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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 February 2022 (24.02.2022)

WIPO | PCT

(10) International Publication Number
WO 2022/036615 A1

(51) International Patent Classification:
G06F 21/62 (2013.01) *H04L 9/12* (2006.01)

(21) International Application Number:
PCT/CN2020/110148

(22) International Filing Date:
20 August 2020 (20.08.2020)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, California 95054 (US).

(72) Inventors; and

(71) Applicants *(for BZ only)*: **CUI, Bo** [CN/CN]; No. 880 Zi Xing Road, Shanghai Zizhu Science Park, Shanghai 200241 (CN). **WEI, Changzheng** [CN/CN]; No. 880 Zi Xing Road, Shanghai Zizhu Science Park, Shanghai 200241 (CN). **SMITH, Ned** [US/US]; 375 SW Delta Drive, Beaverton, Oregon 97006 (US). **YU, Ping** [CN/CN]; No. 880 Zi Xing Road, Shanghai Zizhu Science Park, Shanghai 200241 (CN).

(74) Agent: **SHANGHAI PATENT & TRADEMARK LAW OFFICE, LLC**; 435 Guiping Road, Shanghai 200233 (CN).

(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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, 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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, 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, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

(54) **Title:** DEVICE CHANNEL PROTECTION MECHANISM

(57) **Abstract:** An apparatus to protect memory data transmitted over a device channel is disclosed. The apparatus includes a memory device to store encrypted data, a central processing unit (CPU), including a memory controller to control access to the memory device and a device coupled to the CPU via a communications channel, to receive encrypted memory data, wherein the device includes a cryptographic engine to decrypt the encrypted memory data and a device key table comprising a first set of encryption keys to decrypt the encrypted memory data.

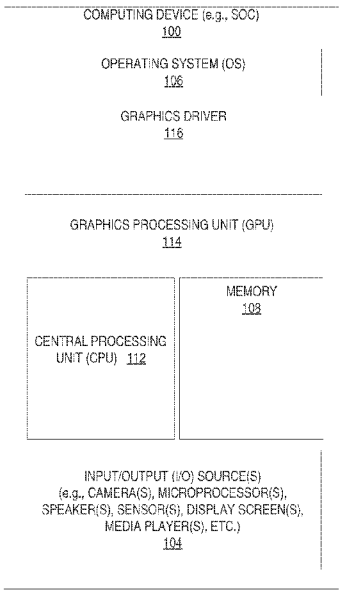

FIG. 1

WO 2022/036615 A1

[Continued on next page]

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

DEVICE CHANNEL PROTECTION MECHANISM

BACKGROUND OF THE DESCRIPTION

Trusted computing platforms have defined secure boot and device authentication that provide security for hardware components included in the platform. However, unprotected communication channels and buses provide system vulnerabilities to snooping attackers. Further, an unprotected bus may enable physical/side channel attacks on the platform security, including fuse injection/alteration attacks on secure boot, or secure bootstrapping of processors.

BRIEF DESCRIPTION OF THE DRAWINGS

So that the manner in which the above recited features can be understood in detail, a more particular description, briefly summarized above, may be had by reference to embodiments, some of which are illustrated in the appended drawings. It is to be noted, however, that the appended drawings illustrate only typical embodiments and are therefore not to be considered limiting of its scope, for the disclosure may admit to other equally effective embodiments.

Figure 1 illustrates one embodiment of a computing device.

Figures 2A&2B illustrate embodiments of a platform.

Figure 3 illustrates another embodiment of a platform.

Figure 4A is a flow diagram illustrating one embodiment of a process for establishing a secure connection.

Figure 4B illustrates another embodiment of the process for establishing a secure connection.

Figures 5A&5B are sequence diagrams illustrating embodiments for performing page encryption key synchronization.

Figure 6 illustrates one embodiment of a schematic diagram of an illustrative electronic computing device.

DETAILED DESCRIPTION

In the following description, numerous specific details are set forth to provide a more thorough understanding. However, it will be apparent to one of skill in the art that the embodiments may be practiced without one or more of these specific details. In other instances, well-known features have not been described in order to avoid obscuring the embodiments.

In embodiments, a mechanism is provided to protect memory data transmitted over a device channel. In such embodiments a peripheral device includes a device key table to store encryption keys shared with a memory controller. The encryption keys are implemented to encrypt/decrypt memory data transmitted between the peripheral device and a memory device via a communications channel.

References to "one embodiment", "an embodiment", "example embodiment", "various embodiments", etc., indicate that the embodiment(s) so described may include particular features, structures, or characteristics, but not every embodiment necessarily includes the particular features, structures, or characteristics. Further, some embodiments may have some, all, or none of the features described for other embodiments.

In the following description and claims, the term "coupled" along with its derivatives, may be used. "Coupled" is used to indicate that two or more elements co-operate or interact with each other, but they may or may not have intervening physical or electrical components between them.

As used in the claims, unless otherwise specified, the use of the ordinal adjectives "first", "second", "third", etc., to describe a common element, merely indicate that different instances of like elements are being referred to, and are not intended to imply that the elements so described must be in a given sequence, either temporally, spatially, in ranking, or in any other manner.

Figure 1 illustrates one embodiment of a computing device 100. According to one embodiment, computing device 100 comprises a computer platform hosting an integrated circuit ("IC"), such as a system on a chip ("SoC" or "SOC"), integrating various hardware and/or software components of computing device 100 on a single chip. As illustrated, in one embodiment, computing device 100 may include any number and type of hardware and/or software components, such as (without limitation) graphics processing unit 114 ("GPU" or simply "graphics processor"), graphics driver 116 (also

referred to as “GPU driver”, “graphics driver logic”, “driver logic”, user-mode driver (UMD), UMD, user-mode driver framework (UMDF), UMDF, or simply “driver”), central processing unit 112 (“CPU” or simply “application processor”), memory 108, network devices, drivers, or the like, as well as input/output (I/O) sources 104, such as touchscreens, touch panels, touch pads, virtual or regular keyboards, virtual or regular mice, ports, connectors, etc. Computing device 100 may include operating system (OS) 106 serving as an interface between hardware and/or physical resources of computing device 100 and a user.

It is to be appreciated that a lesser or more equipped system than the example described above may be preferred for certain implementations. Therefore, the configuration of computing device 100 may vary from implementation to implementation depending upon numerous factors, such as price constraints, performance requirements, technological improvements, or other circumstances.

Embodiments may be implemented as any or a combination of: one or more microchips or integrated circuits interconnected using a parentboard, hardwired logic, software stored by a memory device and executed by a microprocessor, firmware, an application specific integrated circuit (ASIC), and/or a field programmable gate array (FPGA). The terms "logic", “module”, “component”, “engine”, and “mechanism” may include, by way of example, software or hardware and/or a combination thereof, such as firmware.

Embodiments may be implemented using one or more memory chips, controllers, CPUs (Central Processing Unit), microchips or integrated circuits interconnected using a motherboard, an application specific integrated circuit (ASIC), and/or a field programmable gate array (FPGA). The term "logic" may include, by way of example, software or hardware and/or combinations of software and hardware.

Figure 2A&2B illustrates embodiments of a platform 200 including a SOC 210 similar to computing device 100 discussed above. As shown in **Figure 2A**, platform 200 includes SOC 210 communicatively coupled to one or more software components 280 via CPU 112. Additionally, SOC 210 includes other computing device components (e.g., memory 108) coupled via a system fabric 205. In one embodiment, system fabric 205 comprises an integrated on-chip system fabric (IOSF) to provide a standardized on-die

interconnect protocol for coupling interconnect protocol (IP) agents 230 (e.g., IP agents 230A and 230B) within SOC 210. In such an embodiment, the interconnect protocol provides a standardized interface to enable third parties to design logic such as IP agents 230 to be incorporated in SOC 210.

According to embodiment, IP agents 230 may include general purpose processors (e.g., in-order or out-of-order cores), fixed function units, graphics processors, I/O controllers, display controllers, etc. In such an embodiment, each IP agent 230 includes a hardware interface 235 to provide standardization to enable the IP agent 230 to communicate with SOC 210 components. For example, in an embodiment in which IPA agent 230 is a third party visual processing unit (VPU), interface 235 provides a standardization to enable the VPU to access memory 108 via fabric 205.

SOC 210 also includes a security controller 240 that operates as a security engine to perform various security operations (e.g., security processing, cryptographic functions, etc.) for SOC 210. In one embodiment, security controller 240 comprises an IP agent 230 that is implemented to perform the security operations. Further, SOC 210 includes a non-volatile memory 250. Non-volatile memory 250 may be implemented as a Peripheral Component Interconnect Express (PCIe) storage drive, such as a solid state drives (SSD) or Non-Volatile Memory Express (NVMe) drives.

Figure 2B illustrates another embodiment of platform 200 including a component 260 coupled to SOC 210 via IP 230A. In one embodiment, IP 230A operates as a bridge, such as a PCIe root port, that connects component 260 to SOC 210. In this embodiment, component 260 may be implemented as a PCIe device (e.g., switch or endpoint) that includes a hardware interface 235 to enable component 260 to communicate with SOC 210 components.

