Quick Guide to creating SSH Keys

Setting up the required SSH keys for secure remote console connections, secure TTE connections, and SFTP file transfers.

The SSH protocol relies on "asymmetric keys" technology, meaning two keys per side of the connection - one key to encrypt (public key), and one key to decrypt (private key). The public and private key pairs are mathematically related, and so are thus created at the same time. The public key is simply the SSH key that is distributed to trusted parties that you wish to encrypt messages sent to you. Multiple parties can have the same copy of your public key because it is only used for encryption, it cannot be used to decrypt the same messages it encrypts, the private key is the only key that can decrypt messages encrypted by the public key. You will keep the private key "private" in a safe location on your PC, preferably password protected too.
Since there are two sides to a connection (Client and Server), this means that there should be a minimum of two key pairs (four separate key files) that must be created to make a unique, secure connection using SSH.
However, complicating things slightly, there are two "versions" of keys in use, each version is derived by different algorithms, the DSA and RSA algorithms. The SSH server in the RTU requires Server keys generated from both algorithms for its internal use. The SSH server will not operate unless both sets of Server keys are loaded onto the RTU. While it is possible to use Client keys created from both algorithms, it is recommended to use only key pairs created with the RSA algorithm. Each user will require their own key pair.

Client Key pair (1 per user)	Server Key pairs (each version per RTU)
Private Key (RSA)	Private (RSA, DSA)
Public Key (RSA)	Public Key (RSA, DSA)

Table 1. Key pairs needed by "connection side".

All sage RTUs delivered with the J0 and later revision firmware, come pre-loaded with default SSH client and server keys. This makes the default configuration inherently insecure because all customers have the same public and private keys. To be truly secure, each customer must generate unique keys to prevent others from being able to decrypt any commands and data being carried through the protocol.

This tutorial will detail the steps required to create the new, unique keys for the SSH server and each SSH client (user) before the SSH protocol can be considered "secure".

See the "config@WEB Key & Certificate Generation" document for details on how to create the key pair files themselves.

Key Generation steps:

1) Generate unique key pairs for the RTU SSH Server, one key pair created with the RSA 2048-bit algorithm, and one key pair created with the DSA 2048-bit algorithm.

2) Name the RTU SSH server RSA and DSA key pairs as follows:
   RSA public key: RSA_PUB.KEY
   RSA private key: RSA_PVT.KEY

   DSA public key: DSA_PUB.KEY
   DSA private key: DSA_PVT.KEY

3) Generate a unique RSA 2048-bit key pair for each user being given SSH access to the RTU.

4) Name each RSA key pair as follows:
   RSA public key: "username".pk2, where username is the user's login id
   RSA private key: "username".ppk, where username is the user's login id

   Ex: For Admin user, rename public key to Admin.pk2, and private key to Admin.ppk. Spelling is case sensitive.

5) Use the User_Manager.exe utility to create a "Users" package that contains all the user login and password information, and the SSH client Public keys, and the SSH server (RTU) public and private key pairs.

6) Upload the "User's Package" to the RTU using the GUI Up/Download page.