



(51) International Patent Classification:

H04L 9/30 (2006.01) H04L 12/70 (2013.01)

(21) International Application Number:

PCT/US2013/029799

(22) International Filing Date:

8 March 2013 (08.03.2013)

(25) Filing Language:

English

(26) Publication Language:

English

(71) Applicant: HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P. [US/US]; 11445 Compaq Center Drive W., Houston, Texas 77070 (US).

(72) Inventors: WAKUMOTO, Shaun; 8000 Foothills Blvd., Roseville, California 95747 (US). MILLS, Craig, Joseph; 8000 Foothills Blvd., Roseville, California 95747 (US). SYED MOHAMED, Parvez; Address2, 1210 Darling Way, District, Folsom, California 95630 (US).

(74) Agents: PINKSTON, Caroline, S. et al.; Hewlett-Packard Comapny, Intellectual Property Administration, 3404 East Harmony Road Mail Stop 35, Fort Collins, CO 80528 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,

BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

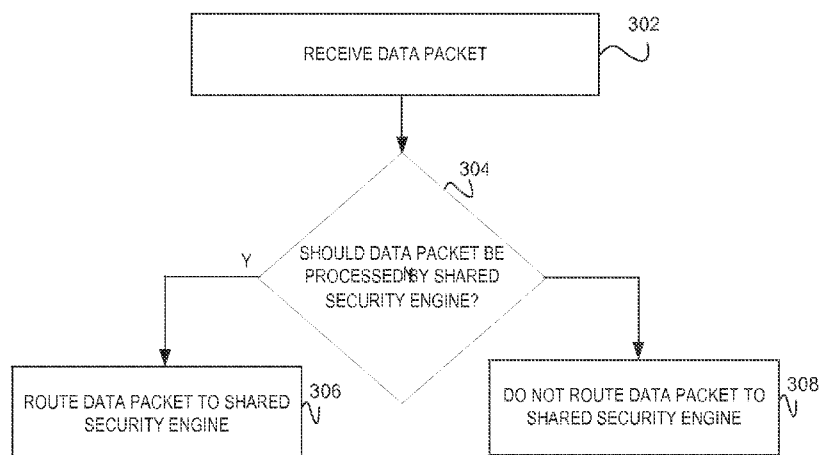
Declarations under Rule 4.17:

- as to the identity of the inventor (Rule 4.17(i))
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

Published:

- with international search report (Art. 21(3))

(54) Title: ROUTING A DATA PACKET TO A SHARED SECURITY ENGINE



(57) Abstract: Examples disclose a system comprising an integrated circuit to determine whether a data packet should be processed by a shared security engine associated with a secure link. Additionally, the examples disclose a first media access control (MAC), as associated with the shared security engine, to receive the data packet for transmission on the secure link based on the determination the data packet should be processed by the shared security engine.



ROUTING A DATA PACKET TO A SHARED SECURITY ENGINE

BACKGROUND

[0001] Computer networks, such as those compliant with IEEE 802® standards, may be deployed in many type of environments, such as a home and/or business. These protocols may establish the security settings to prevent unauthorized access to data traffic. The security settings may include maintaining confidentiality of transmitted data and/or preventing unauthorized access to the transmitted data.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] In the accompanying drawings, like numerals refer to like components or blocks. The following detailed description references the drawings, wherein:

[0003] FIG. 1 is a block diagram of an example switch including an integrated circuit to receive a data packet and to determine whether the data packet should be processed by the shared security engine;

[0004] FIG. 2A is a diagram of an example switch including an integrated circuit to identify whether a data packet is encrypted upon ingress to the switch and route the encrypted data packet to a shared security engine for decryption;

[0005] FIG. 2B is a diagram of an example switch including an integrated circuit to identify a second shared engine with an available secure association based on a determination a shared security engine is operating with a maximum number of secure associations;

[0006] FIG. 3 is a flowchart of an example method to receive a data packet, determine whether the data packet should be processed by a shared security engine, and route the data packet to the shared security engine;

[0007] FIG. 4 is a flowchart of an example method to receive a data packet and depending whether the data packet should be processed by the shared security engine, the data packet is either routed for transmission on a non-secure link or to the shared security engine for transmission on a secure link;

[0008] FIG. 5 is a flowchart of an example method to receive a data packet, route the data packet to either the shared security engine for transmission on a secure link or for transmission on a non-

secure link, the example method further determines whether the shared security engine is operation with a maximum number of associations and routes the data packet to a second shared security engine; and

[0009] FIG. 6 is a block diagram of an example computing device with a processor to execute instructions in a machine-readable storage medium for processing a header of a data packet and routing the data packet accordingly.

DETAILED DESCRIPTION

[0010] Security settings may be implemented by using a dedicated security engine per communication channel, to maintain data confidentiality and integrity. However, this may increase system costs and complexity. Although hardware components may be altered to provide security within the networking system, such alterations may require modifications of various other hardware components. Further, it may be burdensome to the networking system to implement security for each transmission and receipt of client traffic, as this may cause a time latency. For example, the client traffic may include data packets that are processed serially in a queue, exacerbating the time latency associated with processing the data packets in series. Such dedicated security implementations have various other limitations associated with increased costs, space, and design complexity of a networking system.

[0011] To address these issues, examples disclosed herein provide a more cost-efficient and less complex design within the networking system by utilizing a shared security engine among multiple links or communication channels. The examples disclosed herein provide a process to determine whether a data packet should be processed by the shared security engine associated with a secure link. Determining whether the data packet should be processed for security reasons, prevents the time latency as the data packets may be selectively chosen for encryption and/or decryption by the shared security engine.

[0012] Additionally, the examples provide the process for an integrated circuit to route the data packet to the shared security engine upon the determination the packet should be processed by the shared security engine. Routing the data packet to the shared security engine, enables the process to provide decryption at the ingress of a data packet into the networking system and encryption upon egress of the data packet onto the secure link. Providing this encryption and

decryption further enables the networking system to use fewer hardware components while maintaining data confidentiality and integrity. This reduces the overall number of security engines per networking system, as the system may utilize the shared security engine among multiple communication channels or links.

[0013] In another example, the process determines whether the shared security engine is operating with a maximum number of secure associations and routes the data packet to a shared security engine with an available secure association to provide security. This example provides an additional level of efficiency as the number of secure associations per security engine may be limited. Thus routing the client traffic to the security engine with the available secure association provides security to the client traffic which may be unsecured, otherwise.

[0014] In a further example, the process further determines whether the data packet may be encrypted upon ingress into the networking system and routing the data packet for decryption to a first media access control (MAC). The process receives the decrypted data packet from a second MAC. This process provides a cost-efficient process to enable security among data packets.

[0015] In summary, examples disclosed herein provide a shared security engine to maintain data confidentiality and integrity among multiple communication channels or links. The shared security engine reduces the number of security components in a networking system, while also providing data security. Additionally, the examples disclosed herein prevent time latency by enabling selection of data packets for encryption and/or decryption.

[0016] Referring now to the figures, FIG. 1 is a block diagram of an example switch 104 in a networking system. The networking system is a collection of networking devices and other hardware which may be interconnected by communication channels, such as a non-secure link 116 and a secure link 118. The networking system as depicted in FIG. 1 represents an Open Systems Interconnection (OSI) model for communicating with other networking devices. These networking systems may include wide area network (WAN), local area network (LAN), Ethernet, optic cable network, or other type of networking system. The switch 104 includes an integrated circuit 106 to receive a data packet 102 and module 108 to determine whether the data packet should be processed by a shared security engine 112. If the integrated circuit 106 determines the data packet 102 should

be processed by the shared security engine 112, the data packet is routed to a first media access control (MAC) 110. The integrated circuit 106 further includes a hashing engine 120 to choose a port for transmission on one of the links 116 or 118 to transmit the data packet 102. The switch 104 is a networking component that links network devices together. As such, the switch 104 may include multiple links (not illustrated) in addition to the non-secure link 116 and the secure link 118. The switch 104 may route client traffic (e.g., data packet(s) 102) to a data link layer or a network link layer of the OSI model. In implementations, the switch 104 may include a point to point connection with another switch through the links 116 and 118. In further implementations, the switch 104 is included as part of a blade server component. Implementations of the switch 104 include a multi-port network device, multi-layer switch, or other type of networking component capable of providing the physical connections through wired connections or wireless connections between networking devices.