As discussed above, communication channels (or buses) provide system vulnerabilities to platforms, such as platforms 200. For instance, memory encryption technologies provide enhanced memory channel security inside a memory controller. Thus, eavesdroppers have no chance to retrieve plain text data from DRAM bus or DRAM internal for data encrypted with an unexposed page key. However, data vulnerability appears on I/O buses in computing intensive scenarios, while moving data from DRAM to peripheral hardware accelerators (e.g., like GPU, digital signal processors

(DSPs), FPGA or crypto Accelerator. During the accelerator offload, data flowing on a PCIe is in plain text format (e.g., already decrypted), which provides an opportunity for attackers to snoop the content.

According to one embodiment, a device channel mechanism is provided to extend memory protection to support device I/O protections to ensure both platform integrity and the secure transmission of data over memory communication channels (e.g., PCIe bus). **Figure 3** illustrates an embodiment of a platform 300 including a PCIe device 320 coupled to memory 112 to access memory 108. In one embodiment, PCIe device 320 comprises a component coupled external to a secure platform via an interface, such as PCIe bus.

CPU 112 includes a root complex 330, translation agent 340, address translation and protection table 345 and memory controller 350. Root complex 330 generates transaction requests on behalf CPU 112 via PCIe bus. In one embodiment, root complex 330 includes one or more root ports 332 coupled to the PCIe bus. Translation agent 340 uses address translation and protection table 345 to perform virtual-physical address translations. Memory controller 350 manages the flow of data to and from memory 108. In one embodiment, memory 108 stores data in an encrypted format. Thus, memory controller 350 includes a page key table 352 that maintains a plurality of page keys that are implemented to encrypt/decrypt memory pages. In one embodiment, each page key is associated with a unique key identifier (keyID). Additionally, cryptographic (or crypto) engine 358 uses the page keys stored in page key table 352 to perform operations to encrypt or decrypt the memory pages.

PCIe device 310 includes an endpoint 320 having an address translation cache 322, device key table 324 and crypto engine 328. Address translation cache 322 (or device translation lookaside buffer (TLB)) stores virtual-physical address mapping, while device key table 324 maintains a shared copy of the page keys stored in page key table 352 to use for direct memory accesses (DMAs) implemented at a DMA engine 326. In one embodiment, device key table 324 is configured to access page-level physical memory encryption using a page key that is shared with page key table 352 in order to protect data flowing between CPU 112 and device 310. In a further embodiment, crypto

engine 328 is implemented to perform encryption or decryption operations using the page keys stored in page key table 352.

In a yet a further embodiment, a symmetric key (or shared key) is generated between CPU 112 and device 310 during a secure boot phase. Subsequently, device 310 securely synchronizes page keys with CPU 112. Once synchronized, the local copy of page keys stored at page key table 352 enables device 310 to directly access encrypted memory 108 pages without CPU involvement (e.g., for DMAs).

In one embodiment, encrypted DRAM accesses of memory 108 may be enabled by embedding the keyID within the high bits of a physical address. As a result, memory controller 350 performs encrypted DRAM accesses via page key table 352. In another embodiment, device 310 may perform a request without the keyID, thus bypassing page key table 352 for DRAM access. In this embodiment, device key table 324 is used to transform (e.g., decrypt) the encrypted.

According to one embodiment, device 310 communicates with root complex 330 to provide virtual-physical address translation services. In such an embodiment, device 310 generates subsequent DRAM accesses using a translated address returned from a root complex response. Subsequently, memory controller 350 requests the page encryption key. After page encryption key is synchronized (e.g., stored at device key table 324), all subsequent DRAM accesses by device 310 is protected by the synchronized encryption key.

In one embodiment, CPU 112 is booted into a trusted BIOS (e.g., via With BootGuard or similar secure boot technology) that enumerates all I/O devices connected within platform 300 after core component initialization completes. Subsequently, a secure channel is established between the components of the platform. **Figure 4A** is a flow diagram illustrating one embodiment of a process for establishing a secure connection.

At processing block 410, shared keys are provisioned to crypto engine 358 at memory controller 350 and crypto engine 328 at device 320. In one embodiment, CPU 112 exchanges key with a device 310 and negotiates a shared key after a bi-way authentication has been performed (e.g. in BIOS configuration). In such an embodiment, a shared key is stored within both page key table 352 and device key table 324 without an

explicit transfer over the PCIe channel. In a further embodiment, a shared key is used for page key table synchronization only between page key table 352 and device key table 324. In yet a further embodiment, the key is programed into page key table 352 via a write only register. In still a further embodiment, a different set of shared keys are generated for each device 310, and each device-to-device communication depends on a mutual authentication and has independent derived keys.

At processing block 420, memory encryption for DMA is enabled. In one embodiment, a device driver allocates one or more physical memory pages used for DMA and generates page encryption keys (page_keys) associated with the physical memory pages. In this embodiment, the encryption key is generated by the device driver via a hardware random number generator. The encryption key is provisioned into the memory controller once the device driver attributes the pages as Device Memory Page (DMP), which indicates that page key table is to by bypassed for accesses from PCIe devices. Memory accesses from the CPU are handled in a conventional manner using page key table (e.g., with an associated keyID in the physical address which configured in a memory management unit (MMU) TLB.

At processing block 430, the page key table and device key table are synchronized under protection of the shared key. According to one embodiment, the PCIe device retrieves page keys from memory controller within a secure channel. In a further embodiment, page key updates are during runtime in order to support dynamic memory allocation. In such an embodiment, synchronization of keys between CPU and the PCIe device is enabled.

At processing block 440, DRAM accesses are performed. As a result, PCIe device DRAM accesses are performed via a page key in the device key table key, with no plain text being transferred between the PCIe device and DRAM. In one embodiment, a descriptor setup (e.g., tx/rx ring setting) is protected by shared key during the secured boot phase. In a further embodiment, runtime data (e.g., descriptor/data) located in DRAM is protected via the use of the page key table and the device key table. **Figure 4B** provides a graphic illustration of the process performed in **Figure 4A**.

As discussed above, each encryption key in the page key table is encrypted with a shared_key prior to being transmitted over the PCIe bus, where it is decrypted at a device

crypto engine and stored in the device key table. In one embodiment, key table synchronization is initiated by PCIe device in order to request a new entry upon a determination that a device key lookup has failed. In a further embodiment, a key entry in the device key table may be revoked base on capacity and usage of the device key table. In such an embodiment, keys may be revoked based on a revoke/replace algorithm (e.g., least recently used (LRU) defined by the PCIe device. Upon a determination that key synchronization has failed, the operating system reports the failure to a user (e.g., site administrator) via a user interface.

Figures 5A&5B are sequence diagrams illustrating embodiments for performing page encryption key synchronization. **Figure 5A** illustrates one embodiment of an initialization of CPU and device memory encryption. At step S1, a device driver (or operating system) provisions a page key at the memory controller. At step S2, a keyID is transmitted from the memory controller to the device driver. At step S3, memory encryption (ME) is enabled at a memory management unit (MMU) with the CPU, which performs virtual memory management for the CPU. At step S4, device memory encryption (DME) is enabled at an input-output memory management unit (IOMMU) with the CPU. The IOMMU performs virtual memory management for the PCIe device.

Figure 5B illustrates one embodiment of page key synchronization during a DMA. At step S5, a DMA is initiated by the driver at the DMA engine. At step S6, the DMA engine performs a physical address lookup of the device TLB. At step S7, a request from the device is transmitted to the IOMMU. At step S8, a page key request is transmitted to the memory controller. At step S9, encrypted page keys are received from the memory controller. At step S10, the DMA engine transmits a physical page read request to the memory controller. At step S11, the encrypted read data is received . At step S12, the device crypto engine decrypts the data. St step S13, the decrypted read data is forwarded to the DMA engine. At step S14, the DMA engine signals the completion of the DMA.

Figure 6 is a schematic diagram of an illustrative electronic computing device to enable enhanced protection against adversarial attacks according to some embodiments. In some embodiments, the computing device 600 includes one or more processors 610 including one or more processors cores 618 and a TEE 664, the TEE including a machine

learning service enclave (MLSE) 680. In some embodiments, the computing device 600 includes a hardware accelerator 668, the hardware accelerator including a cryptographic engine 682 and a machine learning model 684. In some embodiments, the computing device is to provide enhanced protections against ML adversarial attacks, as provided in **Figures 1-5B**.

The computing device 600 may additionally include one or more of the following: cache 662, a graphical processing unit (GPU) 612 (which may be the hardware accelerator in some implementations), a wireless input/output (I/O) interface 620, a wired I/O interface 630, memory circuitry 640, power management circuitry 650, non-transitory storage device 660, and a network interface 670 for connection to a network 672. The following discussion provides a brief, general description of the components forming the illustrative computing device 600. Example, non-limiting computing devices 600 may include a desktop computing device, blade server device, workstation, or similar device or system.

In embodiments, the processor cores 618 are capable of executing machine-readable instruction sets 614, reading data and/or instruction sets 614 from one or more storage devices 660 and writing data to the one or more storage devices 660. Those skilled in the relevant art will appreciate that the illustrated embodiments as well as other embodiments may be practiced with other processor-based device configurations, including portable electronic or handheld electronic devices, for instance smartphones, portable computers, wearable computers, consumer electronics, personal computers ("PCs"), network PCs, minicomputers, server blades, mainframe computers, and the like.

The processor cores 618 may include any number of hardwired or configurable circuits, some or all of which may include programmable and/or configurable combinations of electronic components, semiconductor devices, and/or logic elements that are disposed partially or wholly in a PC, server, or other computing system capable of executing processor-readable instructions.

