



- (51) **International Patent Classification:**
G06F 21/20 (2006.01) *G06F 12/00* (2006.01)
- (21) **International Application Number:**
PCT/US2011/053587
- (22) **International Filing Date:**
28 September 2011 (28.09.2011)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant** (for all designated States except US): **HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.** [US/US]; 11445 Compaq Center Drive W., Houston, Texas 77070 (US).
- (72) **Inventors; and**
- (75) **Inventors/Applicants** (for US only): **ALI, Valiuddin, Y.** [US/US]; 11445 Compaq Center Drive W., Houston, Texas 77070 (US). **WANG, Lan** [US/US]; 11445 Compaq Center Drive W., Houston, Texas 77070 (US). **WALDRON, James, Robert** [US/US]; 11445 Compaq Center Drive W., Houston, Texas 77070 (US).
- (74) **Agents:** **HABLINSKI, Reed, J.** et al.; Hewlett-packard Development Company, L.P., Intellectual Property Administration, Mail Stop 35 3404 E. Harmony Road, 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, 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, IS, JP, KE, KG, KM, 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, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, 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, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to the identity of the inventor (Rule 4.17(i))
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

Published:

- with international search report (Art. 21(3))

(54) **Title:** UNLOCKING A STORAGE DEVICE

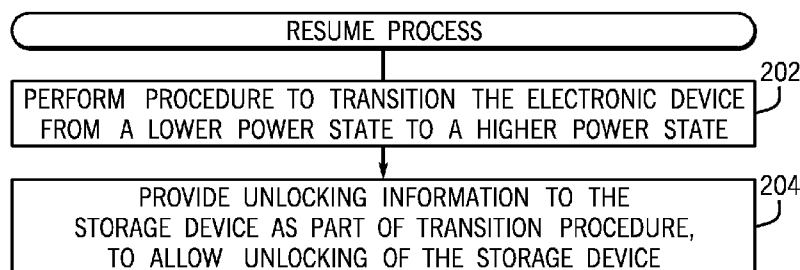


FIG. 2

(57) **Abstract:** An electronic device has a lower power state in which power to a storage device is disabled. Predetermined information stored in a memory is useable to unlock the storage device during a procedure to transition the electronic device from the lower power state to a higher power state. The predetermined information is different from a credential for use in unlocking the storage device.



UNLOCKING A STORAGE DEVICE

BACKGROUND

[0001] A storage device can include an auto-lock feature, in which removal of power from the storage device causes the storage device to become locked (such that data contained in the storage device cannot be accessed). To unlock the storage device, a credential is provided to the storage device when the storage device resumes from a powered-off state.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Some embodiments are described with respect to the following figures:

Fig. 1 is a block diagram of an example electronic device according to some implementations;

Figs. 2-5 are flow diagrams of processes according to various implementations; and

Fig. 6 is a block diagram of another example electronic device according to alternative implementations.

DETAILED DESCRIPTION

[0003] When an electronic device (e.g. a computer, a personal digital assistant, a smartphone, an electronic appliance, etc.) that includes a storage device with an auto-lock feature enters a sleep state, the electronic device may place the storage device in a powered-off state. A “sleep state” refers to a lower power state (e.g. an off state or other lower power state) of an electronic device in which power is removed from the storage device. When the electronic device resumes from the sleep state, the storage device with the auto-lock feature is powered on. As the storage device starts from the powered-off state, the storage device uses a credential to unlock itself. The “credential” can refer to a password, an authentication key, or any other secret information that is used to provide secure access of the storage device. The credential can be provided by a user, an administrator, or another source. If no credential or an incorrect credential is provided, then the storage device remains locked and data in the storage device remains inaccessible.

[0004] An example of a sleep state is the S3 state as defined by the Advanced Configuration and Power Interface (ACPI) specification. In other implementations, other types of sleep states can be employed. More generally, reference is made to a “lower power state” of an electronic device. The electronic device can transition from this lower power state to a “higher power state,” which is a power state in which power is returned to the storage device. Although reference is made to a “sleep state” in the ensuing discussion, it is noted that other lower power states can be used in other examples.

[0005] In the sleep state of the electronic device, even though power is removed from the storage device with the auto-lock feature, power is maintained to a memory of the electronic device, to allow information stored in the memory to be maintained and to be used for resuming the electronic device to a previous state. During resumption from a sleep state, a boot

module may be unable to prompt a user for a credential to be used for unlocking the storage device with the auto-lock feature. This is due to possible corruption of content of the memory should the boot module attempt to perform certain tasks (such as prompting for a credential for unlocking the storage device) other than predefined housekeeping tasks. Prior to entry into the sleep state, an operating system (OS) often stores a current state of the electronic device into the memory—this current state is used upon resumption from the sleep state. The current state can include state information of drivers, components (e.g. graphics component, keyboard, peripheral device, etc.), and other information. For the boot module to perform a prompt for a credential for unlocking the storage device, the boot module may have to modify the content of the memory, which can corrupt the current state information stored in the memory. Another reason that the boot module is not able to obtain a credential may be because the boot module may not have access to system resources employed to gather the credential from the user or other external entity.

[0006] In accordance with some implementations, when an electronic device resumes from a sleep state (transitions from the sleep state to a higher power state, such as the normal state of operation of the electronic device), a boot module is able to retrieve certain information from the memory to derive unlocking information that can be provided to a storage device with an auto-lock feature to allow the storage device to unlock. In this way, the boot module does not have to prompt a user to enter a credential for unlocking the storage device during resumption from the sleep state.

[0007] In some implementations, the boot module can be a Basic Input/Output System (BIOS) module. In other examples, other types of boot modules can be used. More generally, a “boot module” refers to any module that performs at least some tasks before an electronic device transitions from a lower power state to a higher power state.

[0008] In some examples, a storage device with an auto-lock feature can be a self-encrypting storage device. In a self-encrypting storage device, a data encryption key is used to protect data stored in the self-encrypting storage device. Circuitry in the self-encrypting storage device uses the data encryption key to encrypt data, such that encrypted data is stored. When the stored data is later accessed, the circuitry decrypts the encrypted data and provides the decrypted data to the requestor. If a user of the self-encrypting storage device later decides to dispose of the storage device or to erase data stored in the storage device, then the user can simply cause a command to be submitted (by the BIOS or another agent) to the self-encrypting storage device to erase the data encryption key, which effectively erases the stored data since such data would not be recoverable without the data encryption key.

