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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
20 January 2022 (20.01.2022)

WIPO | PCT

(10) International Publication Number
WO 2022/015292 A1

(51) International Patent Classification:
G06F 21/30 (2013.01) *G06F 21/72* (2013.01)

(21) International Application Number:
PCT/US2020/041994

(22) International Filing Date:
14 July 2020 (14.07.2020)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.** [US/US]; 10300 Energy Drive, Spring, Texas 77389 (US).

(72) Inventors: **DOWER, Glen Douglas**; 3390 E Harmony Rd, Fort Collins, Colorado 80528 (US). **SEILER, Peter Andrew**; 3390 E Harmony Rd, Fort Collins, Colorado 80528 (US).

(74) Agent: **GORDON, Erica** et al.; 3390 E. Harmony Road, Mail Stop 35, Fort Collins, Colorado 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, 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).

Declarations under Rule 4.17:
— *as to the identity of the inventor (Rule 4.17(i))*
— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

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

(54) Title: COMPUTE SYSTEMS INCLUDING A SECURITY PROCESSOR

(57) Abstract: A compute system includes a security processor, a component, a component memory, a first communication link, and a second communication link. The component memory stores machine readable instructions executable by the component. The first communication link communicatively couples the component memory to the security processor. The second communication link communicatively couples the component memory to the component. The security processor is to cryptographically authenticate the machine readable instructions stored in the component memory in a boot process.

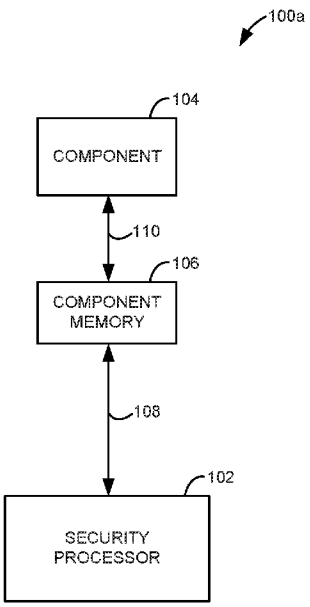

Fig. 1A

WO 2022/015292 A1

COMPUTE SYSTEMS INCLUDING A SECURITY PROCESSOR

Background

[0001] Compute systems may include components that read corresponding machine readable instructions (e.g., firmware) stored in a memory during a boot process. After the boot process, each of the components may execute the corresponding machine readable instructions to implement their individual functions within the compute system.

Brief Description of the Drawings

[0002] Figs. 1A-1F are block diagrams illustrating various examples of a compute system.

[0003] Figs. 2A and 2B are flow diagrams illustrating one example of a method to secure boot a plurality of components within a compute system.

Detailed Description

[0004] In the following detailed description, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific examples in which the disclosure may be practiced. It is to be understood that other examples may be utilized and structural or logical changes may be made without departing from the scope of the present disclosure. The following detailed description, therefore, is not to be taken in a

limiting sense, and the scope of the present disclosure is defined by the appended claims. It is to be understood that features of the various examples described herein may be combined, in part or whole, with each other, unless specifically noted otherwise.

[0005] In a compute system, it may be desirable to cryptographically authenticate machine readable instructions (e.g., firmware) for each component in a secure boot process before the machine readable instructions are loaded by their corresponding components. Cryptographic authentication for machine readable instructions before the machine readable instructions are executed by a component provides a layer of hardware security in a compute system. The cryptographic authentication may include asymmetric key exchange between a component and the machine readable instructions, encrypted certificate exchange from the machine readable instructions to the component, and encrypted hash check of the machine readable instructions against a secure image of the machine readable instructions.

[0006] For a component to cryptographically authenticate its corresponding machine readable instructions as described above, many complex mechanisms may be implemented on each component, such as pseudo random number generators, asymmetric and/or symmetric key generation and exchange, and encrypted hash generation mechanisms. If every discreet component in a compute system is enabled for a secure boot process, these mechanisms would be implemented in numerous components adding cost and complexity to each individual component. In addition, many components do not have the proper architecture to support these cryptographic mechanisms. With multiple components spread across a compute system including private keys, encrypted information, and other security data, there is an increased risk of compromising this information and data. Therefore, it would be advantageous to have secure data and security mechanisms stored in a centralized security processor.

[0007] Accordingly, disclosed herein are examples of compute systems including a centralized security processor to perform a secure boot process of machine readable instructions (e.g., firmware) before the machine readable instructions are read and executed on corresponding components in the

compute system. The machine readable instructions for each corresponding component may be stored in a corresponding memory for each component. The memory for each component includes a dual communication link including a first communication link to the centralized security processor and a second communication link to the corresponding component. The first communication link is used to cryptographically authenticate the machine readable instructions for each component, and the second communication link is used to read the machine readable instructions for execution within the corresponding component once the machine readable instructions have been authenticated by the centralized security processor. The first communication link and the second communication link for the memory for each component may be implemented over the same communication bus, such that both the centralized security processor and the corresponding component share the same communication interface to the memory. Therefore, memory devices including a single communication interface may be used for the component memories.

[0008] The compute systems described herein avoid implementing extensive cryptography capabilities across multiple components in a compute system. Components that use the machine readable instructions are not cryptography enabled. Accordingly, legacy components and subsystems, such as components or subsystems whose architecture prohibits integration of cryptographic capabilities, may be upgraded to be secure boot enabled without adding cryptographic capabilities to the components or subsystems. In addition, the compute systems described herein may save money on cryptographic certifications (e.g., Common Criteria, FIPS, ANSSI, NIST, etc.), since the centralized security processor would include cryptographic capabilities to be certified but each of the components or subsystems would not include cryptographic capabilities to be certified. Thus, one component of a compute system would be certified instead of multiple components of the compute system. Cryptographic certifications generally cost hundreds of thousands of dollars and months to obtain. By certifying one component, compute system designers could use low-demand components in a secure boot environment and

not pay a large overhead to integrate and certify cryptographic capabilities within the low volume components.

[0009] Holding the secure data in a centralized processor versus scattering the secure data over multiple components across a compute system also allows a higher level of security, since there is one access point to secure data rather than multiple access points. Further, in a typical secure boot architecture, multiple components are enabled for secure mechanisms, which generally increases the unit cost and research and development investment into each individual component. In this case, the security architecture tends to become more minimalistic when implemented multiple times on many different components. In contrast, with security mechanisms integrated into a single, centralized processor as disclosed herein, a high investment into the security architecture and processes may be put into that single processor ensuring the greatest protection from physical attacks and software attacks.

[0010] Fig. 1A is a block diagram illustrating one example of a compute system 100a. Compute system 100a includes a security processor 102, a component 104, a component memory 106, a first communication link 108, and a second communication link 110. The component memory 106 stores machine readable instructions executable by the component 104. The first communication link 108 communicatively couples the component memory 106 to the security processor 102. The second communication link 110 communicatively couples the component memory 106 to the component 104. The security processor 102 is to cryptographically authenticate the machine readable instructions stored in the component memory 106 in a boot process.