The computing device 600 includes a bus or similar communications link 616 that communicably couples and facilitates the exchange of information and/or data between various system components including the processor cores 618, the cache 662, the graphics processor circuitry 612, one or more wireless I/O interfaces 620, one or more

wired I/O interfaces 630, one or more storage devices 660, and/or one or more network interfaces 670. The computing device 600 may be referred to in the singular herein, but this is not intended to limit the embodiments to a single computing device 600, since in certain embodiments, there may be more than one computing device 600 that incorporates, includes, or contains any number of communicably coupled, collocated, or remote networked circuits or devices.

The processor cores 618 may include any number, type, or combination of currently available or future developed devices capable of executing machine-readable instruction sets.

The processor cores 618 may include (or be coupled to) but are not limited to any current or future developed single- or multi-core processor or microprocessor, such as: on or more systems on a chip (SOCs); central processing units (CPUs); digital signal processors (DSPs); graphics processing units (GPUs); application-specific integrated circuits (ASICs), programmable logic units, field programmable gate arrays (FPGAs), and the like. Unless described otherwise, the construction and operation of the various blocks shown in **Figure 6** are of conventional design. Consequently, such blocks need not be described in further detail herein, as they will be understood by those skilled in the relevant art. The bus 616 that interconnects at least some of the components of the computing device 600 may employ any currently available or future developed serial or parallel bus structures or architectures.

The system memory 640 may include read-only memory ("ROM") 642 and random access memory ("RAM") 646. A portion of the ROM 642 may be used to store or otherwise retain a basic input/output system ("BIOS") 644. The BIOS 644 provides basic functionality to the computing device 600, for example by causing the processor cores 618 to load and/or execute one or more machine-readable instruction sets 614. In embodiments, at least some of the one or more machine-readable instruction sets 614 cause at least a portion of the processor cores 618 to provide, create, produce, transition, and/or function as a dedicated, specific, and particular machine, for example a word processing machine, a digital image acquisition machine, a media playing machine, a gaming system, a communications device, a smartphone, or similar.

The computing device 600 may include at least one wireless input/output (I/O) interface 620. The at least one wireless I/O interface 620 may be communicably coupled to one or more physical output devices 622 (tactile devices, video displays, audio output devices, hardcopy output devices, etc.). The at least one wireless I/O interface 620 may communicably couple to one or more physical input devices 624 (pointing devices, touchscreens, keyboards, tactile devices, etc.). The at least one wireless I/O interface 620 may include any currently available or future developed wireless I/O interface. Example wireless I/O interfaces include, but are not limited to: BLUETOOTH®, near field communication (NFC), and similar.

The computing device 600 may include one or more wired input/output (I/O) interfaces 630. The at least one wired I/O interface 630 may be communicably coupled to one or more physical output devices 622 (tactile devices, video displays, audio output devices, hardcopy output devices, etc.). The at least one wired I/O interface 630 may be communicably coupled to one or more physical input devices 624 (pointing devices, touchscreens, keyboards, tactile devices, etc.). The wired I/O interface 630 may include any currently available or future developed I/O interface. Example wired I/O interfaces include, but are not limited to: universal serial bus (USB), IEEE 1394 (“FireWire”), and similar.

The computing device 600 may include one or more communicably coupled, non-transitory, data storage devices 660. The data storage devices 660 may include one or more hard disk drives (HDDs) and/or one or more solid-state storage devices (SSDs). The one or more data storage devices 660 may include any current or future developed storage appliances, network storage devices, and/or systems. Non-limiting examples of such data storage devices 660 may include, but are not limited to, any current or future developed non-transitory storage appliances or devices, such as one or more magnetic storage devices, one or more optical storage devices, one or more electro-resistive storage devices, one or more molecular storage devices, one or more quantum storage devices, or various combinations thereof. In some implementations, the one or more data storage devices 660 may include one or more removable storage devices, such as one or more flash drives, flash memories, flash storage units, or similar appliances or devices capable of communicable coupling to and decoupling from the computing device 600.

The one or more data storage devices 660 may include interfaces or controllers (not shown) communicatively coupling the respective storage device or system to the bus 616. The one or more data storage devices 660 may store, retain, or otherwise contain machine-readable instruction sets, data structures, program modules, data stores, databases, logical structures, and/or other data useful to the processor cores 618 and/or graphics processor circuitry 612 and/or one or more applications executed on or by the processor cores 618 and/or graphics processor circuitry 612. In some instances, one or more data storage devices 660 may be communicably coupled to the processor cores 618, for example via the bus 616 or via one or more wired communications interfaces 630 (e.g., Universal Serial Bus or USB); one or more wireless communications interfaces 620 (e.g., Bluetooth®, Near Field Communication or NFC); and/or one or more network interfaces 670 (IEEE 802.3 or Ethernet, IEEE 802.11, or Wi-Fi®, etc.).

Processor-readable instruction sets 614 and other programs, applications, logic sets, and/or modules may be stored in whole or in part in the system memory 640. Such instruction sets 614 may be transferred, in whole or in part, from the one or more data storage devices 660. The instruction sets 614 may be loaded, stored, or otherwise retained in system memory 640, in whole or in part, during execution by the processor cores 618 and/or graphics processor circuitry 612.

The computing device 600 may include power management circuitry 650 that controls one or more operational aspects of the energy storage device 652. In embodiments, the energy storage device 652 may include one or more primary (i.e., non-rechargeable) or secondary (i.e., rechargeable) batteries or similar energy storage devices. In embodiments, the energy storage device 652 may include one or more supercapacitors or ultracapacitors. In embodiments, the power management circuitry 650 may alter, adjust, or control the flow of energy from an external power source 654 to the energy storage device 652 and/or to the computing device 600. The power source 654 may include, but is not limited to, a solar power system, a commercial electric grid, a portable generator, an external energy storage device, or any combination thereof.

For convenience, the processor cores 618, the graphics processor circuitry 612, the wireless I/O interface 620, the wired I/O interface 630, the storage device 660, and the network interface 670 are illustrated as communicatively coupled to each other via

the bus 616, thereby providing connectivity between the above-described components. In alternative embodiments, the above-described components may be communicatively coupled in a different manner than illustrated in **Figure 6**. For example, one or more of the above-described components may be directly coupled to other components, or may be coupled to each other, via one or more intermediary components (not shown). In another example, one or more of the above-described components may be integrated into the processor cores 618 and/or the graphics processor circuitry 612. In some embodiments, all or a portion of the bus 616 may be omitted and the components are coupled directly to each other using suitable wired or wireless connections.

Embodiments may be provided, for example, as a computer program product which may include one or more machine-readable media having stored thereon machine-executable instructions that, when executed by one or more machines such as a computer, network of computers, or other electronic devices, may result in the one or more machines carrying out operations in accordance with embodiments described herein. A machine-readable medium may include, but is not limited to, floppy diskettes, optical disks, CD-ROMs (Compact Disc-Read Only Memories), and magneto-optical disks, ROMs, RAMs, EPROMs (Erasable Programmable Read Only Memories), EEPROMs (Electrically Erasable Programmable Read Only Memories), magnetic or optical cards, flash memory, or other type of media/machine-readable medium suitable for storing machine-executable instructions.

Moreover, embodiments may be downloaded as a computer program product, wherein the program may be transferred from a remote computer (e.g., a server) to a requesting computer (e.g., a client) by way of one or more data signals embodied in and/or modulated by a carrier wave or other propagation medium via a communication link (e.g., a modem and/or network connection).

Throughout the document, term "user" may be interchangeably referred to as "viewer", "observer", "speaker", "person", "individual", "end-user", and/or the like. It is to be noted that throughout this document, terms like "graphics domain" may be referenced interchangeably with "graphics processing unit", "graphics processor", or simply "GPU" and similarly, "CPU domain" or "host domain" may be referenced

interchangeably with "computer processing unit", "application processor", or simply "CPU".

It is to be noted that terms like "node", "computing node", "server", "server device", "cloud computer", "cloud server", "cloud server computer", "machine", "host machine", "device", "computing device", "computer", "computing system", and the like, may be used interchangeably throughout this document. It is to be further noted that terms like "application", "software application", "program", "software program", "package", "software package", and the like, may be used interchangeably throughout this document. Also, terms like "job", "input", "request", "message", and the like, may be used interchangeably throughout this document.

In various implementations, the computing device may be a laptop, a netbook, a notebook, an ultrabook, a smartphone, a tablet, a personal digital assistant (PDA), an ultra mobile PC, a mobile phone, a desktop computer, a server, a set-top box, an entertainment control unit, a digital camera, a portable music player, or a digital video recorder. The computing device may be fixed, portable, or wearable. In further implementations, the computing device may be any other electronic device that processes data or records data for processing elsewhere.

The drawings and the forgoing description give examples of embodiments. Those skilled in the art will appreciate that one or more of the described elements may well be combined into a single functional element. Alternatively, certain elements may be split into multiple functional elements. Elements from one embodiment may be added to another embodiment. For example, orders of processes described herein may be changed and are not limited to the manner described herein. Moreover, the actions of any flow diagram need not be implemented in the order shown; nor do all of the acts necessarily need to be performed. Also, those acts that are not dependent on other acts may be performed in parallel with the other acts. The scope of embodiments is by no means limited by these specific examples. Numerous variations, whether explicitly given in the specification or not, such as differences in structure, dimension, and use of material, are possible. The scope of embodiments is at least as broad as given by the following claims.

