

SSH Key Management Security

What it is

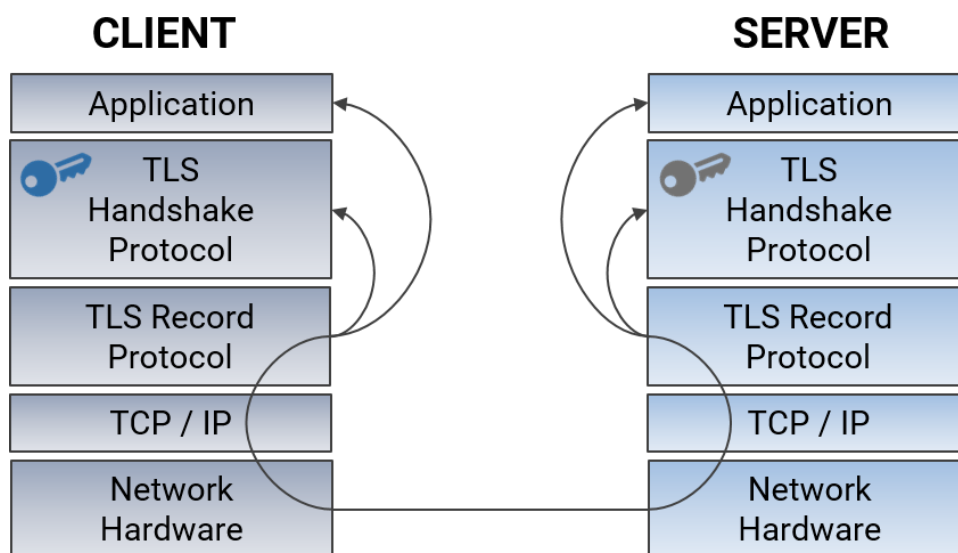
SSH (Secure Shell) is a technology that allows an entity on one system to securely and legitimately access the command shell of a remote system.

The system is implemented on top of “TLS” (Transport-Layer Security), itself implemented on top of the TCP/IP Ethernet protocol stack.

TCP sets up a connection-based tunnel between two IP addresses, bound to arbitrary “ports”. The TCP-based tunnel is actually a series of hops between intermediary systems until the remote end-point is reached. Each of the intermediary systems has full control of the data that it is receiving and then forwarding. Or, receiving, inspecting, capturing, compromising, altering, etc. and only then forwarding.

In order to prevent the intermediaries (the “man in the middle”) from being able to make easy use of the data they are relaying, the data should be protected in such a way that only the two primary endpoints are capable of accessing fungible, in the clear, data.

Application-layer security is sometimes seen as too cumbersome to implement, even when it is not difficult, or that it adds too much overhead (bandwidth, time). Most enterprises fall back on secure transmission at the transport layer (“Transport Layer Security”, or TLS), making it the most implemented method for protection of their data while it is in transit.



How does TLS work

Every application or user that will rely on TLS will be issued an asymmetric key (RSA, EC). The private part (generally stored in a user’s home directory in a hidden folder, accessible only to the logged-in user) is maintained locally, while the public part is exchanged with the remote system of interest. Likewise, the remote system will also have an asymmetric key, and will furnish its public key on request.

Each side of the communication will generate a random nonce, and encrypt it with the other's public key. The encrypted nonce is sent, and on receipt decrypted using the respective private key. The two nonces are combined algorithmically to result in a single shared (symmetric) secret key, which is then used to encrypt/decrypt all further traffic on that channel.

The shared secret key is used, rather than simply using the peer's public key to encrypt everything destined for it, because symmetric cryptology is significantly faster and less CPU intensive than asymmetric cryptology.

What are the risks

If one side or the other allows their asymmetric key to be compromised or extracted, a "man in the middle" (MiM) can change the public key exchange such that the two endpoints receive the MiM's public key instead. The exchange of nonces happens as normal, except the nonce received by each endpoint has been generated by the MiM instead of the remote endpoint. The MiM sets up a secure connection between itself and each endpoint, inserting itself into the channel without either side knowing and can decrypt, inspect and re-encrypt all the data as it moves across the channel. Application layer, or "end to end" security goes a long way to making a MiM attack unproductive.

When only TLS is used, protecting the SSH Keys of a system becomes a serious issue, and since they are mainly stored in "Access Control List" protected directories, the compromise of a user account with the correct roles can lead to the exposure of the SSL key pairs used for communication.

