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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
3 July 2014 (03.07.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/105310 A1

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

(21) International Application Number:
PCT/US2013/071346

(22) International Filing Date:
21 November 2013 (21.11.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/730,469 28 December 2012 (28.12.2012) US

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

(72) Inventors; and

(71) Applicants *(for US only)*: **KOEBERL, Patrick** [DE/DE]; Schlossstrasse 5, 64665 Alsbach-Haenlein (DE). **LI, Jiangtao** [CN/US]; 12572 SW Sheldrake Way, Beaverton, Oregon 97007 (US).

(74) Agent: **DALGLISH, Leslie E.**; Patent Capital Group, C/O CPA Global, P.O. Box 52050, Minneapolis, Minnesota 55402 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, 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,

[Continued on next page]

(54) Title: DEVICE AUTHENTICATION USING A PHYSICALLY UNCLONABLE FUNCTIONS BASED KEY GENERATION SYSTEM

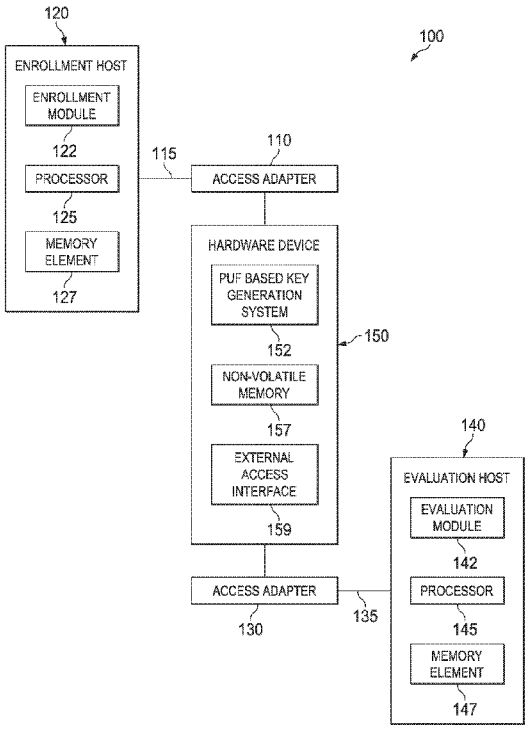


FIG. 1

(57) Abstract: At least one machine accessible medium having instructions stored thereon for authenticating a hardware device is provided. When executed by a processor, the instructions cause the processor to receive two or more device keys from a physically unclonable function (PUF) on the hardware device, generate a device identifier from the two or more device keys, obtain a device certificate from the hardware device, perform a verification of the device identifier, and provide a result of the device identifier verification. In a more specific embodiment, the instructions cause the processor to perform a verification of a digital signature in the device certificate and to provide a result of the digital signature verification. The hardware device may be rejected if at least one of the device identifier verification and the digital signature verification fails.

WO 2014/105310 A1

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).

Published:

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

DEVICE AUTHENTICATION USING A PHYSICALLY UNCLONABLE FUNCTIONS BASED KEY GENERATION SYSTEM

TECHNICAL FIELD

[0001] This disclosure relates in general to the field of semiconductors and, more particularly, to device authentication using a physically unclonable functions (PUF) based key generation system.

BACKGROUND

[0002] The contamination of electronic component supply chains by counterfeit hardware devices is a serious and growing risk in today's globalized marketplace. Copying and counterfeiting devices can occur at different levels. Re-marking is a low-technology technique of counterfeiting a device and accounts for the bulk of the counterfeits detected. In a typical re-marking attack, a device's product markings are misrepresented by replacing the original identification markings with new identification markings indicating a higher specification or higher value part. Such a device, if embedded in an electronic product or system, may fail in the field when subjected to an operational environment for which the original part was not designed. Device failures can lead to the decreased reputation of the device manufacturer. Other counterfeiting techniques can include attempts to clone a device, for example, by copying a fabrication mask or by reverse engineering a device. In these instances, counterfeiters could profit from a remarked device or from a chip design via the unauthorized sale of the cloned chip. Thus, profit margins for legitimate manufacturers of such devices could be detrimentally affected. Additionally, the risk of counterfeit products entering the supply chain generally increase when devices suffer supply shortfalls or have production terminated by the manufacturer. Thus, more effective techniques are needed to protect against unauthorized copying and counterfeiting of hardware devices.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] To provide a more complete understanding of the present disclosure and features and advantages thereof, reference is made to the following description, taken in

conjunction with the accompanying figures, wherein like reference numerals represent like parts, in which:

[0004] FIGURE 1 is a simplified block diagram illustrating a device authentication system using a physically unclonable functions based key generation system, according to an example embodiment;

[0005] FIGURE 2 is a simplified interaction diagram of a hardware device during an enrollment phase of the authentication system according to an embodiment;

[0006] FIGURE 3 is a simplified interaction diagram of a hardware device during an evaluation phase of the authentication system according to an embodiment;

[0007] FIGURE 4 is a simplified flow chart illustrating example operations that may be associated with embodiments of the present disclosure;

[0008] FIGURE 5 is a simplified flow chart illustrating further example operations that may be associated with an embodiment of the present disclosure;

[0009] FIGURE 6 is a block diagram of an exemplary processor according to an embodiment of the present disclosure;

[0010] FIGURE 7 is a block diagram of an exemplary mobile device system according to an embodiment of the present disclosure; and

[0011] FIGURE 8 is a block diagram of an exemplary computing system according to an embodiment of the present disclosure.

DETAILED DESCRIPTION OF EXAMPLE EMBODIMENTS

EXAMPLE EMBODIMENTS

[0012] FIGURE 1 is a simplified block diagram illustrating a device authentication system 100 in accordance with an example embodiment. Authentication system 100 can include an enrollment host 120, an evaluation host 140, and a hardware device 150 to be certified and authenticated. Access adapter 110 can facilitate communication between hardware device 150 and enrollment host 120. Access adapter 130 can facilitate communication between hardware device 150 and evaluation host 140. Communication links 115 and 135 can be configured in any suitable form to enable communication between access adapter 110 and enrollment host 120, and between access adapter 130 and evaluation host

140, respectively. Enrollment host 120 can include an enrollment module 122, a processor 125, and a memory element 127. Evaluation host 140 can include an evaluation module 142, a processor 145, and a memory element 147. Hardware device 150 can include a physically unclonable functions (PUF) based key generation system 152, non-volatile memory 157, and an external access interface 159 that enables access to PUF based key generation system 152.

[0013] In an example implementation, authentication system 100 can be associated with a device manufacturer of hardware device 150 via enrollment host 120 and access adapter 110, when the hardware device is manufactured. Authentication system 100 can also be associated with a manufacturer of other electronic devices (also referred to herein as a 'verifier') via evaluation host 140 and access adapter 130, when hardware device 150 is used to produce other electronic devices.

[0014] For purposes of illustrating the techniques of authentication system 100, it is important to understand the activities and security concerns that may be present in electronic component supply chains and hardware devices in those supply chains, as shown in FIGURE 1. The following foundational information may be viewed as a basis from which the present disclosure may be properly explained. Such information is offered for purposes of explanation only and, accordingly, should not be construed in any way to limit the broad scope of the present disclosure and its potential applications.

[0015] Counterfeit semiconductor devices often enter the electronic component supply chain in manufacturing environments during the assembly phase of an electronic device (e.g., personal computer, laptop, mobile device such as a smartphone, tablet, eBook reader, etc., gaming system, etc.). Generally, a variety of semiconductor devices, also referred to herein as 'hardware devices', such as integrated circuits or chips, are used in the assembly phase to build a given electronic device. Counterfeit chips can cause an electronic device to fail if the chip does not meet the required specifications. In addition, product failures can lead to a diminished reputation of the electronic device and its manufacturer. A diminished reputation often translates to lost revenue and decreased market share. Accordingly, manufacturers of electronic devices want assurances that the chips being used to build the electronic devices are authentic and not counterfeit chips.

[0016] Current practices for detecting counterfeit semiconductor devices include manual inspection such as visual checking, electronic testing, reliability testing, and destructive testing, which can require significant investments in expertise, equipment, and time. Additionally, these methods can be very expensive. Moreover, such methods cannot guarantee the provenance or performance of a device and, in many cases, it may only be feasible to perform testing on a sample of devices, for example when tests are destructive.

[0017] Some methods of device authentication might not be able to protect against certain types of counterfeiting such as cloning. For instance, a unique identifier in a register may not protect against cloning a device. If a unique ID is stored in fuses or some type of non-volatile memory, a counterfeiter could clone the device and store a different unique identifier in the register.

[0018] Other anti-counterfeiting techniques may have other drawbacks. For instance, one commonly used anti-counterfeiting technique is the CPU_ID (central processing unit identifier) mechanism, which supports identification of CPU features. However, the mechanism is not sufficient to differentiate two individual instances of the same device type.

[0019] A physically unclonable function (PUF), may be used as a foundation of a security primitive to build a more effective authentication scheme. A PUF is a physical system that, when measured or challenged, provides unique, repeatable and unpredictable responses. Creating a physical copy of the PUF with an identical challenge-response behavior is difficult, resulting in a structure which is unclonable even by the manufacturer.

[0020] Silicon PUF implementations leverage the complementary metal-oxide-semiconductor (CMOS) manufacturing technology used to fabricate the majority of integrated circuits (ICs) today. Silicon PUFs exploit the uncontrollable manufacturing variations that are a result of integrated circuit fabrication processes. Manufacturing variation of parameters, such as dopant concentrations and line widths, manifest themselves as differences in timing behavior between instances of the same integrated circuit design. These timing differences can be measured using a suitable circuit to extract a fingerprint for the chip.

[0021] Many different silicon PUFs have been proposed. For example, an arbiter PUF compares the relative delay of two delay paths using a series of configurable delay elements terminated by an arbiter. By using a PUF challenge as the delay element configuration vector, the circuit exhibits a challenge space which is exponential in the number of challenge bits.

[0022] A ring oscillator PUF compares the relative frequencies of self-oscillating delay loops in order to generate PUF responses. A single response bit can thus be generated by a pair of oscillators

[0023] Another PUF type is based on the power-up state of uninitialized six-transistor SRAM cells. The storage mechanism in an SRAM cell consists of four cross-coupled transistors that assume one of two stable states after power-up. Which state the cell enters is largely determined by the relative characteristics of the transistors, so any mismatch causes the cell to have a bias to one of the states. The mismatch is fixed at manufacturing time, resulting in a cell that tends to power up in the same state. The power-up behavior is random between cells, but robust for a single cell, resulting in a structure that is well suited for use as a PUF. The challenge in the case of an SRAM PUF can be considered to be a set of SRAM addresses, and the response the contents of those addresses post power-up.

