Lec 10: Cryptography (2)

CSED415: Computer Security

Spring 2025

Seulbae Kim

POSTECH

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

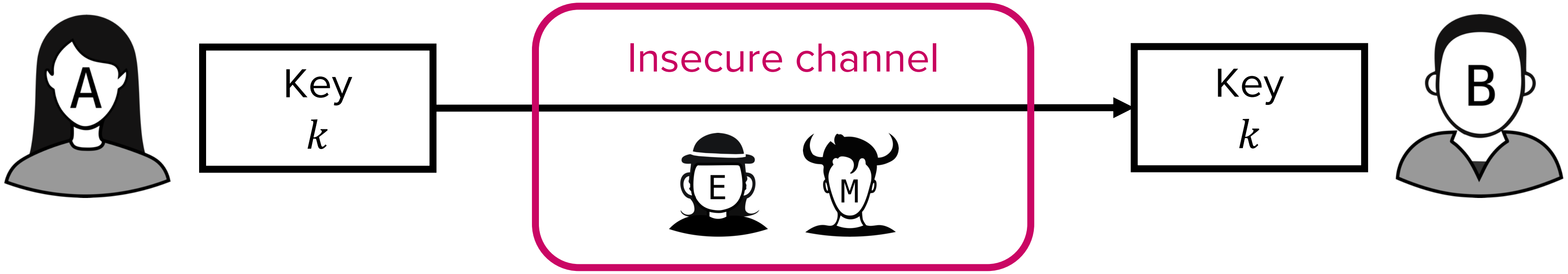
Cryptography roadmap

Goal \ Scheme	Symmetric Key	Asymmetric Key Today's topic
Confidentiality	✅ One Time Pad (OTP) ✅ Block ciphers (DES, AES) ✅ Stream ciphers	• DH secure key exchange • ElGamal encryption • RSA encryption
Integrity & Authentication	• Message Authentication Code (MAC)	• Digital signature

Secure Key Exchange (Diffie-Hellman)

Limitation of symmetric key scheme

- Key sharing problem:
  - Symmetric key cryptography requires key k to be securely shared between Alice and Bob
  - For securely sharing messages over insecure channels, symmetric key cryptography is used
  - However, symmetric schemes do not work without a shared k



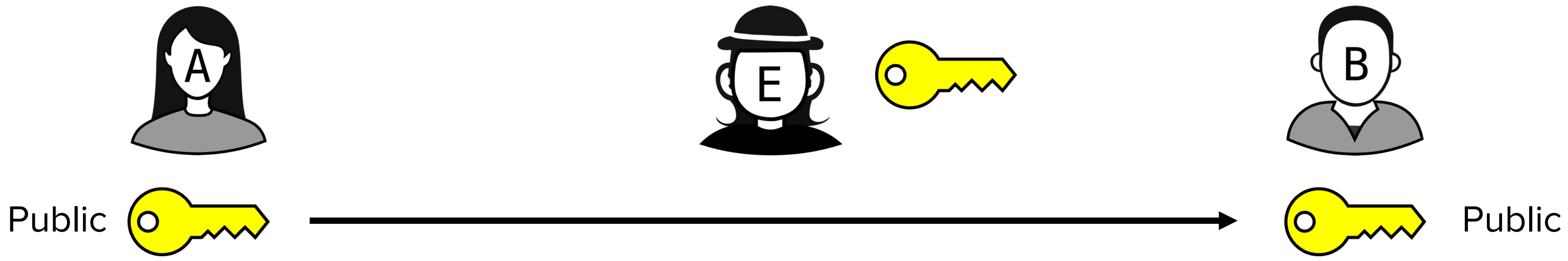
A secure key exchange algorithm is needed

Diffie-Hellman (DH) key exchange

- Named after Whitfield Diffie and Martin Hellman
- Key idea:
  - Derive a shared secret key mathematically, rather than sending it directly

