Diffie-Hellman protocol:- (Deffinition)

→ Using this protocol, we obtain a shared key for the participants.

→ There is an exponential gap betⁿ participant work and attacker work.

→ We have the exponential gap as follows

1) fix ~~some~~ a large prime, p (e.g. 600 digits)
2) fix an integer, g in the range 1 to p $\{1 - - p\}$

$\boxed{\text{Alice}} \longleftrightarrow \boxed{\text{Bob}}$

So, p and g are the parameters of DH protocols

How DH protocol works?

Alice	Bob
choose random a is $\{1 .. p-1\}$	choose random b in $\{1, .. p-1\}$

"Alice", $A \leftarrow g^a \pmod p$ $\longrightarrow$

"Bob", $B \leftarrow g^b \pmod p$ $\longleftarrow$

$B^a \pmod p = (g^b)^a = K_{AB}$

$= g^{ab} \pmod p$

$= (g^a)^b = A^b \pmod p$

even though both parties captured different values, they end up with same $g^{ab} \pmod p$.

Why DH secure :-

1) eavesdropper can't figure out g^{ab} K_{AB}
2) ,, sees prime, p and generator, g fixed forever.
3) ,, sees the value of A ($A \to B$) and B ($B \to A$)

Eavesdropper sees:

$P, g, A = g^a \pmod p$, $B = g^b \pmod p$

Why insecure against MITM?

Alice	MITM	Bob
$A \leftarrow g^a$ $\longrightarrow$	a'	$A' \longrightarrow g^{a'}$ $\longrightarrow$
$\longleftarrow$ $B' \leftarrow g^{b'}$	b'	$\longleftarrow$ $B' \leftarrow g^b$
$g^{ab'}$	$g^{ab'}, g^{a'b}$	$g^{ba'}$

→ Alice computes her part of the secret key and get $g^{ab'}$

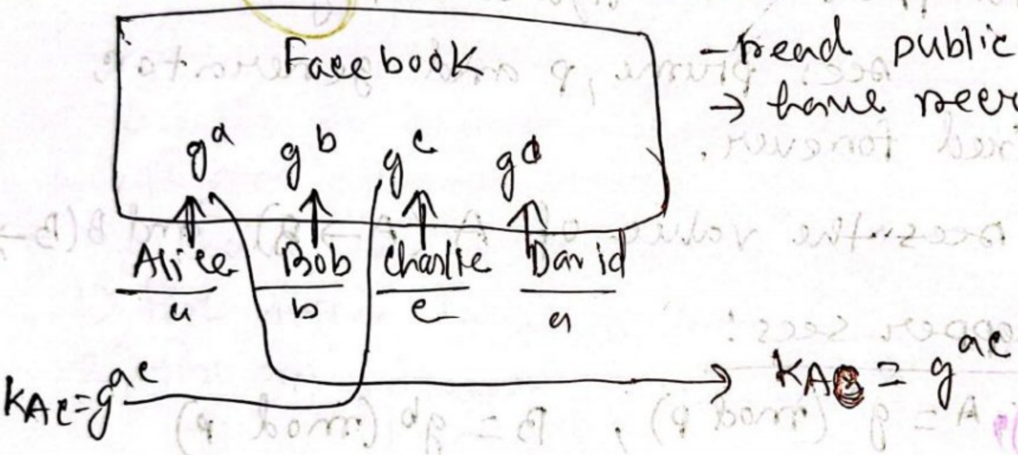
→ Bob gets $g^{a'b}$

which aren't the same keys
But MITM can compute both $g^{ab'}$ and $g^{a'b}$
he knows a', b'

→ If alice sends a msg to Bob, MITM can decrypt it because he knows the secret key.
So, DH protocol is insecure against MITM attack.

DH properties:

