

[0185] After the manufacturer receives an indication from the TPM acknowledging the start command, the manufacturer sends one or more upgrade update commands to the TPM. This command writes secure code into the external flash. In some embodiments this command may only be called after an upgrade start command or a previous upgrade update command. Owner authorization may not be required. However, owner authorization may be used to establish a transport session. Transport sessions may be used to encrypt the secure code information. In some embodiments, to keep the amount of data sent with each command relatively small, multiple upgrade update commands may be used to transfer the secure code information.

[0186] The upgrade update command may include, for example, the total number of input bytes, the size of the secure code information, the maximum size from the upgrade information request, the secure code information (e.g., secure code, rodata, etc.), the authorization handle used for owner authorization, and the authorization digest for the returned parameters.

[0187] As represented by block 814, once the TPM receives this command, it may verify various information and/or conditions associated with the command. For example, the TPM may verify that the previous command was an upgrade start or upgrade update command and that the previous command completed successfully. In addition, if authorization is required the TPM may use the authorization protocol to verify that the command was sent by the owner. The TPM also may verify that the number of bytes received is not greater than that specified in the image size field of the new secure code descriptor (block 816).

[0188] In some embodiments the secure code information may be encrypted as it is loaded into the TPM. Encryption may be desirable to prevent an attacker from gaining knowledge of the firmware functionality. TPM 1.2 transport sessions may be used to encrypt the code as it is transmitted on an internal bus (e.g., an LPC bus). Additionally, software applications could encrypt the secure code when it is delivered to a user's personal computer (PC). In these embodiments, the TPM may decrypt the received secure code information before it is processed for storage into the flash memory. Here, the TPM may use keys that were provided by an entity that created the keys used for encryption. These keys may be securely stored by the TPM, for example, as discussed herein.

[0189] At block 818, the TPM incorporates the blocks of received data into a digest of the existing SHA-1 thread. In some embodiments only integral numbers of complete blocks (e.g., 256 bytes each) may be processed. In this case the secure code size should be modulo 256 bytes.

[0190] In some embodiments the TPM may perform encryption and/or authentication operations on any information (e.g., secure code, rodata, function table, etc.) that is to be stored in the flash memory. In this way, the information may be protected (e.g., encrypted) when it is stored external to the chip. In addition, through the use of authentication, the TPM may verify that the information it retrieves from external memory has not been tampered with or replaced by unauthorized persons or programs.

[0191] Accordingly, as represented by block 820 the TPM may encrypt each secure code block (e.g., 256 bytes) and

generate an authentication digest for each secure code block using the flash 3DES and HMAC-SHA1 keys, respectively. As discussed above, FIG. 9 illustrates one embodiment of how each block may be encrypted and authenticated. For the first block, the IV may be generated by the random number generator. For subsequent blocks, the IV may be the last 8 bytes of the previous encrypted block.

[0192] As represented by block 822, the TPM writes each encrypted block and its corresponding authorization digest (e.g., 16 bytes) to the flash memory. The TPM may then increment a counter that keeps track of the number of secure code blocks. In the event subsequent upgrade update commands are invoked, the TPM may repeat the operations associated with blocks 814-822

[0193] To complete the upgrade process the manufacturer sends an update complete command. This command causes the TPM to write secure code into the external flash and verify the SHA1 digest that was calculated over the new secure code descriptor and the secure code blocks against the signature field. Upon successful verification, the new secure code descriptor may be securely written to flash. Owner authorization may not be required. However, owner authorization may be used to establish a transport session which may be used to encrypt the secure code information.

[0194] The upgrade complete command may include, for example, the total number of input bytes, the size of the secure code information, the maximum size from the upgrade information request, the secure code information (e.g., secure code, rodata, etc.), a signature (e.g., RSASSA-PKCS1-v1.5 2048 bit) over the transmitted secure code information, the authorization handle used for owner authorization, and the authorization digest for the returned parameters.

[0195] As represented by block 824, once the TPM receives this command it may verify various aspects of the command. For example, the TPM may perform operations similar to those discussed above in conjunction with blocks 814 and 816.

[0196] If the update complete command includes secure code information, the TPM may perform operations similar to those discussed above in conjunction with blocks 818-822. For example, the TPM may incorporate the received blocks into the verification digest, perform encryption and authentication operations on the blocks, store the encrypted code blocks and authentication digests into the flash memory and increment the counter that keeps track of the number of secure code blocks.

[0197] As represented by block 826, the TPM completes the SHA-1 digest. The TPM verifies the signature by calculating the signature over the SHA-1 digest.

[0198] At block 828, upon successful verification of the new secure code information, the TPM writes the new secure code descriptor to the flash memory. Here, the TPM may set the firmware loaded and descriptor valid fields of the new secure code descriptor. The TPM may then clear the firmware loaded field of the old secure code descriptor stored in flash. As discussed herein, the secure code descriptor may encrypt the descriptor and/or calculate an authentication digest for the descriptor before it is stored in the flash memory.