[0024] While some standardized methods for providing device traceability and authentication have been defined, at least some of these are serialization mechanisms based on the generation of unpredictable, random codes and are intended to be applied at the device package and higher levels. Authentication methods using PUFs may require on-line access to secure manufacturer databases which could constrain their deployment in production facilities. Additionally, PUFs with large challenge-response pairs may be required. Other mechanisms may require a large number of PUF instances in a single device in order to be robust against hardware simulator attacks. Also, mechanisms that allow off-line authentication can have some amount of false acceptance rates and false rejection rates if the device identifier is changeable (e.g., under different temperatures, voltages, and/or thermal noise).

[0025] Authentication system 100 of FIGURE 1 resolves many of the aforementioned issues (and more). Authentication system 100 prevents counterfeiting of hardware devices by using a physically unclonable functions based key generation system to generate and derive keys for the hardware devices. A PUF based key generation system is embedded in a hardware device during its manufacturing. During an enrollment phase of authentication system 100, a large number of keys (e.g., 1000 keys, etc.) are generated as the intrinsic hardware identifier of the hardware device. The device manufacturer certifies the device by storing a device certificate in the hardware device. The hardware device may be shipped to a

verifier (e.g., a manufacturer of other electronic devices in which the hardware device is integrated) via an untrusted supply chain. During an evaluation phase of authentication system 100, the verifier authenticates the hardware device and validates that the hardware device is a genuine device from the original device manufacturer. Authentication can be performed by generating the same keys from the hardware device to verify the device identifier, and by using the manufacturer public key to verify the signature on the device certificate.

[0026] Turning to the infrastructure of FIGURE 1, a brief description is provided about some of the possible infrastructure that may be included in authentication system 100. Authentication system 100 may span multiple, possibly discrete, network environments in example embodiments. A network environment for an enrollment phase of authentication system 100 may be provided, at least in part, by enrollment host 120 and access adapter 110. Hardware device 150 may be connected to access adapter 110 during the enrollment phase. A separate network environment for an evaluation phase of authentication system 100 may be provided, at least in part, by evaluation host 140 and access adapter 130. Hardware device 150 may be connected to access adapter 130 during the evaluation phase.

[0027] Enrollment host 120 and evaluation host 140 are computing systems of authentication system 100 that provide enrollment (or certification) functions and evaluation (or authentication) functions, respectively. These functions are provided for hardware devices, such as hardware device 150. As used herein, 'computing systems' are intended to encompass servers, appliances, personal computers, laptops, mobile devices, processors, modules, or any other suitable device, component, element, proprietary appliance, or object operable to exchange information in a network environment.

[0028] Hosts 120 and 140 can include logic to achieve (or to foster) the enrollment and evaluation functions, as outlined herein. Each of these elements can have an internal structure (e.g., processors 125, 145, memory elements 127, 147, etc.) to facilitate some of the operations described herein. Computing systems and other network elements (e.g., routers, gateways, switches, bridges, loadbalancers, firewalls, adapters, etc.) may include any suitable algorithms, hardware, software, components, modules, interfaces, communication protocols, or objects that facilitate the operations thereof and that enable receiving, transmitting, and/or otherwise communicating data or information in a network environment.

[0029] In other embodiments, these enrollment and evaluation functions may be executed, in whole or in part, externally to these elements or included in some other computing system to achieve this intended functionality. For instance, enrollment functions could be provided in a server or appliance (e.g., serving a device manufacturer facility) and could communicate with host 120 via a network. The server or appliance could also communicate with multiple other hosts (e.g., hosts that access hardware devices at a device manufacturer facility). Similarly, evaluation functions could be provided in a server or appliance (e.g., serving a verifier that receives hardware devices from device manufacturers) and could communicate with host 140 via a network. The server or appliance could also communicate with multiple other hosts (e.g., hosts accessing hardware devices at a verifier's facility).

[0030] Networks enabling communication between hosts and other computing systems or network elements could each represent a series of points or nodes of interconnected communication paths for receiving and transmitting packets of information. These networks offer a communicative interface between nodes, and may be configured as local area networks (LANs), wireless local area networks (WLANs), wide area networks (WANs), Intranet, Extranet, virtual private networks (VPNs), or any other suitable network configuration, or any combination thereof, including wired and wireless networks. Additionally, radio signal communications over a cellular network may also be provided in authentication system 100, and suitable interfaces and infrastructure may be provided to enable communication with the cellular network.

[0031] Some portions of authentication system 100 may include a configuration capable of transmission control protocol/internet protocol (TCP/IP) communications for the transmission and/or reception of packets in a network. Some portions of authentication system 100 may also operate in conjunction with a user datagram protocol/IP (UDP/IP) or any other suitable protocol, where appropriate and based on particular needs. The term 'data' as used herein, refers to any type of binary, numeric, voice, video, media, textual, or script data, or any type of source or object code, or any other suitable information in any appropriate format that may be communicated from one point to another in electronic devices, hardware devices, and/or networks.

[0032] In the enrollment phase, enrollment host 120 is connected to hardware device 150 through some type of adapter, represented by access adapter 110. In the evaluation phase, evaluation host 140 is connected to hardware device 150 through some type of adapter, represented by access adapter 130. One common way of accessing a PUF based key generation system on a hardware device is the Joint Test Action Group (JTAG) standard, which is formally referred to as the Institute of Electrical and Electronics Engineers 1149.1-2001 Standard Test Access Port and Boundary-Scan Architecture, 7/23/2001. Access adapters 110 and 130 can be JTAG adapters, and can be connected to enrollment host 120 and evaluation host 130, respectively, via any suitable interfaces 115 and 135. Interfaces 115 and 135 can include, but are not limited to, Ethernet, peripheral component interconnect (PCI), and universal serial bus (USB).

[0033] The JTAG standard defines a generic external test access port, which is represented by external access interface 159 in hardware device 150. This external test access port can provide an interface to PUF based key generation system 152 during the enrollment phase, and also during the evaluation phase. Its architecture allows the evaluation phase to occur post assembly on fully populated boards. JTAG is often used to perform boundary scan testing to determine whether openings or shorts were created during the soldering process of a printed circuit board (PCB). JTAG can be used in authentication system 100 to read keys from PUF based key generation system 152 and to read the device certificate from non-volatile memory 157 of the hardware device. Although JTAG is one example industry standard that could be used to obtain the needed information (e.g., keys and device certificate), other standards could be implemented as well.

[0034] Hardware device 150 can be any semiconductor device on which a PUF based key generation system can be embedded. Examples of hardware device 150 include, but are not limited to, integrated circuits, central processing units (CPUs), microprocessors, chipsets, system on a chip (SoCs), random access memory (RAM), read-only memory (ROM), etc.

[0035] PUF based key generation system 152 is embedded on hardware device 150 and is configured to generate a large number of keys. The keys can be used by authentication system 100 as the intrinsic hardware identifier of hardware device 150. As the number of generated keys increases, the security of the system also increases. If the number of keys becomes too large, however, system performance may be affected such that the certification

and authentication activities could potentially experience delays. In an example implementation, PUF based key generation system 152 could generate in the range of 500 to 5000 keys. This range is provided for illustrative purposes, and it will be apparent that authentication system 100 could be configured to generate a number of keys outside the illustrative range. Thus, although many keys may be used by authentication system 100 to authenticate a single hardware device, the actual number of keys that can be generated is flexible and can be configured based on particular requirements and needs of device manufacturers and/or verifiers.

[0036] A PUF based key generation system uses PUFs as an underlying static entropy source from which to generate one or more PUF root keys. The underlying static entropy source is platform-unique and externally-unknown. Because PUF bits may not be completely static and may not have full entropy, PUF bits are not directly used as a PUF root key. Instead, PUF bits are generally post-processed first. A post processing function (also referred to as a fuzzy extractor) can apply error correction and entropy extraction to the PUF bits. Within the key generation system, more keys can be derived from the PUF root keys using cryptographic key derivation functions.

[0037] In one example, a Logically Reconfigurable PUF (LR-PUF) enables multiple outputted keys to be cryptographically derived from the PUF value. A logically reconfigurable PUF provides challenge/response behavior that depends on both the physical properties of the PUF and on the logical state maintained by control logic. Thus, by updating the state of a reconfigurable PUF, its challenge/response behavior can be dynamically changed. Updating the state to dynamically change the challenge/response behavior can be achieved after the reconfigurable PUF has been deployed in a hardware device.

[0038] In a particular example of a reconfigurable PUF, a state is stored in non-volatile memory and is maintained by control logic. The control logic includes an algorithm that queries the LR-PUF and an algorithm that reconfigures the LR-PUF. The query algorithm computes a challenge, evaluates a response, and returns a result. The reconfiguring algorithm reconfigures the LR-PUF by changing the current state to a new independent state. An adversary can know the current and previous states of the LR-PUF, but cannot change the state to a previous LR-PUF state.

[0039] Other PUF based key generation systems may produce a single key, rather than multiple keys. For these systems, a cryptographic key derivation in hardware may be implemented to derive a large number (potentially unlimited) of keys. Additionally, any other PUF based key generation system, from which a large number of keys can be generated, may be embedded in hardware device 150 to generate keys during enrollment and evaluation phases of authentication system 100.

[0040] Domain separation of PUF-based key generation system 152 may be provided to prevent a verifier in the supply chain, or another entity that gains access to the hardware device, from accessing the PUF-based key generation system and learning the keys of the platform. Accordingly, separate physical instances of the PUF-based key generation system can be implemented in a single hardware device. One instance can be dedicated for certification and authentication, while the other instance can be dedicated for internal platform keys. External access interface 159 may provide external access to the PUF-based key generation system dedicated to certification and authentication, but not to the PUF-based key generation system dedicated to internal platform keys. This configuration ensures that verifiers in the supply chain (and others) cannot access internal platform keys.

[0041] Non-volatile memory 157 may be used to store a device certificate generated by enrollment host 120. Examples of non-volatile memory include, but are not limited to, flash memory and fuses. However, not all hardware devices have non-volatile memory. In this scenario, the device certificate could be stored on the IC package using 2D data matrix (e.g., Data Matrix Error Checking and Correction (ECC) 200 Standard). By using 2D data matrix, a 44x44 matrix can encode 1136-bit. At a resolution of 600DPI, the data matrix only takes approximately 8mmx8mm space.

