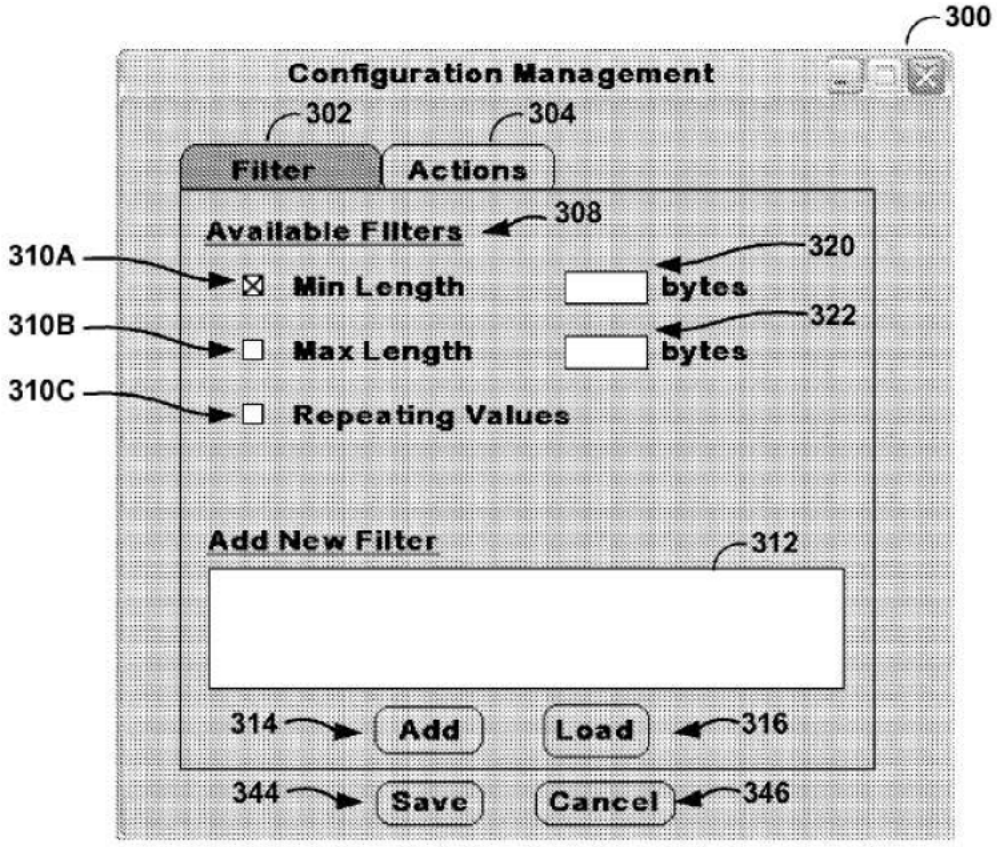
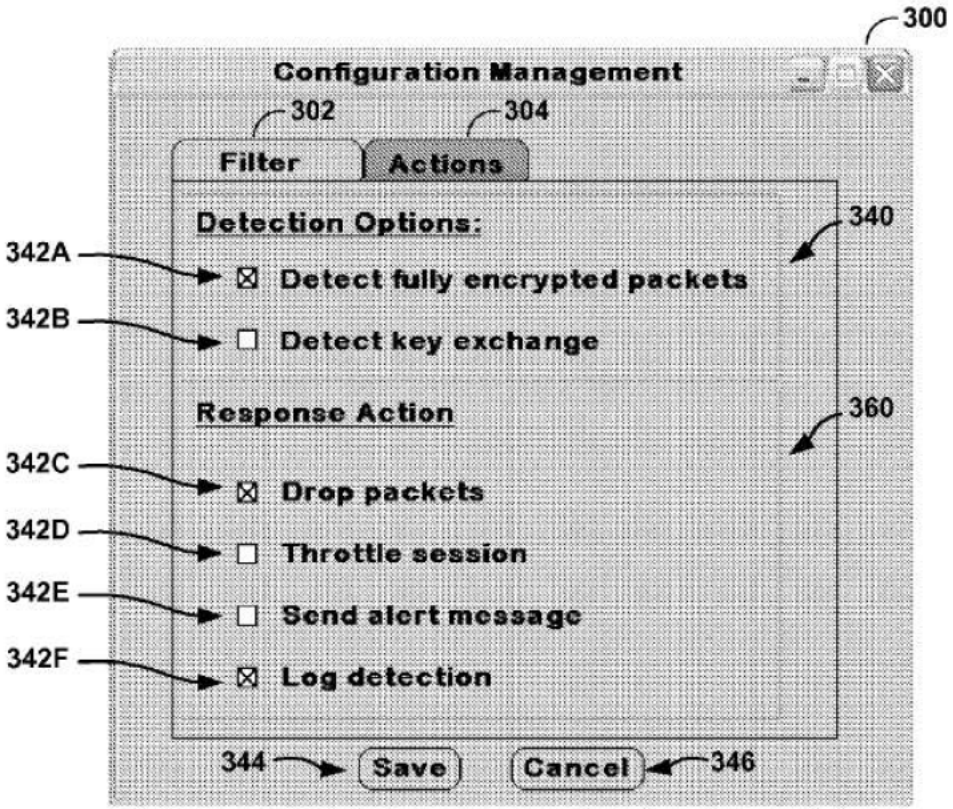
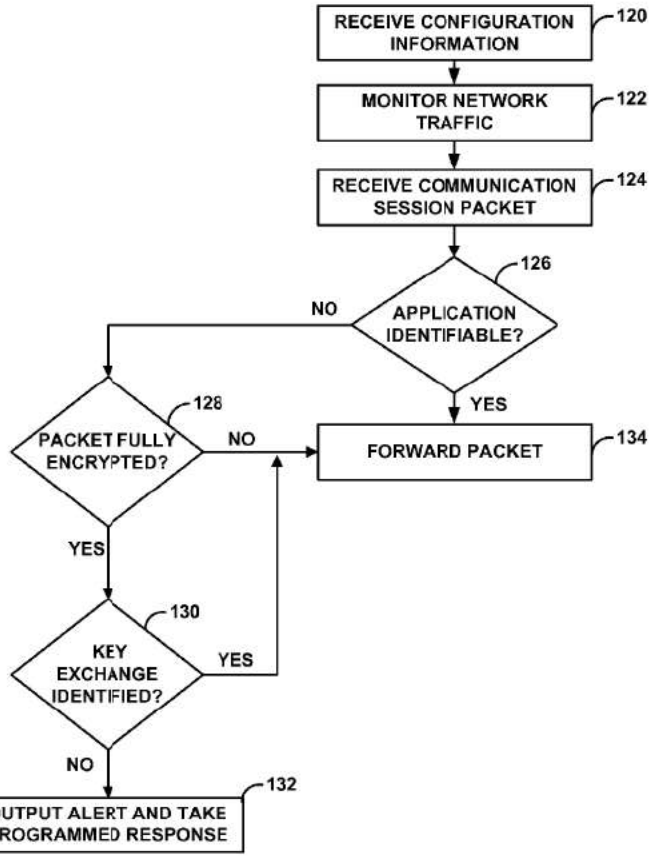
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	 FIG. 7A (Burns, Fig. 7A.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	 FIG. 7B (Burns, Fig. 7B.) *See* also claim element 17[c].

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
1[d]. modifying data packets corresponding to requests for sharing of responses that are not compatible with the received privacy preference; and	Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Burns' IDS provides network security by inhibiting communications (*e.g.*, blocking, closing, or throttling sessions) that are not security-compatible, and each of these security actions involves modifying data packets. *See* also claim 18.
1[e]. maintaining communication between the remote server and the user's device, with sharing of the modified data packets.	Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Burns' IDS operates "transparently" and forwards selectively modified network traffic. "FIG. 2 is a block diagram illustrating an example embodiment of an IDS **20**. In the illustrated example, IDS **20** includes a forwarding plane **22** that transparently monitors inbound network traffic **24** and forwards the network traffic as outbound network traffic **26**. In the example illustrated by FIG. 2, forwarding plane **22** includes flow analysis module **25**, stateful inspection engine **28**, protocol decoders **30**, forwarding component **31** and security management module **44**." (Burns, 6:29-36.) *See* also claim element 17[e].
2. The method of claim 1, wherein the communication data type of the at least one data packet is determined from metadata of communication with the remote server.	Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Burns describes identifying a packet's communication data type based in part on a TCP port value. The TCP port value in a packet header is "metadata." *See* claim element 1[b].

