

1118 to generate a random number that is provided as a seed to a cryptographic processor that generates a public-private key pair.

[**0217**] The master controller stores the private (identity) key in the nonvolatile memory **1116** and never exports this key outside of the security boundary of the module (block **1204**). For example, in some embodiments the key never leaves the chip within which the stateless module resides. In some embodiments this key is encrypted before being stored in off-chip non-volatile memory.

[**0218**] The stateless module also stores the corresponding public key and, upon request, exports the public key (block **1206**) so that the device manufacturer (or some other trusted entity) may publish the public key along with a certificate to a public server.

[**0219**] The stateless module may then be deployed in a computing device that can connect to another device (e.g., a key manager) via a network or some other link. As represented by block **1208**, the stateless module may use its private key to establish a secure communication channel with, for example, a security module (e.g., a key manager) that has access to the stateless module's public key.

[**0220**] As represented by block **1210** the key manager may send keys to the stateless module via the secure communication channel. For example, the key manager and stateless module may negotiate to obtain additional keys that may be used to provide secure communications between the two components. In addition, the key manager may send keys to a remote client via the stateless module. For example, the key manager may generate a private session key (Ka-priv) for a client that incorporates the stateless module. As discussed above, the key manager may encrypt this key using the stateless module's public key (Kdc-pub) or some negotiated key before sending it to the client.

[**0221**] As represented by block **1212**, the keys are decrypted within the security boundary associated with the stateless module. For example, cryptographic processors in the stateless module may decrypt these keys. Alternatively, another cryptographic processor located on the same chip as the stateless module may decrypt the keys.

[**0222**] As represented by block **1214**, the stateless module may then use the keys within the security boundary. For example, cryptographic processors in the stateless module may use these keys to decrypt other keys (e.g., session keys). In addition, the stateless module may enforce key policy within the security boundary (block **1216**).

[**0223**] In some embodiments, as represented by block **1218**, the stateless module may provide keys to one or more cryptographic accelerators within the security boundary. For example, the cryptographic accelerators may be located on the same chip as the stateless module.

[**0224**] Referring now to **FIG. 13**, one embodiment of a stateless secure link module **1300** will be discussed in detail. This embodiment includes, in general, a subset of the functionality of the embodiment of **FIG. 11**. In particular, this embodiment only provides data encryption, decryption, etc. using a symmetric key. One advantage of this configuration is that it may be implemented in other devices with even less impact on the cost and the size of the devices.

[**0225**] In a typical application the embodiment of **FIG. 13** is used to take data that originates from an input device and securely provide that data to a recipient device that uses the data (e.g., an access device or an access server). This process may involve encrypting the data so it does not appear in clear text and/or signing the data to certify to the recipient device that the data originated from a specific input device.

[**0226**] For example, the stateless module may be integrated into a chip for a sensor (e.g., a biometric sensor such as a fingerprint reader). Here, the stateless module may be used to sign and/or encrypt the information generated by the sensor. The stateless module may then securely send the information to a recipient device that uses the information. In this case, the recipient device may use a fingerprint comparison as a means to control access to data or a service.

[**0227**] In some embodiments the sensor data is always maintained within a security boundary. First, by incorporating the stateless module into the sensor chip, the information may be encrypted before it leaves the hardware boundary of the chip. Second, the stateless module may establish a secure channel with the recipient device through a symmetric key exchange. In this way, the information may be securely sent to the recipient device. Third, the recipient device may be secured in a conventional manner or using techniques as described herein.

[**0228**] As an example of the latter scenario, the recipient device may include a stateless module as described above in conjunction with **FIG. 11**. In this case, the recipient device may use other keys to, for example, securely send the information to a remote system. One example of such a remote system is a network access device that enables access to a network based on the user's credentials such as the user's fingerprint.

[**0229**] In other embodiments, it may only be necessary to establish that the data originated from a specific input device. For example, the system may make other provisions to ensure that a copied fingerprint data stream is not being replayed at a later time. In this case, it may be unnecessary to encrypt the information. All that may be needed here is an assurance that the information is being sent by a specific sensor. In this case, adequate security may be provided by simply signing the data.

[**0230**] To provide a solution that is cost effective for a variety of input devices, the stateless module of **FIG. 13** has a reduced set of functionality as compared to, for example, the embodiment of **FIG. 11**. The stateless module includes a master controller **1306** and an external interface **1312** to enable the asymmetric key operations that are performed when the secure link is initially established with, for example, a key manager. Thus, the controller **1306** includes circuitry to generate and verify the validity of its keys. In addition, the module may include assurance logic **1320** similar to that discussed above.

[**0231**] However, because the module only uses a single symmetric key, much of the functionality depicted in **FIG. 11** is not provided in the embodiment of **FIG. 13**. For example, the module does not need to provide management capabilities (e.g., enforcement of key policy) and data storage (e.g., application key cache) for extra keys. Also, the non-volatile ROM ("NVROM") **1310** may be smaller since it may only store, for example, an identity key and a symmetric key.