Embodiments may be provided, for example, as a computer program product which may include one or more transitory or non-transitory machine-readable storage

media having stored thereon machine-executable instructions that, when executed by one or more machines such as a computer, network of computers, or other electronic devices, may result in the one or more machines carrying out operations in accordance with embodiments described herein. A machine-readable medium may include, but is not limited to, floppy diskettes, optical disks, CD-ROMs (Compact Disc-Read Only Memories), and magneto-optical disks, ROMs, RAMs, EPROMs (Erasable Programmable Read Only Memories), EEPROMs (Electrically Erasable Programmable Read Only Memories), magnetic or optical cards, flash memory, or other type of media/machine-readable medium suitable for storing machine-executable instructions.

Some embodiments pertain to Example 1 that includes an apparatus to protect memory data transmitted over a device channel, comprising a memory device to store encrypted data, a central processing unit (CPU), including a memory controller to control access to the memory device and a device coupled to the CPU via a communications channel, to receive encrypted memory data, wherein the device includes a cryptographic engine to decrypt the encrypted memory data and a device key table comprising a first set of encryption keys to decrypt the encrypted memory data.

Example 2 includes the subject matter of Example 1, wherein the memory controller comprises a page key table comprising a second set of encryption keys to decrypt the encrypted memory data, wherein the second set of encryption keys comprises one or more encryption keys that are shared as the first set of encryption keys.

Example 3 includes the subject matter of Examples 1 and 2, wherein sharing one or more encryption keys comprises the CPU provisioning a shared key to the memory controller and the device.

Example 4 includes the subject matter of Examples 1-3, wherein the shared key is stored in the device key table without transmitting via the communications channel.

Example 5 includes the subject matter of Examples 1-4, wherein sharing the encryption key further comprises enabling memory encryption for direct memory accesses from the device.

Example 6 includes the subject matter of Examples 1-5, wherein sharing the encryption key further comprises enabling memory encryption for direct memory accesses from the device.

Example 7 includes the subject matter of Examples 1-6, wherein sharing an encryption key comprises synchronizing the first set of encryption keys and the second set of encryption keys using the shared key.

Example 8 includes the subject matter of Examples 1-7, wherein the device further comprises a translation lookaside buffer to store virtual-physical address translations and a direct memory access (DMA) engine to perform DMAs of the memory device.

Example 9 includes the subject matter of Examples 1-8, wherein the device transmits a page key request for an encryption key to the memory controller.

Example 10 includes the subject matter of Examples 1-9, wherein the device receives the key from the memory controller.

Some embodiments pertain to Example 11 that includes a method to protect memory data transmitted over a device channel, comprising provisioning a shared key to a memory controller at a central processing unit (CPU) and a device peripheral device coupled to the CPU via a communications channel, enabling memory encryption for direct memory access (DMA) of a memory device coupled to the memory controller from the peripheral device and synchronizing a first set of encryption keys stored at a page key table in the memory controller with a second set of encryption keys stored at a device key table in the peripheral device.

Example 12 includes the subject matter of Example 11, further comprising performing a DMA of the memory device from the peripheral device, including requesting access to one or more encrypted pages in the memory device, receiving the encrypted pages via the communications channel and decrypting the pages using the second set of encryption keys.

Example 13 includes the subject matter of Examples 11 and 12, wherein the shared key is stored in the device key table without transmitting via the communications channel.

Example 14 includes the subject matter of Examples 11-13, wherein enabling memory encryption for DMA comprises allocating one or more physical pages of memory to be used for DMA and generating a page encryption key associated with each of the one or more physical pages of memory.

Example 15 includes the subject matter of Examples 11-14, wherein synchronizing the first set of encryption keys and the second set of encryption keys comprises receiving the first set of encryption keys at the device key table via a secure communications channel.

Some embodiments pertain to Example 16 that includes at least one computer readable medium having instructions stored thereon, which when executed by one or more processors, cause the processors to provision a shared key to a memory controller at a central processing unit (CPU) and a device peripheral device coupled to the CPU via a communications channel, enable memory encryption for direct memory access (DMA) of a memory device coupled to the memory controller from the peripheral device and synchronize a first set of encryption keys stored at a page key table in the memory controller with a second set of encryption keys stored at a device key table in the peripheral device.

Example 17 includes the subject matter of Example 16, having instructions stored thereon, which when executed by the one or more processors, further cause the processors to perform a DMA of the memory device from the peripheral device, including requesting access to one or more encrypted pages in the memory device, receiving the encrypted pages via the communications channel and decrypting the pages using the second set of encryption keys.

Example 18 includes the subject matter of Examples 16 and 17, wherein the shared key is stored in the device key table without transmitting via the communications channel.

Example 19 includes the subject matter of Examples 16-18, wherein enabling memory encryption for DMA comprises allocating one or more physical pages of memory to be used for DMA and generating a page encryption key associated with each of the one or more physical pages of memory.

Example 20 includes the subject matter of Examples 16-19, wherein synchronizing the first set of encryption keys and the second set of encryption keys comprises receiving the first set of encryption keys at the device key table via a secure communications channel.

The embodiments of the examples have been described above with reference to specific embodiments. Persons skilled in the art, however, will understand that various modifications and changes may be made thereto without departing from the broader spirit and scope as set forth in the appended claims. The foregoing description and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is:

1. An apparatus to protect memory data transmitted over a device channel, comprising:

a memory device to store encrypted data;

a central processing unit (CPU), including a memory controller to control access to the memory device; and

a device coupled to the CPU via a communications channel, to receive encrypted memory data, wherein the device includes:

a cryptographic engine to decrypt the encrypted memory data; and

a device key table comprising a first set of encryption keys to decrypt the encrypted memory data.

2. The apparatus of claim 1, wherein the memory controller comprises a page key table comprising a second set of encryption keys to decrypt the encrypted memory data, wherein the second set of encryption keys comprises one or more encryption keys that are shared as the first set of encryption keys.

3. The apparatus of claim 2, wherein sharing one or more encryption keys comprises the CPU provisioning a shared key to the memory controller and the device.

4. The apparatus of claim 3, wherein the shared key is stored in the device key table without transmitting via the communications channel.

5. The apparatus of claim 3, wherein sharing the encryption key further comprises enabling memory encryption for direct memory accesses (DMAs) from the device.

6. The apparatus of claim 5, wherein sharing an encryption key comprises synchronizing the first set of encryption keys and the second set of encryption keys using the shared key.

7. The apparatus of claim 6, wherein the second set of encryption keys is received at the device key table via a secure communications channel.

8. The apparatus of claim 5, wherein the device further comprises:

a translation lookaside buffer to store virtual-physical address translations; and

DMA engine to perform the DMAs of the memory device.

9. The apparatus of claim 8, wherein the device transmits a page key request for an encryption key to the memory controller.

10. The apparatus of claim 9, wherein the device receives the page key from the memory controller.

11. A method to protect memory data transmitted over a device channel, comprising:

provisioning a shared key to a memory controller at a central processing unit (CPU) and a peripheral device coupled to the CPU via a communications channel;

enabling memory encryption for direct memory access (DMA) of a memory device coupled to the memory controller from the peripheral device; and

synchronizing a first set of encryption keys stored at a page key table in the memory controller with a second set of encryption keys stored at a device key table in the peripheral device.

12. The method of claim 11, further comprising performing a DMA of the memory device from the peripheral device, including:

requesting access to one or more encrypted pages in the memory device;

receiving the encrypted pages via the communications channel; and

decrypting the pages using the second set of encryption keys.

13. The method of claim 11, wherein the shared key is stored in the device key table without transmitting via the communications channel.

14. The method of claim 11, wherein enabling memory encryption for DMA comprises:

allocating one or more physical pages of memory to be used for DMA; and

generating a page encryption key associated with each of the one or more physical pages of memory.

15. The method of claim 11, wherein synchronizing the first set of encryption keys and the second set of encryption keys comprises receiving the first set of encryption keys at the device key table via a secure communications channel.

16. At least one computer readable medium having instructions stored thereon, which when executed by one or more processors, cause the processors to:

provision a shared key to a memory controller at a central processing unit (CPU) and a peripheral device coupled to the CPU via a communications channel;

enable memory encryption for direct memory access (DMA) of a memory device coupled to the memory controller from the peripheral device; and

synchronize a first set of encryption keys stored at a page key table in the memory controller with a second set of encryption keys stored at a device key table in the peripheral device.

17. The computer readable medium of claim 16, having instructions stored thereon, which when executed by the one or more processors, further cause the processors to perform a DMA of the memory device from the peripheral device, including:

requesting access to one or more encrypted pages in the memory device;

receiving the encrypted pages via the communications channel; and

decrypting the pages using the second set of encryption keys.

18. The computer readable medium of claim 16, wherein the shared key is stored in the device key table without transmitting via the communications channel.

19. The computer readable medium of claim 16, wherein enabling memory encryption for DMA comprises:

allocating one or more physical pages of memory to be used for DMA; and

generating a page encryption key associated with each of the one or more physical pages of memory.

20. The computer readable medium of claim 16, wherein synchronizing the first set of encryption keys and the second set of encryption keys comprises receiving the first set of encryption keys at the device key table via a secure communications channel.

