Exhibit E

U.S. Patent Number US 9,679,289 – Preliminary Infringement Contentions[1]

Assignee:	Proxense, LLC
Title:	Hybrid device having a personal digital key and receiver-decoder circuit and methods of use
Filing Date:	2015-12-07
Publication Date:	2017-06-13
Inventor:	Brown, David L.

'289 Patent Claim[2]		**Accused Instrumentality And Where Each Claim Element Is Found[3]**
14	A method comprising[4]:	This preamble is not limiting. Microsoft sells and offers for sale computer user authentication services, called "passwordless" authentication (and/or sometimes referred to as Microsoft Entra, or Microsoft Identity Services), that practice the claimed methods that directly infringe this claim. Microsoft's servers, in combination with, its subscribers (which Microsoft directs and controls through online instruction, agreements, and direct prompts to action to receive the benefits of not having to maintain an authentication server and/or

[1] The Preliminary Infringement Contentions (PICS) provided herein are based on information obtained to date and may not be exhaustive. Plaintiff's investigation of Defendants' infringement is ongoing. Plaintiff reserves the right to supplement and/or amend these PICS to identify additional instrumentalities and to further identify where each element of each claim is found in each accused instrumentality, including on the basis of discovery obtained from Defendants, and from third parties during the course of this litigation, pursuant to ¶2 of the Order Governing Proceedings – Patent Cases under Hon. Alan D. Albright.

[2] All PICS set forth herein for any independent patent claims are hereby incorporated by reference into the PICS alleged for any dependent patent claims that depend on such independent claims, as if fully set forth therein.

[3] The Accused Instrumentalities and associated exhibits discussed and/or cited for any claim herein are representative in all material respects of all other accused instrumentalities identified for that claim (e.g., a specified device or service may be used as a representative example in these charts since the other accused instrumentalities have immaterial differences in their hardware and/or software configuration, the cited references are believed to be illustrative of all such accused devices).

[4] Plaintiff's inclusion of any claim preamble in this claim chart should not be interpreted as an admission that the preamble is limiting. Plaintiff reserves the right to take the position that the claim preambles are limiting or not limiting on a claim-by-claim basis.

		permit the use of a Microsoft Account to sign on users), perform the claimed method, literally or by the doctrine of equivalents, for at least the reasons set forth below:
	creating a first wireless link between an integrated receiver-decoder circuit (RDC) of a hybrid device and an external device, the hybrid device including an integrated, secure memory and the integrated RDC, wherein the integrated, secure memory stores local, secured information;	*the hybrid device including an integrated, secure memory … wherein the integrated, secure memory stores local, secured information* Microsoft requires its subscribers to use FIDO compliant authenticators (i.e., security keys) to use its "passwordless" authentication services. Security keys contain a personal digital key (PDK). To be used with Microsoft's Identity Platform, an authenticator needs to have FIDO2 certification. PROX_MSFT_002844, Become a Microsoft-Compatible FIDO2 Security Key Vendor for sign-in to Azure AD - Microsoft Entra | Microsoft Learn ("First, your authenticator needs to have a FIDO2 certification. We aren't able to work with providers who don't have a FIDO2 certification... **Microsoft adds your FIDO2 Security Key on Azure Active Directory backend and to our list of approved FIDO2 vendors**.") (emphasis added). Microsoft lists providers of FIDO2 security keys compatible with its "passwordless experience." See PROX_MSFT_002830, Azure Active Directory passwordless sign-in - Microsoft Entra | Microsoft Learn, By way of example, Feitian is a provider listed on Microsoft's website with a hyperlink to a co-branded website at PROX_MSFT_003278, https://shop.ftsafe.us/pages/microsoft. The cobranding is shown as: Microsoft | FEITIAN WE BUILD SECURITY The Microsoft/Feitian site further states: FEITIAN Passwordless Security Keys & OTP Tokens for Microsoft Applications Online Security is our priority! Microsoft users can now be protected with Passwordless authentication. As a member of the Microsoft Intelligent Security Association, FEITIAN Technologies has partnered with Microsoft to provide:

Path to Passwordless Kits, Individual Passwordless Security Keys, and time-based OTP Tokens and Card Options for Multi-factor authentication to safeguard your Microsoft account and access to Microsoft applications.

