Class229: Public Key Cryptography

Virendra Sule

Necessity of Public Key Cryptography

Symmetric key Cryptography suffers from two central issues.

- Symmetric key cryptography requires a secure channel for key exchange.
- Authentication of the identity is not resolved by confidentiality.

Even if symmetric keys are exchanged with most of the users number of keys (one for each pair of users) requires $O(n^2)$ shared keys at any point of time. Persons, servers, business entities who are meeting first time need to share the encryption key over insecure channel.

Proposals

- Diffie Hellman Key exchange (1976). (also Merkel at the same time) using exponential function on groups.
- RSA (1978). Asymmetric key encryption and decryption using two different keys. Based on prime factorization.
- El Gammal (1983). Asymmetric encryption using group exponentiation and Signature (solves the authentication problem).
- Koblitz (1983). Elliptic curve cryptography.

The field grew rapidly after these. However relatively very few public key schemes survived the test of security, relative to number of symmetric key secure algorithms. Currently a very vast field without proof of security.
Security is primarily computational since encryption key is public. Hence Perfect Secrecy is not possible.

A functional description.

- Parties A, B agree with a computational arithmetic.
- Create Private keys R_A, R_B independently.
- Exchange public keys $U_A = \phi(R_A)$, $U_B = \phi(R_B)$ over the public (insecure channel)
- Compute same shared key $S = \psi(R_A, U_B) = \psi(R_B, U_A)$.
- $\phi(.)$ is a OWF.

A functional description.

- Every user X has public key U_X and a private key R_X.
- Encryption algorithm is a TWOF,

$$C = E(U_X, P)$$

- Decryption possible only by private key R_X

$$P = D(R_X, C)$$

- $U_X = F(R_X)$ is a OWF.
- $C = E(F(R_X), P)$ is a TWOF with trapdoor R_X.

Diffie Hellman key exchange

- Choose a cyclic group G in which the discrete logarithm computation is a hard problem. g a generator of G. n is the group order.
- R_A, R_B private keys randomly chosen in $(1, n-1)$.
- $U_A = g^{(R_A)} = \phi(R_A)$, $U_B = g^{(R_B)}$.
- $S = (U_B)^{(R_A)} = (U_A)^{(R_B)}$. Hence $\psi(R_A, U_B) = (U_B)^{(}R_A) = \psi(R_B, U_A) = (U_A)^{(}R_B)$.
- Hardness of discrete log makes $U_A = g^{(R_A)} = \phi(R_A)$ a OWF.

Realization of DH scheme in the group $\mathbb{F}_p^*$

- Let g be a primitive element modulo prime p. Then $\mathbb{F}_p^* =< g >$.
- Let $R_A = a$, $R_B = b$ in $[1, p-1]$ be private keys.
- Public keys exchanged are

$$U_A = g^a \mod p,\ U_B = g^b \mod p$$

- Both parties A, B compute the shared key

$$S = U_B^a \mod p = U_A^b \mod p = g^{(ab)} \mod p$$

- Discrete log computation: computing a given $g^a \mod p$ is (believed to be) computationally infeasible for large enough p.

Realization of Public Key Encryption: RSA

- Private key of Alice $R = (p, q, d)$:
  - Two large primes p, q, $p \neq q$.
  - $n = pq$ the RSA modulus. p, q large enough such that factoring n is difficult.
  - $d < n$ is coprime to $\phi(n) = (p-1)(q-1)$
- Public key of Alice: $U = e$. There exists d such that

$$ed = 1 \mod \phi(n)$$

- Encryption: message m chosen in $[0, n-1]$. Bob computes Ciphertext

$$c = m^e \mod n$$

Decryption: Alice computes

$$\tilde{m} = c^d \mod n$$

We shall prove that $\tilde{m} = m$.

Proof of RSA decryption

Given (n, e) as above

$$(m^e)^d \mod n = m^{ed} \mod n$$

Since $n = pq$ by CRT, $m^{ed} \mod n$ is recovered uniquely by the residues $m^{(ed)} \mod p$ and $m^{(ed)} \mod q$. Since e is coprime to $\phi(n)$ there is an integer l such that

$$ed = 1 + l(p-1)(q-1)$$

Hence $m^{ed} \mod p = m^{(1+l(p-1)(q-1))} \mod p$. It follows that

$$m^{ed} \mod p \quad = \quad m(m^{(p-1)})^{l(q-1)} \mod p$$

If $p|m$ then the RHS is 0. If m is coprime to p then RHS is m. Hence the residues of m^{ed} are $(0, m)$, $(m, 0)$, (m, m) or $(0, 0)$. Hence by CRT the unique solution to $x \mod n$ whose residues match with any of these is m.

El Gammal encryption

- Public parameter p prime, g primitive element of $\mathbb{F}_p^*$.
- Private key of Alice $R = a$, $a \in [1, p-1]$ chosen randomly.
- Public key of Alice

$$A = g^a \mod p$$

- Message m in $[0, p-1]$ for Bob to encrypt. Bob chooses random b in $[1, p-1]$. Computes

$$B = g^b \mod p, \; C = mA^b \mod p$$

- Ciphertext sent to Alice

$$\hat{C} = (B, C)$$

- Decryption by Alice. Alice computes

$$\begin{aligned} S &= B^{-a} \mod p, \\ \tilde{m} &= CS \mod p \end{aligned}$$

Since $S = (B^a)^{-1} \mod p = (A^b)^{-1} \mod p$,

$$CS \mod p = m$$

Attacks on Public Key schemes

- RSA: Factorization of n. If prime factors p or q can be computed, then
$$d = e^{-1} \mod \phi(n)$$
can be computed. Hence m can be recovered from c. Problem of factoring n is practically infeasible for large p, q with additional conditions.
- Note: to set up an RSA p, q need to be chosen large enough such that factorization of n is infesible. However factorization of $(p-1)$, $(q-1)$ is required to choose a public exponent e. A concrete positive application of factorization of large numbers.
- Diffie Hellman key exchange: If the discrete log problem, computing $x < n$ in given $a = g^x$ in a cyclic group $G =< g >$ is feasible, then shared key S can be computed from public keys. Compute a given A, then $S = B^a$ in G.

Attacks...

- Discrete log problem is infeasible in $\mathbb{F}_p^*$ for large enough p and $\mathbb{F}_{2^n}$ for large enough n with minor additional conditions.
- In the elliptic curve group E discrete log problem is harder than the field discrete log.
- Factorization and Discrete log computations are harder problems than actually breaking RSA and DH scheme.
- Does breaking RSA lead to factorization? Computation of $\phi(n)$ is equivalent to prime factorization since p, q are roots of the quadratic equation

$$X^2 - nX + \phi(n) = 0$$

  Breaking RSA is a problem weaker than computing $\phi(n)$.
- In DH scheme, computing S given public keys A, B is weaker than discrete log problem.