



# **EXPLORING CRYPTOGRAPHY PROTOCOLS**

WITH LIMITED EMPHASIS ON MATHEMATICS ☺



## ATTENTION

THESE SLIDES HAVE BEEN CRAFTED USING THE FOUNDATION OF MY MSC COURSE IN CRYPTOGRAPHY PROTOCOLS AT THE UNIVERSITY OF ISFAHAN.

I'VE MADE ADJUSTMENTS TO THE CONTENT TO ALIGN WITH THE SPECIFIC OBJECTIVES OF THIS PRESENTATION.

ALSO, MY INTENTION HAS BEEN TO MINIMIZE THE USE OF MATHEMATICAL CONCEPTS, WHICH MAY RESULT IN SOME CONCEPTS BEING SIMPLIFIED OR LESS PRECISE.

# Agenda

1. Identification and Entity Authentications Protocols
2. Zero Knowledge Protocols
3. Key Establishment Protocols
4. Threshold Cryptography and Secret Sharing Protocols
5. Types of Digital Signatures
6. Special Purpose Protocols (like simultaneous contract signing, mental poker, fair exchange)
7. Identity Based Cryptography
8. Secure Multiparty Computations

# Key Distribution

## Types

- **Key Predistribution Schemes:**

- TA passively delivers some unique keys to each entity through a secure channel in advance.
- The majority, if not all, pairs of entities have a common key.

- **Session Key Distribution**

- TA actively delivers session keys to the entities through an interactive protocol upon their request, ensuring secure encryption during the process.
- Each entity has a common key with TA

- **Key Agreement**

- The entities collaboratively generate the session key, utilizing either symmetric or asymmetric encryption.
- No active involvement of the TA is necessary.

# Key Distribution

## Types of Keys

- **Master Key (Long-Lived Key)**

- If it is a symmetric key, it is pre-shared between two entities or between an entity and the TA. A secure channel is required for its distribution.
- An asymmetric key which is associated with a certified public key.

- **Session Key (Short-Lived Key)**

- A asymmetric key is used for encryption or MAC generation.

# Key Distribution

## Some Terms

- **Known Session Key Security**

- If an enemy discovers a session key, he must not be able to find out other session keys.

- **Prefect Forward Secrecy**

- If the master key is revealed, the subsequent session keys must remain undisclosed.

- **Prefect Backward Secrecy**

- If the master key is revealed, the previous session keys must remain undisclosed.

# Key Predistribution Schemes

## Diffie Hellman

$$e_a = g^{d_a} \bmod n$$

$$1 \leq d_a \leq n-1$$

Public key  $e_a$   
Private key  $d_a$



No interaction is required.

Alice and Bob possess each other's certificates.

$$e_b = g^{d_b} \bmod n$$

$$1 \leq d_b \leq n-1$$

Public key  $e_b$   
Private key  $d_b$



$$K_{ab} = (e_b)^{d_a} = (g^{d_b})^{d_a} \bmod n$$

$$K_{ab} = (e_a)^{d_b} = (g^{d_a})^{d_b} \bmod n$$

# Key Predistribution Schemes

## Unconditional Secure – Obvious Scheme

- TA generates a unique common key for each pair of users.
- The key is delivered through a secure channel.

# Key Predistribution Schemes

## Unconditional Secure - Blom Scheme

- K Is the maximum number of long-lived keys which if disclosed, secrecy of other keys would not be threaten.
- Symmetric Polynomials: if any of the variables in a polynomial are interchanged, then we get the same polynomial.
  - For example:  $f(x,y) = f(y,x) = xy + 5(x+y) + 6$

# Key Predistribution Schemes

## Unconditional Secure – Blom Scheme (K=1)

TA

- chooses prime  $P$
- chooses a random  $1 \leq r_i \leq p-1$  and assign it to user  $i$
- makes  $r_i$  public
- chooses random  $1 \leq a, b, c \leq p-1$
- generates private  $f(x, y) = a + b(x+y) + cx^y \bmod p$
- sends for user  $i$  through a secure channel:
  - $f(x, r_i) = a + b(x+r_i) + cxr_i = a + br_i + (b+cr_i)x$

# Key Predistribution Schemes

## Unconditional Secure – Blom Scheme (K=1)

User  $u, v$  want to generate a common key based on

$$f(x, r_u) = a + b(x + r_u) + cxr_u = a + br_u + (b + cr_u)x$$

$$f(x, r_v) = a + b(x + r_v) + cxr_v = a + br_v + (b + cr_v)x$$

### User $u$

- $f(x, r_u) = g_u(x)$
- $g_u(r_v) = f(r_v, r_u)$

### User $v$

- $f(x, r_v) = g_v(x)$
- $g_v(r_u) = f(r_u, r_v)$

# Key Predistribution Schemes

## Unconditional Secure – Blom Scheme (K=1)

- The scheme for K=1 is unconditionally secure against a single user, say w

Prove:

- User w want to find  $K_{uv} = a + br_u + (b+cr_u)r_v$
- User w knows  $r_u$  and  $r_v$  but does not know a, b, c
- User w has:  $f(x, r_w) = a + b(x+r_w) + cxr_w = a + br_w + (b+cr_w)x$

$$\begin{bmatrix} 1 & r_u+r_v & r_ur_v \\ 1 & r_w & 0 \\ 0 & 1 & r_w \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} k_{uv} \\ a_w \\ b_w \end{bmatrix}$$

