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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 August 2017 (10.08.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/135942 A1

(51) **International Patent Classification**:
G06F 21/57 (2013.01) *G06F 21/56* (2013.01)

(21) **International Application Number**:
PCT/US2016/016369

(22) **International Filing Date**:
3 February 2016 (03.02.2016)

(25) **Filing Language**: English

(26) **Publication Language**: English

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

(72) **Inventors**: **GOMES DE OLIVERIA, Marcelo**; Av. Ipiranga, 6.681, Predio 95 -5, 6 e 7 andares, 90619-900 Porto Alegre (BR). **CESA ROSA, Diogo**; Av. Ipiranga, 6.681, Predio 95 -5, 6 e 7 andares, 90619-900 Porto Alegre (BR). **FERRETTO, Fernando**; Av. Ipiranga, 6.681, Predio 95 -5, 6 e 7 andares, 90619-900 Porto Alegre (BR). **STAUB, Charles R.**; Av. Ipiranga, 6.681, Predio 95 -5, 6 e 7 andares, 90619-900 Porto Alegre (BR). **WANG, Lan**; 11445 Compaq Center Drive West, Houston, Texas 77070 (US). **ALI, Vali**; 11445 Compaq Center Drive West, Houston, Texas 77070 (US). **WALDRON, James R.**; 11445 Compaq Center Drive West, Houston, Texas 77070 (US). **DE QUADROS WEBER, Tiago V**; Av. Ipiranga, 6.681, Predio 95 -5, 6 e 7 andares, 90619-900 Porto Alegre (BR). **SA CAMPAO, Juilo Cesar**; Av. Ipiranga, 6.681, Predio 95 -5, 6 e 7 andares, 90619-900 Porto Alegre (BR).

(74) **Agent**: **LEMMON, Marcus**; HP Inc, 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, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU,

[Continued on next page]

(54) **Title**: HEARTBEAT SIGNAL VERIFICATION

(57) **Abstract**: Examples associated with heartbeat signal verification are disclosed. One example includes verifying a provisioning key using a trusted key. The provisioning key is received from a remote device via an intermediary process. An intermediate key provided by the intermediary process is verified using the provisioning key. A session identifier encrypted using the intermediate key and provided to the intermediary process. A first heartbeat signal is received from intermediary process. The heartbeat signal is generated based on the session identifier.

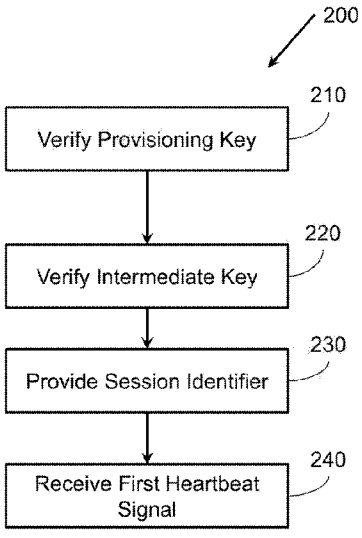


Figure 2

WO 2017/135942 A1

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))*

Published:

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

HEARTBEAT SIGNAL VERIFICATION

BACKGROUND

[0001] Computer security is an important consideration whenever a person or company operates a computer today. Security solutions often include intrusion prevention techniques, threat detection techniques, and so forth, and strong security solutions employ a wide variety of techniques. If a computer is not protected from malicious threats, data on the computer could be inappropriately accessed, the computer could be used by an attacker, the computer could be irreparably damaged, and so forth.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] The present application may be more fully appreciated in connection with the following detailed description taken in conjunction with the accompanying drawings.

[0003] FIG. 1 illustrates an example system associated with heartbeat signal verification.

[0004] FIG. 2 illustrates a flowchart of example operations associated with heartbeat signal verification.

[0005] FIG. 3 illustrates another flowchart of example operations associated with heartbeat signal verification.

[0006] FIG. 4 illustrates an example system associated with heartbeat signal verification.

[0007] FIG. 5 illustrates another flowchart of example operations associated with heartbeat signal verification.

[0008] FIG. 6 illustrates an example computing device in which example systems, and methods, and equivalents, may operate.

DETAILED DESCRIPTION

[0009] Systems, methods, and equivalents associated with heartbeat signal verification are described. A system (e.g., personal computer, mobile device, server) may have several different processes that perform functions associated with ensuring that the system remains secure. These systems may protect data, prevent malware from being installed, and so forth. In some examples, a system may have a first process for security that operates at the firmware (e.g., basic input/output system - BIOS) level of the system. The system may also have a process that operates at an operating system level. Though the firmware level process may be more difficult to remove or subvert because it can be tied to system firmware, the operating system level process may have stronger capabilities than the firmware level process. This may be because, for example, the operating system level process has access to more system resources than the firmware level process, the operating system level process can be more easily configured to protect other applications operating at the operating system level, and so forth.

[0010] Consequently, it may be desirable for the firmware level process to ensure continued operation of the operating system level process. If the operating system level process ceases functioning due to, for example, a malware, a user action, and so forth, the firmware level process can take a remedial action to secure the system. These actions may include, for example, disabling functionality of the system, locking down the system, reinstalling and/or reactivating the operating system level process, and so forth.

[0011] One manner by which the firmware level process can ensure continued operation of the operating system level process is via a heartbeat signal sent periodically from the operating system level process to the firmware level process. However, if an attacker can mimic the heartbeat signal, the attacker may be able to convince the firmware level process that the operating system level process is still functioning when it has actually been disabled. Thus, to establish security of the heartbeat signal, the firmware level process and the operating system level process

may establish a chain of trust for secure communications using a series of public-private keys. First, the operating system level process may obtain a provisioning key from an external device (e.g., remote server). The provisioning key may be digitally signed by a trusted authority using a private key half of a trusted key which is known to be secure by both the firmware level process and the operating system level process. The trusted key may be a public key half of a pair of keys and may be stored in a secure storage accessible to the firmware level process. The operating system level process may then create an intermediate key and have that signed by the external device using the private-key half of the provisioning key. The public key half of the intermediate key may be used by the firmware level process to securely send the operating system level process a session identifier. This session identifier may be used as the basis for a series of heartbeat signals transmitted from the operating system level process to the firmware level process.

[0012] In various examples, the firmware level process may respond to heartbeat signals with signals to the operating system level process containing a series of nonces. Each nonce may be used by the operating system level process, in combination with the session identifier, to generate the next heartbeat signal to be provided by the operating system level process to the firmware level process.

[0013] Figure 1 illustrates an example system associated with heartbeat signal verification. It should be appreciated that the items depicted in figure 1 are illustrative examples and many different systems, and so forth, may operate in accordance with various examples.