[0042] Turning to FIGURE 2, an interaction diagram of an enrollment phase of authentication system 100 is illustrated. The enrollment phase involves certifying a hardware device by a device manufacturer. Certification may be accomplished using a public/private key pair and a unique device identifier. Assume (mpk, msk) are the device manufacturer's public verification key and private signing key pair. The device manufacturer embeds PUF based key generation system 152 into hardware device 150. The manufacturer queries PUF based key generation system 152 to obtain n-number of device keys. The manufacturer hashes, at 160, the outputted n keys into a smaller device identifier (id_D), also referred to

herein as 'device ID'. Any robust, cryptographic hash function may be used. The manufacturer uses the device manufacturer's private signing key (mpk) to sign id_D, at 170, to create digital signature σ. The manufacturer sets device certificate (id_D, σ) 180, and stores the device certificate in non-volatile memory 157 of hardware device 150.

[0043] Public key cryptology is a mechanism in which a mathematically linked key pair, including a private signing key and a public key, can be used to secure data being sent from a sender to a receiver and to verify the authenticity of the data. A digital signature is a mechanism in which the data is proved to have originated from a particular sender. The private key, which may be known only to the sender, can be used to encrypt the data, or a portion thereof, to generate the digital signature. In authentication system 100, the device identifier (id_D) can be the data that is encrypted to create the digital signature (σ). The encrypted data or digital signature can be sent to a receiving entity (e.g., the verifier). The receiving entity can use the public key of the key pair to decrypt the digital signature and verify the authenticity of the data. The public key may be sent to the receiving entity along with the digital signature or may be otherwise provided to the receiving entity.

[0044] Turning to FIGURE 3, an interaction diagram of an evaluation phase of authentication system 100 is illustrated. The evaluation phase involves verifying a hardware device by a verifier (e.g., a manufacturer of electronic devices in which the hardware device is integrated). Once the verifier obtains hardware device 150 from the supply chain, the hardware device can be verified. The verifier can query PUF based key generation system 152 to obtain n keys, which should be the same n keys that were obtained in the enrollment phase. The verifier hashes, at 160, the outputted n keys into a smaller device identifier (id_D'). The same cryptographic hash function used during the enrollment phase is used during the evaluation phase. The verifier reads device certificate (id_D, σ) 180 from non-volatile memory 157 of hardware device 150. The verifier can verify, at 190, the device identity by checking whether id_D matches id_D'. If they do not match, the verifier may reject hardware device 150 as not being authenticated. The verifier can also use the device manufacturer's public verification key (mpk) to verify, at 190, the digital signature σ on id_D. If the signature verification fails, then the verifier may reject hardware device 150 as not being authenticated.

[0045] The queries to the PUF based key generation system to obtain n device keys can be the same query in both the enrollment phase (e.g., by the device manufacturer) and

the evaluation phase (e.g., by the verifier). The number of keys can be set as n=512, n=1024, or n=2048, for example. The queries can be hardcoded in the hardware, or can be provided to hardware device 150 during the query. For instance, the first query can be set as 'anti-counterfeiting-1' to obtain key_1, the second query can be set as 'anti-counterfeiting-2' to obtain key_2, etc. Also, the queries can be public.

[0046] The use of a PUF based key generation system to generate multiple device keys for a digital signature scheme is an effective and reliable anti-counterfeiting solution for hardware devices. In example embodiments, the device certificate is digitally signed and bound to the device ID. Given the collision-resistant property of the hash function (e.g., SHA-256), an attacker has to forge all of the n keys from the PUF based key generation system in order for the forged device to be successfully verified. Because the PUF is unclonable, an attacker of authentication system 100 must be capable of simulating all of the device keys in hardware and outputting the device keys to the verifier during device authentication. If the number of keys is reasonably large, then the expense of hardcoding the keys in hardware can make an attack uneconomic for a potential attacker.

[0047] Additionally, authentication system 100 may be simple and inexpensive to implement. It may not require any secure, on-line database access during the evaluation phase. Any additional non-volatile storage potentially needed for the device may be small, in some embodiments, and thus, a cost effective solution may be implemented. In addition, authentication system 100 may only need a small amount of PUF circuitry used for a PUF based key generation system. Unlike many PUF applications, the PUF queries and device certificates can be public, and unprotected in some embodiments. Additionally, error correction or fuzzy extractors may not be needed in some embodiments. Also, the false acceptance and rejection rates of an authentication system using a PUF based key generation system are minimal to zero.

[0048] Turning to FIGURE 4, an example flow 400 of operations is illustrated, which may be associated with the enrollment phase of an embodiment of authentication system 100. In one example implementation, some or all of flow 400 may be performed by enrollment module 122 of enrollment host 120. Flow 400 may be implemented in hardware, software, firmware, or any suitable combination thereof.

[0049] In the enrollment phase, a hardware device is certified by the device manufacturer. Assume (mpk, msk) are the device manufacturer's public verification key and private signing key of a key pair. In an example embodiment, the manufacturer embeds a PUF based key generation system in the hardware device. The PUF based key generation system may have other usages. In other embodiments, however, the PUF based key generation system may be dedicated to providing device keys for certification and authentication while another PUF system may be embedded in the hardware device and dedicated to other usages (e.g., platform keys).

[0050] At 402, the device manufacturer (e.g., via enrollment module 122 of enrollment host 120) queries the PUF based key generation system embedded in the hardware device for n keys. For example, the number of keys may be set at 512, 1024, or 2048. These queries may be public, and can be hardcoded in the hardware or can be provided to the hardware device during the query. A query may be made for each key of the n keys.

[0051] After receiving all of the n keys from the PUF based key generation system, at 404, the device manufacturer hashes all of the outputted n keys into a smaller device ID (id_D). A cryptographic hash function may be applied to the n keys to obtain the device ID (id_D). A cryptographic hash function is an algorithm that takes an arbitrary block of data and returns a fixed-size bit string, such that any change to the data is very likely to result in a change to the hash value. An example of a robust cryptographic hash function that could be used includes, but is not limited to, SHA-256, designed by the National Security Agency (NSA) and published in 2002 by the National Institute of Standards and Technology (NIST) as a U.S. Federal Information Processing Standard (FIPS).

[0052] At 406, the manufacturer can use its private signing key (msk) to sign the device ID (id_D) and create a digital signature σ. At 408, the manufacturer creates the device certificate (id_D, σ) based on the device ID and the digital signature. At 410, the manufacturer stores the device certificate in the non-volatile memory of the hardware device.

[0053] Turning to FIGURE 5, an example flow 500 of operations is illustrated, which may be associated with the evaluation phase of an embodiment of authentication system 100. In one example implementation, some or all of flow 500 may be performed by

evaluation module 142 of evaluation host 140. Flow 400 may be implemented in hardware, software, firmware, or any suitable combination thereof.

[0054] In the evaluation phase, a verifier receives a hardware device presumably from the supply chain (e.g., from a device manufacturer). In a counterfeiting attack, however, the hardware device may be supplied by a counterfeiting entity. At 502, the verifier (e.g., via evaluation module 142 of evaluation host 140) queries the PUF based key generation system embedded in the hardware device for n keys. For example, the number of keys may be set at 512, 1024, or 2048. The number of keys is set to the same amount as the number of keys that are set in the enrollment phase. These queries may be public, and can be hardcoded in the hardware or can be provided to the hardware device during the query. A query may be made for each key of the n keys.

[0055] At 504, the verifier hashes all of the outputted n keys into a smaller device ID (id_D'). The verifier uses the same cryptographic hash function that is used by the enrollment module to hash the multiple device keys into a smaller device ID. At 506, the verifier can read the device certificate (id_D, σ). For example, the device certificate can be read from the non-volatile memory of the hardware device.

[0056] At 508, the verifier verifies the device identity. If the device ID generated in the enrollment phase does not match the device ID from the evaluation phase (i.e., $id_D \neq id_D'$), then the verification fails. A result that indicates the device identifier verification failed may be provided to the verifier in any appropriate manner (e.g., a report, a display screen message or alert, a text message, an email, etc.). If the device identifier verification fails, then the hardware device may be rejected by the verifier at 514, and not integrated into an electronic device being built.

[0057] Otherwise, if the device ID generated in the enrollment phase matches the device ID from the evaluation phase (i.e., $id_D = id_D'$), then at 510, the digital signature on the device certificate can be verified. The digital signature can be verified using the manufacturer public key (mpk). If the digital signature verification fails, a result that indicates the digital signature verification failed may be provided to the verifier in any appropriate manner (e.g., a report, a display screen message or alert, a text message, an email, etc.). If the digital signature verification fails, then the hardware device may be rejected by the verifier at 514, and not integrated into an electronic device being built. However, if the digital signature is

verified at 510, and if the device identity is verified at 508, then the hardware device may be accepted by the verifier at 512, and integrated into an electronic device.

[0058] FIGURES 6-8 are block diagrams of exemplary computer architectures that may be used in accordance with embodiments disclosed herein. Other computer architecture designs known in the art for processors, mobile devices, and computing systems may also be used. Generally, suitable computer architectures for embodiments disclosed herein can include, but are not limited to, configurations illustrated in FIGURES 6-8.

[0059] FIGURE 6 is an example illustration of a processor according to an embodiment. Processor 600 is one example embodiment of processors 125 and 145 of enrollment host 120 and evaluation host 140, respectively. Additionally, processor 600 is an example of a type of hardware device that can be authenticated by authentication system 100.

[0060] Processor 600 may be any type of processor, such as a microprocessor, an embedded processor, a digital signal processor (DSP), a network processor, a multi-core processor, a single core processor, or other device to execute code. Although only one processor 600 is illustrated in FIGURE 6, a processing element may alternatively include more than one of processor 600 illustrated in FIGURE 6. Processor 600 may be a single-threaded core or, for at least one embodiment, the processor 600 may be multi-threaded in that it may include more than one hardware thread context (or "logical processor") per core.

[0061] FIGURE 6 also illustrates a memory 602 coupled to processor 600 in accordance with an embodiment. Memory 602 may be any of a wide variety of memories (including various layers of memory hierarchy) as are known or otherwise available to those of skill in the art. Such memory elements can include, but are not limited to, random access memory (RAM), read only memory (ROM), logic blocks of a field programmable gate array (FPGA), erasable programmable read only memory (EPROM), and electrically erasable programmable ROM (EEPROM).