Impossible!  
3 equations, 4 variables

# Key Predistribution Schemes

## Unconditional Secure – Blom Scheme (K=1)

- The scheme for K=1 is NOT secure against coalition of 2 users
- Coalition of users x and w can calculate common key of each two others.
- User w:
  - $a_w = a + br_w$
  - $b_w = b + cr_w$  **Possible!**
- User x:
  - $a_x = a + br_x$
  - $b_x = b + cr_x$**4 equations, 3 variables**

# Key Predistribution Schemes

## Unconditional Secure – Generalized Blom Scheme

TA

- chooses prime  $P$
- Chooses a random  $1 \leq r_i \leq p-1$  and assign it to user  $i$
- makes  $r_i$  public
- Chooses a random  $0 \leq a_{i,j} \leq p-1$  which  $a_{ij}=a_{ji}$
- Creates  $f(x,y) = \sum_{i=0}^k \sum_{j=0}^k a_{ij}x^i y^j$
- Sends a user, say  $u$ , through a secure channel:
  - $g_u(x) = f(x, r_u) \bmod p = \sum_{i=0}^k a_{ui}x^i$

Each two users can generates their common key by their  $r$  value

# Key Predistribution Schemes

## Unconditional Secure – Generalized Blom Scheme

The generalized scheme is unconditionally secure against  $K$  users.

But, each coalition of  $K+1$  users can break the scheme.

# Session Key Distribution

## Scheme #1

TA

- $n$  Users:  $\{u_1, u_2, \dots, u_n\}$
- $v$  Keys:  $\{k_1, k_2, \dots, k_v\}$
- a  $v \times n$  binary matrix
- $(i, j)$ -entry is 1 if and only if  $k_i$  is assigned to  $u_j$
- If  $p$  is a subset of users:  $\text{keys}(p) = \{\text{indices of common keys between all of } p \text{ users}\}$
- $\text{Keys}(p) \neq \emptyset \Rightarrow \sum_{i \in \text{keys}(p)} k_i$

# Session Key Distribution

## Scheme #1

	$u_1$	$u_2$	$u_3$	$u_4$
$k_1$	1	1	0	0
$k_2$	1	0	1	0
$k_3$	1	0	0	1
$k_4$	0	1	1	0
$k_5$	0	1	0	1
$k_6$	0	0	1	1

$\text{keys}(u_1): \{1,2,3\}$

$\text{keys}(u_2): \{1,4,5\}$

$\Rightarrow \text{keys}(u_1, u_2): \{1\}$

- Only two 1-entry per row (secure against coalition of others)
- Each two users have exactly one common key
- Same as obvious scheme: a common key per two users
- Count of keys:  $\binom{n}{2} = \binom{4}{2} = 6$

# Session Key Distribution

## Scheme #1

$f$  as a subset of users is able to calculate  $k_p$  if and only if

$$\text{Keys}(p) \subseteq \bigcup_{u_j \in f} \text{keys}(uj)$$

Even if one user of  $\text{keys}(p)$  does not get involve in coalition, the key would not be found.

# Session Key Distribution

## Scheme #2

- Number of keys the user would be stored must be as less as possible
- Most of the time it is not needed to secure against coalition of the all users.
- Securing against a specific number (= less than  $n-2$ ) of users is sufficient

# Session Key Distribution

## Scheme #2

	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$	$u_7$
$K_1$	1	1	1	0	1	0	0
$k_2$	0	1	1	1	0	1	0
$k_3$	0	0	1	1	1	0	1
$k_4$	1	0	0	1	1	1	0
$k_5$	0	1	0	0	1	1	1
$k_6$	1	0	1	0	0	1	1
$k_7$	1	1	0	1	0	0	1

$$\text{keys}(u_1, u_2) \overset{p}{\underset{\cup}{\longrightarrow}} \{1, 7\}$$
$$K_p = k_1 \oplus k_7$$

Secure against one, but Not  
against coalition of two

- Each two users can create a common key secured against others
- Number of keys ( $=7$ ) is much less than  $\binom{n}{2} = 21$
- Each user stores 4 keys which is less than  $n - 1 = 6$

# Session Key Distribution

## Fiat–Naor

For each arbitrary subset of users, the key could be calculated and secured against coalition of up to  $w$  users.

Choose  $w$  in range  $[1, n]$

Number of keys:  $V = \sum_{i=0}^w \binom{n}{i}$

Matrix  $M$  which each its row is a  $n$ -bit sequence with at least  $n - w - 1$  1-bit.

## Fiat–Naor W-KDP

# Session Key Distribution

## Fiat–Naor 1–KDP

$$n = 6, w = 1, n - w = 5$$

$$V = \sum_{i=0}^w \binom{n}{i} = \binom{6}{0} + \binom{6}{1} = 7$$

There is no other user has  
 $\{k_1, k_4, k_5, k_6\}$

	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$
$k_1$	1	1	1	1	1	1
$k_2$	1	1	1	1	1	0
$k_3$	1	1	1	1	0	1
$k_4$	1	1	1	0	1	1
$k_5$	1	1	0	1	1	1
$k_6$	1	0	1	1	1	1
$k_7$	0	1	1	1	1	1

$$\text{keys}(u_1, u_5, u_6) : \{1, 4, 5, 6\}$$
$$K_p = k_1 \oplus k_4 \oplus k_5 \oplus k_6$$

# Session Key Distribution

Mitchel–Piper

There is a key per subset with exactly  $t$  users which is secured against up to  $w$  users

$(X, A)$ -pair:

- $X$  is a set contains  $v$  keys.  $X: \{k_1, k_2, \dots, k_v\}$
- $A$  is a subset of  $X$ , called a block.  $A_i$  is the key assigned to  $u_i$  specifies the column  $i$  of matrix  $M$ .