[0014] Figure 1 illustrates a system 100 associated with heartbeat signal verification. Figure 1 includes a firmware security module 110. System 100 also includes an operating system security module 120. These security modules may be tasked with a variety of functions associated with preventing processes and/or data associated with system 100 from becoming compromised. By way of illustration, operating system security module 120 may hinder installation of malware, prevent unauthorized access to data associated with system 100, prevent interference with other operating system level applications, and so forth. Firmware level security module 110 may be able to, for example, prevent improper use of lower level system

resources, restore a state of system 100 in the event an intrusion is detected, and so forth. Security modules 110 and 120 may also perform a wide variety of other functions depending, for example, on the technical design of system 100 (e.g., system components), the purpose system 100 is designed to achieve (e.g., personal computer, server), other processes and/or applications operating on system 100, and so forth.

[0015] In some examples, firmware security module 110 may be tasked with hindering and/or deterring removal of operating system security module 120. By way of illustration, if firmware security module 110 detects that operating system security module 120 has been removed, firmware security module may take some action to protect system 100, system 100's data, and so forth. These actions may include, for example, reinstalling operating system security module 120, turning off system 100, locking access to data on system 100, and so forth.

[0016] Hindering removal of operating system security module 120 may be achieved using a variety of techniques. One technique may include a periodic exchange of signals between firmware security module 110 and operating system security module 120. In examples described herein, heartbeat signals may be sent from operating system security module 120 to firmware security module 110. Firmware security module 110 may acknowledge the heartbeat signals with information (e.g., a nonce) that may be used to generate a future heartbeat signal.

[0017] However, heartbeat signals on their own may not sufficiently hinder removal of operating system security module if the heartbeat signals can be spoofed or simulated by an attacker. Consequently, firmware security module 110 and operating system security module may seek to encrypt transmissions between one another to increase the complexity of spoofing the heartbeat signals and/or responses to the heartbeat signals.

[0018] To that end, before transmission of heartbeat signals begins, firmware security module 110 and operating system security module 120 may go through a process to securely encrypt their communications. First operating system security module 120 may obtain a provisioning key from a remote device 190. The provisioning key may be a public key half of a public-private key. In some examples, remote device

190 itself may create the provisioning key and the private key half of the provisioning key. Remote device 190 may retain the private key half of the provisioning key, and have a trusted authority 199 sign the provisioning key before providing the provisioning key to operating system security module 120.

[0019] Trusted authority 199 may be some trusted entity (e.g., manufacturer, operating system distributor, security company) that can guarantee an acceptable degree of protection regarding the private key half of the trusted key to prevent an attack involving signing a provisioning key with malicious intent. Trusted authority 199 may sign the provisioning key with the private key half of another public-private key pair. The public key half to this private key may be a trusted key that is stored in a key store 130. Key store 130 may be a secure storage accessible to firmware security module 110. Other resources of system 100 (e.g., operating system security module 120, other processes) may also be able to access key store 130 or portions of key store 130. In various examples, the trusted key may be stored in key store 130 prior to an initial operation of system 100. For example, when system 100 is put together in a factory, the trusted key may be embedded within key store 130. In another example, the trusted key may be stored in key store 130 during operating system installation for system 100.

[0020] Once operating system security module 120 receives the provisioning key from remote device 190, operating system security module 120 may pass the provisioning key to firmware security module 110. At this point firmware security module 110 may verify the provisioning key using the trusted key from key store 130 to check that trusted authority 199 properly signed the provisioning key. In some examples, firmware security module 120 may also compare a timestamp associated with the provisioning key to a timestamp associated with a prior provisioning key. If the timestamp associated with the provisioning key is later than the timestamp associated with the prior provisioning key, then firmware security module 110 may store the provisioning key (e.g., in key store 130). Comparing the timestamps may prevent an attacker from submitting an old provisioning key to, for example, disrupt communications, trick firmware security module 110 into using a compromised provisioning key, and so forth.

[0021] Once firmware security module 110 has verified the provisioning key, operating system security module 120 may then generate an intermediate key. The intermediate key may be a public key half of a public-private key pair. Operating system security module 120 may have remote device 190 sign the intermediate key using the private key half of the provisioning key. Similarly to trusted authority 199, remote device 190 may be operated by some known trusted organization (e.g., manufacturer, security company). By way of illustration, remote device 190 may be a server tasked with generating public-private key pairs for the purpose of securing communications between entities. Though, the trusted key could be used for verifying the intermediate key, the more public-private key pairs are used, the more likely they are to become compromised. Consequently, because the trusted key may have been embedded in key store 130 in a manner that is difficult to modify, it may be desirable to use the trusted key and the private key half of the trusted key on a limited basis.

[0022] Once operating system security module 120 has had remote device 190 sign the intermediate key with the private key half of the provisioning key, operating system security module 120 may provide the intermediate key to firmware security module 110. At this point, firmware security module 110 may verify that remote device 190 has signed the intermediate key using the provisioning key. As with the provisioning key, firmware security module 110 may also compare timestamps of the intermediate key and a prior intermediate key to prevent verification of an out of date intermediate key.

[0023] Using the intermediate key, firmware security module 110 may encrypt a session identifier and transmit the session identifier to firmware security module 110. The session identifier may serve as a basis for communication of heartbeat signals between firmware security module 110 and operating system security module 120. Operating system security module 120 may then decrypt and store the session identifier.

[0024] Using the session identifier, operating system security module 120 may begin transmitting heartbeat signals to firmware security module 110. The heartbeat signals may be generated at least in part based on the session identifier. For example, in some examples, where SID refers to the session identifier a NONCE refers to a

nonce value provided to operating system module from firmware security module, the heartbeat signals HB may have the form generated according to equation 1:

1. $HB_n = SHA2\ (SID\ xor\ NONCE_{n-1})$

[0025] Upon receiving a heartbeat signal, because firmware security module 110 knows both the previous nonce and the session ID, firmware security module 110 may recalculate equation 1 and compare the result to the value received from operating system module 120. If the recalculated result heartbeat signals differ, firmware security module 110 may take some remedial action, wait for a new heartbeat signal, and so forth. Upon receiving the heartbeat signal, firmware security module 110 may then send a new nonce to operating system security module 120 for use in generating the next heartbeat signal.

[0026] In some examples, firmware security module 110 may not have provided a session identifier to operating system security module 120 for the first heartbeat signal. In these examples, the first heartbeat signal may be generated, for example, by taking the SHA2 of the session identifier, the session identifier XOR'ed with itself, and so forth. In other examples, firmware security module 110 may provide an initial nonce to operating system security module 120 with the session identifier, in a separate signal prior to the first heartbeat signal, and so forth.

