Basic Encryption and Decryption Operations

Public Key Cryptography

Public key encryption is a much slower alternative to symmetric cryptography. It is based upon mathematical functions upon two pairs of numbers. For the well-known RSA algorithm, the security comes from the difficulty of factoring large numbers in Galois Fields.

Each key is a pair of keys K and K-1. If a message is encrypted using K then it can only be decrypted using K-1. If A means the application of the encryption function and text is the cleartext, then the following all hold true.

(text) *A* K *A* K-1 = text

(text) *A* K *A* K ≠ text

(text) *A* K-1 *A* K = text

(text) *A* K *A* K ≠ text

Importantly, one cannot derive K from knowledge of K-1 or vice versa. This allows the primary use of public key technology, where one key is made public, and one key remains secret. This provides a much larger degree of functionality, extending the use of cryptography to supply authentication and integrity as well as confidentiality.

Authentication is provided by taking a piece of text, encrypting it using the private key which is only known by you. If it can be decrypted using your public key, then it is known to be encrypted by you. This then functions to authenticate the text.

Example of DNS Server

Client Operation: Retrieve **pub_ency_key_dns** from local resources/website
encrypt_data = **pub_ency_key_dns {**data_sent_to_dns_server**}**

Server Operation: Receive [encrypt_data]
de-crypt_data = **private_ency_key_dns** {encrypted_data}

client = host computer wanting DNS information
server = DNS server

**Scalability of encryption is important. While client to server communication can use server public key (pub_ency_key_dns) and use it to encrypt data sent to the sever, this scheme cannot be applied in the reverse direction. The client cannot generate its own public keys and require the server to use it (since there would be thousands of hosts in a network and it will not be possible for the sever to lookup each client's key). You need to work on determining possible alternates, that are scalable.*