

manufacture at an OEM (e.g., circuit board manufacturer), during OEM testing or during installation for the end user. This technique may be used to inject symmetric and/or asymmetric keys.

[0186] In some embodiments, the stateless module may generate one or more keys (e.g., the root, identity key) internally. For example, the stateless module may include a random number generator (“RNG”) 1118 and other circuitry necessary to generate a key. This embodiment may provide added security in that the generated key may never leave the security boundary of the chip.

[0187] In some embodiments the device identity key comprises a collection of random bits that are used to generate the key material for the long term fixed keys in the stateless module. For example, the RNG 1118 may generate a random number using the internal random number value as a secret initialization seed. The number of bits in the initialization seed may be determined by the amount of key entropy required for the system.

[0188] In some embodiments the value from the random number generator 1118 may not be used directly. For example, it may be post processed using the SHA-1 block 1110 by the master controller before internal usage and before exposing the number external to the stateless module as a random value. The master controller may maintain a cache of post processed random bits (for key generation and for signing) in the data buffer 1126.

[0189] The random number generator 1118 may be a “true” random source. For example, it may utilize free running oscillators to capture thermal noise as the source of randomness.

[0190] The stateless module also may include a privacy (or confidentiality) asymmetric key pair that may be used for transferring secure content to the stateless module device via an intermediate insecure third party such that the third party does not have access to the key material. In some embodiments the confidentiality key is only used to decrypt key material within the stateless module.

[0191] The above keys (e.g., the root, identity key, etc.) may be stored in a nonvolatile data memory (“NVM”) 1116. The NVM may comprise, for example, a one-time programmable (“OTP”) memory or battery backed memory (BBMEM) that is located on-chip or off-chip.

[0192] In some embodiments an on-chip OTP memory (as shown in FIG. 11) may provide certain advantages. For example, in this case the keys may be physically protected within the device so that they cannot be easily altered or observed. In addition, since the use of the keys may be confined within the chip, the keys may not appear in the clear outside of the chip. Moreover, this OTP and stateless module combination may then be implemented using a standard CMOS process. As a result, the stateless module may be readily integrated into a variety of conventional chips that are used in end-user and other devices. Such a combination may provide a very cost effective security solution.

[0193] Examples of architectures and implementations of OTP memory that may be advantageously implemented in CMOS are described in, for example, U.S. Pat. Nos. 6,525,955, 6,693,819, 6,700,176 and 6,704,236 and U.S. patent

application Ser. No. 09/739,952, filed Dec. 20, 2000, the disclosure of each of which is hereby incorporated by reference herein.

[0194] The OTP may be programmed by the master controller 1106 via a programming interface in conjunction with an external programming signal VPP. The master controller may ensure (via local hardware enforcement) that the device keys, authorization and configuration data can be programmed once and only once.

[0195] The key-encryption-key (“KEK”) cache 1124 is a separate memory block sized based on the required number of KEKs in the system. Typically, it is large enough to hold the session private key and a single asymmetric group key.

[0196] The KEK Cache 1124 may be protected in hardware during the execution of any command that does not require a KEK key. For example, a signal from the global registers may be provided to the KEK cache to indicate that the command register is locked, active and contains a command that requires a KEK. Some KEK cache locations are contained in the NVM block that is used to implement the long term keys for the stateless module.

[0197] The application key cache 1104 may be used by the master controller to provide encryption and decryption storage for the internal acceleration cores (such as the public key core 1114 or the 3DES core 1112). The application key cache may enforce key lifetime expiration when the keys are used by either the stateless module commands or the application key cache interface.

[0198] In general, the performance, size and function of the blocks discussed above may be scaled to meet the demands of the system. For example, the basic cryptographic functions that implement the secure channel back to the key manager to transfer and process key material (and/or policy) may be provided at minimal processing performance levels.

[0199] The cryptographic accelerators contained within the stateless module can be used for application data processing when they are not being used for key management functions. For example, a stateless module for an e-commerce application may be used to protect RSA private keys. Here, the public key acceleration required for the secure channel is typically minimal (less than 10 operations/sec). Consequently, any spare processing capacity (e.g., idle cycles of a processor) may be used for other operations.

[0200] In contrast, public key acceleration required for a typical e-commerce accelerator is relatively high (greater than 500 operations/sec). Applications such as this may require the use of cryptographic accelerators that are specially designed to perform cryptographic operations at a high rate of speed.

[0201] One or more cryptographic accelerators may be attached directly to the stateless module via the application key cache interface 1102. Typically, the application key cache interface for the add-on cryptographic acceleration processing is maintained within the security boundary. For example, the stateless module and the cryptographic accelerators may be implemented on the same chip. In this manner, the cleartext keys are not allowed to leave the security boundary which also includes the public key accelerator. However, the external application may use the public