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
3. The method of claim 1, further comprising:	*See* claim 1.
3[a]. determining, by the user's device, at least one communication data pattern of at least one data packet corresponding to the received request; and	Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Burns' IDS analyzes a received packet to identify a corresponding application or protocol and data patterns, such as repeated data and encryption key exchanges. *See* claim elements 17[e] and claim 18.
3[b]. modifying data packets corresponding to requests for sharing of responses corresponding to at least one data pattern, that are not compatible with the received privacy preference,	Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Burns' IDS is configurable to take various actions in handling a packet that is not security-compatible, including modifying a packet by (1) dropping (blocking) the packet, (2) closing the related communication session, or (3) throttling (limiting bandwidth consumption of) the related communication session. These actions may be provided by an administrator. *See* claim element 1[c] and claim 18.
3[c]. wherein the privacy preference comprises a list of allowed data patterns for sharing during communication.	Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Burns teaches data patterns, such as repeated data and encryption key exchanges, and Burns' IDS allows packets to pass according to these data patterns.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	 **FIG. 5** (Burns, Fig. 5.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	"Where IDS **10** is able to identify the application using a particular communication session, IDS **10** may either permit or prevent the communication session from continuing. For example, a system administrator may configure IDS **10** to explicitly allow all identifiable applications, allow all applications except for a specified list of identifiable applications, or prevent all communications except for communications from a specified list of permitted software applications. In cases where IDS **10** is unable to identify the application, IDS **10** may further inspect characteristics of the data from the communication session to determine whether the data is encrypted. As one example, IDS **10** may identify an encrypted communication session by determining whether the data in the communication session exceeds a randomness threshold. A system administrator may configure IDS **10** to establish or modify the randomness threshold. When the data from the communication session exceeds the randomness threshold, IDS **10** determines that the data is encrypted. Upon determining that the packet flows of the communication session are encrypted, IDS **10** applies the principles described herein to determine whether the communication session is associated with an allowed application or a typically unwanted application. For example, IDS **10** may then determine whether a proper key exchange was initially detected for the communication session and/or whether the communication session contains fully encrypted packets or whether only application-layer data within the packets is encrypted." (Burns, 5:27-53.) "Another exemplary method for determining whether a packet is encrypted includes identifying repeating patterns of varying lengths, e.g., one byte, two bytes, four bytes, or other within a packet. That is, encryption detection module **58** may determine that a packet is not random when encryption detection module **58** determines that there exist repeating patterns within the packet." (Burns, 14:32-38.) *See* also claim elements 1[c], 17[c], and 17[e], and claim 18.
4. The method of claim 3, wherein the communication data pattern is determined based on a range selected	Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
from a group consisting of: data packet frequency, data packet size, data packet speed, data packet count, data packet ratio compared to other data type flow, data packet repetition, and data packet order.	For example, Burns describes identifying a repeating pattern (*e.g.*, "repetition"), sequence, or value across multiple packets in a communication session. *See* claims 3 and 18.
5. The method of claim 1, wherein the communication data type is determined based on data packet characteristics selected from a group consisting of: data packet header, data packet footer, version number, IP (Internet Protocol) address, HTTPS (HyperText Transfer Protocol Secure), file extension, encryption method, encoding method, keywords, driver, and communication protocol.	Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Burns determines a packet's communication data type by comparing a TCP port value to a list of known applications. The TCP port value is in the TCP header. *See* claim element 1[b].
6. The method of claim 1, further comprising:	*See* claim 1.
6[a]. determining, by the remote server, a communication data pattern of the at least one data packet of the received request; and	Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Burns' IDS analyzes a received packet to identify a corresponding application or protocol and data patterns, such as repeated data and encryption key exchanges. *See* claim element 17[e] and claim 18.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
6[b]. linking at least one response from the user's device to a type of data packet from the user's device, wherein the linking is based on the communication data type and communication data pattern of the at least one response.	Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Burns' IDS identifies (which involves linking) an application or protocol of packets passing through it, which, in turn, is based on a pattern. For instance, Burns' IDS may identify the packets associated with the request for user input using the HTTP application or protocol. As another example, Burns discloses applying an "attack definition" to identify a packet that is "malicious." "FIG. 1 is a block diagram illustrating an exemplary system **2** in which enterprise computer network **4** includes intrusion detection system (IDS) **10** that intercepts packet flows associated with communication sessions and processes the packet flows to filter unwanted or malicious packet flows in accordance with the principles of the invention. In the example embodiment of FIG. 1, IDS **10** is a single network device. Network **4** also includes a private enterprise computing network **5** that is coupled to public network **6**, such as the Internet. Public network **6** may include, for example, one or more client computing devices. Firewall **9** protects enterprise network **5** and, in particular, internal computing nodes **8**A-**8**N. Computing nodes **8**A-**8**N (computing nodes **8**) represent any private computing device within enterprise network **5**, including workstations, file servers, print servers, database servers, printers and other devices. In one embodiment, IDS **10** attempts to identify applications and protocols for each communication session between computing nodes **8** and other computing devices in public network **6**. Exemplary techniques for identifying specific applications and protocols are described in greater detail in U.S. patent application Ser. No. 11/835,923, Burns et al., "Identifying Applications for Intrusion Detection Systems," filed Aug. 8, 2007, assigned to the assignee of the present application, which is hereby incorporated by reference in its entirety." (Burns, 4:20-45.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	"Security management module **44** presents a user interface by which administrator **42** configures IDS **20**. For example, administrator **42** may configure IDS **20** to monitor particular subnets of the enterprise network. In addition, security management module **44** presents a user interface by which administrator **42** may specify attack definitions **33**, which security management module **44** relays to stateful inspection engine **28**. In one embodiment, attack definitions **33** may be compound attack definitions." (Burns, 6:37-45.) "Initially, security management module **44** receives configuration information from administrator **42** and, in response, configures IDS **20** to monitor a network or portions thereof (subnets) of interest (**70**). During this process, configuration manager **44** presents a user interface by which administrator **42** specifies patterns or other attack definitions **33**. For example, administrator **42** may configure IDS **20** to block all packets deemed to be fully encrypted and to log information about the communication session associated with the blocked packets. Administrator **42** may also configure IDS **20** to throttle down (i.e., bandwidth limit) the communication session associated with the packets to minimize the bandwidth used by the communication session. Administrator **42** may also configure IDS **20** to block future communication sessions from either or both of the communicating devices. IDS **20** may additionally include a default configuration that specifies actions IDS **20** should take when IDS **20** detects a fully encrypted packet. IDS **20** may, for example, be programmed to let such packets pass by default, or to block all fully encrypted packets by default. Once configured, IDS **20** monitors network traffic **24** (**72**). In some configurations, stateful inspection engine **28** of forwarding plane **22** may receive network traffic and mirror the network traffic for purposes of analysis. Forwarding component **31** seamlessly forwards the original network traffic. Stateful inspection engine **28** of forwarding plane **22** next receives a packet, e.g. from flow analysis module **25** (**74**). While stateful inspection engine **28** may receive all packets of a

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	particular communication session, for the purposes of this discussion, it is assumed that this packet is the first packet following the packets used to establish the three-way handshake of TCP/IP. In general, IDS **20** may be capable of identifying certain network attacks as early as with this first packet." (Burns, 9:64-10:31.) *See* also claim element 1[b].
7. The method of claim 1, further comprising:	*See* claim 1.
7[a]. identifying a response corresponding to a request for sharing of data packets, wherein the identified response is not compatible with the received privacy preference; and	Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Burns' IDS monitors network traffic flowing in both directions between a client and server, including "identify[ing] pairs of [unidirectional] packet flows that collectively form a single communication session." Burns' IDS provides network security by inhibiting disallowed communications (*e.g.*, blocking, closing, or throttling sessions in accordance with an administrator's configuration information). Burns' IDS performs its security functions on all packets, including response packets. "Flow analysis module **25** maintains data within flow table **35** that describes each active packet flow present within the network traffic. Flow table **35** specifies network elements associated with each active packet flow, i.e., low-level information such as source and destination devices and ports associated with the packet flow. In addition, flow table **35** may identify pairs of packet flows that collectively form a single communication session between a client and server. For example, flow table **35** may designate communication session as pairs of packet flows in opposite directions for flows sharing at least some common network addresses, ports and protocol." (Burns, 7:3-15.) "Once configured, IDS **20** monitors network traffic **24** (**72**). In some configurations, stateful inspection engine **28** of forwarding plane **22** may receive network

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	traffic and mirror the network traffic for purposes of analysis. Forwarding component **31** seamlessly forwards the original network traffic. Stateful inspection engine **28** of forwarding plane **22** next receives a packet, e.g. from flow analysis module **25** (**74**). While stateful inspection engine **28** may receive all packets of a particular communication session, for the purposes of this discussion, it is assumed that this packet is the first packet following the packets used to establish the three-way handshake of TCP/IP. In general, IDS **20** may be capable of identifying certain network attacks as early as with this first packet." (Burns, 10:18-31.) *See* also claim element 1[d] and claims 6 and 18.
7[b]. providing a response with at least one modified data packet corresponding to the request for sharing types of data packets.	Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Burns' IDS modifies a packet and then shares the modified packet. *See* claim elements 1[d] and 1[e] and claim 18.
8. The method of claim 7, wherein the modification of the at least one modified data packet is selected from the group consisting of data nullification, blocking, data randomization, content modification, change of encoding, change of file template, change of header, change of footer, addition of a predetermined data packet and encryption.	Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Burns describes selectively "dropping" packets based on the IDS's configuration, which corresponds to blocking. Burns also describes "automatically closing the communication session" and "throttle[ing]," which involve a change of header. "In the event a security risk is detected, stateful inspection engine **28** outputs an alert **40** to security management module **44** for logging and further analysis. In addition, stateful inspection engine **28** may take additional actions, such as dropping the packets associated with the communication session, automatically closing the communication session, or other actions. If no security risk is detected for a given

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	application-layer communication session, stateful inspection engine **28** instructs forwarding component **31** to forward the packet flows between the peers. Forwarding component **31** may, for example, maintain a routing table that stores routes in accordance with a topology of enterprise network **5** for use in forwarding the packet flows." (Burns, 7:45-57.) "Initially, security management module **44** receives configuration information from administrator **42** and, in response, configures IDS **20** to monitor a network or portions thereof (subnets) of interest (**70**). During this process, configuration manager **44** presents a user interface by which administrator **42** specifies patterns or other attack definitions **33**. For example, administrator **42** may configure IDS **20** to block all packets deemed to be fully encrypted and to log information about the communication session associated with the blocked packets. Administrator **42** may also configure IDS **20** to throttle down (i.e., bandwidth limit) the communication session associated with the packets to minimize the bandwidth used by the communication session. Administrator **42** may also configure IDS **20** to block future communication sessions from either or both of the communicating devices. IDS **20** may additionally include a default configuration that specifies actions IDS **20** should take when IDS **20** detects a fully encrypted packet. IDS **20** may, for example, be programmed to let such packets pass by default, or to block all fully encrypted packets by default." (Burns, 9:64-10:17.) "Initially, security management module **44** receives configuration information from administrator **42** and, in response, configures IDS **20** to monitor a network or portions thereof (subnets) of interest (**120**). During this process, configuration manager **44** presents a user interface by which administrator **42** specifies patterns or other attack definitions **33**. For example, administrator **42** may configure IDS **20** to block all fully encrypted packets and log information about the communication session associated with the packet. Administrator **42** may also configure IDS **20** to throttle down the communication session associated with the packet to minimize the bandwidth used by the

