(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 May 2016 (06.05.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/068942 A1

(51) International Patent Classification:
G06F 21/79 (2013.01) *G06F 21/62* (2013.01)

(21) International Application Number:
PCT/US2014/063174

(22) International Filing Date:
30 October 2014 (30.10.2014)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP** [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).

(72) Inventors: **JACQUIN, Ludovic Emmanuel Paul Noel**; Longdown Avenue $ Stoke Gifford, Bristol South Gloucestershire BS34 8QZ (GB). **CHEN, Liqun**; Longdown Avenue $ Stoke Gifford, Bristol South Gloucestershire BS34 8QZ (GB). **DALTON, Chris I.**; Longdown Avenue $ Stoke Gifford, Bristol South Gloucestershire BS34 8QZ (GB).

(74) Agents: **SHOOKMAN, Jeb A.** et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, 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, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, 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, SA, 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, ST, 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,

[Continued on next page]

(54) Title: ENCRYPTION FOR TRANSACTIONS IN A MEMORY FABRIC

(57) Abstract: In an example, memory address encryption is facilitated for transactions between electronic circuits in a memory fabric. An electronic circuit may obtain a transaction integrity key and a transaction encryption key. The electronic circuit may encrypt an address using the transaction encryption key and a compute a truncated message authentication code (MAC) using the transaction integrity key.

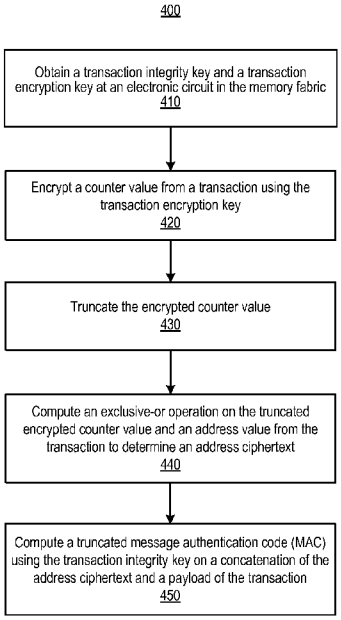

FIG. 4

WO 2016/068942 A1

SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

Published:

— *with international search report (Art. 21(3))*

ENCRYPTION FOR TRANSACTIONS IN A MEMORY FABRIC

BACKGROUND

[0001] Communication protocols exist to provide standard formats for exchanging messages between computers, devices, circuits, etc. In the memory domain, a memory protocol is often used to communication information between memory and a memory controller or other electronic circuits. For example, Double Data Rate (DDR) is a protocol for synchronous dynamic random-access memory (SDRAM). The DDR protocol allows data to be transferred from memory to another electronic circuit on both the rise and fall of a clock cycle. According to the DDR protocol, the data may be transferred from memory using parallel lanes of a data bus.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Features of the present disclosure are illustrated by way of example and not limited in the following figure(s), in which like numerals indicate like elements, in which:

[0003] FIG. 1 shows a block diagram of a memory fabric, according to an example of the present disclosure;

[0004] FIG. 2 shows a packet of a memory fabric protocol, according to an example of the present disclosure;

[0005] FIG. 3 shows a flow chart of a method to facilitate memory address encryption for transactions, according to an example of the present disclosure;

[0006] FIG. 4 shows a flow chart of a method to facilitate pattern analysis protection and memory address encryption for transactions, according to an example of the present disclosure;

[0007] FIG. 5 shows a flow chart of a key management system, according to an example of the present disclosure;

[0008] FIG. 6 shows a flow chart of a method to distribute keys to electronic circuits according to a memory fabric protocol of an example of the present disclosure; and

[0009] FIG. 7 shows a schematic representation of an electronic circuit that is connectable to a memory fabric, according to an example of the present disclosure.

DETAILED DESCRIPTION

[0010] For simplicity and illustrative purposes, the present disclosure is described by referring mainly to an example thereof. In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present disclosure. It will be readily apparent however, that the present disclosure may be practiced without limitation to these specific details. In other instances, some methods and structures have not been described in detail so as not to unnecessarily obscure the present disclosure. As used herein, the terms “a” and “an” are intended to denote at least one of a particular element, the term “includes” means includes but not limited to, the term “including” means including but not limited to, and the term “based on” means based at least in part on.

[0011] Disclosed herein are examples to facilitate memory address encryption for transactions between integrated or discrete electronic circuits of a memory fabric. In one example, the memory fabric is a physical layer topology with electronic circuits that pass data to each other through interconnecting communication channels (e.g., links). Routing on the interconnecting communication channels is performed according to a memory fabric protocol. The memory fabric may be inside of a device such as a mobile device and used to communicate information between electronic circuits inside of the device. A transaction, for instance, includes a data packet with multiple protocol fields that specify an operation and may include an optional payload for exchange between a source electronic circuit and a destination electronic circuit. The electronic circuits of the memory fabric may include processors, memory, memory controllers, input/output (I/O) controllers, storage controllers, field-programmable gate arrays (FPGAs), digital signal processors (DSPs), graphics processing units (GPUs), or any circuit in a computer system. According to the disclosed examples, the transactions may be secured using cryptographic methods for data integrity, data confidentiality, and key management. The memory fabric, as described above, may be internal to a chip or may extend

chip-to-chip interconnections. A memory fabric protocol is a communication protocol specifying rules or standards for communicating in the memory fabric.

[0012] Specifically, the disclosed examples create a separation of ciphered data to prevent redundant encryptions of the same data. By way of example, a memory controller of an electronic circuit may provide both data-in-transit encryption and data integrity for a header of a transaction, while deferring data-at-rest encryption for a payload of the transaction to an end-user. Thus, generally speaking, the disclosed examples may provide data confidentiality and data integrity for transactions while eliminating redundant encryptions of payloads to decrease power consumption and heat generation in the memory fabric.

[0013] Transactions of the disclosed examples may be secured using cryptographic methods for data confidentiality, data integrity, and key management. Data confidentiality refers to placing a limit or restriction on accesses by a third-party to sensitive data in a transaction. Data integrity refers to assuring the accuracy and consistency of a transaction. That is, the transaction is protected against modification in an unauthorized or undetected manner by a third party. Key management refers to the distribution of keys to electronic circuits in the memory fabric. Data-at-rest refers to inactive data that is stored physically in a digital form (e.g., databases, data warehouses, spreadsheets, archives, tapes, off-site backups, mobile devices etc.). Data-in-use refers to active data that is stored in a non-persistent digital state (e.g., in computer random access memory (RAM), computer processing unit (CPU) caches, or CPU registers).

[0014] According to an example of the present disclosure, memory address confidentiality and data integrity may be provided to a transaction between electronic circuits of a memory fabric. In this example, an electronic circuit in the memory fabric may obtain a transaction integrity key and a transaction encryption key. The transaction integrity key and the transaction encryption key are data that can be used to produce a cipher. A truncated key-hashed message authentication code (HMAC) (e.g., 64-bit) may be computed

by the electronic circuit using the transaction integrity key. A hash function, for instance, may be implemented on a concatenation of an address value (e.g., 64-bit) and a payload of the transaction. The electronic circuit may then encrypt a concatenation of the address value and the truncated HMAC using the transaction encryption key to determine a ciphertext (e.g., 128-bit). The ciphertext may be split into an encrypted address (e.g., 64-bit) and an encrypted HMAC (e.g., 64-bit). In this regard, an address field of the transaction may be replaced with the encrypted address and the encrypted HMAC may be placed in a next header field of the transaction according to a memory fabric protocol of the disclosed examples.

