Introduction to Cryptology

Lecture-14
Cryptographic Protocols
Secret Sharing, Threshold Security

22.05.2023, v42

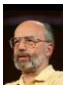
Outlines

- No key Security protocols
  Shamir 3-Pass Protocol
  - Omura-Massey Lock over GF(p)
  - Massey Omura Lock over GF(2^m)
- Secret Sharing Protocols
  - Non-Perfect Secret Sharing
  - Perfect Secret Sharing
- Threshold Schems
  - Shamir's Threshold Scheme

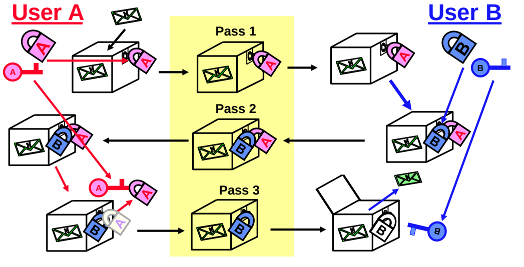
No-Key Secrecy Protocols

Secrecy procedure require
- No secret agreement
- No open keys at all

Cryptographic Protocols



No Key Cryptography : Shamir's 3-Pass Protocol
(Mechanical scenario/simulation)



Vernam One-Time-Pad Lock for: Shamir's 3-Pass Protocol

User A | Vernam Cipher | User B

User A: Z_A; (message) = M

1: $Y_1 = M + Z_A$ →

User B: Z_B; $Y_2 = Y_1 + Z_B$

2: ← $Y_2 = M + Z_A + Z_B$

User A: $Z_A + Y_2 = Y_3$; Remove Key Z_A

3: $Y_3 = M + Z_B$ →

User B: $M = Y_3 + Z_B$

Attack: $Y_1 + Y_2 + Y_3 = M$

Simple Product Cipher Lock for: Shamir's 3-Pass Protocol

User A | Multiplication Cipher | User B

User A: Z_A; (message) = M

1: → $Y_1 = M * Z_A$ →

User B: Z_B; $Y_2 = Y_1 * Z_B$

2: ← $Y_2 = M * Z_A * Z_B$ ←

User A: $Y_3 = Y_2 * Z_A^{-1}$

3: → $Y_3 = M * Z_B$ →

User B: $M = Y_3 * Z_B^{-1}$

Attack: $Y_2 / Y_1 = Z_B$ then $M = Y_3 * Z_B^{-1}$

Matrix Product Lock
for: Shamir's 3-Pass Protocol

User A — Product Cipher — User B

Z_A $\qquad$ Z_B

= M $\xrightarrow{1}$ $Y_1 = [M] * [Z_A]$ $\longrightarrow$

$\longleftarrow$ 2 $Y_2 = [Z_B] * [M] * [Z_A]$ $\longleftarrow$

$Y_3 = Y_2\, [Z_A]^{-1}$ $\xrightarrow{3}$ $Y_3 = [Z_B] * [M]$ $\longrightarrow$ $M = [Z_B]^{-1} * Y_3$

Attack: $Y_2 / Y_1 = Z_B$ then $M = [Z_B]^{-1} * Y_3$

7

Omura-Massey Lock* over GF(p)
for: Shamir's 3-Pass Protocol
Secrecy without Authenticity

p = large prime, arithmetic in GF(p)
[modulus in the exponent is (p-1)]

User A
E_a = secret key
$D_a = E_a^{-1}$ (mod p-1)
gcd (E_a , p -1) = 1

User B
E_b = secret key
$D_b = E_b^{-1}$(mod p-1)
gcd (E_b , p -1) = 1

= M $\xrightarrow{1}$ M^{E_a} $\longrightarrow$ $(M^{E_a})^{E_b}$

$\xleftarrow{2}$ $M^{E_a E_b}$

$(M^{E_a E_b})^{D_a}$ $\xrightarrow{3}$ M^{E_b} $\longrightarrow$ $(M^{E_b})^{D_b}$ = M

gcd (E_i , p -1) = 1
# of keys = φ(p-1)

* J.L. Massey & J. K. Omura, US Patent, 1986

8

A secrecy system based on
Omura-Massey Lock for: Shamir's 3-Pass Protocol

1. Use the Shamir 3-path protocol to exchange a secret key M = Z
(Z must be a unit to be invertible, gcd (Z, p-1) = 1 → $D = Z^{-1}$)