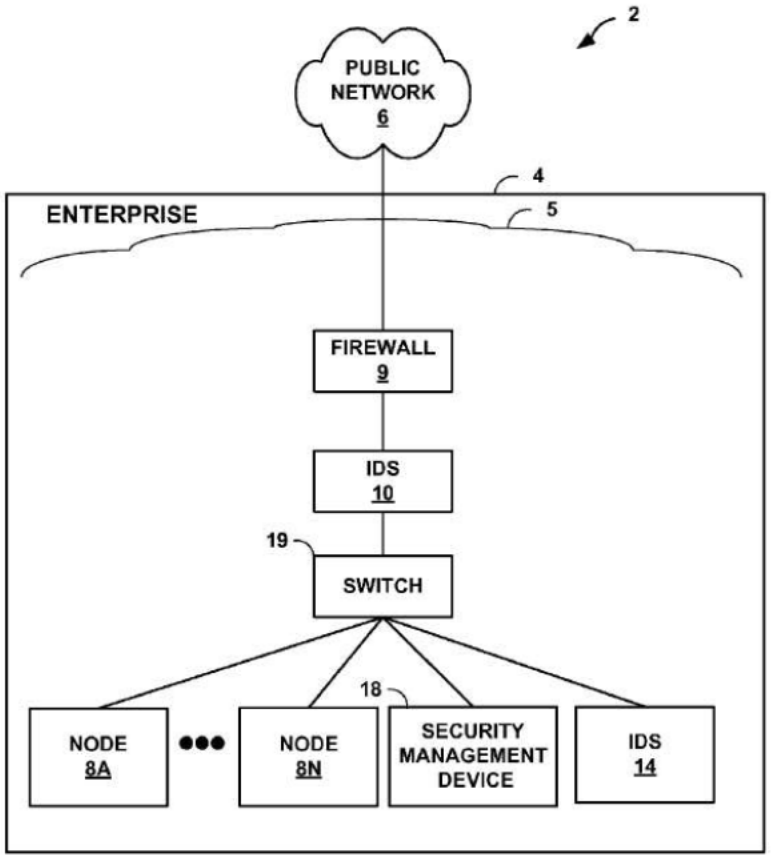
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	communication session. Administrator **42** may also configure IDS **20** to block future communication sessions from either or both of the communicating devices. IDS **20** may additionally include a default configuration that specifies actions IDS **20** should take when IDS **20** detects a fully encrypted packet. IDS **20** may, for example, be programmed to let such packets pass by default, or to block all fully encrypted packets by default." (Burns, 11:44-62.) *See* also claim 18.
9. A method of dynamic management of private data during communication between a remote server and a user's device, the method comprising:	*See* claim element 1[pre].
9[a]. receiving, by the user's device, a request for retrieval of at least one data packet from the user's device, wherein the user's device is configured to provide a response corresponding to the received request;	*See* claim element 1[a].
9[b]. determining, by the remote server, at least one communication data pattern corresponding to the received request, wherein the at least one communication data pattern is determined in accordance with a behavior range of the communication data packet, and wherein the content of the at least one data packet is not read by the remote server for	*See* claim elements 1[b], 6[a], and 17[e], claim 3.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
continued operation by the user's device in real time during communication between the remote server and the user's device;	
9[c]. receiving, by the user's device, a privacy preference for the user's device, wherein the privacy preference compromises a list of allowed data patterns for sharing during communication;	*See* claim element 1[c].
9[d]. modifying data packets corresponding to requests for sharing of responses and corresponding data patterns that are not compatible with the received privacy preference; and	*See* claim elements 1[d] and 6[b], and claim 3.
9[e]. maintaining communication between the remote server and the user's device, with sharing of the modified data packets.	*See* claim element 1[e].
10. The method of claim 9, wherein the communication data pattern of the at least one response is determined from metadata of communication with the remote server.	*See* claim 2 and claim element 6[a]. *See* also Burns, 2:31-43.
11. The method of claim 9, further comprising:	*See* claim 9.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
11[a]. determining, by the remote server, the communication data type of the at least one data packet of the received request; and	*See* claim element 1[b].
11[b]. modifying data packets corresponding to requests for sharing of responses and corresponding data patterns, corresponding to at least one communication data packet type, that are not compatible with the received privacy preference,	*See* claim elements 1[d], 3[b], and 17[e].
11[c]. wherein the privacy preference comprises a list of allowed data packet types for sharing during communication.	*See* claim element 1[c].
12. The method of claim 11, wherein the communication data type is determined based on data packet characteristics selected from a group consisting of: data packet header, data packet footer, version number, IP (Internet Protocol) address, HTTPS (HyperText Transfer Protocol Secure), file extension, encryption method, keywords, driver, and communication protocol.	*See* claim 5.
13. The method of claim 9, wherein the user's device is in communication	*See* claim elements 17[c], 17[e], and claim 18.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
with an external database with at least one communication data pattern corresponding to a request for sharing of data packets from the user's device and wherein at least one data pattern compatible with the received request is determined from the database.	
14. The method of claim 9, further comprising:	*See* claim 9.
14[a]. determining, by the remote server, a type of the at least one data packet of the received request; and	*See* claim element 1[b].
14[b]. linking at least one data pattern to a type of data packet from the user's device, wherein the linking is based on the communication data type and the communication data pattern of the at least one response.	*See* claim element 6[b].
15. The method of claim 9, further comprising:	*See* claim 9.
15[a]. identifying a communication data pattern corresponding to a request for sharing of data packet types, wherein the identified data pattern is not compatible with the received privacy preference; and	*See* claim 3.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
15[b]. providing a response with at least one modified data packet corresponding to the request for sharing types of data packets.	*See* claim 3.
16. The method of claim 15, wherein the modification of the at least one modified data packet is selected from the group consisting of data nullification, blocking, data randomization, content modification, change of encoding, change of file template, change of header, change of footer, addition of a predetermined data packet and encryption.	*See* claim 8.
17. A system for dynamic management of private data during communication between a remote server and at least one user's device, the system comprising:	To the extent the preamble is limiting, Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Burns describes a system for dynamically managing private data, such as encrypted data, between its intrusion detection system (IDS) and a "node" or a security management device.

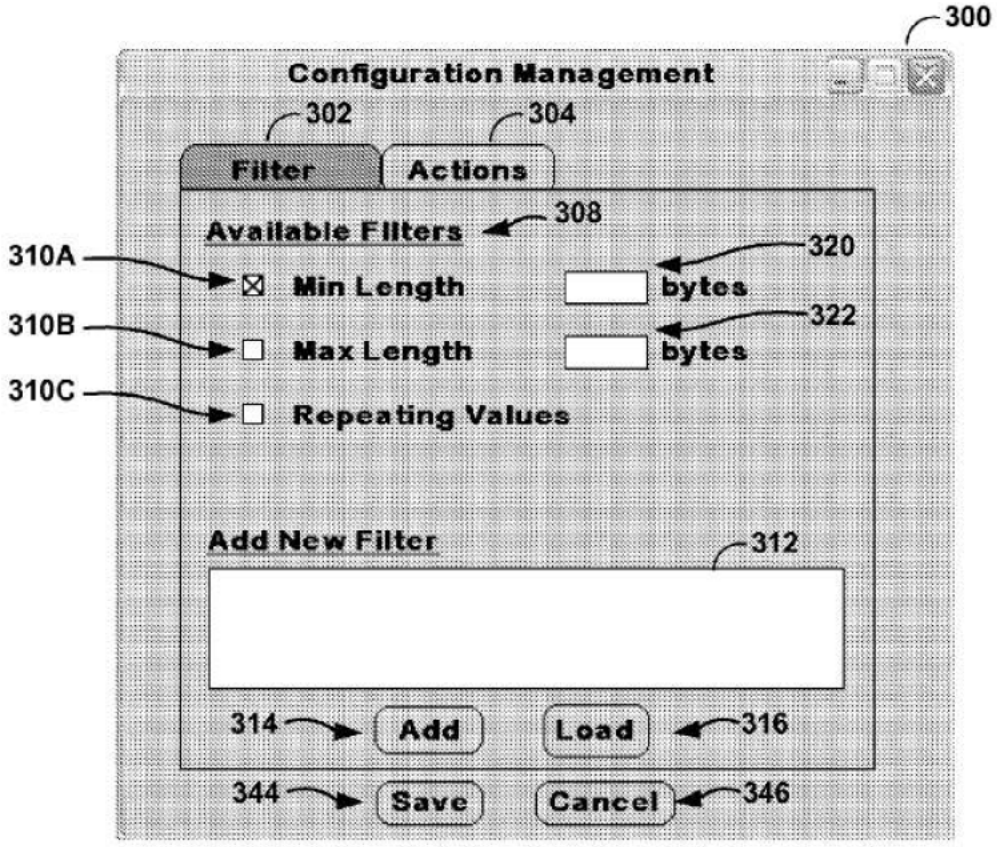
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	 **FIG. 1** (Burns, Fig. 1.)

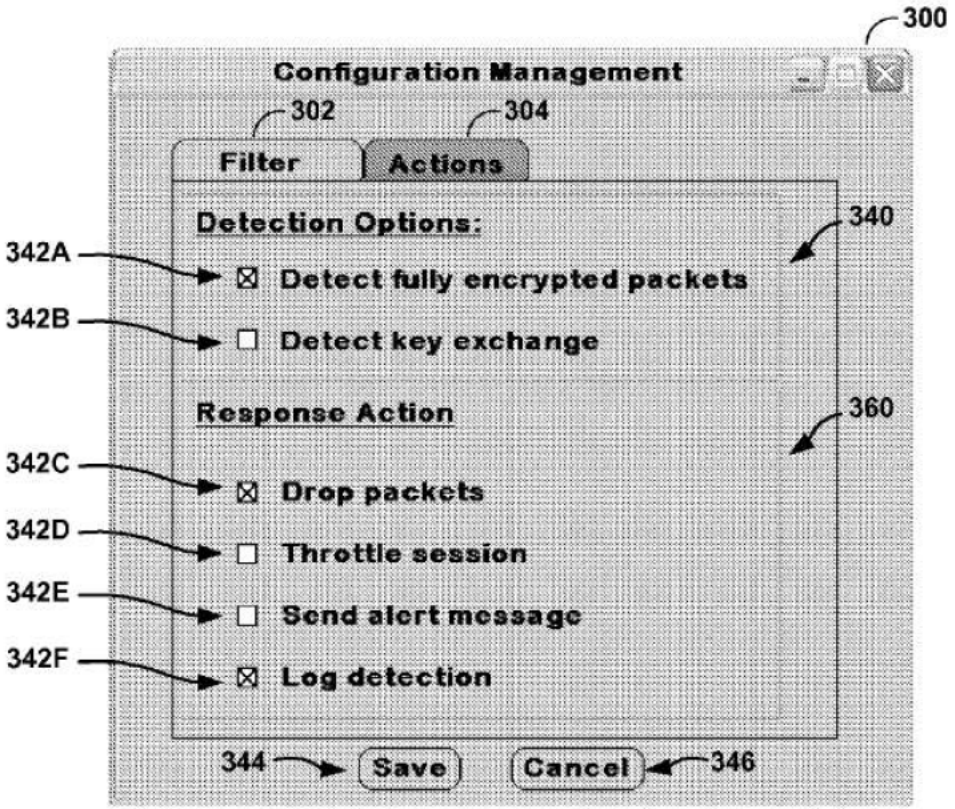
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	"Techniques are described for blocking unidentified encrypted communication sessions. In one embodiment, a device includes an interface to receive a packet, an application identification module to attempt to identify an application associated with the packet, an encryption detection module to determine whether the packet is encrypted when the application identification module is unable to identify an application associated with the packet, and an attack detection module to determine whether the packet is associated with a network attack, to forward the packet when the packet is not associated with a network attack, and to take a response when the packet is associated with a network attack, wherein the encryption detection module sends a message to the attack detection module that indicates whether the packet is encrypted, wherein when the message indicates that packet is encrypted, the attack detection module determines that the packet is associated with a network attack." (Burns, Abstract.) "FIG. 1 is a block diagram illustrating an exemplary system **2** in which enterprise computer network **4** includes intrusion detection system (IDS) **10** that intercepts packet flows associated with communication sessions and processes the packet flows to filter unwanted or malicious packet flows in accordance with the principles of the invention. In the example embodiment of FIG. 1, IDS **10** is a single network device. Network **4** also includes a private enterprise computing network **5** that is coupled to public network **6**, such as the Internet. Public network **6** may include, for example, one or more client computing devices. Firewall **9** protects enterprise network **5** and, in particular, internal computing nodes **8**A-**8**N. Computing nodes **8**A-**8**N (computing nodes **8**) represent any private computing device within enterprise network **5**, including workstations, file servers, print servers, database servers, printers and other devices. In one embodiment, IDS **10** attempts to identify applications and protocols for each communication session between computing nodes **8** and other computing devices in public network **6**. Exemplary techniques for identifying specific applications and protocols are described in greater detail in U.S. patent application Ser. No. 11/835,923, Burns et al., "Identifying Applications for Intrusion Detection Systems," filed Aug. 8, 2007, assigned to