[0062] Processor 600 can execute any type of instructions associated with the certification and authentication operations detailed herein. Generally, processor 600 can transform an element or an article (e.g., data) from one state or thing to another state or thing.

[0063] Code 604, which may be one or more instructions to be executed by processor 600, may be stored in memory 602, or may be stored in software, hardware, firmware, or any suitable combination thereof, or in any other internal or external component, device, element, or object where appropriate and based on particular needs. In one example, processor 600 can follow a program sequence of instructions indicated by code 604. Each instruction enters a front-end logic 606 and is processed by one or more decoders 608. The decoder may generate, as its output, a micro operation such as a fixed width micro operation in a predefined format, or may generate other instructions, microinstructions, or control signals that reflect the original code instruction. Front-end logic 606 also includes register renaming logic 610 and scheduling logic 612, which generally allocate resources and queue the operation corresponding to the instruction for execution.

[0064] Processor 600 can also include execution logic 614 having a set of execution units 616_1 through 616_m. Some embodiments may include a number of execution units dedicated to specific functions or sets of functions. Other embodiments may include only one execution unit or one execution unit that can perform a particular function. Execution logic 614 performs the operations specified by code instructions.

[0065] After completion of execution of the operations specified by the code instructions, back-end logic 618 can retire the instructions of code 604. In one embodiment, processor 600 allows out of order execution but requires in order retirement of instructions. Retirement logic 620 may take a variety of known forms (e.g., re-order buffers or the like). In this manner, processor 600 is transformed during execution of code 604, at least in terms of the output generated by the decoder, hardware registers and tables utilized by register renaming logic 610, and any registers (not shown) modified by execution logic 614.

[0066] Although not shown in FIGURE 6, a processing element may include other elements on a chip with processor 600. For example, a processing element may include memory control logic along with processor 600. The processing element may include I/O control logic and/or may include I/O control logic integrated with memory control logic. The processing element may also include one or more caches. In some embodiments, non-volatile memory (such as flash memory or fuses) may also be included on the chip with processor 600.

[0067] Referring now to FIGURE 7, a block diagram is illustrated of an example mobile device 700. Mobile device 700 is an example of a possible computing system of authentication system 100. In an embodiment, mobile device 700 operates as a transmitter and a receiver of wireless communications signals. Specifically, in one example, mobile device 700 may be capable of both transmitting and receiving cellular network voice and data mobile services. Mobile services include such functionality as full Internet access, downloadable and streaming video content, as well as voice telephone communications.

[0068] Mobile device 700 may correspond to a conventional wireless or cellular portable telephone, such as a handset that is capable of receiving "3G", or "third generation" cellular services. In another example, mobile device 700 may be capable of transmitting and receiving "4G" mobile services as well, or any other mobile service.

[0069] Examples of devices that can correspond to mobile device 700 include cellular telephone handsets and smartphones, such as those capable of Internet access, email, and instant messaging communications, and portable video receiving and display devices, along with the capability of supporting telephone services. It is contemplated that those skilled in the art having reference to this specification will readily comprehend the nature of modern smartphones and telephone handset devices and systems suitable for implementation of the different aspects of this disclosure as described herein. As such, the architecture of mobile device 700 illustrated in FIGURE 7 is presented at a relatively high level. Nevertheless, it is contemplated that modifications and alternatives to this architecture may be made and will be apparent to the reader, such modifications and alternatives contemplated to be within the scope of this description.

[0070] In an aspect of this disclosure, mobile device 700 includes a transceiver 702, which is connected to and in communication with an antenna. Transceiver 702 may be a radio frequency transceiver. Also, wireless signals may be transmitted and received via transceiver 702. Transceiver 702 may be constructed, for example, to include analog and digital radio frequency (RF) 'front end' functionality, circuitry for converting RF signals to a baseband frequency, via an intermediate frequency (IF) if desired, analog and digital filtering, and other conventional circuitry useful for carrying out wireless communications over modern cellular frequencies, for example, those suited for 3G or 4G communications. Transceiver 702 is connected to a processor 704, which may perform the bulk of the digital

signal processing of signals to be communicated and signals received, at the baseband frequency. Processor 704 can provide a graphics interface to a display element 708, for the display of text, graphics, and video to a user. Processor 704 may include an embodiment as shown and described with reference to processor 600 of FIGURE 6.

[0071] In an aspect of this disclosure, processor 704 may be a processor that can execute any type of instructions to achieve authentication of a hardware device using a PUF based key generation system, as detailed herein. Processor 704 may also be coupled to a memory element 706 for storing information to be used in achieving the authentication operations. Additional details of an example processor 704 and memory element 706 are subsequently described herein. In an example embodiment, mobile device 700 may be designed with a system-on-a-chip (SoC) architecture, which integrates many or all components of the mobile device into a single chip, in at least some embodiments.

[0072] In an aspect of this disclosure, memory element 706 of mobile device 700 may also include authentication system modules 712. For example, authentication system modules 712 can include enrollment module 122 when mobile device 700 functions as enrollment host 120. Authentication system modules 712 can include evaluation module 145 when mobile device 700 functions as evaluation host 140.

[0073] FIGURE 8 illustrates a computing system 800 that is arranged in a point-to-point (PtP) configuration according to an embodiment. In particular, FIGURE 8 shows a system where processors, memory, and input/output devices are interconnected by a number of point-to-point interfaces. Generally, one or more of the computing systems or hosts of authentication system 100 may be configured in the same or similar manner as computing system 800. For example, enrollment host 120 and evaluation host 140, shown and described herein, may each be configured in the same or similar manner as exemplary computing system 800.

[0074] Processors 870 and 880 may also each include integrated memory controller logic (MC) 872 and 882 to communicate with memory elements 832 and 834. In alternative embodiments, memory controller logic 872 and 882 may be discrete logic separate from processors 870 and 880. Memory elements 832 and/or 834 may store various data to be used by processors 870 and 880 in achieving operations associated with authentication of devices using a PUF based key generation system, as outlined herein.

[0075] Processors 870 and 880 may be any type of processor, such as those discussed with reference to processor 600 of FIGURE 6, and processors 125 and 145 of FIGURE 1. Processors 870 and 880 may exchange data via a point-to-point (PtP) interface 850 using point-to-point interface circuits 878 and 888, respectively. Processors 870 and 880 may each exchange data with a chipset 890 via individual point-to-point interfaces 852 and 854 using point-to-point interface circuits 876, 886, 894, and 898. Chipset 890 may also exchange data with a high-performance graphics circuit 838 via a high-performance graphics interface 839, using an interface circuit 892, which could be a PtP interface circuit. In alternative embodiments, any or all of the PtP links illustrated in FIGURE 8 could be implemented as a multi-drop bus rather than a PtP link.

[0076] At least one embodiment, as disclosed herein, may be at least partially provided within the processors 1102 and 1104. Other embodiments, however, may exist, at least partially, in other circuits, logic units, or devices within the authentication system 100 of FIGURE 1. Furthermore, other embodiments may be distributed throughout several circuits, logic units, or devices.

[0077] Chipset 890 may be in communication with a bus 820 via an interface circuit 896. Bus 820 may have one or more devices that communicate over it, such as a bus bridge 818 and I/O devices 816. Via a bus 810, bus bridge 818 may be in communication with other devices such as a keyboard/mouse 812 (or other input devices such as a touch screen, trackball, etc.), communication devices 826 (such as modems, network interface devices, or other types of communication devices that may communicate through a computer network 860), audio I/O devices 814, and/or a data storage device 828. Data storage device 828 may store code 830, which may be executed by processors 870 and/or 880. In alternative embodiments, any portions of the bus architectures could be implemented with one or more PtP links.

[0078] The computer system depicted in FIGURE 8 is a schematic illustration of an embodiment of a computing system that may be utilized to implement various embodiments discussed herein. It will be appreciated that various components of the system depicted in FIGURE 8 may be combined in a system-on-a-chip (SoC) architecture or in any other suitable configuration capable of achieving authentication of hardware devices using a PUF based key generation system, as provided herein.

[0079] The enrollment and evaluation functions of authentication system 100, outlined herein, may be implemented by logic encoded in one or more tangible media (e.g., embedded logic provided in an application specific integrated circuit (ASIC), digital signal processor (DSP) instructions, software (potentially inclusive of object code and source code) to be executed by a processor (e.g., processor 600) or other similar machine, etc.). The tangible media may be non-transitory in at least some embodiments. In some of these instances, memory (e.g., memory 602) can store data used for the operations described herein. This includes the memory being able to store software, logic, code, or processor instructions that are executed to carry out the activities described in this Specification. In an embodiment, the tangible media may be provided in each one of hosts 120 and 140.

[0080] Additionally, the information being tracked, sent, received, or stored in authentication system 100 could be provided in any database, register, table, cache, queue, control list, or storage structure, based on particular needs and implementations, all of which could be referenced in any suitable timeframe. Any of the memory items discussed herein should be construed as being encompassed within the broad term 'memory element.' Similarly, any of the potential processing elements, modules, and machines described in this Specification should be construed as being encompassed within the broad term 'processor.'

[0081] Note that with the numerous examples provided herein, interaction may be described in terms of two, three, four, or more network elements, computing systems, modules, and/or other components. However, this has been done for purposes of clarity and example only. It should be appreciated that the system can be consolidated in any suitable manner. Along similar design alternatives, any of the illustrated modules, nodes, elements, and other components of FIGURE 1 may be combined in various possible configurations, all of which are clearly within the broad scope of this Specification. It should be appreciated that the system of FIGURE 1 (and its teachings) is readily scalable and can accommodate a large number of components, as well as more complicated/sophisticated arrangements and configurations. Accordingly, the examples provided should not limit the scope or inhibit the broad teachings of system 100 as potentially applied to a myriad of other architectures.

[0082] It is also important to note that the operations described with reference to the preceding FIGURES illustrate only some of the possible scenarios that may be executed by, or within, the system. Some of these operations may be deleted or removed where

appropriate, or these operations may be modified or changed considerably without departing from the scope of the discussed concepts. In addition, the timing of these operations may be altered considerably and still achieve the results taught in this disclosure. The preceding operational flows have been offered for purposes of example and discussion. Substantial flexibility is provided by the system in that any suitable arrangements, chronologies, configurations, and timing mechanisms may be provided without departing from the teachings of the discussed concepts.