$(X, A)$  is a  $(t, w)$ -CFF (Cover Free Family) if intersection of any  $t$  blocks must not cover union of any  $w$  blocks.

Matrix  $M$  is  $V^*N$  with  $(i, j)$ -entry:  $\begin{cases} 1 & \text{if } k_i \in A_j \\ 0 & \text{otherwise} \end{cases}$

# Session Key Distribution

## Example of Mitchel-Piper

Given  $N=7$  and  $V=7$

$X: \{k_1, k_2, \dots, k_7\}$

$A: \{ A_1: \{k_1, k_4, k_6, k_7\},$

...

$A_7: \{k_3, k_5, k_6, k_7\} \}$

**(2,1)-CFF**

# Session Key Distribution

Mitchel-Piper

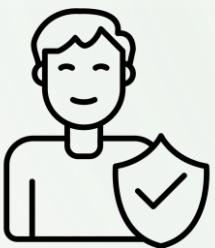
Objective: For given  $t$ ,  $w$  and  $n$ , finding a  $(t, w)$ -CFF( $v, n$ ) with minimal value of  $v$

- ✓ Erdos's estimating algorithm

# Session Key Distribution

Needham-Shroeder (1978)

Session key with Alice:  $K_A$   
Session key with Bob:  $K_B$   
Session key:  $K$   
 $t_B = e_{KB}(K|ID_A)$



$r_A, ID_A, ID_B$

$y_1 = e_{KA}(r_A|ID_B|K|t_B)$



$t_B$

$y_2 = e_K(r_B)$

$y_3 = e_K(r_B-1)$

Bob ensures Alice has the key

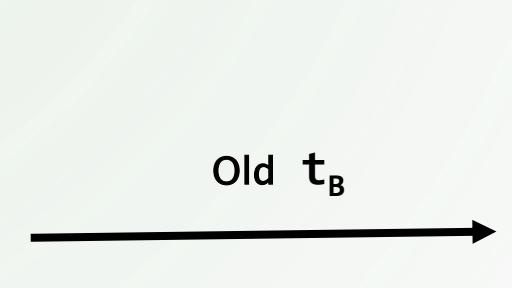
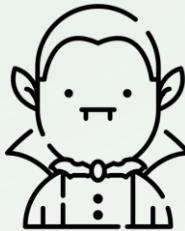
One way key confirmation

# Session Key Distribution

Denning-Succo Attack

Known Session Attack

In role of Alice



# Session Key Distribution

## Simplified Kerberos (80s and 90s)

Session key with Alice:  $K_A$   
Session key with Bob:  $K_B$   
Session key:  $K$   
Validity period:  $L$



$r_A, ID_A, ID_B$

$$y_1 = e_{KA}(r_A | ID_B | K | L)$$

$$t_B = e_{KA}(K | ID_B | L)$$



$$t_B, y_2 = e_K(ID_A | time)$$

$$y_3 = e_K(time+1)$$

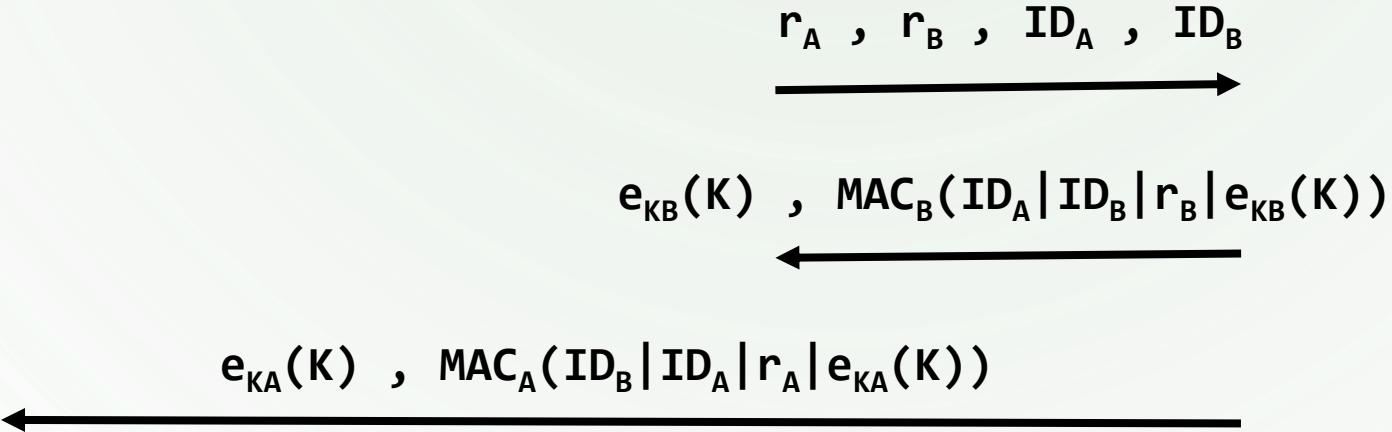
Two way key confirmation

# Session Key Distribution

Bellare–Rogaway (1995)



Session key with Alice:  $K_A$   
Session key with Bob:  $K_B$   
Session key:  $K$



