 1 wherein the processor is further configured to receive, at the authentication translator, a second request to access a second resource and a second authentication input that is different from the received authentication input.

* * * * *

Electronic Acknowledgement Receipt

EFS ID:	14394505
Application Number:	13706254
International Application Number:	
Confirmation Number:	4519
Title of Invention:	AUTHENTICATION TRANSLATION
First Named Inventor/Applicant Name:	Bjorn Markus Jakobsson
Customer Number:	21912
Filer:	Robyn Erinn Wagner/Monique Huang
Filer Authorized By:	Robyn Erinn Wagner
Attorney Docket Number:	MJAKP008
Receipt Date:	05-DEC-2012
Filing Date:	
Time Stamp:	19:49:40
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Application Data Sheet	MJAKP008_ADS.pdf	1063422 [illegible]	no	5

Warnings:

Information:

2		MJAKP008_APP.pdf	147380 [illegible]	yes	21
	Multipart Description/PDF files in .zip description				
	Document Description		**Start**	**End**	
	Specification		1	17	
	Claims		18	20	
	Abstract		21	21	
Warnings:					
Information:					
3	Drawings-only black and white line drawings	MJAKP008_APP_Figures.pdf	112090 [illegible]	no	8
Warnings:					
Information:					
		Total Files Size (in bytes):	1322892		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Attorney Docket No. MJAKP008

APPLICATION FOR UNITED STATES PATENT

AUTHENTICATION TRANSLATION

By Inventor:

Bjorn Markus Jakobsson
Mountain View, CA
A Citizen of SW

Assignee: Extricatus LLC

VAN PELT, YI & JAMES LLP
10050 N. Foothill Blvd., Suite 200
Cupertino, CA 95014
Telephone (408) 973-2585

AUTHENTICATION TRANSLATION

CROSS REFERENCE TO OTHER APPLICATIONS

[0001] This application claims priority to U.S. Provisional Patent Application No. 61/569,112 entitled BACKWARDS COMPATIBLE ROBUST COOKIES filed December 9, 2011, and also claims priority to U.S. Provisional Patent Application No. 61/587,387 entitled BIOMETRICS-SUPPORTED SECURE AUTHENTICATION SYSTEM filed January 17, 2012, both of which are incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

[0002] Providing credentials to a service, whether via a mobile or other device, is often a tedious experience for a user. Unfortunately, to make authentication easier for themselves, users will often engage in practices such as password re-use, and/or the selection of poor quality passwords, which render their credentials less secure against attacks. Accordingly, improvements in authentication techniques would be desirable. Further, it would be desirable for such improvements to be widely deployable, including on existing/legacy systems.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] Various embodiments of the invention are disclosed in the following detailed description and the accompanying drawings.

[0004] Figure 1 illustrates an embodiment of an environment in which authentication translation is provided.

[0005] Figure 2 illustrates an embodiment of credential information stored on a device.

[0006] Figure 3 illustrates an embodiment of a device with secure storage.

[0007] Figure 4 illustrates an example of a renegotiation.

[0008] Figure 5 illustrates an embodiment of a process for performing authentication translation.

[0009] Figure 6 illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator.

[0010] Figure 7 illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator.

[0011] Figure 8 shows the structure of an example of a cache cookie used in some embodiments.

DETAILED DESCRIPTION

[0012] The invention can be implemented in numerous ways, including as a process; an apparatus; a system; a composition of matter; a computer program product embodied on a computer readable storage medium; and/or a processor, such as a processor configured to execute instructions stored on and/or provided by a memory coupled to the processor. In this specification, these implementations, or any other form that the invention may take, may be referred to as techniques. In general, the order of the steps of disclosed processes may be altered within the scope of the invention. Unless stated otherwise, a component such as a processor or a memory described as being configured to perform a task may be implemented as a general component that is temporarily configured to perform the task at a given time or a specific component that is manufactured to perform the task. As used herein, the term 'processor' refers to one or more devices, circuits, and/or processing cores configured to process data, such as computer program instructions.

[0013] A detailed description of one or more embodiments of the invention is provided below along with accompanying figures that illustrate the principles of the invention. The invention is described in connection with such embodiments, but the invention is not limited to any embodiment. The scope of the invention is limited only by the claims and the invention encompasses numerous alternatives, modifications and equivalents. Numerous specific details are set forth in the following description in order to provide a thorough understanding of the invention. These details are provided for the purpose of example and the invention may be practiced according to the claims without some or all of these specific details. For the purpose of clarity, technical material that is known in the technical fields related to the invention has not been described in detail so that the invention is not unnecessarily obscured.

[0014] Figure 1 illustrates an embodiment of an environment in which authentication translation is provided. In the example shown, a variety of client devices 102-108 connect, via one or more networks (represented as a single network cloud 110) to a variety of services 120-124 (also referred to herein as sites 120-124). In particular, client device 102 is a notebook computer owned by a user hereinafter referred to as Alice. Notebook 102 includes a camera, a microphone, and a fingerprint sensor. Client device 104 is a smartphone, also owned by Alice.

Client device 104 includes a camera. Client device 106 is a tablet owned by Bob, and sometimes used by Bob's son Charlie. Client device 106 includes a camera and a fingerprint sensor. Client device 108 is a kiosk located in the lobby of a hotel. Kiosk 108 includes a camera and a microphone. The techniques described herein can be used with or adapted to be used with other devices, as applicable. For example, the techniques can be used in conjunction with gaming systems, with peripheral devices such as mice, and with embedded devices, such as door locks.

[0015] Service 120 is a social networking site. Service 122 is a website of a bank. Service 124 is the online store of a boutique camera retailer. Each of services 120-124 requires a username and password (and/or a cookie) from a user prior to giving that user access to protected content and/or other features. As will be described in more detail below, using the techniques described herein, users need not type such usernames and passwords into their devices whenever required by a service. Instead, users can authenticate themselves to an "authentication translator" via an appropriate technique, and the authentication translator will provide the appropriate credentials to the implicated service on the user's behalf. Also as will be described in more detail below, authentication translators can be located in a variety of places within an environment. For example, notebook computer 102 includes an authentication translator module 132 that provides authentication translation services. The other devices 104-108 can also include (but need not include) their own respective authentication translator modules. The owner of bank website 122 also operates an authentication translator 134 associated with the bank. Finally, authentication translator 136 provides authentication translation services to a variety of businesses, including online camera retailer 124.

[0016] Figure 2 illustrates an embodiment of credential information stored on a device. In particular, device 200 stores three user profiles 202-206, each of which contains a username and one or more templates (e.g., template 210) associated with the user. In various embodiments, a template is a collection of biometric features. Using fingerprints as an example type of biometric, a corresponding template includes a collection of patterns, minutia, and/or other features that can be matched against to determine if a person's fingerprint matches the fingerprint of the registered user (i.e., the owner of a given user profile). A representation of a single fingerprint may be included in multiple templates (e.g., in different resolutions, in accordance with different protocols, as captured during warm vs. cold conditions, and/or by itself

or in combination with multiple fingerprints). When other biometrics are employed (e.g., facial recognition, voiceprint, or retina scan technology), features appropriate to those types of biometrics are included in the template. Other types of features can also be included in templates. As one example, a user's typing speed and/or accuracy can be measured by a device, such as device 102, and used to distinguish between multiple users of a device. For example, suppose Alice types at 100 words per minute and rarely makes mistakes. A representation of this information can be stored in template 212. Also suppose Alice's niece, who sometimes uses Alice's laptop computer when visiting Alice types at 20 words per minute and makes many mistakes. In some embodiments, the fact that a user was recently (e.g., within the last 5 minutes) typing on laptop 102 at 90 words per minute is evidence of a match against template 212. In this case, the typing speed of 90 words per minute is similar enough to Alice's typical behavior, it is considered a match. Various policies can be included in a profile that govern how matches are to be performed. For example, policies can specify thresholds/tolerances for what constitutes a match, and can specify that different levels of matches can result in different levels of access to different resources.

[0017] A profile is associated with a vault (e.g., vault 220). The vault, in turn, contains triples specifying a service provider/domain, a username, and a credential. The vault can also contain other sensitive user information, such as account numbers, address/phone number information, and health care data. The credential for a service provider/domain can be a password (e.g., for legacy servers), and can also take alternate forms (e.g., a cryptographic key for service providers supporting stronger authentication methods).

[0018] In some embodiments, profiles, templates, and vaults (collectively "authentication information") are stored entirely in an unprotected storage area, and are stored in the clear. In other embodiments, secure storage techniques are used to secure at least a portion of the authentication information.

[0019] One example of a device with secure storage is illustrated in Figure 3. In the example shown, a mobile phone 300 includes a large and insecure storage 302 attached to a fast processor 304, and a smaller but secure storage 306 attached to a dedicated processor 308 and a sensor 310 (e.g., a camera or a fingerprint reader). Users (and applications) can read from and

write to the insecure storage area. However, users cannot access the secure storage area, and the fast processor can only communicate with the dedicated processor/sensor via a restricted API. As another example, a unique decryption key associated with a given vault can be stored in a profile. The vault is an encrypted and authenticated container that can be stored on insecure storage, e.g., on the device, and also backed up (e.g., to a cloud storage service 140 or to an alternate form of external storage). As needed, authentication information or portions thereof can be loaded into secure storage and decrypted. For example, one can use AES to encrypt the files one by one, using a key stored on the secured storage. A message authentication technique, such as HMAC, can be used for authenticating the encrypted files to provide tamper prevention. Profiles and vaults can be updated while in secure storage; if this occurs, they are encrypted and MACed before being written back to the insecure storage, which may in turn propagate them to external backup storage. In yet other embodiments, profiles and vaults are stored entirely in secure storage, in plaintext, which allows them to be both read and written -- and in particular, searched.

[0020] Example Transaction Types

[0021] A variety of transaction types can take place in the environment shown in Figure 1, examples of which are discussed in this section.

[0022] Initial Registration

[0023] In order to begin using the techniques described herein, users perform some form of initial registration. As one example, suppose Alice launches an enrollment program installed on laptop 102. She uses the program to capture various biometric information (e.g., fingerprints, photographs of her face, etc.). A user profile is created for Alice, and the biometric information captured about her is encoded into a plurality of templates, such as templates 210 and 214. In some embodiments, Alice is also explicitly asked to supply credential information for services she would like to use, such as by providing the domain name of social networking site 120, along with her username and password for site 120. In other embodiments, domain/username/credential information is at least passively captured on Alice's behalf and included in one or more vaults such as vault 220. Credential information can also be important from a browser password manager already in use by Alice or other appropriate source. In some

embodiments, Alice also registers with cloud storage service 140, which will allow her to back up her authentication information and to synchronize it across her devices (e.g., 102 and 104), as described in more detail below.

[0024] Other registration approaches can also be used. For example, registration can be integrated into the experience the first time a device is used. Thus, when Bob first turns on tablet 106, he may be prompted to take a picture of his face (with a profile/templates being created in response). Similarly, the first time Charlie uses tablet 106, the techniques described herein can be used to determine that Charlie does not yet have a profile (e.g., because none of the templates already present on tablet 106 match his biometrics) and Charlie can be prompted to enroll as a second user of the device.

[0025] Authentication

[0026] Suppose Alice wishes to authenticate to banking website 122. Using a fingerprint reader incorporated into her laptop, she performs a fingerprint scan, which causes her biometric features to be extracted and compared to any stored templates residing on her computer. If a match is found, an associated decryption key is selected, and the associated vault is loaded and decrypted. The vault is scanned for an entry that matches the selected service provider (i.e., website 122). If a matching entry is found, the associated domain, username, and site credential are extracted from the vault. In some embodiments, the validity of the domain name mapping is verified at this point to harden the system against domain name poisoning. Next, a secure connection is established between Alice's computer and the service provider, and Alice is authenticated. For service providers supporting strong user authentication, mutual SSL can be used, for example. A variety of policies can be involved when performing matching. For example, to access certain domains, Alice's print may need only match template 210. To access other domains, Alice may need to match multiple templates (e.g., both 210 and 214). As another example, in order to access social networking site 120, Alice may merely need to be sitting in front of her computer, which has an integrated webcam. Even in relatively low light conditions, a match can be performed against Alice's face and features stored in a template. However, in order to access bank website 122, Alice may need a high quality photograph (i.e., requiring her to turn on a bright light) and may need to demonstrate liveness (e.g., by blinking or turning her

head). As yet another example, other contextual information can be included in policies. For example, if Alice's IP address indicates she is in a country that she is not usually in, she may be required to match multiple templates (or match a template with more/better quality features) in order to access retailer 124, as distinguished from when her IP address indicates she is at home.

[0027] In some embodiments, the biometric sensor used by a user may be a peripheral device (e.g., a mouse with an integrated fingerprint scanner that is connected to the user's primary device via USB). In such scenarios, the peripheral device may be responsible for storing at least a portion of authentication information and may perform at least some of the authentication tasks previously described as having been performed by Alice's computer. For example, instead of processors 304 and 308, and storages 302 and 306 being collocated on a single device (e.g., laptop 102), processor 304 and storage 302 may be present on a primary device, and processor 308 and storage 306 may be present on a peripheral device (e.g., that also includes a sensor, such as a fingerprint reader).

[0028] In such scenarios, once Alice's login to banking website 122 is successfully completed, the secure session can be handed over from the peripheral device to the primary device, in a way that does not allow the primary device retroactive access to the plaintext data of the transcripts exchanged between the peripheral device and the service provider. One way this can be accomplished is by renegotiating SSL keys between the peripheral device and the website, after which the newly negotiated key can be handed off from the peripheral device to the primary device. This avoids retroactive credential capture in a setting where the device is infected by malware.

[0029] An example of renegotiation is depicted in Figure 4. Specifically, after a user has successfully authenticated to a fingerprint reader, a login is performed to a service provider. Using the primary device (404) as a proxy, the peripheral fingerprint reader 402 negotiates a first SSL connection (408) with a service provider 406, over which credentials are exchanged. The proxy then renegotiates SSL (410), which replaces the old key with a new one. The new key is disclosed to the device, which then seamlessly takes over the connection with the service provider and performs the transaction protected by the authentication. The credentials exchanged during the first SSL connection cannot be accessed by device 404, since the key of the

renegotiated session is independent of the key of the original session; this provides protection against malware residing on the device. Renegotiation can be used when the primary device 404 is believed to be in a safe state when performing the negotiation of the SSL connection, but it is not known whether it is in a safe state during the transaction protected by the authentication. Renegotiation can also be used when a secure component of the primary device 404 performs the negotiation of the SSL connection and another and potentially insecure component of the primary device 404 is involved in the transaction protected by the authentication.

[0030] Figure 5 illustrates an embodiment of a process for performing authentication translation. The process begins at 502 when a request to access a resource is received, as is an authentication input. One example of the processing performed at 502 is as follows. Suppose Alice wishes to sign into social networking website 120. She directs a web browser application installed on client 102 to the social networking website. Authentication translator module 132 recognizes, from the context of Alice's actions (e.g., that she is attempting to access site 120 with her browser) that she would like to access a particular resource. Authentication translator module 132 prompts Alice (e.g., by a popup message or via a sound) to provide biometric information (e.g., to use the integrated fingerprint reader on her laptop). In some embodiments, the translator module does not prompt Alice, for example, because Alice has been trained to provide biometric information automatically when attempting to access certain resources. In yet other embodiments, the translator module only prompts Alice if she fails to provide acceptable biometric information within a timeout period (e.g., 30 seconds).

[0031] Module 132 compares Alice's supplied biometric data to the templates stored on her computer. If a suitable match is found, and if an entry for site 120 is present in the applicable vault, at 504, a previously stored credential associated with the resource is accessed. In particular, the username and password for the website, as stored in a vault, such as vault 220, are retrieved from the vault.

[0032] Finally, at 506, the credential is provided to the resource. For example, Alice's username and password for site 120 are transmitted to site 120 at 506. The credential can be transmitted directly (e.g., by the module or by Alice's computer) and can also be supplied

indirectly (e.g., through the use of one or more proxies, routers, or other intermediaries, as applicable).

[0033] Other devices can also make use of process 500 or portions thereof. For example, when Alice launches a banking application on phone 104, implicit in her opening that application is her desire to access the resources of website 134. The application can take Alice's picture and compare it to stored templates/vault information. If an appropriate match is found, a credential can be retrieved from the vault on her phone (or, e.g., retrieved from cloud storage service 140) and provided to website 134.

[0034] As another example, suppose Charlie is using tablet 106 and attempts to visit site 120, whether via a dedicated application or via a web browser application installed on the tablet. Charlie's photograph is taken, and then compared against the profiles stored on tablet 106 (e.g., both Bob and Charlie's profiles). When a determination is made that Charlie's photograph matches a template stored in his stored profile (and not, e.g., Bob's), Charlie's credentials for site 120 are retrieved from a vault and transmitted by an authentication translator module residing on client 106.