Id.

One of the Feitian's Microsoft-compatible security keys shown on the Microsoft/Feitian website, the AllinPass FIDO Security Key, is reproduced below:

See also PROX_MSFT_003276, https://cdn.shopify.com/s/files/1/0053/2889/6034/files/Flyer_AllinPass_FIDO2_K33.pdf?v=1597275700

All Microsoft-compatible authenticators, including the exemplary Feitian security key shown above (e.g., members of the Microsoft Intelligent Security Association) are used to practice the claimed method and have met the standards set by Microsoft and FIDO.

Further information regarding how vendors may become "Microsoft-compatible" is described at the following Microsoft website: PROX_MSFT_002844, https://learn.microsoft.com/en-us/azure/active-directory/authentication/concept-fido2-hardware-vendor

A screenshot of Microsoft's website is reproduced below:

Become a Microsoft-compatible FIDO2 security key vendor

Article • 06/14/2023 • 6 contributors

Feedback

In this article

Current partners
Next steps

Most hacking related breaches use either stolen or weak passwords. Often, IT enforce stronger password complexity or frequent password changes to reduce the risk of a security incident. However, this increases help desk costs and leads to poor user experiences as users are required to memorize or store new, complex passwords.

FIDO2 security keys offer an alternative. FIDO2 security keys can replace weak credentials with strong hardware-backed public/private-key credentials that can't be reused, replayed, or shared across services. Security keys support shared device scenarios, allowing you to carry your credential with you and safely authenticate to an Azure Active Directory joined Windows 10 device that's part of your organization.

Microsoft partners with FIDO2 security key vendors to ensure that security devices work on Windows, the Microsoft Edge browser, and online Microsoft accounts. FIDO2 security keys enable strong password-less authentication.

Notably, in step 2 of the process to become a Microsoft-compatible FIDO2 security key vendor, Microsoft requires that after a vendor obtains FIDO 2 certification, it must submit a request form to become a "Microsoft-compatible FIDO2 security key vendor," whereby Microsoft's "engineering team only confirms the features supported by [the] FIDO2 devices." *Id*.

As noted above, per the specification of the 289 Patent, integrated, secure memory storing local, secured information is memory holding information that can only be accessed with the corresponding access key. Such memory is present in FIDO compliant authenticators. The FIDO CTAP specification incorporates the WebAuthn Specification. PROX_MSFT_002849, Client to Authenticator Protocol (CTAP) (fidoalliance.org), § 1.1 ("This specification is part of the FIDO2 project, which includes this specification and is related to the W3C [WebAuthn] specification."). Under the WebAuthn specification, "compliant authenticators protect public key credentials." PROX_MSFT_003098, Web Authentication: An API for accessing Public Key Credentials - Level 2 (w3.org), § 1. A public key credential refers to a public key credential source, which includes a credential ID. PROX_MSFT_003098, Web Authentication: An API for accessing Public Key Credentials - Level 2 (w3.org), § 4 (Defining "public key credential" and "public key credential source".). The credential ID uniquely identifies its