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	the assignee of the present application, which is hereby incorporated by reference in its entirety." (Burns, 4:20-45.) "IDS **10** may be configurable in a variety of ways. For example, in one embodiment, IDS **10** may permit a communication session that is unencrypted to continue, despite being unable to identify a particular application that is using the communication session. In one embodiment, where IDS **10** determines that a communication session, for which an application cannot be identified, is encrypted, IDS **10** may terminate the communication session. IDS **10** may take other actions as well, such as dropping the packet, blocking future sessions from a computing device that initiated the communication session, or other actions. In one embodiment, where IDS **10** determines that a communication session, for which an application cannot be identified, is encrypted, IDS **10** may further determine whether a key exchange associated with the communication session can be identified; where a key exchange has been identified, IDS **10** may permit the communication session and where a key exchange has not been identified, IDS **10** may terminate the communication session. In one embodiment, IDS **10** may be configured to take any or all of the above actions, in accordance with a user configuration that a user, such as a system administrator, may configure. In some embodiments, enterprise network **5** includes multiple IDSs **10** and **14** located within different regions (e.g., sub-networks) of enterprise network **5**. Security management device **18** may operate as a central device for managing IDSs **10** and **14**. Although the example illustrated in FIG. 1 is described in terms of dedicated IDSs **10** and **14**, the functionality described herein may be incorporated within other devices, such as firewall **9** or switch **19**." (Burns, 5:54-6:17.) "Security management module **44** presents a user interface by which administrator **42** configures IDS **20**. For example, administrator **42** may configure IDS **20** to monitor particular subnets of the enterprise network. In addition, security management module **44** presents a user interface by

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	which administrator **42** may specify attack definitions **33**, which security management module **44** relays to stateful inspection engine **28**. In one embodiment, attack definitions **33** may be compound attack definitions." (Burns, 6:37-45.) "When encryption detection module determines that the packet is fully encrypted, however, ("YES" branch of **80**), encryption detection module **58** may output an alert and take a programmed response (**82**). The response to take may be in accordance with configuration data received from, e.g., administrator **42**, at step (**70**). For example, the programmed response may be to terminate the communication session, to block future communication sessions from the peer that initiated the communication session, or other actions." (Burns, 11:30-38.)
17[a]. a memory;	Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Burns comprises a memory, such as "random access memory (RAM)," "a hard drive, a network drive, [or] a flash memory stick." *See* Burns, 8:4-6, 18:51-53, 21:24-31.
17[b]. a communication data type database, comprising at least one communication data type corresponding to sharing of at least one data packet from the user's device;	Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Burns' IDS analyzes packets to determine an application or protocol for each communication session. These determinations may be saved to a database. "FIG. 1 is a block diagram illustrating an exemplary system **2** in which enterprise computer network **4** includes intrusion detection system (IDS) **10** that intercepts packet flows associated with communication sessions and processes the packet flows to filter unwanted or malicious packet flows in accordance with the principles of the invention. In the example embodiment of FIG. 1, IDS **10** is a single network device. Network **4** also

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	includes a private enterprise computing network **5** that is coupled to public network **6**, such as the Internet. Public network **6** may include, for example, one or more client computing devices. Firewall **9** protects enterprise network **5** and, in particular, internal computing nodes **8**A-**8**N. Computing nodes **8**A-**8**N (computing nodes **8**) represent any private computing device within enterprise network **5**, including workstations, file servers, print servers, database servers, printers and other devices. In one embodiment, IDS **10** attempts to identify applications and protocols for each communication session between computing nodes **8** and other computing devices in public network **6**. Exemplary techniques for identifying specific applications and protocols are described in greater detail in U.S. patent application Ser. No. 11/835,923, Burns et al., "Identifying Applications for Intrusion Detection Systems," filed Aug. 8, 2007, assigned to the assignee of the present application, which is hereby incorporated by reference in its entirety." (Burns, 4:20-45.) *See* also Burns, 9:46-10:45.
17[c]. a privacy preference database, comprising a list of allowed types of data packets for sharing during communication with the at least one user's device;	Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Burns' IDS is configurable, allowing its operation to be customized to match the privacy preferences of an administrator. For instance, the administrator may specify patterns, attack definitions, and desired responses to be applied in the context of encrypted communications. These privacy preferences may be saved to a database.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	 FIG. 7A (Burns, Fig. 7A.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	 FIG. 7B (Burns, Fig. 7B.)

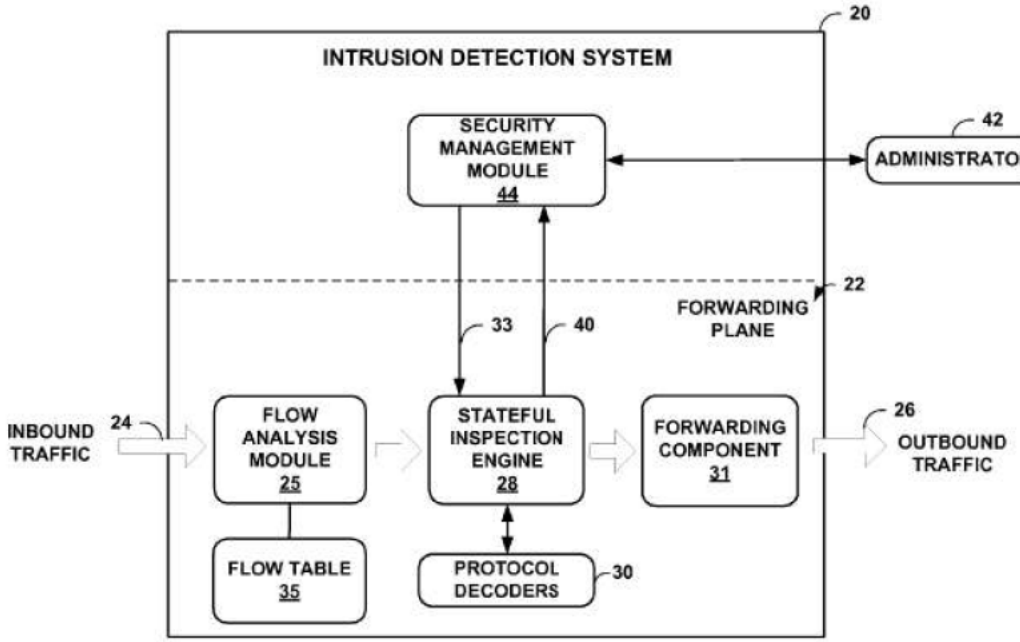
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	"In certain contexts, such as within a corporate enterprise network, a system administrator or other system owner would like to restrict or limit access to and usage of network resources. For example, the system administrator may wish to prevent network access by an unknown application because the unknown application may be malware, such as a virus, trojan, or other malicious software. The system administrator may also wish to prevent users from using certain applications due to various considerations. For example, the system administrator may wish to block network access to applications such as BitTorrent, Skype, or other "grayware" applications due to consumption of network resources or security concerns with respect to the applications. Therefore, the system administrator may configure a firewall or other network device to block certain applications and protocols." (Burns, 1:21-35.) "Where IDS **10** is able to identify the application using a particular communication session, IDS **10** may either permit or prevent the communication session from continuing. For example, a system administrator may configure IDS **10** to explicitly allow all identifiable applications, allow all applications except for a specified list of identifiable applications, or prevent all communications except for communications from a specified list of permitted software applications. In cases where IDS **10** is unable to identify the application, IDS **10** may further inspect characteristics of the data from the communication session to determine whether the data is encrypted. As one example, IDS **10** may identify an encrypted communication session by determining whether the data in the communication session exceeds a randomness threshold. A system administrator may configure IDS **10** to establish or modify the randomness threshold. When the data from the communication session exceeds the randomness threshold, IDS **10** determines that the data is encrypted. Upon determining that the packet flows of the communication session are encrypted, IDS **10** applies the principles described herein to determine whether the communication session is associated with an allowed application or a typically unwanted application. For example, IDS **10** may then determine whether a proper key exchange was initially detected for the communication session and/or whether the communication

