Department of Electronics

Cryptography

Fall 2019

Hasan Mahmood

hasan@qau.edu.pk

Week 11 (13, 14 November 2019)

Diffie-Hellman Key Exchange Revised, III

9. Alice asks Trent to verify that ver_B is Bob's verification algorithm.
10. Alice uses ver_B to verify Bob's signature.
11. Alice sends $E_K(sig_A(\alpha^x, \alpha^y))$ to Bob.
12. Bob decrypts, asks Trent to verify that ver_A is Alice's verification algorithm, and then uses ver_A to verify Alice's signature.

Key Distribution

- Strength of a cryptographic algorithm lies in the security of its keys
- If Alice and Bob are unable to meet in order to exchange their keys, can they still decide on a key without compromising further communications?
- Symmetric key cryptography, both Alice and Bob use same key for encryption and decryption
- Public key methods, RSA, where the sender has one key, and the receiver has another
- Public key encryption, keys are stored in public databases
- Computationally inefficient or slow, e.g., RSA is used to transmit a DES key

Key Pre-distribution

- Alice wants to communicate with Bob
- Key schedule is decided in a
- Vance and this information is sent securely from one to the other
  - Used by German navy in World War II
  - Codebooks were captured from ships
- Problem with this method, Alice and Bob have to meet again, if keys can are compromised
- We need a trusted authority, Trent

Key Pre-distribution

- For every pair of users, call them (*A*, *B*), Trent produces a random key K_{AB} (to be used for symmetric key encryption, $K_{AB}=K_{BA}$)
- It is assumed that Trent is powerful and has established a secure channel to every user
- Each user will receive n-1 keys. Total key distributed is $n(n-1)/2$
- Blom key pre-distribution scheme is used
- Start with a large prime p (known to all), with n users ($p>=n$)

Blom Key Pre-distribution

1. Each user U in the network is assigned a distinct public number r_U (mod p).

2. Trent chooses three secret random numbers a, b, and c mod p.

3. For each user U, Trent calculates the numbers

$$a_U \equiv a + br_U \pmod{p} \quad b_U \equiv b + cr_U \pmod{p}$$

and sends them via his secure channel to U.

Blom Key Pre-distribution

4. Each user U forms the linear polynomial

$$g_U(x) = a_U + b_U x.$$

5. If Alice (A) wants to communicate with Bob (B), then Alice computes $K_{AB} = g_A(r_B)$, while Bob computes $K_{BA} = g_B(r_A)$.

6. It can be shown that $K_{AB} = K_{BA}$ (Exercise 2). Alice and Bob communicate via a symmetric encryption system, for example, DES, using the key (or a key derived from) K_{AB}.

Example

- Network with Alice, Bob and Charlie
- p=23
- Let

$$r_A = 11, \quad r_B = 3, \quad r_C = 2.$$

- Suppose Trent chooses the number a=8, b=3, c=1
- Corresponding linear polynomials are

$$g_A(x) = 18 + 14x, \quad g_B(x) = 17 + 6x, \quad g_C(x) = 14 + 5x.$$

Example

- It is now possible to calculate the keys that this scheme would generate

$$K_{AB} = g_A(r_B) = 14, \quad K_{AC} = g_A(r_C) = 0, \quad K_{BC} = g_B(r_C) = 6.$$

- Can be decoded if Eve and Oscar conspire

$$\begin{pmatrix} 0 & 1 & r_E \\ 1 & r_O & 0 \\ 0 & 1 & r_O \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} \equiv \begin{pmatrix} b_E \\ a_O \\ b_O \end{pmatrix} \pmod{p}.$$

