

[0199] The secure code descriptor may be double-buffered in the external flash. This may guarantee that the flash secure code descriptor is not corrupted on a failing write. This also may prevent the descriptor from being corrupted if power is removed while the descriptor is being updated. A description of one embodiment of the double-buffering scheme follows.

[0200] Upon boot, two descriptors are authenticated using the HMAC-SHA1 flash key. If both authentications are successful, the descriptor with the highest sequence number may be copied to the current descriptor in volatile memory.

[0201] The upgrade start command updates the current descriptor in flash memory. Whenever a descriptor in flash is modified, the older descriptor in the flash is overwritten. In case the power is removed when the descriptor is being updated in flash, the current descriptor is valid and may indicate that valid firmware is loaded in flash.

[0202] The upgrade complete command writes a new descriptor to the flash memory. The current descriptor modified during the upgrade start command is not overwritten. The older descriptor is overwritten. If power is removed, the current descriptor is still valid and indicates there is not valid firmware loaded in flash.

[0203] Referring again to FIG. 8, at block 830 the TPM will increment the internal sequence number and load it into the flash memory. After the upgrade operation is complete, the system will be reset (block 832) so that during the next boot procedure the new function table and secure code descriptor information will be used for subsequent operations.

[0204] In some embodiments multiple signing keys may be used to support firmware for different OEMs and/or to prevent replay attacks. The example of FIG. 10 illustrates how multiple secure code loads may use multiple authentication keys.

[0205] As represented by block 1002, a first private-public key pair (e.g., an RSA asymmetric key) is defined. The corresponding private key (Ksc\_init) may be securely maintained within a secure signing environment. This private key is kept secret and may be used to sign at least the first secure code load for all OEM's. The corresponding public key (Ksc\_init-pub) may be stored in the TPM (e.g., in on-chip ROM).

[0206] A fixed RSA private key (Ksc\_init) may be used to perform the signature over the secure code during a first secure code load and, until the keys are changed, subsequent secure code loads (block 1004). Thus, the TPM will initially use the corresponding public key (Ksc\_init-pub) to verify any new secure code. The secure code may include, for example, one or more of new secure code, a new function table and a secure code descriptor.

[0207] In some embodiments a secure code load may be used to send a secure code descriptor with a new public key to the TPM. In this case, a second private-public key pair is defined (block 1006). Again, the corresponding private key may be securely maintained within the secure signing environment. The corresponding public key (Ksc\_oem0-pub) may be incorporated into the secure code descriptor, signed using the initial private key (block 1008) and then sent to the TPM (block 1010).

[0208] The upgrade commands are used to load the secure code image to flash. The upgrade start command verifies the new secure code descriptor and stores the new code descriptor in TPM volatile memory. Upon successful verification, the TPM may be deactivated, and the on-chip function table may be referenced. Flash may be securely updated to clear the firmware loaded field in the secure code descriptor. The upgrade update command securely writes each block of secure code into flash. Multiple update commands may be called to completely load the new secure code load. The upgrade complete contains a signature that was calculated by the manufacturer over the secure code image. Thus, as represented by block 1012, the TPM firmware verifies the signature calculated over the secure code using the RSA public verification key (Ksc\_init-pub).

[0209] Upon successful verification, the new secure code descriptor may be securely copied to flash. Thus, once the secure code load has been successfully loaded into the TPM, the new RSA public verification key (Ksc\_oem0-pub) may be loaded into the external flash (block 1014).

[0210] For a subsequent secure code load the signing environment uses the new private key to sign the secure code (block 1016). The secure signing environment then sends the new signed secure code to the TPM (block 1018). During the new secure code load, after the operating system TPM driver is loaded, the upgrade information request command may be used by TPM software to determine which Ksc key is used to load the next set of secure code information (block 1020). In addition, the TPM software may determine the TCG version of the current firmware. TPM software may thus determine whether a more up-to-date set of secure code information exists for the current RSA public verification key (Ksc\_oem0-pub).

[0211] The upgrade commands are used to load the secure code image to flash. The TPM firmware verifies the signature calculated over the secure code using the Ksc\_oem0-pub key (block 1022). Once verified, the new code may be loaded into the flash (1024).

[0212] Reiterating, when secure code including a new Ksc public key has been loaded into flash, the TPM uses the new Ksc public key that is stored in the secure code descriptor to verify any new signed code. Conversely, when new secure code has not been loaded into flash, the TPM uses the initial Ksc public key that is stored, for example, in on-chip ROM.

[0213] A system constructed in accordance with the invention may advantageously be configured to resist attacks from malicious persons (e.g., hackers) and/or programs (e.g., viruses).

[0214] In the event an attacker is able to read or write the external flash at will, the following techniques may be used to thwart potential attacks. First, by authenticating the data that is stored in the flash, the system may be able to detect attempts to modify the flash function table, flash read-only data or secure code block. On an authentication failure, the TPM may jump to an exception handler. Second, an attacker may attempt to replace the current flash function table with an older flash function table. This could cause the processor to jump to an unexpected address. An attacker could also replace the flash read-only data or a secure code block. Such attacks may be prevented by authenticating the firmware hash field. Third, an attacker may attempt to reorder chunks