No key confirmation

# Key Agreement

## Diffie Hellman

agreement on:  
multiplicative  
group G of order n,  
with a generator g



a is random from  
range [1, n-1]

$$y^a = g^{ba} \bmod n$$

No Authentication

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

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



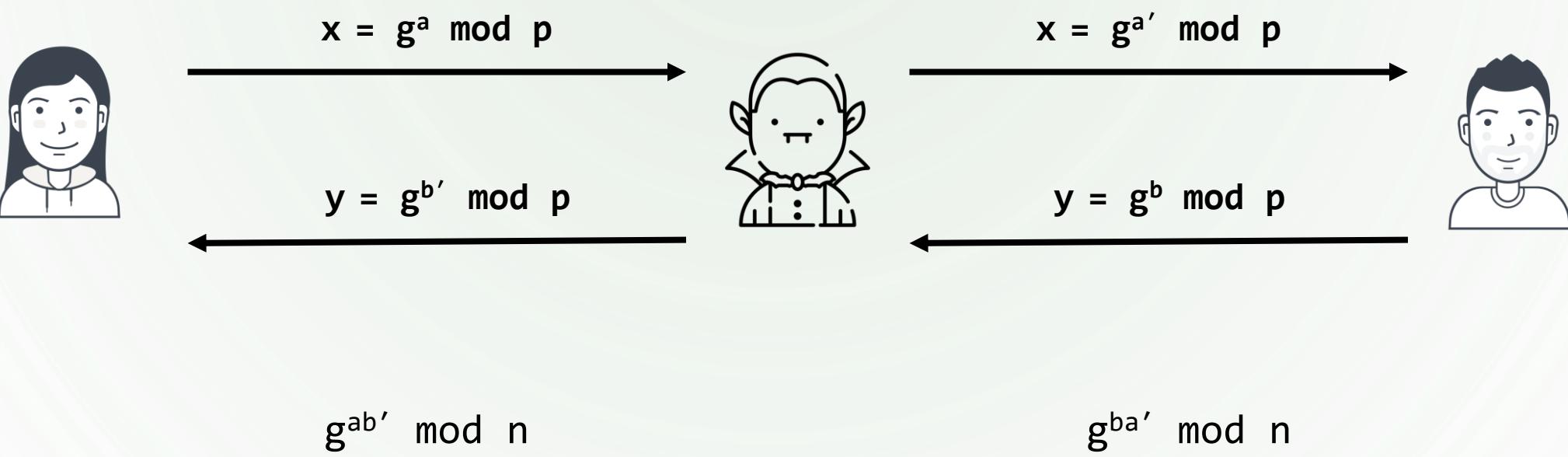
agreement on:  
multiplicative  
group G of order n,  
with a generator g

b is random from  
range [1, n-1]

$$x^b = g^{ab} \bmod n$$

# Key Agreement

Diffie Hellman – MITM Attack



So, we need to authenticate key agreement schemes

# Key Agreement

## Station By Station (STS) Scheme

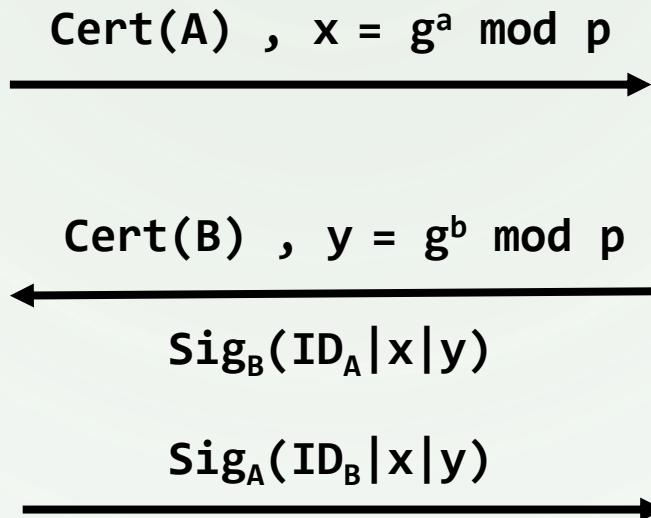
agreement on:  
multiplicative  
group G of order n,  
with a generator g



a is random from  
range [1, n-1]

Verify the sign, then

$$y^a = g^{ba} \bmod n$$



agreement on:  
multiplicative  
group G of order n,  
with a generator g



b is random from  
range [1, n-1]