[0083] The following examples pertain to embodiments in accordance with this Specification. One or more embodiments may provide at least one machine accessible storage medium having instructions stored thereon for certifying a hardware device. The instructions, when executed by a processor, cause the processor to: receive two or more device keys from a physically unclonable function (PUF) on the hardware device; generate a device identifier from the two or more device keys; generate a digital signature based on the device identifier and a private key; create a device certificate based on the device identifier and the digital signature; and store the device certificate in a memory element.

[0084] In an example of an embodiment, the physically unclonable function is configured on the hardware device to generate the two or more device keys, and wherein the keys uniquely identify the hardware device.

[0085] In an example of an embodiment, the memory element is a non-volatile memory element included in the hardware device with the physically unclonable function.

[0086] In an example of an embodiment, the two or more device keys include between 500-5000 keys.

[0087] An example of an embodiment comprises further instructions that when executed by the processor cause the processor to query the physically unclonable function on the hardware device for each of the two or more device keys.

[0088] In an example of an embodiment, generating the device identifier includes applying a cryptographic hash algorithm to the two or more keys.

[0089] One or more embodiments include an apparatus for certifying a hardware device. The apparatus comprises a processor and an enrollment module executing on the processor. The enrollment module may be configured to: receive two or more device keys from a physically unclonable function (PUF) on the hardware device; generate a device

identifier from the two or more device keys; generate a digital signature based on the device identifier and a private key; create a device certificate based on the device identifier and the digital signature; and store the device certificate in a memory element.

[0090] In an example of an embodiment, the physically unclonable function is configured on the hardware device to generate the two or more device keys, and wherein the keys uniquely identify the hardware device.

[0091] In an example of an embodiment, the memory element is a non-volatile memory element included in the hardware device with the physically unclonable function.

[0092] In an example of an embodiment, the two or more device keys include between 500-5000 keys.

[0093] In an example of an embodiment, the enrollment module is further configured to query the physically unclonable function on the hardware device for each of the two or more device keys.

[0094] In an example of an embodiment, the enrollment module is further configured to apply a cryptographic hash algorithm to the two or more keys to generate the device identifier.

[0095] One or more embodiments may provide at least one machine accessible storage medium having instructions stored thereon for authenticating a hardware device. The instructions, when executed by a processor, cause the processor to: receive two or more device keys from a physically unclonable function (PUF) on the hardware device; generate a device identifier from the two or more device keys; obtain a device certificate from the hardware device; perform a verification of the device identifier; and provide a result of the device identifier verification.

[0096] In an example of an embodiment, the hardware device is rejected if the result indicates the device identifier verification failed.

[0097] An example of an embodiment comprises further instructions that when executed by the processor cause the processor to apply a cryptographic hash algorithm to the two or more keys to generate the device identifier.

[0098] An example of an embodiment comprises further instructions that when executed by the processor cause the processor to compare the device identifier to a previously generated device identifier to perform the device identifier verification.

[0099] In an example of an embodiment, if the device identifier and the previously generated device identifier do not match, the result indicates that the device identifier verification failed.

[0100] An example of an embodiment comprises further instructions that when executed by the processor cause the processor to: perform a verification of a digital signature in the device certificate; and provide a result of the digital signature verification.

[0101] In an example of an embodiment, the hardware device is rejected if the result indicates the digital signature verification failed.

[0102] An example of an embodiment comprises further instructions that when executed by the processor cause the processor to use a public key of a key pair to decrypt the digital signature to perform the digital signature verification.

[0103] An example of an embodiment comprises further instructions that when executed by the processor cause the processor to query the physically unclonable function on the hardware device for each of the two or more device keys.

[0104] One or more embodiments include an apparatus for authenticating a hardware device. The apparatus comprises a processor; and an enrollment module executing on the processor. The enrollment module may be configured to: receive two or more device keys from a physically unclonable function (PUF) on the hardware device; generate a device identifier from the two or more device keys; obtain a device certificate from the hardware device; perform a verification of the device identifier; and provide a result of the device identifier verification.

[0105] In an example of an embodiment, the hardware device is rejected if the result indicates the device identifier verification failed.

[0106] In an example of an embodiment, the evaluation module is further configured to apply a cryptographic hash algorithm to the two or more keys to generate the device identifier.

[0107] In an example of an embodiment, the evaluation module is further configured to compare the device identifier to a previously generated device identifier to perform the device identifier verification.

[0108] In an example of an embodiment, if the device identifier and the previously generated device identifier do not match, the result indicates that the device identifier verification failed.

[0109] In an example of an embodiment, the evaluation module is further configured to: perform a verification of a digital signature in the device certificate; and provide a result of the digital signature verification.

[0110] In an example of an embodiment, the hardware device is rejected if the result indicates the digital signature verification failed.

[0111] In an example of an embodiment, the evaluation module is further configured to use a public key of a key pair to decrypt the digital signature to perform the digital signature verification.

[0112] In an example of an embodiment, the evaluation module is further configured to query the physically unclonable function on the hardware device for each of the two or more device keys.

[0113] One or more embodiments may provide a method for certifying a hardware device. The method may comprise: receiving two or more device keys from a physically unclonable function on the hardware device; generating a device identifier from the two or more device keys; generating a digital signature based on the device identifier and a private key; creating a device certificate based on the device identifier and the digital signature; and storing the device certificate in a memory element.

[0114] One or more embodiments may provide a method for authenticating a hardware device. The method may comprise: receiving two or more device keys from a physically unclonable function on the hardware device; generating a device identifier from the two or more device keys; obtaining a device certificate from the hardware device; performing a verification of the device identifier; and providing a result of the device identifier verification.

[0115] One particular example implementation may include means for receiving two or more device keys from a physically unclonable function (PUF) on the hardware device; means for generating a device identifier from the two or more device keys; means for generating a digital signature based on the device identifier and a private key; means for creating a device certificate based on the device identifier and the digital signature; and

means for storing the device certificate in a memory element. The implementation may also include the physically unclonable function being configured on the hardware device to generate the two or more device keys, where the keys uniquely identify the hardware device. Additionally, in the implementation, the memory element may be a non-volatile memory element included in the hardware device with the physically unclonable function. Also in the implementation, the two or more device keys may include between 500-5000 keys. The implementation may further comprise means for querying the physically unclonable function on the hardware device for each of the two or more device keys. Additionally, the means for generating the device identifier includes means for applying a cryptographic hash algorithm to the two or more keys.

[0116] Another particular example implementation may include means for receiving two or more device keys from a physically unclonable function (PUF) on the hardware device; means for generating a device identifier from the two or more device keys; means for obtaining a device certificate from the hardware device; means for performing a verification of the device identifier; and means for providing a result of the device identifier verification. In the implementation, the hardware device may be rejected if the result indicates the device identifier verification failed. The means for generating the device identifier may include applying a cryptographic hash algorithm to the two or more keys. The means for performing the device identifier verification may include comparing the device identifier to a previously generated device identifier. In the implementation, if the device identifier and the previously generated device identifier do not match, the result may indicate that the device identifier verification failed. Additionally, the implementation may include means for performing a verification of a digital signature in the device certificate; and means for providing a result of the digital signature verification. In the implementation, the hardware device may be rejected if the result indicates the digital signature verification failed. The means for performing the digital signature verification may include using a public key of a key pair to decrypt the digital signature. The implementation may further include querying the physically unclonable function on the hardware device for each of the two or more device keys.

WHAT IS CLAIMED IS:

1. At least one machine accessible storage medium having instructions stored thereon for certifying a hardware device, the instructions when executed by a processor cause the processor to:

receive two or more device keys from a physically unclonable function (PUF) on the hardware device;

generate a device identifier from the two or more device keys;

generate a digital signature based on the device identifier and a private key;

create a device certificate based on the device identifier and the digital signature; and

store the device certificate in a memory element.

2. The at least one machine accessible storage medium of Claim 1, wherein the physically unclonable function is configured on the hardware device to generate the two or more device keys, and wherein the keys uniquely identify the hardware device.

3. The at least one machine accessible storage medium of Claim 1, wherein the memory element is a non-volatile memory element included in the hardware device with the physically unclonable function.

4. The at least one machine accessible storage medium of Claim 1, wherein the two or more device keys include between 500-5000 keys.

5. The at least one machine accessible storage medium of any one of Claims 1-4, comprising further instructions that when executed by the processor cause the processor to:

query the physically unclonable function on the hardware device for each of the two or more device keys.

6. The at least one machine accessible storage medium of any one of Claims 1-4, wherein the generating the device identifier includes applying a cryptographic hash algorithm to the two or more keys.

7. An apparatus for certifying a hardware device, the apparatus comprising:

a processor; and

an enrollment module executing on the processor, the enrollment module configured to:

receive two or more device keys from a physically unclonable function (PUF) on the hardware device;

generate a device identifier from the two or more device keys;

generate a digital signature based on the device identifier and a private key;

create a device certificate based on the device identifier and the digital signature; and

store the device certificate in a memory element.

8. The apparatus of Claim 7, wherein the enrollment module is further configured to:

apply a cryptographic hash algorithm to the two or more keys to generate the device identifier.

9. A method for certifying a hardware device, the method comprising:

receiving two or more device keys from a physically unclonable function (PUF) on the hardware device;

generating a device identifier from the two or more device keys;

generating a digital signature based on the device identifier and a private key;

creating a device certificate based on the device identifier and the digital signature; and

storing the device certificate in a memory element.

10. The method of Claim 9, wherein the physically unclonable function is configured on the hardware device to generate the two or more device keys, and wherein the keys uniquely identify the hardware device.

11. An apparatus for certifying a hardware device, the apparatus comprising:

means for receiving two or more device keys from a physically unclonable function (PUF) on the hardware device;

means for generating a device identifier from the two or more device keys;

means for generating a digital signature based on the device identifier and a private key;

means for creating a device certificate based on the device identifier and the digital signature; and

means for storing the device certificate in a memory element.

12. The apparatus of Claim 11, further comprising:

means for querying the physically unclonable function on the hardware device for each of the two or more device keys.

13. At least one machine accessible storage medium having instructions stored thereon for authenticating a hardware device, the instructions when executed by a processor cause the processor to:

receive two or more device keys from a physically unclonable function (PUF) on the hardware device;

generate a device identifier from the two or more device keys;

obtain a device certificate from the hardware device;

perform a verification of the device identifier; and

provide a result of the device identifier verification.

14. The at least one machine accessible storage medium of Claim 13, wherein the hardware device is rejected if the result indicates the device identifier verification failed.