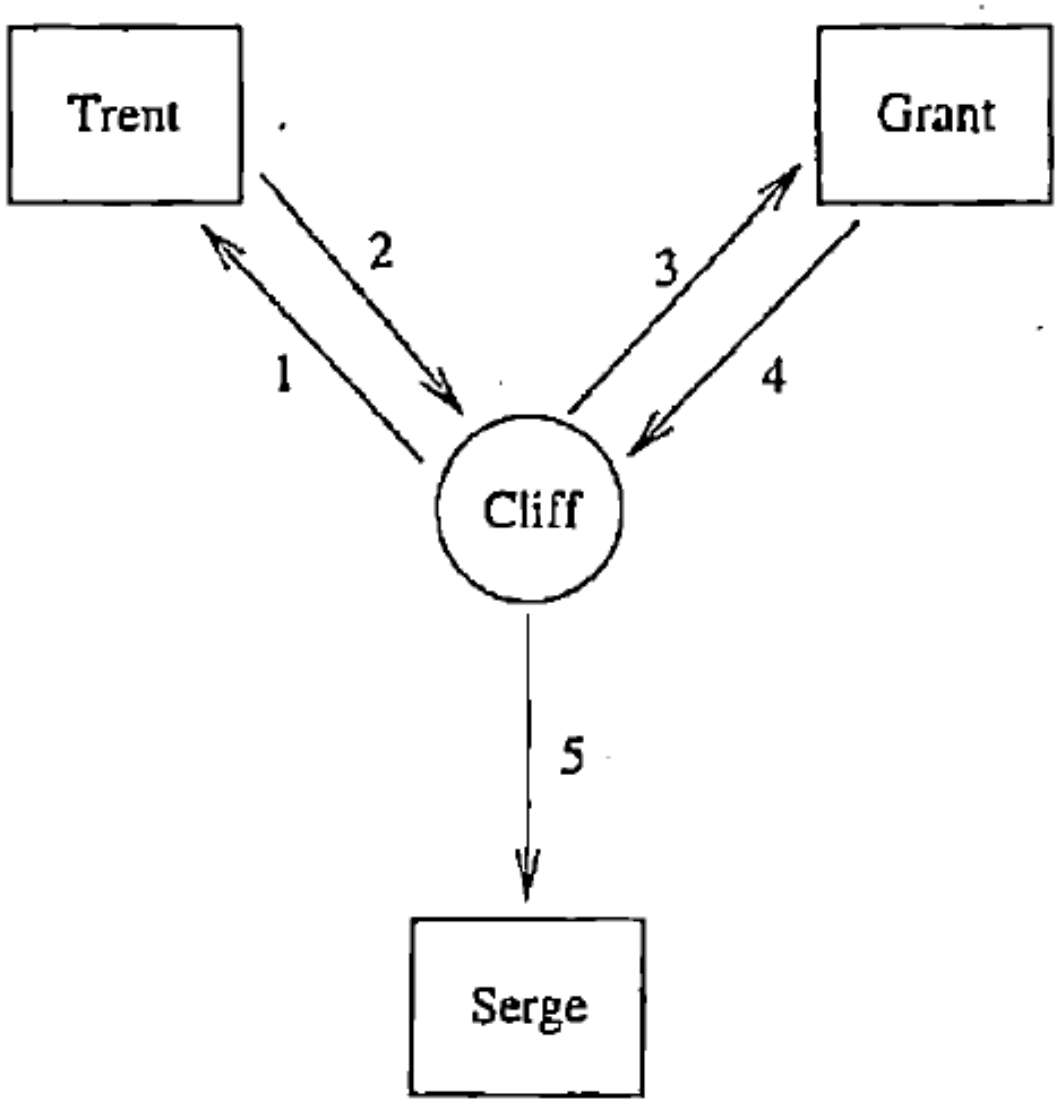
Kerberos

- Kerberos developed at MIT as project Athena
- Objective is to provide huge network of computer workstations to students
- Insecure communications on public networks
- Client/server authentication model architecture

- Cliff: a client
- Serge: a server
- Trent: a trusted authority
- Grant: a ticket-granting server

Kerberos

- Cliff and Serge have ne secret key information shared between them
- Serge needs to verify Cliff's identity



Public Key Infrastructures (PKI)

- Public key cryptography allows for authentication, key distribution and non-repudiation
- Authenticity of published public keys (Eve may substitute her own public key in place of Alice's key
- The benefits of public key depends on authenticity and validity
- Infrastructure to track public keys is required
- PKI infrastructure to define policies, procedures for publishing keys and certificates
- Certification binds a public key to an entity (user, or a piece of info)

Certificate

- Certificate is a quantity of information that has been signed by its publisher, who is commonly referred to as the certification authority (CA)
- Identity certificates, and credential certificates (two types)
- Date encrypted through CA's public key
- CA publishes identity certificates for Alice and Bob
- If Alike knows CA's public key, she can verify Bob's information
- Trusted company publishes manages the public key for Bob (govt or phone company)
- Many CA's are operating

Digital Cash

- Anonymous payments, via cash only
- Is it possible to have electronics cash?
- Duplicate electronics coins
- Double spending
- If coins are recorded as they are spent, anonymity is compromised
- Communications with a central bank could fail temporarily
- Verification required via bank

Properties of Digital Cash

1. The cash can be sent securely through computer networks.
2. The cash cannot be copied and reused.
3. The spender of the cash can remain anonymous. If the coin is spent legitimately, neither the recipient nor the bank can identify the spender.
4. The transaction can be done *offline*, meaning no communication with the central bank is needed during the transaction.
5. The cash can be transferred to others.
6. A piece of cash can be divided into smaller amounts.

