Key Exchange Algorithms

Public Key Cryptosystem

Diffie-Hellman Key Exchange

- first public-key type scheme proposed
- Astounding concept that two parties can carry on a public conversation, and still end up with a secret that only the two of them know!
- Public Secret Sharing
- Shared secret is derived from private secrets and publically shared information

Diffie-Hellman Key Exchange

- first public-key type scheme proposed
- by Diffie & Hellman in 1976 along with the exposition of public key concepts
  - note: now know that Williamson (UK CESG) secretly proposed the concept in 1970
- is a practical method for public exchange of a secret key – in practice, a “pre-key”
- used in a number of commercial products

Diffie-Hellman Key Exchange

- a public-key distribution scheme
  - cannot be used to exchange an arbitrary message
  - rather it can establish a common key
  - known only to the two participants
- value of key depends on the participants (and their private and public key information)

Diffie-Hellman Key Exchange

- Derivation of shared secret is based on exponentiation in a finite (Galois) field (modulo a prime or a polynomial)

- easy

- security relies on the difficulty of working backward to get secrets by computing discrete logarithms (similar to factoring)

– hard

Diffie-Hellman Setup

- all users agree on global parameters:
  - large prime integer or polynomial **q**
  - **a**, which is a primitive root mod **q**
- each user (e.g. A) generates their key
  - chooses a secret key (number): $x_A < q$
  - computes their **public key**: $y_A = a^{x_A} \bmod q$
- each user makes public that key y_A

Diffie-Hellman Key Exchange

- shared session key for users A & B is K_{AB}:

$$K_{AB} = a^{x_A . x_B} \bmod q$$

$$= y_A^{x_B} \bmod q$$

(which B can compute)

$$= y_B^{x_A} \bmod q$$

(which A can compute)

Each principal has the other's public key and their own secret, along with **a** and **q**.

Diffie-Hellman Key Exchange

Bob Agree on $\mathbf{a}$ and $\mathbf{q}$ Alice

$$y_B = a^{x_B} \bmod q$$

$$y_A = a^{x_A} \bmod q$$

$$K_{AB} = y_A^{x_B} \bmod q \qquad K_{AB} = y_B^{x_A} \bmod q$$

Both Alice and Bob have

$$K_{AB} = a^{x_A . x_B} \bmod q$$

Diffie-Hellman Key Exchange

- K_{AB} is used as session key (or pre-key) in private-key encryption scheme between Alice and Bob
- if Alice and Bob subsequently communicate, they will have the **same** key as before, unless they choose new public-keys (can use nonces and pre-key to make session key different)
- attacker needs a private key **`x`**, must solve discrete log base **`a`** modulo **`q`** to get it

Diffie-Hellman Example

- users Alice & Bob who wish to swap keys:
- agree on prime $q=353$ and $a=3$
- select random secret keys:
  - A chooses $x_A=97$, B chooses $x_B=233$
- compute respective public keys:
  - $y_A=3^{97} \bmod 353 = 40$ (Alice)
  - $y_B=3^{233} \bmod 353 = 248$ (Bob)
- compute shared session key as:
  - $K_{AB}= y_B^{x_A} \bmod 353 = 248^{97} = 160$ (Alice)
  - $K_{AB}= y_A^{x_B} \bmod 353 = 40^{233} = 160$ (Bob)

Key Exchange Protocols

- users could create random private/public D-H keys each time they communicate
- users could create a known private/public D-H key and publish in a directory, then consult and use them to securely communicate with them
- both of these are vulnerable to a Man-in-the-Middle Attack
- authentication of the keys is needed

Man-in-the-Middle Attack

1. Darth prepares by creating two private / public keys
2. Alice transmits her public key to Bob
3. Darth intercepts this and transmits his first public key to Bob. Darth also calculates a shared key with Alice
4. Bob receives the public key and calculates the shared key (with Darth instead of Alice)
5. Bob transmits his public key to Alice
6. Darth intercepts this and transmits his second public key to Alice. Darth calculates a shared key with Bob
7. Alice receives the key and calculates the shared key (with Darth instead of Bob)

- ➢ Darth can then intercept, decrypt, re-encrypt, forward all messages between Alice & Bob

Man-in-the-Middle Attack

Bob Darth Alice

$$y_A = a^{x_A} \bmod q$$

$$y'_A = a^{x_{DA}} \bmod q$$

$$y_B = a^{x_B} \bmod q$$

$$y'_B = a^{x_{DB}} \bmod q$$

$$K_{DAB} = {y'_A}^{x_B} \bmod q \quad K_{ADB} = {y'_B}^{x_A} \bmod q$$

Darth has a private, unauthenticated channel with each of Alice and Bob

Man-in-the-Middle Attack

- Also known as “Bucket Brigade” Attack
- Need reliable way to associate public key with principal
- Public key infrastructure (PKI) is one way
- PGP web of trust is another
- In some circumstances, may be possible to use scheduling/timing to prevent MITM

ElGamal Cryptography

- public-key cryptosystem related to D-H
- uses exponentiation in a finite field
- with security based difficulty of computing discrete logarithms, as in D-H
- each user (e.g. A) generates their key
  - chooses a secret key (number): $1 < x_A < q-1$
  - computes their **public key**: $y_A = a^{x_A} \bmod q$

ElGamal Message Exchange

- Bob encrypts a message to send to A computing
  - message M in range $0 <= M <= q-1$
    - longer messages must be sent as blocks
  - chose random integer k, $1 <= k <= q-1$
  - compute one-time key $K = y_A^k \bmod q$
  - encrypt M as a pair of integers (C_1, C_2) where
    - $C_1 = a^k \bmod q$ // like D-H public key
    - $C_2 = KM \bmod q$ // encrypted msg

ElGamal Message Exchange

- 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

ElGamal Example

- use field GF(19) $q=19$ and $a=10$
- Alice computes her key:
  - A chooses $x_A=5$ & computes $y_A=10^5 \bmod 19 = 3$
- Bob send message $m=17$ as $(11,5)$ by
  - chosing random $k=6$
  - computing $K = y_A^k \bmod q = 3^6 \bmod 19 = 7$
  - computing $C_1 = a^k \bmod q = 10^6 \bmod 19 = 11;$
    $C_2 = KM \bmod q = 7.17 \bmod 19 = 5$
- Alice recovers original message by computing:
  - recover $K = C_1^{xA} \bmod q = 11^5 \bmod 19 = 7$
  - compute inverse $K^{-1} = 7^{-1} = 11$
  - recover $M = C_2\ K^{-1} \bmod q = 5.11 \bmod 19 = 17$