Verify the sign, then

$$x^b = g^{ab} \bmod n$$

# Key Agreement

## MTI Scheme

agreement on:  
multiplicative  
group G of order q,  
with a generator g

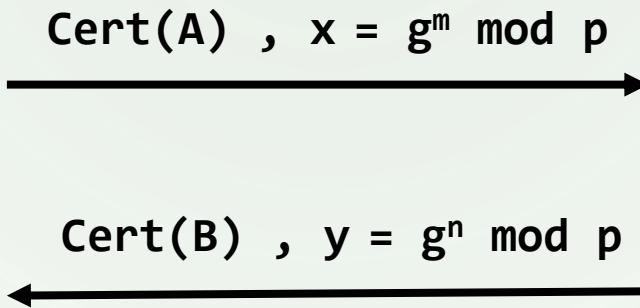


Public Key  
 $P_A = g^a \text{ mod } p$

Private key a is from  
range  $[1, q-1]$

Private m is random  
from range  $[1, q-1]$

$$K = y^a * P_B^m = g^{na+bm} \text{ mod } n$$



agreement on:  
multiplicative  
group G of order n,  
with a generator g

Public Key  
 $P_B = g^b \text{ mod } p$

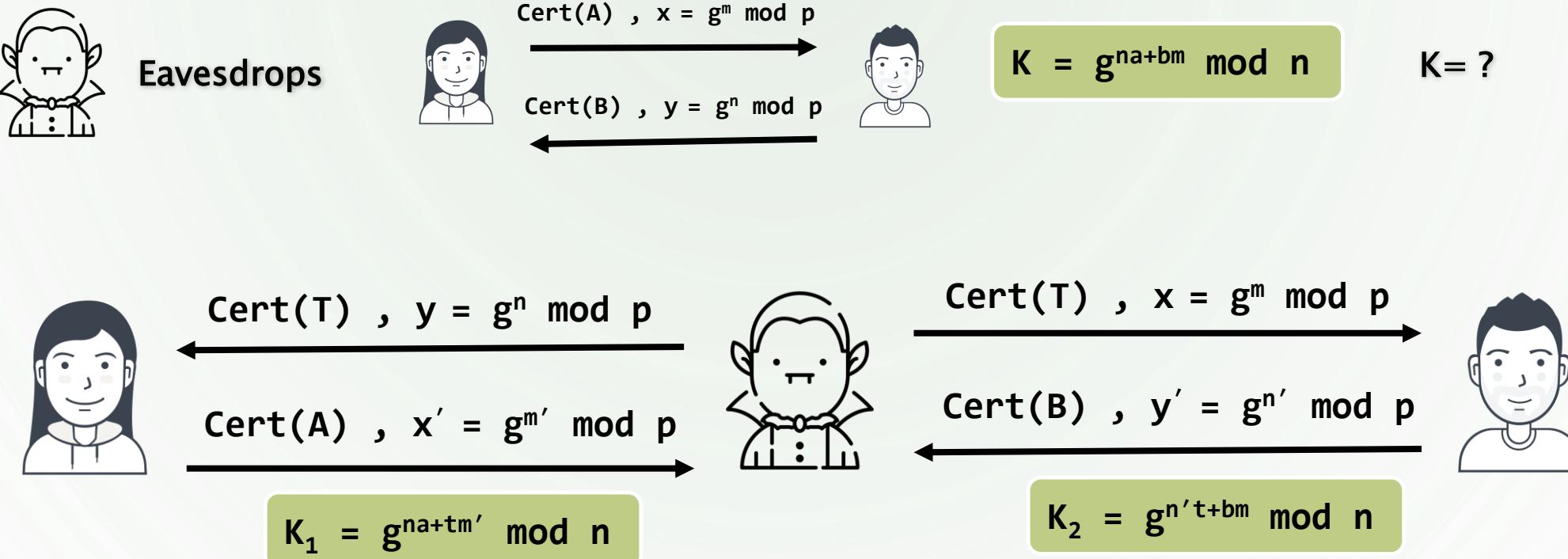
Private key b is from  
range  $[1, q-1]$

Private n is random  
from range  $[1, q-1]$

$$K = x^b * P_A^n = g^{mb+an} \text{ mod } n$$

# Key Agreement

## MTI Scheme – Burmester – Known Session Key Attack



$$\frac{K_1 * K_2}{(x' * y')^t} = \frac{(g^{na} * g^{tm'}) * (g^{n't} * g^{bm})}{g^{m't} * g^{n't}} = g^{na+bm} = K$$

# Key Agreement

## MTI Scheme – Burmester – Countermeasure

Break the symmetry



$\xrightarrow{\text{Cert(A) , } x = g^m \bmod p}$



$\xleftarrow{\text{Cert(B) , } y = g^n \bmod p}$

$$K = y^a * P_B^m = g^{na+bm} \bmod n$$



$$K = h(y^a | P_B^m) = h(g^{na} | g^{bm})$$

$$K = x^b * P_A^n = g^{mb+an} \bmod n$$



$$K = h(P_A^n | x^b) = h(g^{an} | g^{mb})$$

# Key Agreement

## Self Certifying Sign – Givault Scheme

$\phi(n)$  is the number of positive integers less than n that are coprime to n.  
 $\phi(10) = 4$  (1, 3, 7, 9)  
For prime p,  $\phi(p) = p-1$

- No need to certification
- The Public key and ID authenticates each other implicitly.

$$\begin{aligned}n &= pq \\ \text{Public key } e \\ \text{Private key } d &= e^{-1} \bmod \phi(n)\end{aligned}$$

Random a  
Identification  $ID_A$   
Self Certifying  $P_A$



Random b  
Identification  $ID_B$   
Self Certifying  $P_B$



$$A = P_A^e + ID_A \bmod n$$

$$B = P_B^e + ID_B \bmod n$$

Random u



$$\begin{aligned}ID_A, P_A, x &= g^u \bmod n \\ ID_B, P_B, y &= g^v \bmod n\end{aligned}$$



Random v

$$K = y^a * (P_B^e + ID_B)^u = g^{va+bu}$$

$$K = x^b * (P_A^e + ID_A)^v = g^{ub+av}$$

# Key Agreement

Group Key Agreement – burmester–desmedt (1994)

$x_1, x_2, \dots, x_m$

$$x_i = \left(\frac{b_{i+1}}{b_{i-1}}\right)^a a_i$$

Random  $a_i$

$$b_i = g^{a_i} \bmod n$$

$$\begin{aligned} K &= (b_{i-1})^{m-a_i} (x_i)^{m-1} (x_{i+1})^{m-2} \dots (x_{i-2})^1 \\ &= g^{\sum_{i < j} a_i a_j} \end{aligned}$$

$b_{i-1}, b_{i+1}$

$u_i$

$U_{i-1}$

m users

$u_{i+1}$

$$x_{i-1} = \left(\frac{b_i}{b_{i-2}}\right)^a a_{i-1}$$

$$\begin{aligned} \text{Random } a_{i-1} \\ b_{i-1} = g^{a_{i-1}} \bmod n \end{aligned}$$