		public key credential source. *See* PROX_MSFT_003098, Web Authentication: An API for accessing Public Key Credentials - Level 2 (w3.org), § 4 (Defining a credential ID as "A probabilistically-unique byte sequence identifying a public key credential source and its authentication assertions."). In addition to the credential ID, each public key credential source contains a "credential private key". PROX_MSFT_003098, Web Authentication: An API for accessing Public Key Credentials - Level 2 (w3.org), § 4, (Defining "public key credential source" as data structure including the credential private key and the credential ID.). "The credential private key is bound to a particular authenticator" and part of an asymmetric key pair containing a public key returned to a relying party. PROX_MSFT_003098, Web Authentication: An API for accessing Public Key Credentials - Level 2 (w3.org), § 4 (Defining a "credential key pair".). Every FIDO compliant authenticator, therefore, will store within memory a credential comprising a private key of an asymmetric key pair and a credential ID uniquely identifying the private/public key pair to which the private key belongs. The credentials stored within a compliant authenticator can only be accessed with the appropriate access key. "A public key credential can only be used for authentication with the same entity (as identified by the RP ID) it was registered with." PROX_MSFT_003098, Web Authentication: An API for accessing Public Key Credentials - Level 2 (w3.org), § 4 ("A public key credential can only be used for authentication with the same entity (as identified by RP ID) it was registered with."). When generating a response, therefore, the authenticator will only retrieve credentials corresponding to the RP ID provided to it by the external FIDO server. PROX_MSFT_002849, Client to Authenticator Protocol (CTAP) (fidoalliance.org), § 6.2.2 ("7.1 If the allowList parameter is present and is non-empty, locate all denoted credentials created by this authenticator and bound to the specified rpId. 7.2 If an allowList is not present, locate all discoverable credentials that are created by this authenticator and bound to the specified rpId."). As only credentials corresponding to the RP ID will be retrieved, the RP ID is an access key. During a WebAuthn authentication ceremony, an authenticator receives an "authenticatorGetAssertion" request to provide cryptographic proof of user authentication. PROX_MSFT_002849, Client to Authenticator Protocol (CTAP) (fidoalliance.org), § 6.2 (Defining authenticatorGetAssertion as the method "used by a host to request cryptographic proof of user authentication as well as user consent to a given transaction, using a previously generated credential that is bound to the authenticator and relying party identifier."). The authenticatorGetAssertion request contains a relying party identifier (RP ID). PROX_MSFT_002849, Client to Authenticator Protocol (CTAP) (fidoalliance.org), § 6.2 (Defining the input parameters of the authenticatorGetAssertion as including a required relying party identifier.). The authenticatorGetAssertion is called in response to get request issued by the relying party attempting authentication. PROX_MSFT_003098, Web Authentication: An

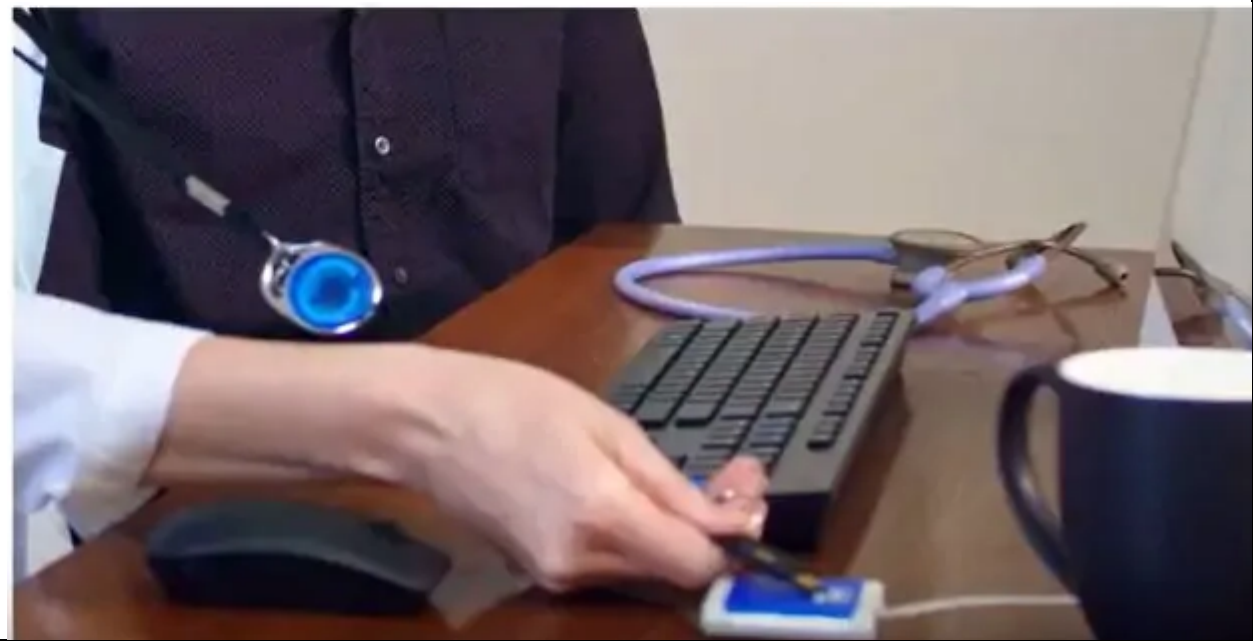
API for accessing Public Key Credentials - Level 2 (w3.org), § 5.1.4 ("WebAuthn Relying Parties call navigator.credentials.get({publicKey:..., ...}) to discover and use an existing public key credential, with the user's consent… If the user picks a credential source, the user agent then uses § 6.3.3 The authenticatorGetAssertion Operation to sign a Relying Party-provided challenge and other collected data into an assertion, which is used as a credential."). When get() is called by the relying party, "The RP ID defaults to being the caller's origin's effective domain unless the caller has explicitly set options.rpId when calling get()."PROX_MSFT_003098, Web Authentication: An API for accessing Public Key Credentials - Level 2 (w3.org), § 5.1.4. The RP ID, therefore, is provided by the by the relying party attempting authentication external authenticator. As an authenticator will only return credentials corresponding to the RP ID access key provided by the external relying party, the authenticator has integrated, secure memory storing local, secured information – i.e., memory holding information that can only be accessed with the corresponding access key.