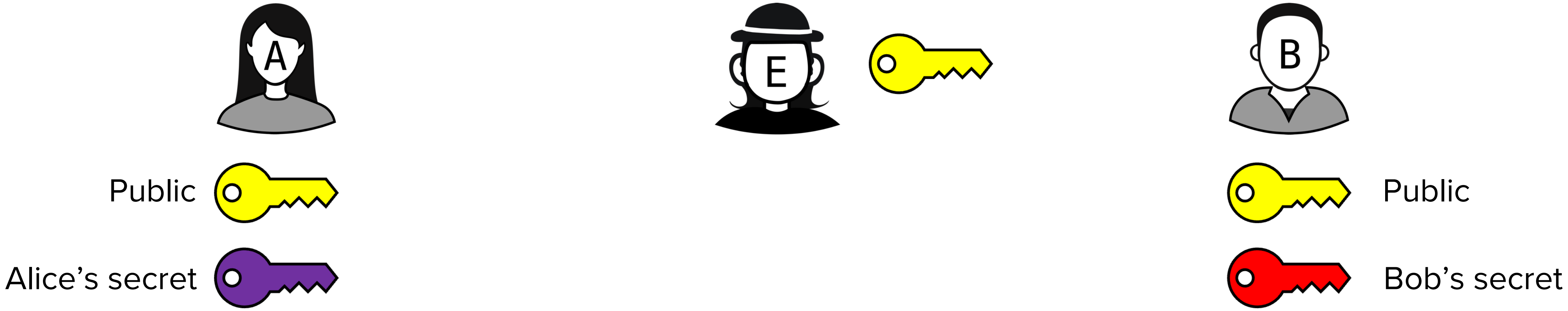
Intuitive example: Colored keys

1. Alice shares a yellow key (public key) with Bob (and Eve)



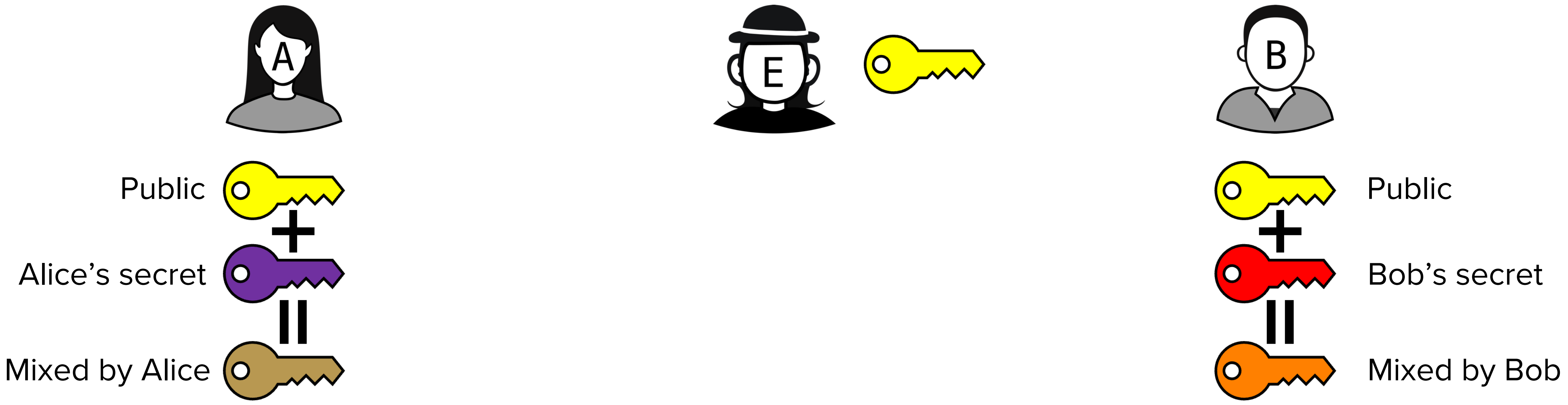
Intuitive example: Colored keys

2. Alice and Bob each select their own colored key (secret key) and keep it to themselves



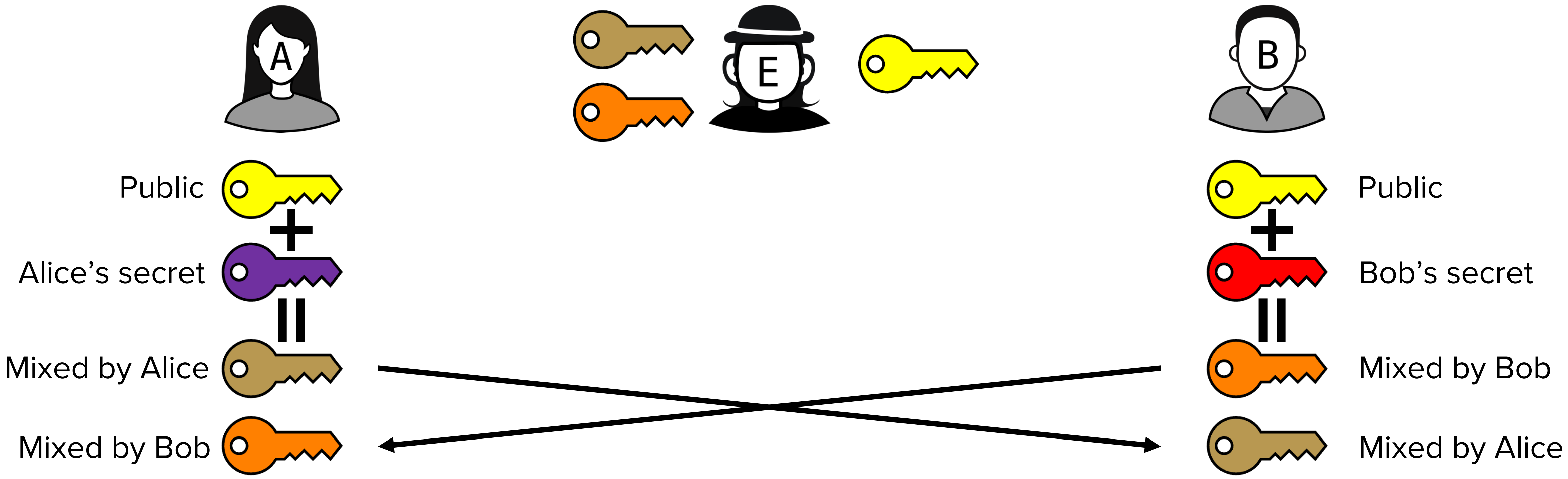
Intuitive example: Colored keys

3. Alice and Bob both mix the colors of the public key and their own secret key, generating mixed keys



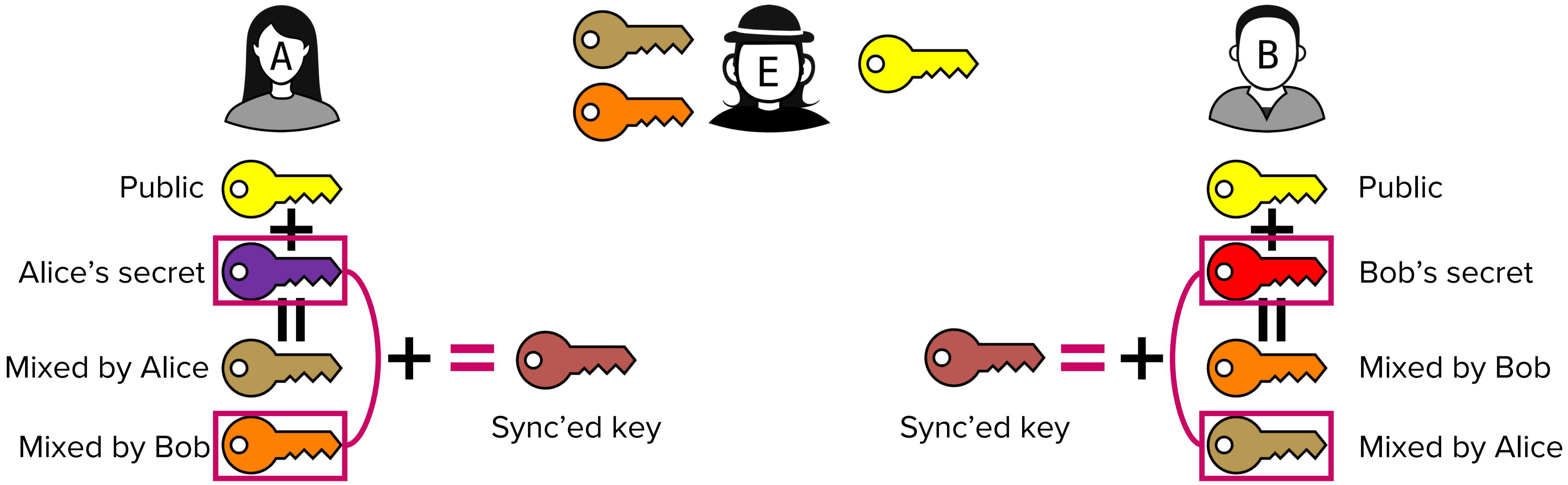
Intuitive example: Colored keys

4. Alice and Bob exchange the mixed keys
   - Eve can see the mixed keys



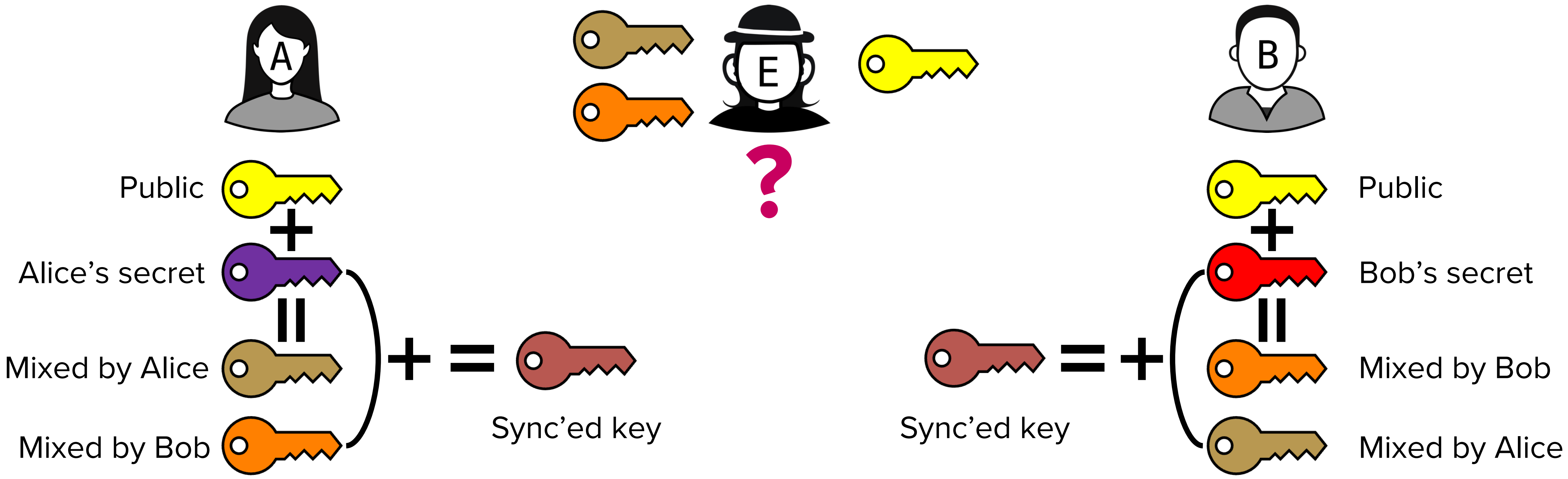
Intuitive example: Colored keys

5. Finally, each party mixes the received mixed key with their own secret key again, resulting in the same sync'ed key



Intuitive example: Colored keys

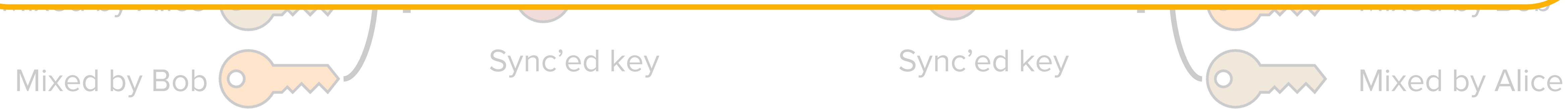
6. Eve cannot derive the sync'ed key without knowing Alice's or Bob's secret keys