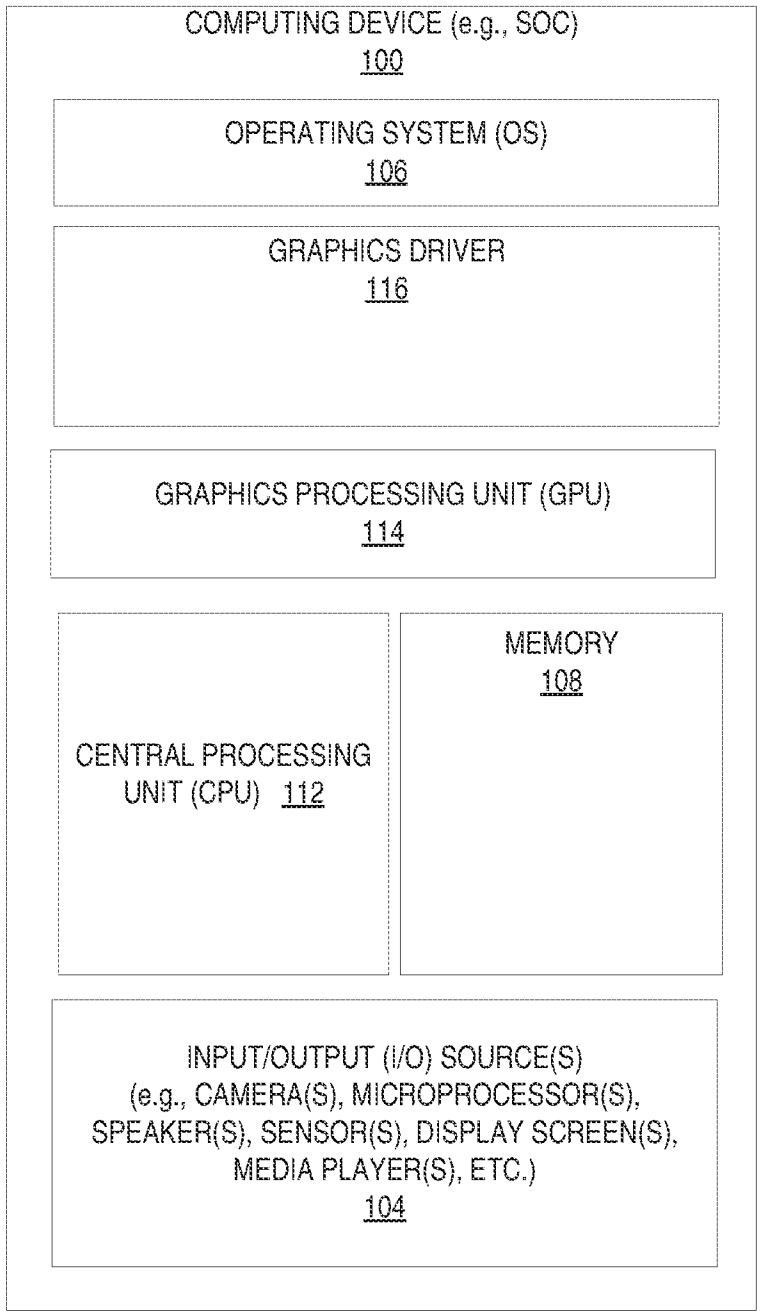
COMPUTING DEVICE (e.g., SOC)
100
OPERATING SYSTEM (OS)
106
GRAPHICS DRIVER
116
GRAPHICS PROCESSING UNIT (GPU)
114
CENTRAL PROCESSING UNIT (CPU) 112
MEMORY
108
INPUT/OUTPUT (I/O) SOURCE(S)
(e.g., CAMERA(S), MICROPROCESSOR(S), SPEAKER(S), SENSOR(S), DISPLAY SCREEN(S), MEDIA PLAYER(S), ETC.)
104

FIG. 1

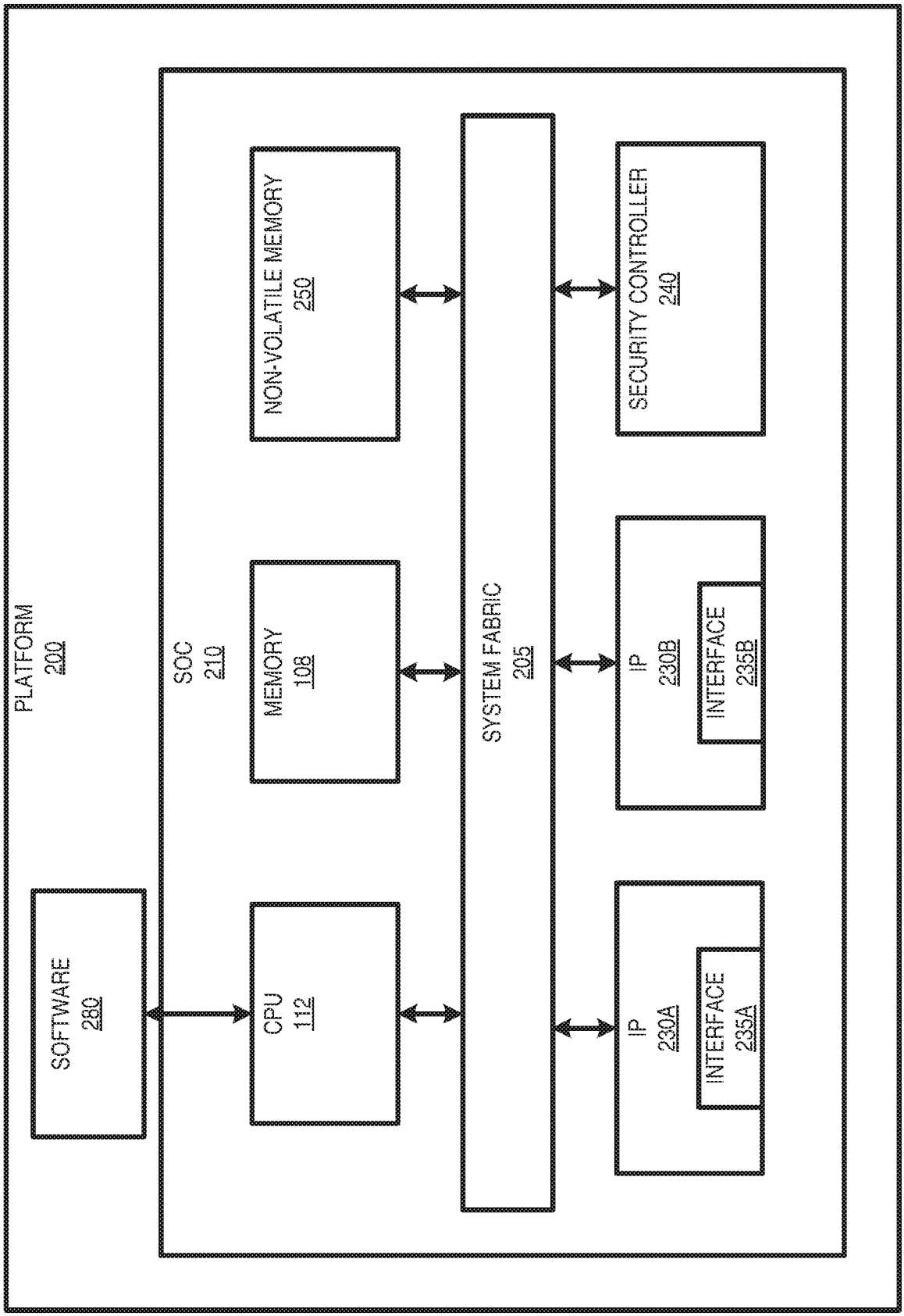
PLATFORM
200
SOC
210
SOFTWARE
280
CPU
112
MEMORY
108
NON-VOLATILE MEMORY
250
SYSTEM FABRIC
205
SECURITY CONTROLLER
240
IP
230A
INTERFACE
235A
IP
230B
INTERFACE
235B