[0011] Compute system 100a may be a computer system (e.g., server, desktop, laptop, etc.), a tablet, a cellular phone, or another system that performs computing functions. In one example, security processor 102, component 104, component memory 106, first communication link 108, and second communication link 110 may be arranged on a motherboard of the compute system 100a. Security processor 102 may include a MicroController Unit (MCU), a Programmable System On a Chip (PSOC), a Central Processing Unit (CPU), an embedded controller, or another suitable processor. Security

processor 102 may be based on the ARM Cortex CyptoIsland or TrustZone architectures or another suitable security architecture.

[0012] The component 104 may include a Universal Serial Bus (USB) hub, a Local Area Network (LAN) controller, a Baseboard Management Controller (BMC), an Embedded Controller (EC), a Super Input/Output (SIO) controller, a high-speed switch controller, a signal conditioning controller, an external port controller, a power delivery controller, or another suitable compute system component. The first communication link 108 may include a Serial Peripheral Interface (SPI) communication link, an enhanced Serial Peripheral Interface (eSPI) communication link, an Inter-Integrated Circuit (I2C) communication link, a SenseWire (I3C) communication link, a System Management Bus (SMBus) communication link, or another suitable communication link. The second communication link 110 may include an SPI communication link, an eSPI communication link, an I2C communication link, a SenseWire (I3C) communication link, an SMBus communication link, or another suitable communication link.

[0013] Thus, component memory 106 includes a dual communication link, however, a single communication link should be active at a given point in time to reduce read and/or write conflicts. First communication link 108 may be active and second communication link 110 may be inactive to allow security processor 102 to cryptographically authenticate the machine readable instructions stored in component memory 106 during a secure boot process before the machine readable instructions are used by the component 104. Second communication link 110 may be active and first communication link 108 may be inactive to allow component 104 to use the machine readable instructions stored in component memory 106.

[0014] The first communication link 108 and the second communication link 110 may be electrically coupled to the same physical communication interface of the component memory 106. In one example, the first communication link 108 and the second communication link 110 may be electrically coupled to the same physical communication interface of the component memory 106 by wire ORing the first communication link 108 and the second communication link 110 and

interfacing to the component memory 106. In another example, as will be described below with reference to Fig. 1F, the first communication link 108 and the second communication link 110 may be electrically coupled to the same physical communication interface of the component memory 106 by externally multiplexing the first communication link 108 and the second communication link 110 to the component memory 106.

[0015] Fig. 1B is a block diagram illustrating another example of a compute system 100b. Compute system 100b includes a security processor 102, a plurality of components 104_1 to 104_N, a plurality of component memories 106_1 to 106_N, a plurality of first communication links 108_1 to 108_N, and a plurality of second communication links 110_1 to 110_N, where "N" is any suitable number of components. Each component memory 106_1 to 106_N corresponds to a component 104_1 to 104_N and stores machine readable instructions executable by the corresponding component 104_1 to 104_N, respectively. Each first communication link 108_1 to 108_N communicatively couples the corresponding component memory 106_1 to 106_N to the security processor 102, respectively. Each second communication link 110_1 to 110_N communicatively couples the corresponding component memory 106_1 to 106_N to the corresponding component 104_1 to 104_N, respectively.

[0016] The security processor 102 is to cryptographically authenticate the machine readable instructions stored in each of the plurality of component memories 106_1 to 106_N in a boot process. In one example, the security processor 102 is configured to cryptographically authenticate the machine readable instructions stored in each of the plurality of component memories 106_1 to 106_N in parallel during the boot process. In another example, the security processor 102 is configured to cryptographically authenticate the machine readable instructions stored in each of the plurality of component memories 106_1 to 106_N in series during the boot process.

[0017] Compute system 100b may be a computer system (e.g., server, desktop, laptop, etc.), a tablet, a cellular phone, or another system that performs computing functions. In one example, security processor 102, components 104_1 to 104_N, component memories 106_1 to 106_N, first communication links 108_1 to

108_N, and second communication links 110_1 to 110_N may be arranged on a motherboard of the compute system 100b.

[0018] Each component 104_1 to 104_N may include a USB hub, a LAN controller, a BMC, an EC, an SIO controller, a high-speed switch controller, a signal conditioning controller, an external port controller, a power delivery controller, or another suitable compute system component. Each first communication link 108_1 to 108_N may include an SPI communication link, an eSPI communication link, an I2C communication link, a SenseWire (I3C) communication link, an SMBus communication link, or another suitable communication link. Each second communication link 110_1 to 110_N may include an SPI communication link, an eSPI communication link, an I2C communication link, a SenseWire (I3C) communication link, an SMBus communication link, or another suitable communication link. The first communication links 108_1 to 108_N may include a single type of communication link or a mixture of different types of communication links. Likewise, the second communication links 110_1 to 110_N may include a single type of communication link or a mixture of different types of communication links.

[0019] Fig. 1C is a block diagram illustrating another example of a compute system 100c. Compute system 100c is similar to compute system 100b previously described and illustrated with reference to Fig. 1B, except that compute system 100c includes additional features. In addition to a security processor 102, a plurality of components 104_1 to 104_N, a plurality of component memories 106_1 to 106_N, a plurality of first communication links 108_1 to 108_N, and a plurality of second communication links 110_1 to 110_N, compute system 100c includes a security processor memory 120. The security processor memory 120 is communicatively coupled to the security processor 102 through a communication link 130. The security processor memory 120 stores machine readable instructions 122 executable by the security processor 102, a private key(s) 124, a secure certificate(s) 126, and backup machine readable instructions 128 for the component memories 106_1 to 106_N.

[0020] The security processor 102 may execute the machine readable instructions 122 stored in security processor memory 120 to cryptographically

authenticate the machine readable instructions stored in each of the component memories 106_1 to 106_N. The security processor 102 is to cryptographically authenticate the machine readable instructions stored in each of the component memories 106_1 to 106_N by asymmetric key exchange, certificate verification, and secure hash algorithm (SHA) checks between each of the component memories 106_1 to 106_N and the security processor 102. In one example, the asymmetric key exchange may be based on the private key(s) 124 stored in the security processor memory 120, and the certificate verification may be based on the secure certificate(s) 126 stored in the security processor memory 120.

[0021] The security processor 102 is to erase a selected component memory 106_1 to 106_N and write backup machine readable instructions to the selected component memory 106_1 to 106_N in response to the cryptographic authentication of the selected component memory 106_1 to 106_N failing. In one example, the security processor 102 writes backup machine readable instructions to the selected component memory 106_1 to 106_N from the backup instructions 128 of the security processor memory 120 in response to the cryptographic authentication of the selected component memory 106_1 to 106_N failing.

[0022] In this example, each of the plurality of components 104_1 to 104_N includes a reset input 132_1 to 132_N, respectively. The security processor 102 is communicatively coupled to each of the plurality of reset inputs 132_1 to 132_N through a reset signal path 134. The security processor 102 is to disable each of the plurality of components 104_1 to 104_N by applying a signal having a first state (e.g., a logic high “1” state or a logic low “0” state) to the reset input 132_1 to 132_N of each of the plurality of components 104_1 to 104_N, respectively, while the security processor 102 is cryptographically authenticating the machine readable instructions stored in each of the plurality of component memories 106_1 to 106_N.