2. Use another exponentiation to exchange data as follows:

User A
Z = secret key from 1
D = Z^{-1}

p = large prime, arithmetic in GF(p)

User B
Z = secret key from 1
D = Z^{-1}

= M $\xrightarrow{1}$ M^Z $\longrightarrow$ $(M^Z)^D$ = M

9

Example : Omura-Massey Lock for Shamir 3-Pass Protocol

Using GF(19) and the secret keys for users A and B as 7 and 11

Hiding/locking function: M^{Key} mod 19, arithmetic in GF(19)

Modulus in the exponent =(19 -1)=18 (Euler theorem)

Prime number

$e_1 = 7$
gcd(18,7)=1
7 x 13 = 1 mod 18

$e_2 = 11$
11 x 5 = 1 mod 18

USER A
$e_1 = 7$
$d_1 = e_1^{-1} = 13$
Message M = 2

USER B
$e_2 = 11$
$d_2 = e_2^{-1} = 5$

$Y_1 = (2^7) = (14)$ mod 19 $\longrightarrow$

$Y_2 = (2^7)^{11 \bmod 18} = 2^5 = 13$ $\longleftarrow$

$Y_3 = (2^5)^{13} = 2^{65 \bmod 18} = 2^{11} = 15$ $\longrightarrow$ $15^5 = (2^{11})^5 = 2 = M$

Numbr of possible secret keys is φ(18) = 18 (1-1/3) (1-1/2) = 6

10

Massey-Omura Lock over GF(2^m)
for: Shamir's 3-Pass Protocol
Secrecy without Authenticity

** Arithmetic in GF(2^m)

User A
E_a = secret key
$D_a = E_a^{-1}$ (mod 2^m -1)
gcd (E_a , 2^m -1) = 1

User B
E_b = secret key
$D_b = E_b^{-1}$ (mod 2^m -1)
gcd (E_b , 2^m -1) = 1

= M $\xrightarrow{1}$ M^{E_a} $\longrightarrow$ $(M^{E_a})^{E_b}$

$\xleftarrow{2}$ $M^{E_a E_b}$

$(M^{E_a E_b})^{D_a}$ $\xrightarrow{3}$ M^{E_b} $\longrightarrow$ $(M^{E_b})^{D_b}$ = M

** For highest security, (2^m -1), need to be a prime, example: **(2^{127} -1) is a** Mersenne prime
# of keys = φ(2^m-1)

11

Example: Massey-Omura Lock for Shamir 3-Pass Protocol

Design a Massey omura Lock over GF(2^3) and exchange the message M=110

1. Select a GF(2^3) generator/modulus P(x)
If P(x)= x^3+ x +1 is the modulus then x^3 + x +1 = 0, thus x^3 = x +1. the exponents of **x** in GF(2^3) are:

x = **x**	010
x^2= **x^2**	100
x^3= **x +1**	011
x^4= **x^2 +x**	110
x^5= x^3 +x^2 = x + 1 + x^2	111
x^6= x^3 + x^2 + x = **x^2 +1**	101
x^7= (x^3 +x)= 1	001

The modulus in the exponent is 2^m-1=2^3-1=7
M=110=(x^4)
M^3=$(x^4)^3$ = x^{12} = x^5 = 111