FIG. 2A

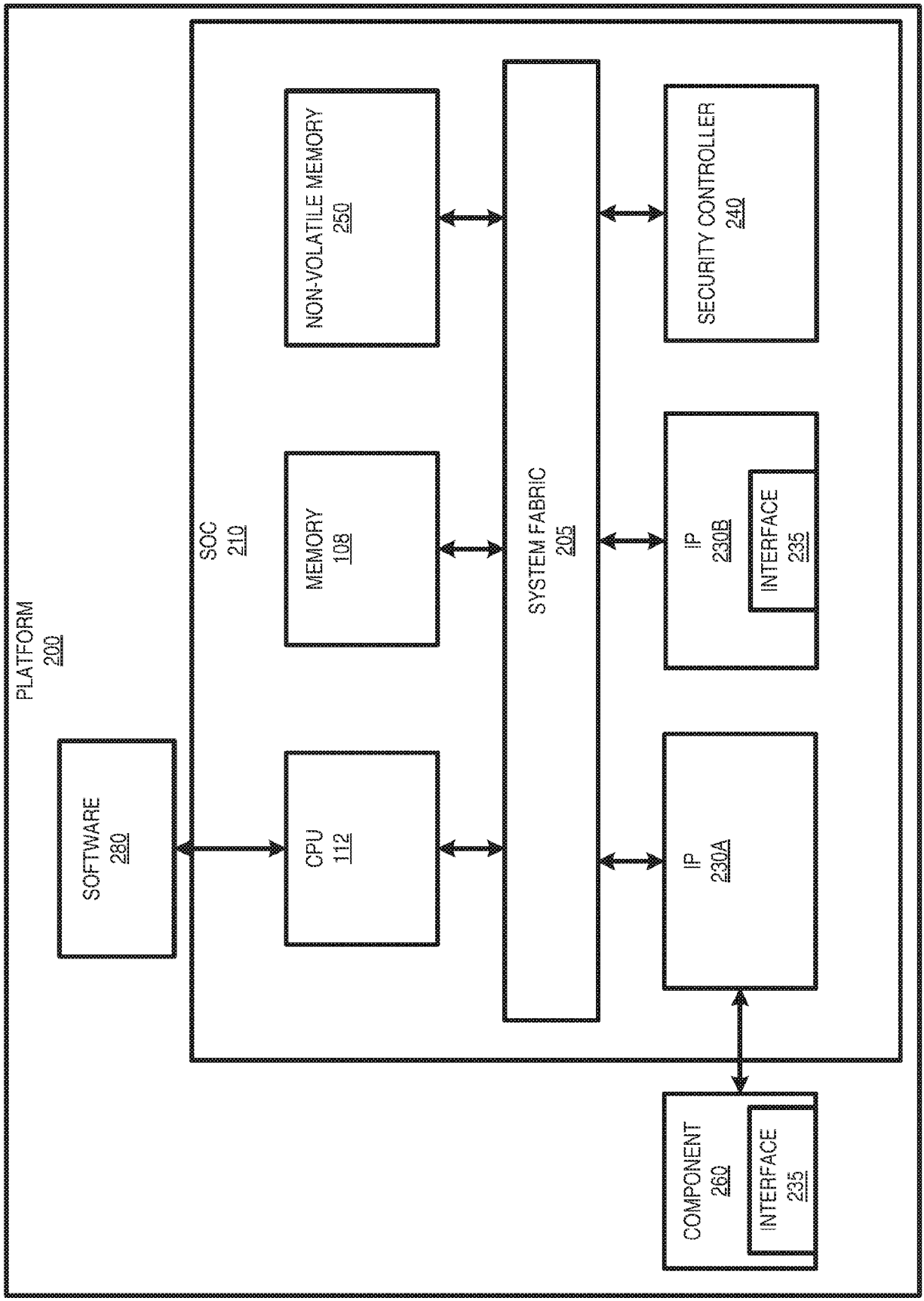
PLATFORM
200
SOC
210
SOFTWARE
280
CPU
112
MEMORY
108
NON-VOLATILE MEMORY
250
SYSTEM FABRIC
205
SECURITY CONTROLLER
240
IP
230B
INTERFACE
235
IP
230A
COMPONENT
260
INTERFACE
235

FIG. 2B

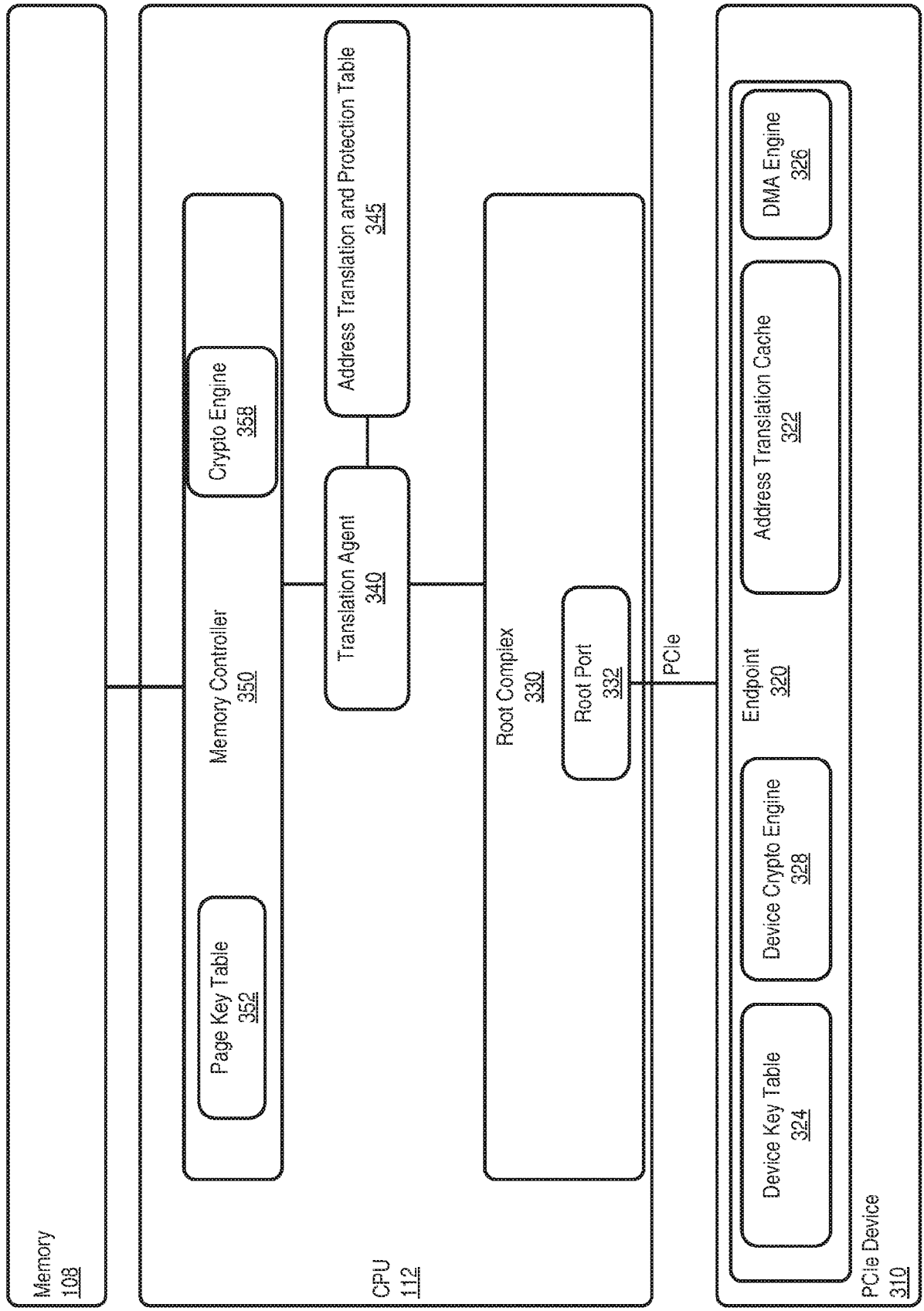
300
Memory
108
CPU
112
Memory Controller
350
Page Key Table
352
Crypto Engine
358
Translation Agent
340
Address Translation and Protection Table
345
Root Complex
330
Root Port
332
PCIe
Endpoint
320
Device Key Table
324
Device Crypto Engine
328
Address Translation Cache
322
DMA Engine
326
PCIe Device
310

FIG. 3

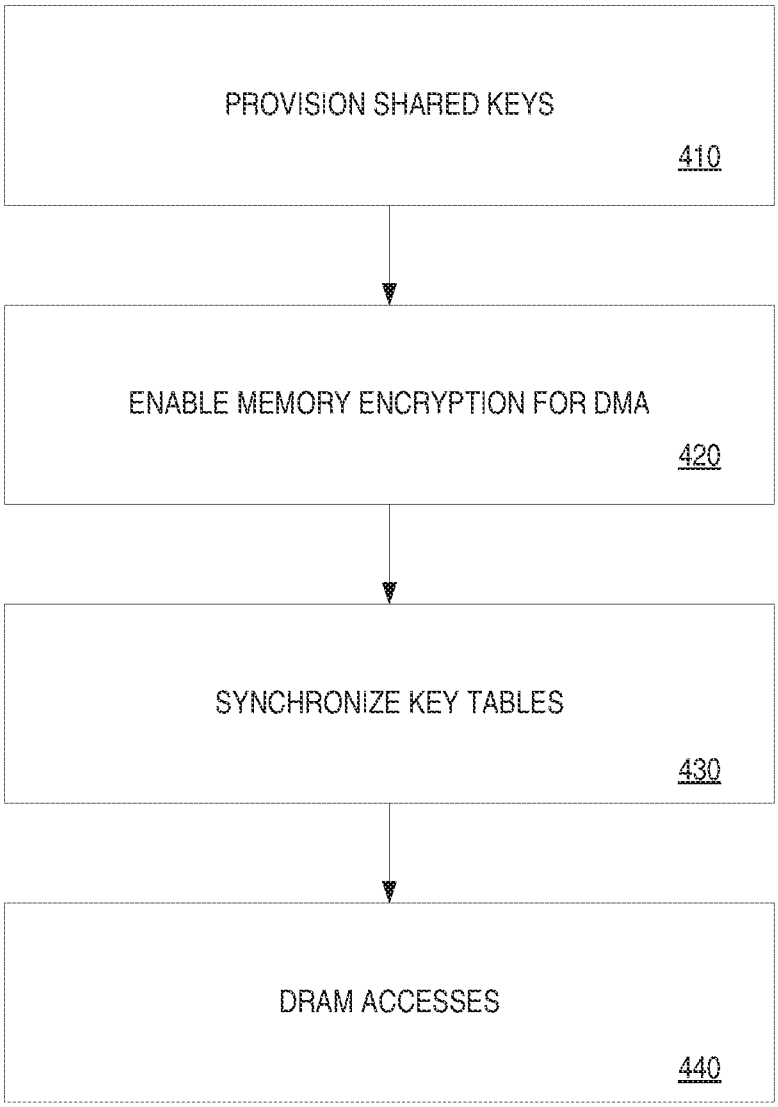
PROVISION SHARED KEYS
410
ENABLE MEMORY ENCRYPTION FOR DMA
420
SYNCHRONIZE KEY TABLES
430
DRAM ACCESSES
440

