1.openssh-client(ssh) openssh-server(sshd)

2.ssh user@ip/domain sftp user@ip/domain xshell xftp

3.negotiation procedure:

①server send *server's public key* to client

②server send *session ID* to client

③client send encrypted *session key* using *server's public key* to server

④server decode *session key* encrypted using *server's private key*

⑤client and server both have *session ID* and *session key*, then data transmission are encrypted and decoded by session ID and key

4.authentication-method: password keyboard_interactive public_key

5.ssh_key: ssh-keygen -t rsa -C "xx(email..)" passphrase

ssh_key_pair: A:~/.ssh/id_rsa (private_key) A:~/.ssh/id_rsa.pub (public_key)

B:~/.ssh/authorized_keys(600/644 public_key of A)

6.tunneling: port_forwarding X11_forwarding(Xming - opensource X server)

1.certmgr.msc

2.certificate is mainly consist of public_key and digital signature, the digital signature is pointed to thumbprint which was encrypted by CA's private key

root certificate is mainly consist of CA_public_key and digital signature, the digital signature is pointed to thumbprint which was encrypted by CA's private key

3.SSL procedure:

①client send a request of secure connection to server

②server send *server's digital certificate* to client

③client find the *CA's root digital certificate* which the server's digital certificate indicate in local certificate manager, then decode the encrypted thumbprint using CA_public_key which the root digital certificate include, then hash the content of server's digital certificate include server's public_key using thumbprint_algorithm, finally to certify if the *server's public_key* provided by server's digital certificate is trustable through compare hash value to thumbprint

④client send encrypted *session key* using *server's public_key* to server

④server decode *session key* encrypted using *server's private_key*

⑤client and server both *session key*, then data transmission are encrypted