15. The at least one machine accessible storage medium of Claim 13, comprising further instructions that when executed by the processor cause the processor to:

apply a cryptographic hash algorithm to the two or more keys to generate the device identifier.

16. The at least one machine accessible storage medium of Claim 13, comprising further instructions that when executed by the processor cause the processor to:

compare the device identifier to a previously generated device identifier to perform the device identifier verification.

17. The at least one machine accessible storage medium of Claim 16, wherein, if the device identifier and the previously generated device identifier do not match, the result indicates that the device identifier verification failed.

18. The at least one machine accessible storage medium of any one of Claims 13-17, comprising further instructions that when executed by the processor cause the processor to:

perform a verification of a digital signature in the device certificate; and

provide a result of the digital signature verification.

19. The at least one machine accessible storage medium of Claim 18, wherein the hardware device is rejected if the result indicates the digital signature verification failed.

20. The at least one machine accessible storage medium of Claim 18, comprising further instructions that when executed by the processor cause the processor to:

use a public key of a key pair to decrypt the digital signature to perform the digital signature verification.

21. The at least one machine accessible storage medium of any one of Claims 13-17, comprising further instructions that when executed by the processor cause the processor to:

query the physically unclonable function on the hardware device for each of the two or more device keys.

22. An apparatus for authenticating a hardware device, the apparatus comprising:

a processor; and

an evaluation module executing on the processor, the enrollment module configured to:

receive two or more device keys from a physically unclonable function (PUF) on the hardware device;

generate a device identifier from the two or more device keys;

obtain a device certificate from the hardware device;

perform a verification of the device identifier; and

provide a result of the device identifier verification.

23. A method for authenticating a hardware device, the method comprising:

receiving two or more device keys from a physically unclonable function (PUF) on the hardware device;

generating a device identifier from the two or more device keys;

obtaining a device certificate from the hardware device;

performing a verification of the device identifier; and

providing a result of the device identifier verification.

24. An apparatus for authenticating a hardware device, the apparatus comprising:

means for receiving two or more device keys from a physically unclonable function (PUF) on the hardware device;

means for generating a device identifier from the two or more device keys;

means for obtaining a device certificate from the hardware device;

means for performing a verification of the device identifier; and

means for providing a result of the device identifier verification.

25. The apparatus of Claim 24, further comprising:

means for comparing the device identifier to a previously generated device identifier to perform the device identifier verification.

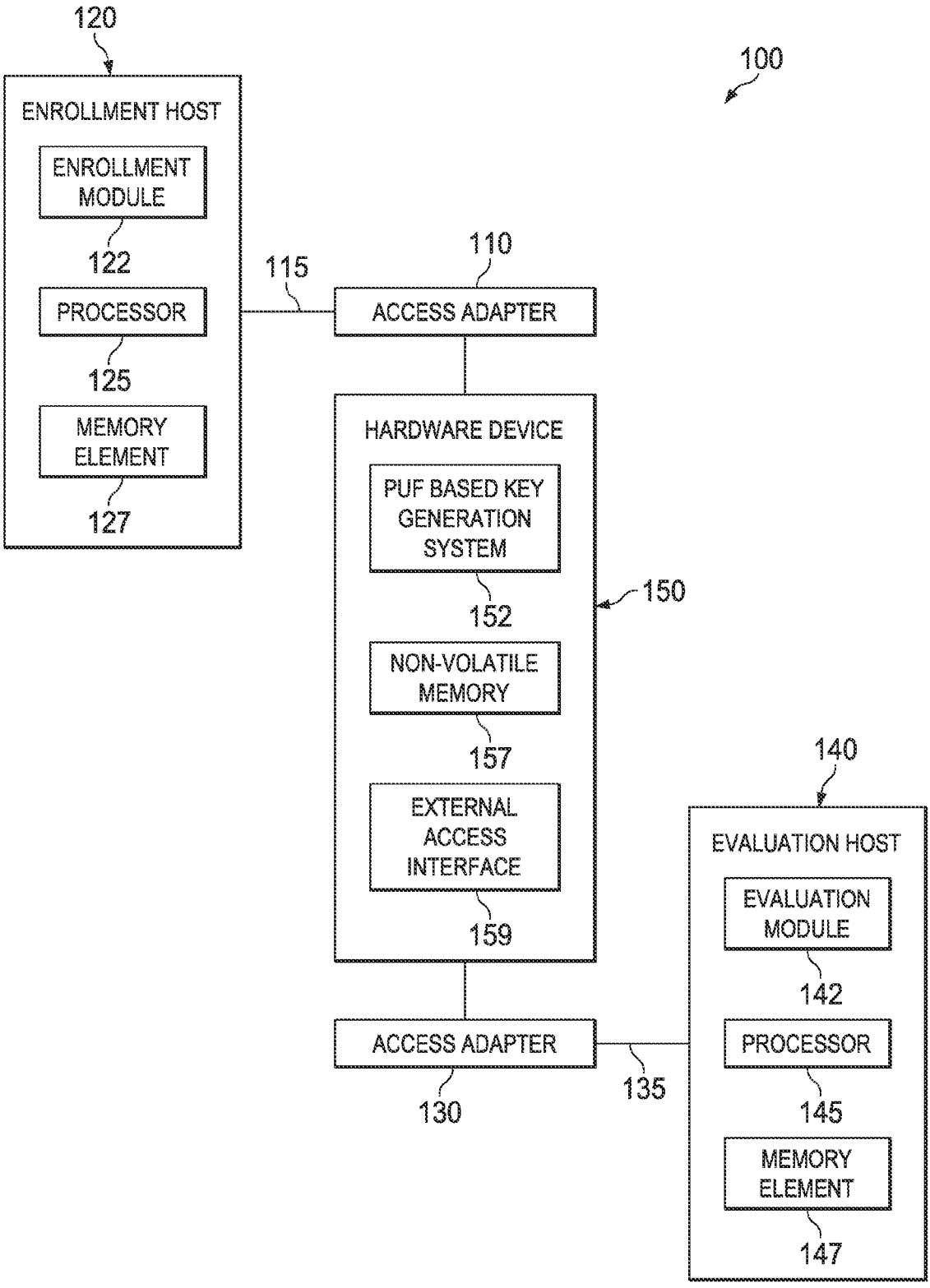

120
100
ENROLLMENT HOST
ENROLLMENT MODULE
122
PROCESSOR
125
MEMORY ELEMENT
127
115
110
ACCESS ADAPTER
HARDWARE DEVICE
PUF BASED KEY GENERATION SYSTEM
152
150
NON-VOLATILE MEMORY
157
EXTERNAL ACCESS INTERFACE
159
140
EVALUATION HOST
EVALUATION MODULE
142
ACCESS ADAPTER
PROCESSOR
135
130
145
MEMORY ELEMENT
147

FIG. 1

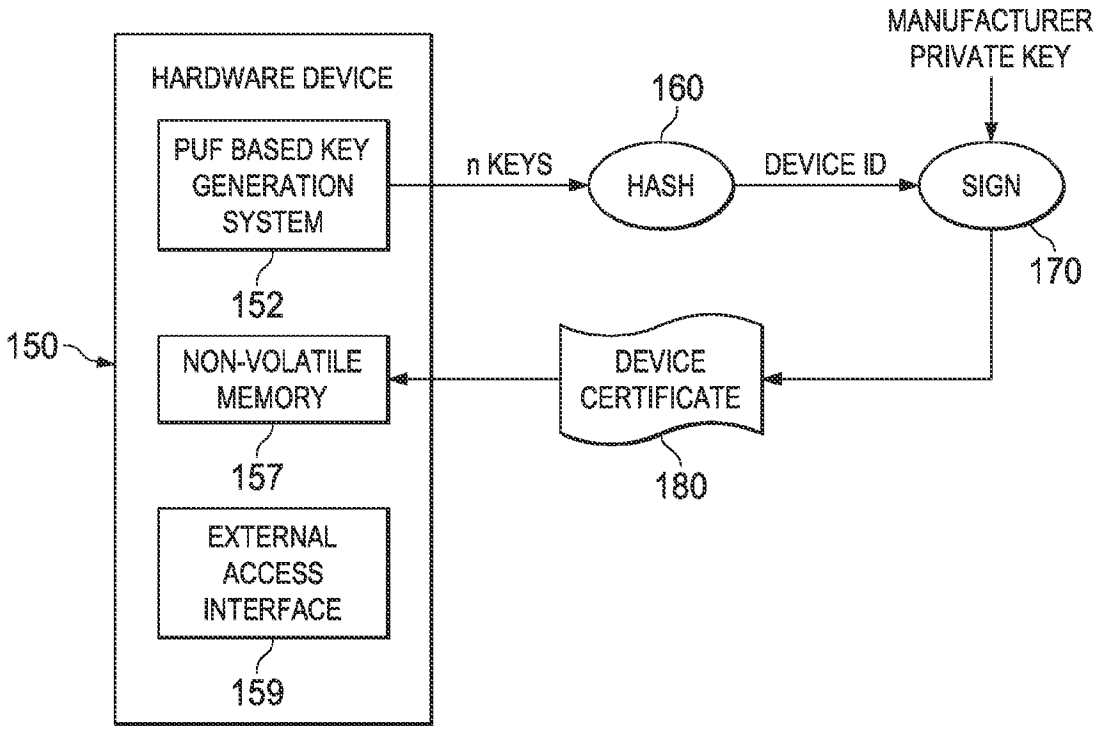

HARDWARE DEVICE
PUF BASED KEY GENERATION SYSTEM
152
NON-VOLATILE MEMORY
157
EXTERNAL ACCESS INTERFACE
159
150
n KEYS
160
HASH
DEVICE ID
MANUFACTURER PRIVATE KEY
SIGN
170
DEVICE CERTIFICATE
180

