CN414
Computer Network Security

Week 5:- Public Key

By

Dr. Piya Techateerawat

Public Key

- **Public Key Cryptosystems**
  - Diffie-Hellman
  - Rivest, Shamir, Adelman (RSA)

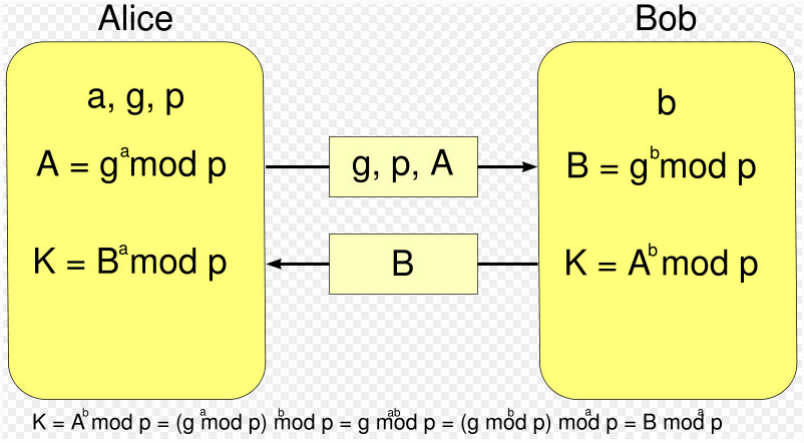
Public Key

- Public Key Cryptosystems
  - **Diffie-Hellman**
  - Rivest, Shamir, Adelman (RSA)

Diffie-Hellman

Diffie-Hellman key exchange (D-H) is a cryptographic protocol that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

Diffie-Hellman



Diffie-Hellman

1. Alice and Bob agree to use a prime number p=23 and base g=5.
2. Alice chooses a secret integer a=6, then sends Bob (g^a mod p)
   - 5^6 mod 23 = 8.
3. Bob chooses a secret integer b=15, then sends Alice (g^b mod p)
   - 5^{15} mod 23 = 19.
4. Alice computes (g^b mod p)^a mod p
   - 19^6 mod 23 = 2.
5. Bob computes (g^a mod p)^b mod p
   - 8^{15} mod 23 = 2.

Diffie-Hellman

- Strength ?
  - Strong protocol
  - Do not have to reveal the secret code
- Weakness ?
  - Man in the middle attack.
  - Authentication
  - Complexity

Public Key

- Public Key Cryptosystems
  - Diffie-Hellman
  - **Rivest, Shamir, Adelman (RSA)**

Rivest, Shamir, Adelman (RSA)

Key Generation Algorithm

1. Generate two large random primes, *p* and *q*, of approximately equal size such that their product $n = pq$ is of the required bit length, e.g. 1024 bits. [See note 1].
2. Compute $n = pq$ and $\varphi = (p-1)(q-1)$.
3. Choose an integer *e*, $1 < e < \varphi$, such that $\gcd(e, \varphi) = 1$. [See note 2].
4. Compute the secret exponent *d*, $1 < d < \varphi$, such that $ed \equiv 1 \pmod{\varphi}$. [See note 3].
5. The public key is (n, e) and the private key is (n, d). Keep all the values d, p, q and phi secret.

- n is known as the *modulus*.
- e is known as the *public exponent* or *encryption exponent* or just the *exponent*.
- d is known as the *secret exponent* or *decryption exponent*.

Rivest, Shamir, Adelman (RSA)

Encryption

Sender A does the following:-

1. Obtains the recipient B's public key (n, e).
2. Represents the plaintext message as a positive integer *m* [see note 4].
3. Computes the ciphertext $c = m^e \bmod n$.
4. Sends the ciphertext *c* to B.

Decryption

Recipient B does the following:-

1. Uses his private key (n, d) to compute $m = c^d \bmod n$.
2. Extracts the plaintext from the message representative *m*.

Rivest, Shamir, Adelman (RSA)

- Try to play it here.

http://www.cs.pitt.edu/~kirk/cs1501/notes/rsademo/alice.html

N = P x Q = 5x 3= 15
PHI = (P-1)(Q-1) = 8
The public exponent E will be generated by the computer so that the greater common divisor of E and PHI is 1. In other words, E is relatively prime with PHI.
E = 3 N and E are your public keys. Your private key (D) is the inverse of E modulo PHI.

By using extended Euclidian algorithm, the private key, **D, is 3 Don't forget to record N, E and D!**

Now, you can give your pair of public keys, E and N, to Bob so that he can send you encrypted letter by using those key. Later, you can decrypt it, using D and N.

Rivest, Shamir, Adelman (RSA)

- Coding in Java

RSA.doc

Rivest, Shamir, Adelman (RSA)

- Build-in function in Java

Create

```
KeyPairGenerator keyGen =
KeyPairGenerator.getInstance("DSA", "SUN");
```

Initialize

```
SecureRandom random =
  SecureRandom.getInstance("SHA1PRNG", "SUN");
  keyGen.initialize(1024, random);
```

Generate Key

```
KeyPair pair = keyGen.generateKeyPair(); PrivateKey priv =
  pair.getPrivate(); PublicKey pub = pair.getPublic();
```

Reference

- http://java.sun.com/docs/books/tutorial/security/apisign/step2.html@ 28 OCT 2008
- http://www.di-mgt.com.au/rsa_alg.html @ 28 OCT 2008
- http://en.wikipedia.org/wiki/Image:Diffie-Hellman-Schl%C3%BCsselaustausch.svg @ 28 OCT 2008


Q & A