$b_{i-2}, b_i$

$x_1, x_2, \dots, x_m$

$$x_{i+1} = \left(\frac{b_{i+2}}{b_i}\right)^a a_{i+1}$$

$$\begin{aligned} \text{Random } a_{i+1} \\ b_{i+1} = g^{a_{i+1}} \bmod n \end{aligned}$$

$b_i, b_{i+2}$

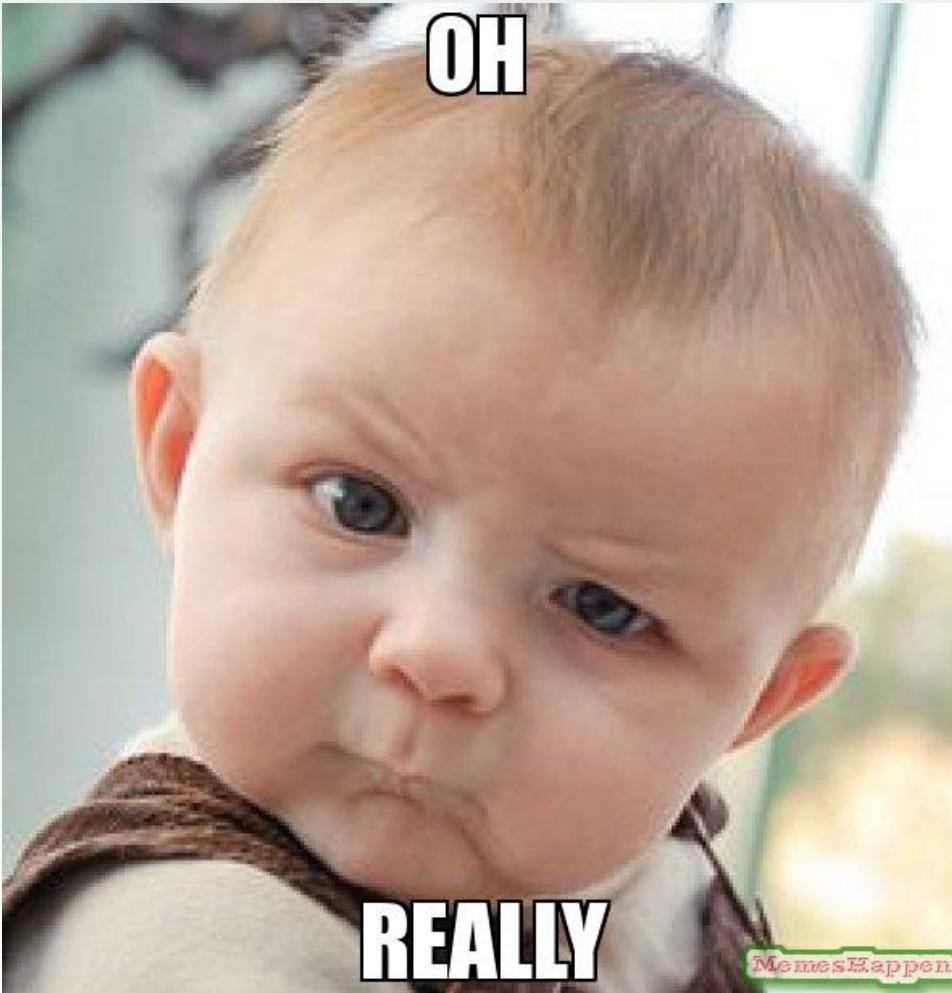
$x_1, x_2, \dots, x_m$

$x_1, x_2, \dots, x_m$

$x_1, x_2, \dots, x_m$

# Key Agreement

Group Key Agreement – burmester-desmedt (1994)



# Key Agreement

## Group Key Agreement – burmester–desmedt (1994)

$$K = (b_{i-1})^{m*a_i} (x_i)^{m-1} (x_{i+1})^{m-2} \dots (x_{i-2})^1$$
$$= g^{\sum_{i < j} a_i a_j}$$

$$K = g^{\sum_{i < j} a_i a_j} = g^{a_1 a_2 + a_2 a_3 + a_1 a_3}$$

$x_1, x_3$

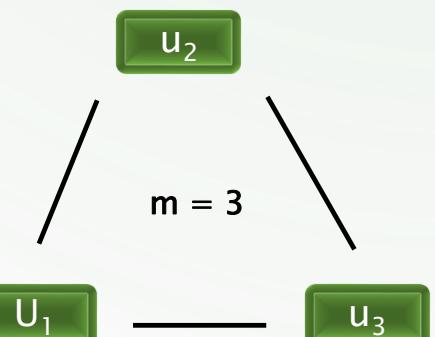
$$x_2 = \left(\frac{b_3}{b_1}\right)^\wedge a_2$$

**Random  $a_2$**   
 $b_2 = g^{a_2} \bmod n$

$b_1, b_3$

$$x_1 = \left(\frac{b_2}{b_3}\right)^\wedge a_1$$

**Random  $a_1$**   
 $b_1 = g^{a_1} \bmod n$   
 $b_2, b_3$



$$K_1 = (b_3)^{3a_1} (x_1)^2 (x_2)^1 = (b_3)^{3a_1} \left(\frac{b_2}{b_3}\right)^{2a_1} \left(\frac{b_3}{b_1}\right)^{1a_2}$$
$$= (b_3)^{a_1+a_2} (b_2)^{2a_1} (b_1)^{-a_2} = (g^{a_3})^{a_1+a_2} (g^{a_2})^{2a_1} (g^{a_1})^{-a_2}$$