[0017] The data packet 102 includes data and a header for the integrated circuit 106 to process at module 108. At module 108, a decision is determined of whether the data packet 102 should be routed to the shared security engine 112 for processing or to the hashing engine 120. The data packet 102 is considered a portion of aggregate data and metadata. The aggregate data may include client traffic, a data stream, and/or data frames. In one implementation, the integrated circuit 106 may receive the data packet 102 as it ingresses into the switch 104. In this implementation, the data packet 102 may be processed to determine whether it is encrypted and routed to the shared security engine 112 for decryption. In another implementation, the data packet 102 may be processed as it egresses from the switch 104. In this implementation, the data packet 102 may be processed to determine if the data packet 102 contains sensitive data, indicating the data packet 102 should be processed through the shared security engine 112 to provide protection. For example, the data packet 102 may include personal information, indicating this data should be encrypted prior to transmission on the secure link 118 to maintain confidentiality. The sensitive data may be identified using a header of the data packet 102 and as such may be identified by the application type of the data packet 102, the source of the data packet 102, the destination of the data packet 102, and/or whether the data packet 102 may have been received upon ingress or egress to the switch 104.

[0018] The integrated circuit 106 may receive the data packet 102 upon ingress or egress of the data packet 102 from the switch 104. Upon ingress, the integrated circuit 106 receives the data packet 102 for determining whether the data packet may be encrypted at module 108 and routing the encrypted data packet to the shared security engine 112 for decryption. Upon egress, the integrated circuit 106 receives the data packet 102 and determines whether the data packet 102 contains sensitive data that indicates it should be encrypted by the shared security engine 112. Once encrypting the data packet 102 by the shared security engine 112, the data packet 102 is transmitted on the secure link 118. In a further example, if the integrated circuit 106 determines the data within the data packet 102 is without sensitive data, the integrated circuit 106 may route the data packet 102 to the hashing engine 120 for transmission on the non-secure link 116. In one implementation, the integrated circuit 106 may communicate with a controller (not illustrated) to obtain a secure association for encryption and/or decryption of the data packet 102. Implementations of the integrated circuit 106 include an application-specific integrated circuit (ASIC), controller, microprocessor, microchip, chipset, electronic circuit, semiconductor, microcontroller, central processing unit (CPU), or other programmable device capable of receiving the data packet 102 and executing module 108.

[0019] The module 108 processes the header of the data packet 102 for the sensitive type of data which indicates to the integrated circuit 106, the data packet 102 should be processed for security reasons to maintain data confidentiality and integrity. Implementations of the module 108 include a set of instructions, instruction, process, operation, logic, algorithm, technique, logical function, firmware, and or software executable by the integrated circuit 106 to determine whether the data packet 102 contains sensitive data that may warrant processing through the shared security engine 112 to maintain data confidentiality and integrity.

[0020] The first media access control (MAC) 110 acts as an interface between the logical link layer and the network's physical layer (PHY). As such, the first MAC 110 is considered a sub-layer to the data link layer in the OSI model of the networking system. In this regard, the first MAC 110 emulates a full-duplex logical communication channel in the networking system.

[0021] The shared security engine 112 is a hardware component within the networking system which is configured to provide network security for the data packet 102. The shared engine 112 may be shared among several client devices using the same secure channel or multiple secure

channels (i.e., links 116 and 118). Sharing the engine 112 among multiple links 116 and 118 decreases the number of security components for encryption or decryption. The security techniques employed by the shared security engine 112 may include the exchange of cipher keys, symmetric keys, associated encryption/decryption, and authentication techniques to provide network security. In one implementation the shared security engine 112 provides network security by decrypting the data packet 102 upon ingress and encrypting the data packet 102 upon egress from the switch 104. In this implementation, shared security engine 112 may include multiple secure associations and as such, may include a secure association table. The table includes the multiple secure associations with the secure identifiers and the corresponding keys. The table may be used to identify the given corresponding key to encrypt or decrypt the data packet 102. This implementation is described in detail in later figures. In other implementations, the shared security engine 112 may be specific to a given MAC, such as the first MAC 110 or specific to multiple MACs. In another further implementation, the shared security engine 112 may include a media access control security (MACSec). The MACSec is a hardware component which defines connectionless data confidentiality and integrity among the first MAC 110 and/or other MACs. Although FIG. 1 illustrates the shared security engine 112 as independent of the integrated circuit 106, implementations should be limited to this illustration as the shared security engine 112 may be part of the first MAC 110, the integrated circuit 106, or as part of a network PHY.

[0022] The non-secure link 116 and the secure link 118 include physical ports on the switch 104 which and operate as communication channels or links between the switch 104 and another security-enabled networking device for data communications. The secure link 118 may be aggregated with the non-secure link 116 by combining these links in parallel to increase the throughput beyond what a single link may be capable of transmitting as symbolized by 114. For example, the link aggregation may include the logical formation of multiple links to form a single logical link between the switch 104 and another networking device. The aggregation of these links may also provide redundancy if either the secure link 118 or the non-secure link 116 may fail. The links are aggregated together at the second layer and third layers of open systems interconnection (OSI) model or combination thereof. These layers which implement the aggregation may include the data link layer and/or network layer. The secure link 118, secures the data packet 102 that was encrypted by the shared security engine 112; however the shared

security engine 112 rather than the secure link 118 performs the encryption. Specifically, the secure link 118 transmits this encrypted data packet to another networking device. In another implementation, the shared security engine 112 may decrypt the data packet 102. The decrypted data packet is routed back to back to the integrated circuit 106. The integrated circuit 106 then processes the decrypted packet. In this regard, the shared security engine 112 may appear transparent to the networking system which enables the shared security engine 112. The non-secure link 116 is considered the communication channel for the transmission of the data packet 102 without the security means. Specifically, the data packet 102 may not include sensitive data that would trigger the integrated circuit 106 to route the data packet to the first MAC 110 for encryption by the shared security engine 112. Additionally, although FIG. 1 illustrates two links 116 and 118, implementations should not be limited to this illustration as there may be multiple non-secure links and/or multiple secure links.

[0023] The hashing engine 120 is a data structure enabled to identify the port for use with the corresponding link 116 or 118 for transmission. If it is determined the data packet 102 should be processed by the shared security engine 112 for encryption, the hashing engine 120 may be bypassed as the encrypted data packet is transmitted on the secure link 118. In turn, if it is determined the data packet 102 is without sensitive data, the hashing engine 120 directs the data packet 102 to the particular port for transmission on the corresponding link 116 and/or 118.

