

# Asymmetric Cryptography and Key Management

## **Diffie-Hellman Key Exchange**

Sang-Yoon Chang, Ph.D.

# **Module: Diffie-Hellman Key Exchange**

Discrete logarithm problem

Diffie-Hellman Key Exchange

Man-in-the-Middle Attack



# Ordinary Logarithm

$$y = a^b$$

$$\Leftrightarrow b = \log_a y$$

## Discrete Logarithm

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

$$\Leftrightarrow b = \text{dlog}_{a,p} y$$

$b$  is called the discrete logarithm of  $y$   
base  $a$  mod  $p$

## Discrete Logarithm

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

$$\Leftrightarrow b = \text{dlog}_{a,p} y$$

$b$  is called the discrete logarithm of  $y$   
base  $a$  mod  $p$

When does discrete logarithm  $b$  exist  
and is unique?

# Discrete Logarithm

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

$$\Leftrightarrow b = \text{dlog}_{a,p} y$$

Given that  $p$  is prime,  
 $b$  exists and is unique when  
 $a$  is a primitive root of  $p$ , i.e.,  
 $a^1, \dots, a^{p-1} \pmod{p}$  produce distinct  
integers between  $1, \dots, p-1$

## Discrete Logarithm

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

$$\Leftrightarrow b = \text{dlog}_{a,p} y$$

Given that  $p$  is prime,  
 $b$  exists and is unique when  
 $a$  is a primitive root of  $p$ , i.e.,  
 $a^1, \dots, a^{p-1} \pmod{p}$  produce distinct  
integers between  $1, \dots, p-1$





# Discrete Logarithm

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

$$\Leftrightarrow b = \text{dlog}_{a,p} y$$

Given that  $p$  is prime,  
 $b$  exists and is unique when  
 $a$  is a primitive root of  $p$ , i.e.,

$a^1, \dots, a^{p-1} \pmod{p}$  produce distinct  
integers between  $1, \dots, p-1$

## Primitive Root of a Prime Number

For modulus  $p=5$

$a$	$a^2$	$a^3$	$a^4$	$(\text{mod } p)$
-----	-------	-------	-------	-------------------

1				
---	--	--	--	--

2				
---	--	--	--	--

3				
---	--	--	--	--

4				
---	--	--	--	--

## Primitive Root of a Prime Number

For modulus  $p=5$

$a$	$a^2$	$a^3$	$a^4$	$(\text{mod } p)$
1	1	1	1	
2				
3				
4				

## Primitive Root of a Prime Number

For modulus  $p=5$

$a$	$a^2$	$a^3$	$a^4$	$(\text{mod } p)$
1	1	1	1	
2	4	3		
3				
4				

## Primitive Root of a Prime Number

For modulus  $p=5$

$a$	$a^2$	$a^3$	$a^4$	$(\text{mod } p)$
1	1	1	1	
2	4	3	1	
3				
4				

## Primitive Root of a Prime Number

For modulus  $p=5$

$a$	$a^2$	$a^3$	$a^4$	$(\text{mod } p)$
1	1	1	1	
2	4	3	1	2
3				
4				

## Primitive Root of a Prime Number

For modulus  $p=5$

$a$	$a^2$	$a^3$	$a^4$	(mod $p$ )
1	1	1	1	
2	4	3	1	
3	4	2	1	
4	1	4	1	



# Primitive Root of a Prime Number

For modulus  $p=5$

$a$	$a^2$	$a^3$	$a^4$	$(\text{mod } p)$
1	1	1	1	
2	4	3	1	
3	4	2	1	
4	1	4	1	

# Primitive Root of a Prime Number

For modulus  $p=5$

$a$	$a^2$	$a^3$	$a^4$	$(\text{mod } p)$
1	1	1	1	
2	4	3	1	
3	4	2	1	
4	1	4	1	

# Primitive Root of a Prime Number

For modulus  $p=5$

$a$	$a^2$	$a^3$	$a^4$	$(\text{mod } p)$
-----	-------	-------	-------	-------------------

1	1	1	1	
---	---	---	---	--

2	4	3	1	
---	---	---	---	--

3	4	2	1	
---	---	---	---	--

4	1	4	1	
---	---	---	---	--

## Primitive Root of a Prime Number

For modulus  $p=5$

$a$	$a^2$	$a^3$	$a^4$	$(\text{mod } p)$
-----	-------	-------	-------	-------------------