2. Secret Key selection
gcd(7,3)=1, => 3 is possible to use
gcd(7,6)=1, => 6 is possible to use

3. System setup, transactions and message exchange
GF(2^3) generated by: x^3 + x +1
Modulus in the exponent is 2^3-1=7

USER A
$e_1 = 3$
$d_1 = 3^{-1}$ mod 7 = 5
Message
M = x^4 =110

USER B
$e_2 = 6$
$d_2 = 6^{-1}$ mod 7 = 6

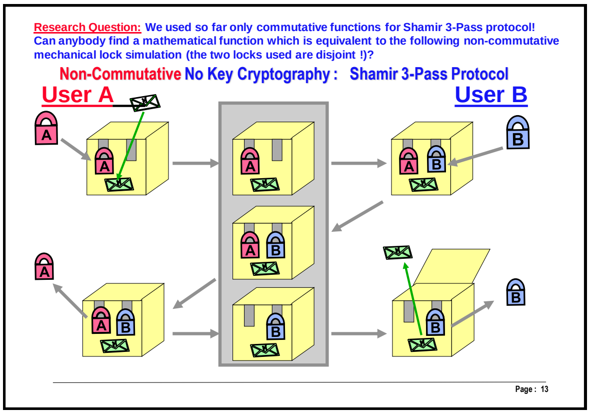
$Y_1 = (M)^3 = x^{4.3 \bmod 7} = x^{12 \bmod 7} = x^5 = (111)$ $\longrightarrow$

$Y_2 = (x^5)^6 = x^{30 \bmod 7} = x^2 = 100$ $\longleftarrow$

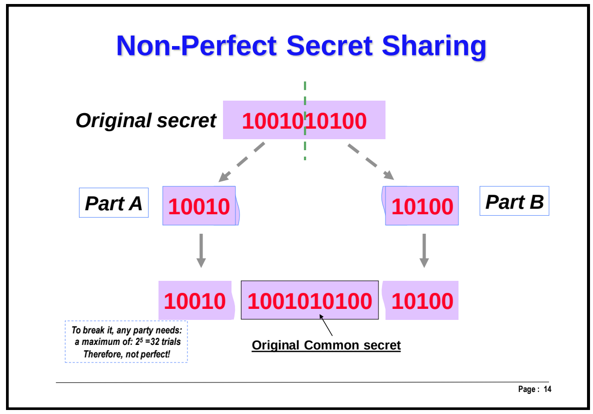
$Y_3 = (x^2)^5 = x^{10} = x^3 = 011$ $\longrightarrow$ $(X^3)^6 = x^{18} = x^4 = 110 = M$