[0024] FIG. 2A is a block diagram of an example switch 104 including an integrated circuit 106 to identify whether a data packet is encrypted upon ingress to the switch 104. The data packet 102, in accordance with FIG. 1, enters the switch 104 for the integrated circuit 106 to determine whether the data packet is encrypted. Upon determining the data packet is encrypted, the integrated circuit 106 routes the encrypted data packet to the first MAC 110 for decryption. The data packet further includes a secure association identifier for the shared security engine 112 to identify the corresponding key as in a table 212. The table 212 may be stored in memory component within the shared security engine 112 and in another memory component internal to the controller 214. The secure identifier is a value that enables the security engine 112 to identify the key for encryption and/or decryption as the shared security engine 112 may contain multiple keys for multiple communication channels. For example, the shared security engine

112 includes the table 212 for multiple secure associations and includes secure identifiers (i.e., SA1, SA2, and SA3), and the corresponding keys (KEY1, KEY2, KEY3). If a secure association has not yet been established by the shared security engine 112, the integrated circuit 106 communicates with the controller 214 to negotiate the secure association and obtain keys for encryption at the shared security engine 112. The table 212 may also be used by the controller 214 to track the number of secure associations available at each security engine 112 and 206 as in FIG. 2B. In this implementation, the controller 214 includes a table 212 internal to itself to track the secure associations. Although FIGS. 2A-2B illustrate the shared security engines 112 and 206 as separate from the integrated circuit 106 and/or PHY 204, implementations should not be limited as this was done for illustration purposes. For example, the shared security engines 112 and 206 may be included in the integrated circuit 106 or as part of the PHY 204.

[0025] In FIG. 2A, the MACS 110 and 208 are dedicated to receiving the data packet for encryption and/or decryption from the integrated circuit 106. The data packet may then be routed to a port for transmission on a link to establish a communication channel with another networking device, such as a switch. In this implementation, the MACS 110 and 208 receive communications from the integrated circuit 106 and as such, a third MAC (not pictured) routes the encrypted data packet on a selected port to establish a communication channel with other networking components. For example, the integrated circuit 106 may receive client traffic (e.g., data packet) from third MAC connected to an external port. The integrated circuit 106 will route this data packet to the first MAC 110 for decryption. Likewise, the client traffic (e.g., data packet) for transmission or egress of the data packet from the switch 104, may be received on the second MAC 208 and encrypted by the shared security engine 112. The integrated circuit 106 receives the encrypted packet and routes the encrypted packet out to the third MAC for transmission on the secure link. This reduces the number of individual security engines per networking system, as the system may utilize the shared security engine 112 among multiple communication channels for providing data security. The two MACS 110 and 208 are connected together through the shared security engine 112 and a physical interface (PHY) 204. The PHY 204 is the physical interface and includes the hardware to connect the MACS 110 and 208. The PHY 204 is the hardware mechanism that contains a transceiver which electrically interfaces to a physical link and is the common abbreviation for the physical layer of the OSI model. This

creates an internal loopback connection for the shared security engine 112 to decrypt the data packet upon ingress into the switch 104 or encrypt the data packet upon egress from the switch 104.

[0026] FIG. 2B is a block diagram of an example switch 104 including integrated circuit 106 to identify a second shared engine 206 with an available secure association. The second shared engine 206 is identified based on a determination the shared security engine 112 is operating with a maximum number of secure associations. The second MAC 208 associated with the second shared engine 206 receives the data packet for processing. The ports 202 serve as an interface between the switch 104 and other networking devices to receive client traffic (e.g., data packets). These ports 202 may be used to receive and transmit the data packets through the links 116 and 118.

[0027] FIG. 2B illustrates the situation when the maximum number of secure associations (i.e., pair of keys) are exhausted at a particular engine 112. Once the number of maximum secure associations has been reached by the shared security engine 112, the integrated circuit 210 redirects the client traffic from the first port as indicated by the arrow 210. There are a limited number of keys to use for a given system, as defined by a vendor for the shared security engine 112. Thus, the maximum number of secure associations is a pre-defined number of pairs of keys to ensure security for a given number of clients in the networking system. Rather than ignore client traffic which should be processed by the shared security engine 112, the integrated circuit 106 utilizes the available secure association and redirects the data packets accordingly.

[0028] For example, the first port 202 may be servicing four clients by providing security protection with these data packets upon ingress and egress from the first port 202. In this example, shared security engine 112 may include available secure associations for the four clients and may not have an available secure association to service an additional client. Thus, the controller 214 as in FIG. 2A may utilize the table 212 internal to itself to track the number of secure associations at each shared security engine 112 and 206. The controller 214 may track when the shared security engine 112 has the maximum number of secure associations and communicate with the integrated circuit 106 to direct the additional client to the second MAC 208 for processing by the second shared security engine 206. In a further implementation, the key pair may specific to a particular client and as such, the keys for encryption and decryption

may be obtained by the integrated circuit 106 from the controller 214 for placement in the second shared engine 206.

[0029] FIG. 3 is a flowchart of an example method to receive a data packet, determine whether the data packet should be processed by a shared security engine, and route the data packet to the shared security engine. In discussing FIG. 3, references may be made to the components in FIGS. 1-2B to provide contextual examples. Further, although FIG. 3 is described as implemented by an integrated circuit 106 as in FIG. 1, it may be executed on other suitable components. For example, FIG. 3 may be implemented in the form of executable instructions on a machine readable storage medium, such as machine-readable storage medium 604 as in FIG. 6.

[0030] At operation 302, the integrated circuit receives a data packet. The data packet may also include a portion of a data stream and as such, FIG. 3 may include selectively securing particular packets in the given data stream. The data packet includes header information for the integrated circuit to process to determine whether the data packet should be processed by the shared security engine at operation 304 for transmission on a secure link. The secure link, associated with the shared security engine, provides the ability to secure the data packet by encryption for transmission to another type of security-enabled networking device. In this regard, since the data packet is encrypted prior to transmission, the secure link is considered dedicated to the secure transmission of the data packet. The non-secure link is considered the link for transmission of data without the security for encryption. The secure link is aggregated with a non-secure link by combining these links in parallel to increase the throughput beyond what a single link may be capable of transmitting.

[0031] At operation 304, the integrated circuit determines whether the data packet should be processed by a shared security engine for transmission on the secure link. The data packet is analyzed to determine whether it may contain sensitive data. The sensitive data may be identified using a header of the data packet, application type of the data packet, and whether the data packet may have been received in ingress or egress from a networking system. At operation 304, data packets may be selectively chosen for encryption based on the sensitive data. In one implementation, operation 304 processes the header corresponding to the data packet for the sensitive data. For example, the data packet may include credit card information and as such,

should be encrypted to maintain confidentiality to the data packet as it is transmitted on the secure link. In this implementation, operation 304 may include the hashing engine 120 as in FIG. 1, which may be bypassed if the determination is the data packet should be processed by the shared security engine for security reasons. Additionally, if the switch includes multiple ports with multiple secure links, each link associated with the shared security engine, the hashing engine 120 may be utilized to determine which link (i.e., communication channel) to transmit the encrypted data packet. In another implementation, the integrated circuit may process the data packet for time sensitive data. In this implementation, the data packet may be directed to the non-secure link for immediate transmission. In this implementation, the shared security engine may cause time latency as the data packet may be encrypted or decrypted prior to transmission, thus to prevent the time latency the data packet may be transmitted on the non-secure link. In a further implementation, if the integrated determines the data packet need not be processed (i.e., no sensitive data), the data packet is routed to operation 308. Operation 308 may include routing the data packet to the non-secure link for transmission. These implementations are explained in further detail in accordance with FIGS. 4-5.

[0032] At operation 306, the integrated circuit routes the data packet to the shared security engine for transmission on the secure link. The data packet is routed based on the determination at operation 304 the data packets should be processed for encryption by the shared security engine. The shared security engine maintains confidentiality of transmitted and received data packets to prevent unauthorized data and/or devices from propagating within a network. As such, the shared security engine may be used across multiple links (e.g., secure links) to provide cryptography techniques. The cryptography techniques may include the exchange of cipher keys, associated encryption/decryption, and authentication techniques to provide network security. In one implementation, operation 306 may determine whether the shared security engine is operating with the maximum number of secure associations and then identify a second shared security engine with an available secure association. The second shared security engine to process the data packet instead of the shared security engine. In another implementation, the integrated circuit identifies whether the data packet was encrypted and routes the data packet to a first MAC associated with the shared security engine. The shared security engine decrypts the

data packet and the integrated circuit receives the decrypted data packet from a second MAC. These implementations are discussed in further detail in accordance with FIGS. 4-5.

