Other Public Key Cryptosystems

ITC 3093 Principles of Computer Security

Based on Cryptography and Network Security by William Stallings
and Lecture slides by Lawrie Brown
and Introduction to Cryptography and Security Mechanisms by Dr Keith Martin

Diffie-Hellman Key Exchange

Diffie-Hellman Key Exchange

- first public-key type scheme proposed
- by Diffie & Hellman in 1976 along with the exposition of public key concepts
  - *note: we now know that Williamson (UK CESG) secretly proposed the concept in 1970, and subsequently declassified in 1987*
- is a practical method for public exchange of a secret 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)
- based on exponentiation in a finite (Galois) field (modulo a prime or a polynomial) – easy
- security relies on the difficulty of computing discrete logarithms (similar to factoring) – hard

Diffie-Hellman Setup

- all users agree on global parameters:
  - large prime integer or polynomial q
  - a being a primitive root mod q
- each user (eg. A) generates their key
  - chooses a secret key (number): $x_A < q$
  - compute 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}:

$$K_{AB} = a^{x_A . x_B} \bmod q$$
$$= y_A^{x_B} \bmod q \quad \text{(which } \mathbf{B} \text{ can compute)}$$
$$= y_B^{x_A} \bmod q \quad \text{(which } \mathbf{A} \text{ can compute)}$$

- K_{AB} is used as session 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
- attacker needs an x, must solve discrete log

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 consulted and used to securely communicate with them
- both of these are vulnerable to a meet-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)
8. Darth can then intercept, decrypt, re-encrypt, forward all messages between Alice & Bob

ElGamal Cryptography

ElGamal

- We will also take a look at the ElGamal public key cipher system for a number of reasons:
  - To show that RSA is not the only public key system
  - To exhibit a public key system based on a different one way function
  - ElGamal is the basis for several well-known cryptographic primitives

Setting up ElGamal

- Let **p** be a large prime
  - By “large” we mean here a prime rather typical in length to that of an RSA modulus
- Select a special number **g**
  - The number g must be a **primitive element** modulo **p**.
- Choose a private key **x**
  - This can be any number bigger than 1 and smaller than **p**-1
- Compute public key **y** from **x**, **p** and **g**
  - The public key **y** is **g** raised to the power of the private key **x** modulo **p**. In other words:

$$y = g^x \bmod p$$

Setting up ElGamal: example

Step 1: Let p = **23**

Step 2: Select a primitive element g = **11**

Step 3: Choose a private key x = **6**

Step 4: Compute y = $11^6 \pmod{23}$
= **9**

Public key is **9**
Private key is **6**

ElGamal encryption

- The first job is to represent the plaintext as a series of numbers modulo p.
- Then:
  1. Generate a random number k
  2. Compute two values C_1 and C_2, where

$$\mathbf{C_1 = g^k \bmod p} \quad \text{and} \quad \mathbf{C_2 = My^k \bmod p}$$

  3. Send the ciphertext C, which consists of the two separate values C_1 and C_2.

ElGamal encryption: example

To encrypt M = 10 using Public key **9**

1 - Generate a random number k = **3**

2 - Compute $C_1 = 11^3 \bmod 23 = 20$

$C_2 = 10 \times 9^3 \bmod 23$

$= 10 \times 16 = 160 \bmod 23 = 22$

3 - Ciphertext C = (**20** , **22**)

ElGamal decryption

$$C_1 = g^k \bmod p \qquad C_2 = My^k \bmod p$$

1 - The receiver begins by using their private key **x** to transform C_1 into something more useful:

$$C_1^x = (g^k)^x \bmod p$$

NOTE: $$C_1^x = (g^k)^x = (g^x)^k = (y)^k = y^k \bmod p$$

2 - This is a very useful quantity because if you divide C_2 by it you get **M**. In other words:

$$C_2 / y^k = (My^k) / y^k = M \bmod p$$

ElGamal decryption: example

To decrypt C = (20 , 22)

1 - Compute $20^6 = 16 \bmod 23$

2 - Compute $22 / 16 = 10 \bmod 23$

3 - Plaintext = 10

Security of ElGamal

- Recall the two different strategies for trying to “break” RSA:
  - Trying to decrypt a ciphertext without knowledge of the private key
  - Trying to determine the private key

What hard problems do you come across if you try to follow these two different strategies to break ElGamal?

ElGamal v RSA

PROS of ElGamal

- Does not rely on factorisation being hard

CONS of ElGamal

- Requires a random number generator
- Message expansion

While regarded as similar from a security perspective, are there any differences between ElGamal and RSA from an efficiency perspective?

Public key systems in practice

- Public key cipher systems led to mini revolution in cryptography in the mid 1970's, with a further boom in interest since the development of the Internet in the 1990's.
- Public key cipher systems are only likely to grow in importance in the coming years.
- In upcoming lessons we discuss:
  - cryptographic services, some of which involve public key techniques.
  - digital signatures, one of the major applications of public key cipher systems
  - the big problem of authenticating public keys