Introduction to Cryptology

Lecture-09
Public-Key Cryptography
Diffie-Hellman Key-Exchange System

09.05.2023, v41

1

Lecture Outlines

- Historical Overview
- Public Key Principles and its Breakthrough
- Diffie Hellman Public Key Exchange System

2

Historical Public-Key Breakthrough in Cryptography

Shannon's Breakthrouh in Communication Technology in 1948

1- Shannon (AT&T) 1948 'A Mathematical Theory of Communication,

Shannon's Breakthrough in Communication:
Error-free transmission is possible on noisy channels! $C = B \log_2 (1 + S/N)$

His second publication on secrecy systems one year later:

2- Shannon (AT&T) 1949 'Communication Theory of Secrecy Systems'

However, Shannon introduced no breakthrough in cryptography but he introduced mathematical mean to deal with security systems and proved that: Vernam Cipher is perfect and is unbreakable

The Breakthrough to the "Modern Cryptology" came in 1976

Diffie and Hellman (Stanford University) introduced the Concept of the Public key Cryptography in 1976

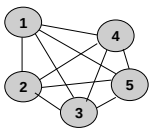
Diffie Hellman 's Breakthrough in Cryptography:
Secured transmission is possible on unsecured channels without any secret agreement!

Unfortunately: No unbreakable (perfect) public-key system is so far known !!!!
A possible future breakthrough is to find a non-breakable public-key system?

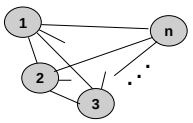
3

Why Public-Key Cryptography ?

Question: How many secret-keys needed to be exchanged in order to set up a system of n-users?



10 key-exchanges for 5 users



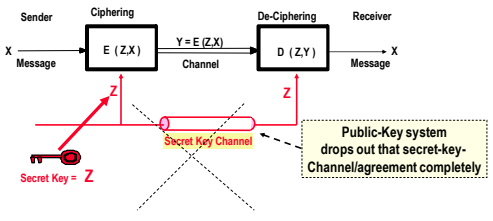
$\frac{n(n-1)}{2}$ keys for n users

For 10 000 users, we need 50 million key- exchanges. This is very hard !

Public-key systems makes this exchange unnecessary

4

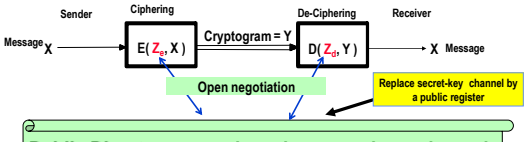
The Target of Public Key Cryptography is to communicate securely without prior secret key exchange



Secret key exchange is very hard and non-practical!:

5

The Solution was
The breakthrough discovery of Public-Key Cryptography by Diffie and Hellman in 1976



Public Directory to replace the secret keys channel

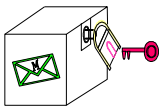
- Public Open Agreement
- No Prior Secret Communication !

6

Basic Public-Key Changes Concept

(Mechanical Model)

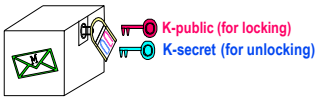
Conventional Secret Key Systems



K-open = K-close
(Symmetric System)

-Open and close using the same key which need to be agreed on secretly !!

Key Idea of Public Key Systems: Using two different keys!



K-open ≠K-close
(Asymmetric System)

- Open and close with different keys!!
- No Secret Key Agreement required

First two schemes in Public Key Cryptography:

- Diffie-Hellman key exchange scheme 1976
- RSA public key secrecy system 1978

7

Public-Key Cryptography

First introduced by Diffie and Hellman 1976 (Stanford University)

Key Revolutionary Idea for sharing a secret:

no need for any prior secret agreement in order to share a common secrets

HOW ?

1. Every user generates two keys:

kept secret
published

2. Any two users start open negotiations resulting with exclusively shared secret !

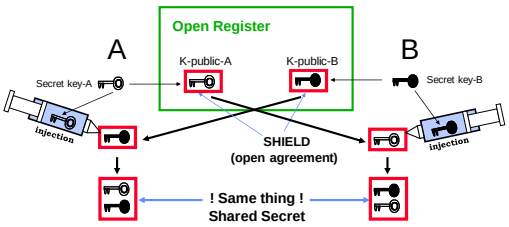
Diffie and Hellman showed how could that work securely under some assumptions!