[0015] According to another example of the present disclosure, data pattern analysis protection may be provided in addition to memory address confidentiality and data integrity for the transaction. That is, data patterns in the ciphertext may be obscured to prevent a third-party from discerning the ciphertext. In this example, an electronic circuit in the memory fabric may obtain a transaction integrity key and a transaction encryption key. The electronic circuit may encrypt a counter value (e.g., 128-bit) from the transaction using the transaction encryption key and then truncate the encrypted counter value (e.g., 64-bit). An exclusive-or operation may be applied to the truncated encrypted counter value and an address value (e.g., 64-bit) from the transaction to compute an address ciphertext (e.g., 64-bit). The address ciphertext may then be used as an input to a key-hashed message authentication code (HMAC) function or a message authentication code (MAC) function. For example, a truncated MAC or HMAC (e.g., 64-bit) may be computed using the transaction integrity key on a concatenation of the address ciphertext and a payload of the transaction. The truncated MAC or HMAC (e.g., 64-bit), for example, may be computed by applying the transaction integrity key and the concatenation of the address ciphertext and the payload as inputs to an HMAC or MAC function. In an example, standard MAC or HMAC functions may be used. RFC 2104 is a standard promulgated by the Internet Engineering Task Force (IETF) describing an HMAC function that may be used. An address field

of the transaction may be replaced with the address ciphertext and the truncated MAC may be placed in a next header field of the transaction according to a memory fabric protocol of the disclosed examples.

[0016] The examples disclosed herein may be implemented according to a scalable and extensible memory fabric protocol that interconnects numerous electronic circuits at high data transfer rates. The disclosed memory fabric protocol operates on physical layer communications and does not operate on layer 3 of the Open Systems Interconnection (OSI) protocol stack (e.g., Transmission Control Protocol (TCP)/Internet Protocol (IP); TCP/IP). For example, the disclosed memory fabric protocol may include an abstract physical layer interface to support multiple physical layers, thus allowing the disclosed memory fabric protocol to be tailored to market needs independent of devices (e.g., network interface controller (NIC)) or operating systems.

[0017] The disclosed memory fabric protocol may be a serial protocol to provide high-bandwidth in the communication channels of the memory fabric and enhance the scalability of the electronic circuits of the memory fabric. In other words, the serial memory fabric protocol may provide scalability of bandwidth, as it is easier to add more communication channels to scale the bandwidth.

[0018] The improved scalability and extensibility of the disclosed memory fabric protocol, however, may expose an electronic circuit to malicious or compromised electronic circuits in the distributed memory fabric. Accordingly, examples of the present disclosure use cryptography to safeguard the integrity and confidentiality of transactions between electronic circuits. In particular, the disclosed examples provide cryptographic security for a high-performance, scalable and extensible memory fabric protocol with small minimum-size packets.

That is, the disclosed examples provide authentication and access control in a memory fabric protocol where minimal overhead space (e.g., 8 bytes) is allotted for security of transmissions.

[0019] In contrast to the disclosed memory fabric protocol, software layer security protocols, such as TCP/IP security, operate over slower network technologies and tolerate more latency, more cost, more overhead and more complexity. For example, the minimum overhead for a Transport Layer Security/Secure Sockets Layer (TLS/SSL) packet is 90 bytes per packet, which is about a 6% overhead, given 1500 byte packets. According to an example, the minimum size of a packet for the disclosed memory fabric protocol may be as low as 24 bytes. Accordingly, minimal bytes may be allocated in the packet for security. As a result, the disclosed memory fabric protocol provides lower-overhead and lower-cost to maximize the scalability and extensibility of electronic circuits.

[0020] With reference to FIG. 1, there is shown a block diagram of a memory fabric 100, according to an example of the present disclosure. It should be understood that the memory fabric 100 may include additional components and that one or more of the components described herein may be removed and/or modified without departing from a scope of the memory fabric 100.

[0021] The memory fabric 100 is depicted by way of example as including electronic circuits such as a processor 102, a memory controller 104, volatile memory 106a-f, non-volatile memory 108a-b, a storage controller 110, a graphical processor unit (GPU) 112, a field-programmable gate array (FPGA) 114, and a digital signal processor (DSP) 116. Each electronic circuit may include one or more physical interfaces to communicate with another physical interface via a link 105 (i.e., communication channel) between two electronic circuits.

[0022] According to an example, a link 105 may include at least one transmitter lane and one receiver lane and may be either symmetric or asymmetric. A link includes a physical medium for transmitting signals. A symmetric link is a link where the number of transmitter lanes is equal to the number of receiver lanes. Conversely, an asymmetric link is a link where the number of transmitter lanes is not equal to the number of receiver lanes.

Depending upon the underlying physical layer capabilities, the number of transmitter and receiver lanes may be statically provisioned or dynamically adjusted on a per-link basis. Additional capacity and performance scaling can be achieved through the use of integrated or discrete switches. As a result, a variety of topologies may be constructed from simple daisy chains to star to 3D-Torus to increase aggregate performance and optionally improve resiliency of the memory fabric.

[0023] The processor 102, which may be a microprocessor, a micro-controller, an application specific integrated circuit (ASIC), or the like, is to perform various processing functions in the memory fabric 100. For example, the processor may perform the function of securing transactions between the electronic circuits of a memory fabric 100. The memory controller 104 is an electronic circuit that manages the flow of data (e.g., high-level transactions such as reads, writes, etc.) going to and from the memory 106a-f. The memory controller 104 may be a separate electronic circuit or may be integrated into the die of the processor 102. The memory 106a-f may include static RAM (SRAM), dynamic RAM (DRAM), or the like. Moreover, each of the memory 106a-f may include a separate media controller to service the high-level transactions and perform media-specific services and management.

[0024] The processor 102, memory controller 104, and memory 106a-f may be coupled by links to the non-volatile memory 108a-b, the storage controller 110, the GPU 112, the FPGA 114, and the DSP 116. The non-volatile memory 108a-b may include read-only memory (ROM), flash memory, magnetic computer storage devices, and the like. The storage controller 110, for instance, may communicate with a hard disk or disk drive. The GPU 112 may manipulate and alter memory 106a-f to accelerate the creation of images in a frame buffer intended for output to a display. The FPGA 114, for example, is an electronic circuit that may be programmed after manufacturing. Lastly, the DSP 116 may be used to measure, filter and/or compress continuous real-world analog signals. These electronic circuits (e.g., the non-volatile memory 108a-b,

the GPU 112, the FPGA 114, and the DSP 116) may access the memory 106a-f through the memory controller 104 of the memory fabric 100.

[0025] As noted above, the protocol for the memory fabric 100 may be optimized to support memory semantic communications using a scalable packetized transport with scalable and power-proportional link, physical layers, and underlying memory media access. That is, each market segment may require one or more market-specific physical layers to be supported. According to an example, the memory fabric protocol includes an abstract physical layer interface to support multiple physical layers and media. As a result, the physical layers may evolve or be replaced without disrupting or waiting for the entire ecosystem to move in lock-step. The abstract physical layer may include the media access control sublayer, physical coding sublayer, and physical medium attachment sublayer. The electronic circuits of the memory fabric 100 may implement the abstract physical layer to facilitate interoperable communications.

[0026] With reference to FIG. 2, there is shown an example of a packet 200 of the memory fabric protocol of the present disclosure. The packet 200, for instance, may include a Next Header field that carries security information (e.g., SMH) to enable electronic circuits of the memory fabric to authenticate that a transaction was transmitted by an authorized source and was not tampered with during transit.

[0027] It should be understood that the packet 200 may include additional protocol fields and that one or more of the protocol fields described herein may be removed and/or modified without departing from a scope of the packet 200. The packet 200 is depicted as including the protocol fields shown in TABLE 1 below.