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	session contains fully encrypted packets or whether only application-layer data within the packets is encrypted." (Burns, 5:27-53.) "Security management module **44** presents a user interface by which administrator **42** configures IDS **20**. For example, administrator **42** may configure IDS **20** to monitor particular subnets of the enterprise network. In addition, security management module **44** presents a user interface by which administrator **42** may specify attack definitions **33**, which security management module **44** relays to stateful inspection engine **28**. In one embodiment, attack definitions **33** may be compound attack definitions." (Burns, 6:37-45.) "Security management module 44 may further present a user interface by which administrator 42 may configure IDS 20 to take particular actions when a fully encrypted packet is identified. For example, the user interface may permit the administrator to drop all fully encrypted packets, log details about the communication session associated with the fully encrypted packets, drop the packets only if a key exchange has not been previously identified for the communication session, throttle down the communication session with fully encrypted packets to reduce the bandwidth usage of that communication session, or other actions. The user interface may also enable administrator 42 to configure a randomness threshold that IDS 20 uses to determine whether a packet is encrypted, as described further herein." (Burns, 6:46-59.) "Initially, security management module **44** receives configuration information from administrator **42** and, in response, configures IDS **20** to monitor a network or portions thereof (subnets) of interest (**70**). During this process, configuration manager **44** presents a user interface by which administrator **42** specifies patterns or other attack definitions **33**. For example, administrator **42** may configure IDS **20** to block all packets deemed to be fully encrypted and to log information about the communication session associated with the blocked packets. Administrator **42** may also configure IDS **20** to throttle down (i.e., bandwidth limit) the communication session

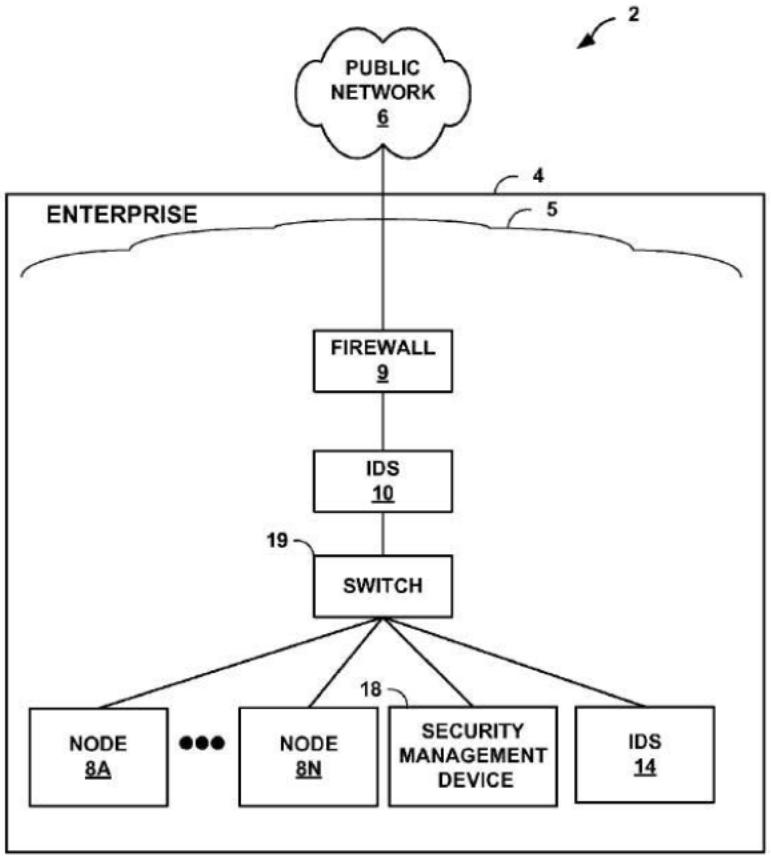
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	associated with the packets to minimize the bandwidth used by the communication session. Administrator **42** may also configure IDS **20** to block future communication sessions from either or both of the communicating devices. IDS **20** may additionally include a default configuration that specifies actions IDS **20** should take when IDS **20** detects a fully encrypted packet. IDS **20** may, for example, be programmed to let such packets pass by default, or to block all fully encrypted packets by default." (Burns, 9:64-10:17.) "Application identification module **51** attempts to identify an application associated with the received packet (**126**). For example, application identification module **51** inspects the packet header to determine the destination port of the packet. In some cases, the destination port is associated with the protocol or application being used for the communication session. For example, port **80** is typically associated with HTTP traffic, while port **22** is typically associated with SSH traffic. Application identification module **51** may also attempt to identify an application-layer header of the packet and use the application-layer header to identify the application associated with the communication session. Application identification module **51** may use any known or future technique to attempt to determine the application associated with the communication session. When application identification module **51** is able to determine the application associated with the communication session ("YES" branch of **126**), application identification module **51** forwards the packet (**128**). In one embodiment, application identification module **51** may send the packet to protocol decoders **30** to identify the protocol in use for the communication session. In another embodiment, application identification module **51** may send the packet to attack detection module **52** to inspect the packet and to determine whether the packet is associated with a network attack. In yet another embodiment, IDS **20** may simply forward the packet towards its destination in the network." (Burns, 12:7-23.)

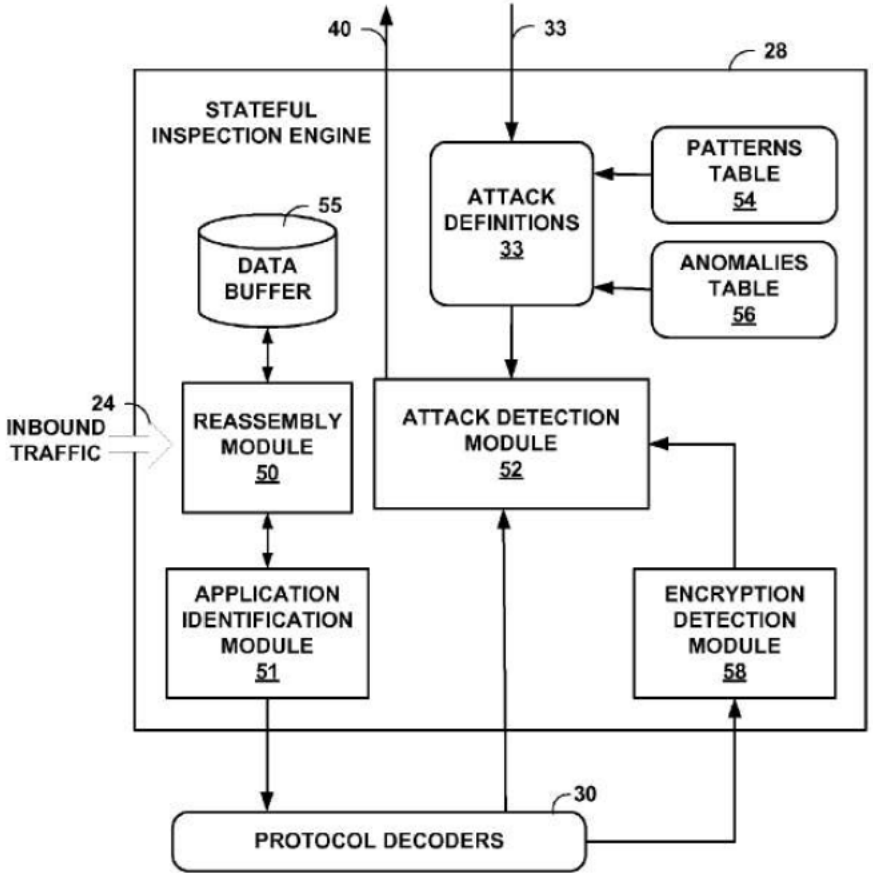
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	"When application identification module **51** is unable to determine the application associated with the communication, however, ("NO" branch of **126**), application identification module **51** sends the packet to encryption detection module **58**. Encryption detection module **58** inspects the packet when application detection module **51** cannot identify an application associated with the communication session and when protocol decoders **30** cannot identify a protocol associated with the communication session. Encryption detection module **58** attempts to determine whether the packet is fully encrypted (**128**). Encryption detection module **58** determines a randomness value for the TCP/IP payload of the packet and compares the randomness value to a randomness threshold. Where the randomness value exceeds the randomness threshold, encryption detection module **58** determines that the packet is fully encrypted. When encryption detection module **58** determines that the packet is not fully encrypted ("NO" branch of **128**), encryption detection module **58** may forward the packet. For example, in one embodiment, IDS **20** may simply forward the packet toward its network destination. In another embodiment, encryption detection module **58** may send the packet to attack detection module **52** to determine whether the packet represents a network attack. When encryption detection module determines that the packet is fully encrypted, however, ("YES" branch of **128**), encryption detection module **58** may determine whether a key exchange has already been identified for the communication session (**130**). As discussed above, protocol decoders **30** may have determined that one or more previously exchanged packet represented a key exchange and logged the fact that a non-hidden (i.e., readily detectable) key exchange has occurred for the communication session. If this has taken place earlier in the communication session, ("YES" branch of **130**), in this embodiment encryption detection module **58** determines that the current packet, while potentially fully encrypted, is not associated with an invalid or otherwise unwanted application. Therefore, encryption detection module **58** may forward the packet. For example, in one embodiment, IDS **20** may simply forward the packet toward its network