Non interactive Protocol of DH :-

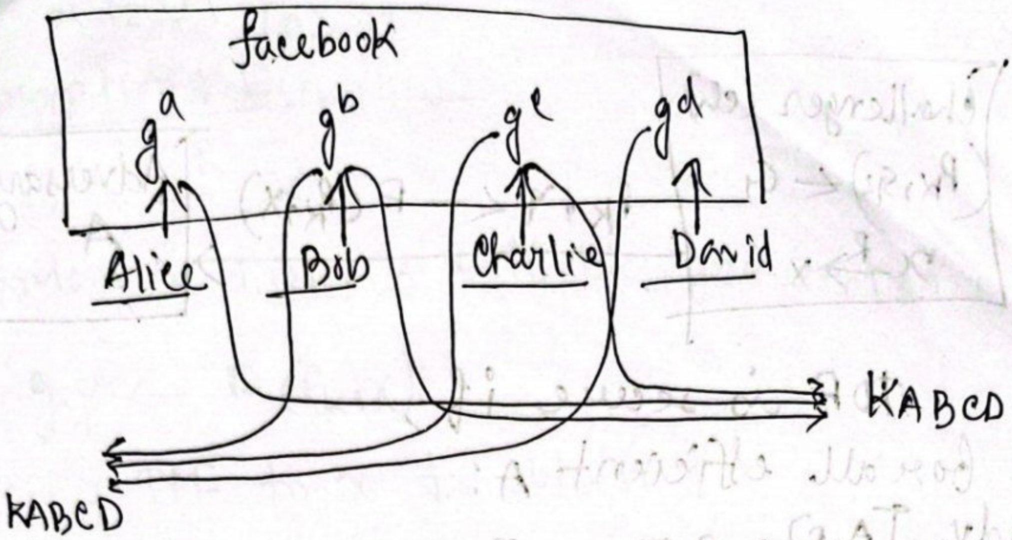


- read public profile
→ have secret key

Joint shared key/open problem :-

for, $N = 2$ → DH protocol
$N = 3$ → Four "
$N = 4/5/$up → open problem
→ anybody can read anyone's public profile
→ have a joint shared key

CS CamScanner
CS CamScanner


facebook
g^a
g^b
g^c
g^d
Alice
Bob
Charlie
David
K_{ABCD}
K_{ABCD}

Information Security
CT-01 13/03/21

⑤

Basic key Exchange

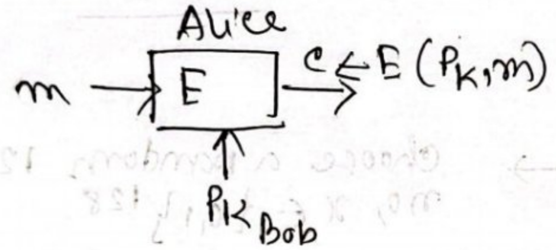
Public key exchange:-

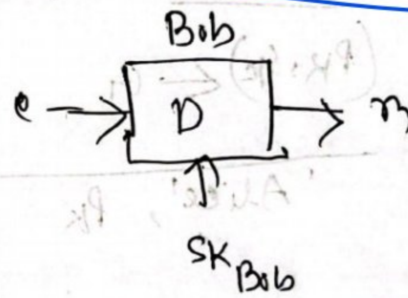
Public key encryption is triple algorithm (G, E, D)

$G()$: generate key pair (P_K, S_K)

$E(P_K, m)$: generate $c \leftarrow E(P_K, m)$ P_K : Public key

$D(S_K, m)$: generate $m \leftarrow D(S_K, C)$ S_K : secret key





→ Alice sends m to Bob. 'm' is encrypted by using P_K of Bob which results is C.

→ Bob decrypts C by using S_K of Bob and gets m.

Establishing a shared secret key using PKC:

goal: Alice and Bob want shared secret, unknown to eavesdropper.

(1) Alice generates P_K and S_K by using G.

(2) Alice sends P_K to Bob and says that "this msg came from Alice".

(3) Bob will generate a random 128 bit value, x and sends back to Alice saying "this msg is from Bob"

He also sends encryption of x under Alice's private key (P_K)

(4) Alice will receive the ciphertext she will decrypt it by using her S_K and give him the value, X

(5) X is the shared secret key.

(6) Bob can't send msg to Alice without X.

Alice | **Bob**

$(P_K, S_K) \leftarrow G$

'Alice', P_K $\longrightarrow$ choose a random 128 bit no, $x \in \{0,1\}^{128}$

$D(S_K, c) \rightarrow X \longleftarrow$ 'Bob', $c \leftarrow E(P_K, x)$

X : share secret key

~~why~~ public key encryption (PKC) secure?

Secure against eavesdropping.

→ attacker can see P_K, $E(P_K, x)$ and wants $x \in m$.

→ attacker can't ~~disgusti~~ distinguish

$\{P_K, E(P_K, x), x\}$ from $\{P_K, E(P_K, x)$ rand $\in m\}$

So it is semantically secure.

→ if only encryption is given, attacker can't tell whether the plaintext is X or just a random junk.

CamScanner
CamScanner

→ x can be used as a session key against 02 parties.

Why PKC is insecure against MITM?

Alice — MITM — Bob

$(P_K, S_K) \leftarrow G()$ $(P_K', S_K') \leftarrow G()$

"Alice", P_K → 'Alice', P_K' → choose random $x \in \{0,1\}^{128}$

x ← "Bob", $E(P_K, x)$ | 'Bob', $E(P_K', x)$

→ Alice ~~needs~~ sends P_K to Bob. MITM intercepts it and sends P_K' to Bob.

→ Bob receives the msg without knowing that MITM intercepted.

→ Bob will choose x and send $E(P_K', x)$

→ MITM intercepts this msg and reply this msg by something else.

→ Alice decrypts c by using her own secret key S_K and reveals x to MITM.

→ Alice obtains x and thinks she did a key exchange with Bob. But MITM also knows x.

So PKC is insecure against MITM.

(i) Alice generate P_K & S_K using $G()$ (ii) At the same time, MITM generates its own public key & secret key pair (P_K', S_K')