the hybrid device including … the integrated RDC

See also PROX_MSFT_003276, https://cdn.shopify.com/s/files/1/0053/2889/6034/files/Flyer_AllinPass_FIDO2_K33.pdf?v=1597275700

Microsoft-compatible FIDO compliant BLE capable authenticators, like the Feitian AllinPass FIDO2 shown above, contain an RDC.

The FIDO standard, which Microsoft complies with and requires its partners to comply with, requires all communications with BLE authenticators to be encrypted. PROX_MSFT_002849, Client to Authenticator Protocol (CTAP) (fidoalliance.org), § 11.4.2, ("[C]lients and authenticators MUST create and use a long-term link key (LTK) and SHALL encrypt all

		communications. Authenticator MUST never use short term keys."); *see also* PROX_MSFT_002844, Become a Microsoft-Compatible FIDO2 Security Key Vendor for sign-in to Azure AD - Microsoft Entra | Microsoft Learn. Microsoft-compatible FIDO compliant BLE capable authenticators, accordingly, include an RDC. *creating a first wireless link between an integrated receiver-decoder circuit (RDC) of a hybrid device and an external device* A first wireless link may be created between a BLE or NFC capable Microsoft-compatible FIDO2 security key (i.e., authenticator/RFC) and a Windows device (i.e., PDK) may be created. Exemplary screenshots of a video from another Microsoft-compatible security key provider, Yubico, showing an NFC wireless link between a Yubico device (i.e., RFC) and a Windows device (i.e., PDK) is shown below: 