[0009] Note that data delete operations available using operating system or file system based interfaces do not actually erase the underlying data on a persistent storage medium (e.g. magnetic storage medium, optical storage medium, flash memory, etc.). Even when files or directories are “deleted,” the underlying data still remains on the persistent storage medium. To effectively erase the underlying data, the data may have to be overwritten, or alternatively, the persistent storage medium can be destroyed. More generally, a “persistent storage medium” refers to any storage medium that maintains data stored on the storage medium even after system power is removed from the storage medium.

[0010] Note that if the access to the encryption key inside the self-encrypting storage device is not controlled, then there will be no meaningful protection of the data in the self-encrypting storage device. The self-encrypting storage device is provided with the credential discussed above to control access to the encryption key. Various mechanisms can be used to allow access to the encryption key using the credential.

[0011] Although reference is made to self-encrypting storage devices, note that techniques or mechanisms according to some implementations can also be applied to other types of storage devices that employ an auto-lock feature.

[0012] Fig. 1 is a block diagram of an example electronic device 100. The electronic device includes a processor (or multiple processors) 102. The processor(s) 102 can be connected to a memory 104 and a storage device 112 that has an auto-lock feature. The auto-lock feature locks the storage device 112 when power is removed from the storage device 112. Subsequent unlocking of the storage device 112 is accomplished by using a credential (as discussed above).

[0013] As examples, the storage device 112 can be a persistent storage device, which is a storage device that includes a persistent storage medium. The persistent storage device 112 can be a magnetic disk drive, an optical disk drive, a flash memory, or another type of storage device. As examples, the memory 104 can be implemented with integrated circuit memory device(s), such as dynamic random access memory (DRAM) device(s), static random access memory (SRAM) device(s), flash memory device(s), or any other type of memory device.

[0014] The memory 104 stores “predetermined” information 106, which is accessible by the processor(s) 102 (under control of a boot module such as a BIOS module, for example) during resumption from a sleep state. The predetermined information 106 is stored in the memory 104 prior to the electronic device entering the sleep state. Note that in implementations where the memory 104 includes volatile memory, the memory 104 remains powered (even though the storage device 112 is powered off) to allow the predetermined information 106 to remain available for subsequent use. As further discussed above, in addition to the predetermined information 106, memory 104 can further store current state information of the electronic

device 100 relating to drivers, components, and so forth. Such current state information is used to restore the state of the electronic device 100 when the electronic device 100 next resumes from the sleep state.

[0015] During a procedure in which the electronic device transitions from the sleep state to a higher power state, the processor(s) 102 can retrieve (108) the predetermined information 106 from the memory 104, and can use the predetermined information to derive unlocking information (110) that is provided to the storage device 112. An unlocking module 114 in the storage device 112 uses the unlocking information 110 to unlock the storage device 112. Such unlocking of the storage device 112 is accomplished without prompting a user for a credential to unlock the storage device 112.

[0016] By using the predetermined information 106 and unlocking information 110, unlocking of the storage device 112 during resumption from a sleep state can be accomplished without having to prompt for input of a credential. In this manner, automated unlock of the storage device 112 upon resumption from the sleep state is possible.

[0017] In some examples, the predetermined information 106 can be stored in a portion of the memory 104 that is allocated for system management mode (SMM). The electronic device 100 can enter SMM to perform certain types of tasks, such as error management, power management tasks, security tasks, and so forth. The portion of the memory 104 allocated to SMM is referred to as SMM memory. SMM is an operating mode in which execution of the operating system of the electronic device 100 is suspended.

[0018] In other implementations, the predetermined information 106 can be stored in another portion of the memory 104.

[0019] The predetermined information 106 and unlocking information 110 can differ in implementations. In some implementations, the predetermined

information 106 includes a seed key (also referred to as a secret or shared secret) and a random number (or counter value). The seed key and random number (or counter value) are used to generate an encryption key—this encryption key is part of the unlocking information 110 that is provided to the storage device 112. The unlocking module 114 generates the credential for unlocking the storage device 112 based on the encryption key. For example, the unlocking module 114 can decrypt an encrypted version of the credential stored in the storage device 112 using the encryption key—the decrypted credential is then useable to unlock the storage device 112.

[0020] In alternative implementations, the predetermined information 106 can include a random number. In such implementations, the unlocking information 110 also includes the random number. Thus, in such implementations, during a procedure to transition the electronic device 100 from a sleep state to a higher power state, the random number is retrieved from the memory 104, and the processor(s) 102 send(s) the unlocking information 110 that includes the retrieved random number to the storage device 112. The unlocking module 114 in the storage device 112 produces the credential to unlock the storage device 112 based on the random number. For example, the unlocking module 114 can decrypt an encrypted version of the credential using the random number.

[0021] In other implementations, other forms of the predetermined information 106 stored in the memory 104 can be used. For enhanced security, the predetermined information 106 is different from the credential that is used to unlock the storage device 112. The portion of the memory 104 (such as the SMM memory) that stores the predetermined information 106 can be accessed by an unauthorized entity, such as malware. If the predetermined information 106 contains the credential, then unauthorized access of the credential in the memory 104 can be achieved

[0022] To further enhance the protection afforded by some implementations, once the boot module unlocks the storage device when resuming from a lower power state, new predetermined information 106 (e.g. new seed key or counter value or random number) can be generated and exchanged with the storage device, which can later use the new predetermined information 106 to generate the credential to allow access to the storage device (such as to allow access of an encryption key that encrypts the data on the storage device). In this manner, even if an unauthorized entity is able to observe the predetermined information 106 stored in the memory 104, the changing nature of the predetermined information 106 can reduce the chance of previously obtained predetermined information being usable when the unauthorized entity later gains physical access to the electronic device.

[0023] Although various alternative implementations have been discussed above, note that further alternative implementations are also contemplated that use different predetermined information 106 and unlocking information 110.