[0035] As yet another example, kiosk 108 can be configured to provide certain local resources (e.g., by displaying a company directory or floor plan on demand) when users speak certain requests into a microphone. Enrolled users (e.g., with stored voiceprint or facial recognition features) can be granted access to additional/otherwise restricted services in accordance with the techniques described herein and process 500.

[0036] New device

[0037] In some embodiments, to register a new device, a user provides an identifier, such as a username or an account number to the device. The new device connects to an external storage (such as cloud storage 140), provides the user identifier and credential, and downloads the user's templates/vaults from the service. In some embodiments, the templates/vaults are encrypted. Once downloaded, the template is decrypted and stored in a secure storage area, while the still encrypted vault can be stored in insecure storage. The decryption key can be generated from information the user has/knows, or from biometric data -- such as features

extracted from fingerprinting of all ten fingers. In some embodiments, more arduous fingerprinting is required for the setup of a new device than for regular authentication to avoid that a new device gets registered by a user thinking she is merely authenticating -- or worse still, simply touching the device. Moreover, it translates into higher entropy of the decryption keys.

[0038] Backup Authentication

[0039] Backup authentication allows a user, such as Alice, to access resources in the event she is unable to or unwilling to interact with a particular biometric sensor. As one example, instead of having a single template associated with her profile, Alice can have multiple templates associated with it, e.g., where the first template includes fingerprint features and the second template includes voice biometric, facial recognition, or iris detection features. As a second example, where the service Alice is connecting to is a legacy website (i.e., one that users authenticate to using usernames and passwords), such a service would allow the use of passwords and password reset mechanisms by Alice without requiring Alice to use a fingerprint reader.

[0040] In various embodiments, environment 100 supports the ability of users (e.g., under duress) to release the contents of their vaults. For example, if Alice was physically threatened with the loss of a finger by a criminal, Alice could instead release the contents of her vault(s) – the ultimate goal of the criminal. As one example, in the event Alice supplies all 10 fingerprints to the sensor, provides a special password, or supplies a fingerprint and a second identifier, a cleartext version of her vault(s) could be made available.

[0041] Access Policies

[0042] In various embodiments, cloud storage service 140 is configured to accept backups from multiple devices associated with a single account, and synchronize the updates so that all devices get automatically refreshed. For example, Alice's laptop 102 and phone 104 could both communicate with cloud storage service 140 which would keep their authentication information synchronized. Refreshes can also be made in accordance with user-configured restrictions. For example, Alice's employer could prevent privileged employer data from being stored on shared personal devices, or on any device that was not issued by the employer. As

another example, arbitrary policies can be defined regarding the access to and synchronization of software and data, and to tie a license or access rights to a person (and her fingerprint) rather than to a device. As yet another example, in some embodiments (e.g., where a device is made publicly available or otherwise shared by many users), no or a reduced amount of authentication information resides on a device, and at least a portion of authentication information is always retrieved from cloud storage service 140.

[0043] Remote wiping

[0044] Remote wiping of a user's authentication information (e.g., templates) can be used both to "unshare" previously shared devices (e.g., where Bob and Charlie both have user profiles on their shared tablet 106), and to avoid that criminals with physical component access to lost devices gain access to templates and vault contents. In some embodiments, polices such as ones where a template self-wipes if it is not matched within a particular duration of time are supported. Since user data can be frequently backed up to the cloud storage, and recovered from this using the new device registration process, inconvenience to the user will be minimized.

[0045] **Legacy Server Support**

[0046] New authentication schemes typically require changes to a significant codebase residing with service providers. If the code is well written and documented, such changes may be relatively simple. Commonly, though, this may not be so. The engineers who originally wrote the code of relevance may have long since left the company; the code they left behind may be poorly documented -- if documented at all. In severe cases, the legacy code may have been written in an outdated programming language or written in a way that does not follow guidelines for good code. This makes updates to the codebase impractical or virtually impossible in many common cases. Even if none of these challenges complicate the desired modifications, it is commonly a great bureaucratic burden to obtain permission to make modifications (e.g., to store an extra field in a backend database), because every affected part of the organization may need to review the request.

[0047] As will be described in the following section, the technologies described herein can be used in conjunction with legacy servers (e.g., existing servers that rely on usernames and

passwords to authenticate users), and in particular, can be used without requiring modification to such legacy servers.

[0048] Cookies

[0049] Cookies are commonly used by legacy servers for user authentication. Unfortunately, cookies have several problems. For one thing, they are sometimes deleted -- whether explicitly/intentionally by the end user or by the user's software. In addition, cookies are commonly stolen. Approaches such as cache cookies and identification using user agents can be more resistant to these problems, however, they have their own problems. For example, their use requires new code and new fields in the credential database stored by the server.

[0050] In some embodiments, authentication translators, such as translators 134 and 136 (also referred to herein as proxies) provide authentication translation services on behalf of associated services. Translators 134 and 136 are illustrated as single logical devices in Figure 1. In some embodiments, the translators comprise standard commercially available server hardware (e.g., a multi-core processor, 4+ Gigabytes of RAM, and one or more Gigabit network interface adapters) and run typical server-class operating systems (e.g., Linux). Translators 134 and 136 can also be implemented using a scalable, elastic architecture and may comprise several distributed components, including components provided by one or more third parties. For example, translators 134 and 136 may store user credential information or may task cloud storage service 140 with storing at least a portion of that information.

[0051] In the case of authentication translator 134, service is provided with respect to bank website 122 only. Authentication translator 134 is positioned between a legacy web server (122) and the Internet (110) -- and therefore between the legacy server and any client devices. Authentication translator 134 is configured to translate traffic between the legacy server and client devices so that the client devices (and respective users) perceive the new authentication mechanism, while the legacy server remains unchanged. Authentication translator 136 works similarly, but it provides authentication translation as a third party service to multiple providers, an example of which is online camera retailer 124.

[0052] Authentication translators 134 and 136 can also perform process 500. For example, when a device transmits a request to access website 122, the request is intercepted by translator 134, as is cookie/user agent information. The received information can be used to determine a username/password associated with the device, and that information can be passed by translator 134 to website 122 on behalf of the device.

[0053] Figure 6 illustrates an example of what occurs when a client device first visits the site of a legacy server via an authentication translator. The translator (referred to in the figure as a "proxy") fails to identify the client, and passes on the request to the legacy server. The legacy server responds to the request and sets a cookie. The proxy passes on the response, including the cookie and also a cache cookie. The proxy stores the information about both these types of identifiers, along with the user agent of the client device. This triplet of information is also referred to herein as an identifier.

[0054] Figure 7 illustrates an example of what occurs when a device subsequently visits the site of a legacy server via an authentication translator. In this scenario, the device's request is accompanied by some form of identifying information. The proxy uses this information to identify the associated cookie and passes this along to the legacy server, along with the request. In some embodiments, additional processing is also involved. For example, reading a cache cookie may require user interaction. Moreover, if not all of the identifying information is present in the request, the proxy can be configured to set the missing information again by sending a corresponding request to the client device.

[0055] The translation of cache cookies and user agent information to cookies involves a two-way translation. First, when the legacy server sets a cookie, the proxy will set the two types of cookies -- both an HTML cookie and a cache cookie -- and then create a new record in which the two cookies are stored, along with the user agent information of the client device. The user agent information can include quite a bit of data associated with a browser -- such as the browser type and version, the clock skew, and the fonts that are installed. While each of these pieces of information only contributes a small amount of entropy, the collection of items can be sufficient to identify the device in most cases. Moreover, while some of these types of data may change over time -- in fact, all of them may -- they do not typically change, and when one or two of

them do, the others typically do not. When the client device is used to visit a site controlled by the legacy server, the cookie, cache cookie and user agent information are read (if available), the record identified, and the request translated and sent to the legacy server. When a legacy server requests that the user password is updated (e.g., as part of an annual or other periodic requirement), the transmission of this request to the user can be suppressed -- in which case the database of the proxy is updated to create the illusion of an update. The user can be involved in authentication as needed, e.g., if, in addition to supplying a credential, a user must also solve a CAPTCHA, the CAPTCHA can be displayed to the user (with the user's credentials being handled in accordance with the techniques described herein).

[0056] Figure 8 shows the structure of an example of a cache cookie used in some embodiments. Cache cookies can be associated with a particular webpage, just like an HTML cookie can. In the example shown in Figure 8, the proxy wishes to associate a page "calling page.html" with a cache cookie. It embeds a request for a second object, "samename.html" in callingpage.html for every visitor. However, as the cache cookie is set for one visitor, a customized samename.html is served to this visitor. The page samename.html refers to an object with a different name for each user; that object is referred to as "uniquename.jpg." The cache cookie is set by embedding the request for samename.html in callingpage.html. The client browser attempts to render this, causing a request for samename.html from the server. The server configures samename.html to refer to a uniquely named file uniquename.jpg, and serves samename.html to the client. For the client browser to render samename.html, it requests the file uniquename.jpg, which is intentionally not served. That concludes the setting of a cache cookie. As a user returns to the page callingname.html, the browser again attempts to render the entire page, which causes it to load the object samename.html from its cache. As that is rendered, the client browser requests uniquename.jpg, which is not in its cache (since it was not served previously). The server still does not serve it, but takes note of the name of the file being requested, as it identifies the client device. Note that samename.html can be displayed in a zero-sized iframe, which makes the end user unaware of it being rendered.

[0057] A cache cookie is an implementation of the typical cookie functionality that uses the client device's browser cache. Unlike user agents, it does not change over time, and like standard HTML cookies, it cannot be read by a party other than that which set it. However, like

HTML cookies, it could be deleted -- by the user clearing his or her browser cache. Cache cookies are not automatically transmitted with GET requests, unless the cache elements are embedded in the referring pages. This adds a potential round of communication in some settings. By relying on user agent information, cache cookies, and HTML cookies to identify the client device, it is much more likely that a machine will be recognized than if only HTML cookies are used.

[0058] Although the foregoing embodiments have been described in some detail for purposes of clarity of understanding, the invention is not limited to the details provided. There are many alternative ways of implementing the invention. The disclosed embodiments are illustrative and not restrictive.

[0059] WHAT IS CLAIMED IS:

CLAIMS

1. A system, comprising:

a processor configured to:

receive, at an authentication translator, a request to access a resource and an authentication input, wherein the authentication input corresponds to at least one stored record and wherein the stored record is associated at least with the resource;

in response to the receiving, access a previously stored credential associated with the resource; and

cause the credential to be provided to the resource; and

a memory coupled to the processor and configured to provide the processor with instructions.

2. The system of claim 1 wherein the authentication input comprises a biometric input.

3. The system of claim 1 wherein the authentication input is received in response to a user-supplied biometric input matching a template.

4. The system of claim 1 wherein the system comprises a device and wherein the device further includes a biometric input component.

5. The system of claim 1 wherein the request to access the resource and the authentication input are received from a client device that is different and physically separate from the authentication translator.

6. The system of claim 1 wherein the processor is further configured to establish a session between the authentication translator and the resource.

7. The system of claim 6 wherein the processor is further configured to facilitate a session renegotiation.

8. The system of claim 1 wherein the processor is further configured to receive, from a remote system, an encrypted container that includes the credential.

9. The system of claim 1 wherein the processor is further configured to receive, from a remote system, an encrypted container that includes at least one template containing biometric features.

10. The system of claim 1 wherein the authentication input comprises user agent information.

11. The system of claim 1 wherein the authentication input comprises at least one cookie.

12. The system of claim 1 wherein the processor is further configured to receive a password change request from the resource and wherein the processor is configured to update the credential without user input.

13. A method, comprising:

receiving, at an authentication translator, a request to access a resource and an authentication input, wherein the authentication input corresponds to at least one stored record and wherein the stored record is associated at least with the resource;

in response to the receiving, accessing a previously stored credential associated with the resource; and

causing the credential to be provided to the resource.

14. The method of claim 13 further comprising establishing a session between the authentication translator and the resource.

15. The method of claim 13 further comprising facilitating a session renegotiation.

16. The method of claim 13 further comprising receiving, from a remote system, an encrypted container that includes the credential.

17. The method of claim 13 further comprising receiving, from a remote system, an encrypted container that includes at least one template containing biometric features.

18. The method of claim 13 further comprising receiving a password change request from the resource and updating the credential without user input.

19. A system, comprising:

a processor configured to:

receive a biometric input; and

determine whether to provide unlocked access to one or more of a plurality of entries associated with a first user profile based at least on the biometric input matching one or more templates in accordance with at least one policy;

wherein an entry includes an identifier of a resource and a credential associated with the resource; and

wherein access to a first entry is allowed and access to a second entry is disallowed based on the determination; and

a memory coupled to the processor and configured to provide the processor with instructions.

20. The system of claim 19 wherein the processor is further configured to establish a connection to the resource.

ABSTRACT OF THE DISCLOSURE

Authentication translation is disclosed. A request to access a resource is received at an authentication translator, as is an authentication input. The authentication input corresponds to at least one stored record. The stored record is associated at least with the resource. In response to the receiving, a previously stored credential associated with the resource is accessed. The credential is provided to the resource.

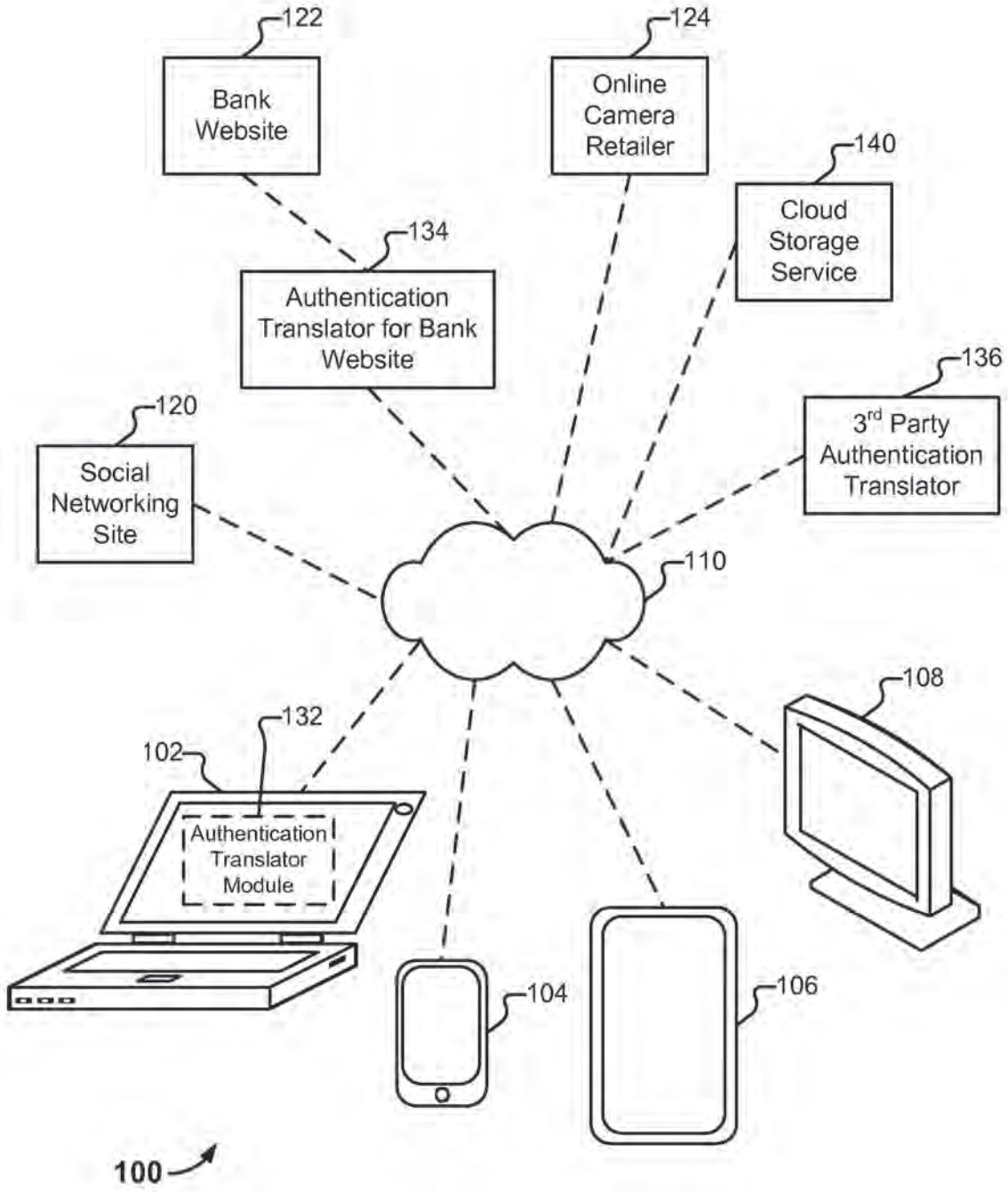

122
Bank Website
124
Online Camera Retailer
140
Cloud Storage Service
134
Authentication Translator for Bank Website
136
3rd Party Authentication Translator
120
Social Networking Site
110
108
132
102
Authentication Translator Module
104
106
100