[0023] Once the security processor 102 has cryptographically authenticated the machine readable instructions stored in each of the plurality of component memories 106_1 to 106_N, the security processor 102 enables each of the plurality of components 104_1 to 104_N by applying a signal having a second state (e.g.,

opposite to the first state) to the reset input 132_1 to 132_N of each of the plurality of components 104_1 to 104_N, respectively. With each of the plurality of components 104_1 to 104_N enabled, each of the components 104_1 to 104_N may load and execute the machine readable instructions stored in the corresponding component memories 106_1 to 106_N, respectively.

[0024] Fig. 1D is a block diagram illustrating another example of a compute system 100d. Compute system 100d includes a single security processor 102, a plurality of components 104_1 to 104_N, a plurality of Read-Only Memories (ROMs) 106_1 to 106_N, a first communication link 108, and a plurality of second communication links 110_1 to 110_N. In addition, compute system 100d includes a security processor ROM 120, a power source 144, and a plurality of switches 146_1 to 146_N. Each ROM 106_1 to 106_N corresponds to a component 104_1 to 104_N and stores machine readable instructions 107_1 to 107_N executable by the corresponding component 104_1 to 104_N. Each ROM 106_1 to 106_N is communicatively coupled to the single security processor 102 through a first communication link 108 and to the corresponding component 104_1 to 104_N through a corresponding second communication link 110_1 to 110_N. Each of the plurality of ROMs 106_1 to 106_N may include a Serial Peripheral Interface (SPI) ROM, an Electrically Erasable Programmable Read-Only Memory (EEPROM), a flash memory, or another suitable ROM. Security processor ROM 120 may include an SPI ROM, an EEPROM, a flash memory, or another suitable ROM.

[0025] In this example, each of the plurality of components 104_1 to 104_N includes a power input 142_1 to 142_N, respectively. Each switch 146_1 to 146_N is electrically coupled between the power source 144 and a power input 142_1 to 142_N of the corresponding component 104_1 to 104_N, respectively. The control input of each switch 146_1 to 146_N is electrically coupled to the security processor 102 through a reset signal path 148. In response to the security processor 102 applying a reset signal having a first state to reset signal path 148, each of the switches 146_1 to 146_N are opened to remove power from the corresponding components 104_1 to 104_N. With power removed from each of the components 104_1 to 104_N, the components 104_1 to 104_N are disabled. In response to the security processor 102 applying a reset signal having a second

state to reset signal path 148, each of the switches 146_1 to 146_N are closed to apply power to the corresponding components 104_1 to 104_N. With power applied to the components 104_1 to 104_N, the components 104_1 to 104_N are enabled to read and execute the corresponding machine readable instructions 107_1 to 107_N stored in the corresponding ROMs 106_1 to 106_N, respectively. Each switch 146_1 to 146_N may include a transistor(s), a logic gate(s), and/or other suitable circuitry for selectively applying and removing power from power source 144 to each component 104_1 to 104_N.

[0026] The security processor 102 is to, in a boot process, disable each of the plurality of components 104_1 to 104_N, cryptographically authenticate the machine readable instructions 107_1 to 107_N stored in each of the ROMs 106_1 to 106_N with each of the plurality of components 104_1 to 104_N disabled, and enable each of the plurality of components 104_1 to 104_N with the machine readable instructions 107_1 to 107_N stored in each of the ROMs 106_1 to 106_N cryptographically authenticated. In one example, the security processor 102 is to cryptographically authenticate the machine readable instructions 107_1 to 107_N stored in each of the ROMs 106_1 to 106_N by asymmetric key exchange, certificate verification, and SHA checks between each of the ROMs 106_1 to 106_N and the security processor 102.

[0027] In this example, the security processor 102 is to control each of the plurality of switches 146_1 to 146_N to remove power from each of the plurality of components 104_1 to 104_N while the security processor 102 is cryptographically authenticating the machine readable instructions 107_1 to 107_N stored in each of the plurality of ROMs 106_1 to 106_N. The security processor 102 is to erase a selected ROM 106_1 to 106_N and write backup machine readable instructions to the selected ROM 106_1 to 106_N in response to the cryptographic authentication of the selected ROM 106_1 to 106_N failing.

[0028] In this example, first communication link 108 is a common communication link communicatively coupled between the security processor 102 and each of the plurality of ROMs 106_1 to 106_N. In other examples, a first subset of the plurality of ROMs 106_1 to 106_N may be communicatively coupled to the security processor 102 through a corresponding discrete communication

link as previously described and illustrated with reference to Figs. 1B and1C, and a second subset of the plurality of ROMs 106_1 to 106_N may be communicatively coupled to the security processor 102 through a shared communication link.

[0029] Fig. 1E is a block diagram illustrating another example of a compute system 100e. Compute system 100e is similar to compute system 100d previously described and illustrated with reference to Fig. 1D, except that compute system 100e includes a single component or multiple components that are enabled and disabled by the security processor 102 via a corresponding reset input of the component(s) and a single component or multiple components that are enabled and disabled by the security processor 102 via a power input of the component(s). In this example, components 104_1 and 104_2 include power inputs 142_1 and 142_2, respectively, as previously described and illustrated with reference to Fig. 1D, and component 104_N includes a reset input 132_N as previously described and illustrated with reference to Fig. 1C. Each switch 146_1 and 146_2 is electrically coupled between the power source 144 and the power input 142_1 and 142_2 of the corresponding component 104_1 and 104_2, respectively. The control input of switches 146_1 and 146_2 are electrically coupled to security processor 102 through a first reset (RESET1) signal path 148. The reset input 132_N of component 104_N is electrically coupled to security processor 102 through as second reset (RESET2) signal path 134.

[0030] In this example, the security processor 102 is to control each of the switches 146_1 and 146_2 to remove power from each of the components 104_1 and 104_2 and to disable component 104_N through reset input 132_N while the security processor 102 is cryptographically authenticating the machine readable instructions 107_1 to 107_N stored in each of the plurality of ROMs 106_1 to 106_N. Once the machine readable instructions 107_1 to 107_N stored in each of the plurality of ROMs 106_1 to 106_N are cryptographically authenticated, the security processor 102 is to control each of the switches 146_1 and 146_2 to apply power to each of the components 104_1 and 104_2 and to enable component 104_N through reset input 132_N.

[0031] Fig. 1F is a block diagram illustrating another example of a compute system 100f. Compute system 100f is similar to compute system 100a previously described and illustrated with reference to Fig. 1A, except that compute system 100f includes a multiplexer 150. A first input of multiplexer 150 is electrically coupled to the first communication link 108. A second input of multiplexer 150 is electrically coupled to the second communication link 110. The output of multiplexer 150 is electrically coupled to a single communication interface of the component memory 106 through a communication link 152. The control input of multiplexer 150 is electrically coupled to the security processor 102 through a control signal path 154.