[0033] At operation 308, if the integrated circuit determines the data packet is without sensitive data, the data packet is not routed to the shared security engine. In another implementation, operation 308 may include transmitting the data packet on the non-secure link.

[0034] FIG. 4 is a flowchart of an example method to receive a data packet and depending whether the data packet should be processed by the shared security engine, the data packet is either routed for transmission on a non-secure link or to the shared security engine for transmission on a secure link. In discussing FIG. 4, references may be made to the components in FIGS. 1-2B to provide contextual examples. Further, although FIG. 4 is described as implemented by an integrated circuit 106 as in FIG. 1, it may be executed on other suitable components. For example, FIG. 4 may be implemented in the form of executable instructions on a machine readable storage medium, such as machine-readable storage medium 604 as in FIG. 6.

[0035] At operations 402-404, the data packet is received by the integrated circuit for determination of whether the data packet should be processed by the shared security engine. Operation 404 includes the process in which the data packet may either be transmitted on a secure link or non-secure link as at operations 406 or 408. The determination of which link to transmit the data packet depends on whether the integrated circuit determines the data packet should be processed by the shared security engine for security purposes. For example, if the integrated circuit determines the data packet may not be processed, the process moves to operation 406 for transmission on the non-secure link. If the integrated circuit determines the data packet may contain sensitive data, the data packet should be processed to provide security and the process moves to operation 408. Operations 402-404 may be similar in functionality to operations 302-304 as above in accordance with FIG. 3.

[0036] At operation 406, the integrated circuit routes the data packet to the non-secure link for transmission upon determination the data packet is without sensitive data and as such, may not need to be processed by the shared security engine at operation 404. This indicates particular data packets may be selected by the sensitivity of the data itself and processed by the share security engine as at operation 408. For example, certain types of data packets may need to be

secured and with a single link, each data packet may be encrypted prior to transmission. In this example, the packets are processed serially and non-sensitive type of data packets may be in queue behind a sensitive type of data packet, causing a time latency with the data packets. Selecting particular types of data packets for processing by the shared security engine at operation 408 prevents this type of time latency as the data packets may be transmitted on multiple links. In another implementation, operation 406 includes a hashing engine to select a port to direct the data packet to for transmission on the non-secure link.

[0037] At operation 408 the integrated circuit routes the data packet to the shared security engine for transmission on the secure link. In one implementation, the integrated circuit may further determine whether the data packet is encrypted as at operation 410 to either move the process to operations 412-414 or 416-418. Operation 408 may be similar in functionality to operation 306 as in accordance with FIG. 3.

[0038] At operation 410, the integrated circuit may further determine whether the data packet is encrypted. If the data packet is encrypted, then the data packet is routed to a FIRST MAC for the shared security engine to decrypt as at operations 412-414. Operations 412-414 occur upon ingress of the data packet in the networking system for routing to an internal component within the networking system. In turn, if the data packet is not encrypted (i.e., decrypted), the integrated circuit routes the data packet to the second MAC for encryption as at operations 416-418. Operations 416-418 occur upon egress of the data to another networking device through the secure link. This provides an internal type of loopback connection to the integrated circuit for encrypting and/or decrypting data packets prior to transmission to an internal component or to another networking device. This reduces the number of security engines as the process may utilize the shared security engine to maintain data confidentiality.

[0039] At operation 412 the integrated circuit routes the data packet to the first MAC associated with the shared security engine upon determination the data packet is encrypted at operation 410. In one implementation, the shared security engine may include secure associations for multiple links, and as such, may include a type of secure association table. In this implementation, the data packet may include a secure association identifier for the shared security engine to identify a specific key corresponding to the secure association identifier. This

provides another level of security to each of the links to maintain data confidentiality among the links.

[0040] At operation 414 the integrated circuit receives the decrypted data packet from the second MAC. The integrated circuit may then route the decrypted data packet to the authorized networking component or device. In one implementation, the key used to encrypt the data packet at operation 416 may correspond to a key used at operation 414 to decrypt the encrypted data packet. In this operation, the integrated circuit receives the decrypted data packet for internal processing within the network system.

[0041] At operations 416-418, the integrated circuit routes the data packet to the second MAC associated with the shared security engine upon determination the data packet is not encrypted (i.e., decrypted) at operation 410. It may be assumed if the data packet is not encrypted, the data packet should be encrypted upon egress from the networking system. Once receiving the encrypted data packet at operation 418, the encrypted data packet may be transmitted on the secure link.

[0042] FIG. 5 is a flowchart of an example method to receive a data packet and determine the data packet should be processed by the shared security engine. Based upon this determination, a networking component may further determine whether the shared security engine is operation with a maximum number of associations. If the shared security engine is operating with the maximum number of secure association, the integrated circuit routes the data packet to a second shared security engine. FIG. 5 represents the situation of when the shared security engine may be overloaded with client traffic for encrypting and decrypting each respective data packet. Accordingly, the client traffic may be redirected to the available shared security engine. Additionally, FIG. 5 represents the situation of sharing available resources (e.g., secure associations) across multiple security engines. In discussing FIG. 5, references may be made to the components in FIGS. 1-2B to provide contextual examples. Further, although FIG. 5 is described as implemented by an integrated circuit 106 as in FIG. 1, it may be executed on other suitable components. For example, FIG. 5 may be implemented in the form of executable instructions on a machine readable storage medium, such as machine-readable storage medium 604 as in FIG. 6.

Operations 502-508 may be similar in functionality to operations 402-408 as above in accordance with FIG. 4.

[0043] At operation 510, the integrated circuit determines whether the shared security engine at operation 508 is operating with a maximum number of secure associations. At operations 510, a controller may communicate with the integrated circuit to track the number of secure associations in use by each of the shared security engines. The controller may then communicate with the integrated circuit to redirect the data packet to the security engine which has an available secure association. The secure association represents a set of keys used for the security at each security engine; one key for encryption and the other key for decryption. The controller negotiates for each set of keys used in a secure association and as such, may utilize an internal table to track the number of secure associations at each security engine. These sets of keys may be programmed in each of shared security engines for providing data integrity; however, the number of keys may be limited according to the vendor of the shared security engine and as such, each shared security engine may be capable of handling a limited number secure associations. For example, the maximum number of keys may be exhausted at the shared security engine. In this example, the shared security engine may include the number of keys to support four clients for encryption and decryption, as such, when a fifth client requests a secure channel, the integrated circuit may utilize another shared security engine for the encryption and/or decryption. The controller may be part of the switch 104 and 204 as in FIGS. 1-2.

[0044] At operation 512, the integrated circuit identifies the second shared security engine with an available secure association. In this operation, the controller in communication with the integrated circuit, identifies the available secure association at the second shared security engine using a table internal to the networking system as discussed in accordance with operation 510. The table is utilized to track which secure association is in operation at each shared security engine for reference to identify which shared security engine may have the capabilities for the secure association.

[0045] At operation 514, the integrated circuit routes the data packet from the shared security engine to the second shared security engine with the available secure association. In this operation, the controller communicates with the integrated circuit to direct the client traffic in excess of the maximum number to the security engine in which has the available secure

association. In one implementation, the second shared security engine may use secure associations different from the secure associations used at the shared security engine, thus the controller communicates with the integrated circuit for the integrated circuit to obtain the appropriate secure association for a given client's traffic (e.g., data packets). In this regard, the controller instructs the integrated circuit to redirect the traffic to the second shared security engine.