[0027] In the event that firmware security module 110 fails to receive a heartbeat signal after a predetermined period of time, firmware security module may take some form of remedial action. These actions may include, for example, re-initiating operating system security module 120, alerting an operator of system 100, preventing use system 100, preventing access to certain resources (e.g., data) available to system 100, alerting an administrator, and so forth. These remedial actions may help to protect system 100 from attacks that remove operating system security module 120.

[0028] It is appreciated that, in the following description, numerous specific details are set forth to provide a thorough understanding of the examples. However, it is appreciated that the examples may be practiced without limitation to these specific

details. In other instances, methods and structures may not be described in detail to avoid unnecessarily obscuring the description of the examples. Also, the examples may be used in combination with each other.

[0029] "Module", as used herein, includes but is not limited to hardware, firmware, software stored on a computer-readable medium or in execution on a machine, and/or combinations of each to perform a function(s) or an action(s), and/or to cause a function or action from another module, method, and/or system. A module may include a software controlled microprocessor, a discrete module, an analog circuit, a digital circuit, a programmed module device, a memory device containing instructions, and so on. Modules may include gates, combinations of gates, or other circuit components. Where multiple logical modules are described, it may be possible to incorporate the multiple logical modules into one physical module. Similarly, where a single logical module is described, it may be possible to distribute that single logical module between multiple physical modules.

[0030] Figure 2 illustrates an example method 200 associated with heartbeat signal verification. Method 200 may be embodied on a non-transitory computer-readable medium storing processor-executable instructions. The instructions, when executed by a processor, may cause the processor to perform method 200. In other examples, method 200 may exist within logic gates and/or RAM of an application specific integrated circuit (ASIC).

[0031] Method 200 includes verifying a provisioning key at 210. The provisioning key may be provided by a remote device. The provisioning key may be provided by an intermediary process. The intermediary process and a process performing actions associated with method 200 may operate on the same device (e.g., computer). The verification at action 210 may be performed using a trusted key. The trusted key may be a public key half of a public-private key generated according to RSA. In this example, the private key half may be controlled by a trusted authority. The public key half may have been embedded in a secure storage of a computer on which method 200 is being performed prior to initial operation of the computer. For example, the public key half may be embedded into the secure storage during initial

setup of the computer, while or prior to the computer being physically constructed, and so forth.

[0032] The provisioning key may also be a public key half of a public-private key generated according to RSA. The provisioning key may operate on a temporary basis and/or be used more widely than the trusted key. In some examples, the remote device may generate the provisioning key. Subsequently, the provisioning key may be signed by a trusted authority using a trusted private key that is the private key half of the trusted key.

[0033] Consequently, verifying the provisioning key may include ensuring the provisioning key is properly signed by the trusted authority. The trusted authority may sign the provisioning key using the trusted private key, which the process may detect using the trusted key. Verifying the provisioning key may also include ensuring a timestamp associated with the provisioning key exceeds a timestamp of a previous provisioning key. This may prevent an attacker from convincing the process to use an old provisioning key that could have been compromised or otherwise disrupt communications between the process and the intermediary process.

[0034] Method 200 also includes verifying an intermediate key at 220. The intermediate key may be provided by the intermediary process. The intermediate key may be verified using the provisioning key. The intermediate key may also be a public key half of a public-private key generated according to RSA. The intermediate key may be generated by the intermediary process. The intermediate key may be signed by the remote device using a provisioning private key that is paired with the provisioning key.

[0035] Similar to the trusted key, verifying the intermediate key may include ensuring the intermediate key is properly signed by the remote device. The remote device may sign the intermediate key using the provisioning private key, which may be verified using the provisioning key. Verifying the intermediate key may also include ensuring a timestamp associated with the intermediate key exceeds a timestamp associated with a previous intermediate key.

[0036] Method 200 also includes providing a session identifier at 230. The session identifier may be provided to the intermediary process. The session identifier may be provided in an encrypted form using the intermediary key. This may allow the intermediary process to decrypt the session identifier using a private key half of the intermediary key.

[0037] Method 200 also includes receiving a first heartbeat signal at 240. The first heartbeat signal may be received from the intermediary process. The first heartbeat signal may be generated based on the session identifier. Additionally, the first heartbeat signal may be generated by encrypting the session identifier using SHA-2.

[0038] Figure 3 illustrates a method 300 associated with heartbeat signal verification. Figure 3 includes several actions similar to those described above with reference to method 200. For example, method 300 includes verifying a provisioning key at 310, verifying an intermediate key at 320, providing a session identifier at 330, and receiving a first heartbeat signal 340.

[0039] Method 300 also includes storing the provisioning key at 315. Storing the provisioning key in, for example, a secure storage, may facilitate verifying the intermediate key at action 320. Method 300 also includes storing the intermediate key at 325. This may facilitate secure provision of the session identifier to the intermediary process at action 330.

[0040] Method 300 also includes providing a nonce at 350. The nonce may be provided to the intermediary process in response to the first heartbeat signal. Method 300 also includes receiving a second heartbeat signal from the intermediary process at 360. The second heartbeat signal may be generated based on the session identifier and the nonce. By way of illustration, the second heartbeat signal may be generated by applying an XOR operator to the session identifier and the nonce, and then encrypting the result using SHA-2.

[0041] Subsequently, actions 350 and 360 may be repeated as long as a device performing method 300 is operating. When the intermediary process receives

a heartbeat signal, a new nonce may be provided, which may be used to generate the next heartbeat signal.

[0042] Figure 4 illustrates a system 400 associated with heartbeat signal verification. System 400 includes a first security module 410. First security module 410 may verify a series of public-private keys received from a second security module 420. Verifying the series of public-private keys may establish a chain of trust for a session identifier. The session identifier may be used during communications between first security module 410 and second security module 420. First security module 410 may operate within the firmware of system 400.

[0043] System 400 also includes second security module 420 that operates at an operating system level of system 400. Second security module 420 may provide period heartbeat signals to first security module 420. The periodic heartbeat signals may be generated as a function of the session identifier. Additionally, the periodic heartbeat signals may be generated based on nonces received from first security module 410. The nonces may be received in response to previous heartbeat signals, and nonces may be used to generate a single periodic heartbeat signal. The periodic heartbeat signals may be provided from first security module 420 to second security module 410 in an encrypted form.

[0044] In one example, the heartbeat signals may be generated by applying a SHA-2 function to an XOR of the session identifier and a previous nonce. The first security module, which knows bot the session identifier and the nonce, may verify the validity of the heartbeat signal, by recalculating the heartbeat signal and comparing the result to the heartbeat signal actually received. In some examples, second security module 420 may provide first security module 410 an initial heartbeat signal generated without a nonce provided by first security module 410. First security module 410 may provide an initial nonce in response to the initial heartbeat signal, and subsequent heartbeat signals may be generated based on the nonce. In an alternative example, first security module 410 may provide second security module 420 an initial nonce in the signal containing the session identifier, allowing the initial heartbeat signal to be generated based on both the initial nonce and the session identifier.