SSH Key Management

Protecting the suite of SSH Keys that a company has is best done through either a stand-alone SSH Key management application or by using tools built on top of the Oasis open standard "Key Management Interoperability Protocol" (KMIP).

Protection of the SSH keys used to protect TLS channels is paramount to information security.

In order to provide the highest level of protection, some SSH Key managers and KMIP server tools store their system keys within a NIST FIPS 140-2 certified Hardware Security Module (HSM), such as the Utimaco CryptoServer.

Hardware Security Modules are designed to securely store cryptographic keys and other secrets, by

- Allowing for four-eyes (or greater!) requirements for use of the secrets contained,
- Providing policies around (not) allowing the export of secrets (if not permitted, the keys must be used in situ -- input is sent to the HSM, which uses the key and returns just the results. The key never leaves the protection boundary), and
- Providing for physical protection of the secrets (in the event of physical attack, well-designed HSMs will erase the key material, preventing its compromise).

Use of an HSM-protected primary system key should also be protected by policies and procedures, implemented around the HSMs own native user management scheme.

In addition to using the HSM to store the Key manager's system key, the enterprise might also set up the HSM to do the actual SSH key exchange/secret key agreement steps internally. By using the key entirely within the HSM itself, it protects and prevents exposure of the secret key to outside observation and compromise.

The key management client requests a key from the manager, which it receives 'wrapped' (encrypted using a key encryption key stored on the target HSM). The client loads the

wrapped key into the HSM where it is unwrapped and prepared for usage. The key agreement step then is handed off to the HSM, and the resulting agreed-upon secret key remains within so that all data received can be decrypted only by the HSM, without exposing the agreed secret key to any third party.

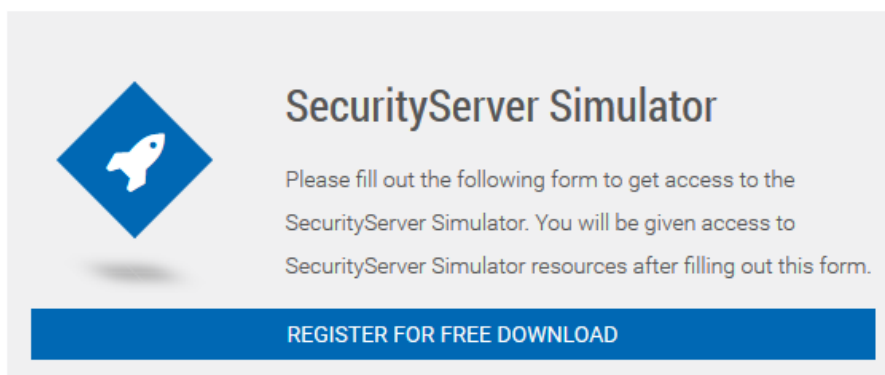
About Utimaco

Utimaco is a leading manufacturer of hardware security modules (HSMs) that provide the Root of Trust to the IoT. We keep cryptographic keys and digital identities safe to protect critical digital infrastructures and high value data assets. Our products enable innovations and support the creation of new business by helping to secure critical business data and transactions. Founded in 1983, Utimaco HSMs today are deployed across more than 80 countries in more than 1,000 installations. Utimaco employs a total of 170 people, with sales offices in Germany, the US, the UK and Singapore. For more information, visit hsm.utimaco.com



Download the Utimaco Software HSM Simulator to get started immediately learning about HSM devices. It is a FREE fully functioning Software version of the Hardware HSM. The download package includes documentation on our product. Here is the link to register for the download.

<https://support.hsm.utimaco.com/hsm-simulator>



The banner features a blue diamond icon with a white rocket ship on the left. To the right, the text reads: "SecurityServer Simulator", "Please fill out the following form to get access to the SecurityServer Simulator. You will be given access to SecurityServer Simulator resources after filling out this form.", and a prominent blue button with the text "REGISTER FOR FREE DOWNLOAD".

The Simulator download includes tools for creating user accounts, sample code and libraries for PKCS#11 Microsoft CNG, Java JCE and the Utimaco CXI API to link and test your code. The Simulator will run on a Windows or Linux platform. This is your opportunity to try out the Utimaco HSM technology easily and without initial cost.