[0046] FIG. 6 is a block diagram of computing device 600 with a processor 602 to execute instructions 608-624 within a machine-readable storage medium 604. Specifically, the computing device 600 with the processor 602 is to process a header of a data packet and routing the data packet accordingly. Although the computing device 600 includes processor 602 and machine-readable storage medium 604, it may also include other components that would be suitable to one skilled in the art. For example, the computing device 600 may include the integrated circuit 106 and/or switch 104 as in FIG. 1. The computing device 600 is an electronic device with the processor 602 capable of executing instructions 606-624, and as such embodiments of the computing device 600 include a computing device, mobile device, client device, personal computer, desktop computer, laptop, tablet, video game console, or other type of electronic device capable of executing instructions 606-624. The instructions 606-624 may be implemented as methods, functions, operations, and other processes implemented as machine-readable instructions stored on the storage medium 604, which may be non-transitory, such as hardware storage devices (e.g., random access memory (RAM), read only memory (ROM), erasable programmable ROM, electrically erasable ROM, hard drives, and flash memory).

[0047] The processor 602 may fetch, decode, and execute instructions 606-624 to process a header of a data packet and route the data packet accordingly. In one implementation, once executing instructions 606-610, the processor may then execute instructions 620-624. In another implementation, once executing instructions 606-610, the processor 602 may then execute instructions 612-618. Specifically, the processor 602 executes instructions 606-610 to: process the header of the data packet to determine if the data packet contains sensitive material; then determine if the data packet contains time-sensitive material; and if the data packet contains no sensitive nor time-sensitive material, the data packet is routed to the non-secure link for transmission. The

processor may then execute instructions 612-618 to: route the data packet to the shared security engine; determine if the shared security engine is operating with the maximum number of secure associations; identify a second shared engine with an available secure association; and route the data packet to the second shared security engine. Once executing instructions 606-610, the processor 602 may other execute instructions 620-624 to: identify whether the data packet is encrypted; route the encrypted data packet to a first MAC associated with the shared security engine for decryption; and receive the decrypted data packet from a second MAC.

[0048] The machine-readable storage medium 604 includes instructions 606-624 for the processor to fetch, decode, and execute. In another embodiment, the machine-readable storage medium 604 may be an electronic, magnetic, optical, memory, storage, flash-drive, or other physical device that contains or stores executable instructions. Thus, the machine-readable storage medium 604 may include, for example, Random Access Memory (RAM), an Electrically Erasable Programmable Read-Only Memory (EEPROM), a storage drive, a memory cache, network storage, a Compact Disc Read Only Memory (CDROM) and the like. As such, the machine-readable storage medium 604 may include an application and/or firmware which can be utilized independently and/or in conjunction with the processor 602 to fetch, decode, and/or execute instructions of the machine-readable storage medium 604. The application and/or firmware may be stored on the machine-readable storage medium 604 and/or stored on another location of the computing device 600.

[0049] In summary, examples disclosed herein provide a shared security engine to maintain data confidentiality and integrity among multiple communication channels or links. The shared security engine reduces the number of security components in a networking system, while also providing data security. Additionally, the examples disclosed herein prevent time latency by enabling selection of data packets for encryption and/or decryption.

CLAIMS

We claim:

1. A system comprising:
 - an integrated circuit to determine whether a data packet should be processed by a shared security engine associated with a secure link; and
 - a first media access control (MAC), associated with the shared security engine, to receive the data packet for transmission on the secure link, based on the determination the data packet should be processed by the shared security engine.
2. The system of claim 1 wherein the integrated circuit is further to:
 - determine whether the shared security engine is operating with a maximum number of secure associations;
 - identify a second shared engine associated with an available secure association;
 - route the data packet from the shared security engine to the second shared engine based on the determination the shared security engine is operating with the maximum number of secure associations; and
 - the second media access control (MAC), associated with the second shared security engine, to receive the data packet for processing through the second shared security engine.
3. The system of claim 1 wherein the integrated circuit is further to:
 - identify whether the data packet is encrypted;
 - route the data packet to the first media access control (MAC) associated with the shared security engine for decryption, the shared security engine associated with the first MAC and a second MAC; and
 - receive the decrypted packet from the second MAC.
4. The system of claim 1 further comprising:
 - a hashing engine to determine which non-secure link to transmit the data packet based upon the determination the data packet should not be processed by the shared security engine.

5. The system of claim 1 wherein the shared security engine is a media access control security engine and is associated with a physical interface.

6. The system of claim 1 further comprising:

the shared security engine to process the data packet by encrypting upon egress of the data packet to the secure link or decrypting the data packet upon ingress of the data packet, the shared security engine includes a table of secure associations for specifying a key for the encryption or decryption of the data packet.

7. A non-transitory machine-readable storage medium encoded with instructions executable by a processor of a computing device, the storage medium comprising instructions to:

process a header of a data packet for a determination of whether the data packet should be processed by a shared security engine associated with a secure link; and

route the data packet to the shared security engine for transmission on the secure link based on the determination the data packet should be processed by the shared security engine.

8. The non-transitory machine-readable storage medium including the instructions of claim 7 and further comprising instructions to:

determine whether the shared security engine is operating with a maximum number of secure associations upon ingress of the data packet;

identify a second shared engine associated with an available secure association;

route the data packet to the second shared security engine instead of the shared security engine based on the determination the data packet should be processed.

9. The non-transitory machine-readable storage medium including the instructions of claim 7 and wherein to process the header of the data packet is further comprising instructions to:

identify whether the data packet is encrypted;

route the data packet to a first media access control (MAC) associated with the shared security engine for decryption based on the identification of the encrypted data packet, the shared

security engine associated with the first MAC and a second MAC; and
receive the decrypted data packet from the second MAC.

10. The non-transitory machine-readable storage medium including the instructions of claim 7 and wherein if the determination is the data packet should not be processed by the shared security engine, the non-transitory machine-readable storage medium is further comprising instructions to:

route the data packet to the non-secure link for transmission.

11. The non-transitory machine-readable storage medium including the instructions of claim 7 wherein to process the header of the packet data is further comprising instructions to:

determine whether the data packet includes time-sensitive data.

12. A method, executed by a networking device, the method comprising:

receiving a data packet;

determining whether the data packet should be processed by a shared security engine associated with a secure link; and

routing the data packet to the shared security engine for transmission on the secure link based on the determination the packet should be processed by the shared security engine.

13. The method of claim 12 wherein determining whether the data packet should be processed by the shared security association is further comprising:

processing a header of the data packet to determine whether the data packet is encrypted;

based upon the determination the data packet is encrypted, routing the data packet to a first media access control (MAC) associated with the shared security engine for decryption, the shared security engine associated with the first MAC and a second MAC; and

receiving the decrypted data packet from the second MAC.

14. The method of claim 12 further comprising:

determining whether the shared security engine is operating with a maximum number of