[0045] When first security module 410 fails to receive a valid heartbeat signal after a predetermined period of time, first security module may perform a remedial action. This may hinder undetected removal of second security module 420. Further due to the manner of encrypting heartbeat signals, spoofing the heartbeat signals may also be difficult for an attacker attempting to remove second security module 420, further hindering undetected removal of second security module 420. In various examples, the remedial action may include, forcing shutdown of system 400, reinstalling second security module 420, alerting an external device, and so forth.

[0046] Figure 5 illustrates a method 500 associated with heartbeat signal verification. Method 500 includes storing a provisioning key at 510. The provisioning key may be received from a remote device via an intermediary process. The provisioning key may be stored when the provisioning key has been signed by a trusted authority and when the provisioning key has a first timestamp that exceeds a timestamp associated with a prior provisioning key.

[0047] Method 500 also includes storing an intermediate key at 520. The intermediate key may be received from the intermediary process. The intermediate key may be stored when the intermediate key has been signed by the remote device and has a second timestamp that exceeds a timestamp associated with a prior intermediate key.

[0048] Method 500 also includes providing a session identifier to the intermediary process using the intermediate key. Method 500 also includes receiving a first heartbeat signal from the intermediary process. The heartbeat signal may be generated based on the session identifier. Method 500 also includes providing a nonce to the intermediary process in response the first heartbeat signal.

[0049] Method 500 also includes providing nonces to the intermediary process. Each nonce may be generated in response to a subsequent heartbeat signal received from the intermediary process. Consequently, each subsequent heartbeat signal may be generated based on the session identifier and on a nonce provided in response to a prior heartbeat signal.

[0050] In some examples, method 500 also includes performing a remedial action (not shown). In these actions, the remedial action may be taken upon failing to receive a heartbeat signal from the intermediary process after a predetermined period of time. As discussed above, a missed heartbeat signal may indicate a security breach and the remedial action may attempt to fix or limit effects of the security breach.

[0051] Figure 6 illustrates an example computing device in which example systems and methods, and equivalents, may operate. The example computing device may be a computer 600 that includes a processor 610 and a memory 620 connected by a bus 630. Computer 600 includes a heartbeat signal verification module 640. Heartbeat signal verification module 640 may perform, alone or in combination, various functions described above with reference to the example systems, methods, apparatuses, and so forth. In different examples, heartbeat signal verification module 640 may be implemented as a non-transitory computer-readable medium storing processor-executable instructions, in hardware, software, firmware, an application specific integrated circuit, and/or combinations thereof.

[0052] The instructions may also be presented to computer 600 as data 650 and/or process 660 that are temporarily stored in memory 620 and then executed by processor 610. The processor 610 may be a variety of processors including dual microprocessor and other multi-processor architectures. Memory 620 may include non-volatile memory (e.g., read only memory) and/or volatile memory (e.g., random access memory). Memory 620 may also be, for example, a magnetic disk drive, a solid state disk drive, a floppy disk drive, a tape drive, a flash memory card, an optical disk, and so on. Thus, memory 620 may store process 660 and/or data 650. Computer 600 may also be associated with other devices including other computers, computers, peripherals, and so forth in numerous configurations (not shown).

[0053] It is appreciated that the previous description of the disclosed examples is provided to enable any person skilled in the art to make or use the present disclosure. Various modifications to these examples will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other examples without departing from the spirit or scope of the disclosure. Thus, the

present disclosure is not intended to be limited to the examples shown herein but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.

WHAT IS CLAIMED IS:

1. A method, comprising:

verifying a provisioning key provided by a remote device via an intermediary process, where the verification is performed using a trusted key;

verifying an intermediate key provided by the intermediary process, where the intermediate key is verified using the provisioning key;

providing a session identifier to the intermediary process, the session identifier encrypted using the intermediate key; and

receiving, from the intermediary process, a first heartbeat signal generated based on the session identifier.

2. The method of claim 1, further comprising:

providing to the intermediary process, a nonce; and

receiving from the intermediary process, a second heartbeat signal generated based on the session identifier and the nonce.

3. The method of claim 2, where the second heartbeat signal is generated in part by applying an XOR operation to the session identifier and the nonce, and encrypting the result using SHA-2.

4. The method of claim 1, where the first heartbeat signal is generated in part by encrypting the session identifier using SHA-2.

5. The method of claim 1 where the trusted key, the provisioning key, and the intermediate key are public key halves of public-private keys generated according to RSA.

6. The method of claim 5, comprising:

storing the provisioning key; and

storing the intermediate key.

7. The method of claim 5, where the provisioning key is generated by the remote device and where the provisioning key is signed by a trusted authority using a trusted private key paired with the trusted key.

8. The method of claim 5, where the intermediate key is generated by the intermediary process, and where the intermediate key is signed by the remote device using a provisioning private key paired with the provisioning key.

9. The method of claim 1, where a condition of verifying the provisioning key includes ensuring a timestamp associated with the provisioning key exceeds a timestamp of a previous provisioning key and where a condition of verifying the intermediate key includes ensuring a timestamp associated with the intermediate key exceeds a timestamp of a previous intermediate key.

10. A system, comprising:

a first security module to verify a series of public-private keys received from a second security module to establish a chain of trust for a session identifier to be used when the first security module and the second security module communicate with one another; and

the second security module to provide periodic heartbeat signals to the first security module, where the periodic heartbeat signals are generated as a function of (i) a session identifier passed from the second security module to the first security module using the pair of public-private keys, (ii) nonces received from the first security module in response to previous heartbeat signals, and where the periodic heartbeat signals are provided in an encrypted form.

11. The system of claim 10, where the first security module operates at a firmware level and where the second security module operates at an operating system level.

12. The system of claim 11, where the first security module performs a remedial action after failing to receive a heartbeat signal after a predetermined period of time.

13. The system of claim 10, where the second security module provides the first security module an initial heartbeat signal generated without a nonce, where the first security module provides an initial nonce in response to the initial heartbeat signal, and where nonces are used to generate a single periodic heartbeat signal.

14. A non-transitory computer-readable medium storing computer-executable instructions that when executed by a computer cause the computer to:

store a provisioning key, received from a remote device via an intermediary process, when the provisioning key has been signed by a trusted authority and has a first timestamp that exceeds a timestamp associated with a prior provisioning key;

store an intermediate key, received from the intermediary process, when the intermediate key has been signed by the remote device and has a second timestamp that exceeds a timestamp associated with a prior intermediate key;

provide a session identifier to the intermediary process using the intermediate key;

receive a first heartbeat signal from the intermediary process, where the heartbeat signal is generated based on the session identifier;

provide a nonce in response to the first heartbeat signal; and

provide nonces in response to subsequent heartbeat signals from the intermediary process, where each subsequent heartbeat signal is generated based on the session identifier and on a nonce provided in response to a prior heartbeat signal.