8

Open Key-Agreement Breakthrough 1976

Shared Secret without exchange of secrets "Mechanical simulation"

Diffie-Hellman proposed mechanism

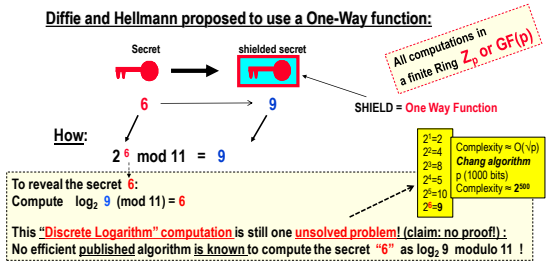


9

The key-question in Public-key systems is How to "*publicly*" hide (shield) a secret ?

Key idea: by using the so called ***One-way Functions (OWF)***

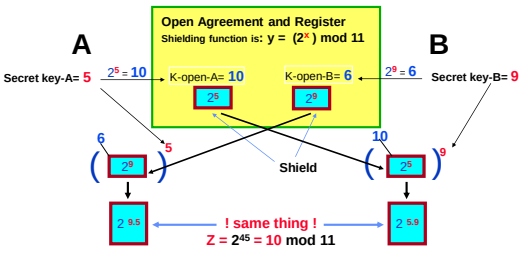
Diffie and Hellmann proposed to use a One-Way function:



10

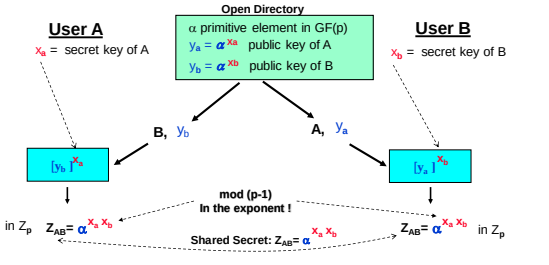
Example for *Diffie-Hellman* key exchange scheme 1976

Widely use in internet and banking ...



11

Conventional Diffie-Hellman Public Key Distribution System



12

Example:

A public key exchange system is setup according to Diffie-Hellman key exchange scheme with $y = 7^x \bmod 23$

1. Compute the public keys Y_A, Y_B of users A and B if their secret keys are X_A=5 and X_B=4 respectively.
2. Compute the common shared secret Z_{AB} between users A and B according to Diffie-Hellman key exchange scheme

Solution two-way exchange :

Secret Key $X_A = 5$ — Secret Key $X_B = 4$

Public key for user A is : $Y_A = 7^5 \bmod 23 = 17$

Public key for user B is : $Y_B = 7^4 \bmod 23 = 9$

$Z_{AB} = 9^5 \bmod 23 = 8$

$Z_{AB} = 17^4 \bmod 23 = 8$

Common Secret is = 8

13

Practical Use Cases of "Diffie-Hellman-Lock" as Public-Key Systems

Three selected application scenarios:

1- SEEK: (Secure Electronic Exchange of Keys)

2- Authenticated Public Key Distribution System

3- Two-Way Authenticated Public Key Distribution System

- Many other scenarios ….

14

"SEEK" Secure Electronic Exchange of Keys

A Public-Key Secrecy System

Based on Diffie Hellman Key exchange Scheme [SEEK (Omura) cryptosystem 1985]

Step 1: DH-Key agreement

Open Directory: Arithmetic in GF(q); α = primitive element in GF(q)

User A: E_a = secret key — User B: E_b = secret key

1: α^{E_a} → 2: α^{E_b}

$Z = (\alpha^{E_b})^{E_a} = \alpha^{E_b E_a}$ = shared secret key = Z = $(\alpha^{E_a})^{E_b} = \alpha^{E_a E_b}$

If gcd(Z, q-1) = 1, then Z is adopted as a suitable shared key. [the inverse of Z is computed as D = Z^{-1} (mod q -1)]
If gcd(Z, q-1) ≠1, then repeat step 1 above until Z becomes invertible.