[0028] TABLE 1

Field Name	Size (Bits)	Description

Access Keys	Variable (e.g., 16)	Indicates restricted or unrestricted access to targeted resources.
Transaction-Specific	Variable	May have different meanings for different transactions.
Next Header Present (N)	1	Indicates if a 64-bit next header field follows the initial 64-bits of protocol header.
Operation Code (OpCode)	5	Indicates an operation type as well as an operation payload size (if present).
OpClass (OCL)	Variable (e.g., 4)	Indicates an operation class.
OpClass Present (O)	1	Indicates whether the OCL field is present. The presence of the OCL may indicate the presence of the access key field presence.
Length (Len)	6	Indicates the encoded transaction length in 64-bit multiples.
Destination Component Identifier (DCID)	11	Identifies a destination electronic circuit.
Virtual Channel (VC)	3	Identifies a virtual channel for the transaction.
Transaction Type (TT)	2	Indicates the type of the transaction (e.g., link-local, unicast end-to-end transaction, multicast end-to-end, etc.).
Address	64	Indicates a unique identifier to access or target an electronic circuit's resources (e.g., memory).

Payload	Variable	Transaction-specific payload.
End-to-End Cyclic Redundancy Check (ECRC)	24	Data Integrity Field
Priority (Pri)	3	Differentiates transaction processing within the receiving electronic circuit by, for instance, determining the transaction execution order.
Operation Flags (OFlags)	6	Various operation flags.
Tag	20	Associates a request with a response or acknowledgement.
Source Component Identifier (SCID)	11	Identifies a source electronic circuit.

[0029] According to an example, the Next Header field may be included in the packet 200 to allow multiple semantics to be attached based on solution needs. For instance, the Next Header may be a security message header (SMH) that includes a 64-bit HMAC. In this example, the Next Header enables electronic circuits to authenticate that the transaction was transmitted by an authorized source and was not tampered with during transit. The presence of the Next Header does not impact transaction relay and if an electronic circuit does not support the configured meaning, the Next Header may be ignored upon receipt. That is, the electronic circuit may locate the subsequent payload or protocol fields even if it does not support the Next Header.

[0030] With reference to FIG. 3, there is shown a flow chart of a method 300 to facilitate memory address encryption for transactions according to a memory fabric protocol of an example of the present disclosure. The method

300 may be implemented, for example, by one of the electronic circuits depicted in FIG. 1.

[0031] At block 310, an electronic circuit in the memory fabric may obtain a transaction integrity key (TIK) and a transaction encryption key (TEK). According to one example, the TIK and TEK may be received from a key distribution server as further discussed below with reference to FIGS. 5 and 6. According to another example, the TIK and TEK may be provided by to the electronic circuit during manufacture.

[0032] At block 320, the electronic circuit may compute a truncated message authentication code (MAC), such as a key-hashed message authentication code (HMAC), using the TIK. According to an example, a hash function may be applied to a concatenation of an address value (addr) and a payload of a transaction (e.g., HMAC(TIK, addr + payload)). The HMAC may then be truncated to fit the 64-bits allocated to the Next Header field of the transaction according to the disclosed memory fabric protocol.

[0033] At block 330, the electronic circuit may then encrypt a concatenation of the address value (e.g., 64-bits) and the truncated HMAC (e.g., 64-bits) using the TEK to determine a ciphertext (e.g., 128-bit). According to an example, the plaintext of the address value and the truncated HMAC may be encrypted using an electronic codebook (ECB) encryption mode since the address value and the truncated HMAC may both be 64-bits each (e.g., addr + HMAC = ECB_encrypt(TEK, addr + HMAC)).

[0034] At block 340, the ciphertext (e.g., 128-bit) may be split into an encrypted address (e.g., 64-bit) and an encrypted HMAC (e.g., 64-bit). In this regard, for example, an address field of the transaction may be replaced with the encrypted address and the encrypted HMAC may be placed in the Next Header field of the transaction, as shown in block 350.

[0035] According to an example, the electronic circuit may also decrypt and verify an HMAC of a received transaction. For instance, the electronic circuit may receive a transaction from another electronic circuit in the memory

fabric and decrypt a concatenation of the encrypted address and the encrypted HMAC using the TIK (e.g., addr + HMAC = ECB_decrypt(TEK, addr + HMAC)). The electronic circuit may then, for example, verify the integrity of the decrypted HMAC using the TIK.

[0036] With reference to FIG. 4, there is shown a flow chart of a method 400 to facilitate pattern analysis protection and memory address encryption for transactions according to a memory fabric protocol of an example of the present disclosure. That is, the method 400 may further obscure data patterns in the ciphertext to prevent a third-party from discerning the ciphertext. The method 400 may be implemented, for example, by one of the electronic circuits depicted in FIG. 1.

[0037] At block 410, an electronic circuit in the memory fabric may obtain a transaction integrity key (TIK) and a transaction encryption key (TEK). According to an example, the TIK and TEK may be received from a key distribution server as further discussed below with reference to FIGS. 5 and 6. According to another example, the TIK and TEK may be provided by to the electronic circuit during manufacture.

[0038] In block 420, the electronic circuit may encrypt a counter value from a transaction using the TEK. The counter value may be taken from a field in the transaction, which may be a packet field. The counter value, for instance, may be 128-bits and may be included in a tag field in a footer of the transaction. Alternatively, a random number (e.g., nonce) may be encrypted using the TEK. The encrypted counter value or random number may then be truncated to the first 64-bits from the 128-bits for example, as shown in block 430.

[0039] At block 440, the electronic circuit may compute an exclusive-or operation on the truncated encrypted counter value and a 64-bit address value (addr) from the transaction to determine an address ciphertext. The address ciphertext, for instance, may be 64-bits in length. According to an example, a counter (CTR) mode of encryption may be used to determine the address

ciphertext (addr_ciph) of the transaction (e.g., addr_ciph = CTR_encrypt(TEK, counter, addr) = CTR_encrypt(TEK, counter) XOR addr).

[0040] At block 450, the electronic circuit may compute a truncated message authentication code (MAC) using the TIK on a concatenation of the address ciphertext and a payload of the transaction. According to an example, the truncated MAC may be 64-bits and may be a keyed-hash message authentication code (HMAC) (e.g, HMAC(TIK, ciph_addr + payload). As a result, therefore, an address field of the transaction may be replaced with the address ciphertext and the truncated MAC or HMAC may be placed in the Next Header field of the transaction.

[0041] According to an example, the electronic circuit may also decrypt and verify a MAC or HMAC of a received transaction. For instance, the electronic circuit may receive a transaction from another electronic circuit in the memory fabric and verifying a MAC or HMAC of the received transaction using the TIK. The electronic circuit may then encrypt a counter value (or an encrypted nonce) of the received transaction using the TEK and compute an exclusive-or operation on the encrypted counter value and an address ciphertext to determine the address value for the received transaction (e.g,. addr = CTR_encrypt(TEK, counter, addr_cipher) = decrypt(TEK, counter) XOR addr_cipher).

[0042] FIG. 5 shows a block diagram of a key management system 500, according to an example of the present disclosure. The key management system 500 according to the memory fabric protocol may include a source electronic circuit 510, a destination electronic circuit 520, a key distribution server 530, and intervening electronic circuits 540.

[0043] According to an example, cryptographic keys are managed by a centralized third-party key distribution server 530. Each electrical circuit in the memory fabric 100 (source electronic circuit 510, destination electronic circuit 520, and key distribution server 530) may have a unique identifier and a public/private key pair. The public/private key pair, for example, may be

generated during manufacture of the respective electronic circuits. A key may be cryptographically bound with the unique identifier. For example, the public key may directly serve as the identifier for the respective electronic circuit.

[0044] An electrical circuit in the memory fabric 100 may be assigned to be the key distribution server 530. Each transaction according to the memory fabric protocol may include the source electronic circuit 510, intervening electronic circuits 540, and a destination electronic circuit 520 (or multiple destination electronic circuits). Accordingly, during the operation of the memory fabric protocol, if the source electronic circuit 510 and the destination electronic circuit 520 have not yet established a shared TIK, the key distribution server by 530 may distribute the TIK to the source electronic circuit 510 and the destination electronic circuit 520 using the key management method described below in FIG. 6.