[0024] Fig. 2 is a flow diagram of a resume process of the electronic device 100, in accordance with some implementations. The process can be performed by the processor(s) 102 of Fig. 1, for example, such as under control of a boot module. The process performs (at 202) a procedure to transition the electronic device 100 from a lower power state (e.g. sleep state) to a higher power state, where power to the storage device 112 of the electronic device 100 is disabled in the lower power state

[0025] The process further provides (at 204) unlocking information (110 in Fig. 1) to the storage device 112 to allow unlocking of the storage device as part of the procedure to transition the electronic device 100 from the lower power state to the higher power state, where the provided unlocking

information is based on the predetermined information (106 in Fig. 1) stored in the memory 104 prior to placing the electronic device 100 in the lower power state.

[0026] Fig. 3 is a flow diagram of a sleep process of the electronic device 100, in accordance with some implementations. The process can be performed by the processor(s) 102 of Fig. 1, for example, such as under control of a boot module. The process stores (at 302) predetermined information 106 (Fig. 1) in the memory 104 of the electronic device.

[0027] After storing the predetermined information 106 in the memory, the process transitions (at 304) the electronic device from a higher power state to a lower power state (e.g. sleep state), where power is removed from the storage device 112 when the electronic device is in the lower power state. The stored predetermined information 106 is useable to unlock the storage device 112 when the electronic device next transitions from the lower power state to the higher power state.

[0028] Fig. 4 is a flow diagram of a process according to alternative implementations. During a cold boot procedure, a boot module (400) prompts (at 402) a user to enter a credential for unlocking the storage device 112. The user can be located at the electronic device (in which case the prompt can be presented in a display device of the electronic device), or alternatively the user can be located at a remote device (in which case the prompt is communicated over a network to the remote device). "Cold boot" can refer to starting the electronic device 100 from a low power state or an off state in which power to components, including the memory 104 of Fig. 1, is removed. Examples of the low power state and off state include the ACPI S4 and S5 states, respectively. In other examples, other low power and off states can be used.

[0029] In response to the prompt, the user inputs the credential (referred to as "P"), which is received (at 404) by the boot module 400. Alternatively, the provision of the prompt at 402 can be performed by another module in the electronic device 100 instead of the boot module 400. Such other module can be a full volume encryption module that is responsible for obtaining a password for unlocking the storage device 112. In implementations where the credential is obtained by the full volume encryption module, the full volume encryption module can communicate this credential to the boot module 400 using one of several different types of interfaces, such as a Windows Management Interface (WMI), or an interrupt interface (e.g. INT 15h or INT 1a), or any other type of interface.

[0030] In implementations according to Fig. 4, the boot module 400 generates (at 406) a random number C as a counter, and further generates a shared secret K between the boot module 400 and the storage device 112. In some examples, the shared secret K can be an SHA-1 (secure hash algorithm-1) hash of a value that is an exclusive-OR (XOR) of the counter C with the credential P. SHA-1 is a cryptographic hash function. In some examples, an HOTP technique is used, where HOTP refers to HMAC-SHA-1 based-one-time-password, and where HMAC stands for hash-based message authentication code. Information relating to the HOTP technique can be found in Request for Comments (RFC) 4226, entitled "HOTP: An HMAC-Based One-Time Password Algorithm," dated December 2005. In other examples, other techniques for computing the secret K and random number C can be used.

[0031] According to the HOTP technique, K is referred to as a shared secret, and C is referred to as a counter. The counter C can be initialized to a random number. The shared secret K can be generated randomly. As noted above, in some examples, K is computed according to

$$K = \text{SHA-1}(P \text{ XOR } C).$$

The operation $P \text{ XOR } C$ represents an exclusive-OR of P and C . The cryptographic hash function (SHA-1) produces a hash value (K) based on $P \text{ XOR } C$. In other examples, other functions for producing K can be used.

[0032] Next, the boot module 400 saves (at 408) the shared secret K and the counter C in the SMM memory (or in some other memory portion). In the context of Fig. 1, the K and C values are part of the predetermined information 106 stored in the memory 104.

[0033] The boot module 400 then issues (at 410) a clear command to the storage device 112, to cause the storage device 112 to clear any previous information relating to unlocking the storage device 112. The boot module 400 then sends (at 412) the values P , K , and C to the storage device 112. After sending the credential P to the storage device 112, the boot module 400 deletes the credential P from the memory 104.

[0034] The received credential P is used by the storage device 112 to unlock (at 414) the storage device 112. The storage device 112 also calculates (at 416) an encryption key K' based on the K and C values received from the boot module 400, as follows:

$$K' = \text{HOTP}(K, C).$$

[0035] The foregoing operation, $\text{HOTP}(K, C)$ can be defined as follows:

$$\text{HOTP}(K, C) = \text{Truncate}(\text{HMAC-SHA-1}(K, C)).$$

[0036] The $\text{Truncate}()$ operation selects some predefined number of bytes (e.g. 4 bytes) from the 160-bit value resulted from SHA-1. In some examples, the $\text{HMAC}(K, C)$ operation, represented by $\text{HMAC-SHA-1}(K, C)$ above, is defined as follows:

$H(\cdot)$ is a cryptographic hash function,

\parallel denotes concatenation,

opad is the outer padding (0x5c5c5c...5c5c, one-block-long hexadecimal constant), and

ipad is the inner padding (0x363636...3636, one-block-long hexadecimal constant).

Then HMAC(K, C) is mathematically defined as

$$\text{HMAC}(K, C) = H((K \text{ XOR opad}) \parallel H((K \text{ XOR ipad}) \parallel C)).$$

[0037] Although an example HOTP() definition is provided above, different functions for producing the encryption key K' based on K and C can be used in other examples.

[0038] The encryption key K' is then used by the storage device 112 to encrypt (at 418) the credential P to produce an encrypted credential P' :

$$P' = \text{Encrypt}(P \text{ with } K').$$

[0039] The encrypted credential P' and the counter C are then saved (at 420) in the storage device 112. Note that the clear text (unencrypted) version of the credential, P , is deleted by the storage device 112.

[0040] The boot module 400 can next send (at 422) a lock command to the storage device 112. This lock command is to indicate that no further modifications of the encrypted credential P' are to be performed until after the next boot cycle (when the electronic device 100 is again in the boot environment). The lock command is effectively a write-once command to prevent modification of P' outside the boot environment (to prevent malicious entities from modifying P' to gain access to the storage device 112). By encrypting the credential P , physical attacks on the storage device 112 may

yield just the encrypted credential P' (and not the clear text version of the credential P).