1	1	1	1	
---	---	---	---	--

2	4	3	1	
---	---	---	---	--

3	4	2	1	
---	---	---	---	--

4	1	4	1	
---	---	---	---	--

2 is primitive root 5

and so is 3

## Primitive Root of a Prime Number

For modulus  $p=5$

$a$	$a^2$	$a^3$	$a^4$	(mod $p$ )
-----	-------	-------	-------	------------

1	1	1	1
---	---	---	---

2	4	3	1
---	---	---	---

3	4	2	1
---	---	---	---

4	1	4	1
---	---	---	---

2 is primitive root 5

and so is 3

$\Rightarrow \text{dlog}_{a,p} y$  unique

# Primitive Root of a Prime Number

For modulus  $p=5$

$a$	$a^2$	$a^3$	$a^4$	$(\text{mod } p)$
1	1	1	1	
2	4	3	1	2 is primitive root 5
3	4	2	1	and so is 3
4	1	4	1	$\Rightarrow \text{dlog}_{a,p} y$ unique

$2 = \text{dlog}_{3,5} 4$



## Discrete Logarithm Problem

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

$$\Leftrightarrow b = \text{dlog}_{a,p} y$$

If  $a$  is a primitive root of  $p$ ,  
then  $\text{dlog}_{a,p} y$  exist and unique



## Discrete Logarithm Problem

$y = a^b \bmod p$  Easy

$\Leftrightarrow b = \text{dlog}_{a,p} y$  Difficult

If  $a$  is a primitive root of  $p$ ,  
then  $\text{dlog}_{a,p} y$  exist and unique

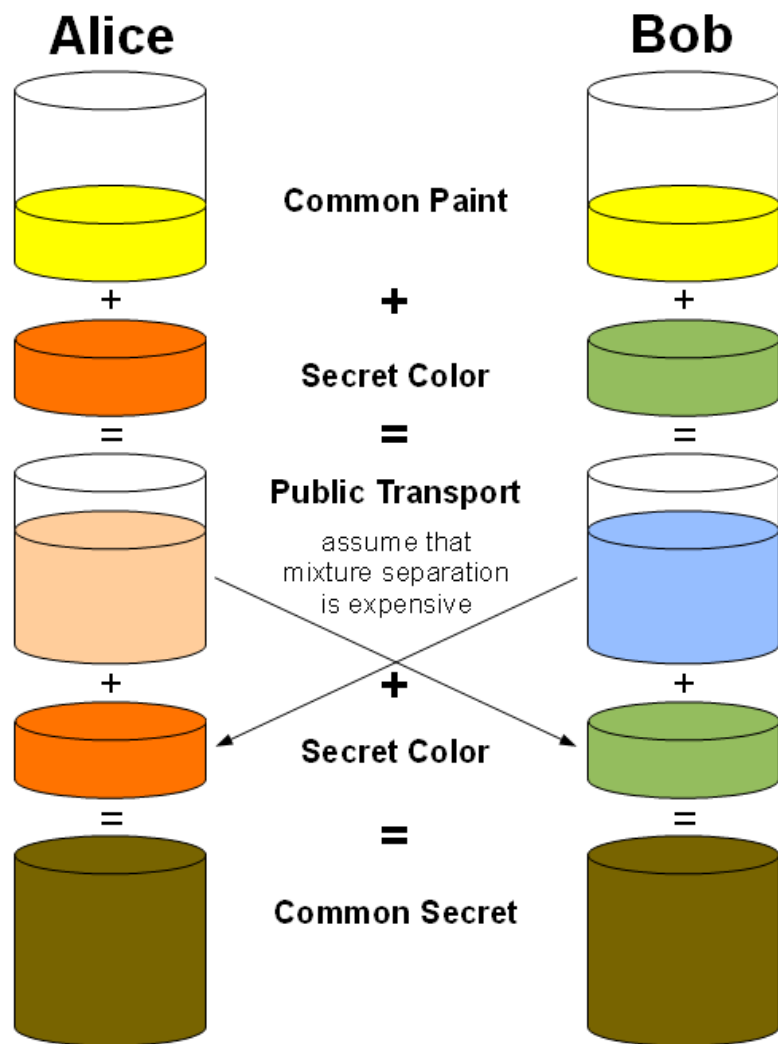


# Diffie-Hellman Key Exchange

The first published asymmetric algorithm

Practical method to exchange secret  
key over public channel

Security relies on Discrete Log Problem



## Diffie-Hellman Key Exchange Setup

Alice and Bob want to exchange secret key

They agree on the global parameters:  $p$ ,  $a$

Each user randomly selects  $X < p$ , and computes  $Y = a^X \bmod p$

$X$  is private and  $Y$  is public, i.e.,  
 $\{X_A, Y_A\}$  for Alice and  $\{X_B, Y_B\}$  for Bob



