$\boxed{N(N-1)}$ ✓

$A \xrightarrow{k_1} B$

$\xleftarrow[k_2]{}$

1 Million - 1 trillion - 10^{12} keys

✓ Alice $\longrightarrow$ $\boxed{\text{Key Distribution Center}}$ = ✓ $\longrightarrow$ Bob

Key Distribution Centre (KDC) : Refer. C&NS by Forouzan.

Diffie - Hellman Key Exchange

Alice

Public Parameter	p - Large prime
	$\alpha \in \mathbb{Z}_p^*$
α is a generator of $\mathbb{Z}_p^*$	

Bob

$a \in K_{prA} \in \{2, 3, ---, p-2\}$

$b \in K_{prB} \in \{2, 3, ---, p-2\}$

$A = \alpha^a \bmod p \longrightarrow A$

$B \longleftarrow B = \alpha^b \bmod p$

$K_{AB} = B^a \bmod p$

$= (\alpha^b)^a \bmod p$

$\boxed{K_{AB} = \alpha^{ab} \bmod p}$

$K_{AB} = A^b \bmod p = (\alpha^a)^b \bmod p$

$\boxed{K_{AB} = \alpha^{ab} \bmod p}$

$K_{AB} = \alpha^{ab} \bmod p$ is the session key.

Recall: A cyclic group is a group in which there is an element which generates the whole.

$(G, *)$ is a cyclic group $\Rightarrow$ $\exists$ an elmt $g \in G$ s.t

$G = \{g^n \mid n \in \mathbb{Z}\} = \{\underbrace{g * g * --- * g}_{n\text{-times}} \mid n \in \mathbb{N}$

Such an element g in a cyclic grp $(G, *)$ is called primitive element or a generator of $(G, *)$

Ex: $(\mathbb{Z}_n, +_n)$ - Cyclic group.

g is a generator of $(\mathbb{Z}_n, +_n)$ if $\gcd(g, n) = 1$

$(\mathbb{Z}_n^*, \times_n)$ is cyclic ?

Theorem: $(\mathbb{Z}_p^*, \times_n)$ is a cyclic group for all prime p.

$$(\mathbb{Z}_7^*, \times) = \{1, 2, 3, 4, 5, 6\}$$

$2^1 = 2$
$2^2 = 4$
$2^{\boxed{3}} = \boxed{1}$
$2^4 = 2$

$O(2) = 3$

$3^1 = 3$
$3^2 = 2$
$3^3 = 3^2 \cdot 3 = 6$
$3^4 = 6 \cdot 3 = 4$
$3^5 = 4 \times 3 = 5$

$3^6 = 5 \times 3 = 1$

$O(3) = 6 = |\mathbb{Z}_7^*|$

$\Rightarrow 3$ is a generator of $(\mathbb{Z}_7^*, \times_n)$

Consider the following eq^n in $(\mathbb{Z}_7^*, \times_n)$

$$3^x \equiv 5 \mod 7$$

$x = 5$ is the sol^n.

$$\boxed{x = \log_3 5 \mod 7}$$

$(\mathbb{Z}_{47}^*, \times)$: $a = 5$ is a generator.

$$5^x \equiv 41 \mod 47$$

Discrete Logarithm Problem: Given a prime p, $\beta \in \mathbb{Z}_p^*$. Let α be a primitive element of $\mathbb{Z}_p^*$. Find x such that

$$\boxed{\alpha^x \equiv \beta \bmod p} \checkmark$$

Note: When p is large (≥ 300 decimal digits) DL problem is computationally very hard to solve.