4. Numbr of possible secret keys is φ(7)=(7-1)=6
Which is the number of invertible elements in the exponent!

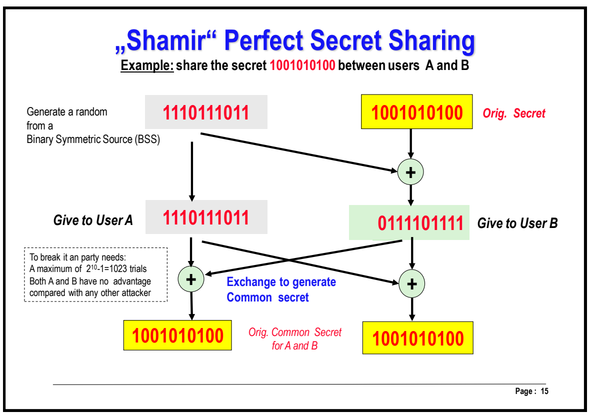
12



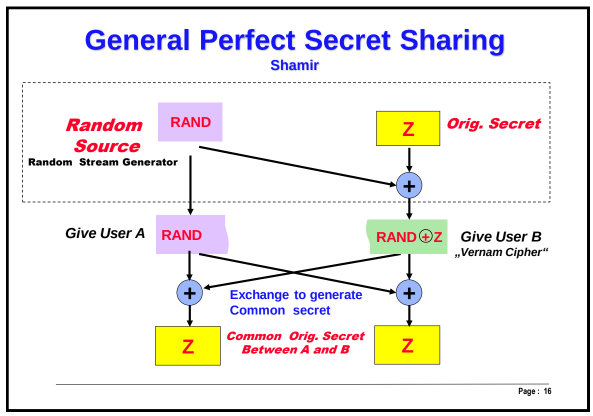
13



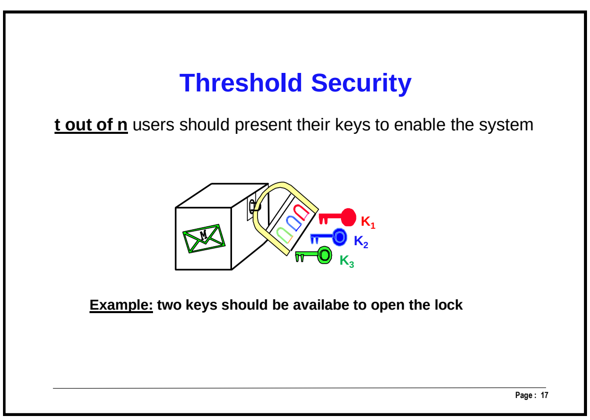
14



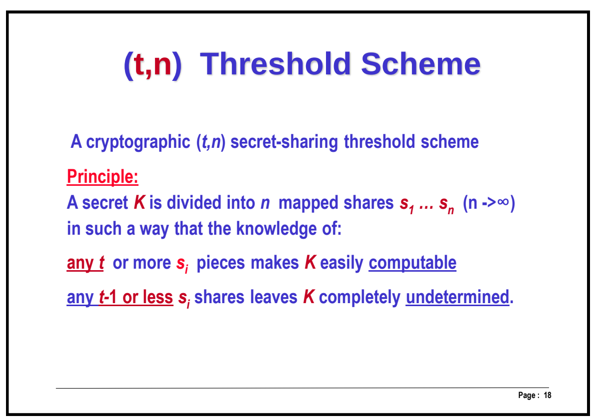
15



16



17



18

Shamir's Threshold Scheme

Basic idea* :

Shamir's t out of n Threshold Scheme is based on the fact that a polynomial y = f(x) of degree (t-1) can only be uniquely defined by at least t points (x_i, y_i) with distinct x_i.

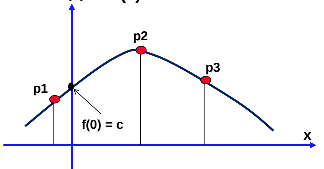
This means that if we have n users each knows only one point on f(x), then any group of at least t users can cooperate to generate the polynomial f(x) as a common secret.

In other words: If less that t users cooperate they would not be able to construct f(x) and share the secret

* (Lagrange Interpolation: A polynomial of degree t-1 can be uniquely interpolated from at least t points).

Basic Concept: Example of Lagrange Interpolation Shamir's Threshold Scheme

f(x) = a x^2 + b x + c



At least 3 points are required to define f(x) with degree 2 !

To find a, b and c
Solving 3 linear equation with 3 unknowns
At least 3 points are necessary

Example p1(-1,2), p2 (3,5), p3 (5,3):

$$\begin{aligned} 2 &= a - b + c \\ 5 &= 9a + 3b + c \\ 3 &= 25a + 5b + c \end{aligned}$$

Solving 3 linear equation having 3 unknowns a,b, and c reveals the curve f(x).
Notice: Less than 3 points are not sufficient!

Shamir's Threshold Scheme set up

System set-up:

n secrets are distributed securely to n users. The (secret distributor), called here Dealer should then perform the following steps:

1. for Threshold =t, choose a polynomial $f(x) = f_0 + f_1 x + f_2 x^2 + \ldots + f_{t-1} x^{t-1}$
   With the secret K = f_0 = f(0), where $f_0 \in GF(p)$, *p is a large prime integer.*
2. The public values x_1 to x_n are selected randomly for n users.
   Dealer then computes the corresponding n shares for n participants
   $S_i = f(x_i)$ and sends securely every share S_i to the corresponding participant P_i.

Revealing the secret K:

The above function f(x) can be reconstructed to get K if at least t participants cooperate and disclose their shares to each other to get K (that is, t-shares (S_i s) need to be disclosed together).

Shamir's Threshold Scheme

Secret reconstruction by t users:

Using *Lagrange interpolation formula*, any t cooperating participants can find the secret K = f(0) = f_0 by *Lagrange Interpolation*:

$$f(x) = \sum_{i=0}^{t-1} S_i\, L_i(x), \quad \text{Where} \quad L_i(x) = \prod_{\substack{j=0 \\ J \neq i}}^{t-1} (x-x_j) / (x_i-x_j)$$

Only t-S_i's [t points on f(x)] are necessary to find K=f(0)

$$\text{that is for } x=0, \quad K = f(0) = \sum_{i=0}^{t-1} S_i\, L_i, \quad \text{where } L_i = \prod_{\substack{j=0 \\ J \neq i}}^{t-1} -x_i / (x_i-x_j)$$

All computations are modulo p (over GF(p)), where p is a large prime.
The system works similarly over GF (2^m).