# Alice

Randomly select  $X_A < p$   
Compute  $Y_A = a^{X_A} \bmod p$

# Bob

Randomly select  $X_B < p$   
Compute  $Y_B = a^{X_B} \bmod p$

**Alice**

**Bob**

Randomly select  $X_A < p$   
Compute  $Y_A = a^{X_A} \bmod p$

Randomly select  $X_B < p$   
Compute  $Y_B = a^{X_B} \bmod p$

Send  $Y_A$  to Bob ----->

<----- Send  $Y_B$  to Alice



**Alice**

**Bob**

Randomly select  $X_A < p$   
Compute  $Y_A = a^{X_A} \bmod p$

Randomly select  $X_B < p$   
Compute  $Y_B = a^{X_B} \bmod p$

Send  $Y_A$  to Bob ----->

<----- Send  $Y_B$  to Alice

Compute  $K = Y_B^{X_A} \bmod p$

Compute  $K = Y_A^{X_B} \bmod p$

# Alice

# Bob

Randomly select  $X_A < p$   
Compute  $Y_A = a^{X_A} \bmod p$



$a, p$

Randomly select  $X_B < p$   
Compute  $Y_B = a^{X_B} \bmod p$

$Y_A, Y_B$

Send  $Y_A$  to Bob



Send  $Y_B$  to Alice

Compute  $K = Y_B^{X_A} \bmod p$

Compute  $K = Y_A^{X_B} \bmod p$

# Alice

# Bob

Randomly select  $x_A < p$   
Compute  $Y_A = a^{x_A} \bmod p$



$a, p$

Send  $Y_A$  to Bob



Randomly select  $x_B < p$   
Compute  $Y_B = a^{x_B} \bmod p$

----->

<----- Send  $Y_B$  to Alice

Compute  $K = Y_B^{x_A} \bmod p$

Compute  $K = Y_A^{x_B} \bmod p$

# Alice

# Bob

Randomly select  $X_A < p$   
Compute  $Y_A = a^{X_A} \bmod p$  ← D. log problem

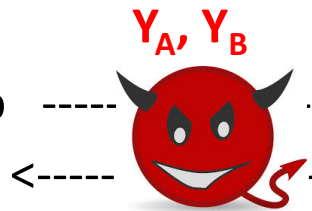


$a, p$

D. log prob.

Randomly select  $X_B < p$   
Compute  $Y_B = a^{X_B} \bmod p$

Send  $Y_A$  to Bob



<-----

----->

Send  $Y_B$  to Alice

Compute  $K = Y_B^{X_A} \bmod p$

Compute  $K = Y_A^{X_B} \bmod p$

# Alice

# Bob

Randomly select  $X_A < p$   
Compute  $Y_A = a^{X_A} \bmod p$  ← D. log problem



$a, p$

D. log prob.

Randomly select  $X_B < p$   
Compute  $Y_B = a^{X_B} \bmod p$

Send  $Y_A$  to Bob



<-----

----->

Send  $Y_B$  to Alice

Compute  $K = Y_B^{X_A} \bmod p$

Compute  $K = Y_A^{X_B} \bmod p$

Since  $X_A, X_B$  are secret,  $K$  is also secret

**Alice**

**Bob**

Randomly select  $X_A < p$   
Compute  $Y_A = a^{X_A} \bmod p$

Randomly select  $X_B < p$   
Compute  $Y_B = a^{X_B} \bmod p$

Send  $Y_A$  to Bob ----->

<----- Send  $Y_B$  to Alice

Compute  $K = Y_B^{X_A} \bmod p$

Compute  $K = Y_A^{X_B} \bmod p$

**K is the secret key for Alice and Bob:**

$$\begin{aligned} K &= Y_B^{x_A} \bmod q && // \text{ A can compute} \\ &= (a^{x_B} \bmod q)^{x_A} \bmod q \\ &= (a^{x_B})^{x_A} \bmod q \\ &= a^{x_B x_A} \bmod q \\ &= (a^{x_A})^{x_B} \bmod q \\ &= (a^{x_A} \bmod q)^{x_B} \bmod q \\ &= Y_A^{x_B} \bmod q && // \text{ B can compute} \end{aligned}$$