[0045] According to an example, an asymmetric key belonging to the source electronic circuit 510 or the destination electronic circuit 520 may be an encryption/decryption key pair, and the asymmetric key belonging to a key distribution server 530 may be a digital signature/signature verification key pair. The key distribution server 530 may have an authentic copy of both the source electronic circuit's public key and destination electronic circuit's public key. Additionally, both the source electronic circuit 510 and the destination electronic circuit 530 may have an authentic copy of the key distribution server's public key. According to an example, a key certification service may provide authentic copies of the public keys to the respective electronic circuits as discussed above. Alternatively, during a first computer boot sequence, each electronic circuit may register their identifier and public key with the key distribution server 530 assuming that the communication channel between the key distribution server 530 and the source electronic circuit 510 or destination electronic circuit 520 is safe.

[0046] With reference to FIG. 6, there is shown a flow chart of a method 600 to distribute keys to electronic circuits according to a memory fabric protocol of example of the present disclosure. The method 600 may be

implemented, for example, by one of the electronic circuits depicted in FIG. 1. The method 600 allows the key distribution server 530 to generate and distribute a TK to the source electronic circuit 510 and to one or more destination electronic circuits.

[0047] At block 610, the key distribution server 530 may generate a TK, TIK and/or TEK from scratch. Alternatively, the key distribution server 530 may derive a TIK and a TEK from a TK using a master key with a key derivation function (KDF).

[0048] At block 620, the key distribution server 530 may distribute the TK, TIK, and/or the TEK to the source electronic circuit 510 and destination electronic circuit 520 using a combined encryption and signature algorithm. For example, the key distribution server (KDS) 530 may encrypt a plaintext under the receiver's public key (e.g., source electronic circuit (SEC) 510 or the destination electronic circuit (DEC) 520) and sign the ciphertext under the key distribution server's private key.

[0049] Accordingly, the key distribution server 530 may then transmit the encrypted and signed TIK to the source electronic circuit 510 (e.g., sig_KDS(enc_SEC(TK, t1), t2)) and to the destination electronic circuit 520 (e.g., sig_KDS(enc_DEC(TK, t1), t2), where t1 and t2 denote optional tests such as a time value, sequence value or nonce, which may be used against replay attacks.

[0050] Thus, the source electronic circuit 510 or the destination electronic circuit 520 may receive the encrypted and signed TK, TIK, and/or TEK from the key distribution server 530. If the source electronic circuit 510 or the destination electronic circuit 520 receives a TK, the source electronic circuit 510 or the destination electronic circuit 520 may derive a TIK and TEK from the TK using a key derivation function (e.g., TIK = KDF(TK, "Integrity") and TEK = KDF(TK, "Encryption"), where "Integrity" and "Encryption" represent distinct salt values). In order to maintain the security level for encryption and MAC using the derived keys, the TK should have sufficient entropy. For example, the length of TK may

be longer than the length of max (TIK, TEK). The source electronic circuit 510 or the destination electronic circuit 520 may then verify the integrity of the received TK, TIK, and/or TEK using an authentic copy of a public digital verification key of the key distribution server 530, and decrypt the received TK, TIK, and/or TEK using a private decryption key of the respective electronic circuit.

[0051] At block 630, the key distribution server 530 may certify the source electronic circuit 510 and destination electronic circuit 520. According to an example, the key distribution server 530 may introduce the source electronic circuit 510 and destination electronic circuit 520 by including their identifiers along with the distributed TIK. For instance, the key distribution server may transmit the TK to the source electronic circuit 510 as sig_KDS(enc_SEC(DEC, TK, t1), t2) and to the destination electronic circuit 520 as sig_KDS(enc_DEC(SEC, TK, t1), t2).

[0052] While FIGS. 5 and 6 show key management and subsequent communication between a pair of electronic circuits, the method 600 may be used for a larger number of communicating peers. In the case in which the source electronic circuit 510 communicates with multiple destination electronic circuits, the source electronic circuit 510 and the multiple destination electronic circuits may belong to a group that all share the same TIK. Alternatively, the source electronic circuit 510 and the multiple destination electronic circuits may be partitioned into several subgroups, each of the subgroups sharing a single TIK.

[0053] Some or all of the operations set forth in the methods 300, 400, and 600 may be contained as utilities, programs, or subprograms, in any desired computer accessible medium. In addition, the methods 300, 400, and 600 may be embodied by computer programs, which may exist in a variety of forms both active and inactive. For example, they may exist as machine readable instructions, including source code, object code, executable code or other formats. Any of the above may be embodied on a non-transitory computer readable storage medium.

[0054] Examples of non-transitory computer readable storage media include conventional computer system RAM, ROM, EPROM, EEPROM, and magnetic or optical disks or tapes. It is therefore to be understood that any electronic device capable of executing the above-described functions may perform those functions enumerated above.

[0055] Turning now to FIG. 7, a schematic representation of an electronic circuit 700 of the memory fabric is shown according to an example of the present disclosure. Examples of the electronic circuit 700 may include a central processor, memory controller, GPU, etc. that can send and receive secure transactions in the memory fabric shown in FIG. 1. The electronic circuit 700 may be employed to perform various functions of methods 300, 400, and 600 as depicted in FIGS. 3, 4 and 6 according to an example implementation. The electronic circuit 700 may include a hardware controller 702, a local memory 708, and a memory fabric interface 710. Each of these components may be operatively coupled to a link 712.

[0056] The local memory 708 may be a computer readable medium that stores machine readable instructions which are executable by the hardware controller 702 to perform the various functions of methods 300, 400, and 600 as depicted in FIGS. 3, 4 and 6. For example, the local memory 708 may store a transaction module 712 that is executable by the hardware controller 702 to obtain a transaction integrity key and a transaction encryption key. The local memory 708 may also store an encryption module 714 that is executable by the hardware controller 702 to encrypt an arbitrary number using the transaction encryption key, wherein the encrypted arbitrary number is truncated in length, and to calculate an exclusive-or operation on the encrypted arbitrary number and an address value from the transaction to determine an address ciphertext. Further, the local memory 708 may store an authentication module 716 that is executable by the hardware controller 702 to calculate a truncated keyed-hash message authentication code (HMAC) using the transaction integrity key on a concatenation of the address ciphertext and a payload of the transaction. The

transaction may be transmitted via the memory fabric interface 710, which connects the electronic circuit 700 to a memory fabric.

[0057] What has been described and illustrated herein are examples of the disclosure along with some variations. The terms, descriptions and figures used herein are set forth by way of illustration only and are not meant as limitations. Many variations are possible within the scope of the disclosure, which is intended to be defined by the following claims -- and their equivalents -- in which all terms are meant in their broadest reasonable sense unless otherwise indicated.

CLAIMS

What is claimed is:

1. A method to facilitate memory address encryption for transactions between electronic circuits in a memory fabric, comprising:

obtaining, at an electronic circuit in the memory fabric, a transaction integrity key and a transaction encryption key;

encrypting a counter value from a transaction using the transaction encryption key;

truncating the encrypted counter value;

computing an exclusive-or operation on the truncated encrypted counter value and an address value from the transaction to determine an address ciphertext; and

computing a truncated message authentication code (MAC) using the transaction integrity key on a concatenation of the address ciphertext and a payload of the transaction.

2. The method of claim 1, comprising:

replacing an address field of the transaction with the address ciphertext; and

placing the truncated MAC in a next header field of the transaction.

3. The method of claim 1, wherein the counter value is 128-bit and the truncated encrypted counter value is a first 64-bits of the encrypted counter value.

4. The method of claim 1, wherein the address value is 64-bits and the

truncated MAC is 64-bits.