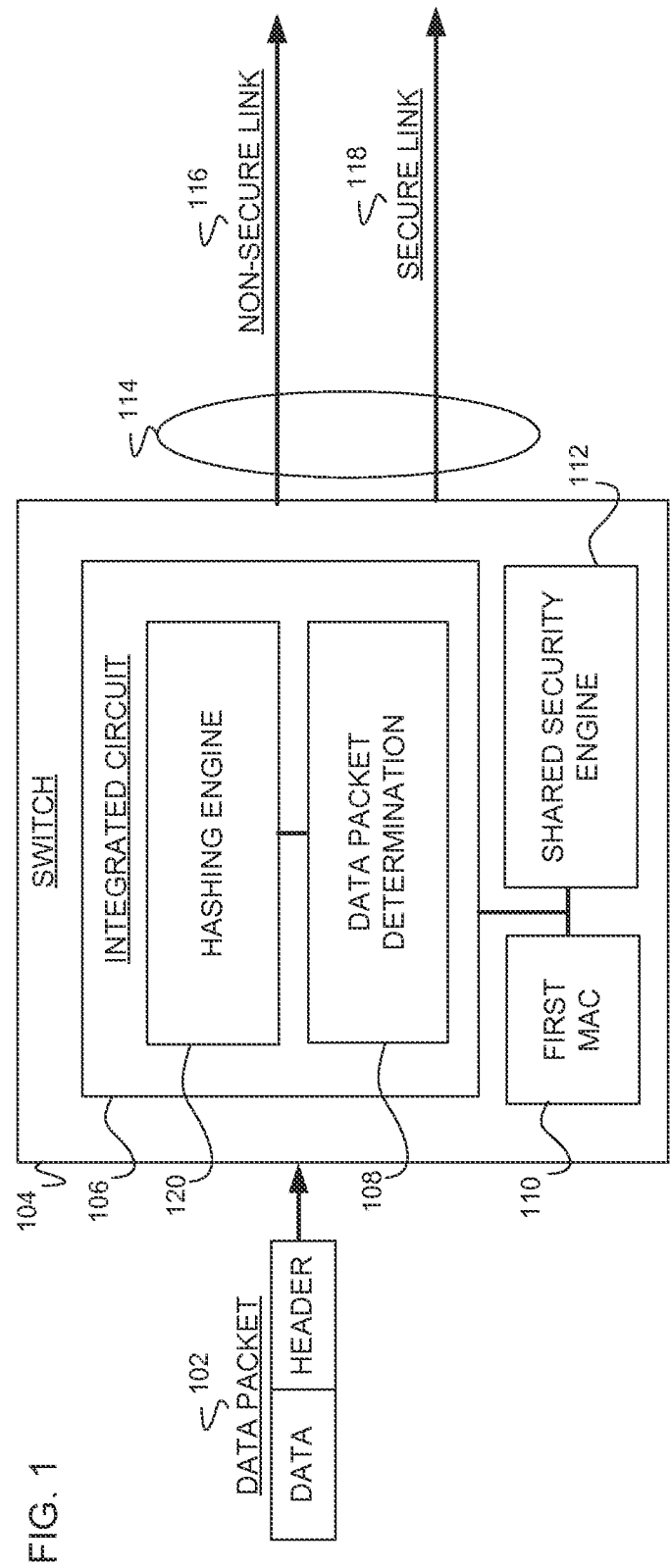
secure associations according to a table;

identify a second shared security engine with an available secure association; and

route the data packet from the shared security engine to the second shared security engine for processing based on the determination the shared security engine is operating with the maximum number of secure associations.

15. The method of claim 12 wherein if the determination is the data packet should not be processed by the shared security engine, the method is further comprising:

routing the data packet to the non-secure link for transmission.



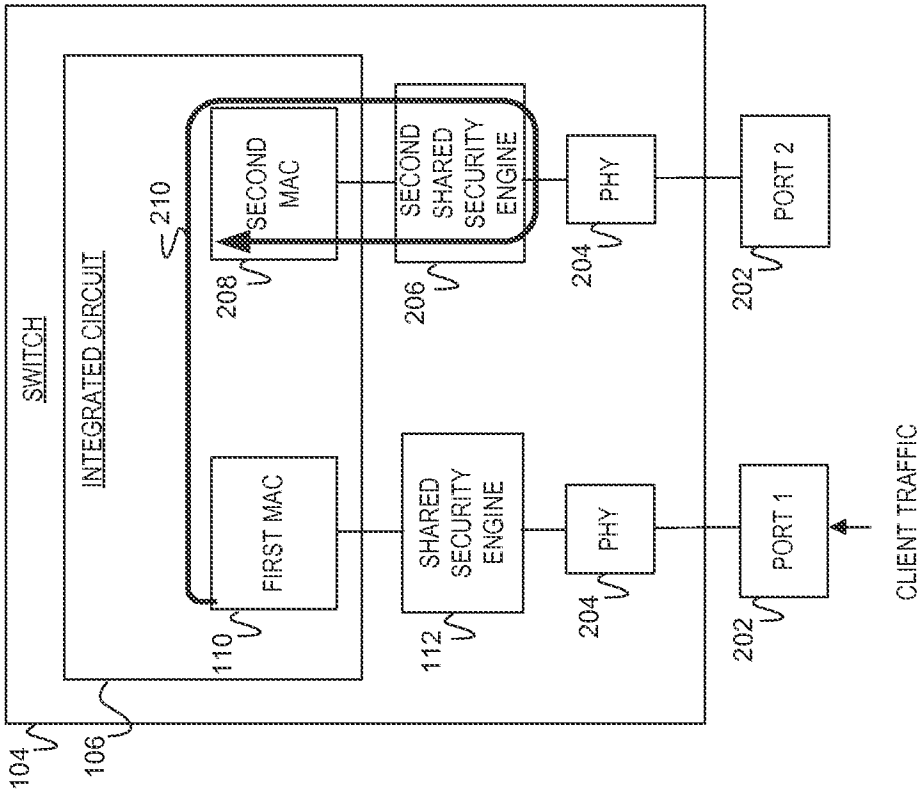


FIG. 2B

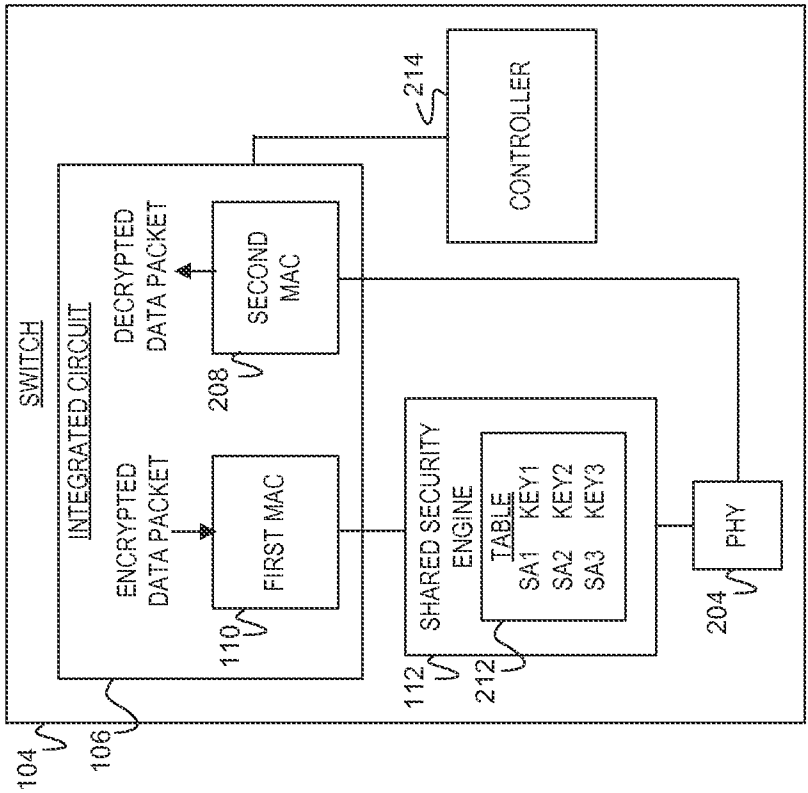
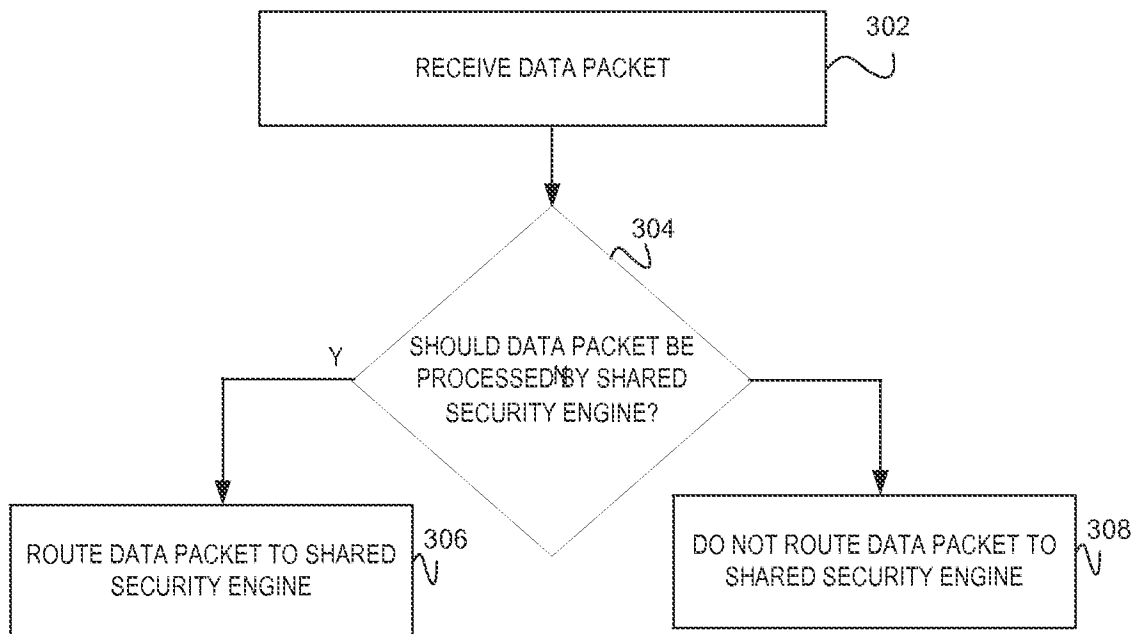