# Man-in-the-Middle Attack

Randomly select  $X_A < p$   
Compute  $Y_A = a^{X_A} \bmod p$



$a, p$

Randomly select  $X_B < p$   
Compute  $Y_B = a^{X_B} \bmod p$

$Y_A, Y_B$

Send  $Y_A$  to Bob



Send  $Y_B$  to Alice

Compute  $K = Y_B^{X_A} \bmod p$

Compute  $K = Y_A^{X_B} \bmod p$

# Man-in-the-Middle Attack

Randomly select  $X_A < p$   
Compute  $Y_A = a^{X_A} \bmod p$



Randomly select  $X_B < p$   
Compute  $Y_B = a^{X_B} \bmod p$

Send  $Y_A$  to Bob -----> Receive  $Y_A$ ; Send  $Y_{M1}$  ----->  
<--- Send  $Y_{M2}$ ; Receive  $Y_A$ ; <--- Send  $Y_B$  to Alice

Compute  $K = Y_B^{X_A} \bmod p$

Compute  $K = Y_A^{X_B} \bmod p$

# Man-in-the-Middle Attack

Randomly select  $X_A < p$   
Compute  $Y_A = a^{X_A} \bmod p$



Randomly select  $X_B < p$   
Compute  $Y_B = a^{X_B} \bmod p$

Send  $Y_A$  to Bob -----> Receive  $Y_A$ ; Send  $Y_{M1}$  ----->  
<--- Send  $Y_{M2}$ ; Receive  $Y_A$ ; <--- Send  $Y_B$  to Alice

Compute  $K_2 = Y_{M2}^{X_A} \bmod p$       Compute  $K_1 = Y_{M1}^{X_B} \bmod p$

# Man-in-the-Middle Attack

Randomly select  $X_A < p$   
Compute  $Y_A = a^{X_A} \bmod p$



Randomly select  $X_B < p$   
Compute  $Y_B = a^{X_B} \bmod p$

Send  $Y_A$  to Bob -----> Receive  $Y_A$ ; Send  $Y_{M1}$  ----->  
<--- Send  $Y_{M2}$ ; Receive  $Y_B$ ; <--- Send  $Y_B$  to Alice

Compute  $K_2 = Y_{M2}^{X_A} \bmod p$       Compute  $K_1 = Y_{M1}^{X_B} \bmod p$   
Knows  $Y_A$  and can compute  $K_2$

# Man-in-the-Middle Attack

Randomly select  $X_A < p$   
Compute  $Y_A = a^{X_A} \bmod p$



Randomly select  $X_B < p$   
Compute  $Y_B = a^{X_B} \bmod p$

Send  $Y_A$  to Bob -----> Receive  $Y_A$ ; Send  $Y_{M1}$  ----->  
<--- Send  $Y_{M2}$ ; Receive  $Y_B$ ; <--- Send  $Y_B$  to Alice

Compute  $K_2 = Y_{M2}^{X_A} \bmod p$       Compute  $K_1 = Y_{M1}^{X_B} \bmod p$   
Knows  $Y_A$  and can compute  $K_2$       Knows  $Y_B$  and can compute  $K_1$

# Man-in-the-Middle Attack

Randomly select  $X_A < p$   
Compute  $Y_A = a^{X_A} \bmod p$



Randomly select  $X_B < p$   
Compute  $Y_B = a^{X_B} \bmod p$

Send  $Y_A$  to Bob -----> Receive  $Y_A$ ; Send  $Y_{M1}$  ----->  
<--- Send  $Y_{M2}$ ; Receive  $Y_B$ ; <--- Send  $Y_B$  to Alice

Compute $K_2 = Y_{M2}^{X_A} \bmod p$	Compute $K_1 = Y_{M1}^{X_B} \bmod p$
Knows $Y_A$ and can compute $K_2$	Knows $Y_B$ and can compute $K_1$
Alice uses $K_2$	Bob uses $K_1$

## **Man-in-the-Middle Attack Countermeasure**

Vulnerable because no authentication

Authenticate Alice and Bob, e.g.,  
certificates and digital signatures





## El Gamal Encryption

El Gamal encryption related to D.-H.:

- Relies on Discrete Log problem
- Use exponentiation

Sends one-time key with the message

Used in Digital Signature Standards