5. The method of claim 1, wherein the truncated MAC is keyed-hash message authentication code (HMAC).

6. The method of claim 1, comprising:

receiving a transaction from another electronic circuit in the memory fabric;

verifying a MAC of the received transaction using the transaction integrity key;

encrypting a counter value of the received transaction using the transaction encryption key; and

computing an exclusive-or operation on the encrypted counter value and an address ciphertext to determine the address value for the received transaction.

7. The method of claim 1, wherein obtaining the transaction integrity key and the transaction encryption key comprises receiving the transaction integrity key and the transaction encryption key from a key distribution server in the memory fabric.

8. A method to facilitate memory address encryption for transactions between electronic circuits in a memory fabric, comprising:

obtaining, at an electronic circuit in the memory fabric, a transaction integrity key and a transaction encryption key;

computing a truncated message authentication code (MAC) using the transaction integrity key, wherein a hash function is applied to a concatenation of an address value and a payload of a transaction;

encrypting a concatenation of the address value and the truncated MAC using the transaction encryption key to determine a ciphertext;

splitting the ciphertext into an encrypted address and an encrypted MAC; and

placing the encrypted address in an address field of the transaction and the encrypted MAC in a next header field of the transaction.

9. The method of claim 8, wherein to compute a truncated MAC, wherein computing the truncated MAC comprises computing a 64-bit key-hashed message authentication code (HMAC).

10. The method of claim 8, wherein splitting the ciphertext comprises splitting the ciphertext into a 64-bit encrypted address and a 64-bit encrypted MAC.

11. The method of claim 8, comprising:

receiving a transaction from another electronic circuit in the memory fabric;

decrypting a concatenation of the encrypted address and the encrypted MAC using the transaction encryption key; and

verify the integrity of the decrypted MAC using the transaction integrity key.

12. An electronic circuit connectable to a memory fabric, the electronic circuit comprising:

a hardware controller; and

a local memory storing machine readable instructions, executable by the hardware controller, including:

a transaction module to obtain a transaction integrity key and a transaction encryption key;

an encryption module to encrypt an arbitrary number using the transaction encryption key, wherein the encrypted arbitrary number is truncated in length, and calculate an exclusive-or operation on the encrypted arbitrary number and an address value from the transaction to determine an address ciphertext; and

an authentication module to calculate a truncated keyed-hash message authentication code (HMAC) using the transaction integrity key on a concatenation of the address ciphertext and a payload of the transaction.

13. The electronic circuit of claim 12, comprising machine readable instructions that are executable by the hardware controller to:

replace an address field of the transaction with the address ciphertext; and

place the truncated HMAC in a next header field of the transaction.

14. The electronic circuit of claim 12, comprising machine readable instructions that are executable by the hardware controller to:

receive a transaction from another electronic circuit in the memory fabric;

verify a HMAC of the received transaction using the transaction integrity key;

encrypt an arbitrary number of the received transaction using the transaction encryption key; and

compute an exclusive-or operation on the encrypted arbitrary number and an address ciphertext to determine the address value for the received transaction.

15. The electronic circuit of claim 12, wherein to obtain the transaction integrity key and the transaction encryption key, the machine readable instructions are executable by the hardware controller to obtain the transaction integrity key and the transaction encryption key from a key distribution server in the memory fabric.

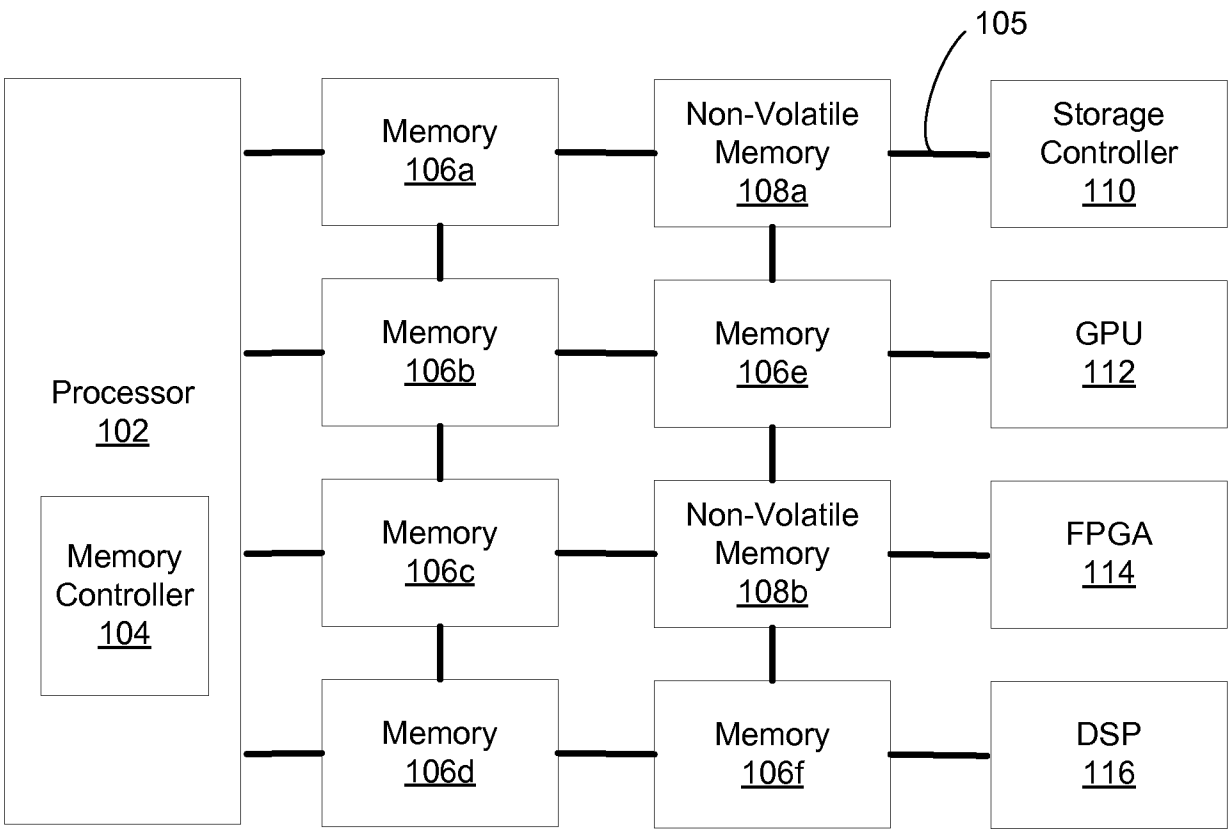
100
105
Processor
102
Memory
Controller
104
Memory
106a
Memory
106b
Memory
106c
Memory
106d
Non-Volatile
Memory
108a
Memory
106e
Non-Volatile
Memory
108b
Memory
106f
Storage
Controller
110
GPU
112
FPGA
114
DSP
116