15. The computer-readable media of claim 14, stores further instructions that cause the computer to: perform a remedial action upon failing to receive a heartbeat signal after a predetermined period of time.

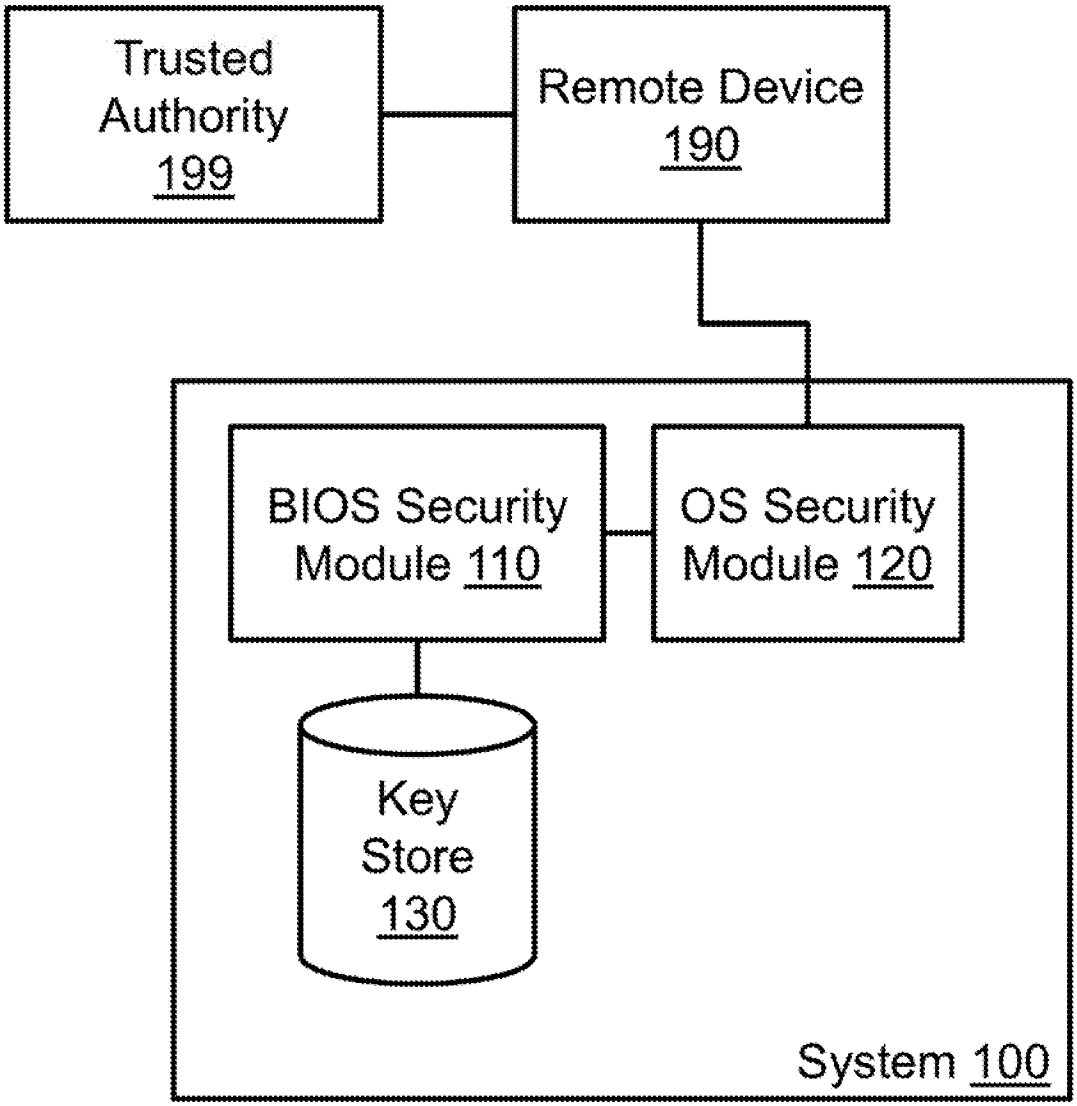

Trusted Authority 199
Remote Device 190
BIOS Security Module 110
OS Security Module 120
Key Store 130
System 100