FIG. 1

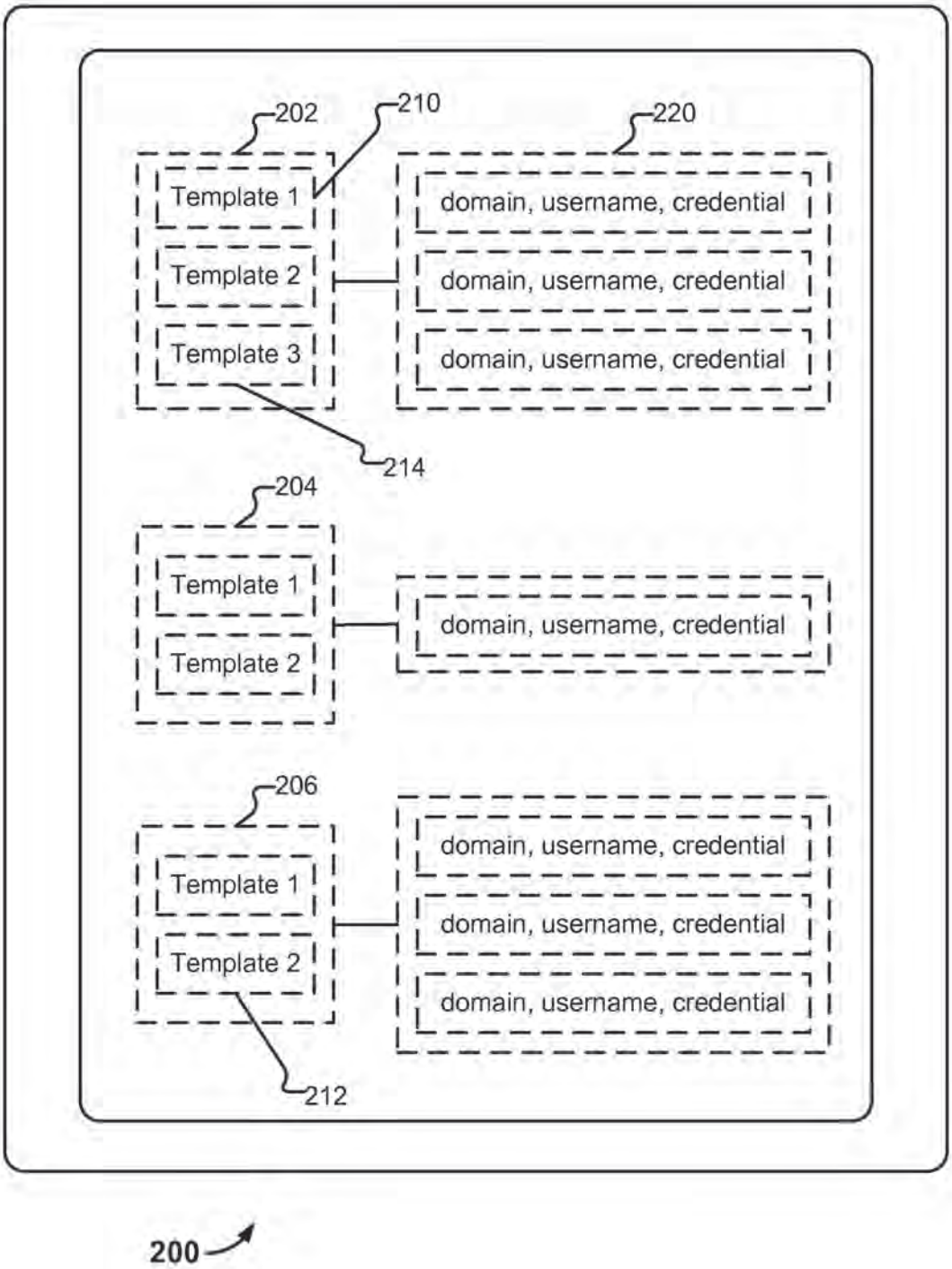
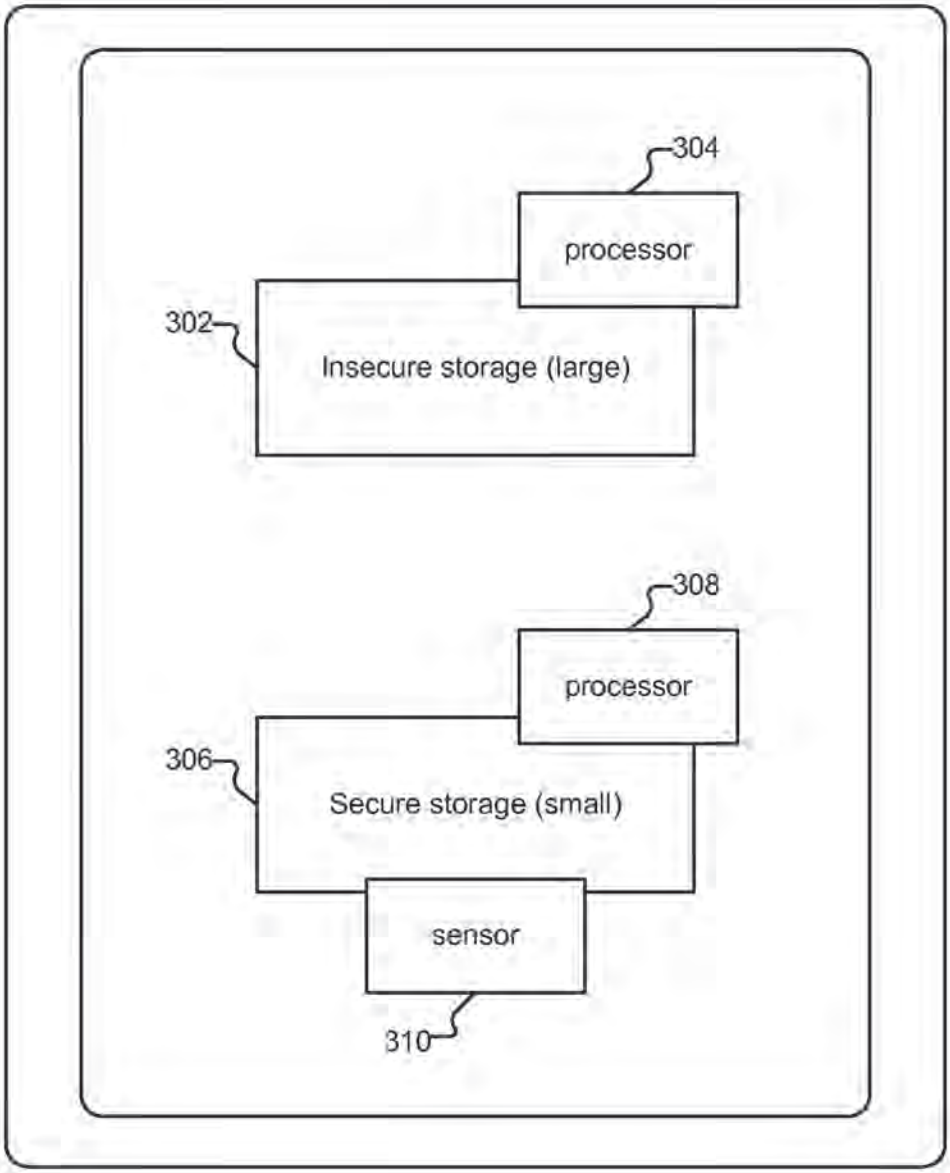

202
210
220
Template 1
Template 2
Template 3
domain, username, credential
domain, username, credential
domain, username, credential
214
204
Template 1
Template 2
domain, username, credential
206
Template 1
Template 2
domain, username, credential
domain, username, credential
domain, username, credential
212
200

FIG. 2


304
processor
302
Insecure storage (large)
308
processor
306
Secure storage (small)
sensor
310

300

FIG. 3

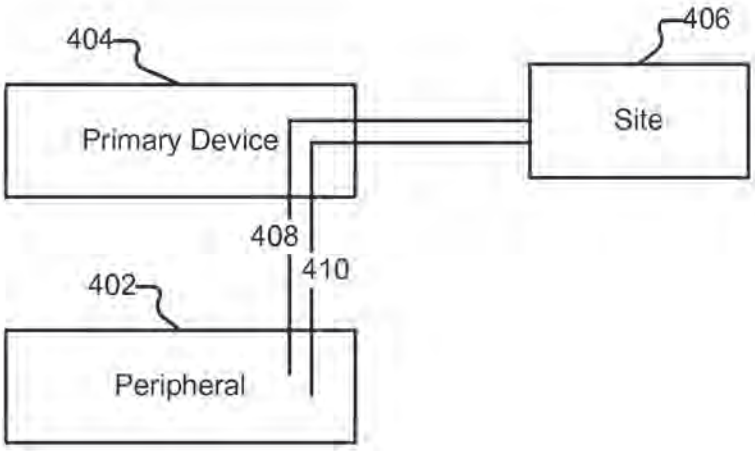


FIG. 4

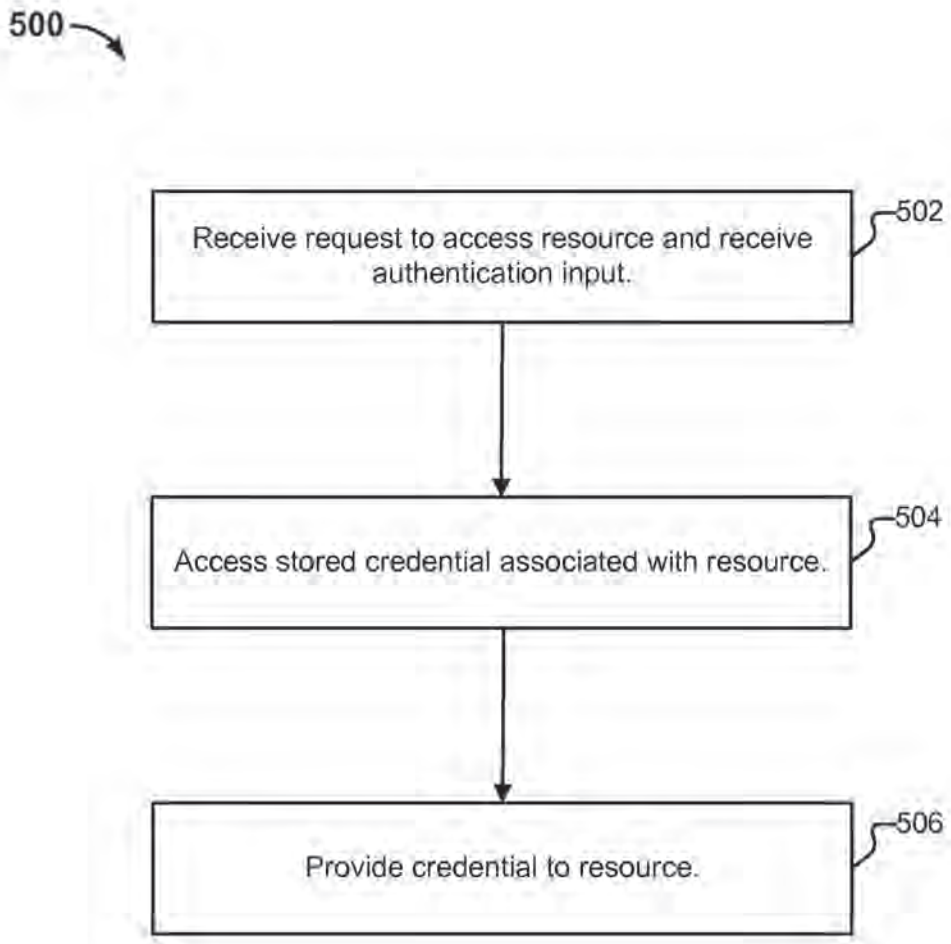

500
Receive request to access resource and receive authentication input.
502
Access stored credential associated with resource.
504
Provide credential to resource.
506

FIG. 5

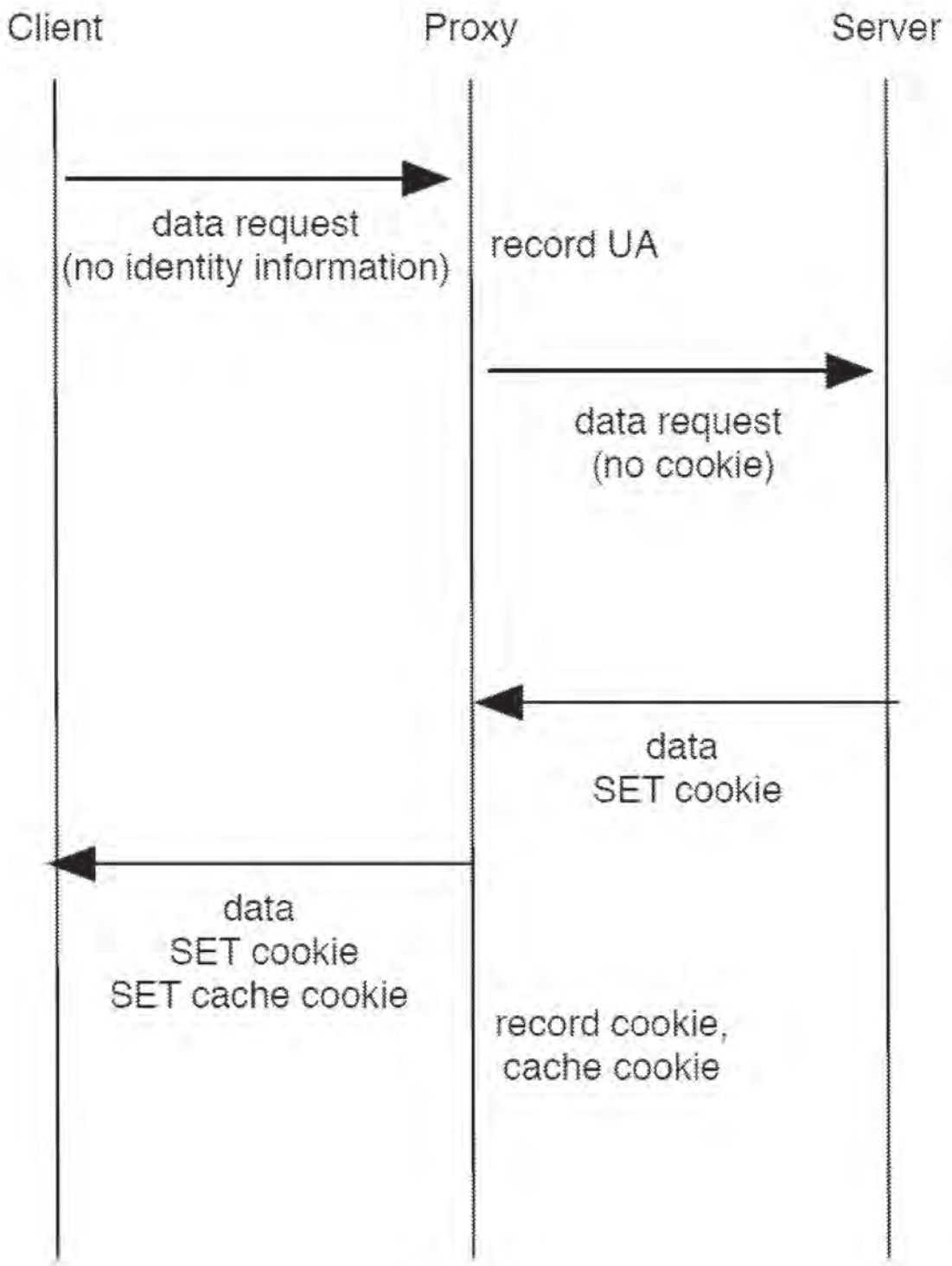

Client
Proxy
Server
data request
(no identity information)
record UA
data request
(no cookie)
data
SET cookie
data
SET cookie
SET cache cookie
record cookie,
cache cookie