[0032] In this example, security processor 102 controls multiplexer 150 to connect either the first communication link 108 or the second communication link 110 to the component memory 106. Security processor 102 connects the first communication link 108 to the component memory 106 with the security processor 102 cryptographically authenticating the machine readable instructions stored in the component memory 106 during a secure boot process. Security processor 102 connects the second communication link 110 to the component memory 106 once the secure boot process is complete so that component 104 may use the machine readable instructions stored in the component memory 106.

[0033] Figs. 2A and 2B are flow diagrams illustrating one example of a method 200 to secure boot a plurality of components within a compute system. In one example, method 200 may be implemented by compute system 100a, 100b, 100c, 100d, or 100e previously described and illustrated with reference to Figs. 1A-1E, respectively. As illustrated in Fig. 2A at 202, method 200 includes disabling the plurality of components (e.g., components 104_1 to 104_N), each of the plurality of components communicatively coupled to a corresponding component memory (e.g., component memory or ROM 106_1 to 106_N) of the compute system. At 204, method 200 includes with each of the plurality of components disabled, cryptographically authenticating, via a security processor (e.g., security processor 102) of the compute system, machine readable instructions (e.g., instructions 107_1 to 107_N) stored in each corresponding

component memory. At 206, method 200 includes enabling each of the plurality of components with the machine readable instructions stored in each corresponding component memory cryptographically authenticated. At 208, method 200 includes with each of the plurality of components enabled, executing the machine readable instructions stored in each corresponding component memory via the corresponding component.

[0034] In one example, disabling the plurality of components comprises applying a signal having a first state to a reset input (e.g., reset input 132_1 to 132_N) of each of the plurality of components. In this case, enabling each of the plurality of components comprises applying a signal having a second state to the reset input of each of the plurality of components. In another example, disabling the plurality of components comprises removing power from each of the plurality of components (e.g., via switches 146_1 to 146_N). In this case, enabling each of the plurality of components comprises applying power to each of the plurality of components.

[0035] As illustrated in Fig. 2B at 210, method 200 may further include erasing a selected component memory in response to the machine readable instructions stored in the selected component memory failing the cryptographic authentication. At 212, method 200 may further include writing backup machine readable instructions (e.g., from backup instructions 128) to the selected component memory.

[0036] Although specific examples have been illustrated and described herein, a variety of alternate and/or equivalent implementations may be substituted for the specific examples shown and described without departing from the scope of the present disclosure. This application is intended to cover any adaptations or variations of the specific examples discussed herein. Therefore, it is intended that this disclosure be limited only by the claims and the equivalents thereof.

CLAIMS

1. A compute system comprising:

a security processor;

a component;

a component memory storing machine readable instructions executable by the component;

a first communication link communicatively coupling the component memory to the security processor; and

a second communication link communicatively coupling the component memory to the component,

wherein the security processor is to cryptographically authenticate the machine readable instructions stored in the component memory in a boot process.

2. The compute system of claim 1, further comprising:

a security processor memory communicatively coupled to the security processor, the security processor memory storing machine readable instructions executable by the security processor, a private key, a secure certificate, and backup machine readable instructions for the component memory.

3. The compute system of claim 1, wherein the component comprises a reset input, and

wherein the security processor is to disable the component by applying a signal having a first state to the reset input of the component while the security processor is cryptographically authenticating the machine readable instructions stored in the component memory.

4. The compute system of claim 1, further comprising:

a switch electrically coupled between a power source and a power input of the component,

wherein the security processor is to control the switch to remove power from the component while the security processor is cryptographically authenticating the machine readable instructions stored in the component memory.

5. The compute system of claim 1, wherein the component comprises a Universal Serial Bus (USB) hub, a Local Area Network (LAN) controller, a Baseboard Management Controller (BMC), an Embedded Controller (EC), a Super Input/Output (SIO) controller, a high-speed switch controller, a signal conditioning controller, an external port controller, or a power delivery controller.

6. The compute system of claim 1, wherein each of the first communication link and the second communication link comprise a Serial Peripheral Interface (SPI) communication link, an enhanced Serial Peripheral Interface (eSPI) communication link, an Inter-Integrated Circuit (I2C) communication link, a SenseWire (I3C) communication link, or a System Management Bus (SMBus) communication link.

7. The compute system of claim 1, wherein the first communication link and the second communication link are electrically coupled to a single communication interface of the component memory.

8. A compute system comprising:

a single security processor;

a plurality of components; and

a plurality of Read-Only Memories (ROMs), each ROM corresponding to a component of the plurality of components, each ROM storing machine readable instructions executable by the corresponding component, and each ROM communicatively coupled to the single security processor and the corresponding component;

wherein the security processor is to, in a boot process, disable each of the plurality of components, cryptographically authenticate the machine

readable instructions stored in each of the ROMs with each of the plurality of components disabled, and enable each of the plurality of components with the machine readable instructions stored in each of the ROMs cryptographically authenticated.

9. The compute system of claim 8, wherein each of the plurality of ROMs comprises a Serial Peripheral Interface (SPI) ROM, an Electrically Erasable Programmable Read-Only Memory (EEPROM), or a flash memory.

10. The compute system of claim 8, wherein the security processor is to cryptographically authenticate the machine readable instructions stored in each of the ROMs by asymmetric key exchange, certificate verification, and secure hash algorithm (SHA) checks between each of the ROMs and the security processor.

11. The compute system of claim 8, wherein the security processor is to erase a selected ROM of the plurality of ROMs and write backup machine readable instructions to the selected ROM in response to the cryptographic authentication of the selected ROM failing.

12. A method to secure boot a plurality of components within a compute system, the method comprising:

disabling the plurality of components, each of the plurality of components communicatively coupled to a corresponding component memory of the compute system;

with each of the plurality of components disabled, cryptographically authenticating, via a security processor of the compute system, machine readable instructions stored in each corresponding component memory;

enabling each of the plurality of components with the machine readable instructions stored in each corresponding component memory cryptographically authenticated; and

with each of the plurality of components enabled, executing the machine readable instructions stored in each corresponding component memory via the corresponding component.

13. The method of claim 12, wherein disabling the plurality of components comprises applying a signal having a first state to a reset input of each of the plurality of components, and

wherein enabling each of the plurality of components comprises applying a signal having a second state to the reset input of each of the plurality of components.

14. The method of claim 12, wherein disabling the plurality of components comprises removing power from each of the plurality of components, and

wherein enabling each of the plurality of components comprises applying power to each of the plurality of components.

15. The method of claim 12, further comprising:

erasing a selected component memory in response to the machine readable instructions stored in the selected component memory failing the cryptographic authentication; and

writing backup machine readable instructions to the selected component memory.