FIG. 4A

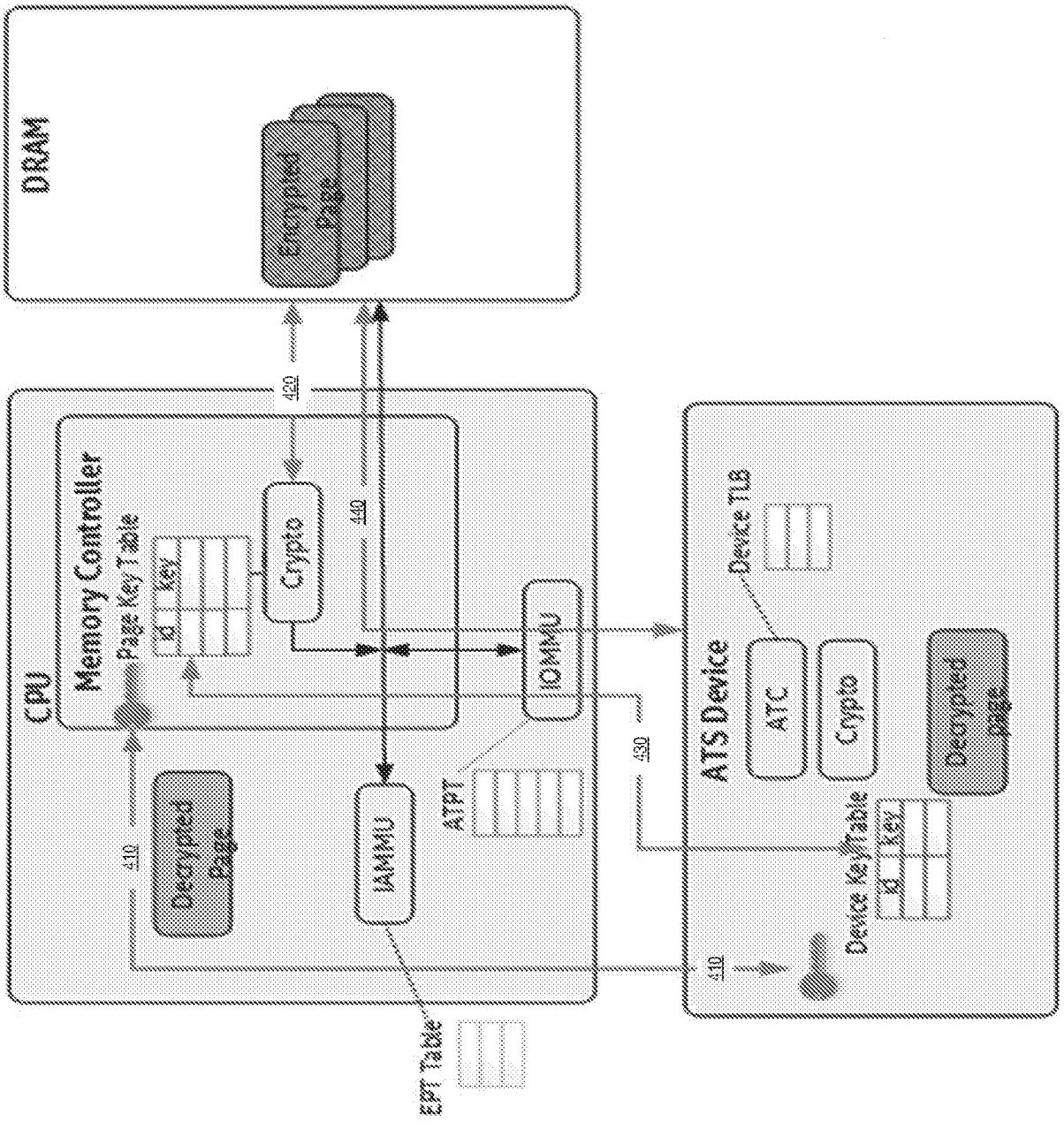
DRAM
Encrypted Page
420
CPU
Memory Controller
Page Key Table
Id
Key
Crypto
440
IOMMU
ATPT
IAMMU
Decrypted Page
410
430
ATS Device
Device TLB
ATC
Crypto
Decrypted page
Device Key Table
Id
Key
410
EPT Table

FIG. 4B

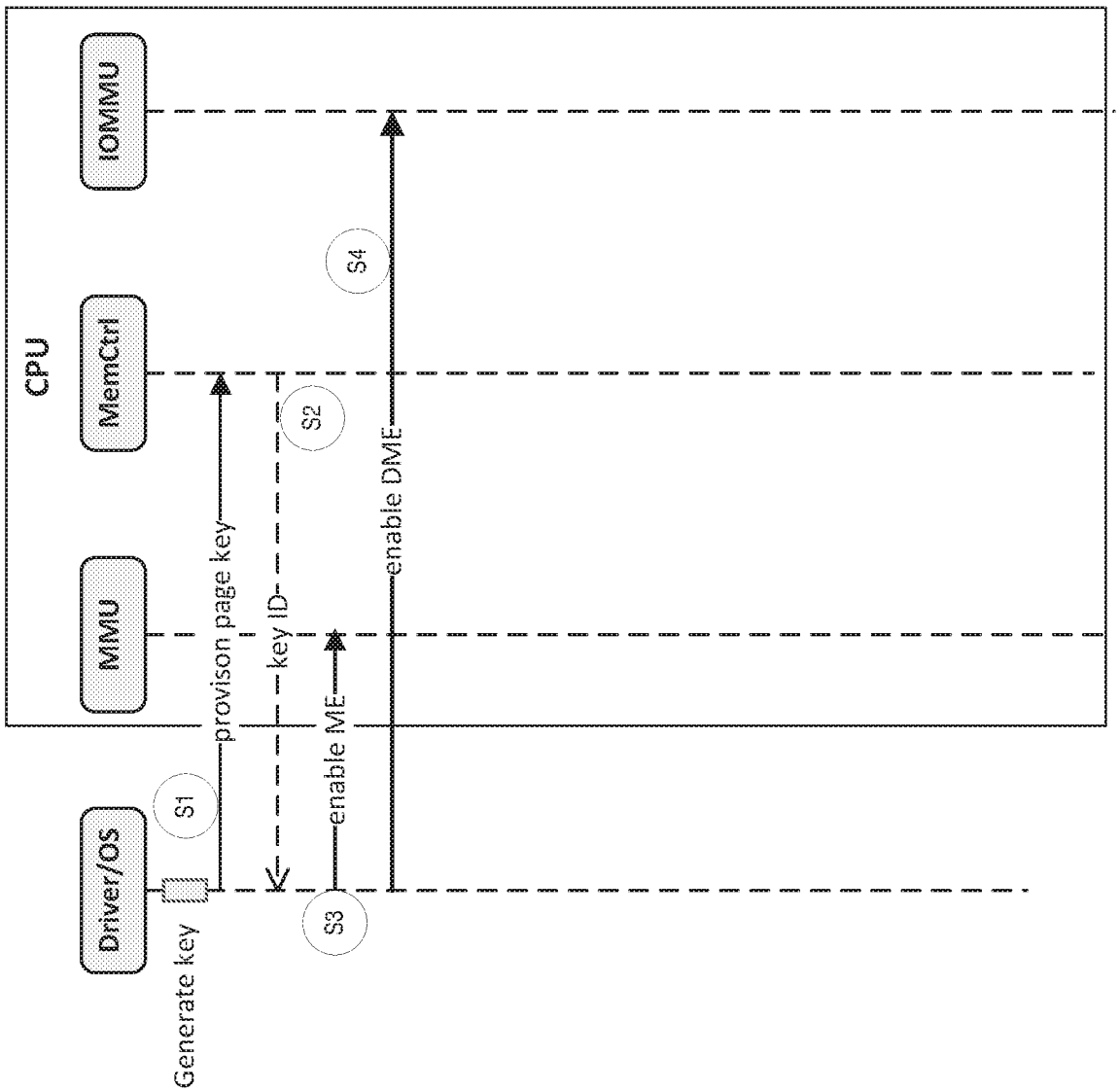
CPU
Driver/OS
MMU
MemCtrl
IOMMU
Generate key
S1
provison page key
S2
key ID
S3
enable ME
S4
enable DME