FIG. 2

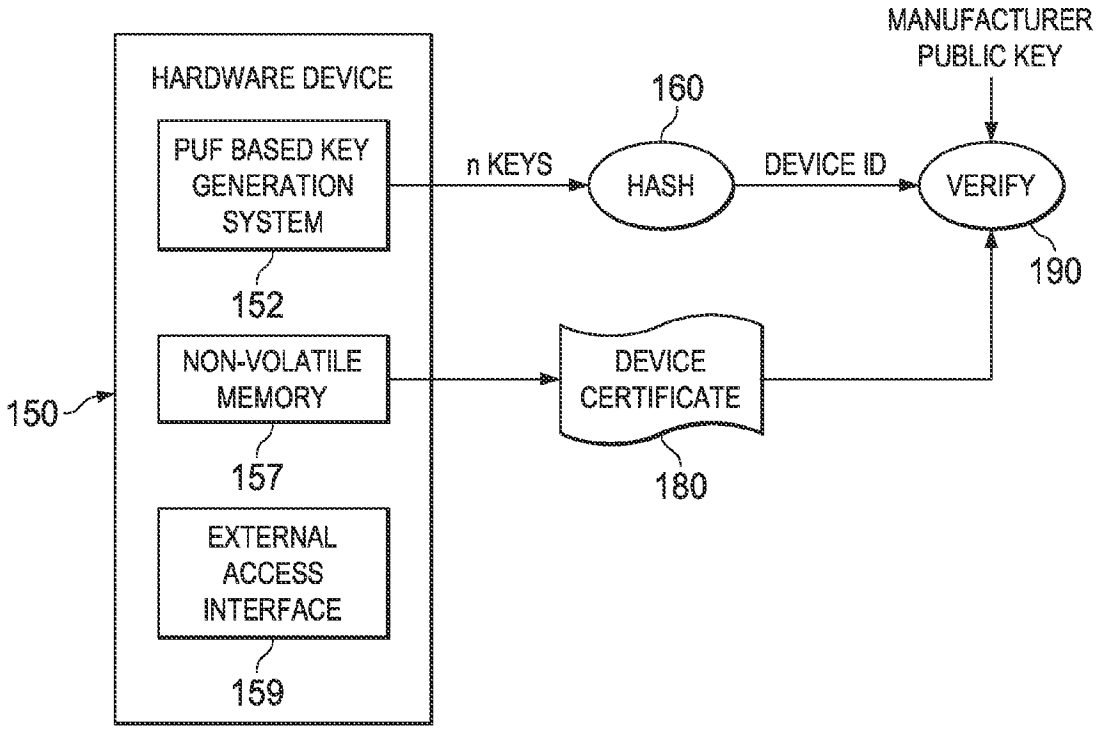

HARDWARE DEVICE
PUF BASED KEY GENERATION SYSTEM
152
NON-VOLATILE MEMORY
157
EXTERNAL ACCESS INTERFACE
159
150
n KEYS
160
HASH
DEVICE ID
MANUFACTURER PUBLIC KEY
VERIFY
190
DEVICE CERTIFICATE
180

FIG. 3

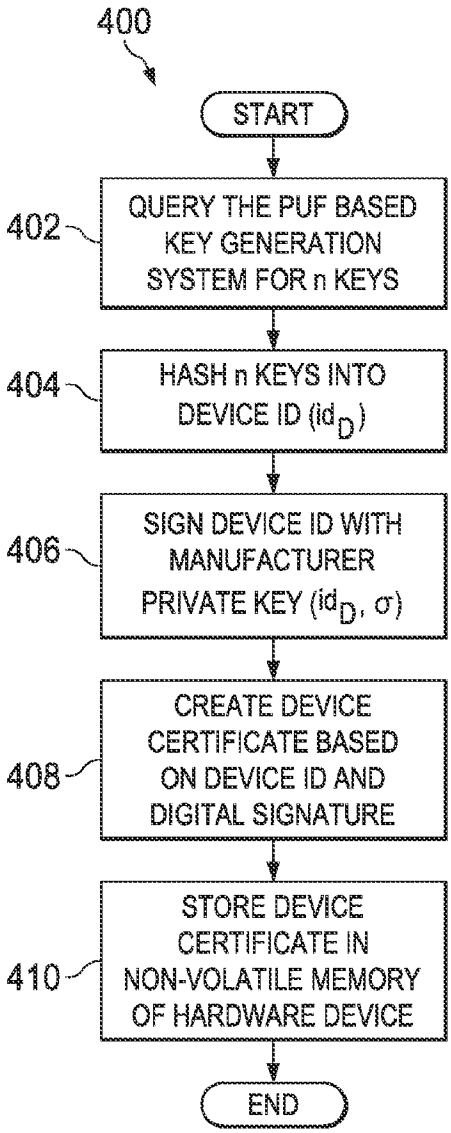

400
START
402
QUERY THE PUF BASED KEY GENERATION SYSTEM FOR n KEYS
404
HASH n KEYS INTO DEVICE ID (id_D)
406
SIGN DEVICE ID WITH MANUFACTURER PRIVATE KEY (id_D, σ)
408
CREATE DEVICE CERTIFICATE BASED ON DEVICE ID AND DIGITAL SIGNATURE
410
STORE DEVICE CERTIFICATE IN NON-VOLATILE MEMORY OF HARDWARE DEVICE
END

FIG. 4

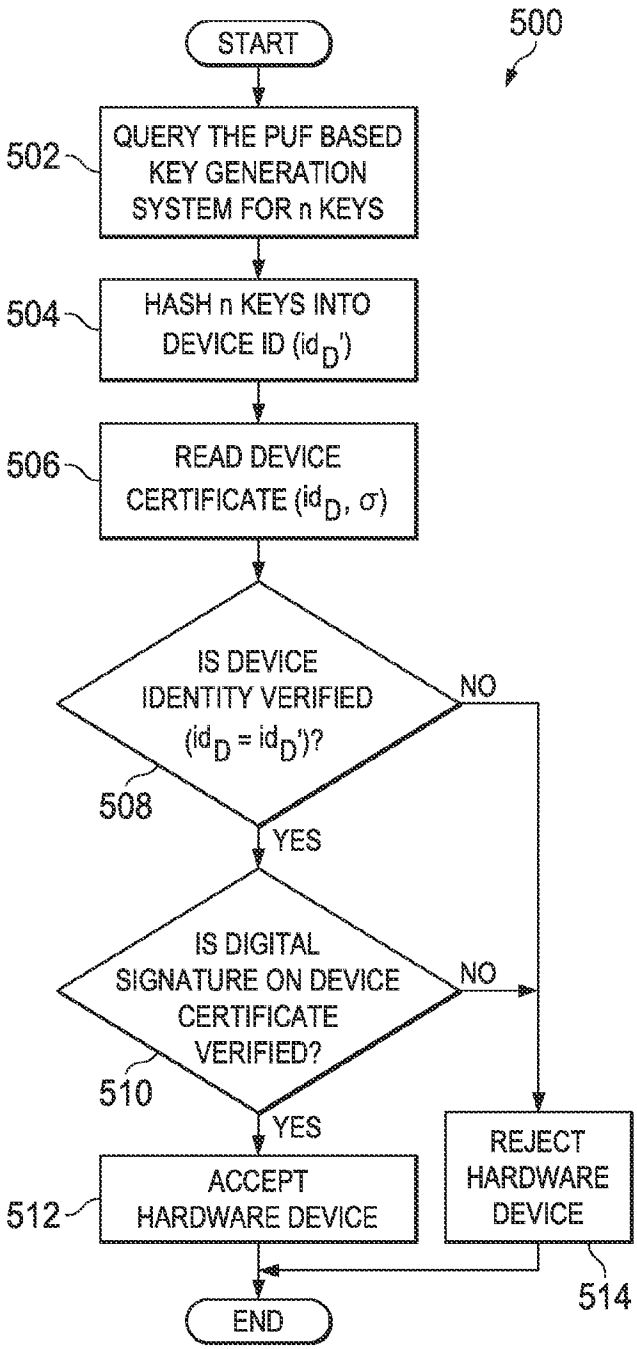

500
START
502
QUERY THE PUF BASED KEY GENERATION SYSTEM FOR n KEYS
504
HASH n KEYS INTO DEVICE ID (id_D')
506
READ DEVICE CERTIFICATE (id_D, σ)
IS DEVICE IDENTITY VERIFIED ($id_D = id_D'$)?
508
NO
YES
IS DIGITAL SIGNATURE ON DEVICE CERTIFICATE VERIFIED?
510
NO
YES
512
ACCEPT HARDWARE DEVICE
REJECT HARDWARE DEVICE
514
END

FIG. 5

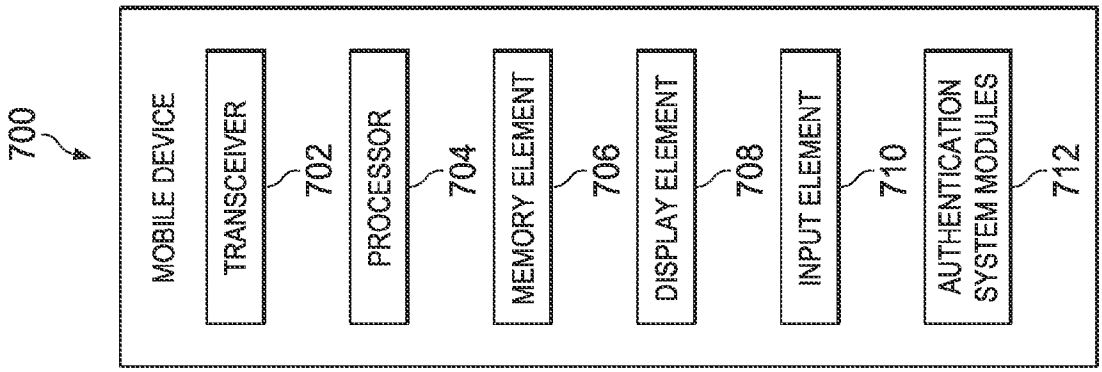

700
MOBILE DEVICE
TRANSCEIVER
702
PROCESSOR
704
MEMORY ELEMENT
706
DISPLAY ELEMENT
708
INPUT ELEMENT
710
AUTHENTICATION SYSTEM MODULES
712

