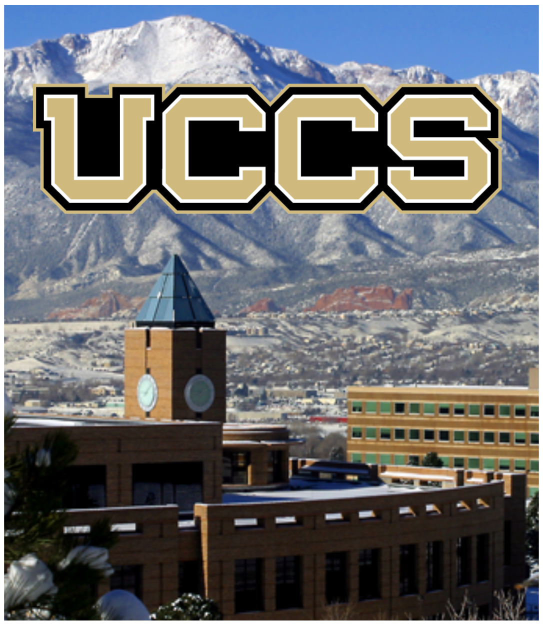


Asymmetric Cryptography and Key Management

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

$y = a^b \bmod p$

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

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

Discrete Logarithm

$y = a^b \bmod p$

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

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

When does discrete logarithm b exist and is unique?

Discrete Logarithm

$y = a^b \bmod p$
$\Leftrightarrow$ $b = \mathrm{dlog}_{a,p} y$

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

Discrete Logarithm

$y = a^b \bmod p$

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

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

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

Discrete Logarithm

$y = a^b \bmod p$

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

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

Primitive Root of a Prime Number

For modulus p=5

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

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	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	(mod p)
$\boxed{1}$	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	(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	(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	(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	

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	2 is primitive root 5
3	4	2	1	and so is 3
4	1	4	1	=> $\mathrm{dlog}_{a,p} y$ unique

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	2 is primitive root 5
3	4	2	1	and so is 3
4	1	4	1	=> $\mathrm{dlog}_{a,p} y$ unique

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

Discrete Logarithm Problem

$y = a^b \bmod p$

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

If a is a primitive root of p,

then $\mathrm{dlog}_{a,p} y$ exist and unique

Discrete Logarithm Problem

$y = a^b \bmod p$ Easy

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

If a is a primitive root of p,

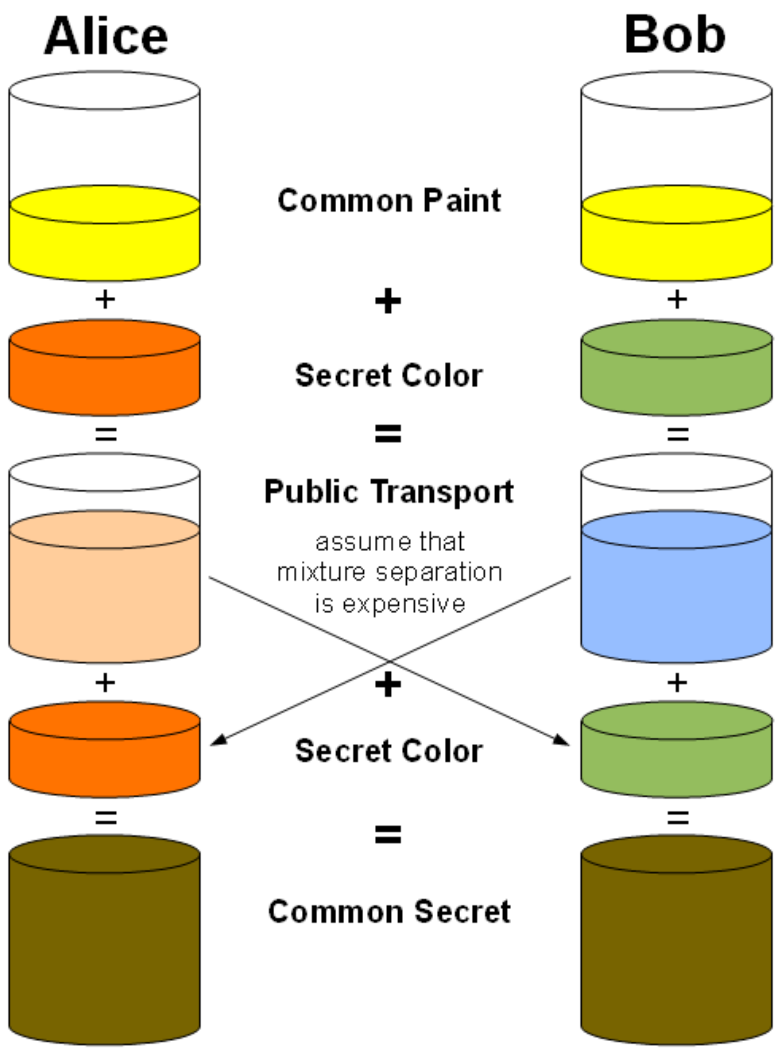
then $\mathrm{dlog}_{a,p} y$ exist and unique

Diffie-Hellman Key Exchange

The first published asymmetric algorithm

Practical method to exchange secret
key over public channel

Security relies on Discrete Log Problem


Alice
Bob
Common Paint
+
Secret Color
=
Public Transport
assume that
mixture separation
is expensive
+
Secret Color
=
Common Secret

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 **Bob**

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

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

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

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 $Y_A = a^{X_A} \bmod p$ ← **D. log problem**

Randomly select X_B < p

a, p **D. log prob.** → 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 $Y_A = a^{X_A} \bmod p$ ← **D. log problem**

Randomly select X_B < p

a, p   **D. log prob. →** 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 $K = Y_A^{X_B} \bmod p$

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

$\boxed{\text{Compute } K = Y_B^{X_A} \bmod p \qquad \text{Compute } K = Y_A^{X_B} \bmod p}$

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 $\mathbf{Y_A}$; Send $\mathbf{Y_{M1}}$ ---->

<--- Send $\mathbf{Y_{M2}}$; Receive $\mathbf{Y_A}$; <--- 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 $\mathbf{Y_A}$; Send $\mathbf{Y_{M1}}$ ---->

<--- Send $\mathbf{Y_{M2}}$; Receive $\mathbf{Y_A}$; <--- 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 $\mathbf{Y_A}$; Send $\mathbf{Y_{M1}}$ ---->

<--- Send $\mathbf{Y_{M2}}$; Receive $\mathbf{Y_B}$; <--- Send Y_B to Alice

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

Knows $\mathbf{Y_A}$ 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 $\mathbf{Y_A}$; Send $\mathbf{Y_{M1}}$ ---->

<--- Send $\mathbf{Y_{M2}}$; Receive $\mathbf{Y_B}$; <--- Send Y_B to Alice

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

Knows $\mathbf{Y_A}$ and can compute K_2

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

Knows $\mathbf{Y_B}$ and can compute K_1

Man-in-the-Middle Attack

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

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

Send Y_A to Bob ----> Receive $\mathbf{Y_A}$; Send $\mathbf{Y_{M1}}$ ---->

<--- Send $\mathbf{Y_{M2}}$; Receive $\mathbf{Y_B}$; <--- Send Y_B to Alice

Compute $K_2 = Y_{M2}^{X_A} \mod p$

Knows $\mathbf{Y_A}$ and can compute K_2
Alice uses K_2

Compute $K_1 = Y_{M1}^{X_B} \mod p$

Knows $\mathbf{Y_B}$ and can compute K_1
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