FIG. 1

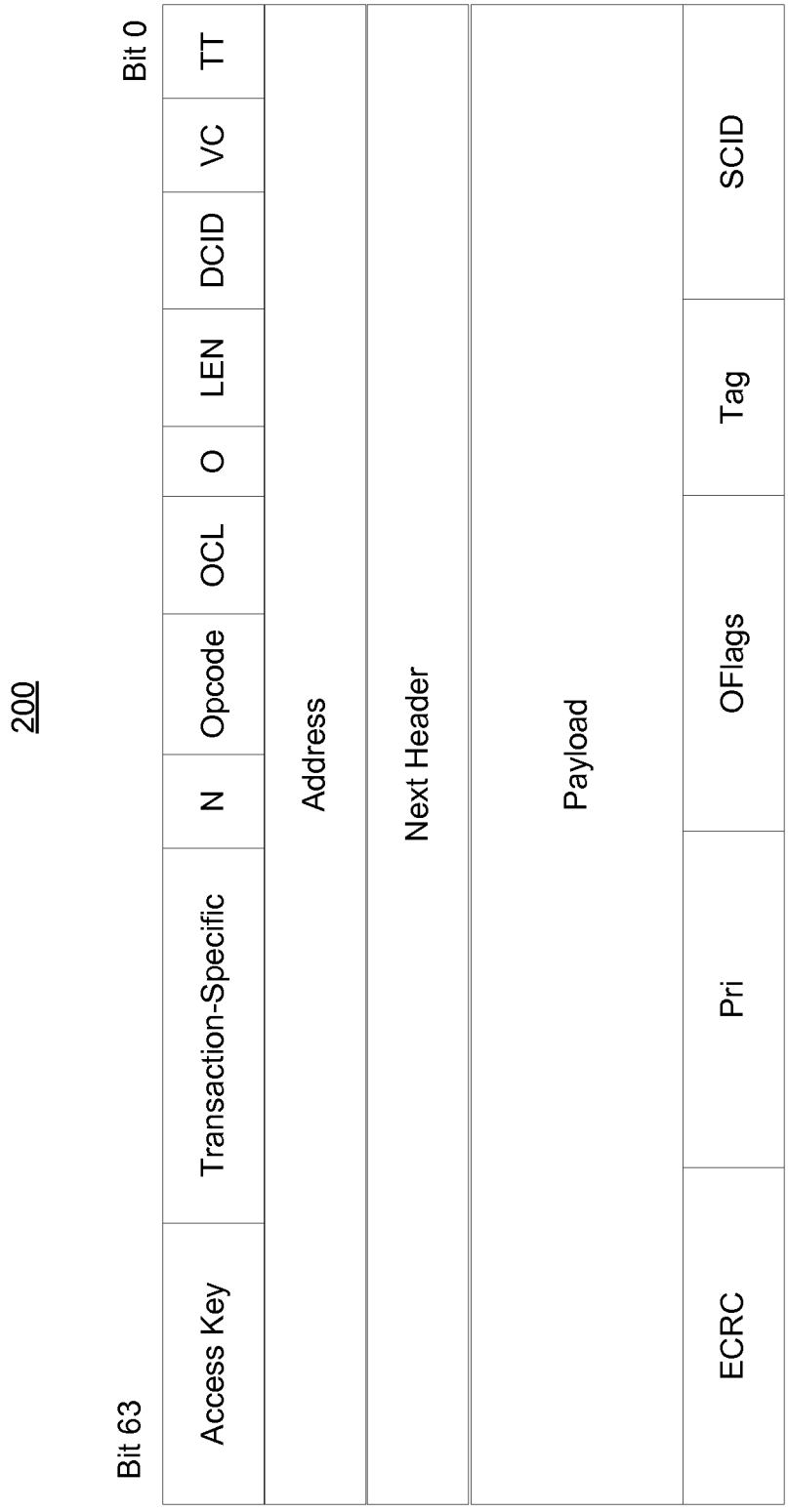
200
Bit 63
Bit 0
Access Key
Transaction-Specific
N
Opcode
OCL
O
LEN
DCID
VC
TT
Address
Next Header
Payload
ECRC
Pri
OFlags
Tag
SCID

FIG. 2

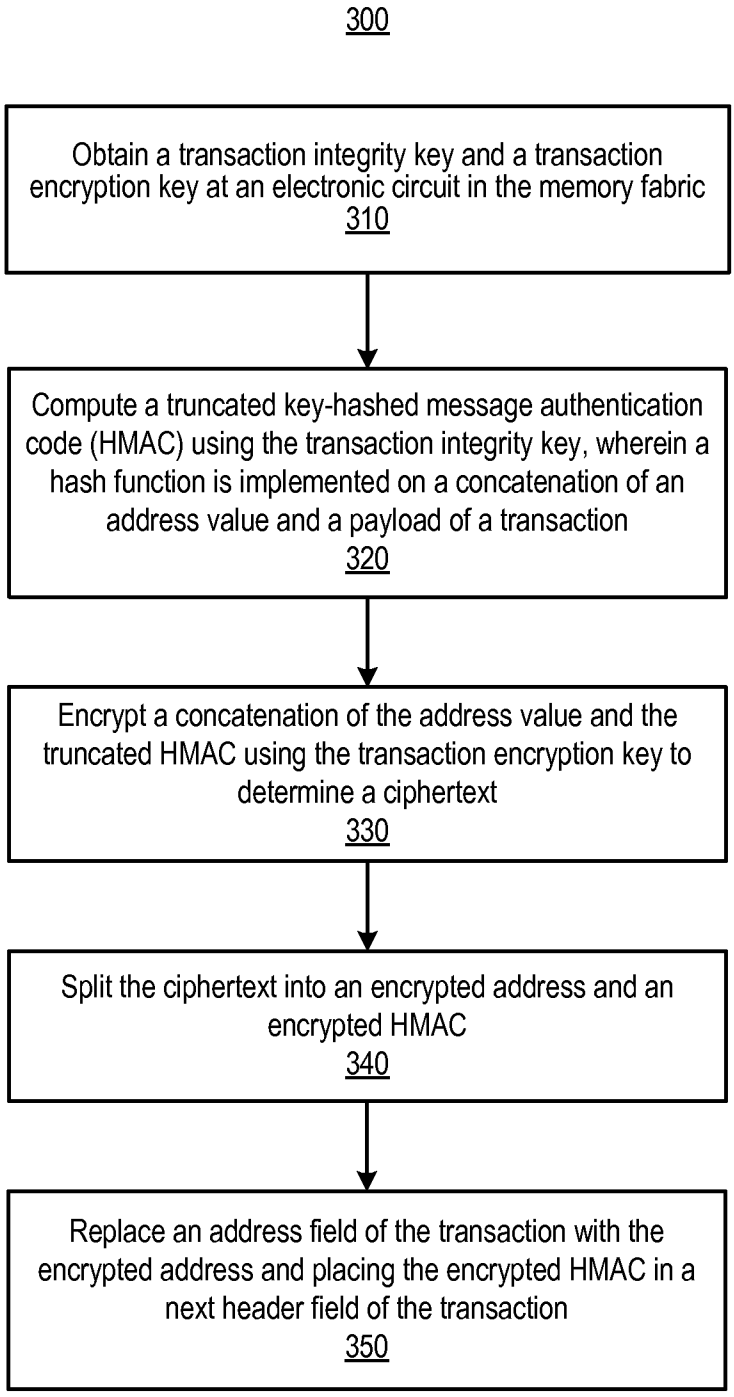
300
Obtain a transaction integrity key and a transaction encryption key at an electronic circuit in the memory fabric
310
Compute a truncated key-hashed message authentication code (HMAC) using the transaction integrity key, wherein a hash function is implemented on a concatenation of an address value and a payload of a transaction
320
Encrypt a concatenation of the address value and the truncated HMAC using the transaction encryption key to determine a ciphertext
330
Split the ciphertext into an encrypted address and an encrypted HMAC
340
Replace an address field of the transaction with the encrypted address and placing the encrypted HMAC in a next header field of the transaction
350