		  See PROX_MSFT_003281, https://youtu.be/wl479T2t6eo (Yubico/Microsoft video showing wireless communication between a key fob, i.e., RDC, and computer, i.e. external PDK at 0:26 seconds).
	receiving a first signal, at the integrated RDC, via the first wireless link, from the external device;	During authentication, an authenticator receives an "authenticatorGetAssertion" request to provide cryptographic proof of user authentication. PROX_MSFT_002849, Client to Authenticator Protocol (CTAP) (fidoalliance.org), § 6.2 (Defining authenticatorGetAssertion as the method "used by a host to request cryptographic proof of user authentication as well as user consent to a given transaction, using a previously generated credential that is bound to the authenticator and relying party identifier."). Browsers, such as Microsoft Edge and Google Chrome, operating on Windows 11 forward the authenticatorGetAssertion to the external authenticator. PROX_MSFT_003282, Passwordless login with passkeys | Authentication | Google Developers ("When a user wants to sign in to a service that uses passkeys, their browser or operating system will help them select and use the right passkey."); and PROX_MSFT_002819, All about FIDO2, CTAP2 and WebAuthn - Microsoft Community Hub ("CTAP2 and WebAuthn define an abstraction layer that creates an ecosystem for strongly authenticated credentials. Any interoperable client (such as a native app or browser) running on a given 'client device' can use a standardized method to interact with any interoperable authenticator – which could mean a platform authenticator that is built into the client device or a roaming authenticator that is connected to the client device through USB, BLE, or NFC."). Windows supports use of BLE and NFC roaming authenticators with Edge and Chrome web browsers. *See* PROX_MSFT_002847, Browser support of FIDO2 passwordless authentication - Microsoft Entra | Microsoft Learn (Presenting a table indicating Windows supports NFC and BLE passkeys with Microsoft Accounts "created by consumers for services such as Xbox, Skype, or Outlook.com" via the Chrome and Edge web browsers.). Regardless of the web browser, Windows provides WebAuthn APIs enabling interactions with authenticators to take place. PROX_MSFT_002819, All about FIDO2, CTAP2 and WebAuthn - Microsoft Community Hub ("Windows 10 plays the part of the platform, hosting Win32 Platform WebAuthn APIs that enable clients to interact with Windows Hello in order for users to be prompted and interactions with authenticators to take place.") As such, Windows will connect with a roaming authenticator over BLE or NFC to forward to the authenticator the authenticatorGetAssertion request received via either the Edge or Chrome web browser.

	generating an enablement signal enabling one or more of an application, a function and a service.	An enablement signal is the message comprising information received from a PDK in exchange for the access key provided by the application, function, or service to be enabled that permits the application, function, or service to operate. During a WebAuthn authentication ceremony, a Microsoft-authorized authenticator receives an "authenticatorGetAssertion" request to provide cryptographic proof of user authentication. PROX_MSFT_002849, Client to Authenticator Protocol (CTAP) (fidoalliance.org), § 6.2 (Defining authenticatorGetAssertion as the method "used by a host to request cryptographic proof of user authentication as well as user consent to a given transaction, using a previously generated credential that is bound to the authenticator and relying party identifier."). When successfully invoked, the authenticatorGetAssertion causes the authenticator to return a response including "the credential identifier whose private key was used to generate the assertion." PROX_MSFT_002849, Client to Authenticator Protocol (CTAP) (fidoalliance.org), § 6.2.2. As noted above, a credential can only be accessed when an authenticator is applied by the appropriate relying party identifier, such that the relying party identifier is an access key. When generating a response, therefore, the authenticator will only retrieve credentials corresponding to the RP ID provided to it by the external FIDO server. PROX_MSFT_002849, Client to Authenticator Protocol (CTAP) (fidoalliance.org), § 6.2.2 ("7.1 If the allowList parameter is present and is non-empty, locate all denoted credentials created by this authenticator and bound to the specified rpId. 7.2 If an allowList is not present, locate all discoverable credentials that are created by this authenticator and bound to the specified rpId."). Authentication is a service provided by a FIDO server operated by Microsoft, and the credential ID is necessary for Microsoft's FIDO server to perform the authentication function. Upon receiving the response (i.e., enablement signal), the WebAuthn/FIDO server will use the credential ID to locate the appropriate public key to verify a signature generated with the private key held by the authenticator. Web Authentication: An API for accessing Public Key Credentials - Level 2 (w3.org), § 7.2 ("7. Using credential.id (or credential.rawId, if base64url encoding is inappropriate for your use case), look up the corresponding credential public key and let credentialPublicKey be that credential public key… 20. Using credentialPublicKey, verify that sig is a valid signature over the binary concatenation of authData and hash… 22. I f all the above steps are successful, continue with the authentication ceremony as appropriate. Otherwise, fail the authentication ceremony.") "[I]f an authenticator returns the wrong credential ID, or if an attacker intercepts and manipulates the credential ID, is that the WebAuthn Relying Party would not look up the correct credential public key with which to verify the returned signed authenticator data (a.k.a., assertion), and thus the interaction would end in an error." PROX_MSFT_003098, Web Authentication: An API for accessing Public Key Credentials - Level 2 (w3.org), § 13.1. As the proper credential ID is needed for Microsoft's FIDO server to authenticate a user, and the credential ID is included within a response to a get request having the appropriate relying party ID received from the Microsoft's FIDO server, the response to the authenticatorGetAssertion