[0041] Also, since just K and C are stored in the memory 104, any unauthorized access of the memory 104, such as by using a memory freeze attack, would yield just the K and C information, which cannot be used by an unauthorized entity to derive the credential P. A memory freeze attack refers to an attack in which the temperature of the memory 104 is reduced to a freezing temperature such that the data loss process when power is removed from the memory 104 is slowed down. This allows an unauthorized user to remove the memory 104 from the electronic device (which causes power to be lost), followed by re-connecting the memory 104 in another system, where power is again provided to the memory 104 in an attempt to retrieve information previously stored in the memory 104. Also, the K and C values in the memory 104 are lost upon a system cold boot, which protects against replay attacks.

[0042] As further shown in Fig. 4, the electronic device 100 is next transitioned (at 424) to the sleep state.

[0043] Upon resuming (at 425) from the sleep state, the boot module 400 is initiated and performs a procedure to transition the electronic device 100 from the sleep state to a higher power state. During this transition procedure, the boot module 400 retrieves the K and C values from the memory 104, and calculates (at 426) the encryption key K' using, for example, $K' = \text{HOTP}(K, C)$, similar to the calculation of K' at 416 in the storage device 112.

[0044] Additionally, in some examples, the boot module 400 updates (at 428) the K and C values as follows: increment the counter C, and set $K = K'$. The updated K and C values are to be used in the next sleep state resume cycle (after the electronic device 100 next transitions to the sleep state and then subsequently resumes from the sleep state). In other examples,

updating of the K and C values is not performed. Updating the K and C values such that different K and C values are used in different resume cycles provides enhanced protection.

[0045] The boot module 400 sends (at 430) the encryption key K' to the storage device 112. In the context of Fig. 1, the encryption key K' sent at 430 is part of the unlocking information 110.

[0046] The storage device 112 uses (at 432) the encryption key K' to decrypt the encrypted credential P' that is stored in the storage device 112, to produce decrypted credential P as follows:

$$P = \text{Decrypt}(P' \text{ with } K').$$

[0047] The storage device 112 then uses (at 434) the decrypted credential P to unlock the storage device 112.

[0048] In addition, in some examples, the storage device 112 also updates (at 436) its K and C values, to synchronize with the K and C update (428) performed by the boot module 400. The update (436) is performed as follows: increment the counter C, and set $K = K'$. In other examples, the update of K and C is not performed.

[0049] In examples where K and C have been updated, the storage device 112 then computes $\text{HOTP}(K, C)$ to derive K', and re-encrypts the credential P with K' to produce P'. The storage device 112 then saves (at 438) the encrypted credential, P', with the counter C.

[0050] Fig. 5 is a flow diagram of a process according to further alternative implementations. During a cold boot procedure, the boot module (400) prompts (at 502) a user to enter a credential for unlocking the storage device 112. In response to the prompt, the user inputs the credential (referred to as "P"), which is received (at 504) by the boot module 400. Alternatively, the provision of the prompt at 502 can be performed by the full volume encryption module, as discussed above. The full volume encryption module can communicate the user-input credential to the boot module 400 through an interface.

[0051] Next, the boot module 400 generates (at 506) a random number, R, such as by using a pseudo-random number generator. The boot module 400 saves (at 508) the random number R in the memory 104 (Fig. 1). In the context of Fig. 1, the random number R is part of the predetermined information 106 stored in the memory 104.

[0052] The boot module 400 sends (at 509) a clear command to the storage device 112 (similar to clear command sent at 410 in Fig. 4). The boot module 400 also sends (at 510) the user-input credential, P, and the random number R to the storage device 112. After sending the credential P to the storage device 112, the boot module 400 deletes the credential P from the memory 104.

[0053] The received credential P is used by the storage device 112 to unlock (at 512) the storage device 112. The storage device 112 then encrypts (at 514) the credential P with the random number, R, to produce encrypted credential P':

$$P' = \text{Encrypt}(P \text{ with } R).$$

[0054] The encrypted credential P' is saved (at 516) in the storage device 112. Note that the storage device 112 deletes the random number R. The

boot module 400 can next send (at 518) a lock command to the storage device 112 (similar to lock command sent at 422 in Fig. 4). As further shown in Fig. 5, the electronic device 100 is next transitioned (at 520) to the sleep state.

[0055] As part of resuming (at 521) from the sleep state, the boot module 400 is initiated and performs a procedure to transition the electronic device 100 from the sleep state to a higher power state. During this transition procedure, the boot module 400 generates (at 522) a new random number, R2. The boot module 400 also retrieves (at 524) the random number R previously stored to the memory 104. The random numbers R and R2 are sent (at 526) from the boot module 400 to the storage device 112. In the context of Fig. 1, the random number R at 526 is part of the unlocking information 110.

[0056] The new random number R2 is to be used in the next sleep state resume cycle. Using different random numbers in different sleep state resume cycles provides enhanced protection. In other examples, the same random number can be used in multiple sleep state resume cycles.

[0057] The storage device 112 uses (at 528) the received random number R to decrypt the encrypted credential P' that is stored in the storage device 112, to produce decrypted credential P as follows:

$$P = \text{Decrypt}(P' \text{ with } R).$$

[0058] The storage device 112 then uses (at 530) the decrypted credential P to unlock the storage device 112.

[0059] In addition, in some examples, the storage device 112 next re-encrypts the credential P with the new random number R2, also received from the boot module 400, as follows:

$P' = \text{Encrypt}(P \text{ with } R2).$

[0060] The new random number R2 is used to perform the decryption in the next sleep state resume cycle (this is part of the examples in which different random numbers can be used in different sleep state resume cycles). The re-encrypted credential P' (encrypted with R2) is next saved (at 532) in the storage device 112.

[0061] Using techniques or mechanisms according to some implementations, enhanced protection is provided against various forms of attacks that may seek unauthorized access of a storage device that has an auto-lock feature.