Step 2: Data Encryption and Decryption by Exponentiation in GF(q)

M^Z → M^Z → $(M^Z)^D = M$

15

Two-Way Authenticated Public Key Distribution System

Based on Diffie-Hellman Scheme

(Alexandris, Burmester, Chrissikopoulos, Desmedt ,1993)

Open Directory: α primitive element in GF(p); $y_a = \alpha^{X_a}$ public key of A; $y_b = \alpha^{X_b}$ public key of B

User A: x_a = secret key of A; r_1 = random in Z_{p-1}

User B: x_b = secret key of B; r_2 = random in Z_{p-1}

$E_a = (r_1 - x_a)$ in Z_{p-1} — $E_b = (r_2 - x_b)$ in Z_{p-1}

B, α^{E_b} — A, α^{E_a}

$[\, y_b \cdot \alpha^{E_b}\,]^{r_1}$ in Z_p — $[\, y_a \cdot \alpha^{E_a}\,]^{r_2}$ in Z_p

$[\alpha^{x_b} \cdot \alpha^{r_2 - x_b}\,]^{r_1} = \alpha^{r_1 r_2}$ — $[\alpha^{x_a} \cdot \alpha^{r_1 - x_a}\,]^{r_2} = \alpha^{r_1 r_2}$

Shared Secret: $Z_{AB} = \alpha^{r_1 r_2}$

16

Two-Way Authenticated Public Key Distribution System

Based on Diffie-Hellman-Hughs Scheme

(Shared-key enforced by user A !)

Open Directory: α : as primitive element in GF(p)

User A: x_a = secret key of A

User B: r = random in Z_{p-1} such that: gcd(p-1, r) = 1; Compute r^{-1} in Z_{p-1}

$Y_1 = \alpha^r$

$Z = \alpha^{Xa}$

$Y_2 = [Y_1]^{Xa} = \alpha^{r \cdot Xa}$

Y_2

$[Y_2]^{r^{-1}} = [\alpha^{r \cdot Xa}]^{r^{-1}}$

$= \alpha^{Xa} = Z$

User A can enforce a certain key value Z !

17

Remarks on Diffie-Hellman (DH) Public Key System

Security considerations and few known facts:

1. **Based on** the claim that the **discrete logarithm** which is **claimed to be** not efficiently computable
2. **Breaking Complexity**: A primitive element α from GF(p) or GF(2^m) is used to make exhaustive search algorithms infeasible. If $y = \alpha^i$, only y and α are known. To break the system, we need to find i. To find i, the powers α^i are computed in the given GF until $y = \alpha^i$, then i=t. A maximum of **p-1 or 2^m-1 cycles** are required by a primitive search to find the discrete logarithm t (smarter algorithms require less complexity to compute t). The order of α as a primitive element is p-1 in GF(p) or 2^m-1.in GF(2^m). Therefore, p is selected as 1000 to 4000 bit prime or m> 1000 to attain a good security level. **Best known cryptanalysis algorithms are proportional to $\sqrt{p}$ (Chang).**

   Caution: There is no evidence that no efficient algorithms can be found to break the system .
3. **Hint: When designing DH system, (p-1)** should have large prime factor to make the discrete logarithm computation infeasible (p is called then a **strong prime**).

18

Design Summary for

Diffie-Hellman (DH) Public Key Exchange System

A possible design procedure for DH system over GF(p) is:

1. Select a strong prime number **p** such that **p-1 = 2 q** where q is a prime.
   A possible procedure is to use Pocklington's Theorem to find such a prime:

   - Select N = 2q + 1 where q is a large prime
     Check if the resulting N is prime according to Pocklington's theorem
   - If N is prime, take p=N to define GF(p)

2. Find a primitive element α in GF(p) by selecting any non-zero random value and checking if its order is p-1. The order of any element in GF(p) is a divisor of p-1=2q. That is the order can be either 1, 2, q or 2q.
   If $\alpha^2 \neq 1$ and $\alpha^q \neq 1$ then the order of α is 2q and α is a primitive element.
   Repeat 3 until you get a primitive element.

3. Publish GF(p) and α in a public directory. The system is ready for use.

NOTICE: the one who generates p should be trustable! (fake prime!)

Notice: There are other algebraic groups for creating DH systems other than those in GF(p) or GF(2^m).
One widely-used system would be shown in a later sections in this lecture. (Additive Groups in Elliptic-Curves)