FIG. 6

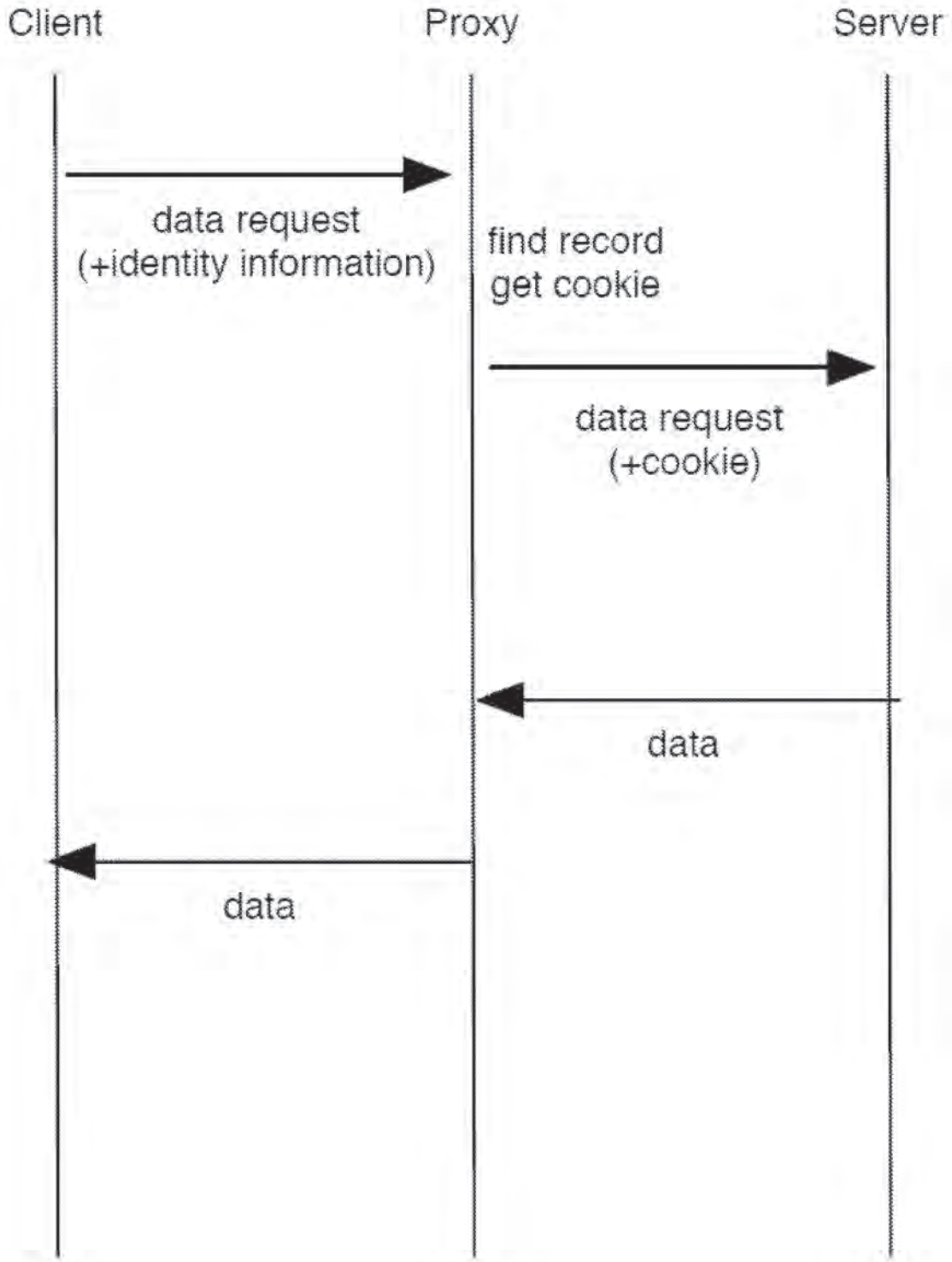

Client
Proxy
Server
data request
(+identity information)
find record
get cookie
data request
(+cookie)
data
data

FIG. 7

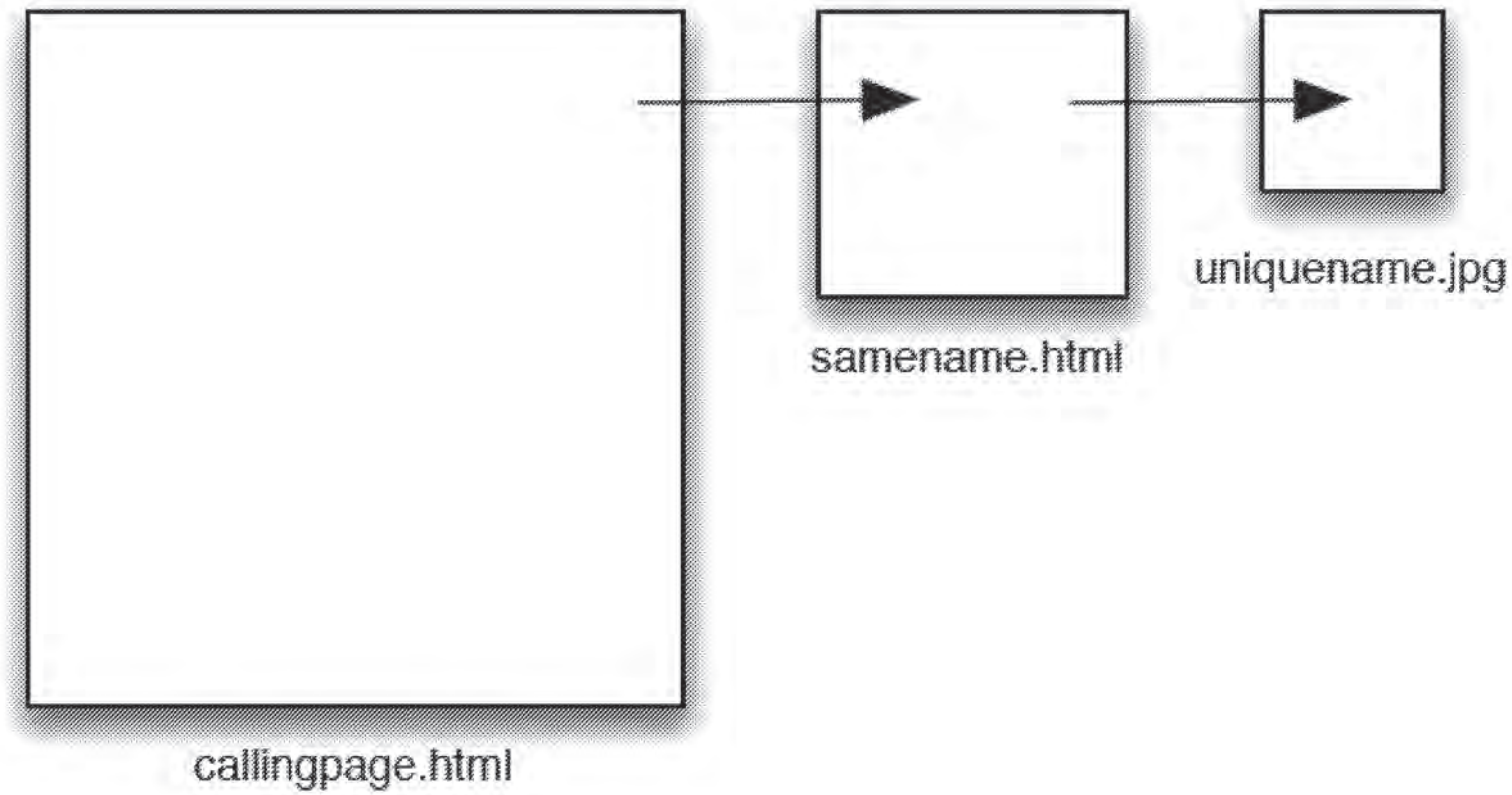


FIG. 8

Electronic Acknowledgement Receipt

EFS ID:	11851374
Application Number:	61587387
International Application Number:	
Confirmation Number:	2362
Title of Invention:	BIOMETRICS-SUPPORTED SECURE AUTHENTICATION SYSTEM
First Named Inventor/Applicant Name:	Bjorn Markus Jakobsson
Customer Number:	21912
Filer:	Robyn Erinn Wagner/Monique Huang
Filer Authorized By:	Robyn Erinn Wagner
Attorney Docket Number:	MJAKP008+
Receipt Date:	17-JAN-2012
Filing Date:	
Time Stamp:	16:25:14
Application Type:	Provisional

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$125
RAM confirmation Number	3933
Deposit Account	500685
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Provisional Cover Sheet (SB16)	MJAKP008plus_SB16.pdf	2071614 [illegible]	no	3
Warnings:					
Information:					
2	Specification	MJAKP008plus_App.pdf	1791098 [illegible]	no	13
Warnings:					
Information:					
3	Fee Worksheet (SB06)	fee-info.pdf	29963 [illegible]	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	3892675		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

<u>New Applications Under 35 U.S.C. 111</u>
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

<u>National Stage of an International Application under 35 U.S.C. 371</u>
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

<u>New International Application Filed with the USPTO as a Receiving Office</u>
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Doc Code: **TR.PROV**
Document Description: Provisional Cover Sheet (SB16)

PTO/SB/16 (11-08)
Approved for use through 01/31/2014 OMB 0651-0032
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

Provisional Application for Patent Cover Sheet

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c)

Inventor(s)

Inventor 1 — Remove

Given Name	Middle Name	Family Name	City	State	Country
Bjorn	Markus	Jakobsson	Mountain View	CA	US

All Inventors Must Be Listed – Additional Inventor Information blocks may be generated within this form by selecting the **Add** button. Add

Title of Invention	BIOMETRICS-SUPPORTED SECURE AUTHENTICATION SYSTEM
Attorney Docket Number (if applicable)	MJAKP008+

Correspondence Address

Direct all correspondence to (select one):

◉ The address corresponding to Customer Number	○ Firm or Individual Name
Customer Number	21912

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

◉ No.

○ Yes, the name of the U.S. Government agency and the Government contract number are:

Doc Code: **TR.PROV**
Document Description: Provisional Cover Sheet (SB16)

PTO/SB/16 (11-08)
Approved for use through 01/31/2014 OMB 0651-0032
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

Entity Status

Applicant claims small entity status under 37 CFR 1.27

- (●) Yes, applicant qualifies for small entity status under 37 CFR 1.27
- ( ) No

Warning

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

Signature

Please see 37 CFR 1.4(d) for the form of the signature.

Signature	/Robyn Wagner/			Date (YYYY-MM-DD)	2012-01-17
First Name	Robyn	Last Name	Wagner	Registration Number (If appropriate)	50575

This collection of information is required by 37 CFR 1.51. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **This form can only be used when in conjunction with EFS-Web. If this form is mailed to the USPTO, it may cause delays in handling the provisional application.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or paten. Accordingly, pursuant to the requirements of the Act, please be advised that : (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, t o a n other federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Attorney Docket No. MJAKP008+

PROVISIONAL APPLICATION FOR
UNITED STATES PATENT

BIOMETRICS-SUPPORTED SECURE AUTHENTICATION SYSTEM

By Inventor:

Bjorn Markus Jakobsson
Mountain View, CA
A Citizen of Sweden

Assignee: Bjorn Markus Jakobsson

VAN PELT, YI & JAMES LLP
10050 N. Foothill Blvd., Suite 200
Cupertino, CA 95014
Telephone 408-973-2585

Introduction

Increasingly, small and accurate finger print readers, voice identifiers, or other biometric devices are being built into consumer electronics such as phones, laptops, mice, tablet computers, game computers, car locks, car ignitions, and door locks. This poses a collection of complex problems for which no solutions to date have been developed.

In addition to the physical considerations associated with building fingerprint readers with low error rates, there are many other issues that need to be addressed to create a practical biometric system. Unfortunately, some of the issues are very problematic. Described herein are an array of issues of relevance, and techniques that minimize the need for passwords.

Example Issues with Biometric Devices

• New device. When a user acquires a new device, her experience should be smooth and intuitive. If a user feels like that task of transferring an established profile from an old device to a new one is too burdensome, this will hamper deployment and cause frustration. On the other hand, it is equally undesirable for a user to register a new device unintentionally – by simply touching an object with a fingerprint reader that a fraudster has placed in a strategic position, or making utterances used as a voice-based authenticator. New device registration, in other words, should be reasonably simple but not automatic.

• Finger theft. While nobody likes the idea of their password being stolen, having one's finger stolen is even less attractive. While there are fingerprint readers that check for the liveness of the tissue, there may always be the nagging question in the minds of would-be users: can those readers be tricked to accept a stolen finger? Similarly, nobody likes the idea of being held captive in order to provide occasional fingerprints to authenticate. A user should be allowed to authenticate without fingerprints. This mechanism can be significantly less practical for the intended user than to use a fingerprint reader, as long as it is more practical to the rough criminal than kidnapping or finger theft would be.

• Legacy systems. Legacy systems speak passwords, and must continue to be allowed to. It is beneficial to be able to operate the fingerprint reader in a way that creates the server-side illusion of passwords being entered. Moreover, when the legacy system demands that the password is updated, this should happen – and in particular, invisibly to the user!

• Credential theft. Credentials should be secure against various abuses, including phishing attacks, DNS poisoning and various malware attacks. Whereas it might seem that these are orthogonal issues that can be dealt with as phishing and malware would otherwise be addressed, the best end result may not be achieved through such approaches. Benefits can be achieved by taking advantage of the opportunities provided by the infrastructure designed to support the fingerprint-based authentication.

• No fingerprint. Using current consumer technology, approximately 98% of the population have fingerprints that can be reliably read. Conversely, approximately two in a

hundred do not have fingerprints that are easy to read using today's technology. While this number will hopefully shrink over the next few years, as the cost goes down and technology improves, the question remains how to design a system that respects those who have difficulties – or apprehension – using fingerprint readers. Developers should keep in mind that while users of biometrics may not mind if they are constantly authenticated, this is not true for people who use passwords instead of biometric approaches.

• Forgotten passwords. The less users rely on passwords, the more likely it will be that once they do need them, they have forgotten them. Similarly, the more anxiety one feels about potentially not remembering a password when it is required, the harder it will be to recall it.

• Device sharing. The system should preferably support multiple users per device, without exposing the credentials of one user to another or leaving one user accidentally logged in after handing the device over to another. This problem exists to some extent already today, as some sites may keep users logged in without them being aware of it. However, the problem is made worse by the possibility of authentication that takes place without the user having to be aware of it – although that may be a desirable feature in most situations.

Example Embodiments

Users interact with fingerprint readers, embedded in devices, and facilitating authentication to websites. Examples of devices include a computer, a handset, a mouse, and a door lock. A cloud storage is used to back up information from fingerprint readers and devices.

Fingerprint readers perform image capture and feature extraction, followed by a comparison to one or more templates. Templates are either stored on the fingerprint reader, or in an encrypted format on the associated device, using a secret key stored by the fingerprint reader. If the extracted features match a given template, the corresponding secure vault is accessed by the fingerprint reader. The vaults, which can be stored on the device, are encrypted using a secret key associated with the matched template and stored by the fingerprint reader. It is assumed that fingerprint readers cannot be infected by malware, but that devices can.

In some embodiments, vaults are encrypted profiles containing templates to which extracted features are compared. They contain user data, such as the user name the user has registered an account under, and the associated site where it is registered. Classification data, such as "email" or "banking" is stored in some embodiments. User addresses, configuration data, credit card details, shipping addresses, frequent flier number, and other such information is also stored in some embodiments. In general, a vault may contain any type of information. In some embodiments, the profile associated with a vault is a user's work persona, and associated information, which in some embodiments contain proprietary files, such as employee data, trade secrets that the employee has access to, etc.

In some embodiments, portions of the data stored in a profile are stored on a storage associated with the fingerprint reader or other biometric device; some portions of data in the associated device; and some portions on external resources, including third-party storage such as cloud servers. In embodiments where not all the profile data is stored on the biometric device storage, the biometric device can encrypt such portions that are stored elsewhere, and store a key associated with the profile. The encrypted data can also be hierarchically encrypted, with some of the decryption keys stored in an encrypted format in the storage external to the biometric device.

One example of a data encryption technique employed in some embodiments is a symmetric-key based encryption method, such as AES. It can also be a public key encryption method such as RSA, or a hybrid encryption technique, such as one using AES and RSA.

In one embodiment, an entire record is encrypted as one unit, which allows the associated ciphertext block to be decrypted without decrypting other ciphertext blocks.

The first time a user registers, he inputs his information, which is then stored in his profile. If he ever changes some aspect of his profile, such as his phone number or credit card number, this can be automatically propagated to all the places in the user's profile where it is used, such as for different accounts.

When a user gets a new device that he wishes to have configured according to previously stored data that he input on another device, then his new device obtains a copy of that profile from a storage where it is stored, after which the user either inputs a key or some data from which a key is computed, from which the received encrypted data can be decrypted and the data be accessed. Alternatively, instead of inputting a key or data from which a key is computed, the user can provide a biometric reading from which features are extracted and either used as a key or used to compute a key, or used along with other material as a key or to compute a key. Examples of such other material include user-provided information, such as a password, responses to life questions, or other data used to authenticate the user. Alternatively, the key or a portion of the key can be stored on separate hardware, such as a phone, a dongle, in the form of a QR code on a sticker, etc.

Example of information that can be stored in a template includes the relative location of features of a fingerprint or retina, a frequency spectrum description of a user's voice, and data describing behavioral treats, such as the manner in which a user moves.

