

[0057] Various types of information may be stored in the flash memory 306 to support the secure code update capability of the TPM 302. A description of one embodiment of such information follows.

[0058] A secure code structure 336 may contain encrypted executable code. This code may include, for example, any modified and/or new functions for the TPM 302.

[0059] A secure code descriptor structure 338 may describe the secure code information stored in the flash memory 306. For example, the secure code descriptor may describe the size of the secure code information and include keys that are used for decrypting or verifying information to be stored in the flash memory 306. In some embodiments the secure code descriptor structure 338 may be encrypted and/or authenticated using functions and/or key(s) stored in the OTP memory 332.

[0060] A function table 340 may contain a pointer to (e.g., an address of) every function that resides in on-chip ROM 310 and external flash 306. The function table 340 may be updated each time new secure code is written to the flash memory 306. In some embodiments the flash function table 340 may be both encrypted and authenticated by keys that are stored in the OTP memory 332. A flash function table 340 may be defined even when there is no code stored in the flash. In some embodiments the function table structure 340 may be read by the processor 308 via the instruction cache controller 326.

[0061] A read-only data (“rodata”) segment 342 contains data for functions that are stored in the flash memory 306. For example, the flash rodata table 342 may contain constants used by functions such as cryptographic routines, self-test routines, and functions with switches. The flash rodata segment 342 may be updated each time new secure code information is stored in the flash memory 306. The rodata segment 342 may be encrypted and authenticated. In some embodiments the rodata structure 342 may be read by the processor 308 via the instruction cache controller 326.

[0062] In some embodiments the flash function table 340, rodata 342, and secure code 336 may be authenticated in, for example, fixed-sized blocks (e.g., 256 bytes) using HMAC-SHA1. For each encrypted block, the resulting HMAC may be stored in a secure code authentication region 344.

[0063] In some embodiments the flash 306 includes data used by the he system after the secure code has been loaded and the system is operating in a secure mode. For example, the flash 306 may store keys and data 356 and non-volatile RAM data for a TPM.

[0064] In some embodiments the flash memory 306 is hard partitioned. For example, a hard partition may be defined for the flash memory 306 between code used by the TPM (e.g., the secure code information discussed above) and code used by one or more other devices, components or processes. In this case, the flash may contain components 336, 338, 340, 342 and 344 used for secure code load operations. In addition, the flash may contain other firmware 346 and a flash directory 348 that is used by the other devices/components/processes. For example, the flash memory may store code and data 346 that is used by another processing component 222 (e.g., a local area network device) as shown in FIG. 2.

[0065] With the above component configuration in mind, a brief overview of one embodiment of data flow in the system of FIG. 3 will now be provided. In particular, information flow relating to writes to and reads from the flash memory 306 and reads from the instruction ROM 310 will be discussed.

[0066] The processor 302 writes data to flash memory 306 through the external interface module (“EIM”) 322. For example, a command may be invoked to move information from the LPC bus 324 to the data buffer 318. The cryptographic processor(s) 330 may then verify a signature over the received information. Next, the cryptographic processor(s) 330 may encrypt the information and generate an authentication digest for the information. Finally, the encrypted information is sent to the flash memory 306 via the external interface module 322 and the flash controller 304.

[0067] In some embodiments the TPM may access the external flash through three separate mechanisms. These mechanisms may be used to perform general data fetches, instruction fetches, and function table and rodata fetches.

[0068] The processor 308 (e.g., TPM firmware executing on the processor) reads data stored in the external flash through the external interface module 322 that is attached to the data bus 316. For example, TPM non-volatile data stored in a TPM non-volatile RAM portion 358 of the flash memory 306 may be read by the processor 308 in this manner. This information may be requested, for example, via a load-word (“lw”) instruction. The external interface module 322 interfaces with the flash controller 304 to retrieve the requested data.

[0069] The processor accesses executable code through the instruction multiplexer 312. The processor 308 issues an instruction fetch over the instruction bus 314. The instruction multiplexer 312 determines whether the requested code is stored in the instruction ROM 310 or the flash memory 306. After the instruction multiplexer 312 retrieves the code from the appropriate data memory, the instruction multiplexer 312 returns the code to the processor 308 via the instruction bus 314.

[0070] The processor (e.g., the TPM firmware) issues reads for the flash function table or the flash rodata segment on the data bus 316. These reads are serviced by the instruction multiplexer 312. After decoding a function table or rodata segment read access, the instruction multiplexer 312 may issue a request to the instruction cache controller 326 to retrieve the requested information. The instruction multiplexer 312 then returns the information to the processor via the data bus 316.

[0071] There are two consumers of the instruction cache controller 326: a processor instruction fetch and a flash function table (or rodata) access. If the instruction cache controller 326 is busy responding to a cache miss, the other request may be registered by the instruction multiplexer 312 until the original cache miss has been serviced. When the instruction multiplexer 312 simultaneously receives a flash instruction fetch and a flash function table access, the processor 308 may be stalled until both requests have been serviced. An instruction ROM fetch and flash function table request may be serviced simultaneously by the instruction multiplexer 312.