FIG. 2A

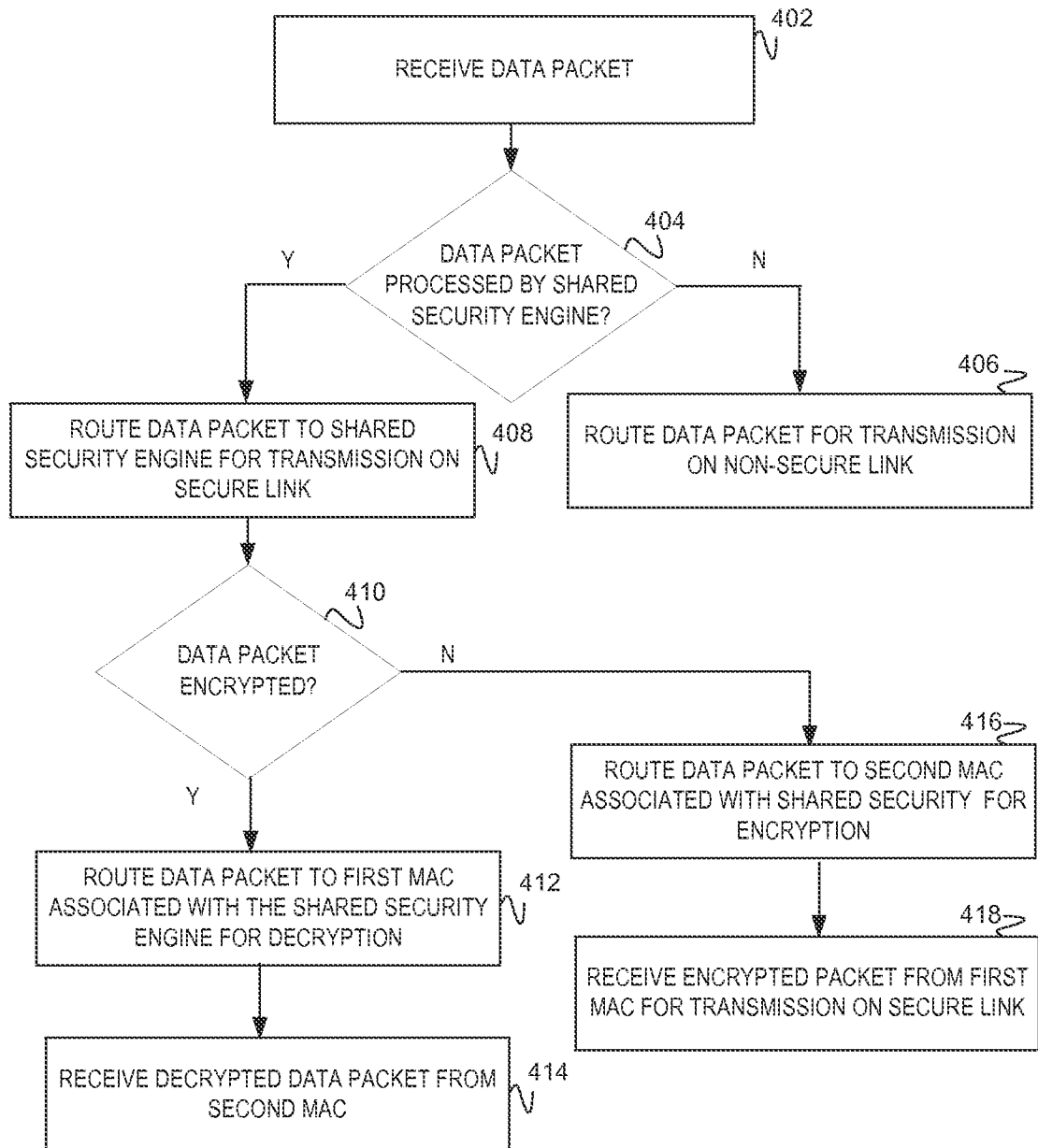
FIG. 3

3/6



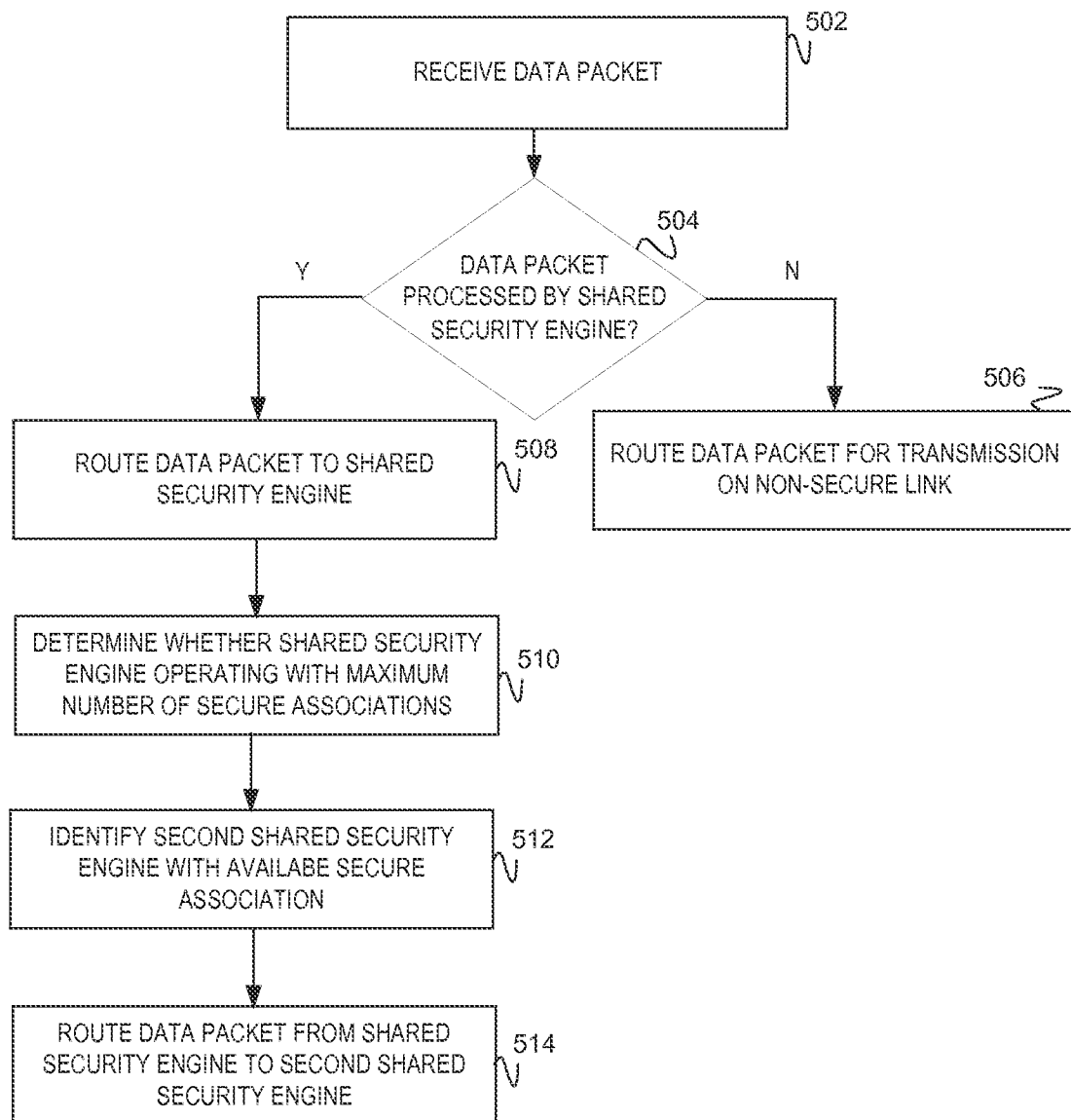
4/6

FIG. 4



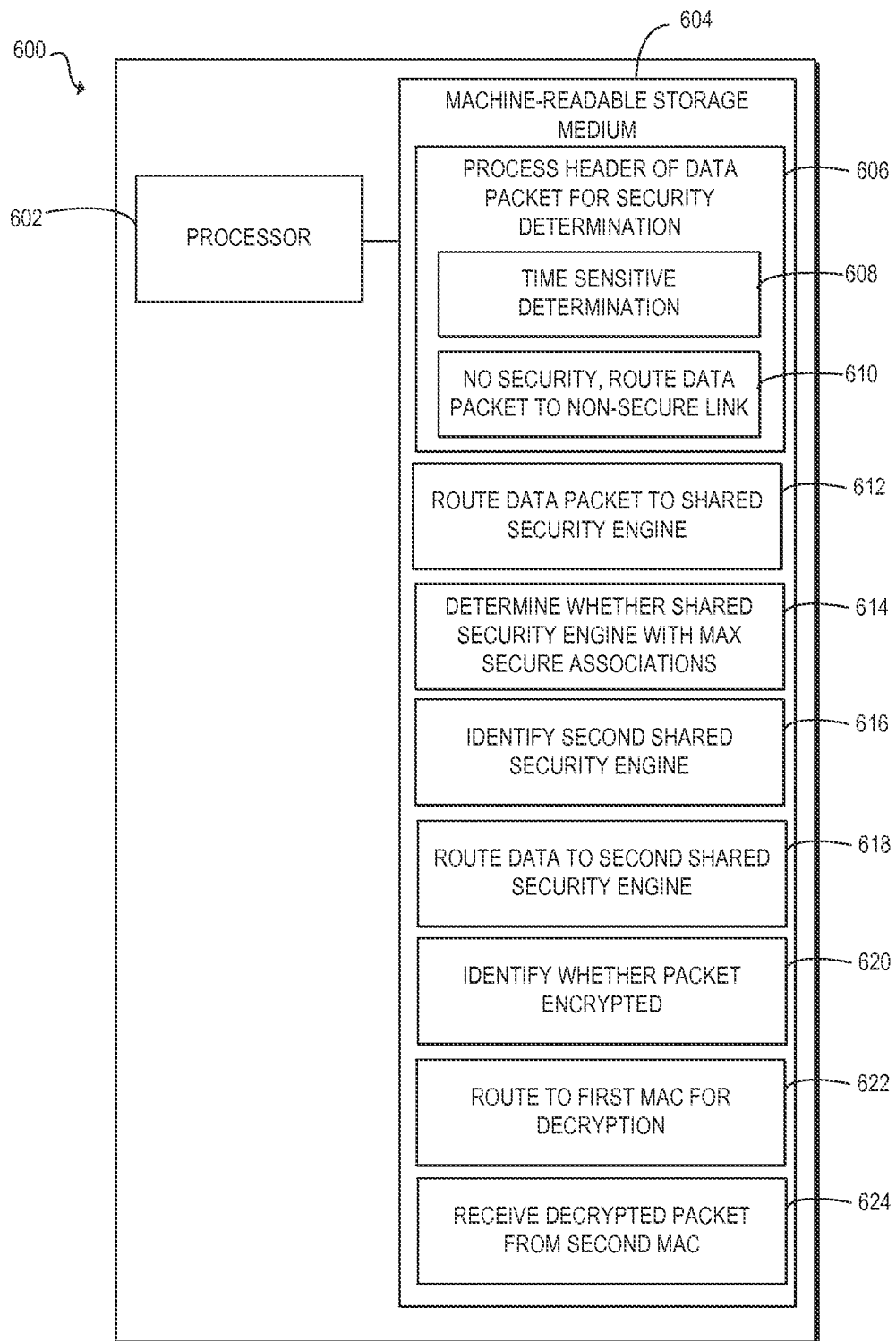
5/6

FIG. 5



6/6

FIG. 6



INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2013/029799**A. CLASSIFICATION OF SUBJECT MATTER****H04L 9/30(2006.01)i, H04L 12/70(2013.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L 9/30; G06F 9/02; H04L 29/06; H04L 9/32; G06F 15/76; G06F 9/00; H04L 9/00; H04L 12/28; H04L 12/70

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: shared security engine, MAC, packet, routing, secure association, encrypt, decrypt

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2010-0217971 A1 (CHANDRAMOULI RADHAKRISHNAN et al.) 26 August 2010 See paragraphs 13, 16, 19-24, 31, 52; claims 1-4, 10-13; and figure 1.	1-10, 12-15
A		11
A	US 2011-0153985 A1 (ASHOKE SAHA et al.) 23 June 2011 See paragraphs 232-236; and figure 6B.	1-15
A	US 7634650 B1 (SHREYAS SHAH et al.) 15 December 2009 See column 1, line 65 - column 2, line 48; and figure 9.	1-15
A	US 2012-0011360 A1 (DANIEL W. ENGELS et al.) 12 January 2012 See paragraphs 26-48; claims 1-9; and figure 1.	1-15
