

[0139] Alternatively, as represented by block 606, if the requested function table is stored in the external flash memory, the instruction multiplexer requests the information from the instruction cache controller. If at block 608 the function table is in the instruction cache (a cache “hit”), the instruction multiplexer returns the function table to the instruction multiplexer. The instruction multiplexer then returns the function table to the processor via the data bus (block 612).

[0140] As represented by block 610, if the function table is not in the instruction cache (a cache “miss”), the instruction multiplexer stalls the processor while the instruction cache controller retrieves the function table from the flash memory. The instruction cache controller issues a request to the flash controller to retrieve the block or blocks of data from the flash memory that contain the external function table.

[0141] Upon receipt of the requested block(s) the cryptographic processor(s) 334 use the instruction cache keys (e.g., 3DES and HMAC-SHA1 keys) to decrypt and authenticate the block(s) of data. In some embodiments, if authentication fails the TPM is reset and enters failure mode. The TPM may exit this failure mode after a reset (e.g., after an LPC reset or a POR reset)

[0142] If the authentication is successful, the instruction cache controller stores the block(s) of data in the instruction cache and the cache line is marked as valid. In addition, the instruction cache controller returns the function table to the instruction multiplexer. The instruction multiplexer then terminates the processor stall.

[0143] As represented by block 612, once the instruction multiplexer receives the table, it returns the function table to the processor via the data bus.

[0144] To initiate a function call, the processor sends the instruction address to the instruction multiplexer via the instruction bus (block 614). The instruction multiplexer either reads the internal ROM or sends a request to the instruction cache controller depending on the address associated with the function (block 616). As discussed above, this address may be obtained from the appropriate function table. Advantageously, the instruction multiplexer may access the instruction ROM or the instruction cache without any involvement from TPM firmware.

[0145] As represented by block 618, when the address is an internal address, the instruction ROM returns the requested instruction in the next cycle. In addition, in some embodiments the instruction ROM also returns security assurance logic (“SAL”) bits. The SAL bits may be used to prevent an attacker from jumping the program counter of the processor without being detected.

[0146] As represented by block 620, if the requested code is stored in the external flash memory, the instruction multiplexer requests the information from the instruction cache controller. If at block 622 the code is in the instruction cache (a cache “hit”), the instruction cache controller returns the code to the instruction multiplexer in the next cycle.

[0147] In some embodiments the program counter assurance logic may be disabled when operating out of the instruction cache. For example, the SAL bits may be set high when returning data from the instruction cache.

[0148] As represented by block 624, if the code is not in the instruction cache (a cache “miss”), the instruction multiplexer stalls the processor while the instruction cache controller retrieves the code from the flash memory. The instruction cache controller issues a request to the flash controller to retrieve the block or blocks of data from the flash memory that contain the external code.

[0149] Upon receipt of the requested block(s) the cryptographic processor(s) 334 use the instruction cache keys (e.g., 3DES and HMAC-SHA1 keys) to decrypt and authenticate the block(s) of data. If the authentication is successful, the instruction cache controller stores the block(s) of data in the instruction cache and the cache line is marked as valid. The instruction cache controller returns the requested code to the instruction multiplexer and the instruction multiplexer terminates the processor stall.

[0150] In some embodiments if authentication fails, the instruction cache may raise an error signal. When this signal is raised, the TPM is reset. After reset, the boot routine generates an error response for any interrupted command and the TPM enters failure mode. The exception handler places the TPM in a failure mode until the next reboot. In this mode, secure code stored in external flash may not be accessed.

[0151] As represented by block 626, after the instruction multiplexer receives the requested code, the instruction multiplexer returns the code to the processor via the instruction bus. The processor is thus able to execute the instruction associated with the appropriate (e.g., original, modified or new) function.

[0152] FIG. 9 illustrates one embodiment of how data may be decrypted and authenticated in the system. As discussed herein, the secure code may be encrypted and an authentication digest over the secure code may be generated before the secure code is stored in external memory. The block 902 on the left side of FIG. 9 represents the secure code and authentication information that is stored in the flash memory. The block 904 on the right side of FIG. 9 illustrates the various parameters that are used to decrypt and authenticate the secure code.

[0153] The secure code 906 and the code authentication 908 may be stored as represented in FIG. 3 in the secure code location 336 and in the secure code authentication location 344, respectively, in the external flash 306. In some embodiments the secure code 906 comprises a set of secure code blocks (e.g., blocks 0, 1, 2, etc.) and the code authentication code comprises a set of corresponding code authentication blocks (e.g., blocks 0, 1, 2, etc.). As shown in FIG. 9, the secure code blocks and their corresponding HMACs each may be stored contiguously in separate memory blocks in memory.

[0154] In the example of FIG. 9 an initialization vector (“IV”) 926 begins 8 bytes before the start of the secure code block. In some embodiments the IV for the first code block is stored in the flash memory. This IV may be generated by a random number generator (not shown) that may be one of the secure code components. The IV for subsequent code blocks may be, for example, the last 8 bytes of the previous encrypted block. This association is represented by the line 912 in FIG. 9.