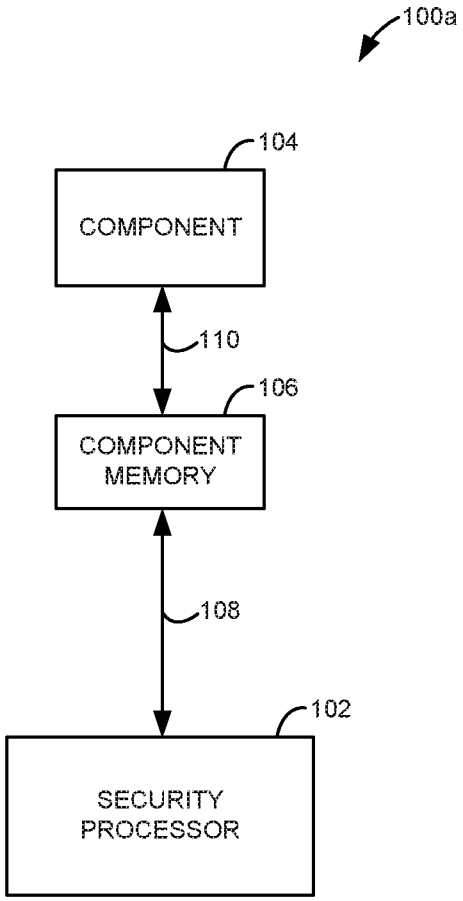
100a
104
COMPONENT
110
106
COMPONENT MEMORY
108
102
SECURITY PROCESSOR

Fig. 1A

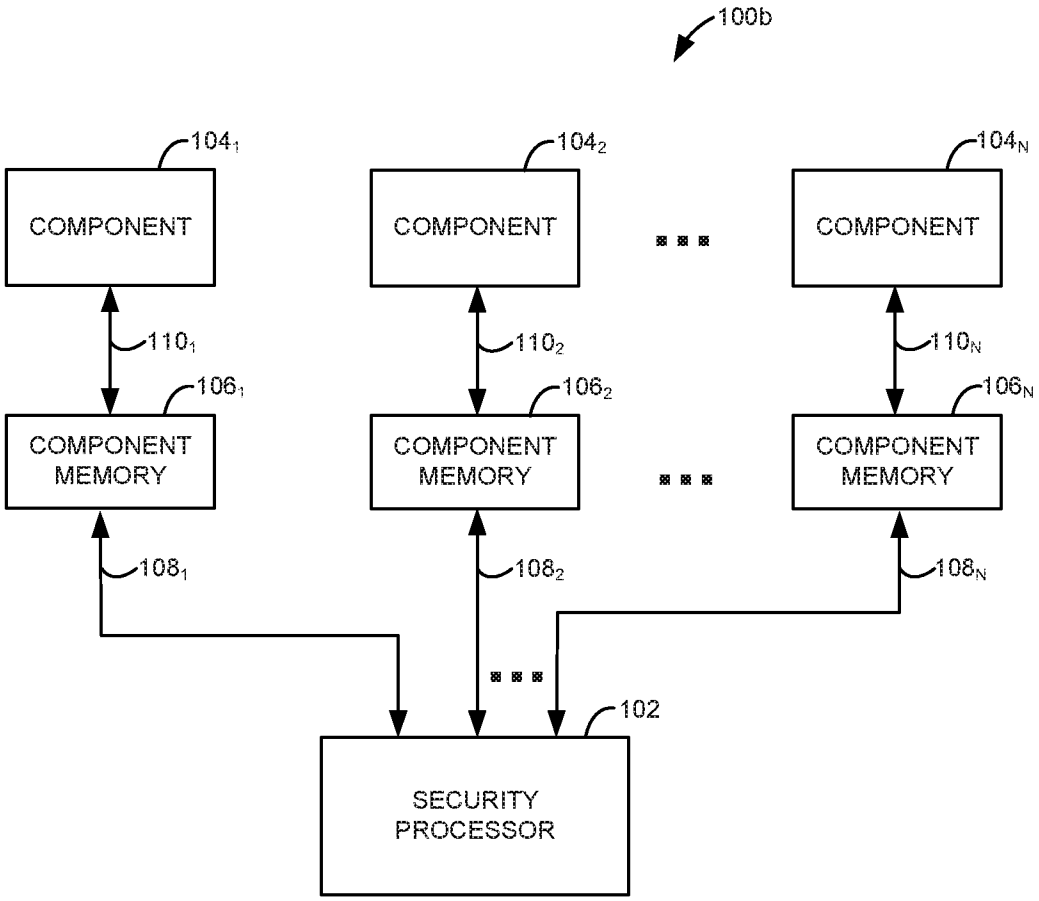
100b
104₁
COMPONENT
104₂
COMPONENT
104N
COMPONENT
110₁
110₂
110N
106₁
106₂
106N
COMPONENT MEMORY
COMPONENT MEMORY
COMPONENT MEMORY
108₁
108₂
108N
102
SECURITY PROCESSOR

Fig. 1B

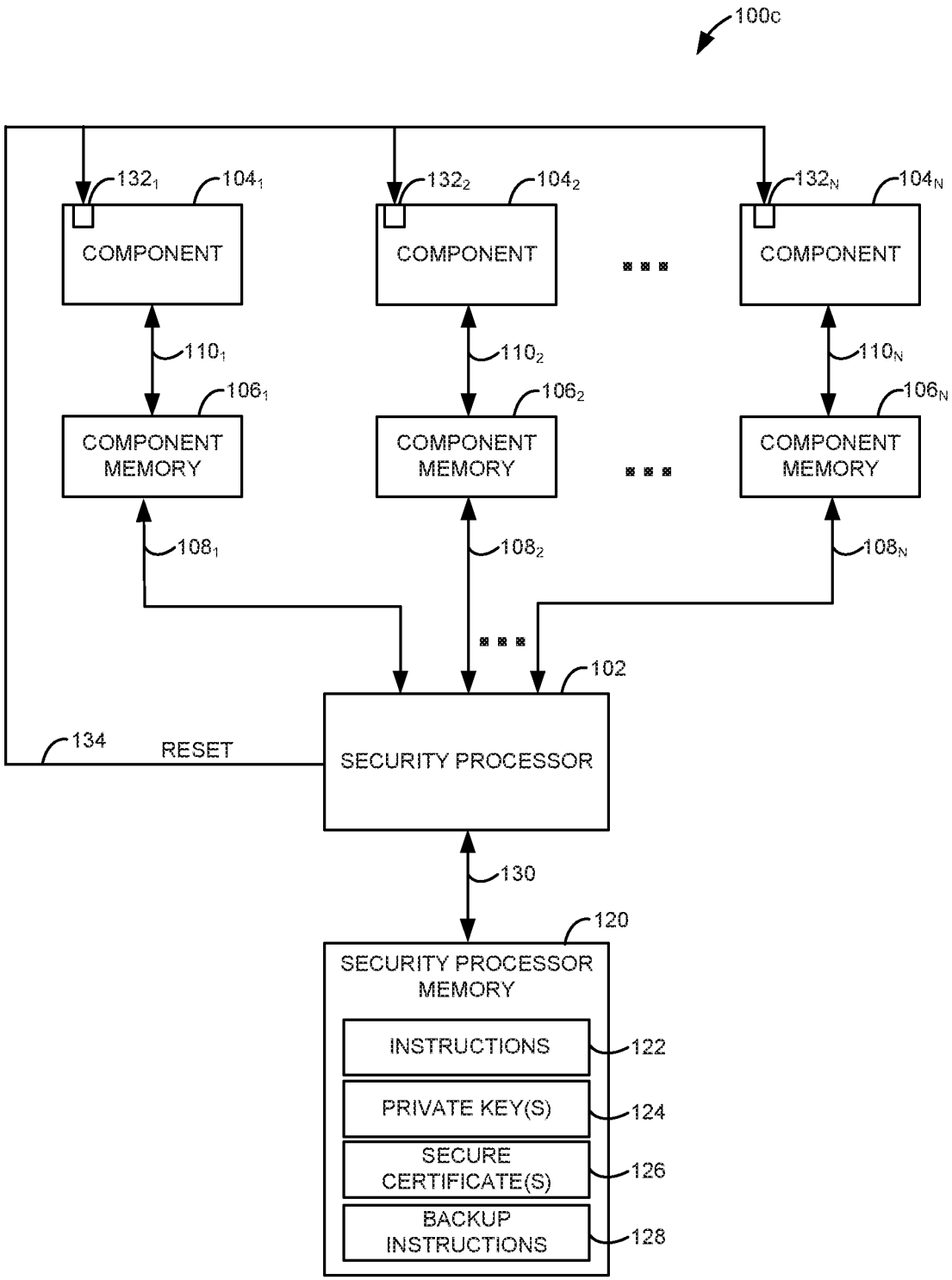
100c
132$_1$
104$_1$
COMPONENT
132$_2$
104$_2$
COMPONENT
132$_N$
104$_N$
COMPONENT
110$_1$
110$_2$
110$_N$
106$_1$
106$_2$
106$_N$
COMPONENT MEMORY
COMPONENT MEMORY
COMPONENT MEMORY
108$_1$
108$_2$
108$_N$
102
134
RESET
SECURITY PROCESSOR
130
120
SECURITY PROCESSOR MEMORY
INSTRUCTIONS
122
PRIVATE KEY(S)
124
SECURE CERTIFICATE(S)
126
BACKUP INSTRUCTIONS
128