Figure 1

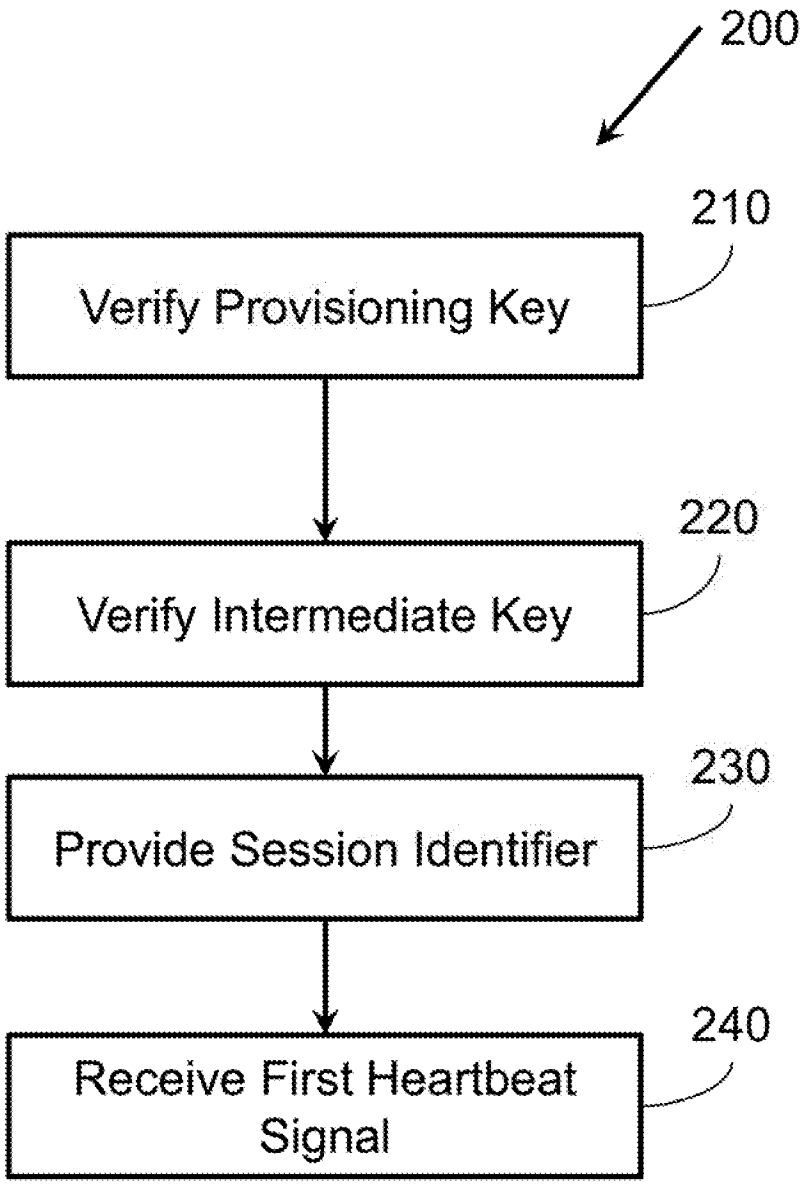

200
210
Verify Provisioning Key
220
Verify Intermediate Key
230
Provide Session Identifier
240
Receive First Heartbeat Signal

Figure 2

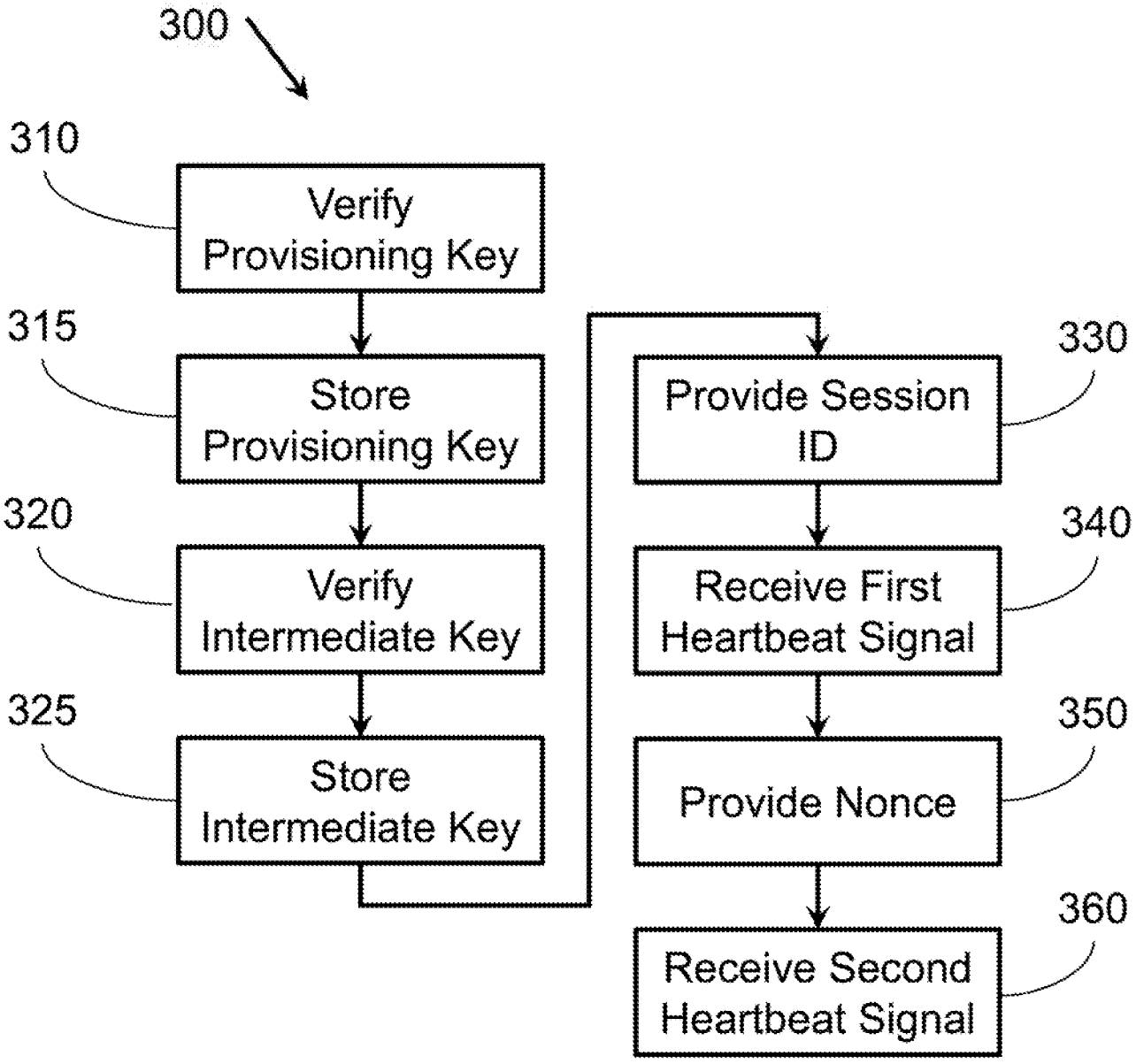

300
310
Verify Provisioning Key
315
Store Provisioning Key
320
Verify Intermediate Key
325
Store Intermediate Key
330
Provide Session ID
340
Receive First Heartbeat Signal
350
Provide Nonce
360
Receive Second Heartbeat Signal