Intuitive example: Colored keys

6. Eve cannot derive the sync'ed key without knowing Alice's or Bob's secret keys

Some operations are easy to perform in one direction but extremely hard to reverse



Background: Number theory

- Greatest common denominator $d = \gcd(a, b)$:
  - Largest integer d such that d divides a and d divides b
- Relatively prime (or, coprime)
  - If $\gcd(a, b) = 1$, then a and b are relatively prime
    - Is 15 relatively prime to 28? Yes. $\gcd(15,28) = 1$
    - Is 14 and 49 relatively prime? No. $\gcd(14,49) = 7$
    - Are 23 and 443 coprime? Yes. Two prime numbers are always coprime
      - Hint: 23 and 443 are prime numbers

Diffie-Hellman key exchange

POSTECH

1. Choose a prime num p and its generator g such that $g < p$
   - Both p and g are shared (public keys)
   - g is a generator of p if $g^k \, mod \, p$ can take any value in $[1, \ldots, p-1]$
   - Example: $p = 11$

$g^k \, mod \, p$	$i = 0$	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$	$i = 7$	$i = 8$	$i = 9$	10	Generator?
$2^i \, mod \, 11$	1	2	4	8	5	10	9	7	3	6	1	Y
$3^i \, mod \, 11$	1	3	9	5	4	1	3	9	5	4	1	N
$4^i \, mod \, 11$	1	4	5	9	3	1	4	5	9	3	1	N
$5^i \, mod \, 11$	1	5	3	4	9	1	5	3	4	9	1	N
$6^i \, mod \, 11$	1	6	3	7	9	10	5	8	4	2	1	Y
…												

→ We can select $g = 6$

Diffie-Hellman key exchange

2. Alice and Bob each choose a secret key
   - Assume Alice's secret key $a = 15$
     and Bob's secret key $b = 8$

Public	Secret
$p = 11$	$a = 15$
$g = 6$	$b = 8$

Diffie-Hellman key exchange

POSTECH

Public	Secret
$p = 11$	$a = 15$
$g = 6$	$b = 8$

2. Alice and Bob each choose a secret key
   - Assume Alice's secret key $a = 15$ and Bob's secret key $b = 8$

3. Alice and Bob compute $g^x \ mod \ p$ where x is the secret key
   - $A = g^a \ mod \ p = 6^{15} \ mod \ 11$
   - $B = g^b \ mod \ p = 6^8 \ mod \ 11$

← Too large to be calculated by hand?