A	US 2008-0031155 A1 (MICHAEL F. KORUS et al.) 07 February 2008 See paragraphs 51-56; claims 1, 2; and figures 5, 8.	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

05 December 2013 (05.12.2013)

Date of mailing of the international search report

06 December 2013 (06.12.2013)

Name and mailing address of the ISA/KR


 Korean Intellectual Property Office
 189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City,
 302-701, Republic of Korea

Facsimile No. +82-42-472-7140

Authorized officer

KANG, Hee Gok

Telephone No. +82-42-481-8264



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2013/029799

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2010-0217971 A1	26/08/2010	CN 102549998 A EP 2401853 A1 WO 2010-126634 A1	04/07/2012 04/01/2012 04/11/2010
US 2011-0153985 A1	23/06/2011	None	
US 7634650 B1	15/12/2009	US 2013-145072 A1 US 7502884 B1 US 7937447 B1 US 8041875 B1 US 8180949 B1 US 8291148 B1	06/06/2013 10/03/2009 03/05/2011 18/10/2011 15/05/2012 16/10/2012
US 2012-0011360 A1	12/01/2012	TW 201215070 A WO 2011-159715 A2	01/04/2012 22/12/2011
US 2008-0031155 A1	07/02/2008	US 7804807 B2 WO 2008-016819 A2 WO 2008-016819 A3	28/09/2010 07/02/2008 23/10/2008