		request generated by the authenticator is an enablement signal enabling authentication by Microsoft's FIDO server.
16	The method of claim 14 further comprising:	
	sending the enablement signal	The responsibility for sending responses received from an authenticator via BLE or NFC falls upon the WebAuthn Client. PROX_MSFT_003098, Web Authentication: An API for accessing Public Key Credentials - Level 2 (w3.org), § 4 ("A WebAuthn Client is an intermediary entity typically implemented in the user agent (in whole, or in part). Conceptually, it underlies the Web Authentication API and embodies the implementation of the [[Create]](origin, options, sameOriginWithAncestors) and [[DiscoverFromExternalSource]](origin, options, sameOriginWithAncestors) internal methods. It is responsible for both marshalling the inputs for the underlying authenticator operations, and for returning the results of the latter operations to the Web Authentication API's callers."). Microsoft identifies its Edge Browser as the WebAuthn client. PROX_MSFT_002819, All about FIDO2, CTAP2 and WebAuthn - Microsoft Community Hub ("Microsoft Edge plays the part of a WebAuthn Client. Edge can handle the UI for the above listed WebAuthn/CTAP2 features, and also supports the AppID extension. Edge is capable of interacting with both CTAP1 and CTAP2 authenticators, which means that it can facilitate the creation and use of both U2F and FIDO2 credentials…"). As such, Microsoft's Edge Browser will forward the enablement signal received from an authenticator to Microsoft's FIDO server. Other web browsers, like Chrome, also act as a WebAuthn client and are compatible with Microsoft's UPA (e.g., by forwarding the enablement signal received from a roaming authenticator to Microsoft's FIDO server. See e.g. PROX_MSFT_003062, Passkey support on Android and Chrome | Authentication | Google Developers ("Chrome on all desktop platforms supports using passkeys from mobile devices."); and PROX_MSFT_003062, Passkey support on Android and Chrome | Authentication | Google Developers (Providing a table indicating "Can sign in with phone" is supported on Chrome running on the macOS, iOS, and Windows.).
	wherein at least one of the one or more of the application, the function, and the service is enabled at least in part on a device external to the hybrid device and communicatively coupled to an external RDC.	Microsoft's FIDO server is communicatively coupled with either an instance the Edge or Chrome browser running on the Windows device linked to the authenticator. PROX_MSFT_003282, Passwordless login with passkeys | Authentication | Google Developers ("When a user wants to sign in to a service that uses passkeys, their browser or operating system will help them select and use the right passkey."); and PROX_MSFT_002819, All about FIDO2, CTAP2 and WebAuthn - Microsoft Community Hub ("CTAP2 and WebAuthn define an abstraction layer that creates an ecosystem for strongly authenticated credentials. Any interoperable client (such as a native app or browser) running on a given 'client device' can use a standardized method to interact with any interoperable authenticator – which could mean a platform authenticator that is built into the client device or a roaming authenticator that is connected to the client device through USB, BLE, or NFC."). The authenticator, accordingly, generates an enablement signal enabling one or more of an application, a

		function and a service on a device external to the hybrid device and communicatively coupled to an external RDC.