Diffie-Hellman key exchange

- Note: Modular exponentiation
  - We can compute $x^y \ mod \ n$ by breaking y down into powers of 2
  - e.g., $6^{15} \ mod \ 11$ → 15 = 8 + 4 + 2 + 1
    - $6^{15} = 6^8 \times 6^4 \times 6^2 \times 6$
    - $6 \ mod \ 11 = 6$
    - $6^2 \ mod \ 11 = 36 \ mod \ 11 = 3$
    - $6^4 \ mod \ 11 = (6^2)^2 \ mod \ 11 = 3^2 \ mod \ 11 = 9$
    - $6^8 \ mod \ 11 = (6^4)^2 \ mod \ 11 = 9^2 \ mod \ 11 = 81 \ mod \ 11 = 4$
    - Thus, $6^{15} \ mod \ 11 = (4 \times 9 \times 3 \times 6) \ mod \ 11 = 648 \ mod \ 11 = 10$

Diffie-Hellman key exchange

2. Alice and Bob each choose a secret key
   - Assume Alice's secret key $a = 15$
     and Bob's secret key $b = 8$

Public	Secret
$p = 11$	$a = 15$
$g = 6$	$b = 8$

3. Alice and Bob compute $g^x \ mod \ p$ where x is the secret key
   - Alice's mixed key $A = g^a \ mod \ p = 6^{15} \ mod \ 11 = 10$
   - Bob's mixed key $B = g^b \ mod \ p = 6^8 \ mod \ 11 = 4$

Modular exponentiation!

Diffie-Hellman key exchange

4. Alice and Bob exchange their mixed keys
   - $A = 6^{15}\ mod\ 11 = 10$
   - $B = 6^{8}\ mod\ 11 = 4$

Public	Secret
$p = 11$	$a = 15$
$g = 6$	$b = 8$
$A = 10$	
$B = 4$	

Diffie-Hellman key exchange

5. Alice and Bob then can generate a common shared key k by raising the exchanged mixed key to their respective secret keys
   - Alice: $k = B^a \ mod\ p = 4^{15} \ mod\ 11$
   - Bob: $k = A^b \ mod\ p = 10^8 \ mod\ 11$

Public	Secret
$p = 11$	$a = 15$
$g = 6$	$b = 8$
$A = 10$	
$B = 4$	

Diffie-Hellman key exchange

POSTECH

5. Alice and Bob then can generate a common shared key k by raising the exchanged mixed key to their respective secret keys
   - Alice: $k = B^a \bmod p = 4^{15} \bmod 11$
   - Bob: $k = A^b \bmod p = 10^8 \bmod 11$

Public	Secret
$p = 11$	$a = 15$
$g = 6$	$b = 8$
$A = 10$	
$B = 4$	

$$4^{15} \bmod 11 = 4^8 \times 4^4 \times 4^2 \times 4 \bmod 11$$

$$4 \bmod 11 = 4$$
$$4^2 \bmod 11 = 16 \bmod 11 = 5$$
$$4^4 \bmod 11 = (4^2)^2 \bmod 11 = 5^2 \bmod 11 = 25 \bmod 11 = 3$$
$$4^8 \bmod 11 = (4^4)^2 \bmod 11 = 9 \bmod 11 = 9$$

Diffie-Hellman key exchange

5. Alice and Bob then can generate a common shared key k by raising the exchanged mixed key to their respective secret keys
   - Alice: $k = B^a \bmod p = 4^{15} \bmod 11 = 1$
   - Bob: $k = A^b \bmod p = 10^8 \bmod 11$

Public	Secret
$p = 11$	$a = 15$
$g = 6$	$b = 8$
$A = 10$	
$B = 4$	

$$4^{15} \bmod 11 = 4^8 \times 4^4 \times 4^2 \times 4 \bmod 11 = 9 \times 3 \times 5 \times 4 \bmod 11 = 1$$

$$4 \bmod 11 = 4$$
$$4^2 \bmod 11 = 16 \bmod 11 = 5$$
$$4^4 \bmod 11 = (4^2)^2 \bmod 11 = 5^2 \bmod 11 = 25 \bmod 11 = 3$$
$$4^8 \bmod 11 = (4^4)^2 \bmod 11 = 9 \bmod 11 = 9$$

Diffie-Hellman key exchange

5. Alice and Bob then can generate a common shared key k by raising the exchanged mixed key to their respective secret keys
   - Alice: $k = B^a \bmod p = 4^{15} \bmod 11 = 1$
   - Bob: $k = A^b \bmod p = 10^8 \bmod 11 = 1$

Public	Secret
$p = 11$	$a = 15$
$g = 6$	$b = 8$
$A = 10$	
$B = 4$	

$10 \bmod 11 \equiv -1 \bmod 11$
$10^8 \bmod 11 = (-1)^8 \bmod 11 = 1 \bmod 11 = 1$

Diffie-Hellman key exchange

5. Alice and Bob then can generate a common shared key k by raising the exchanged mixed key to their respective secret keys
   - Alice: $k = B^a \ mod \ p = 4^{15} \ mod \ 11 = 1$
   - Bob: $k = A^b \ mod \ p = 10^8 \ mod \ 11 = 1$

