ARTICLE

Forward Secrecy Made Real Easy

February 15, 2016

By Dan Ujvari, Senior Field Applications Engineer

Forward Secrecy (FS), often referred to as Perfect Forward Secrecy (PFS), is essentially the protection of ciphertext with respect to time and changes in security of your primary cryptographic keying material over time.

A cryptographic session key is used to authenticate messages and encrypt text into ciphertext before it is transmitted. Confidentiality of this session key thwarts a “man in the middle” from understanding the message and/or altering that message. These keys are derived from primary keying material. In the case of Public Key Cryptography this primary key would be the private key.

Unless you are implementing your own security in the application layer, you probably rely on the TLS/SSL in the transport layer.

The Problem

One can envision a scenario in which ciphertext was recorded by an eavesdropper over time. For a variety of reasons out of your control, the primary keying material (private key) or a current session key may be discovered and using this, the eavesdropper could potentially decipher all of those previously recorded transmissions.

Release of your private keys could be the result of a deliberate act, as with a bribe, a disgruntled employee, or even someone thinking they are “doing the right thing” by exposing your secrets, or it could be the result of an unwitting transgression, aka bug, from your security software. Equipment could be decommissioned and disposed of improperly. The hard drives could be recovered using the infamous *dumpster dive attack methodology* thereby exposing your secrets.

Recently leaked NSA documents shows powerful government agencies can, *and do*, record ciphertext and store that data for extended periods of time.

You have to worry not only about your own system, but the system with which you are communicating. Your security could be challenged knowingly or unknowingly by third parties controlling servers on their end since they of course need to be able to decrypt your messages under normal operation. There are many ways your forward security could be compromised at the server level. Server managers can unwittingly compromise it with bad practices; inadequate cipher suites, leaving session keys on the server too long, use of resumption mechanisms, and so on.

Let’s just say there are many, many ways the security of your session keys and/or primary keying material could eventually be compromised. It only takes one of them. The damage is irreversible and the result is the same: Those recorded ciphertext transmissions are now open to entities they should not be open to.

The Solution

A public/private key system demonstrates the property of forward secrecy if it creates new key pairs for each communication session. These key pairs are generated on an as-needed basis and are destroyed after the session is over. Their generation must be truly random. They cannot be the result of a deterministic algorithm. Once a session key is derived from the public/private key pair, that key pair must not be reused.

The Atmel® ATECC508A CryptoAuthentication™ device meets these criteria. It has the ability to generate new key pairs using a high quality truly random number generator. The ATECC508A supports ECDH, a method to spawn a cryptographic session key by knowing the public key of the recipient. When these spawned session keys are purposely short-lived, or *ephemeral*, the process is known as ECDHE.

Using this method, each communications session has its own unique keying material. Any compromise of this material only compromises that one transmission. The secrecy of all other transmissions remains secure.

Transport layer security such as SSL/TLS may support ephemeral key generation but you may or may not have control over the actions of the standard security layers in the overall system. So while we always recommend you employ transport layer security, you can wipe out much of your liability by simply changing where encryption takes place. If encryption and forward secrecy are addressed in the application layer on both sides, you can sidestep liabilities associated with the security layers on both sides that may be out of your control.

This, of course, does not imply transport layer security should be discarded.

The Need for Robust Authentication in the First Place

Before any of the above can occur, the identity of the correspondents needs to be robustly authenticated, a process usually done with public key signatures. Their identities need to be assured without doubt (non-repudiation) because accepting an unknown public key without robust authentication of origin could authorize an attacker as a valid user. The ATECC508A provides this required level of authentication and non-repudiation.

The ATECC508A is a cost effective asymmetric authentication device available in a tiny package, is easy to design in, and is ultra-secure. Offering secure hardware key storage on-board, it also has an on-board ECC cryptographic block for ECDSA and ECDH(E), a high quality random number generator, a monotonic counter, and unique serial number.

All Atmel CryptoAuthentication devices have active defenses, such as an active shield protecting the entire device, voltage/temperature/clock tamper monitors, and active defenses against power analysis attacks on the dynamic current consumption of the chip. In addition, it also offers an external tamper pin that can be connected to the case of your system, so unauthorized opening of your product can be detected.

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**