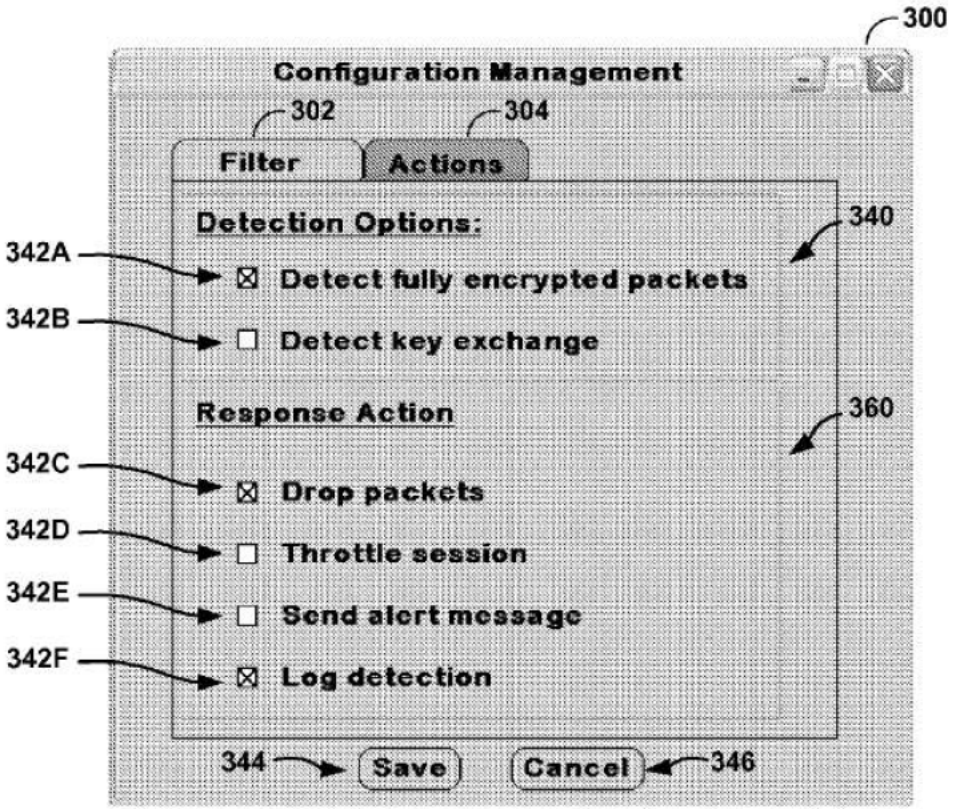
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	destination. In another embodiment, encryption detection module **58** may send the packet to attack detection module **52** to determine whether the packet represents a network attack. However, if there has been no earlier key exchange ("NO" branch of **130**), encryption detection module may output an alert and take a programmed response (**132**). The response to take may be in accordance with configuration data received from, e.g., administrator **42**, at step (**120**)." (Burns, 13:9-57.) *See* also Burns, 17:58-20:67. *See* also claim 4.
17[d]. a communication module, to allow communication between the remote server and the at least one user's device;	Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Burns discloses that network traffic flows into and out of the IDS. The IDS may comprise a network interface for allowing the network traffic flows

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	 FIG. 2 (Burns, Fig. 2.) "FIG. 1 is a block diagram illustrating an exemplary system **2** in which enterprise computer network **4** includes intrusion detection system (IDS) **10** that intercepts packet flows associated with communication sessions and processes the packet flows to filter unwanted or malicious packet flows in accordance with the principles of the invention. In the example embodiment of FIG. 1, IDS **10** is a single network device. Network **4** also includes a private enterprise computing network **5** that is coupled to public network **6**, such

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	as the Internet. Public network **6** may include, for example, one or more client computing devices. Firewall **9** protects enterprise network **5** and, in particular, internal computing nodes **8**A-**8**N. Computing nodes **8**A-**8**N (computing nodes **8**) represent any private computing device within enterprise network **5**, including workstations, file servers, print servers, database servers, printers and other devices. In one embodiment, IDS **10** attempts to identify applications and protocols for each communication session between computing nodes **8** and other computing devices in public network **6**. Exemplary techniques for identifying specific applications and protocols are described in greater detail in U.S. patent application Ser. No. 11/835,923, Burns et al., "Identifying Applications for Intrusion Detection Systems," filed Aug. 8, 2007, assigned to the assignee of the present application, which is hereby incorporated by reference in its entirety." (Burns, 4:20-45.) *See* also Burns, 6:61-63.
17[e]. and a processor, coupled to a response database and to the privacy preference database, wherein the processor is configured to instruct the remote server to determine at least one data type for sharing of data packet that is compatible with the list of allowed patterns of data packets for sharing, and wherein the at least one data type is determined in accordance with characteristics of the communication data packet, and wherein the content of the at least one data packet is not read by the remote server for continued operation by the	Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Burns comprises a processor coupled to memories for storing the response actions and privacy preferences. The processor is configured to instruct the IDS to perform packet processing operations in real time (*e.g.*, the IDS "transparently monitors… and forwards"). Burns' IDS analyzes a received packet to identify a corresponding application or protocol and data patterns, such as repeated data and encryption key exchanges. The packets are not read because they are not decrypted.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
user's device in real time during communication between the remote server and the user's device.	 **FIG. 1** (Burns, Fig. 1.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	 FIG. 3 (Burns, Fig. 3.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	 FIG. 7B (Burns, Fig. 7B.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	"As one example, the Back Orifice protocol depends upon a random number generator that is seeded, i.e. initialized, with a secret key. Both parties to a Back Orifice communication session know the secret key in advance of the communication session, and both parties have the same random number generator. By seeding the random number generator with the same secret key, both parties are able to encrypt data without exchanging a key during the communication session. In this way, the parties seek to avoid detection of any key exchange and session establishment. Moreover, parties to the Back Orifice protocol encrypt all data in the packet, including the Back Orifice application-layer header. Back Orifice uses a 17-byte application-layer header that is encrypted, in addition to the payload, by one party and decrypted by the other party to a communication session." (Burns, 1:56-2:3.) "In one embodiment, a method includes receiving a network packet, using an application-layer header of the packet, determining whether the packet is associated with an identifiable network application, when the packet is not determined to be associated with an identifiable network application, determining whether data in the packet is encrypted by calculating a randomness value of the packet from a payload of the packet that includes the application-layer header and an application-layer payload and determining that the packet is encrypted when the randomness value exceeds a randomness threshold, and, when the data in the packet is determined to be encrypted, executing a programmed response." (Burns, 3:6-17.) "In another embodiment, a computer-readable medium stores instructions. The computer-readable medium may be a computer-readable storage medium. The instructions cause a programmable processor to receive a network packet, determine whether the packet is associated with a network application using an application-layer header of the packet, determine whether data in the packet is encrypted, when the packet is not determined to be associated with an identifiable network application, by calculating a randomness value of the packet from a payload of the packet that includes the application-layer header and an application-layer payload and determining that the packet is encrypted when the

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	randomness value exceeds a randomness threshold, and, when the data in the packet is determined to be encrypted, execute a response, and, when the data in the packet is determined not to be encrypted, forwarding the packet." (Burns, 3:34-49.) "FIG. 1 is a block diagram illustrating an exemplary system **2** in which enterprise computer network **4** includes intrusion detection system (IDS) **10** that intercepts packet flows associated with communication sessions and processes the packet flows to filter unwanted or malicious packet flows in accordance with the principles of the invention. In the example embodiment of FIG. 1, IDS **10** is a single network device. Network **4** also includes a private enterprise computing network **5** that is coupled to public network **6**, such as the Internet. Public network **6** may include, for example, one or more client computing devices. Firewall **9** protects enterprise network **5** and, in particular, internal computing nodes **8**A-**8**N. Computing nodes **8**A-**8**N (computing nodes **8**) represent any private computing device within enterprise network **5**, including workstations, file servers, print servers, database servers, printers and other devices. In one embodiment, IDS **10** attempts to identify applications and protocols for each communication session between computing nodes **8** and other computing devices in public network **6**. Exemplary techniques for identifying specific applications and protocols are described in greater detail in U.S. patent application Ser. No. 11/835,923, Burns et al., "Identifying Applications for Intrusion Detection Systems," filed Aug. 8, 2007, assigned to the assignee of the present application, which is hereby incorporated by reference in its entirety." (Burns, 4:20-45.) "Where IDS **10** is able to identify the application using a particular communication session, IDS **10** may either permit or prevent the communication session from continuing. For example, a system administrator may configure IDS **10** to explicitly allow all identifiable applications, allow all applications except for a specified list of identifiable applications, or prevent all communications except for communications from a specified list of permitted software applications. In cases where IDS **10** is unable to identify the

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	application, IDS **10** may further inspect characteristics of the data from the communication session to determine whether the data is encrypted. As one example, IDS **10** may identify an encrypted communication session by determining whether the data in the communication session exceeds a randomness threshold. A system administrator may configure IDS **10** to establish or modify the randomness threshold. When the data from the communication session exceeds the randomness threshold, IDS **10** determines that the data is encrypted. Upon determining that the packet flows of the communication session are encrypted, IDS **10** applies the principles described herein to determine whether the communication session is associated with an allowed application or a typically unwanted application. For example, IDS **10** may then determine whether a proper key exchange was initially detected for the communication session and/or whether the communication session contains fully encrypted packets or whether only application-layer data within the packets is encrypted." (Burns, 5:27-53.) "IDS **10** may be configurable in a variety of ways. For example, in one embodiment, IDS **10** may permit a communication session that is unencrypted to continue, despite being unable to identify a particular application that is using the communication session. In one embodiment, where IDS **10** determines that a communication session, for which an application cannot be identified, is encrypted, IDS **10** may terminate the communication session. IDS **10** may take other actions as well, such as dropping the packet, blocking future sessions from a computing device that initiated the communication session, or other actions. In one embodiment, where IDS **10** determines that a communication session, for which an application cannot be identified, is encrypted, IDS **10** may further determine whether a key exchange associated with the communication session can be identified; where a key exchange has been identified, IDS **10** may permit the communication session and where a key exchange has not been identified, IDS **10** may terminate the communication session. In one embodiment, IDS **10** may be configured to take any or all of the above actions, in accordance with a user configuration that a user, such as a system administrator, may configure." (Burns, 5:54-6:8.)

