Plan of Talk

- ElGamal Cryptosystem

ElGamal Cryptography

- A public-key cryptosystem related to D-H
- Uses exponentiation in a finite (Galois) field
- Security is based
  - difficulty of computing discrete logarithms, as in D-H
  - difficulty of computational D-H problem.
- The goal here is to motivate how ElGamal came up with the scheme, nearly after eight years of the discovery of DH protocol.
- Let us look at the DH protocol again.

Diffie-Hellman Key Establishment Protocol

p=11, g =2: Public

- Alice — Bob
- Choose Na=2 — Choose Nb=6
- $(g^{Na} \bmod p)=2^2=4 \pmod{11} =Ma$ $\longrightarrow$
- $\longleftarrow$ $g^{Nb} \bmod p= (2^6 \bmod 11)= 9=Mb$
- Compute
- $K_{ab} = Mb^{Na} \bmod 11 = 9^2=4$
- Compute
- $K_{ba} = Ma^{Nb} \bmod p = (4^6 \bmod 11) =4$
- $K_{ab} = K_{ba}=4$

Note that we may use variables p and q for representing primes. And, g and a for generators.

Salient Features

- DH protocol can be formulated over any cyclic group where computing discrete logarithm over the group is hard.
- What is the main objective?
  - Two users connected over insecure channel arrive at a common secret by using only public parameters.
  - In our case, they arrive at g^(ab), g is a generator of the group; a, b are random secrets chosen by the participants respectively.

Different Cyclic Groups

- Z_n: Integers modulo n, n is a positive integer.
- Z_p: Integer modulo p, p is a prime number.
- Residues of Polynomials over Z_p.
- Elliptic Curves over Z_p.

Order of Cyclic Groups

- What is the maximum size of cyclic groups obtained from Z_p?
- (p-1)
- What is the maximum size of cyclic groups obtained from Z_n?
- $\phi(n)$ = Numbers of integers < n but relatively prime to n.
- What is the maximum size of cyclic groups obtained from $Z_p[x]$ mod m(x), deg(m(x)) =k?
- P^k-1

A variation of DH

- Let us now assume that one of the users in the DH protocol is fixed in advance. Assume computations mod q, q is a prime. “a”: generator of the group.
- Alice generates the key in advance
  - chooses a secret key (number): $1 < x_A < q-1$
  - compute her **public key**: $y_A = a^{x_A} \bmod q$
- Bob knows this public key in advance

A variation of DH

- Bob
  - Choose a random k and compute $a^k \bmod q$
  - **Send** $a^k \bmod q$ **to Alice**
  - Since y_A is available, compute the DH common
  - key $y_A^k = a^{k\ x_A}$
  - Hide the message in the common key and send it to Alice
  - Bob to Alice: $C = M\ a^{k\ x_A}$
- Alice knows her secret x_A
- Obtain the common key in the cipher $(a^k)^{x_A} = a^{k\ x_A}$
- Recover Message $M = C / a^{k\ x_A}$

The scheme ElGamal Cryptography

- Public-key cryptosystem related to D-H
- Uses exponentiation in a finite (Galois)
- with security based difficulty of computing discrete logarithms, as in D-H
- each user (eg. A) generates their key
  - chooses a secret key (number): $1 < x_A < q-1$
  - compute their **public key**: $y_A = a^{x_A} \bmod q$
- NOTE: a is the generator here.

ElGamal Message Exchange

- Bob encrypt a message to send to A computing
  - represent message M in range $0 <= M <= q-1$
    - longer messages must be sent as blocks
  - chose random integer k with $1 <= k <= q-1$
  - compute one-time key $K = y_A^k \bmod q$
  - encrypt M as a pair of integers (C_1, C_2) where
    - $C_1 = a^k \bmod q$; $C_2 = KM \bmod q$
- A then recovers message by
  - recovering key K as $K = C_1^{xA} \bmod q$
  - computing M as $M = C_2 K^{-1} \bmod q$
- a unique k must be used each time
  - otherwise result is insecure

If k is not unique

- Let (M_1, C_1 = [C_{11}, C_{12}]) and
- (M_2, C_2 = [C_{21}, C_{22}]) be two message and ciphertext pais using the same randomization parameter k.
- What does this imply for C_1 and C_2 ?
- $C_{11} = a^k \bmod q = C_{21} = a^k \bmod q$
- If Adversary knows M_1, he can then recover M_2