FIG. 7

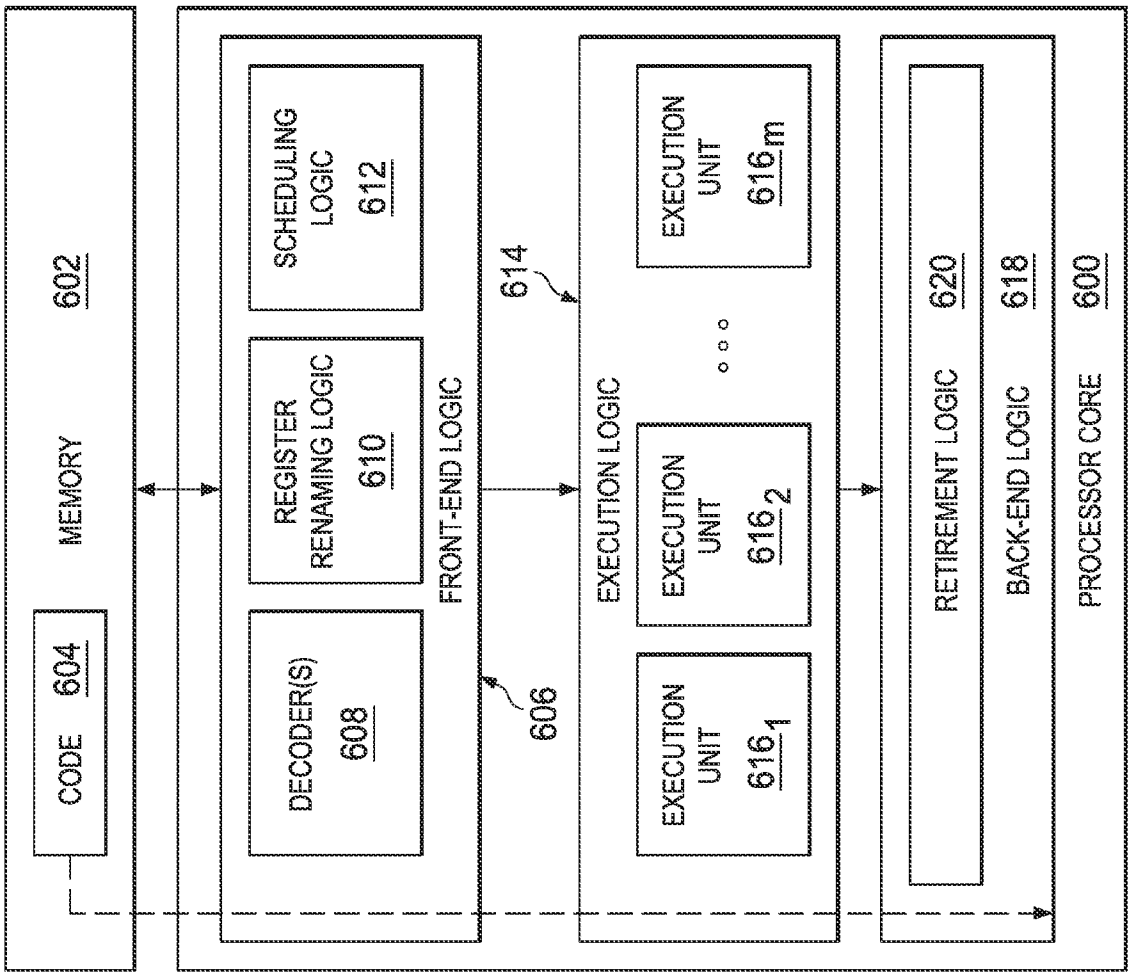
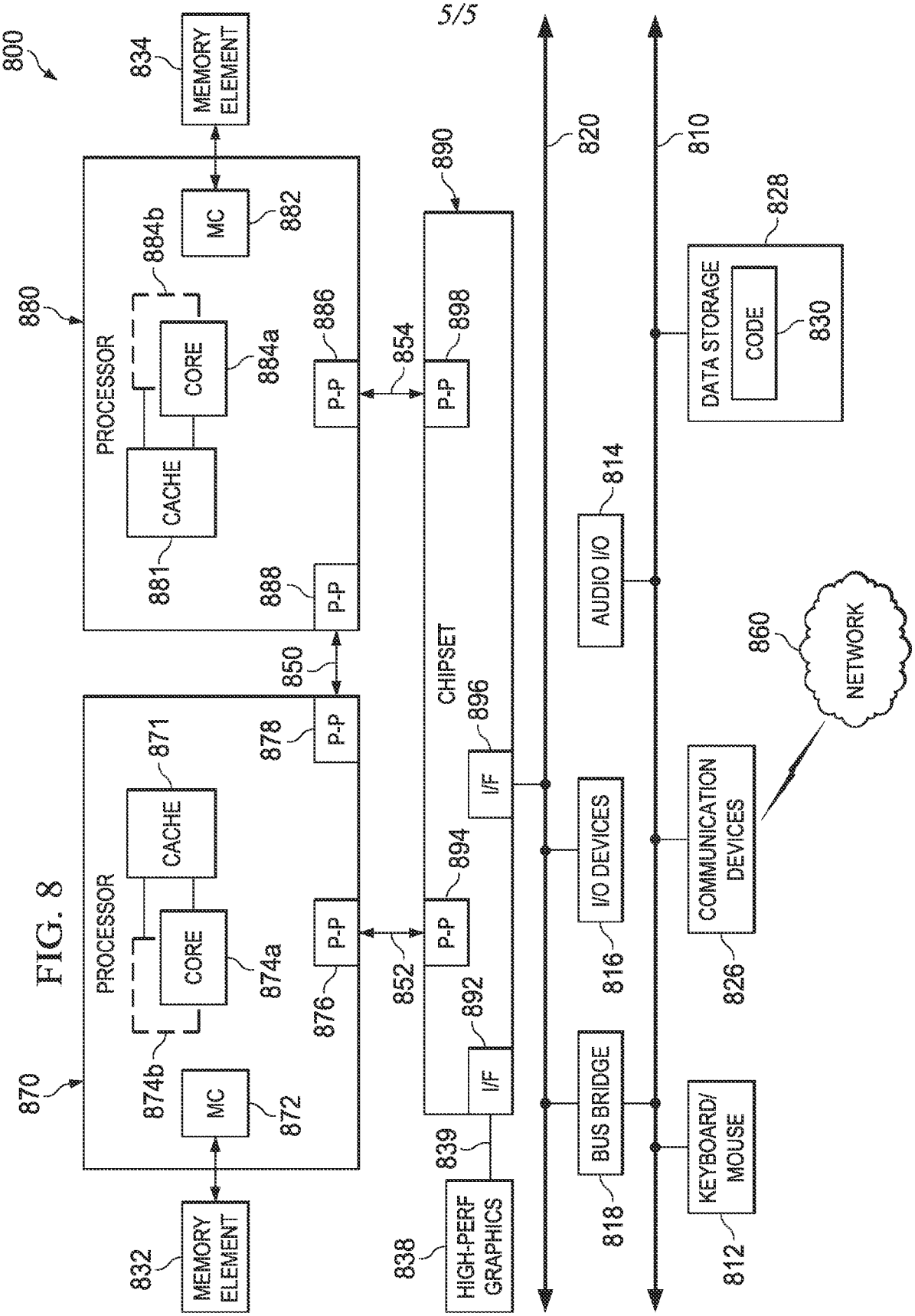

CODE 604
MEMORY 602
DECODER(S) 608
REGISTER RENAMING LOGIC 610
SCHEDULING LOGIC 612
FRONT-END LOGIC
606
614
EXECUTION UNIT 616_1
EXECUTION UNIT 616_2
EXECUTION UNIT 616_m
EXECUTION LOGIC
RETIREMENT LOGIC 620
BACK-END LOGIC 618
PROCESSOR CORE 600

FIG. 6

FIG. 8


800
834
MEMORY ELEMENT
880
PROCESSOR
MC
882
884b
CORE
884a
CACHE
881
888
P-P
886
854
890
P-P
898
CHIPSET
820
810
828
DATA STORAGE
CODE
830
850
878
P-P
871
CACHE
PROCESSOR
CORE
874a
874b
MC
872
870
876
P-P
852
P-P
894
896
I/F
892
I/F
839
838
HIGH-PERF GRAPHICS
832
MEMORY ELEMENT
AUDIO I/O
814
I/O DEVICES
816
BUS BRIDGE
818
KEYBOARD/ MOUSE
812
COMMUNICATION DEVICES
826
860
NETWORK

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2013/071346

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/30(2013.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F 21/30; H04L 9/08; H04K 1/00; H05K 13/00; G06F 12/14; H04L 9/30; H03K 19/003; G06F 9/455; G06F 21/22

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: PUF, hardware, key, identifier, signature, verification and similar terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2012-0321077 A1 (SHIGEMASA SHIOTA et al.) 20 December 2012 See paragraphs 59-118; figures 1, 6; and claims 1,11.	1-25
Y	US 2010-0199103 A1 (SANDER M. VAN RIJNSWON) 05 August 2010 See paragraphs 18, 37-45; and figure 1.	1-25
A	US 2010-0322418 A1 (MIODRAG POTKONJAK) 23 December 2010 See paragraphs 48-61; and figures 5-7.	1-25
A	US 2011-0215829 A1 (JORGE GUAJARDO MERCHAN et al.) 08 September 2011 See paragraphs 77-78.	1-25
A	US 2011-0239002 A1 (NATHAN ZACHARY BECKMANN et al.) 29 September 2011 See paragraphs 97-99; and figure 8.	1-25

☐ 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
27 February 2014 (27.02.2014)	**27 February 2014 (27.02.2014)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701,
Republic of Korea
Facsimile No. +82-42-472-7140

Authorized officer
BYUN, Sung Cheal
Telephone No. +82-42-481-8262

Form PCT/ISA/210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/US2013/071346

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012-0321077 A1	20/12/2012	CN 102843231 A	26/12/2012
		EP 2544400 A2	09/01/2013
		EP 2544400 A3	17/07/2013
		JP 2013-031151 A	07/02/2013
		KR 10-2012-0140224 A	28/12/2012
US 2010-0199103 A1	05/08/2010	CN 101681657 A	24/03/2010
		EP 2156437 A1	24/02/2010
		WO 2008-152547 A1	18/12/2008
US 2010-0322418 A1	23/12/2010	CN 102461063 A	16/05/2012
		JP 2012-529867 A	22/11/2012
		KR 10-1336004 B1	03/12/2013
		KR 10-2012-0069655 A	28/06/2012
		US 8379856 B2	19/02/2013
		WO 2011-016900 A2	10/02/2011
		WO 2011-016900 A3	31/03/2011
US 2011-0215829 A1	08/09/2011	EP 2191410 A2	02/06/2010
		TW 200917085 A	16/04/2009
		WO 2009-024913 A2	26/02/2009
		WO 2009-024913 A3	19/11/2009
		WO 2009-024913 A4	18/03/2010
		WO 2009-024913 A9	03/06/2010
US 2011-0239002 A1	29/09/2011	US 2013-246809 A1	19/09/2013
		US 8458489 B2	04/06/2013