[0062] Fig. 6 illustrates another example electronic device 600, which includes various components. The components in the electronic device 600 that are the same as the electronic device 100 are referenced with the same reference numerals. In addition to the components already discussed in connection with Fig. 1, the electronic device 600 further includes a BIOS module 602, which can perform tasks as discussed above in connection with Figs. 2-5. Also, the electronic device 600 includes an operating system 604. In some examples, the electronic device 600 also includes a full volume encryption module 606, discussed above in connection with Fig. 4 or 5.

[0063] A user interface module 608 in the electronic device 600 presents a user interface 612 in a display device 610. A user can enter information (such as credential P) through the user interface 612. Alternatively, the credential P can be provided by a user at a remote device, where the credential P is communicated over a network to the electronic device 600. The electronic device 600 includes a network interface 614 to communicate over such network.

[0064] The various modules discussed above (including the modules 602, 604, 608, and 608) can be implemented as machine-readable instructions that are loaded for execution on a processor or processors (e.g. 102 in Fig. 1 or 6). A processor can include a microprocessor, microcontroller, processor module or subsystem, programmable integrated circuit, programmable gate array, or another control or computing device.

[0065] Data and instructions are stored in respective storage devices, which are implemented as one or more computer-readable or machine-readable storage media. The storage media include different forms of memory including semiconductor memory devices such as dynamic or static random access memories (DRAMs or SRAMs), erasable and programmable read-only memories (EPROMs), electrically erasable and programmable read-only memories (EEPROMs) and flash memories; magnetic disks such as fixed, floppy and removable disks; other magnetic media including tape; optical media such as compact disks (CDs) or digital video disks (DVDs); or other types of storage devices. Note that the instructions discussed above can be provided on one computer-readable or machine-readable storage medium, or alternatively, can be provided on multiple computer-readable or machine-readable storage media distributed in a large system having possibly plural nodes. Such computer-readable or machine-readable storage medium or media is (are) considered to be part of an article (or article of manufacture). An article or article of manufacture can refer to any manufactured single component or multiple components. The storage medium or media can be located either in the machine running the machine-readable instructions, or located at a remote site from which machine-readable instructions can be downloaded over a network for execution.

[0066] In the foregoing description, numerous details are set forth to provide an understanding of the subject disclosed herein. However, implementations may be practiced without some or all of these details. Other

implementations may include modifications and variations from the details discussed above. It is intended that the appended claims cover such modifications and variations.

What is claimed is:

1 1. A method comprising:
2 performing a procedure to transition an electronic device from a lower
3 power state to a higher power state, wherein power to a storage device of the
4 electronic device is disabled in the lower power state; and
5 providing unlocking information to the storage device to allow for
6 generation of a credential to unlock the storage device as part of the
7 procedure to transition the electronic device from the lower power state to the
8 higher power state, wherein the provided unlocking information is based on
9 predetermined information stored in a memory prior to placing the electronic
10 device in the lower power state, the predetermined information different from
11 the credential.

1 2. The method of claim 1, further comprising:
2 as part of the procedure to transition the electronic device from the
3 lower power state to the higher power state, computing an encryption key
4 based on the predetermined information stored in the memory,
5 wherein providing the unlocking information comprises providing the
6 computed encryption key to the storage device, wherein the computed
7 encryption key is useable by the storage device to generate a credential to
8 unlock the storage device.

1 3. The method of claim 2, wherein computing the encryption key
2 comprises computing the encryption key based on a seed key and a counter
3 value that were stored in the memory prior to placing the electronic device in
4 the lower power state, where the predetermined information includes the seed
5 key and the counter value.

- 1 4. The method of claim 3, wherein computing the encryption key
2 comprises computing the encryption key using a HMAC (hash-based
3 message authentication code)-based-one-time-password technique.
- 1 5. The method of claim 1, wherein providing the unlocking information
2 comprises providing a random number useable by the storage device to
3 compute the credential for unlocking the storage device, where the
4 predetermined information includes the random number.
- 1 6. The method of claim 1, further comprising updating the predetermined
2 information for use in producing unlocking information to be provided to the
3 storage device in a next transition from the lower power state to the higher
4 power state.
- 1 7. The method of claim 1, further comprising:
2 as part of a cold boot procedure, providing a prompt to a user to enter
3 a credential to unlock the storage device; and
4 storing the predetermined information in the memory after obtaining the
5 credential in response to the prompt.
- 1 8. The method of claim 7, further comprising computing a secret based on
2 the credential, wherein the predetermined information includes the secret.
- 1 9. The method of claim 7, further comprising computing a random number
2 after obtaining the credential, wherein the predetermined information includes
3 the random number.

1 10. An article comprising at least one machine-readable storage medium
2 storing instructions that upon execution cause an electronic device to:
3 store predetermined information in a memory of the electronic device;
4 and
5 after storing the predetermined information in the memory, transition
6 the electronic device from a higher power state to a lower power state,
7 wherein power to a storage device is disabled when the electronic device is in
8 the lower power state, and wherein the predetermined information is useable
9 to unlock the storage device when the electronic device transitions from the
10 lower power state to the higher power state, the predetermined information
11 being different from a credential for use in unlocking the storage device.

1 11. The article of claim 10, wherein the predetermined information is
2 maintained in the memory during the lower power state.

1 12. The article of claim 10, wherein the instructions upon execution cause
2 the electronic device to further:
3 during a procedure transitioning the electronic device from the lower
4 power state to the higher power state,
5 retrieve the predetermined information from the memory;
6 produce unlocking information based on the predetermined
7 information; and
8 send the unlocking information to the storage device to allow the
9 storage device to generate the credential to unlock the storage device.

1 13. The article of claim 12, wherein the retrieving, producing, and sending
2 are performed by a Basic Input/Output System (BIOS) module.

1 14. The article of claim 10, wherein the instructions upon execution cause
2 the electronic device to further:
3 update the predetermined information;
4 store the updated predetermined information in the memory for use in a
5 subsequent procedure to transition the electronic device from the lower power
6 state to the higher power state, to unlock the storage device.