FIG. 3

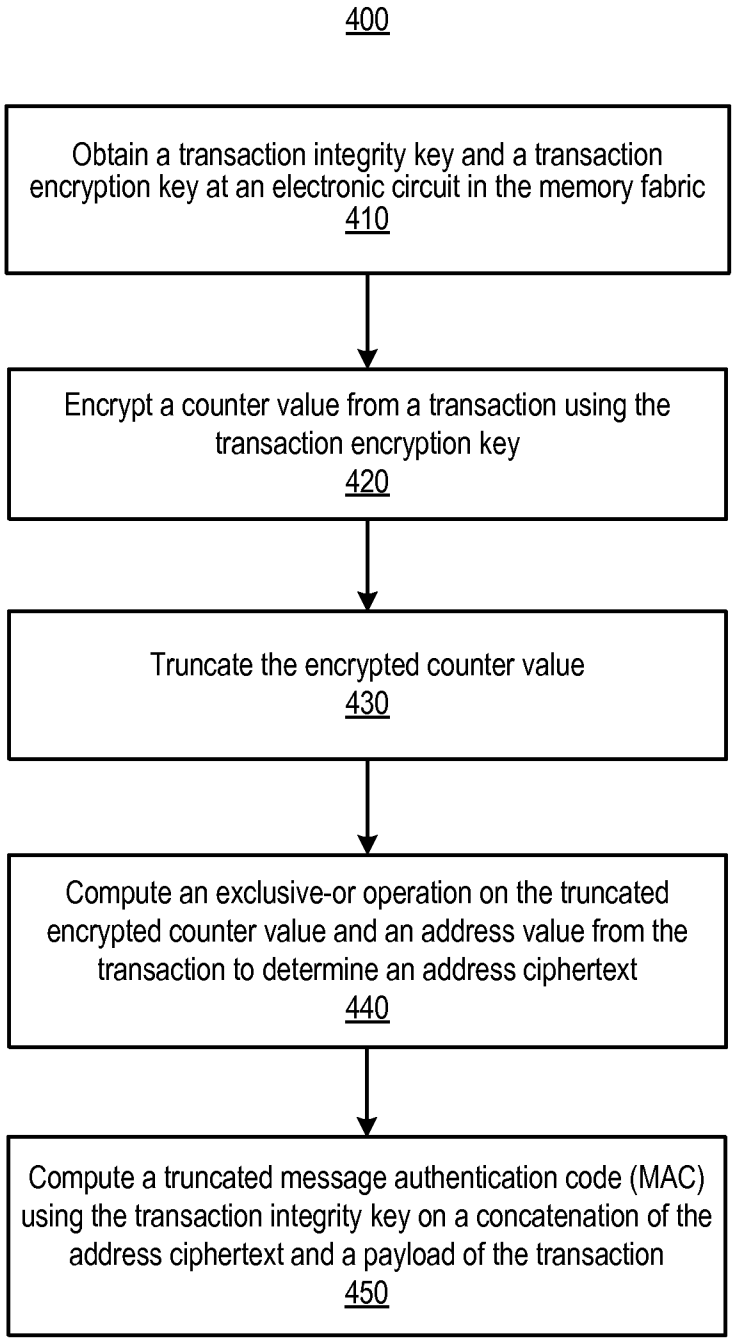
400
Obtain a transaction integrity key and a transaction encryption key at an electronic circuit in the memory fabric
410
Encrypt a counter value from a transaction using the transaction encryption key
420
Truncate the encrypted counter value
430
Compute an exclusive-or operation on the truncated encrypted counter value and an address value from the transaction to determine an address ciphertext
440
Compute a truncated message authentication code (MAC) using the transaction integrity key on a concatenation of the address ciphertext and a payload of the transaction
450

FIG. 4

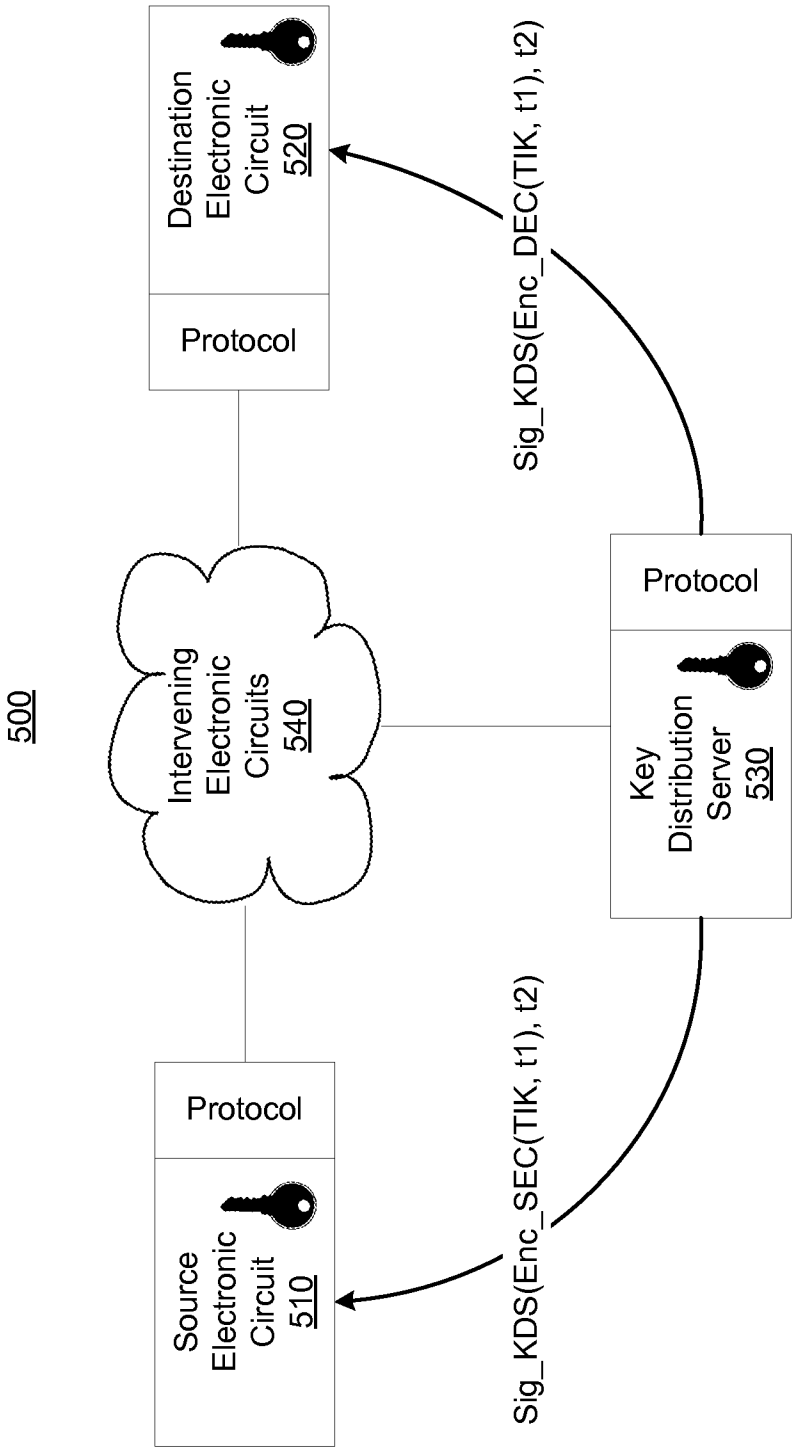
500
Destination Electronic Circuit 520
Protocol
Sig_KDS(Enc_DEC(TIK, t1), t2)
Intervening Electronic Circuits 540
Protocol
Key Distribution Server 530
Protocol
Source Electronic Circuit 510
Sig_KDS(Enc_SEC(TIK, t1), t2)