Let us now consider some transaction types, and how they can be performed:

• New device. For a user to register a new device, the user would provide a user identifier (such as an account name), and the new device would connect to the cloud storage or other storage where the encrypted profile resides, provide the account name or other identifier, and be authenticated and associated with the account. The encrypted user template and vault would be sent to the device. The key for decryption could be

generated from information the user has (e.g., stored on a personal identification chip in the user's safekeeping). It can also be generated from biometric data with sufficient entropy – such as features extracted from fingerprinting of all ten fingers. (Note that this type of data would not be collected without the user's knowledge, especially if the scans use portions of the finger that are not normally used for touching the screen or fingerprint reader.) The threshold for new device authorization can also be different from that of a typical authentication session. This combines the benefits of low false accept rates for new device enrollment with low false reject rates for the typical authentication session, requiring a higher quality fingerprint for the rarer new-device authentication than it does for the common authentication session. This translates into a reduced risk of impersonation during the critical enrollment phase and lower risks for accidental enrollment (since even the valid user may have to take special care to produce perfect fingerprints); and improved convenience during typical authentication sessions. If the appropriate key is obtained, the template and the vault can be decrypted and stored on a secure portion of the device, such as the biometric reader or other secure storage that is believed not possible to be affected by malware. In some embodiments, portions of this data is stored on insecure storage, such as the device's storage or external storage, and the key used to decrypt it stored on a secure storage, such as a storage associated with the biometric device. Note that it is possible to register a new device without performing the above process, but at the cost of not gaining access to the vault. This can be done later on, at the user's convenience. This would correspond to the process by which the user registers a device the first time, when he or she does not have a stored profile available anywhere. If the user configures a new device like this, the user can later elect to merge the two or more profiles he may have, where two profiles can be merged by downloading them; obtaining access to the keys (e.g., as described above); combining records; and storing the new combined profile. In some embodiments this new combined profile is encrypted before being stored. In some embodiments it is made to replace the profiles it was produced from, whereas in others it is not. When two or more records are combined and they both contain data of a related type, the system may perform tie-breaking either by the time and date during which the different data elements were first input; last used or confirmed; or by another rule, including rules in which the user performs tie-breaking element by element.

• Authentication. For a user to authenticate to a website or other resource, he performs a fingerprint scan, which causes a verification of the extracted features and a comparison to the templates stored on the fingerprint reader and associated device. If a match is made, the associated decryption key is selected by the fingerprint reader, and the associated vault decrypted by it. The vault is scanned for an entry that matches the website or other resource – e.g., domain, IP address range, certificate, etc. If a matching entry is found, the associated user name and password are extracted, and a secure connection is established between the fingerprint reader and the website/resource; the user name and password are submitted over this connection. After the login is successfully completed, the secure session can be handed over from the fingerprint reader to the device, in a way that does not allow the device retroactive access to the transcripts exchanged between the fingerprint reader and the website/resource (This avoids credential capture in a setting where the device is infected by malware.) This can be achieved using SSL renegotiation,

which causes a new SSL session to be built, using new keys; the new encryption/decryption/authentication key can then be handed off from the fingerprint reader (or other biometric device) to the associated device. Knowledge of this new key does not alow the device to access the contents of any past encrypted contents, since those used a different and unrelated key, thereby achieving "forward security" for the session.

• Backup Authentication. Backup authentication can be implemented in a variety of ways. First of all, if legacy websites retain password reset mechanisms, this will provide an out-of-band approach for users to access their accounts without using fingerprint readers. For example, if so-called life questions are used, then the user can prove his identity by answering these questions correctly; or a system such as the Blue Moon Authentication system can be used to authenticate. Second, a form of emergency access that users can use under duress can be implemented to release the contents of their vaults – to make sure that nobody has to fear losing a finger to violent criminals. As one example, the emergency access is local to a registered device, and works in the same principal way as the new device authentication, except that it does not rely on data that the user might not carry with himself, nor on information she memorizes (as people typically forget under duress.) In one such embodiment, the user would swipe all his fingers to perform an authentication that could either be the authentication for a new device, or an emergency authentication.

• Synchronization. The cloud storage can accept backups from multiple devices associated with one and the same account, and synchronize the updates so that all devices get automatically refreshed. Refreshes can also be made according to user-configured restrictions. For example, such policies can block privileged employer data from being stored on shared personal devices, or on any device that was not issued by the employer.

• Remote wiping. Remote wiping of a user's template is beneficial, both to "unshare" previously shared devices, and to avoid that criminals with physical component access to lost devices gain access to templates and vault contents. It is possible to set policies such as one where a template self-wipes if it is not matched within a particular duration of time, or if none of the profiles of the device are matched within this time. In one embodiment, this is achieved by having the vault contain an encrypted profile and a date and time, where the date and time is associated with the time when the vault should be erased by the storage, but where this data and time is refreshed with a new date and time every time the profile is accessed, thereby postponing erasure as long as the profile is accessed every so often. This way, the operating system associated with the storage of the vault will read the date and time every once in a while and remove "expired" profiles. Since user data can be frequently backed up to the cloud storage, and recovered from this using the new device registration process, the user will not lose any data by performing a remote wipe when in doubt.

There should be no reason to fear finger theft, since it is possible for users to release the entire contents of their vaults without providing a fingerprint, e.g., by answering the life questions or other backup authentication questions associated with the profile. It is also

possible to release the entire contents of the profile in a plaintext format by performing an action of the same kind that is used to register a new device, e.g., swiping all ten fingers. At the same time, if data access is limited by policy to employer-issued devices, a criminal would need access to such a device to gain access to privileged data – whether he plans to steal a finger or force the release of vault contents of his victim.

Legacy systems are respected, which could maintain a world view of user names and passwords. At the same time, though, the passwords could be generated by the fingerprint devices in ways that result in much safer passwords than those produced by typical users – while respecting the password formatting rules of sites. Requests to update old passwords can be automatically intercepted and acted on by devices and fingerprint readers.

We note that credentials – whether passwords, templates or biometric features – are never accessible in plaintext outside the fingerprint devices. This prevents the theft of credentials by malware. Likewise, phishing attempts become largely pointless, as credentials are never exposed to sites that do not match the accounts contained in the vault. The verification of IP ranges and certificates prevents attacks based on DNS poisoning, as credentials would not be released if only the observed website domain is matched.

The resulting automation results both in increased user convenience and improved security.

Figures

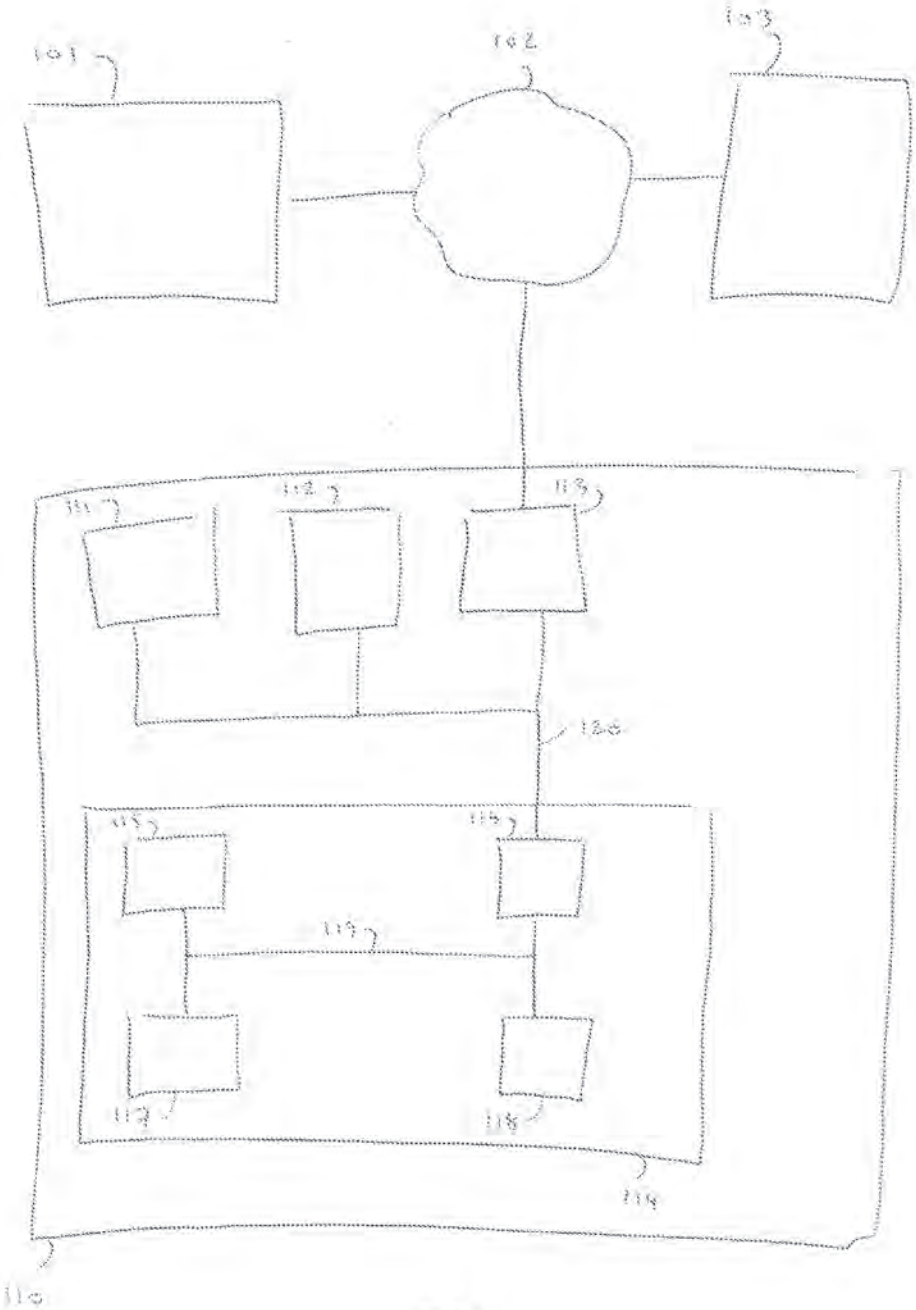


FIG 1

Fig 1 shows a schematic overview of the architecture. A cloud storage or other storage facility 101 is connected using a network or communication channel 102 to device 110. In various embodiments, device 110 includes a storage 111, processor 112, and communication hardware 113, such as radio, all connected using a bus 120. It also

includes or is otherwise in communication with a biometric component 114 that has a reader 115, a processor 117 and storage 118, connected by a bus 119, and to an interface 116 to bus 120. Device 110 is connected to a website or other resource 103 using network or communication channel 102.

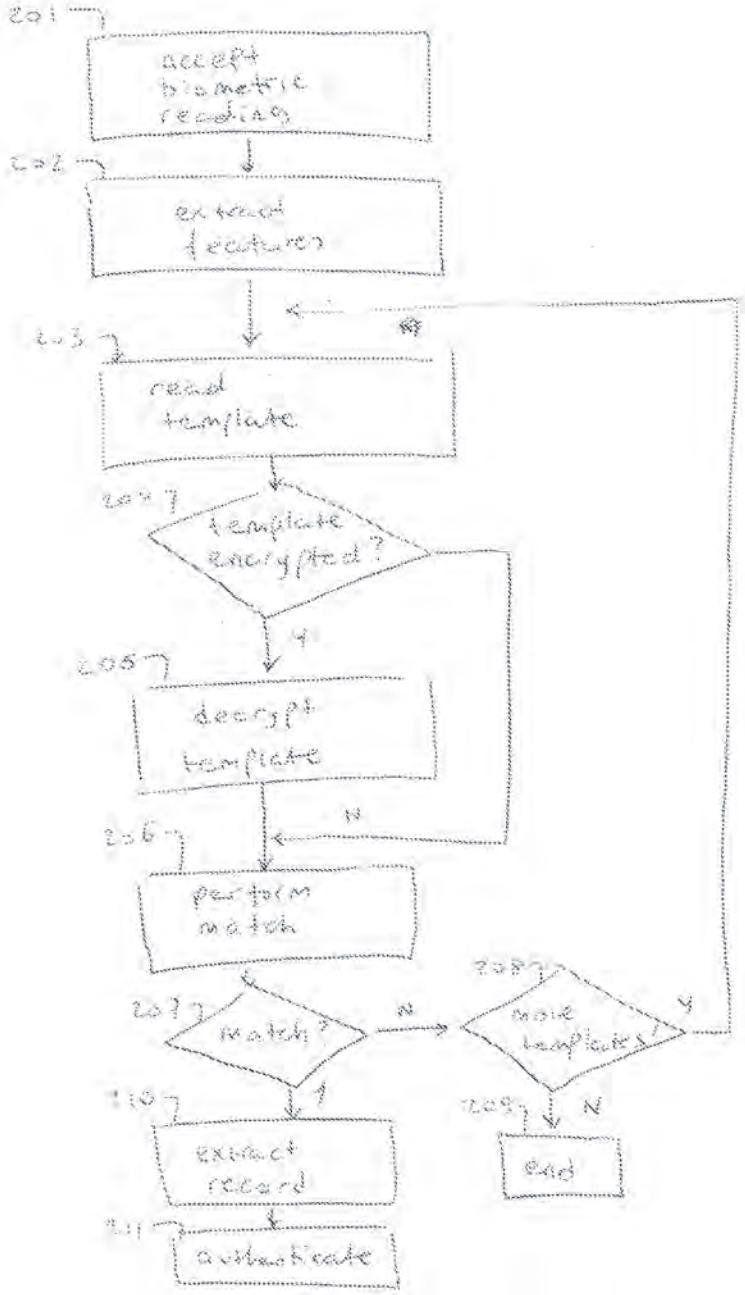


FIG 2

Fig 2 shows an example of a process of authenticating. In 201, the biometric component

114 accepts a reading, extracts features 202 using processor 117 and storage 118, reads a template 203 from storage 118 or storage 111, determines 204 whether the template is encrypted, if applicable decrypts template 205 using a key stored at least partially in storage 118, perform a match 206 between said template and the extracted features, determines if the match is sufficiently good 207. If it is not then determines 208 if there are further templates that can be matched, and iterates 203 if so; otherwise ends the authentication process 209. If the match 207 is sufficiently good, then extracts the appropriate record 210 from a template or associated file, stored either in storage 118 or storage 111, and uses data from the extracted record to authenticate 211 to entity 103, shown in more detail in Fig 4. If match is sufficiently good 207, device 110 can also be unlocked, and contents stored in storage 111 made accessible to the user.

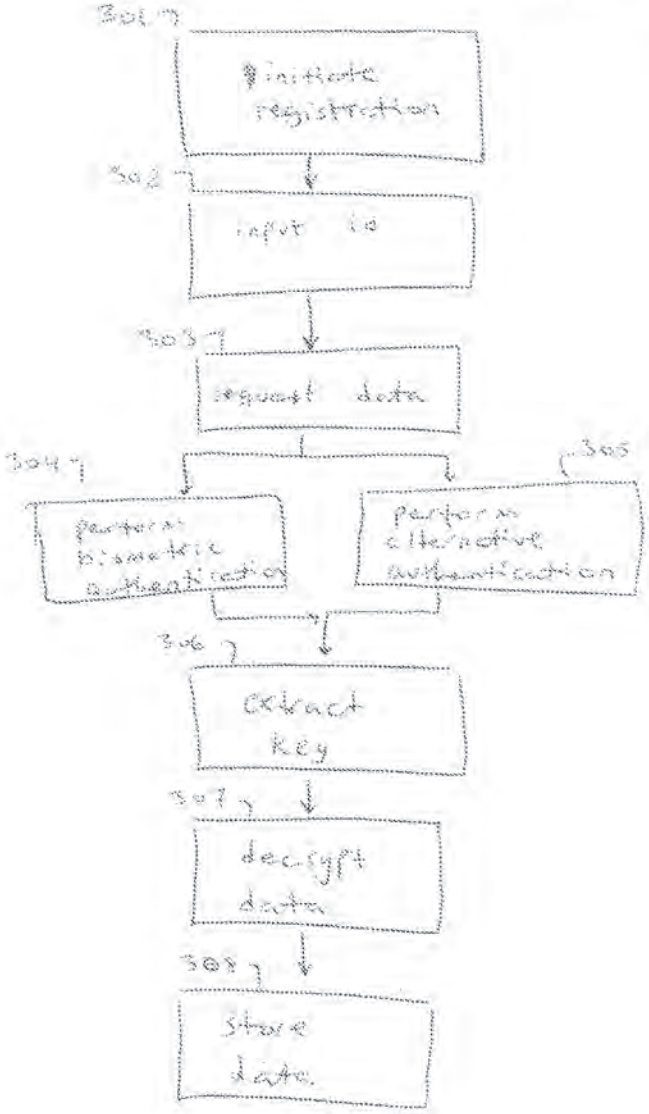


FIG 3

Fig 3 shows an example of a registration of a new device by a user who already has a profile stored in cloud storage or other storage 101. To register a new device 110 and copy user data from previously used devices, the user would initiate a registration 301, input or otherwise obtain a user identifier 302, and request data 303 from storage 101. The user then performs a biometric authentication using biometric component 114, resulting in extracted features 202. Alternatively, device 110 lets the user perform an alternative form of authentication 305 such as life questions or Blue Moon Authentication. Using the data obtained in alternative authentication 305 or using extracted features 202, a key is extracted 306. This key is used to decrypt requested data obtained from storage 101, and the decrypted data is stored 308 on the fingerprint reader device, i.e., 118. The part of decrypted data that contains keys is stored on storage 118, and other parts are either stored on storage 118 or storage 111, where data in storage 118 does not have to be encrypted. Said data contains templates used in 203 and records used in 210. There may be several templates, e.g., one for each type of biometric reader utilized by the user, on the device in question or other devices. These may all be associated with one and the same user, which may be associated with multiple records, where each record describes the access to one particular resource, or contains other user data, such as personal or proprietary files. This data is encrypted when stored in a place that is not considered secure; when the data is secure, this collection of templates and records. On one device, several such vaults or collections of templates and records may be stored; e.g., one for each user who is commonly accessing the device or using it for storage of vault data.