**Random  $a_3$**   
 $b_3 = g^{a_3} \bmod n$   
 $b_1, b_2$

$$x_3 = \left(\frac{b_1}{b_2}\right)^\wedge a_3$$

$x_1, x_2$

# Key Agreement

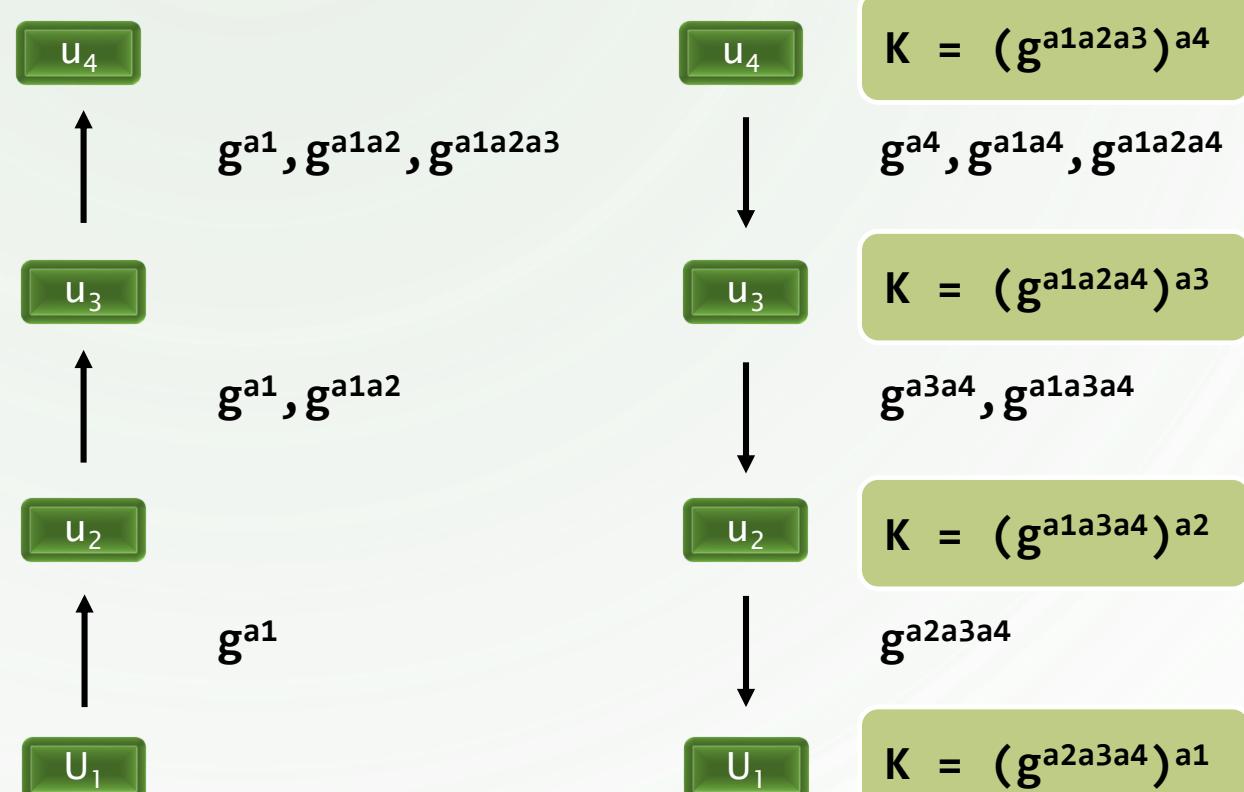
## Group Key Agreement – Steiner

Multiplicative group  $G$   
with a generator  $g$   
with order  $n$   
For  $n$  users