FIG. 5

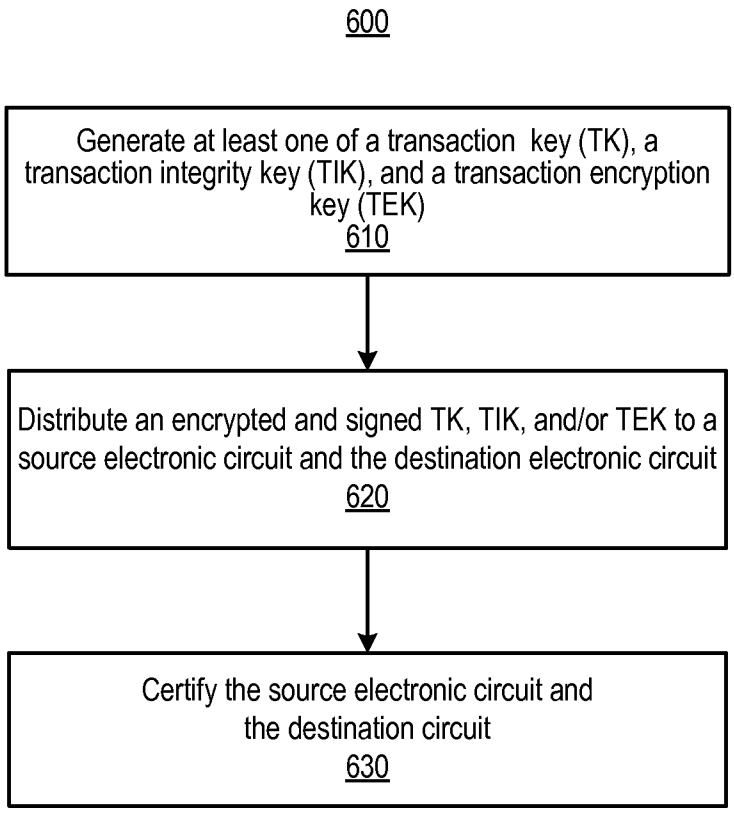
600
Generate at least one of a transaction key (TK), a transaction integrity key (TIK), and a transaction encryption key (TEK)
610
Distribute an encrypted and signed TK, TIK, and/or TEK to a source electronic circuit and the destination electronic circuit
620
Certify the source electronic circuit and the destination circuit
630

FIG. 6

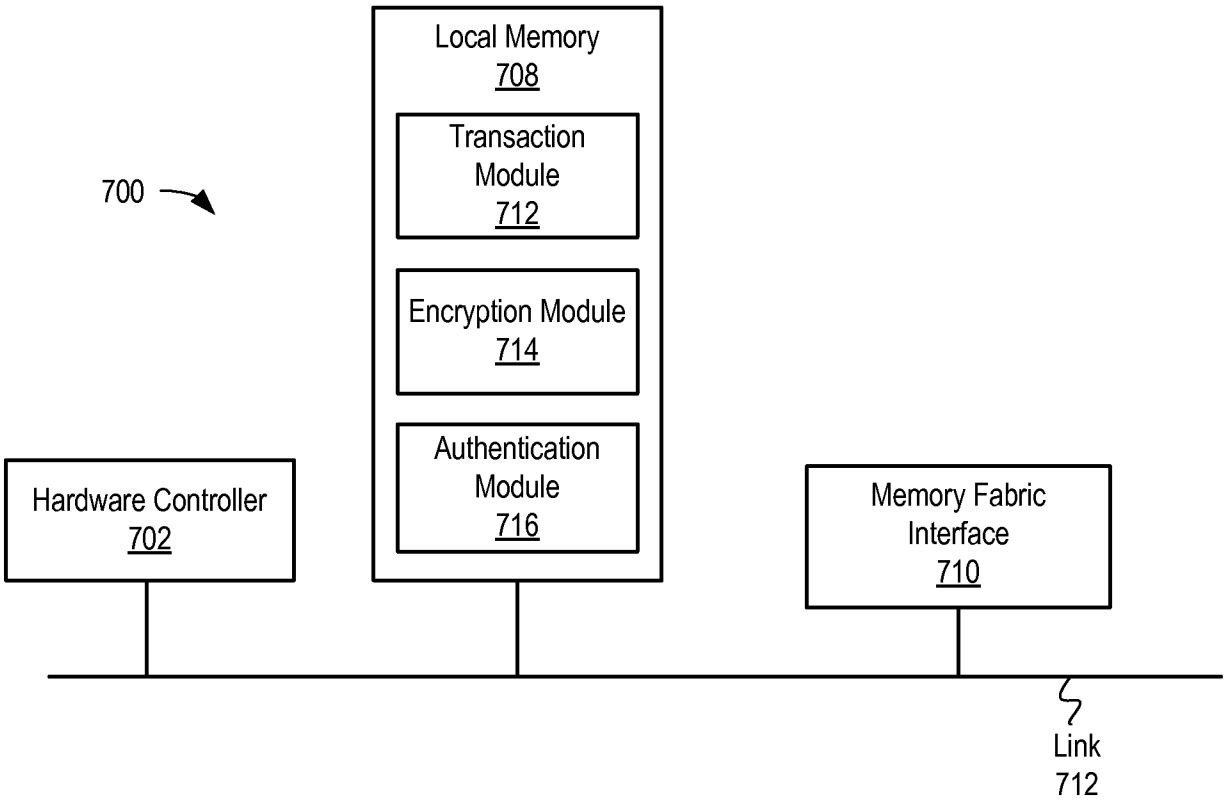
700
Local Memory
708
Transaction Module
712
Encryption Module
714
Authentication Module
716
Hardware Controller
702
Memory Fabric Interface
710
Link
712

FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2014/063174

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/79(2013.01)i, G06F 21/62(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)
G06F 21/79; G06F 21/24; G06F 17/30; G06F 12/14; H04L 29/06; H04L 9/00; G06F 21/62

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: integrity, encrypt, message authentication code, address

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2013-0117577 A1 (CPU TECHNOLOGY, INC. (77615)) 09 May 2013 See paragraphs [0037]-[0052], [0060]-[0061] and figures 2-6, 9.	1-15
A	US 2007-0130470 A1 (ROLF BLOM et al.) 07 June 2007 See paragraphs [0020]-[0047], claims 1-6 and figures 2-3.	1-15
A	US 2013-0080790 A1 (GUILLAUME PEAN et al.) 28 March 2013 See paragraphs [0024]-[0026], [0031]-[0039], claims 1-11 and figures 2, 4A-4B.	1-15
A	US 2013-0191649 A1 (ADAM J. MUFF et al.) 25 July 2013 See abstract, paragraphs [0089]-[0102] and figures 8-12.	1-15
A	US 2014-0075189 A1 (QUALCOMM INCORPORATED) 13 March 2014 See abstract, claims 1, 5 and figures 6-11.	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	Date of mailing of the international search report
25 June 2015 (25.06.2015)	**29 June 2015 (29.06.2015)**

Name and mailing address of the ISA/KR
International Application Division
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
AHN, Jeong Hwan
Telephone No. +82-42-481-8440

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/US2014/063174

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2013-0117577 A1	09/05/2013	US 8843767 B2	23/09/2014
US 2007-0130470 A1	07/06/2007	EP 1958114 A2	20/08/2008
		JP 2009-517939 A	30/04/2009
		TW 200805978 A	16/01/2008
		US 7681050 B2	16/03/2010
		WO 2007-062941 A2	07/06/2007
		WO 2007-062941 A3	26/07/2007
US 2013-0080790 A1	28/03/2013	US 8726037 B2	13/05/2014
US 2013-0191649 A1	25/07/2013	CN 103218572 A	24/07/2013
		DE 102013200161 A1	25/07/2013
		GB 2500458 A	25/09/2013
		GB 2500458 B	31/12/2014
		US 8751830 B2	10/06/2014
US 2014-0075189 A1	13/03/2014	KR 10-2015-0052276 A	13/05/2015
		US 2014-0071850 A1	13/03/2014
		US 2014-0071881 A1	13/03/2014
		US 2014-0071882 A1	13/03/2014
		US 2014-0071883 A1	13/03/2014
		WO 2014-039266 A1	13/03/2014
		WO 2014-039273 A1	13/03/2014
		WO 2014-039276 A1	13/03/2014
		WO 2014-039279 A1	13/03/2014
		WO 2014-039280 A2	13/03/2014