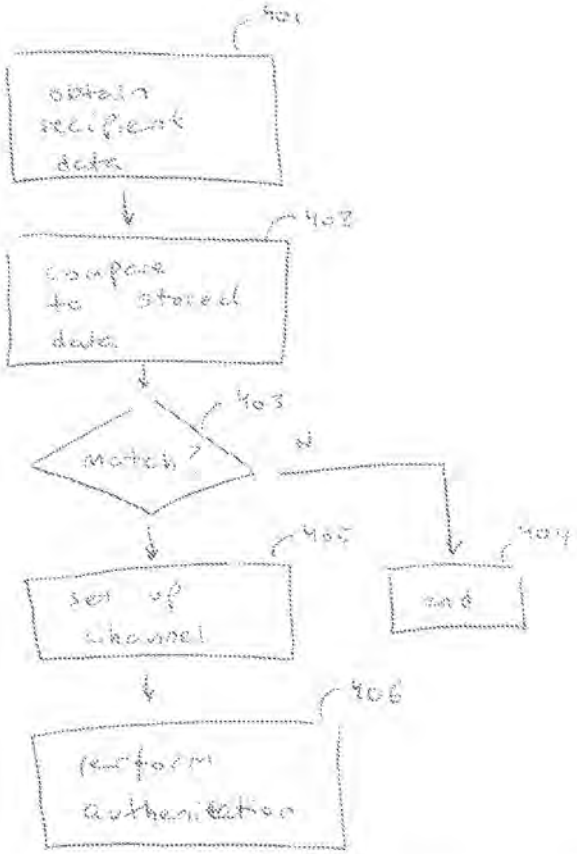


FIG 4

Fig 4 shows an example of authentication 211 to entity 103 in more detail. In 401, recipient data is obtained from entity 103 or a proxy that represents entity 103. This data may include domain name, IP address, certificates, and other data used to authenticate sites, apps and entities. The obtained data is compared to stored data 402, which, in some embodiments, is recorded by device 114 or device 110 or other entity during previous interactions with 103. It is determined whether there is a sufficiently good match 403 between the obtained and the stored data. If there is not then the authentication session is said to have failed, and ends 404. Otherwise, device 110 or preferably component 114 of device 110 sets up a secure channel 405 with entity 103, e.g., using SSL, over which authentication data is communicated 406; such authentication data can include user name and passwords, that were extracted 210 from a matched template or associated file. After device 103 has accepted the authentication 406, a new secure channel can be set up between 110 and 103, or existing secure channel transferred to 110 and 103. Alternatively secure channel 405 may be set up inside a second secure channel, where said second secure channel is set up between device 110 and entity 103; in which case secure channel 405 can plainly be terminated after the authentication 406 has completed.

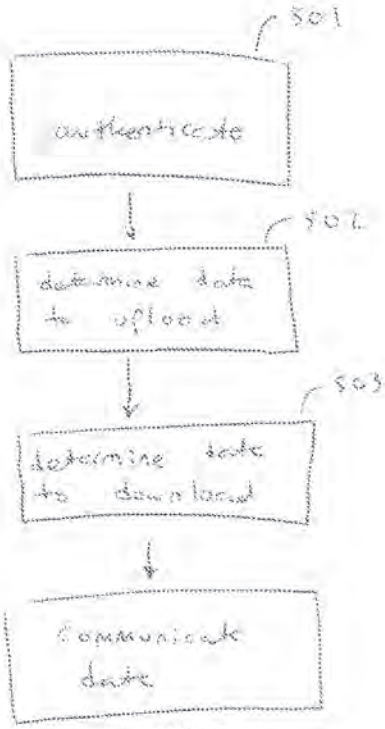


FIG 5

Fig 5 shows an example of synchronization of data between device 110, including component 114, and cloud storage or other storage 101. A secure channel can have been established, e.g., using SSL, after which the entities 110 and 101 authenticate to each other 501. It is determined, based on policies and the identity and access history of device 110 what data to upload 502 to storage 101 from device 110 and component 114, and similarly, what data to download from storage 101 to device 110 and component 114. All data is encrypted using keys stored at least in part by component 114.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
61/587,387	01/17/2012		125	MJAKP008+		

CONFIRMATION NO. 2362

21912
VAN PELT, YI & JAMES LLP
10050 N. FOOTHILL BLVD #200
CUPERTINO, CA 95014

FILING RECEIPT

"OC000000052393989"

Date Mailed: 02/07/2012

Receipt is acknowledged of this provisional patent application. It will not be examined for patentability and will become abandoned not later than twelve months after its filing date. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Applicant(s)

Bjorn Markus Jakobsson, Mountain View, CA;

Power of Attorney:
Robyn Wagner--50575

If Required, Foreign Filing License Granted: 02/03/2012

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 61/587,387**

Projected Publication Date: None, application is not eligible for pre-grant publication

Non-Publication Request: No

Early Publication Request: No
**** SMALL ENTITY ****
Title

BIOMETRICS-SUPPORTED SECURE AUTHENTICATION SYSTEM

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international

patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4158).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and

Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation and commercialization of new technologies. The USA offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to encourage, facilitate, and accelerate business investment. To learn more about why the USA is the best country in the world to develop technology, manufacture products, and grow your business, visit SelectUSA.gov.

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

ACKNOWLEDGEMENT OF LOSS OF ENTITLEMENT TO ENTITY STATUS DISCOUNT

APPLICATION #	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET #	REQUEST ID
61/587,387	01/17/2012	Bjorn Markus Jakobsson	CARBP002+	63421

The entity status change request below filed through Patent Center on 02/13/2023 has been accepted.

Certifications

APPLICANT CHANGING TO REGULAR UNDISCOUNTED FEE STATUS

Signature

I certify, in accordance with 37 CFR 1.4(d)(4), that I am one of the signatories making the entity status change.

Signature	Name	Registration #
/Robyn Wagner/	R Wagner	50575

Electronic Acknowledgement Receipt	
EFS ID:	11587264
Application Number:	61569112
International Application Number:	
Confirmation Number:	5370
Title of Invention:	BACKWARDS COMPATIBLE ROBUST COOKIES
First Named Inventor/Applicant Name:	Bjorn Markus Jakobsson
Customer Number:	21912
Filer:	Robyn Erinn Wagner/Monique Huang
Filer Authorized By:	Robyn Erinn Wagner
Attorney Docket Number:	
Receipt Date:	09-DEC-2011
Filing Date:	
Time Stamp:	18:32:36
Application Type:	Provisional

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$125
RAM confirmation Number	5616
Deposit Account	500685
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Provisional Cover Sheet (SB16)	MJAKP007plus_SB16.pdf	2071601 [illegible]	no	3
Warnings:					
Information:					
2	Specification	MJAKP007plus_App.pdf	597245 [illegible]	no	15
Warnings:					
Information:					
3	Fee Worksheet (SB06)	fee-info.pdf	29533 [illegible]	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	2698379		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Attorney Docket No. MJAKP007+

PROVISIONAL APPLICATION FOR
UNITED STATES PATENT

BACKWARDS COMPATIBLE ROBUST COOKIES

By Inventor:

Bjorn Markus Jakobsson
Mountain View, CA
A Citizen of Sweden

VAN PELT, YI & JAMES LLP
10050 N. Foothill Blvd., Suite 200
Cupertino, CA 95014
Telephone 408-973-2585

Consider a server S and a client C that communicate over a network N. S sets cookies for C, but occasionally, C erases them. Disclosed herein are techniques that modify S in a way that makes C possible to identify with a greater probability than if only traditional cookies are used. As one example, the techniques can be implemented as a filter F installed between S and N, requiring no modifications to S or C. The filter can also be integrated into S.

Traditionally, the server S can SET a cookie and READ a (HTML) cookie. The setting (or writing) of a cookie is initiated by S, while reading of the cookie is a passive action for S, as S simply receives information transmitted by the client C. In an augmented system with a proxy P, the server S will act the same as in the traditional context, but the proxy P will filter and inject traffic to increase the likelihood that a correct cookie value is received by S for a given client. The probability is increased without having to modify S.

The proxy P will add "cache cookies" to traffic going to C. When P sets a cookie to C, then P will cause the cookie to be set, and also cause a cache cookie to be set to C. When a cookie but no cache cookie is received from C, then P will communicate it to C, and cause a new cache cookie to be set to C. When a cache cookie but no cookie is received from C, then P will determine what the corresponding cookie value is, and communicate that to S; it will also cause this cookie value to be set to C. The translation between cache cookie values and cookie values can be performed in a variety of ways, such as by using a lookup table, or using a computation or mapping of values.

One way to set a cache cookie is for P to send to C a document A.HTML that references an element V. To render A.HTML, C will have to request V, since it does not have this in its cache. However, P will not serve V. A.HTML can be named the same for all clients, but the value V is unique to the cookie to be set to a given client C. Here, V is associated with a domain controlled either by P or S, or a party associated with either of these.

Later, to read a cache cookie, P can send a document B.HTML that references A.HTML. To render B.HTML, C determines whether it has A.HTML in its cache. If it does, it retrieves it. Then, to render A.HTML, it will request V from P. Doing this will identify it, since V is unique to the client C. P will not serve V - therefore, V will remain not known by C, and therefore, will cause subsequent cache misses when B.HTML is served again at some later time.

In some embodiments the value V incorporates the value of a cookie associated with C. For example, if S has set a cookie "abc123" for C, then V can be the object "abc123.jpg." When P learns that a client wishes to render "abc123.jpg" it would conclude that this is a client whose cookie value is "abc123"; it would then transmit that cookie value to S. It would also set that cookie from S to C, if the cookie was not transmitted along with the cache cookie value.

P can also perform a mapping of cookie values to cache cookie values V. For example, it can encrypt all cookie values individually, using a cryptographic key K1 known by P. Doing this may cause a cookie "abc123" to be encrypted to a value "AA311B2723," which would in turn be converted to a cache cookie value V="AA311B2723.jpg." The use of the extension jpg here is only for the sake of an example, and it could be any file type of applicability in the context of a

document that can be rendered by a browser. Similarly, HTML is also only an example of a document type that can be rendered.

Consider a cache cookie that was computed by mapping a cookie value to a cache cookie value, as described above. When such a cache cookie value is read, and V="AA311B2723.jpg" is obtained, P would convert that to a value "AA311B2723" and then decrypt that using the key K2, where decrypting with K2 results in the original cookie value "abc123". This is then communicated to the server S. It can also be set to the client C.

Cache cookie values can also be computed from cookie values using another form of mapping, or using lookup tables. One form of mapping is the identity function, in which essentially the same value is used for the cookie value and the cache cookie value.

The proxy can add additional functionality, such as sequence numbers. For example, if a sequence number 1 is associated by P with a client C, and S wishes to set a cookie "abc123" to C, then P can attach the value 1 to the cookie string, obtaining an augmented cookie value "abc1231"; after which it maps to a suitable cache cookie value. As before, this can be done using mappings or lookup tables. The proxy P can use a function f1 to map the augmented cookie value to a cache cookie value V. In the example above, the result of applying f1 to "abc1231" may be the value V="01000100010000001.gif". Both the file name and the file type may be the result of this function. To perform the reverse mapping, a function f2 will be applied to any received cache cookie value; here f2(f1(x)) = x. This would provide P with the augmented cookie value, e.g., "abc1231", from which it can compute the associated cookie value "abc123" and transmit that to C; and use it to set a cookie to C. The cookie sent to C may also be augmented, but it does not have to be augmented with the same number as the cache cookie was augmented with. For example, there could be two serial numbers associated with C; one for cookies and one for cache cookies. These two numbers may be independent. If, when reading cookies and cache cookies from C, P can determine if the two values added in the augmentation are consistent with each other and consistent with the state kept by P, and potentially associated with C. If the values are not consistent, P may communicate an alert to C, or may decide not to communicate the cookie value to C. P may also modify the sequence numbers associated with C.

In addition to using cache cookies to help S recognize client devices C, P can add other features that augments the data flow to help increase the probability of correct recognition. For example, when P transmits cookies or cache cookies to C, it can also read the browser agent of C, and store information associated with this. When a client device C connects to S via P without transmitting any cookie or cache cookie information, then P can read the browser agent and determine whether it recognizes C with a sufficiently high assurance; and if so, determine the cookie value associated with C and transmit this to S; it will also cause the setting of the cookie and cache cookie values to C.

P can augment the cookie value and cache cookie value using a value associated with the browser agent. For example, instead of using a sequence number, or in addition to using this, P can use the browser agent or part thereof. For example, P can determine that the browser agent

corresponding to C corresponds to a value "501", and map this to a first sequence number or value, say "110", to be used to augment the cookie; and map the same value to a second sequence number of value, say "F9j", and augment the cache cookie string with this. The augmentation may be performed before or after the mapping or lookup.

P can also store a browser agent (also referred to as a user agent) associated with C in a record associated with C, where P also stores the cookie value associated with C. If P receives a browser agent value that identifies C with a sufficiently high certainty, but not any cookie or cache cookie value, then it may transmit the cookie value associated with the record containing the received browser agent value to S, and use the same value to set cookies and cache cookies for C on behalf of S. Multiple browser agents can be associated with one and the same user or client.

P may also store additional information associated with C or a user of C in a record. Examples of this additional information include the credit card number and CVV of the user, or a life question, or a password. These may be stored in cleartext or in a secured format. If P cannot identify a client C, or there is inconsistent data, then P may request such additional information from the user of C. If the user enters this information, then this can be used to look up the correct record and then validated. The associated record would also contain the associated cookie value for C; this value can then be transmitted to S, and used to set cookies and cache cookies to C; and the observed browser agent can be recorded in the record.

P may also utilize an auto-fill feature to request data to be filled by C's browser, on behalf of the user associated with C. This way, the user associated with C would not have to enter the information that would be used to identify the proper record. The user could be requested to click on a "submit" button, causing the auto-filled information (that does not have to be visible to the user) to be submitted to P. Techniques such as those used for "click-jacking" and "like-jacking" can be used to channel user clicks from a user to P. Other techniques for initiating or causing a click can also be used. The initiated click may be performed by a user who believes he or she is simply pressing a submit button, whereas he or she is also submitting values that are auto-filled in a portion of a page that are not necessarily visible to the user. This initiated click can be used to transmit information that was auto-filled by the browser associated with C, and then used to identify the record kept by P and associated with C.

In the above, C will not have to be aware of the presence of P, but its traffic will appear as if the data was sent by S. Similarly, S will not have to be aware of P, but the traffic it receives will appear to come from C. P may be integrated in S, or used as a filter, as described above.

The proxy P can combine the functionality of multiple servers S1 and S2. In one embodiment, P includes two components P1 and P2, where P1 is associated with S1 and P2 with S2, and where P1 and P2 share data, such as information about clients C. In this embodiment, P1 may send packets both on behalf of S1, and cause the sending of data on behalf of S2, where a response by C to either of these packets would cause the identification of C to P1. P1 may then cause the rewriting of data, according to the above descriptions, on behalf of S1, but may also

cause the rewriting of data on behalf of S2, thereby causing the identifying information associating C with both S1 and S2 to become hardened in the sense that as long as one of the servers S1 and S2 would be able to recognize C, so would the other. Again, this is transparent to the servers. One way in which this can be performed is that proxy P1 sends a cookie read request on behalf of P2; this could be an element that, to be rendered, causes C to request data from S2 from C. P2 would filter out this request so that it does not necessarily reach S2, and would notify P1. Both P1 and P2 could then perform actions to write cookies to the client C. The element causing a request by C to S2 via P2 may either be the same as what S2 would use to detect C for S2, or it could be an associated item that P2 knows corresponds to a request emanating from S1 and P1, but which P2 maintains when C visits S2. As an alternative, S1 and S2 may both, via P1 and P2, set cache cookies of the following format: A.HTML would reference one object V1 associated with domain S1 and one object V2 associated with domain S2. As soon as A.HTML is referenced, both V1 and V2 will be requested from their associated domains; the associated cookies will be sent at the same time, if available. That way, both P1 and P2 can rewrite cookies when either of them reads a cache cookie.