Fig. 1C

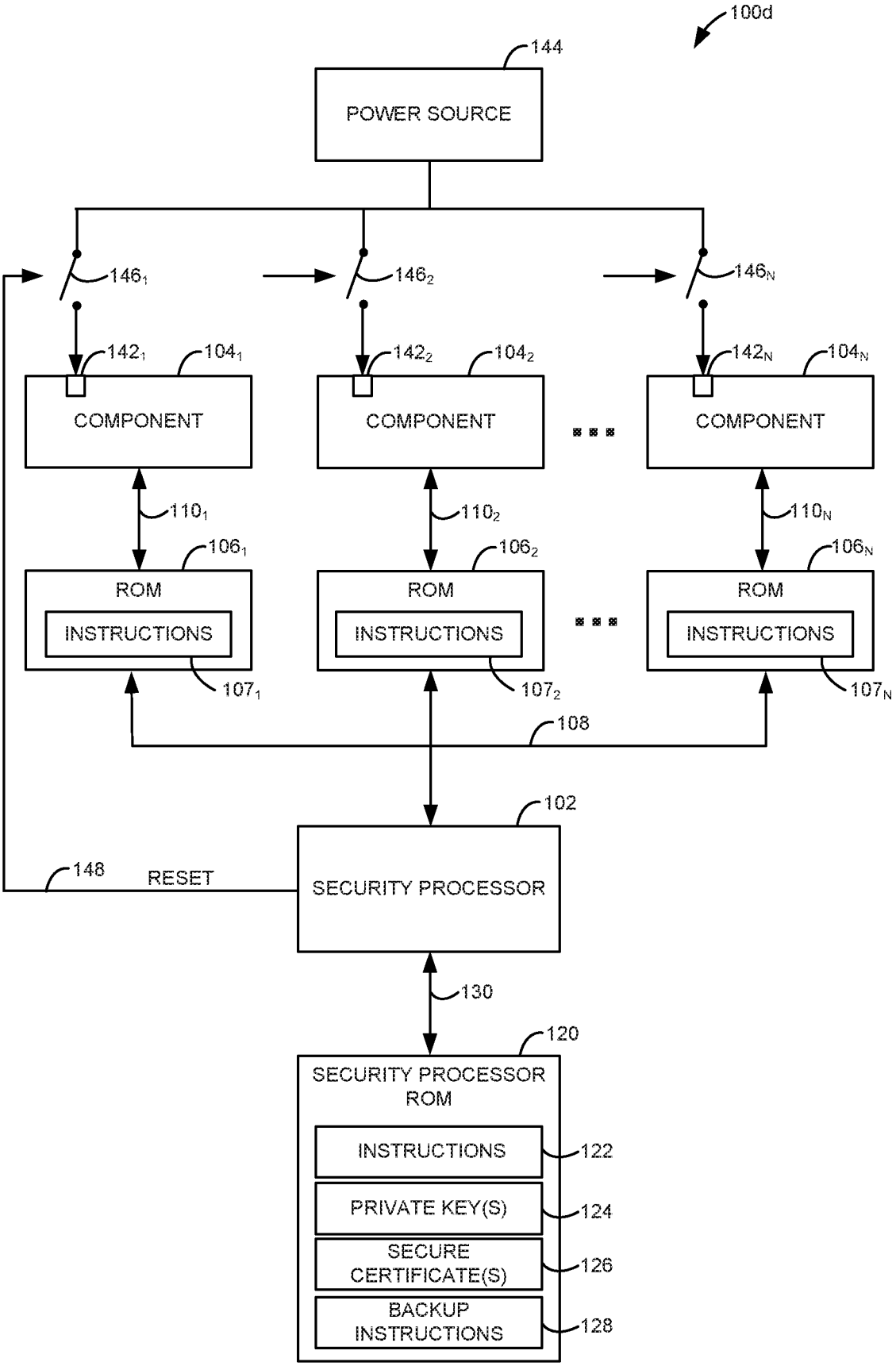
100d
144
POWER SOURCE
146$_1$
146$_2$
146$_N$
142$_1$
104$_1$
142$_2$
104$_2$
142$_N$
104$_N$
COMPONENT
COMPONENT
COMPONENT
110$_1$
110$_2$
110$_N$
106$_1$
106$_2$
106$_N$
ROM
INSTRUCTIONS
ROM
INSTRUCTIONS
ROM
INSTRUCTIONS
107$_1$
107$_2$
107$_N$
108
102
148
RESET
SECURITY PROCESSOR
130
120
SECURITY PROCESSOR ROM
INSTRUCTIONS
122
PRIVATE KEY(S)
124
SECURE CERTIFICATE(S)
126
BACKUP INSTRUCTIONS
128