Figure 3

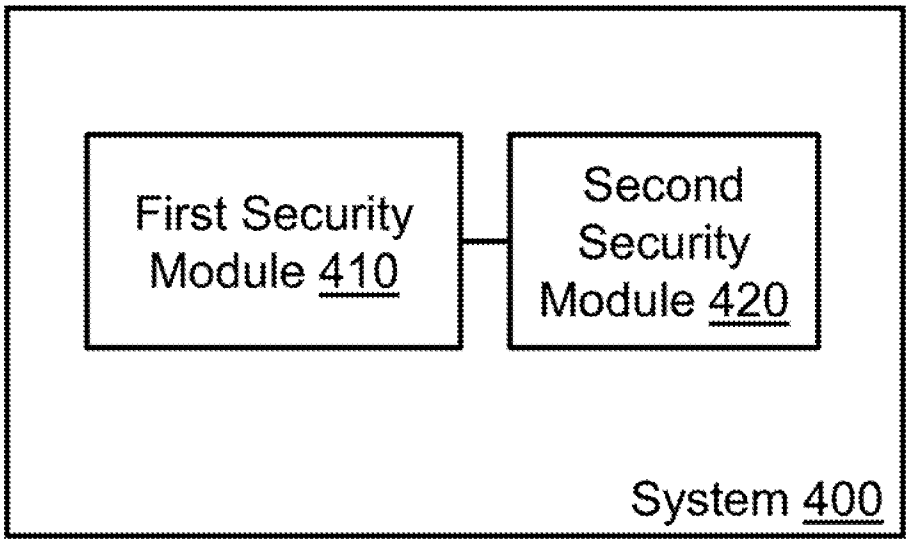

First Security Module 410
Second Security Module 420
System 400

Figure 4

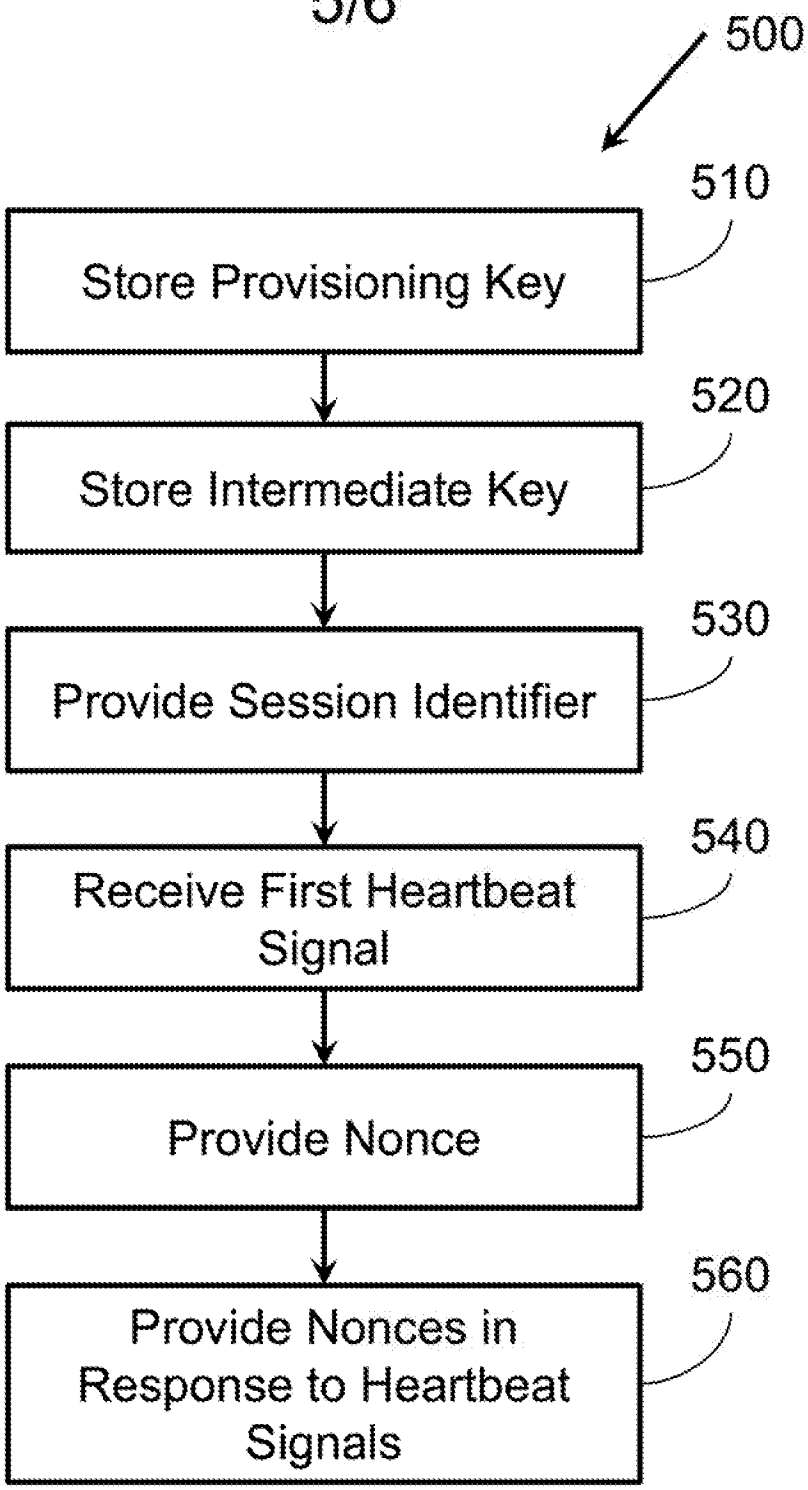

500
510
Store Provisioning Key
520
Store Intermediate Key
530
Provide Session Identifier
540
Receive First Heartbeat Signal
550
Provide Nonce
560
Provide Nonces in Response to Heartbeat Signals