For authentication, each user must sign his message and pass his certificate

This scheme is generalized form of Diffie–Hellman

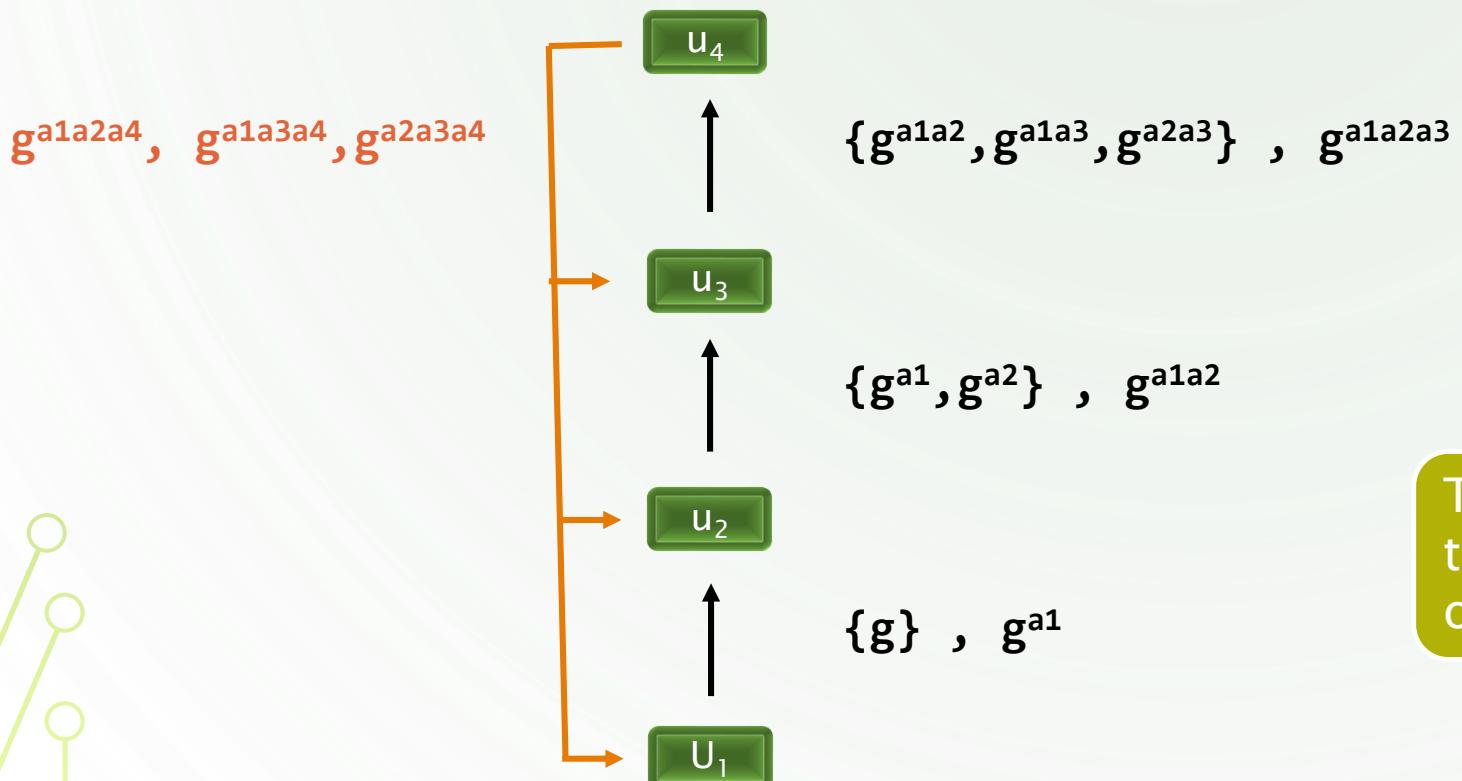
Description for  
Number of users  $n = 4$



# Key Agreement

## Group Key Agreement - Steiner #2 (Cliques)

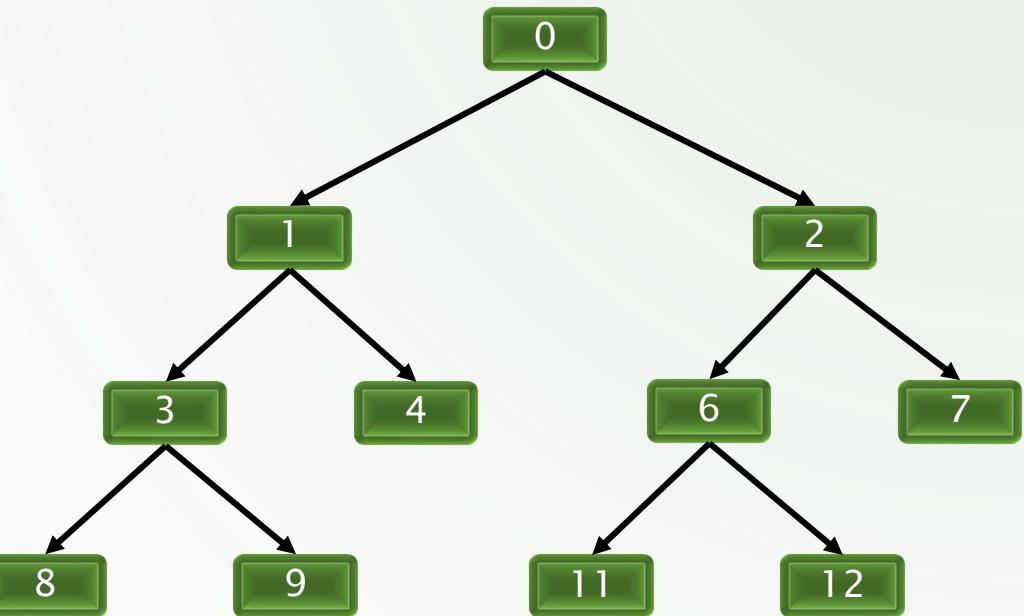
Description for  
Number of users n = 4



This scheme supports  
the addition and removal  
of users from the group

# Key Agreement

## Group Key Agreement - Tree Based Group Diffie Hellman



Blinded Public Key ( $BK_v$ )  
Private Key ( $K_v$ )

Each user knows private key of his parents

Each user knows public key of all others

Common Key of All Users =  
Private Key of the Root

# Key Agreement

## Group Key Agreement - Tree Based Group Diffie Hellman

$$K_v = BK_a^{K_b} = g^{K_a K_b}$$
$$K_v = BK_b^{K_a} = g^{K_b K_a}$$

$$V$$

$$BK_v = g^{g^{K_a K_b}}$$

$$K_a$$
$$BK_a = g^{K_a}$$

$$2V+1$$
  
(a)

$$2V+2$$
  
(b)

$$K_b$$
$$BK_b = g^{K_b}$$

Private key of each non-leaf node would be generated by public key of one child and private key of another.

By continuing this process, each user can generate the private key of the root.

# Ref

1. Cryptography Protocols Course, Dr. Hamid Mala, University of Isfahan
2. [https://www.math.union.edu/~hatleyj/student\\_theses/kender.pdf](https://www.math.union.edu/~hatleyj/student_theses/kender.pdf)
3. <https://www.iconfinder.com/UsersInsights>
4. <https://www.iconfinder.com/Chanut-is>
5. <https://www.iconfinder.com/iconsets/softwaredemo>