FIG. 5A

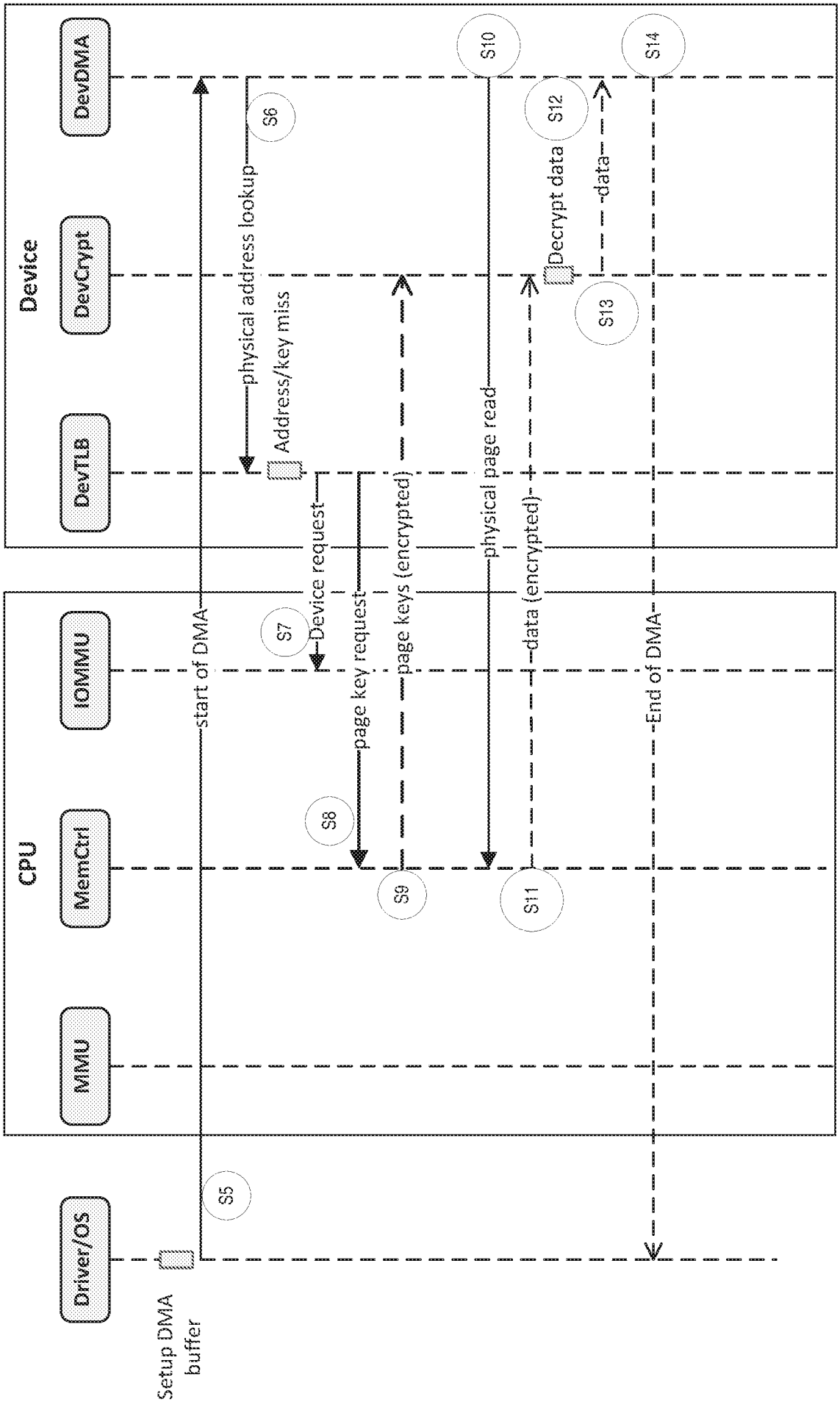
Driver/OS
CPU
MMU
MemCtrl
IOMMU
Device
DevTLB
DevCrypt
DevDMA
Setup DMA buffer
S5
start of DMA
S6
physical address lookup
Address/key miss
S7
Device request
S8
page key request
S9
page keys (encrypted)
S10
physical page read
S11
data (encrypted)
S12
Decrypt data
S13
data
S14
End of DMA

FIG. 5B

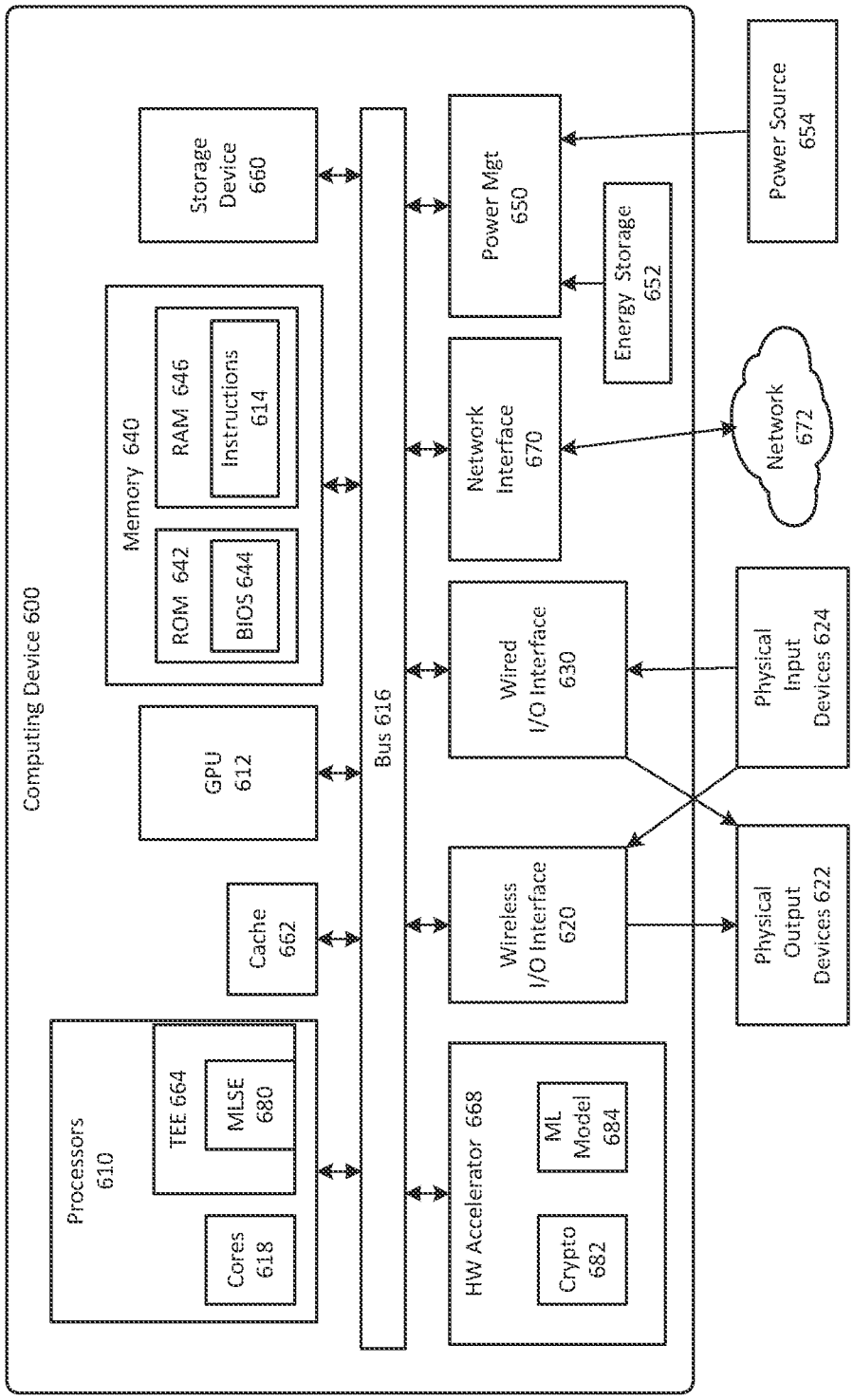
Computing Device 600
Processors 610
Cores 618
TEE 664
MLSE 680
Cache 662
GPU 612
Memory 640
ROM 642
BIOS 644
RAM 646
Instructions 614
Storage Device 660
Bus 616
HW Accelerator 668
Crypto 682
ML Model 684
Wireless I/O Interface 620
Wired I/O Interface 630
Network Interface 670
Power Mgt 650
Energy Storage 652
Power Source 654
Network 672
Physical Input Devices 624
Physical Output Devices 622

FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2020/110148

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/62(2013.01)i; H04L 9/12(2006.01)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 H04L

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

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

CNPAT, CNKI, EPODOC, WPI: memory store share channel encrypted encryption keys decrypt synchronize DMA CPU peripheral

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2016188911 A1 (RAMBUS INC.) 30 June 2016 (2016-06-30) description, paragraphs [0072]-[0092], and figures 1,6	1
A	US 2014068280 A1 (SONY CORPORATION) 06 March 2014 (2014-03-06) the whole document	1-20
A	CN 107643992 A (CHENGDU YIXIN TECHNOLOGY CO., LTD.) 30 January 2018 (2018-01-30) the whole document	1-20
A	CN 106657052 A (HUNAN GOKE MICROELECTRONICS CO., LTD.) 10 May 2017 (2017-05-10) the whole document	1-20

☐ 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
07 May 2021	**21 May 2021**

Name and mailing address of the ISA/CN	Authorized officer
National Intellectual Property Administration, PRC **6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088** **China**	**AN,Xiaolan**
Facsimile No. **(86-10)62019451**	Telephone No. **86-(10)-53961732**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2020/110148

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
US 2016188911 A1	30 June 2016	US 2014237261 A1	21 August 2014
US 2014068280 A1	06 March 2014	EP 2706520 A2	12 March 2014
		CN 103678990 A	26 March 2014
		JP 2014053675 A	20 March 2014
CN 107643992 A	30 January 2018	CN 111797046 A	20 October 2020
CN 106657052 A	10 May 2017	None	