Digital Cash System

- Participants:
  - Bank
  - Spender
  - Merchant
  - Fraud Control

Initialization

- Done once by some central authority
- Choose a large prime such that $q=(p-1)/2$
- Let g be the square of a primitive root mode p
- This implies:

$$g^{k_1} \equiv g^{k_2} \pmod{p} \iff k_1 \equiv k_2 \pmod{q}$$

- Two secret exponents are chosen
- g, g_1, and g_2 are made public

Hash Functions

- Two public hash functions are chosen
- H
  - Takes 5-tuple of integers as input and outputs an integer mod q
- H_0
  - Takes a 5-tuple of integers as input and outputs an integer mod q

The Bank

- The bank chooses its secret identity number x and computes

$$h \equiv g^x, \quad h_1 \equiv g_1^x, \quad h_2 \equiv g_2^x \pmod{p}.$$

- The numbers h, h_1 and h_2 are made public
- These numbers identity the bank

The Spender

- The spender chooses a secret identity number *u* and computes the account number

$$I \equiv g_1^u \pmod{p}.$$

- The number *I* is sent to bank, which stores *I* along with the information identifying the spender
- Name, address, ID, etc. stored
- The spender does not send *u* to the bank. The bank sends to the Spender
-

$$z' \equiv (I g_2)^x \pmod{p}$$

The Merchant

- The Merchant chooses an identification number *M* and registers it the bank.

Creating a Coin

- The Spender contacts the bank, asking for a coin
- The bank requires proof of identity (similar to withdrawing classical cash from the bank)
- All coins in this scheme have same value
- A coin is represented by a 6-tuple number

$$(A, B, z, a, b, r).$$

- Most of the effort in this system is needed to preserve anonymity and double spending

Creating a Coin

Here is how the numbers are constructed.

1. The Bank chooses a random number w (a different number for each coin), computes

$$g_w \equiv g^w \text{ and } \beta \equiv (Ig_2)^w \pmod{p},$$

and sends g_w and β to the Spender.

Creating the Coin, Spender Secret Numbers

2. The Spender chooses a secret random 5-tuple of integers

$$(s, x_1, x_2, \alpha_1, \alpha_2).$$

3. The Spender computes

$$\begin{aligned} A &\equiv (Ig_2)^s, \quad B \equiv g_1^{x_1} g_2^{x_2}, \quad z \equiv z'^s, \\ a &\equiv g_w^{\alpha_1} g^{\alpha_2}, \quad b \equiv \beta^{s\alpha_1} A^{\alpha_2} \pmod{p}. \end{aligned}$$

Creating the Coin, Conditions

- Coins with A=1 are not allowed
- This may result in Spender solving the discrete log problem

Spender Calculations

4. The Spender computes

$$c \equiv \alpha_1^{-1} H(A, B, z, a, b) \pmod{q}$$

and sends c to the Bank. Here H is the public hash function mentioned earlier.

5. The Bank computes $c_1 \equiv cx + w \pmod{q}$ and sends c_1 to the Spender.

Spender Calculations

6. The Spender computes

$$r \equiv \alpha_1 c_1 + \alpha_2 \pmod{q}.$$

The coin (A, B, z, a, b, r) is now complete. The amount of the coin is deducted from the Spender's bank account.

Spending the Coin

The Spender gives the coin (A, B, z, a, b, r) to the Merchant. The following procedure is then performed:

1. The Merchant checks whether

$$g^r \equiv a\, h^{H(A,B,z,a,b)} \quad A^r \equiv z^{H(A,B,z,a,b)} b \pmod{p}.$$

If this is the case, the Merchant knows that the coin is valid. However, more steps are required to prevent double spending.

2. The Merchant computes

$$d = H_0(A, B, M, t),$$

where H_0 is the hash function chosen in the initialization phase and t is a number representing the date and time of the transaction. The number t is included so that different transactions will have different values of d. The Merchant sends d to the Spender.

3. The Spender computes

$$r_1 \equiv dus + x_1, \quad r_2 \equiv ds + x_2 \pmod{q},$$

where u is the Spender's secret number, and s, x_1, x_2 are part of the secret random 5-tuple chosen earlier. The Spender sends r_1 and r_2 to the Merchant.

4. The Merchant checks whether

$$g_1^{r_1} g_2^{r_2} \equiv A^d B \pmod{p}.$$

If this congruence holds, the Merchant accepts the coin. Otherwise, the Merchant rejects it.

The Merchant Deposits the Coin in the Bank

A few days after receiving the coin, the Merchant wants to deposit it in the Bank. The Merchant submits the coin (A, B, z, a, b, r) plus the triple (r_1, r_2, d). The Bank performs the following:

1. The Bank checks that the coin (A, B, z, a, b, r) has not been previously deposited. If it hasn't been, then the next step is performed. If it has been previously deposited, the Bank skips to the Fraud Control procedures discussed in the next subsection.

2. The Bank checks that

$$g^r \equiv a\, h^{H(A,B,z,a,b)} \quad A^r \equiv z^{H(A,B,z,a,b)} b, \text{ and } g_1^{r_1} g_2^{r_2} \equiv A^d B \pmod{p}.$$

If so, the coin is valid and the Merchant's account is credited.