Diffie-Hellman Problem: Eve knows α, p, A & B

She wants to find the key $K_{AB} = \alpha^{ab} \bmod p$

$A = \alpha^a \bmod p$

Solution: 1. Compute $\boxed{a = \log_\alpha A \bmod p} \checkmark$

2. " $B^a = \underline{\underline{K_{AB}}} = \alpha^{ab} \bmod p$

Security of Diffie-Hellman

1. Brute Force : Take large p (≥ 300 decimal digits)
2. Discrete Logarithm :
3. Man in the Middle Attack :

Alice	Eve	Bob
Chooses $x \in \{2, 3, \dots, p-2\}$	chooses $z \in \{2, \dots, p-2\}$	chooses $y \in \{2, \dots, p-2\}$
$R_1 = \alpha^x \bmod p$ $\longrightarrow$	R_1	
R_2 $\longleftarrow$	$R_2 = \alpha^z \bmod p$	$\longrightarrow$ R_2
	R_3	$\longleftarrow$ $R_3 = R_2^y \bmod p$
$K_1 = R_2^x \bmod p$	$K_1 = R_1^z \bmod p$ $K_2 = R_3^z \bmod p$	$K_2 = R_2^y \bmod p$

1. Alice send a message using the key K_1.
2. Eve intercept it and decrypt it using K_1.
3. Eve read the message, encrypt it using K_2 and she will send it to Bob.
4. If Bob send a send a message to Alice, Eve will intercept it, read it, encrypt it using K_1 and she will send it to Alice.

Generalized Discrete Logarithm Problem :

Given a cyclic group $(G, *)$ and $|G| = n$. Let α be a generator of $(G, *)$ and let $\beta \in G$.

Find x s.t.

$$\beta = \underbrace{\alpha * \alpha * \cdots * \alpha}_{x \text{ times}}$$

Elgamal Cryptosystem

Alice | **Bob**

Key Generation:

1. Select a large prime p
2. Select $K_{pr} = d \in \{2, 3, \ldots, p-2\}$
3. Select a generator e_1 of $\mathbb{Z}_p^*$
4. Compute $e_2 = e_1^d \bmod p$.

$K_{pub} = (e_1, e_2, p)$ - Public Key.

$(e_1, e_2, p) \longleftarrow$

Encryption:

1. Select a random integer $r \in \{1, 2, \ldots, p-1\}$
2. $c_1 = e_1^r \bmod p$
3. $c_2 = (\underbrace{X}_{\text{plaintext}} \cdot e_2^r) \bmod p$

$(c_1, c_2) \longrightarrow (c_1, c_2)$

Decryption:

1. $c_1' = c_1^d \bmod p$
2. $x = c_2 \cdot (c_1')^{-1} \bmod p$

Proof:

$$c_2 \cdot (c_1')^{-1} \bmod p = c_2 \cdot (c_1^d)^{-1} \bmod p$$
$$= c_2 \cdot (e_1^{rd})^{-1} \bmod p$$
$$= X \cdot e_2^r \cdot (e_1^{rd})^{-1} \bmod p$$
$$= X \cdot (\underline{e_1^{rd}}) \cdot (\underline{e_1^{rd}})^{-1} \bmod p$$
$$= X \bmod p$$

$$\boxed{X = c_2 (c_1')^{-1} \bmod p}$$

Note:

$$X = c_2 \cdot (c_1')^{-1} \bmod p$$
$$= c_2 \times (c_1^d)^{-1} \bmod p$$
$$= c_2 \times c_1^{-d} \bmod p$$
$$= c_2 \times c_1^{p-1} \times c_1^{-d} \bmod p$$
$$X = c_2 \times c_1^{p-1-d} \bmod p$$

Fermat's little theorem

If a & p are coprime then $a^{p-1} = 1 \bmod p$

Security of Elgamal

1. **Brute Force**: Take p very large.

2. **Known Plaintext attack**:

Let Alice uses the same random exponent r for two plaintexts P & P'

Let Eve knows P and its encryption.

Let $$C_2 = P \times e_2^r \bmod p \quad \text{—— (1)}$$
$$C_2' = P' \times e_2^r \bmod p \quad \text{—— (2)}$$

Eve can find P' as follows:

1. $e_2^r = C_2 \cdot P^{-1} \bmod p$ by (1)
2. $P' = C_2' (e_2^r)^{-1} \bmod p$ by (2)

To avoid known plaintext attack Alice has to use a different exponent r each time.