A third-party proxy P3 can be associated with proxies P1 and P2, which in turn are associated with server S1 and S2. When either S1 or S2 sets a cookie to C, their associated proxy, P1 or P2, sends a cookie set request to C. This can contain a reference to material associated with the domain of P3. For example, S1 wishes to set a cookie to C. P1 sends the set-cookie request to C, and also sets a cache cookie to C. This cache cookie includes an object A.HTML that references V1 and V3, where V1 is an element associated with a domain associated with P1 or S1, and V3 is an element associated with a domain associated with P3. C will request V1 and V3 to render A.HTML, but will not be served these elements. If S1 wishes to read a cache cookie from C (which happens when C visits S1), the proxy P1 sends an element B.HTML to C; B.HTML references A.HTML, which C has stored in its cache. To render A.HTML, C has to request V1 and V3. Neither P1 nor P3 sends the associated objects. P1 detects the cache cookie value associated with the value of V1, and transmits information to S1 as described before. P1 may also write a cookie to Cm or perform other actions to update records associated with C. When P3 receives the request for V3, it may also receive cookies associated with its domain. Using the value associated with V3, and potentially the cookie value it received, it identifies a record associated with C. It then may write a cache cookie associated with S2 and P2 to C, which may cause a request to be sent from C to P2, and may cause a request to be sent from C to P3. P2 will react to this by setting a cookie to C, but will not report to S2, since C did not visit S2, but this was a result of visiting S1, and then interacting with P3.

Since P can identify a device C based on evidence that is not a hundred percent certain, there is a parameter or a configuration that determines what certainty is sufficient. This is a per-client parameter or configuration that is a function of the risk exposure, the case history, and previous accesses, including the score associated with each such access. This constitutes a threshold determining what actions to perform upon receiving data from a client. In general, the threshold applies to all the inputs, including one or more cookies; one or more cache cookies; browser agent data; and auxiliary data that may be provided by the user, or which may be provided by

the browser of the user. This threshold can be a simple value or an n-dimensional boundary specifying what is sufficient certainty given n input features.

In the above, C has been described as a separate component between S and the network N. In some embodiments, P is a component that is part of the network, and which is contacted by S when communicating with C. Instead of being a filtering proxy, P is therefore a stand-alone service provider that receives requests from S associated with a client. Ways of doing this include by forwarding or rerouting packets, and by including elements in the packets that cause C to communicate with P -- such as references to A.HTML, which is associated with P and served by P, and which makes references to a cache cookie value V associated with V or S, or both. P can also be part of the server.

In some embodiments, biometric sensors are integrated. When server S sets a cookie, proxy P sends a request to read biometric data to associate said biometric sensor with the account. Client S responds with data that identifies the biometric readings. Proxy P records these associated with said cookie. When client C makes a connection with server S, proxy P again requests biometric data to be read. Client C makes a biometric read and responds with corresponding biometric data. Proxy P determines what account is associated with said corresponding data and sends the associated cookie to server S. In another embodiment, proxy P also sets cookies and cache cookies during the set operation, and records user agent data; then reads the cookie, cache cookie and user agent during the cookie read operation.

The multiple sources of identifying data serve as backup for absent or deteriorated data, and to provide evidence that the data is valid. If some of said data including cookie data, is stolen by an attacker, but other data is not stolen, then an attacker would only be able to present some of the data to the proxy P, causing the detection of said attack.

In some embodiments, multiple readings correspond to the registration of a new client device C, whereas fewer readings correspond with the cookie set and cookie read operations. For example, a user may use his thumb to authenticate to a device during an operation associated with the cookie set or cookie read operations. The same user may have to use both thumbs and index finger to register a new device, or to set up an account. This protects the user by making sure that he or she does not accidentally register a new device in his name, where the registration of a device would associate it with his account and cause a cookie associated with a previous device to be observed by server S when new client C' communicates with S via proxy P.

In some embodiments, proxy P is not filtering traffic between server S and network N, but is listening in to the traffic between the two, and injecting additional traffic. In these embodiments, when S sets a cookie to C, P adds a request to set a cache cookie to C, and to read the user agent and other potential auxiliary data. When S reads a cookie from C, which is an action initiated by C sending data to S over N, then P sends C a request that causes the reading of the associated cache cookie, browser agent, and other auxiliary data. If the client C initiates interaction with server S without sending the cookie (e.g., the cookie has been erased), then P

does not know the identity of the client C, but can determine this by reading the user agent, the cache cookie, and other auxiliary data. Once this has been read, P determines the associated cookie value and injects that in the traffic from C to S, causing S to perceive that the cookie was sent by C. In one such embodiment, P monitors and injects traffic on a bus, LAN or other wired network; in another such embodiment, P monitors and injects traffic over a wireless channel.

In some embodiments, the traffic that is being filtered or monitored and injected is not encrypted traffic, and the proxy P does not need to know the key material to decrypt or encrypt packets. In other embodiments, the traffic is encrypted, and P knows the key, allowing P to decrypt and encrypt traffic. In another embodiment where the traffic also is encrypted, P does not know the key K, but injects unencrypted traffic.

Description of Figures

Fig. 1 shows a server 100 that interfaces a proxy 120 using an interface 110; the proxy 120 interfacing a network 140 using an interface 130; a client 160 interfacing said network using an interface 150.

Fig. 2 shows a server 100 making a request 210 to set a cookie for a client 160; where request 210 is converted at proxy 120 to a request 221 to set a cookie and a request 222 to set a cache cookie. Fig 2 also shows a potential read 231 of an HTML cookie, a potential read 232 of a cache cookie, a potential read 233 of a browser agent or auxiliary data supplied by a user or his or her browser. The data being obtained by proxy 120 in one or more of these read operations is combined and compared to a threshold. If the threshold is reached a read cookie operation 220 is performed to inform server 100 that the cookie was read.

Fig. 3 shows an element 300 that references 310 a second element 320.

Fig. 4 shows an element 400 that references 410 a second element 420 that when rendered references 310 a third element 320. Element 300 and element 410 correspond to each other.

Fig. 5 shows an example of what is performed at a client 160 as it receives a request 222 to set a cache cookie 300 in some embodiments. In 500, this element 300 is rendered, which causes 510 the client 160 to attempt to load the element 320 referenced 310 by element 300. It is determined 520 whether this loading was successful 550 or not 530. If it was not successful 530, then the element 320 is requested by client 160 from proxy 120 in 530. The client 160 may not be aware of the proxy 120, but think that it requests 530 the element 320 from server 100. The state 550 should never be reached, since the element 320 is preferably never served to client 160. The state 540 signifies that the request has been completed.

Fig. 6 shows what is performed at a client 160 as it receives a request 232 to read a cache cookie 300 in some embodiments. This is done by proxy 120 sending element 400 to client 160. Client 160 attempts to render element 400 in step 600. This causes 610 the element 420 referenced 410 by 400 to be attempted to be loaded from the cache of client 160. It is determined 620 whether this is successful or not. If it is not then the cache cookie is not present

621. If it is, then the browser associated with client 160 attempts to render element 420, which is obtained from the cache of client 160. This element corresponds to 300. Therefore, when 420 is rendered in 630 the reference 310 to element 320 causes the browser associated with client 160 to attempt to load 640 element 320 from its cache. It is determined 650 whether this is successful or not. If it is successful then state 651 is entered, otherwise state 660 is entered. State 651 should never be reached, since the element 320 is preferably never served to client 160. The state 660 signifies that the request has been completed.

Fig. 7 shows actions taken by proxy 120 in some embodiments. In step 700 it receives a cookie value x from server 100 after interacting 210 with server 100. Proxy 120 computes a function f(x) from cookie value x in step 710, and then determines value V in step 720, where V is a value that is associated with element 320.

Fig. 8 shows actions taken by proxy 120 in some embodiments. In step 800, proxy 120 receives a value V from interaction 232 with client 160. Proxy 120 then computes 810 the value of f^{-1}(V), which is the inverse of the function f computed on the value V. In 820 proxy 120 determines the value of cookie x that corresponds to value V. Here, C is a value associated with element 320.

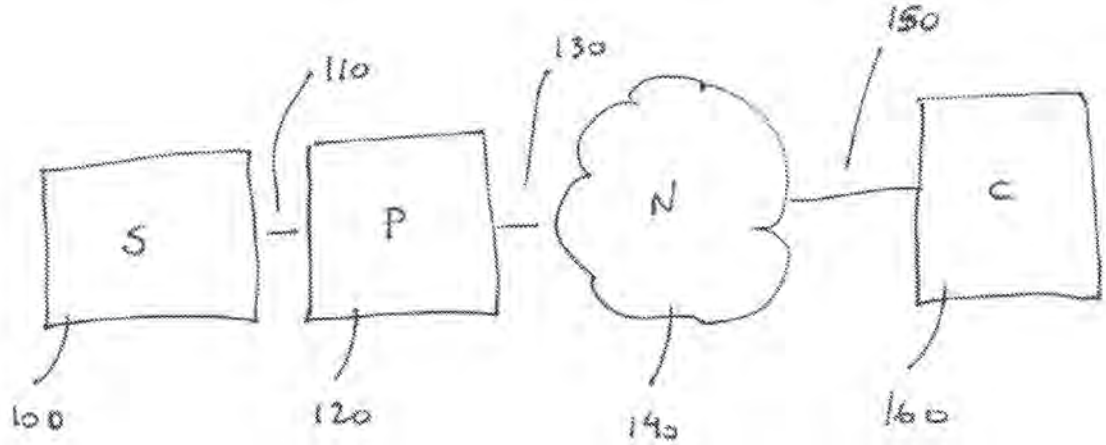


Fig 1.

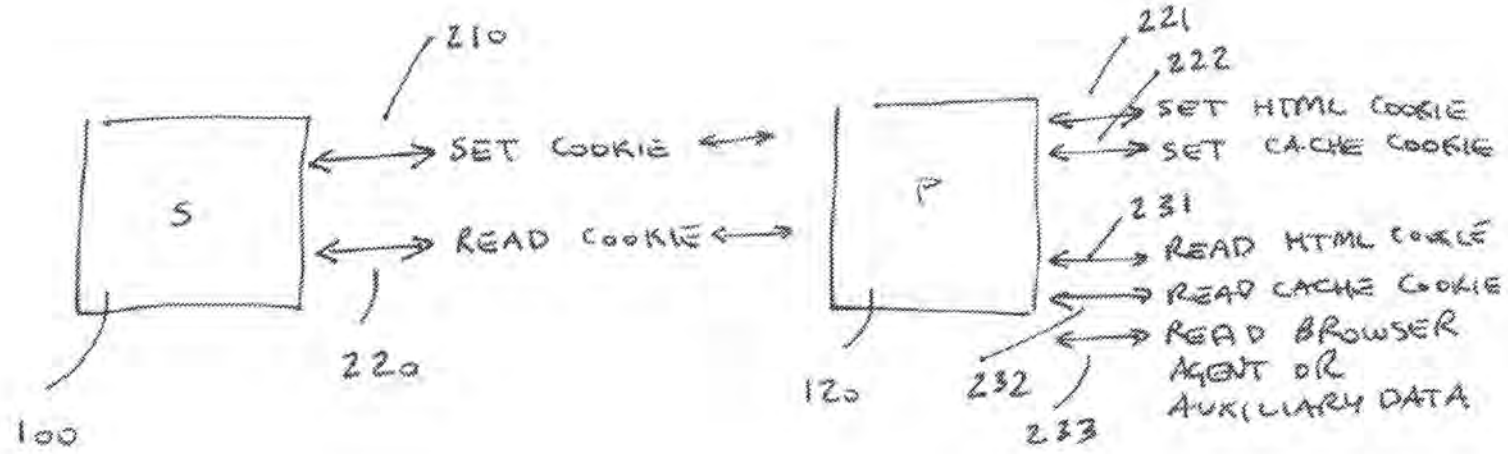

210
S
SET COOKIE
READ COOKIE
220
100
P
120
221
222
SET HTML COOKIE
SET CACHE COOKIE
231
READ HTML COOKIE
READ CACHE COOKIE
READ BROWSER AGENT OR AUXILIARY DATA
232
233

Fig 2.

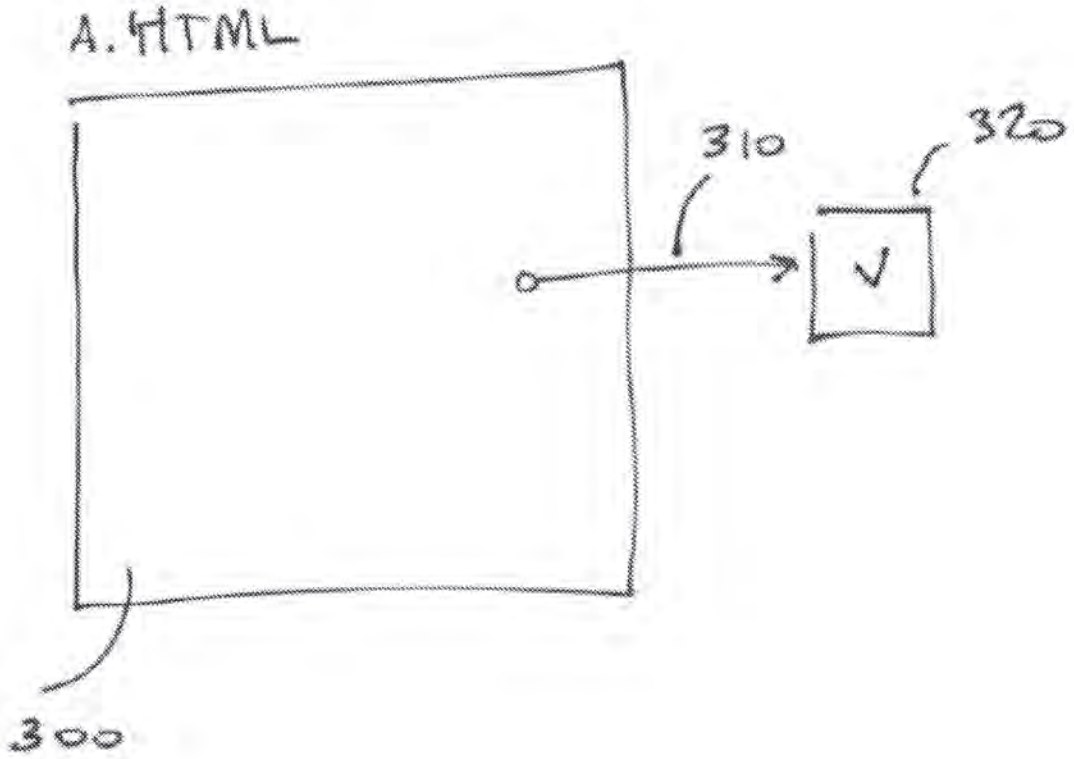


Fig 3.

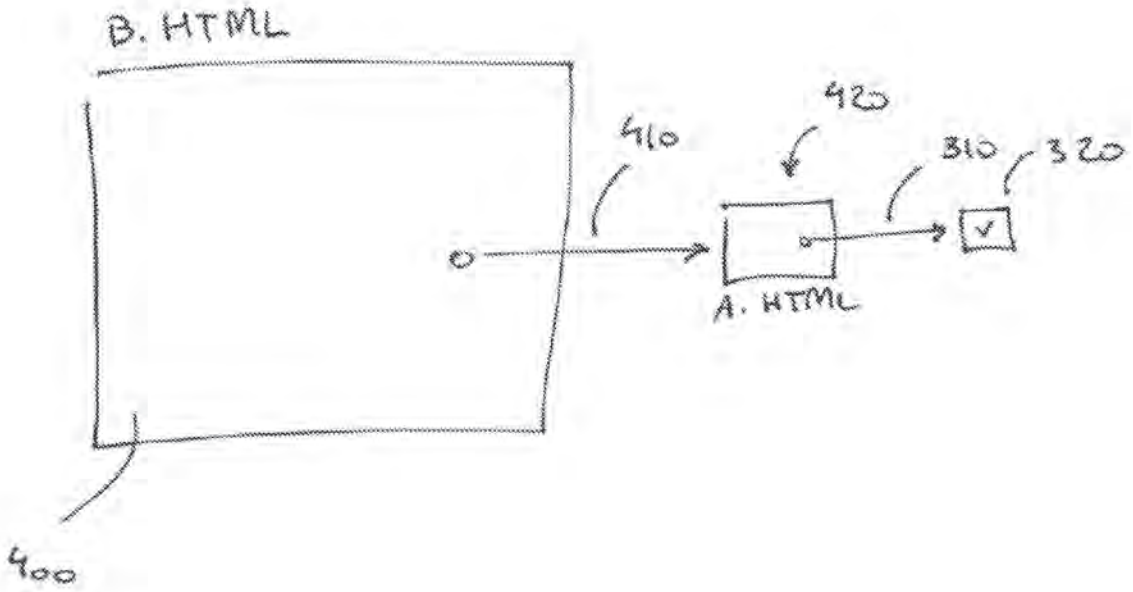


Fig 4.