Public	Secret
$p = 11$	$a = 15$
$g = 6$	$b = 8$
$A = 10$	$k = 1$
$B = 4$	

Alice and Bob have successfully generated a shared secret key k

Diffie-Hellman key exchange

- Can Eve deduce the shared secret key?
  - Problem that Eve needs to solve:
    - Given p, g, A, and B, find a, b, and k such that $A^b \bmod p = B^a \bmod p = k$.
  - Discrete log problem (DLP):
    - Given p, g, and $B = g^a \bmod p$, it is computationally difficult to find a, especially for large prime number p
      - e.g., $g^a \bmod p = 6^a \bmod 11 = 4$ → Can you find a?
      - How about $43^a \bmod 170141183460469231731687303715884105727 = 107658615995071204650478536027214115641$?

Public	Secret
$p = 11$	$a = 15$
$g = 6$	$b = 8$
$A = 10$	$k = 1$
$B = 4$	

Generalization of Diffie-Hellman key exchange

- A large prime p and a generator g are shared

Generalization of Diffie-Hellman key exchange

- A large prime p and a generator g are shared
- Mixing:
  - Alice chooses a secret integer a and computes $A = g^a \bmod p$
  - Bob chooses a secret integer b and computes $B = g^b \bmod p$

Generalization of Diffie-Hellman key exchange

- A large prime p and a generator g are shared
- Mixing:
  - Alice chooses a secret integer a and computes $A = g^a \; mod \; p$
  - Bob chooses a secret integer b and computes $B = g^b \; mod \; p$
- Deducing the shared secret:
  - Alice computes $k = B^a \; mod \; p = \left(g^b\right)^a \; mod \; p = g^{ab} \; mod \; p$
  - Bob computes $k = A^b \; mod \; p = (g^a)^b \; mod \; p = g^{ab} \; mod \; p$

Generalization of Diffie-Hellman key exchange

- A large prime p and a generator g are shared
- Mixing:
  - Alice chooses a secret integer a and computes $A = g^a \mod p$
  - Bob chooses a secret integer b and computes $B = g^b \mod p$
- Deducing the shared secret:
  - Alice computes $k = B^a \mod p = \left(g^b\right)^a \mod p = g^{ab} \mod p$
  - Bob computes $k = A^b \mod p = (g^a)^b \mod p = g^{ab} \mod p$
- Eve knows p, g, A, and B
  - Eve cannot feasibly solve DLP to compute a nor b if p is large

Generalization of Diffie-Hellman key exchange

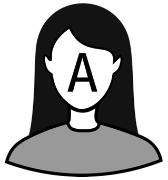
- A large prime p and a generator g are shared
- Mixing:

DH is secure against passive attacks

- Eve knows p, g, A, and B
  - Eve cannot feasibly solve DLP to compute a nor b if p is large

Diffie-Hellman key exchange

- Intended key exchange



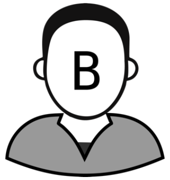
Select a

Compute $g^{a}\ mod\ p$

Receive $g^{b}\ mod\ p$

Compute $\left(g^{b}\right)^{a}\ mod\ p$

Shared secret



Select b

Compute $g^{b}\ mod\ p$

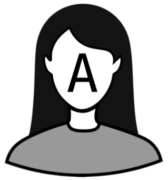
Receive $g^{a}\ mod\ p$

Compute $(g^{a})^{b}\ mod\ p$

Shared secret

Diffie-Hellman – Man in the Middle (MitM) attack

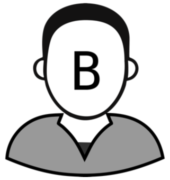
- What if Mallory actively alters key exchange messages?



Select a
Compute $g^a \bmod p$

MitM

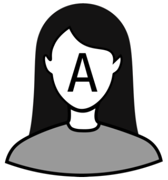
Select m
Compute $g^m \bmod p$



Select b
Compute $g^b \bmod p$

Diffie-Hellman – Man in the Middle (MitM) attack

- What if Mallory actively alters key exchange messages?



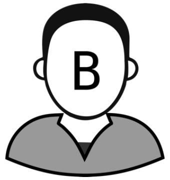
Select a
Compute $g^a \bmod p$

Receive $g^m \bmod p$
Compute $(g^m)^a \bmod p$

Select m
Compute $g^m \bmod p$

Receive $g^a \bmod p$
Compute $(g^a)^m \bmod p$

Receive $g^b \bmod p$
Compute $(g^b)^m \bmod p$

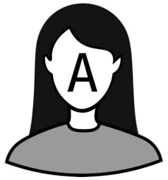
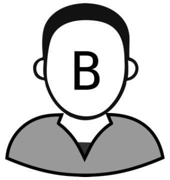


Select b
Compute $g^b \bmod p$

Receive $g^m \bmod p$
Compute $(g^m)^b \bmod p$

Diffie-Hellman – Man in the Middle (MitM) attack

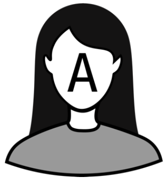
- What if Mallory actively alters key exchange messages?