1 15. An electronic device comprising:
2 a memory;
3 a storage device having an auto-lock feature that specifies that
4 unlocking of the storage device is to be performed based on a credential; and
5 at least one processor to:
6 perform a procedure to transition the electronic device from a
7 lower power state to a higher power state, wherein power to the storage
8 device is disabled in the lower power state; and
9 provide unlocking information to the storage device to allow
10 unlocking of the storage device using the credential as part of the procedure
11 to transition the electronic device from the lower power state to the higher
12 power state, wherein the provided unlocking information is based on
13 predetermined information stored in the memory prior to placing the electronic
14 device in the lower power state, the predetermined information being different
15 from the credential.

1 / 4

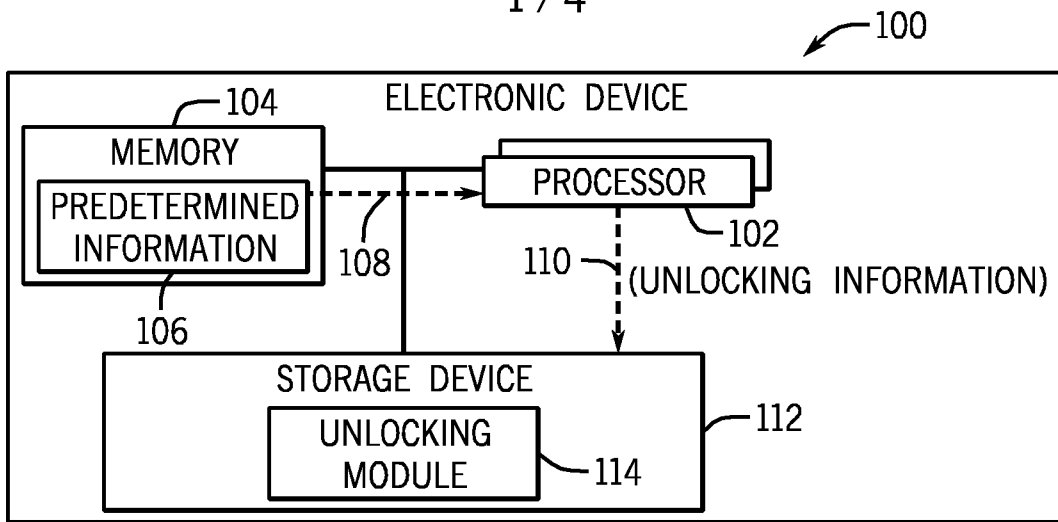


FIG. 1

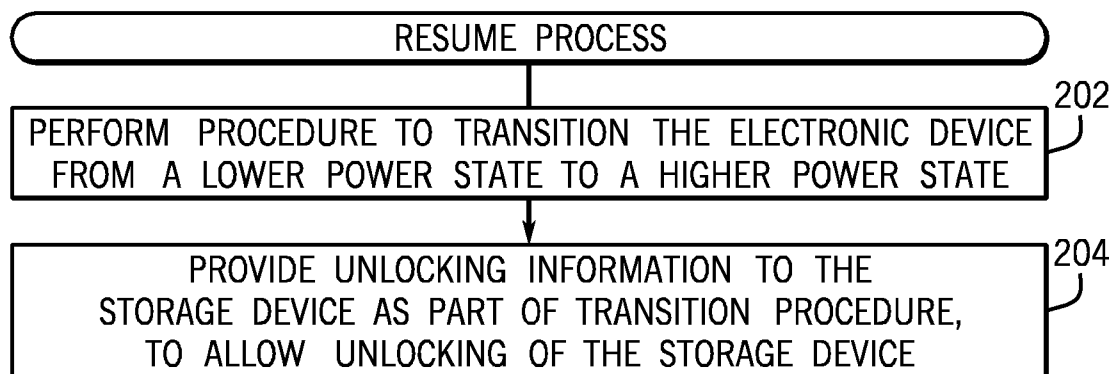


FIG. 2

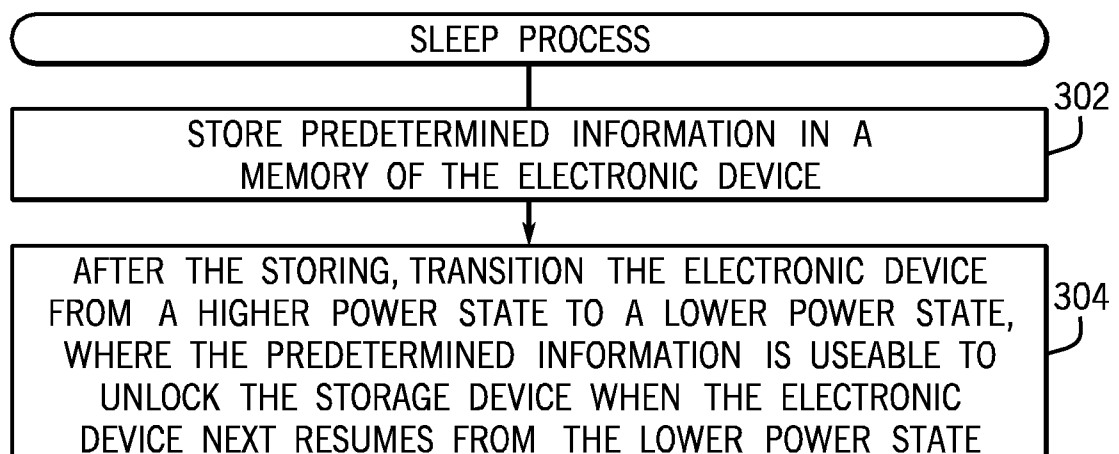


FIG. 3