Fig. 1D

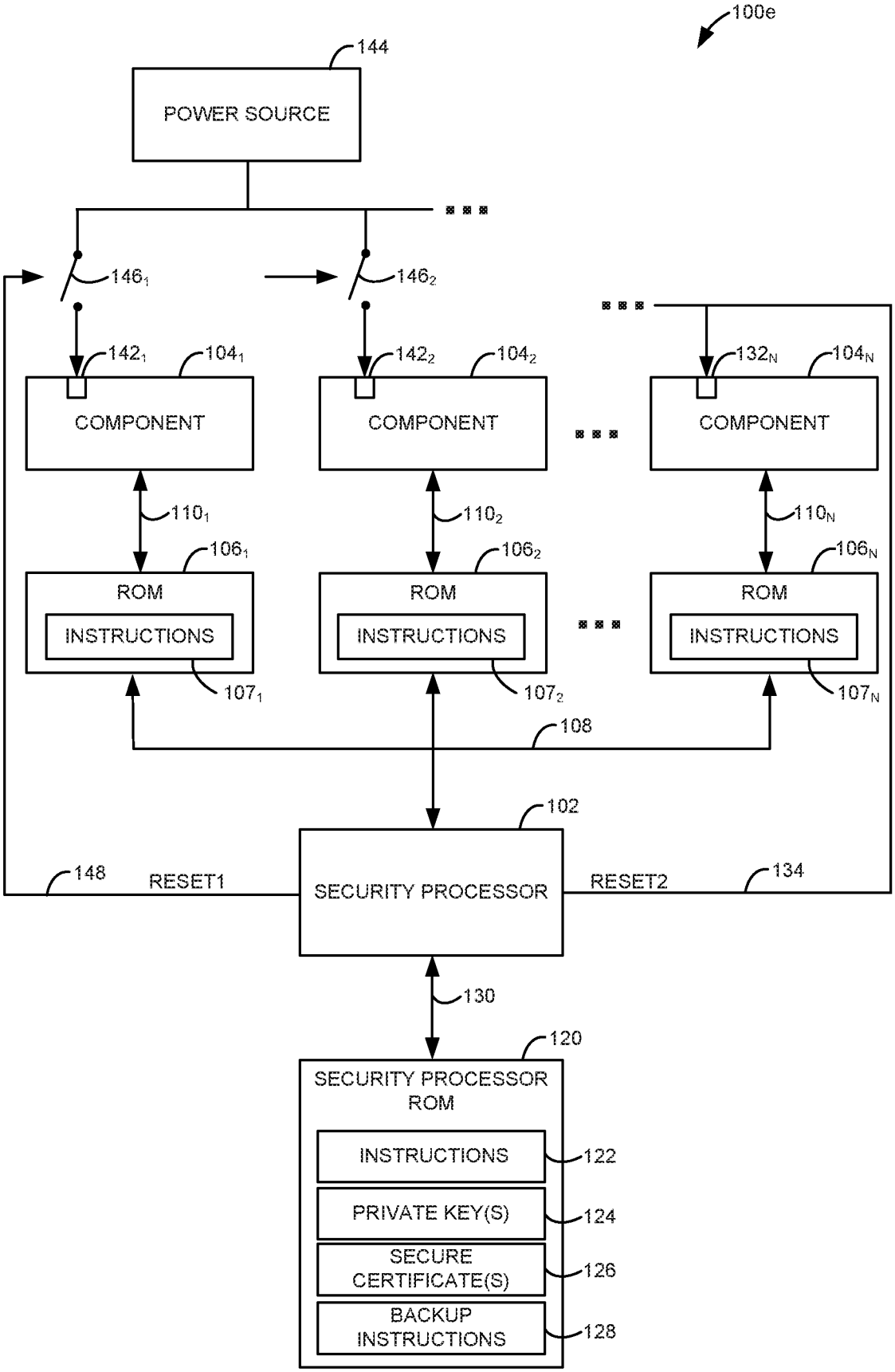
100e
144
POWER SOURCE
146$_1$
146$_2$
142$_1$
104$_1$
142$_2$
104$_2$
132$_N$
104$_N$
COMPONENT
COMPONENT
COMPONENT
110$_1$
110$_2$
110$_N$
106$_1$
106$_2$
106$_N$
ROM
INSTRUCTIONS
ROM
INSTRUCTIONS
ROM
INSTRUCTIONS
107$_1$
107$_2$
107$_N$
108
102
148
RESET1
SECURITY PROCESSOR
RESET2
134
130
120
SECURITY PROCESSOR ROM
INSTRUCTIONS
122
PRIVATE KEY(S)
124
SECURE CERTIFICATE(S)
126
BACKUP INSTRUCTIONS
128

Fig. 1E

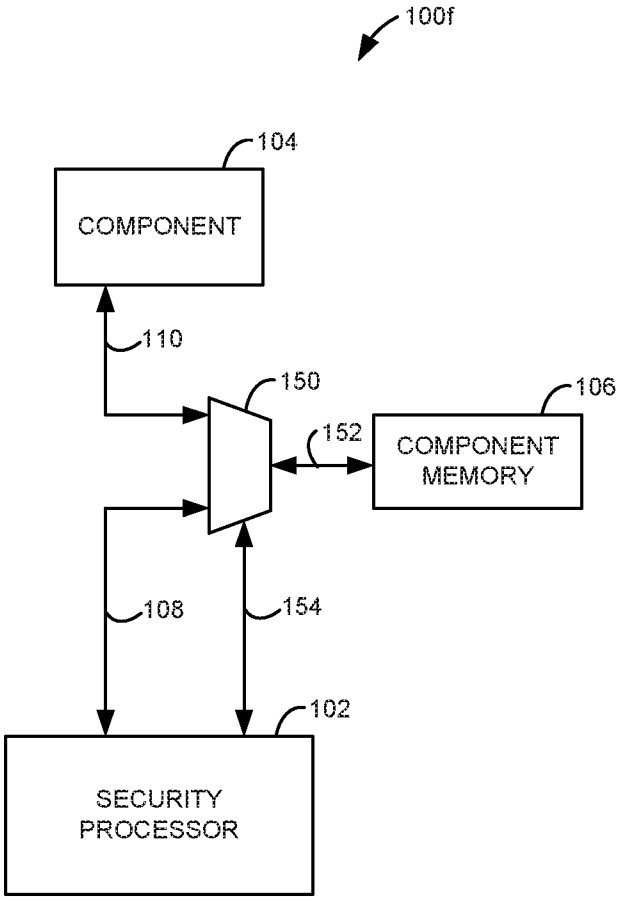
100f
104
COMPONENT
110
150
106
152
COMPONENT MEMORY
108
154
102
SECURITY PROCESSOR