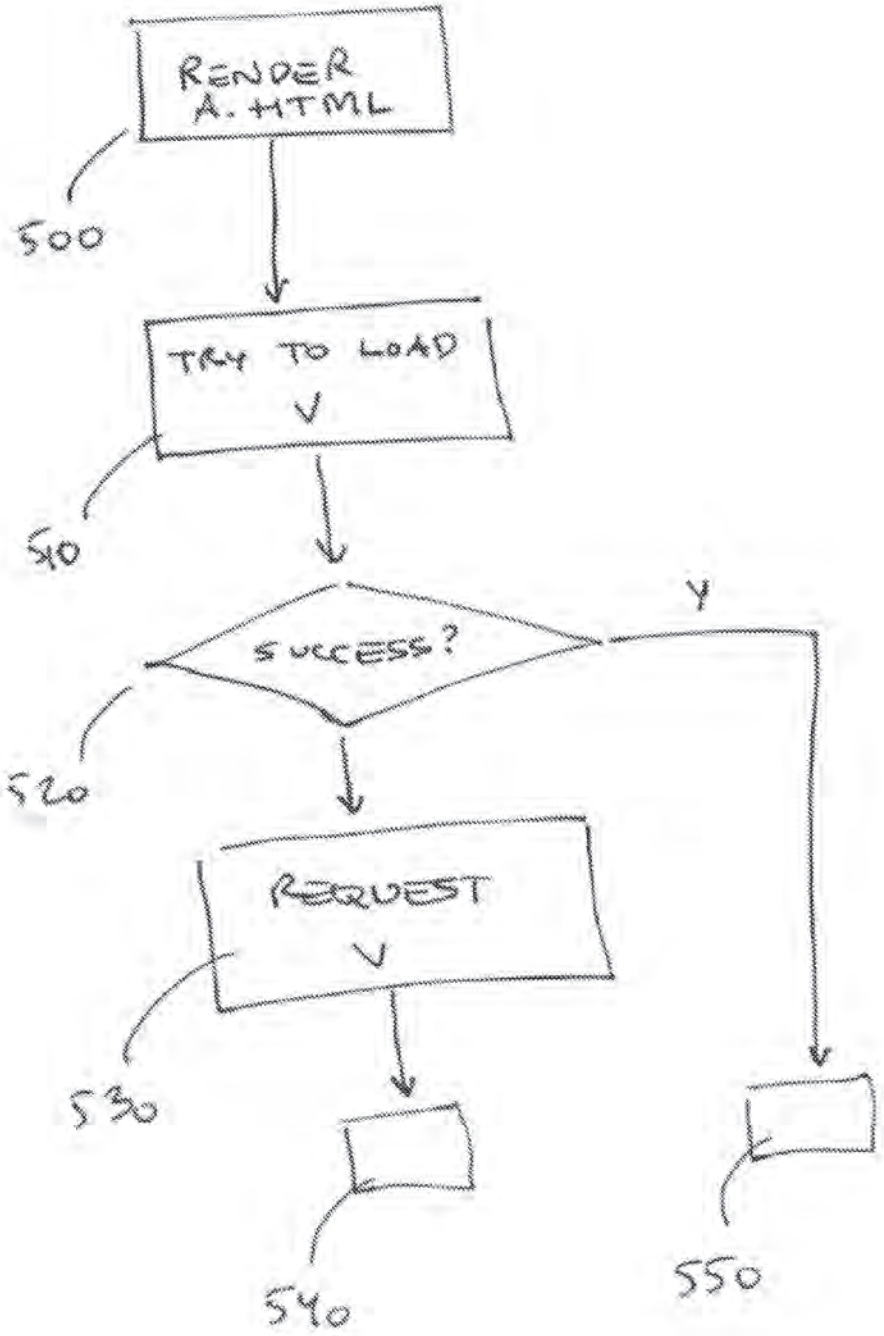

RENDER
A.HTML
500
TRY TO LOAD
V
510
SUCCESS?
Y
520
REQUEST
V
530
540
550

Fig 5.

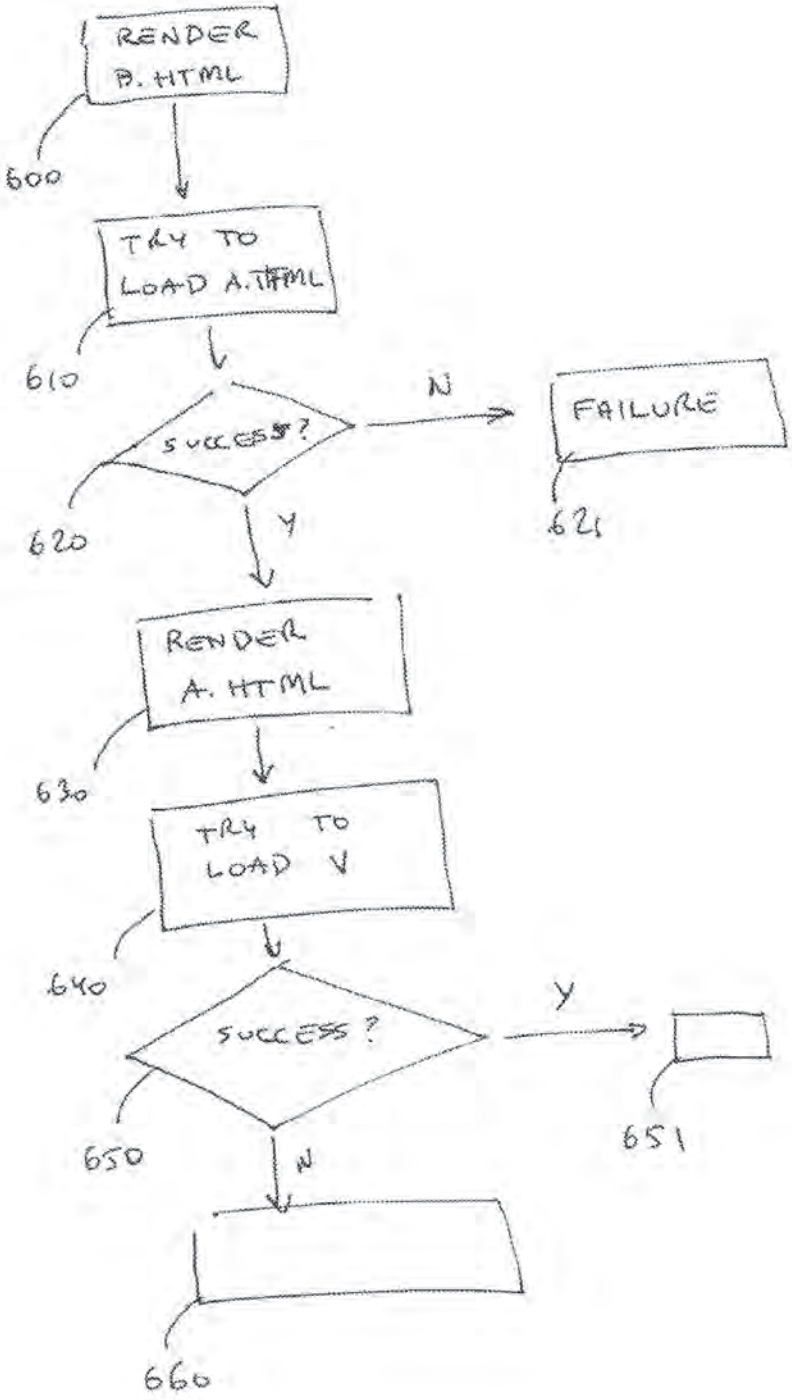

RENDER B. HTML
600
TRY TO LOAD A. HTML
610
SUCCESS?
620
N
FAILURE
621
Y
RENDER A. HTML
630
TRY TO LOAD V
640
SUCCESS?
650
Y
651
N
660

Fig. 6

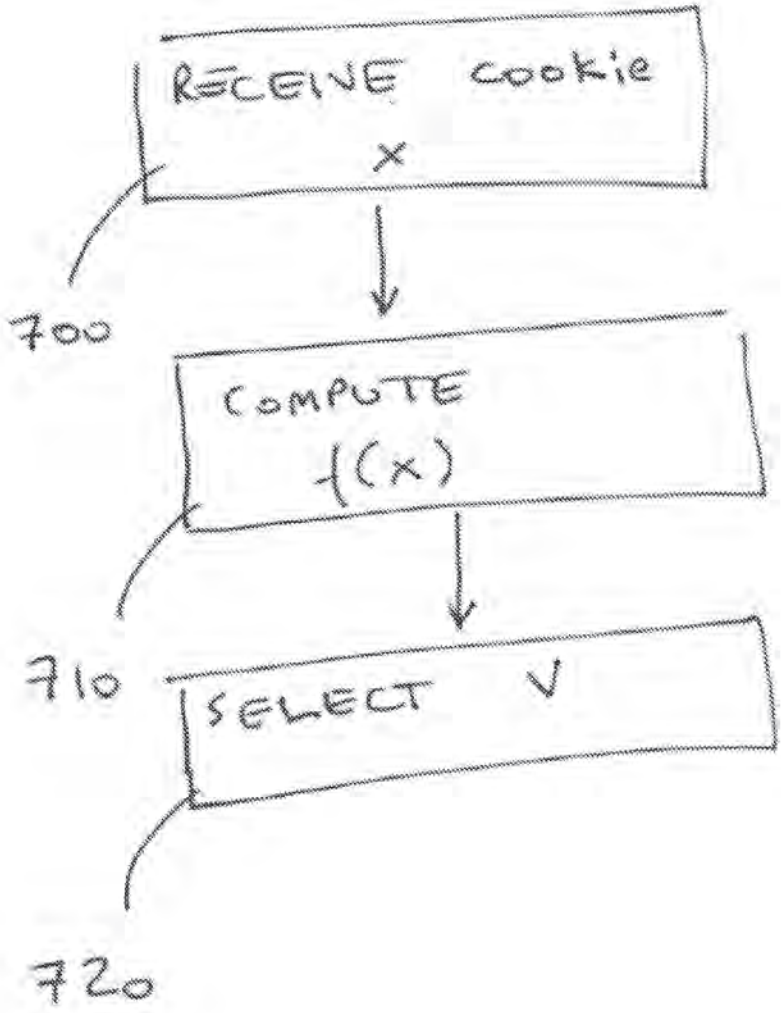


Fig 7.

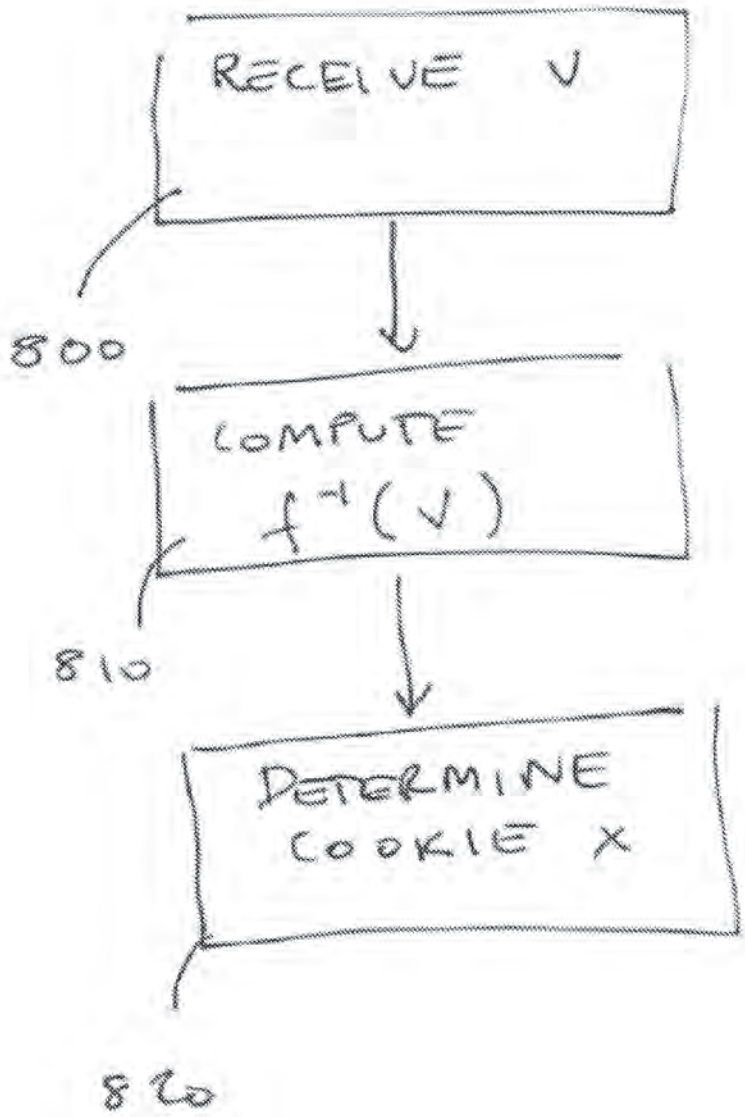


Fig 8.

Doc Code: **TR.PROV**
Document Description: Provisional Cover Sheet (SB16)

PTO/SB/16 (11-08)
Approved for use through 01/31/2014 OMB 0651-0032
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

Provisional Application for Patent Cover Sheet

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c)

Inventor(s)

Inventor 1 — Remove

Given Name	Middle Name	Family Name	City	State	Country
Bjorn	Markus	Jakobsson	Mountain View	CA	US

All Inventors Must Be Listed – Additional Inventor Information blocks may be generated within this form by selecting the **Add** button. Add

Title of Invention	BACKWARDS COMPATIBLE ROBUST COOKIES
Attorney Docket Number (if applicable)	MJAKP007+

Correspondence Address

Direct all correspondence to (select one):

◉ The address corresponding to Customer Number	○ Firm or Individual Name
Customer Number	21912

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

◉ No.

○ Yes, the name of the U.S. Government agency and the Government contract number are:

Doc Code: **TR.PROV**
Document Description: Provisional Cover Sheet (SB16)

PTO/SB/16 (11-08)
Approved for use through 01/31/2014 OMB 0651-0032
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

Entity Status

Applicant claims small entity status under 37 CFR 1.27

- (●) Yes, applicant qualifies for small entity status under 37 CFR 1.27
- ( ) No

Warning

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

Signature

Please see 37 CFR 1.4(d) for the form of the signature.

Signature	/Robyn Wagner/			Date (YYYY-MM-DD)	2011-12-09
First Name	Robyn	Last Name	Wagner	Registration Number (If appropriate)	50575

This collection of information is required by 37 CFR 1.51. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **This form can only be used when in conjunction with EFS-Web. If this form is mailed to the USPTO, it may cause delays in handling the provisional application.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or paten. Accordingly, pursuant to the requirements of the Act, please be advised that : (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, t o a n other federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
61/569,112	12/09/2011		125	MJAKP007+		

CONFIRMATION NO. 5370

FILING RECEIPT

21912
VAN PELT, YI & JAMES LLP
10050 N. FOOTHILL BLVD #200
CUPERTINO, CA 95014

OC000000051697626

Date Mailed: 01/03/2012

Receipt is acknowledged of this provisional patent application. It will not be examined for patentability and will become abandoned not later than twelve months after its filing date. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Applicant(s)

Bjorn Markus Jakobsson, Mountain view, CA;

Power of Attorney:
Robyn Wagner--50575

If Required, Foreign Filing License Granted: 12/28/2011

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 61/569,112**

Projected Publication Date: None, application is not eligible for pre-grant publication

Non-Publication Request: No

Early Publication Request: No
**** SMALL ENTITY ****
Title

BACKWARDS COMPATIBLE ROBUST COOKIES

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international

patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4158).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and

Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation and commercialization of new technologies. The USA offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to encourage, facilitate, and accelerate business investment. To learn more about why the USA is the best country in the world to develop technology, manufacture products, and grow your business, visit SelectUSA.gov.

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

ACKNOWLEDGEMENT OF LOSS OF ENTITLEMENT TO ENTITY STATUS DISCOUNT

APPLICATION #	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET #	REQUEST ID
61/569,112	12/09/2011	Bjorn Markus Jakobsson	MJAKP007+	63422

The entity status change request below filed through Patent Center on 02/13/2023 has been accepted.

Certifications

APPLICANT CHANGING TO REGULAR UNDISCOUNTED FEE STATUS

Signature

I certify, in accordance with 37 CFR 1.4(d)(4), that I am one of the signatories making the entity status change.

Signature	Name	Registration #
/Robyn Wagner/	R Wagner	50575