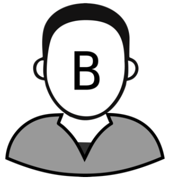
			
Select a Compute $g^a \bmod p$	Select m Compute $g^m \bmod p$		Select b Compute $g^b \bmod p$
Receive $g^m \bmod p$ Compute $(g^m)^a \bmod p$	Receive $g^a \bmod p$ Compute $(g^a)^m \bmod p$	Receive $g^b \bmod p$ Compute $\left(g^b\right)^m \bmod p$	Receive $g^m \bmod p$ Compute $(g^m)^b \bmod p$
$k = g^{ma} \bmod p$	$k = g^{ma} \bmod p$	$k' = g^{mb} \bmod p$	$k' = g^{mb} \bmod p$

Mallory keeps two shared keys: k for Alice, and k' for Bob, respectively

Diffie-Hellman – Man in the Middle (MitM) attack

- Then, Mallory can tamper with messages



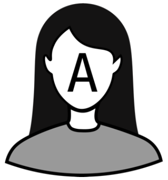


Alice	Mallory	Mallory	Bob
Select a	Select m		Select b
Compute $g^a \bmod p$	Compute $g^m \bmod p$		Compute $g^b \bmod p$
Receive $g^m \bmod p$	Receive $g^a \bmod p$	Receive $g^b \bmod p$	Receive $g^m \bmod p$
Compute $(g^m)^a \bmod p$	Compute $(g^a)^m \bmod p$	Compute $\left(g^b\right)^m \bmod p$	Compute $(g^m)^b \bmod p$
$k = g^{ma} \bmod p$	$k = g^{ma} \bmod p$	$k' = g^{mb} \bmod p$	$k' = g^{mb} \bmod p$
$E(k, msg)$ →	$D\big(k, E(k, msg)\big)$		

Mallory can decrypt Alice's msg using k

Diffie-Hellman – Man in the Middle (MitM) attack

- Then, Mallory can tamper with messages



Select a

Compute $g^{a}\ mod\ p$

Receive $g^{m}\ mod\ p$

Compute $(g^{m})^{a}\ mod\ p$

$k = g^{ma}\ mod\ p$

$E(k, msg)$

Select m

Compute $g^{m}\ mod\ p$

Receive $g^{a}\ mod\ p$

Compute $(g^{a})^{m}\ mod\ p$

$k = g^{ma}\ mod\ p$

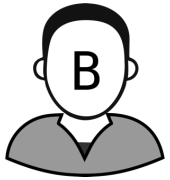
$D(k, E(k, msg))$

Receive $g^{b}\ mod\ p$

Compute $\left(g^{b}\right)^{m}\ mod\ p$

$k' = g^{mb}\ mod\ p$

$E(k', msg')$



Select b

Compute $g^{b}\ mod\ p$

Receive $g^{m}\ mod\ p$

Compute $(g^{m})^{b}\ mod\ p$

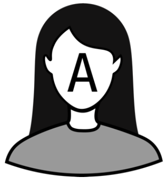
$k' = g^{mb}\ mod\ p$

$D(k', E(k', msg'))$

Mallory can modify the msg to msg' and encrypt it using k' for Bob

Diffie-Hellman – Man in the Middle (MitM) attack

- Then, Mallory can tamper with messages



Select a
Compute $g^a \; mod \; p$

Receive $g^m \; mod \; p$
Compute $(g^m)^a \; mod \; p$

$k = g^{ma} \; mod \; p$

$E(k, msg)$

Select m
Compute $g^m \; mod \; p$

Receive $g^a \; mod \; p$
Compute $(g^a)^m \; mod \; p$

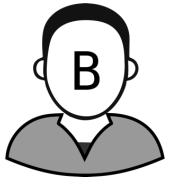
$k = g^{ma} \; mod \; p$

$D\big(k, E(k, msg)\big)$

Receive $g^b \; mod \; p$
Compute $\left(g^b\right)^m \; mod \; p$

$k' = g^{mb} \; mod \; p$

$E(k', msg')$



Select b
Compute $g^b \; mod \; p$

Receive $g^m \; mod \; p$
Compute $(g^m)^b \; mod \; p$

$k' = g^{mb} \; mod \; p$

$D\big(k', E(k', msg')\big)$

Alice and Bob are tricked into believing that they are securely communicating

Diffie-Hellman – Man in the Middle (MitM) attack

- What if Mallory actively changes key exchange messages?