Fig. 1F

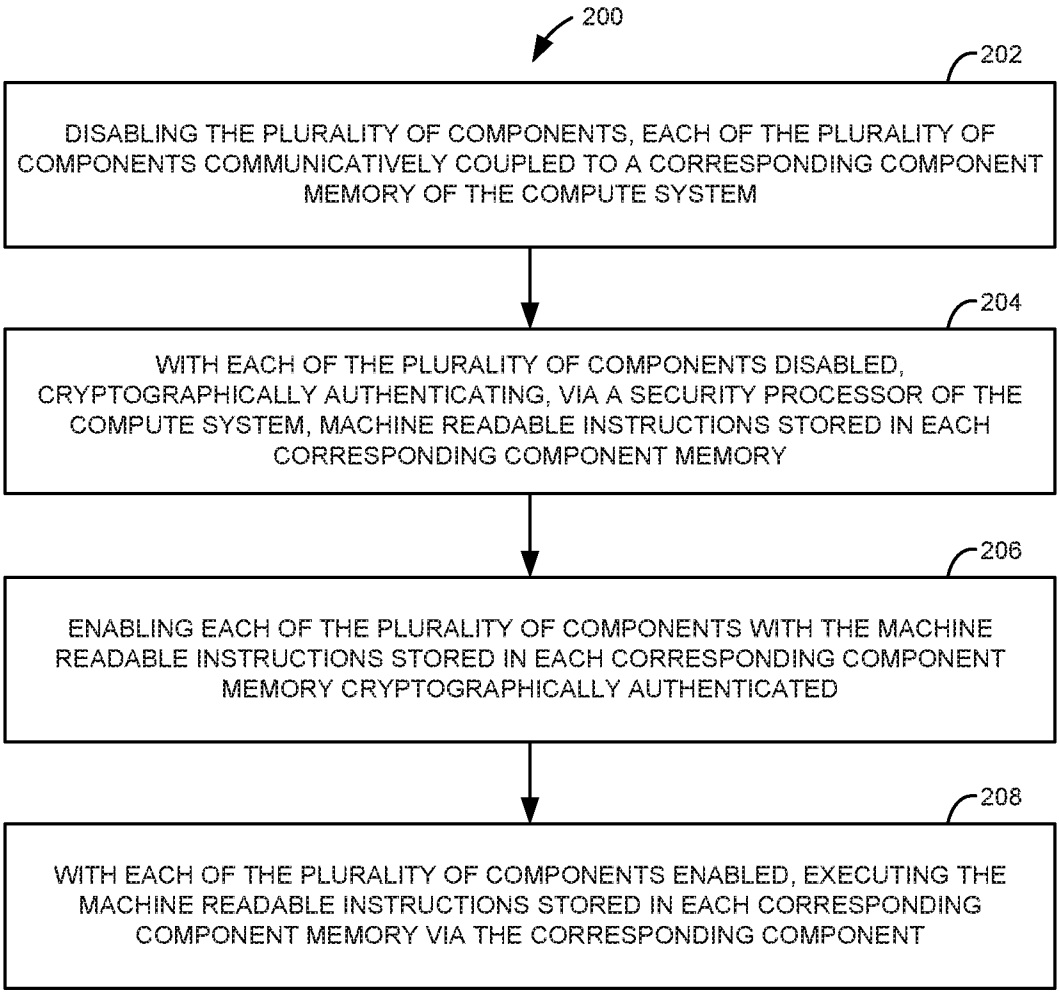
200
202
DISABLING THE PLURALITY OF COMPONENTS, EACH OF THE PLURALITY OF COMPONENTS COMMUNICATIVELY COUPLED TO A CORRESPONDING COMPONENT MEMORY OF THE COMPUTE SYSTEM
204
WITH EACH OF THE PLURALITY OF COMPONENTS DISABLED, CRYPTOGRAPHICALLY AUTHENTICATING, VIA A SECURITY PROCESSOR OF THE COMPUTE SYSTEM, MACHINE READABLE INSTRUCTIONS STORED IN EACH CORRESPONDING COMPONENT MEMORY
206
ENABLING EACH OF THE PLURALITY OF COMPONENTS WITH THE MACHINE READABLE INSTRUCTIONS STORED IN EACH CORRESPONDING COMPONENT MEMORY CRYPTOGRAPHICALLY AUTHENTICATED
208
WITH EACH OF THE PLURALITY OF COMPONENTS ENABLED, EXECUTING THE MACHINE READABLE INSTRUCTIONS STORED IN EACH CORRESPONDING COMPONENT MEMORY VIA THE CORRESPONDING COMPONENT

Fig. 2A

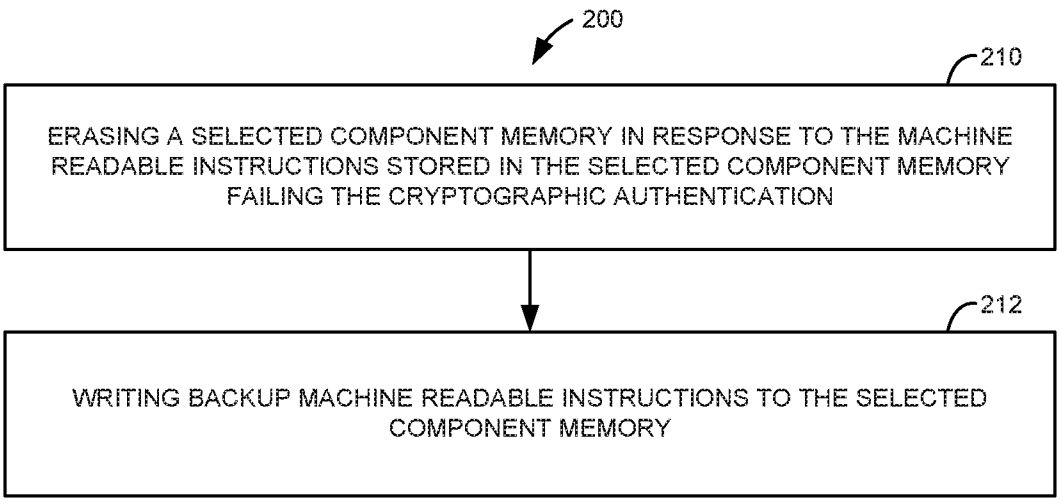
200
210
ERASING A SELECTED COMPONENT MEMORY IN RESPONSE TO THE MACHINE READABLE INSTRUCTIONS STORED IN THE SELECTED COMPONENT MEMORY FAILING THE CRYPTOGRAPHIC AUTHENTICATION
212
WRITING BACKUP MACHINE READABLE INSTRUCTIONS TO THE SELECTED COMPONENT MEMORY

Fig. 2B

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 2020/041994

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/30 *(2013.01)*
G06F 21/72 *(2013.01)*

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 9/00-9/445, 12/00, 13/00, 21/00-21/82, H04L 9/00-9/08

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)

PatSearch (RUPTO internal), USPTO, PAJ, Esp@cenet, Information Retrieval System of FIPS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	US 2015/0121054 A1 (ADVANCED MICRO DEVICES, INC.) 30.04.2015, abstract, paragraphs [0015]-[0028], [0032]-[0040], [0045], claims 1, 3, 5, fig. 1	1-3, 5-10, 12-14 4, 11, 15
A	US 8726364 B2 (INTEL CORPORATION) 13.05.2014	1-15
A	US 9479331 B2 (APPLE INC.) 25.10.2016	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

"D" document cited by the applicant in the international application

"E" earlier document 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
01 March 2021 (01.03.2021)	11 March 2021 (11.03.2021)

Name and mailing address of the ISA/RU:	Authorized officer
Federal Institute of Industrial Property, Berezhkovskaya nab., 30-1, Moscow, G-59, GSP-3, Russia, 125993 Facsimile No: (8-495) 531-63-18, (8-499) 243-33-37	D. Starshinov Telephone No. (495) 531-64-81