

may be implemented in end-user client devices such as cell phones, laptops, etc., that need some form of data security, authentication, etc. In some embodiments the security module may be integrated into previously existing chips (e.g., a main processor) within these devices.

[0173] The security module may be configured as part of and to enforce a security boundary. For example, the security module may be configured to never allow clear text keys to exit, for example, the security module or the chip within which the security module is implemented. As a result, the security module may be safely integrated into other devices or systems regardless of whether the system outside of the security boundary is secure.

[0174] In this way, the security module may provide highly secure and cost effective remote key management for a client device. The security module may provide and/or support any required cryptographic processing. A security boundary is established within the device to securely maintain and use keys and key material. Yet the system may be securely managed by a remote key management system (e.g., a hardware security module, a TPM, etc.) via the security module. Accordingly, a high level of security functionality may be provided for the end-user device using a relatively small security module that has minimal impact on the rest of the device.

[0175] To support this key usage and management scheme, a security module provides mechanisms for securely loading one or more keys into the module, securely storing the keys and securely using the keys. One embodiment of a stateless hardware security module 1100 that provides such mechanisms is depicted in FIG. 11.

[0176] The stateless module 1100 includes a master controller 1106 for controlling the overall operation of the module. For example, the controller may control boot operations, key management operations (if applicable) and data and key traffic flow into and out of the module. The controller may comprise, for example, a processor and associated code (e.g., ROM 1108) and/or a state machine or other hardware. The controller and/or any other component in the stateless module may communicate with other components in the stateless module via an internal bus 1130.

[0177] In some embodiments the master controller 1106 comprises a RISC processor with ROM code to execute the various commands necessary for the operation of the stateless module. The master controller block also may include the address decoder for each of the slave blocks on the internal bus 1130. The RISC engine may use a protected portion of a data buffer 1126 for temporary stack and scratch data space.

[0178] A bi-directional external interface 1120 provides a mechanism to send keys and/or data to or receive keys and/or data from the module. For example, the external interface may include registers that may be written to or read by the controller and external devices (e.g., a host) that are connected to the stateless module. In this case, the controller may be configured so that it never writes certain data (e.g., unencrypted keys) to the registers.

[0179] The external data interface 1120 may be used by a local host to read global registers, issue commands and place data into the data buffer 1126 for processing by the stateless module. The external interface may be controlled through a

global register block 1122 by the master controller. These global registers may include, for example, command (“CMD”), timer and configuration (“CONFIG.”) registers. The master controller transfers the data between the global registers block and a data buffer memory 1126.

[0180] The command interface provides a streaming data interface directly into the data input and data output registers. It allows an external FIFO to be used for data input and data output (separate FIFOs). This interface allows the stateless module to be easily embedded into a packet based system.

[0181] In some embodiments, data (e.g., data to be processed or key material) to be encrypted or decrypted may be sent to or sent from the stateless module 1100 via one or more data interfaces. For example a data interface 1102 may be used to send encrypted data or keys (e.g., that were decrypted by the module) to a cryptographic accelerator and vice versa. In addition, a data interface may be connected to an input device (e.g., a sensor) that generates data that needs to be encrypted by the stateless module. This encrypted data may then be sent to an external processing component via the external interface 1120.

[0182] One or more cryptographic processing blocks perform any cryptographic processing that needs to be done to acquire or use keys or to cryptographically process data flowing through the module. For example, separate processing blocks may be used to perform asymmetric key algorithms such as DSA, RSA Diffie-Hellman (block 1114), key exchange protocols or symmetric key algorithms such as 3DES, AES (block 1112) or authentication algorithms such as HMAC-SHA1 (block 1110). The cryptographic processing block may be implemented, for example, in hardware and/or using a processor that executes code stored in a data memory (e.g., ROM).

[0183] Typically this embodiment includes processing to generate asymmetric keys that are used to establish a secure channel with a remote device and to authenticate information sent from the module to the remote device and vice versa. Here, the private portion of the asymmetric key may be maintained within the security boundary of the chip. In addition, the stateless module also will include a mechanism for exporting the public version of the asymmetric key. For example, the public value may be loaded into the external interface register discussed above so that it may then be read by an external device. The public key value may be read from the stateless module by issuing a public key read command to the stateless module. In response to this command the module returns the public key value and any non-secure configuration information for the device (authorization data, product configuration data, etc.).

[0184] In some embodiments a root, identity key serves as the basis for the asymmetric key. For example, the root key for the module may comprise an asymmetric key pair (secret or private, public) that is used to uniquely identify the stateless module. In some embodiments this key is only used for digital signatures to securely identify the stateless module.

[0185] In some embodiments, one or more keys (e.g., the root, identity key for the module) may be injected into the stateless module. This may be performed, for example, when the chip is manufactured, when the chip is tested, during