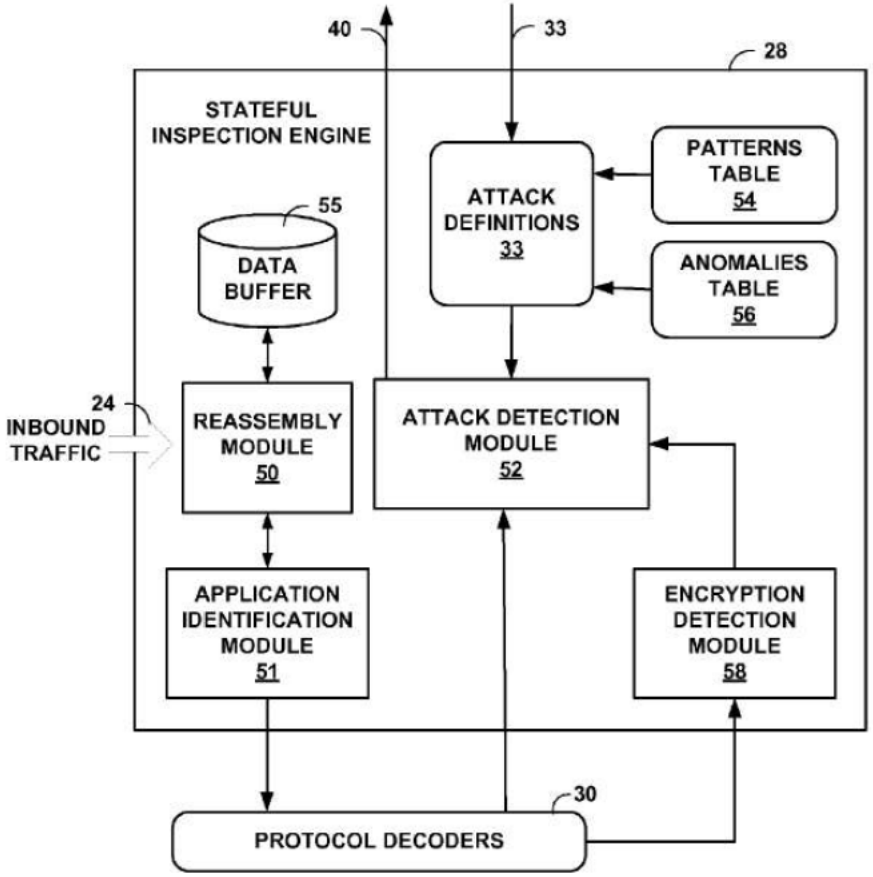
U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	"FIG. 2 is a block diagram illustrating an example embodiment of an IDS **20**. In the illustrated example, IDS **20** includes a forwarding plane **22** that transparently monitors inbound network traffic **24** and forwards the network traffic as outbound network traffic **26**. In the example illustrated by FIG. 2, forwarding plane **22** includes flow analysis module **25**, stateful inspection engine **28**, protocol decoders **30**, forwarding component **31** and security management module **44**." (Burns, 6:29-36.) "Application identification module **51** attempts to identify an application associated with each of the intercepted communication sessions. In one embodiment, when stateful inspection engine **28** receives a packet as part of a packet flow, reassembly module **50** buffers the packet in data buffer **55**. Reassembly module **50** attempts to reconstruct application layer data from the packets in data buffer **55**. Application identification module **51** then attempts to identify the application associated with the packets in accordance with this reconstructed data. In other embodiments, application identification module **51** may use other techniques to attempt to identify the application associated with the communication session." (Burns, 8:11-23.) "In one embodiment, one of protocol decoders **30** corresponds to a default protocol decoder. When application identification module **51** is unable to determine an application corresponding to a communication session, application identification module **51** sends data from that communication session to the default protocol decoder. The default protocol decoder attempts to identify various elements of the communication session that make the communication session appear to conform to a protocol. For example, the default protocol decoder attempts to analyze the bi-directional communications for the communication session to identify a key exchange for cryptography. As another example, the default protocol decoder attempts to identify regularly occurring patterns that correspond to protocol headers in the packets. Other embodiments may utilize other detection algorithms to determine if the communication session is following a protocol, even if the protocol is unknown." (Burns, 8:51-67.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	"Application identification module **51** attempts to identify an application associated with the received packet (**76**). For example, application identification module **51** may inspect the packet header to determine the destination port of the packet. In some cases, the destination port is associated with the protocol or application being used for the communication session. For example, port **80** is typically associated with HTTP traffic, while port **22** is typically associated with SSH traffic. Application identification module **51** may also inspect an application-layer header, within a TCP/IP payload of the packet, to identify the application associated with the packet. Application identification module **51** may use any known or future technique to attempt to identify the application associated with the communication session." (Burns, 10:32-45.) "When encryption detection module **58** determines that the packet is not fully encrypted ("NO" branch of **80**), either because the packet is not encrypted at all or because the packet is part of a valid encryption scheme, for example an encryption scheme that does not encrypt the application-layer header, encryption detection module **58** may forward the packet (**78**). In one embodiment, IDS **20** may simply forward the packet toward its network destination. In another embodiment, encryption detection module **58** may send the packet to attack detection module **52** to determine whether the packet represents a network attack." (Burns, 11:19-29.) "As an example, the communication session may be in accordance with the secure socket layer (SSL) protocol. Although application identification module **51** may not be able to determine the specific application associated with the communication, protocol decoders **30** may determine that the communication session is using the SSL protocol. The SSL protocol uses several messages to indicate the use of the SSL protocol after the TCP/IP three-way handshake. A client using the SSL protocol will first send a ClientHello message to a server that includes the highest transportation layer security (TLS) protocol version the client supports, a random number, a cipher suite that the client supports, and compression methods supported by the client. The server will respond with a ServerHello message including a selected TLS protocol version, a random number, a cipher suite, and a

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	compression method, each of which the server may select from the client's ClientHello message. Other standard messages are also exchanged between the client and the server to initiate an SSL session. Protocol decoders **30** may include an SSL decoder that may identify the protocol in use for the communication session as SSL. Moreover, protocol decoders **30** may recognize a packet that includes key exchange information that is exchanged as part of the SSL protocol. For example, protocol decoders **30** may recognize a ClientMasterKey message sent by the client to the server. Protocol decoders **30** may also recognize a ServerVerify message sent by the server to the client. Protocol decoders **30** may recognize any or all of these messages, for example, by inspecting the application layer packet header." (Burns, 12:47-13:8.) "Another exemplary method for determining whether a packet is encrypted includes identifying repeating patterns of varying lengths, e.g., one byte, two bytes, four bytes, or other within a packet. That is, encryption detection module **58** may determine that a packet is not random when encryption detection module **58** determines that there exist repeating patterns within the packet." (Burns, 14:32-38.) "When administrator **42** selects check box **310**A and selects save button **344**, administrator user interface **300** instructs security management module **44** to use a filter to determine whether data in a packet has repeating values. Repeated data in a packet may be an indication that the packet is not random, or that the packet is associated with a protocol that should not be blocked. In one embodiment, encryption detection module **58** compares data in the packet to other data in the packet to find a repeating sequence. In one embodiment, encryption detection module **58** stores a sequence of data from each packet in a communication session and determines whether there exists a repeated value in each packet of the communication session; in this case, the repeated value may indicate a standard header of an unknown protocol. In any case, IDS **20** may utilize a repeating values filter." (Burns, 18:30-44.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	"When administrator **42** selects "Detect key exchange" check box **342**B, and then selects save button **344**, administrator user interface **300** sends a corresponding message to security management module **44**. In particular, when "Detect key exchange" check box **342**A is selected, administrator user interface **300** sends a message to security management module **44** to detect key exchanges for monitored communication sessions, or to operate in a detect key exchange mode. Accordingly, security management module **44** instructs stateful inspection engine **28** to analyze incoming packets to determine whether those packets represent a key exchange for their corresponding communication session. Stateful inspection engine **28** then begins to determine whether incoming packets represent a key exchange as part of the communication session. In one embodiment, stateful inspection engine **28** may only determine whether a first set of packets of a communication session represents a key exchange. In one embodiment, stateful inspection engine **28** may only determine whether packets represent a key exchange when those packets correspond to a communication session for which an application and/or a protocol cannot be identified." (Burns, 19:22-42.) "Actions tab **304** of administrator user interface **300** also presents response actions panel **360** to a user, such as administrator **42**. Response actions panel **360** presents four options to administrator **42**: "Drop packets", "Throttle session", "Send alert message", and "Log detection." Administrator **42** may select these options using check boxes **342**C-**342**F, respectively. Other embodiments may include other responses, such as blocking future communication sessions from a computing device that sent a fully encrypted packet, or blocking the communication session in which a fully encrypted packet was identified." (Burns, 19:61-20:4.) "Methods described herein may be performed in hardware, software, or any combination thereof within a network device. For example, methods described herein may be performed by an application specific integrated circuit (ASIC) or a general-purpose processor. Methods described herein may also be embodied in a computer readable medium containing instructions. Instructions embedded in a computer readable medium

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	may cause a programmable processor, or other processor, to perform the method, e.g. when the instructions are executed. A computer readable medium may be a computer readable storage medium. Computer readable storage media may include, for example, random access memory (RAM), read only memory (ROM), programmable read only memory (PROM), erasable programmable read only memory (EPROM), electronically erasable programmable read only memory (EEPROM), flash memory, a hard disk, a CD-ROM, a floppy disk, a cassette, magnetic media, optical media, or other computer readable media." (Burns, 21:14-31.) *See* also claim 1 and claim elements 17[b] and 17[c].
18. The system of claim 17, further comprising a communication data pattern database, coupled to the processor and comprising at least one data pattern corresponding to sharing of at least one data packet from the user's device, wherein the processor is configured to modify data packet corresponding to requests for retrieval of data packets and communication data types that are not compatible with communication data patterns from a communication data pattern database.	Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Burns discloses that security management module, which is coupled to the processor, can store information in "a database, a text file, or any appropriate data structure." The data pattern can be repeated data pattern, encryption key exchanges, or patterns and anomalies. Burns' IDS is configurable to take various actions in handling a packet based in part on the packet's application or protocol ("data packets corresponding to… communication data types"). Exemplary potential actions include (1) dropping (blocking) the packet, (2) closing the related communication session, or (3) throttling (limiting bandwidth consumption of) the related communication session.