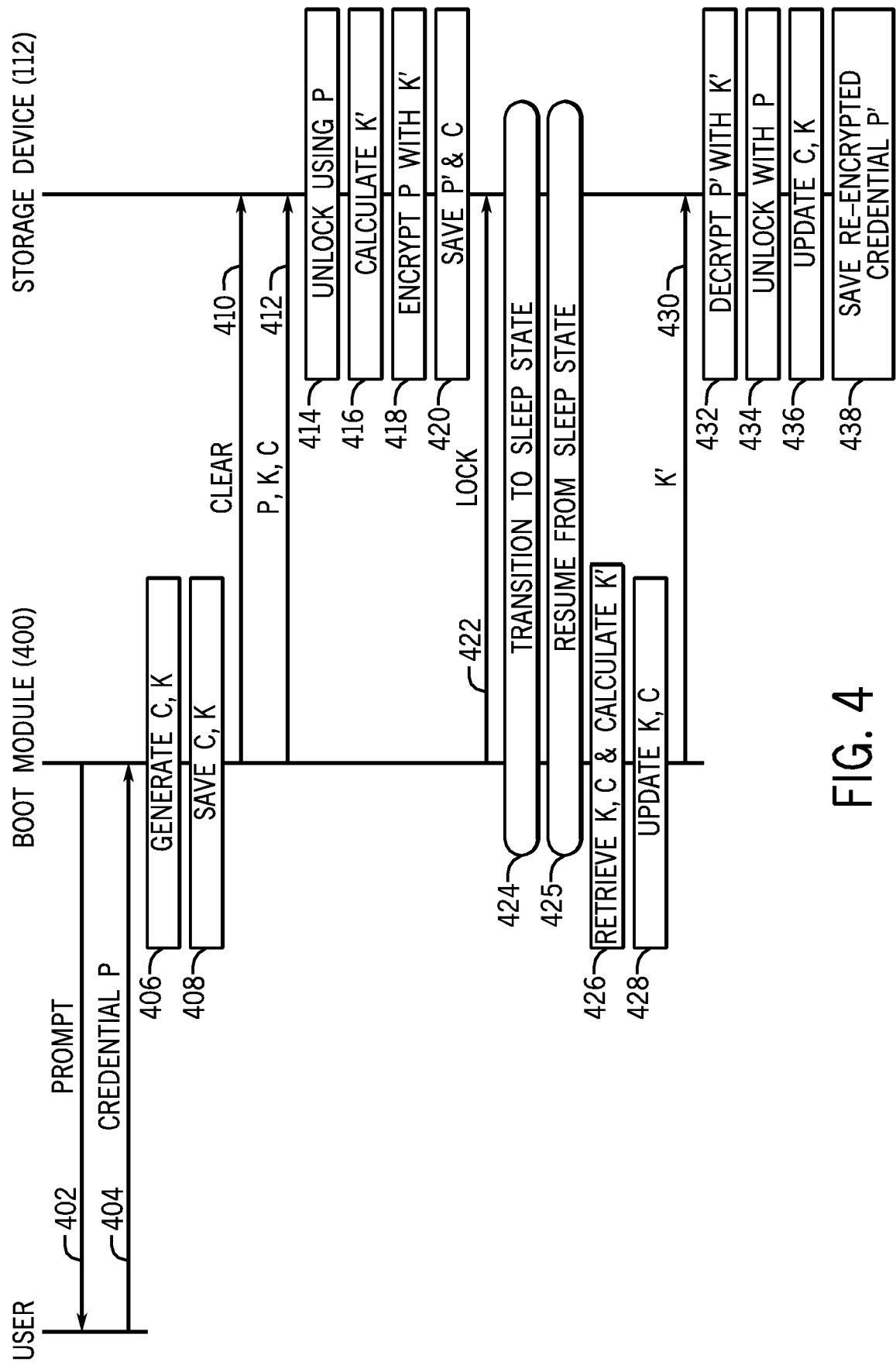


FIG. 4

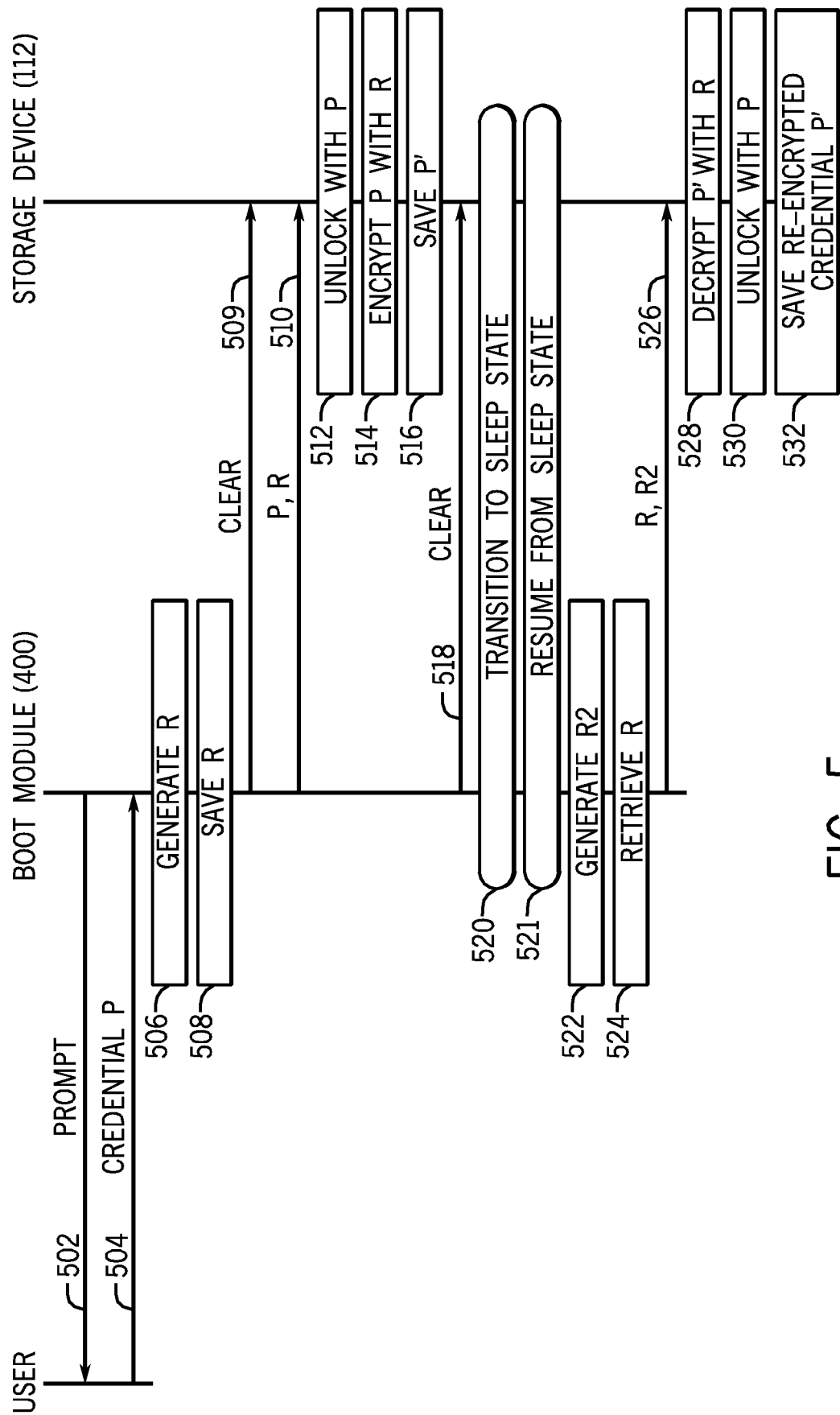


FIG. 5

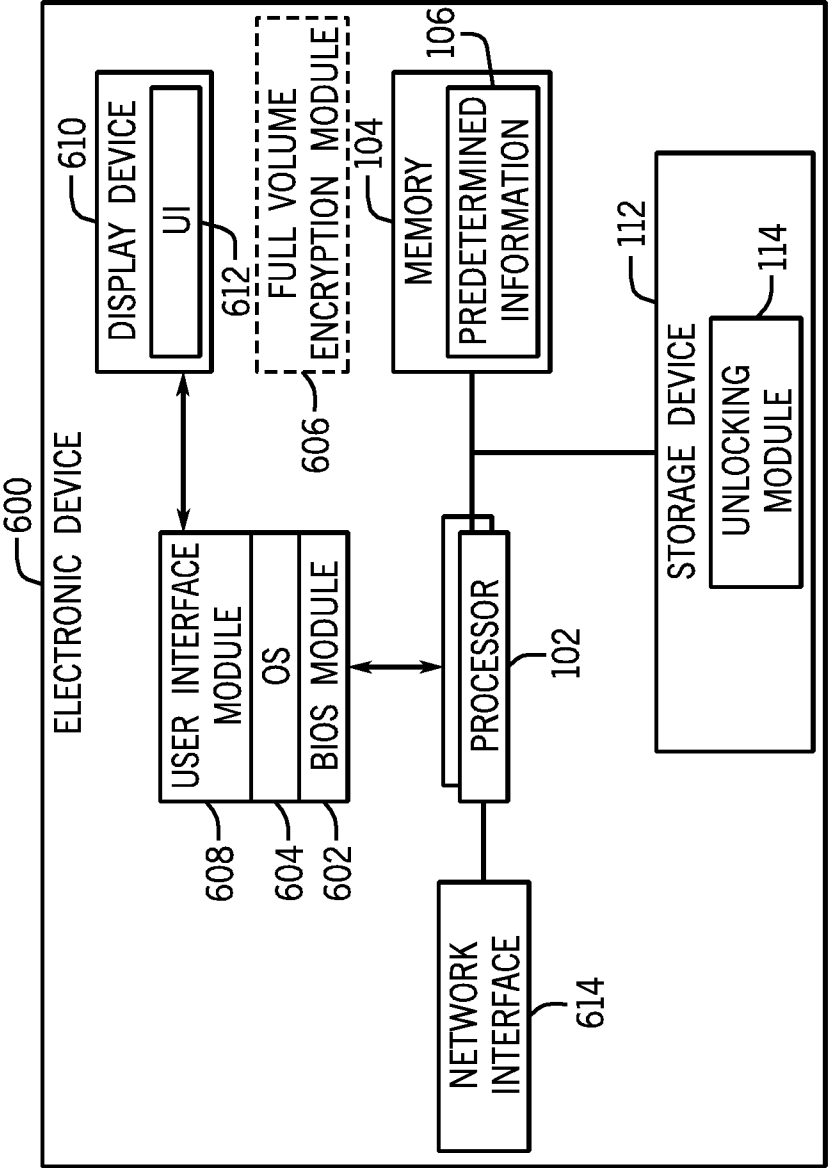


FIG. 6

A. CLASSIFICATION OF SUBJECT MATTER**G06F 21/20(2006.01)i, G06F 12/00(2006.01)i**

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 21/20; H04L 9/32; G06F 12/14; H04L 29/06; G06F 9/00; G06F 7/04

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: unlock, storage, memory, transition, lower, higher, power, credential, encryption, key, hash, random, boot

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2008-0222423 A1 (RODRIGUEZ FELIPE et al.) 11 September 2008 See abstract; paragraphs [12] - [51]; figures 3, 4.	1,2,5,7,10-13,15
A	See the whole document.	3,4,6,8,9,14
Y	US 2004-0236919 A1 (OKAUE TAKUMI et al.) 25 November 2004 See abstract; paragraphs [102] - [118]; figures 7-8.	1,2,5,7,10-13,15
A	See the whole document.	3,4,6,8,9,14
A	US 2009-0006857 A1 (CHENG ANTON) 01 January 2009 See abstract; paragraphs [15] - [19]; figure 2.	1-15
A	US 7743241 B1 (MOORE KEVIN) 22 June 2010 See abstract; column 4, line 18 - column 11, line 44; figures 2-4.	1-15
A	US 2006-0156035 A1 (TRAN et al.) 13 July 2006 See abstract; paragraphs [16] - [21]; figures 2-5.	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 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

28 MARCH 2012 (28.03.2012)

Date of mailing of the international search report

09 APRIL 2012 (09.04.2012)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon, 189 Cheongsu-ro,
Seo-gu, Daejeon 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

Shin Sang Gil

Telephone No. 82-42-481-8480



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2011/053587

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2008-0222423 A1	11.09.2008	WO 2008-109150 A1	12.09.2008
US 2004-0236919 A1	25.11.2004	CN 1292357 C0	27.12.2006
		CN 1564981 A	12.01.2005
		CN 1564981 C0	27.12.2006
		EP 1519277 A1	30.03.2005
		JP 03-979194 B2	19.09.2007
		JP 2004-030101 A	29.01.2004
		KR 10-2005-0013523 A	04.02.2005
		WO 04-001610 A1	31.12.2003
US 2009-0006857 A1	01.01.2009	None	
US 7743241 B1	22.06.2010	None	
US 2006-0156035 A1	13.07.2006	US 7685634 B2	23.03.2010