Figure 5

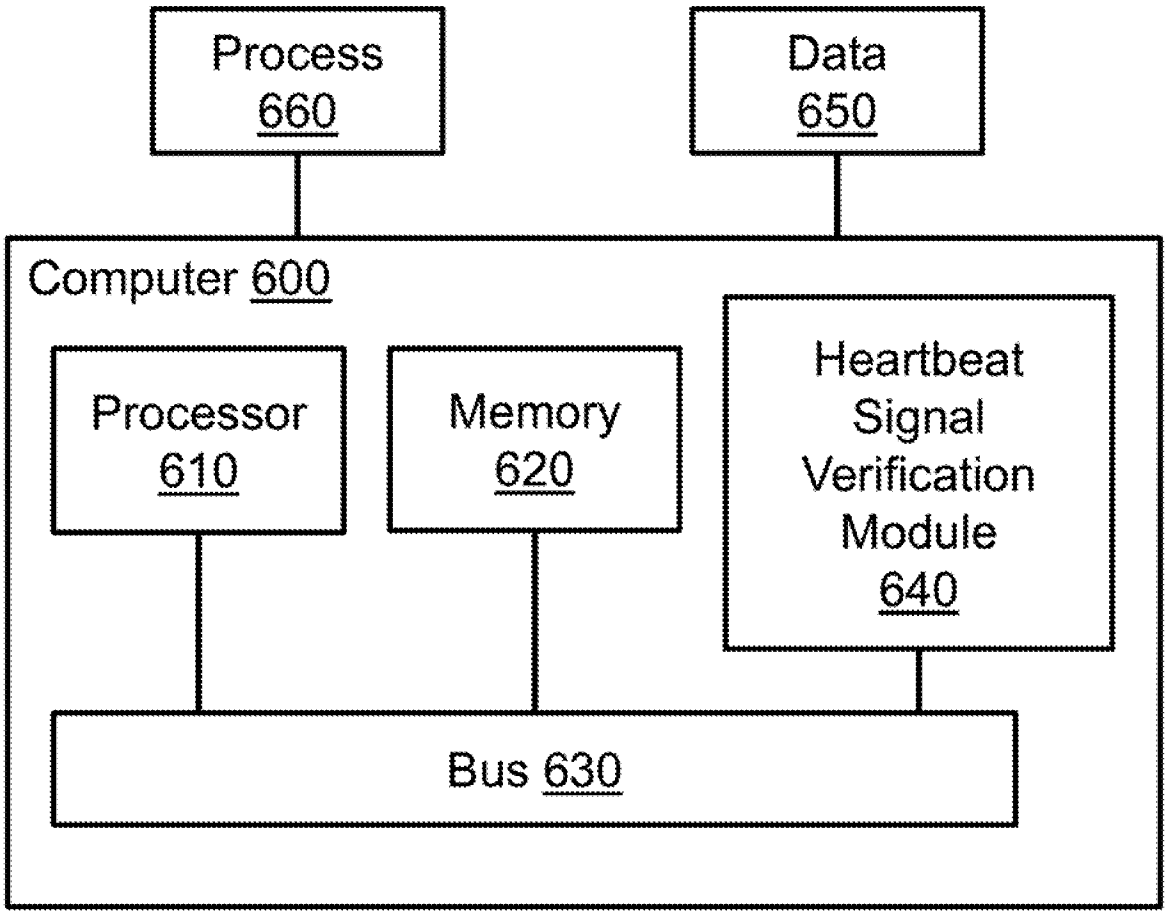

Process
660
Data
650
Computer 600
Processor
610
Memory
620
Heartbeat
Signal
Verification
Module
640
Bus 630

Figure 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2016/016369

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/57(2013.01)i, G06F 21/56(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 21/57; H04L 9/08; H04K 1/00; G06F 21/62; H04L 9/32; G08C 17/00; G06F 21/00; G06F 21/56

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: verification, heartbeat signal, provisioning key, intermediate key

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2008-0310337 A1 (DEVON WELLES) 18 December 2008 See paragraphs [0022], [0060]; and figures 1-3.	1-15
A	US 2013-0340088 A1 (PAUL THADIKARAN et al.) 19 December 2013 See paragraphs [0026], [0071]; and figures 1, 6.	1-15
A	US 2009-0217034 A1 (FRANK W. SUDIA et al.) 27 August 2009 See paragraphs [0030]-[0045]; and figure 1.	1-15
A	US 2012-0201381 A1 (PAUL TIMOTHY MILLER et al.) 09 August 2012 See paragraphs [0048]-[0070]; and figures 1-3.	1-15
A	US 2010-0037323 A1 (JACQUES LEMIEUX et al.) 11 February 2010 See paragraphs [0030]-[0042]; and figures 1A-2.	1-15

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
25 October 2016 (25.10.2016)	**26 October 2016 (26.10.2016)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea
Facsimile No. +82-42-481-8578

Authorized officer
CHIN, Sang Bum
Telephone No. +82-42-481-8398

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/US2016/016369

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2008-0310337 A1	18/12/2008	None	
US 2013-0340088 A1	19/12/2013	CN 104205044 A	10/12/2014
		EP 2828783 A1	28/01/2015
		EP 2828783 A4	25/11/2015
		KR 10-2014-0125445 A	28/10/2014
		US 9027148 B2	05/05/2015
		WO 2013-141838 A1	26/09/2013
US 2009-0217034 A1	27/08/2009	AU 1996-55521 B2	13/04/2000
		AU 1996-60842 B2	20/05/1999
		AU 1999-44619 A1	28/10/1999
		CN 1138927 A	25/12/1996
		CN 1192834 A	09/09/1998
		EP 0739560 A1	16/08/2000
		EP 0739560 B1	20/06/2001
		EP 0872080 A1	22/06/2005
		EP 0872080 B1	15/12/2010
		GB 2301919 A	18/12/1996
		GB 2337145 A	10/11/1999
		JP 04083218 B2	30/04/2008
		JP 09-507729 A	05/08/1997
		JP 11-506222 A	02/06/1999
		JP 2005-328574 A	24/11/2005
		JP 2006-246543 A	14/09/2006
		JP 2006-333520 A	07/12/2006
		JP 2007-282295 A	25/10/2007
		KR 10-1996-0706726 A	09/12/1996
		KR 10-1999-0022451 A	25/03/1999
		US 2001-0050990 A1	13/12/2001
		US 2002-0013898 A1	31/01/2002
		US 2005-0204129 A1	15/09/2005
		US 5799086 A	25/08/1998
		US 5825880 A	20/10/1998
		US 5841865 A	24/11/1998
		US 5850451 A	15/12/1998
		US 5857022 A	05/01/1999
		US 5867578 A	02/02/1999
		US 5872849 A	16/02/1999
		US 6009177 A	28/12/1999
		US 6209091 B1	27/03/2001
		US 6411716 B1	25/06/2002
		US 8364967 B2	29/01/2013
		WO 95-19672 A3	08/09/1995
		WO 96-39765 A1	12/12/1996
US 2012-0201381 A1	09/08/2012	EP 2810401 A1	10/12/2014
		HK 1204719 A1	27/11/2015
		JP 2015-510729 A	09/04/2015

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/US2016/016369

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		US 2015-0033027 A1	29/01/2015
		US 8817984 B2	26/08/2014
		US 9294448 B2	22/03/2016
		WO 2013-116019 A1	08/08/2013
US 2010-0037323 A1	11/02/2010	AU 2009-279430 A1	11/02/2010
		AU 2009-279430 B2	10/04/2014
		AU 2009-279431 A1	11/02/2010
		AU 2009-279431 B2	10/04/2014
		AU 2010-324789 A1	03/06/2011
		AU 2010-324789 B2	17/07/2014
		CA 2732830 A1	11/02/2010
		CA 2732830 C	19/01/2016
		CA 2732831 A1	11/02/2010
		CA 2732831 C	14/07/2015
		CA 2778913 A1	12/05/2011
		CA 2778925 A1	03/06/2011
		EP 2327036 A1	01/06/2011
		EP 2327037 A1	01/06/2011
		EP 2497051 A2	12/09/2012
		EP 2507736 A2	10/10/2012
		EP 2507736 B1	11/05/2016
		US 2010-0037291 A1	11/02/2010
		US 2010-0037312 A1	11/02/2010
		US 2010-0050244 A1	25/02/2010
		US 2010-0100972 A1	22/04/2010
		US 8332953 B2	11/12/2012
		US 8510825 B2	13/08/2013
		US 8556991 B2	15/10/2013
		US 8566961 B2	22/10/2013
		US 8745383 B2	03/06/2014
		WO 2010-017516 A1	11/02/2010
		WO 2010-017517 A1	11/02/2010
		WO 2011-056700 A2	12/05/2011
		WO 2011-056700 A3	18/08/2011
		WO 2011-066331 A2	03/06/2011
		WO 2011-066331 A3	29/09/2011