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	 FIG. 3 (Burns, Fig. 3.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	"Flow analysis module **25** maintains data within flow table **35** that describes each active packet flow present within the network traffic. Flow table **35** specifies network elements associated with each active packet flow, i.e., low-level information such as source and destination devices and ports associated with the packet flow. In addition, flow table **35** may identify pairs of packet flows that collectively form a single communication session between a client and server. For example, flow table **35** may designate communication session as pairs of packet flows in opposite directions for flows sharing at least some common network addresses, ports and protocol." (Burns, 7:3-15.) "In one embodiment, to determine whether a packet flow is associated with an identifiable application, stateful inspection engine **28** buffers a copy of the packet flow and reassembles the buffered packet flow to form application-layer communications. For example, stateful inspection engine **28** may reconstruct TCP segments into application-layer communications, which represent protocol-specific messages. Where stateful inspection engine **28** is unable to determine an identity of an application associated with a packet flow, stateful inspection engine **28** performs additional analysis on the packet flow. In one embodiment, for example, stateful inspection engine **28** attempts to determine whether the packet flow is encrypted. When stateful inspection engine **28** determines that the packet flow is encrypted, stateful inspection engine **28** further determines whether characteristics of the communication session indicates that the packet flow is associated with an unwanted software application. Stateful inspection engine **28** may, in one embodiment, determine whether a key exchange has occurred for the communication session when stateful inspection engine **28** identifies an encrypted packet of the communication session. In this embodiment, when stateful inspection engine **28** determines that a key exchange has occurred, stateful inspection engine **28** permits the communication session to continue, but when stateful inspection engine **28** determines that a key exchange has not occurred, stateful inspection engine **28** takes a programmed response, e.g., to block the communication session, block

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	future communication sessions from a peer of the communication session, output an alert, or other actions. In the event a security risk is detected, stateful inspection engine **28** outputs an alert **40** to security management module **44** for logging and further analysis. In addition, stateful inspection engine **28** may take additional actions, such as dropping the packets associated with the communication session, automatically closing the communication session, or other actions. If no security risk is detected for a given application-layer communication session, stateful inspection engine **28** instructs forwarding component **31** to forward the packet flows between the peers. Forwarding component **31** may, for example, maintain a routing table that stores routes in accordance with a topology of enterprise network **5** for use in forwarding the packet flows." (Burns, 7:16-57.) "Stateful inspection engine **28** stores attack definitions **33** received from security management module **44**. Attack definitions **33** may be stored, for example, in a computer-readable medium, such as random access memory (RAM). Each of attack definitions **33** specifies a combination of one or more patterns specified within patterns table **54** and one or more protocol-specific anomalies specified within anomalies table **56**." (Burns, 8:3-10.) "Encryption detection module **58** attempts to determine whether the packets of the communication session are fully encrypted, i.e., that the application-layer header as well as the application-layer payload are encrypted. For example, encryption detection module **58** may extract the payload from a TCP/IP packet, which frequently includes an application-layer packet header (e.g., HTTP header) and application-layer payload. Encryption detection module **58** then analyzes characteristics of the TCP/IP payload to determine whether the payload of the TCP/IP packet is fully encrypted. That is, encryption detection module **58** examines the entire payload of a TCP/IP packet, including a region that would typically be an application-layer header, and compare

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	characteristics of the TCP/IP payload to a randomness threshold to determine if the TCP/IP payload that would normally correspond to the application-layer header as well as the application-layer payload are encrypted." (Burns, 9:9-25.) "In one embodiment, administrator **42** (FIG. 2) may configure a randomness threshold of IDS **20** (FIG. 2). In another embodiment, IDS **20** utilizes a randomness threshold specified by a randomness detection algorithm, such as the Wald-Wolfowitz runs test algorithm. Encryption detection module **58** determines a randomness value for a packet of the communication session being examined, i.e., the TCP/IP payload of a packet of the communication session. When the randomness value of a particular TCP/IP payload of the communication session exceeds the randomness threshold, encryption detection module **58** determines that the communication session is encrypted. Accordingly, encryption detection module **58** sends a message to attack detection module **52** that the communication session is being conducted by an undesirable application. Attack detection module **52**, therefore, may perform one or more of a variety of actions. For example, attack detection module **52** may terminate the communication session, log the communication session, generate a message for, e.g., administrator **42**, or take other actions." (Burns, 9:26-44.) "Initially, security management module **44** receives configuration information from administrator **42** and, in response, configures IDS **20** to monitor a network or portions thereof (subnets) of interest (**70**). During this process, configuration manager **44** presents a user interface by which administrator **42** specifies patterns or other attack definitions **33**. For example, administrator **42** may configure IDS **20** to block all packets deemed to be fully encrypted and to log information about the communication session associated with the blocked packets. Administrator **42** may also configure IDS **20** to throttle down (i.e., bandwidth limit) the communication session associated with the packets to minimize the bandwidth used by the communication session. Administrator **42** may also configure IDS **20** to block future communication sessions from either or both of the communicating devices. IDS **20** may additionally

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	include a default configuration that specifies actions IDS **20** should take when IDS **20** detects a fully encrypted packet. IDS **20** may, for example, be programmed to let such packets pass by default, or to block all fully encrypted packets by default." (Burns, 9:64-10:17.) "When application identification module **51** is able to determine the application associated with the communication session ("YES" branch of **76**), application identification module **51** forwards the packet (**78**). In one embodiment, application identification module **51** may send the packet to protocol decoders **30** to identify the protocol in use for the communication session. In another embodiment, application identification module **51** sends the packet to attack detection module **52** to inspect the packet and to determine whether the packet is associated with a network attack. In yet another embodiment, IDS **20** simply forwards the packet towards its destination in the network." (Burns, 10:46-57.) "As an example, the communication session may be in accordance with the secure socket layer (SSL) protocol. Although application identification module **51** may not be able to determine the specific application associated with the communication, protocol decoders **30** may determine that the communication session is using the SSL protocol. The SSL protocol uses several messages to indicate the use of the SSL protocol after the TCP/IP three-way handshake. A client using the SSL protocol will first send a ClientHello message to a server that includes the highest transportation layer security (TLS) protocol version the client supports, a random number, a cipher suite that the client supports, and compression methods supported by the client. The server will respond with a ServerHello message including a selected TLS protocol version, a random number, a cipher suite, and a compression method, each of which the server may select from the client's ClientHello message. Other standard messages are also exchanged between the client and the server to initiate an SSL session. Protocol decoders **30** may include an SSL decoder that may identify the protocol in use for the communication session as SSL. Moreover, protocol decoders **30** may recognize a packet that includes key exchange information that is

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	exchanged as part of the SSL protocol. For example, protocol decoders **30** may recognize a ClientMasterKey message sent by the client to the server. Protocol decoders **30** may also recognize a ServerVerify message sent by the server to the client. Protocol decoders **30** may recognize any or all of these messages, for example, by inspecting the application layer packet header." (Burns, 12:47-13:8.) "Another exemplary method for determining whether a packet is encrypted includes identifying repeating patterns of varying lengths, e.g., one byte, two bytes, four bytes, or other within a packet. That is, encryption detection module **58** may determine that a packet is not random when encryption detection module **58** determines that there exist repeating patterns within the packet." (Burns, 14:32-38.) "In one embodiment, an administrator, such as administrator **42**, may add, edit, or remove filters of encryption detection module **58**. A filter instructs encryption detection module **58** to determine that a packet is random or not random based on certain filtering characteristics. The filtering characteristics may include, for example, the number of bytes in the packet, examining bytes in the packet for a certain pattern, determining whether a minimum number of byte values above or below the mean occur, or other characteristics of a random or nonrandom packet. The following pseudocode shows one exemplary embodiment of a function performed by encryption detection module **58** on a packet to determine whether the packet is random or not" (Burns, 16:43-56.) "When administrator **42** selects check box **310**A and selects save button **344**, administrator user interface **300** instructs security management module **44** to use a filter to determine whether data in a packet has repeating values. Repeated data in a packet may be an indication that the packet is not random, or that the packet is associated with a protocol that should not be blocked. In one embodiment, encryption detection module **58** compares data in the packet to other data in the packet to find a repeating sequence. In one embodiment, encryption detection module **58** stores a sequence of data from each packet in a communication session and determines whether there exists a repeated value in each

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	packet of the communication session; in this case, the repeated value may indicate a standard header of an unknown protocol. In any case, IDS **20** may utilize a repeating values filter." (Burns, 18:30-44.) "When administrator **42** selects "Detect key exchange" check box **342**B, and then selects save button **344**, administrator user interface **300** sends a corresponding message to security management module **44**. In particular, when "Detect key exchange" check box **342**A is selected, administrator user interface **300** sends a message to security management module **44** to detect key exchanges for monitored communication sessions, or to operate in a detect key exchange mode. Accordingly, security management module **44** instructs stateful inspection engine **28** to analyze incoming packets to determine whether those packets represent a key exchange for their corresponding communication session. Stateful inspection engine **28** then begins to determine whether incoming packets represent a key exchange as part of the communication session. In one embodiment, stateful inspection engine **28** may only determine whether a first set of packets of a communication session represents a key exchange. In one embodiment, stateful inspection engine **28** may only determine whether packets represent a key exchange when those packets correspond to a communication session for which an application and/or a protocol cannot be identified." (Burns, 19:22-42.) "When administrator **42** selects log detection check box **342**F, and then selects save button **344**, administrator user interface **300** sends a corresponding message to security management module **44**. In particular, when "Log detection" check box **342**A is selected, administrator user interface **300** sends a message to security management module **44** to make a log of the event that was identified. For example, security management module **44** records the source port, destination port, source IP address, destination IP address, time of discovery, packet size, a copy of the packet, other actions taken in response to the detection, or any other information that administrator **42** may find useful. Security management module **44** may create this log in, e.g., a database, a text file, or any appropriate data structure." (Burns, 20:54-67.)

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	See also claim elements 17[c] and 17[e].
19. The system of claim 18, wherein data packets from the user's device are selected from the group consisting of user's device files, user's device characteristics, user's device indirect attributes, user's device sensor data, user's device browser data, user's device form data, user's device dynamic memory and user's device static memory.	Burns discloses this element and/or renders it obvious either alone or in combination with other references and/or the knowledge of one of ordinary skill in the art. For example, Burns discloses analyzing HTTP traffic, which includes packets corresponding to the "user's browser data." "Encryption detection module **58** attempts to determine whether the packets of the communication session are fully encrypted, i.e., that the application-layer header as well as the application-layer payload are encrypted. For example, encryption detection module **58** may extract the payload from a TCP/IP packet, which frequently includes an application-layer packet header (e.g., HTTP header) and application-layer payload. Encryption detection module **58** then analyzes characteristics of the TCP/IP payload to determine whether the payload of the TCP/IP packet is fully encrypted. That is, encryption detection module **58** examines the entire payload of a TCP/IP packet, including a region that would typically be an application-layer header, and compare characteristics of the TCP/IP payload to a randomness threshold to determine if the TCP/IP payload that would normally correspond to the application-layer header as well as the application-layer payload are encrypted." (Burns, 9:9-25.) "Application identification module **51** attempts to identify an application associated with the received packet (**76**). For example, application identification module **51** may inspect the packet header to determine the destination port of the packet. In some cases, the destination port is associated with the protocol or application being used for the communication session. For example, port **80** is typically associated with HTTP traffic, while port **22** is typically associated with SSH traffic. Application identification module **51** may also inspect an application-layer header, within a TCP/IP payload of the packet, to identify the application associated with the packet. Application identification

U.S. Patent No. 11,106,824	
Claim	**Relevant Disclosure of Burns**
	module **51** may use any known or future technique to attempt to identify the application associated with the communication session." (Burns, 10:32-45.)
20. The system of claim 18, wherein at least the communication module and the processor are embedded on a single hardware component.	*See* claim elements 17[d] and 17[e]. *See* also Burns, 21:14.