DH key exchange is insecure against active attacks

$k = g^{ma} \bmod p$ | $k = g^{ma} \bmod p$ | $k' = g^{mb} \bmod p$ | $k' = g^{mb} \bmod p$

$E(k, msg) \rightarrow D(k, E(k, msg))$

$E(k', msg') \rightarrow D(k', E(k', msg'))$

Alice and Bob are tricked into believing that they are securely communicating

Key exchange in the presence of active attacker

- When Mallory (an active attacker) exists, it is impossible for Alice and Bob to start from scratch and exchange messages to derive a shared key unknown to the adversary
- Why?
  - Bob cannot distinguish Alice from Mallory because DH does not provide authentication
- Solution:
  - Alice and Bob needs an "information advantage" over the adversary
    - Typically, in the form of long-lived keys (e.g., previously shared keys)
    - More on this next week!

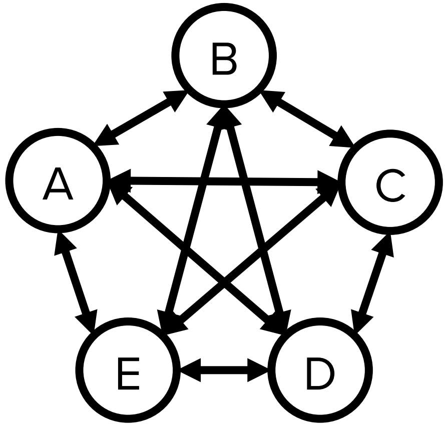
Cryptography roadmap

Goal \ Scheme	**Symmetric Key**	**Asymmetric Key**
Confidentiality	✅ One Time Pad (OTP) ✅ Block ciphers (DES, AES) ✅ Stream ciphers	✅ DH secure key exchange • ElGamal encryption • RSA encryption
Integrity & Authentication	• Message Authentication Code (MAC)	• Digital signature

Asymmetric Cryptography (Public key Scheme)

Motivation

- Another limitation of symmetric key schemes
  - The number of symmetric keys needed grows exponentially



→ $\binom{n}{2} = \frac{n(n-1)}{2}$ keys are needed for n people to securely communicate using symmetric key schemes

Solution: Public-key cryptography

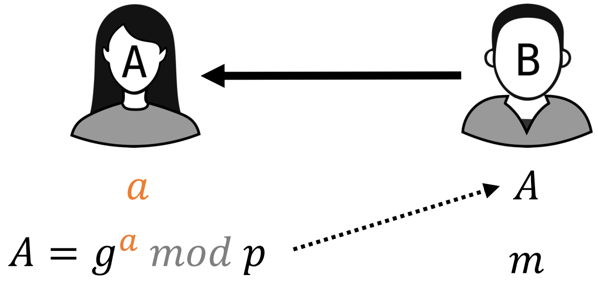
- Idea:
  - Utilize an asymmetric key pair $< k_p, k_s >$, where
    - k_p: Public key that is publicly released
    - k_s: Secret key, which is kept secret
  - Any sender can encrypt a message using the receiver's public key
    - $c = E(k_p, m)$
  - Only the receiver can decrypt the ciphertext using his or her own private key
    - $m = D(k_s, c)$

ElGamal encryption

- An extension of Diffie-Hellman key exchange
  - DH only provides only a shared secret derivation
  - ElGamal supports direct encryption and decryption on top of DH key exchange

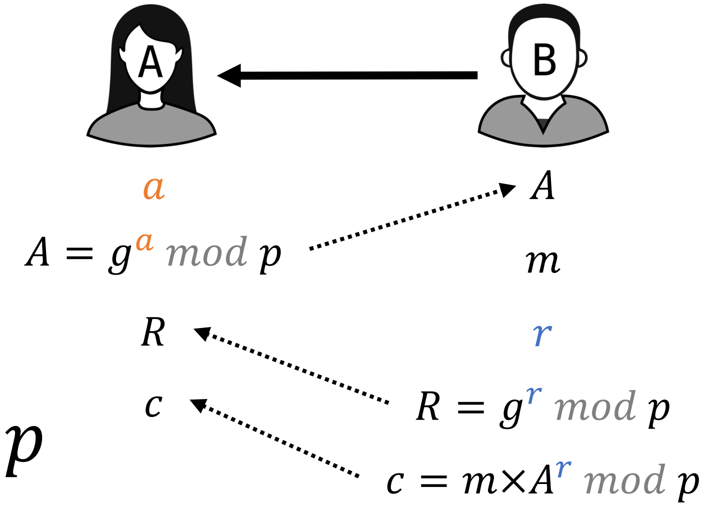
ElGamal encryption

- Alice chooses a secret key a
- Alice generates a public key $A = g^{a}\ mod\ p$
  - p (prime number) and g (generator) are public
- Bob wants to encrypt m for Alice

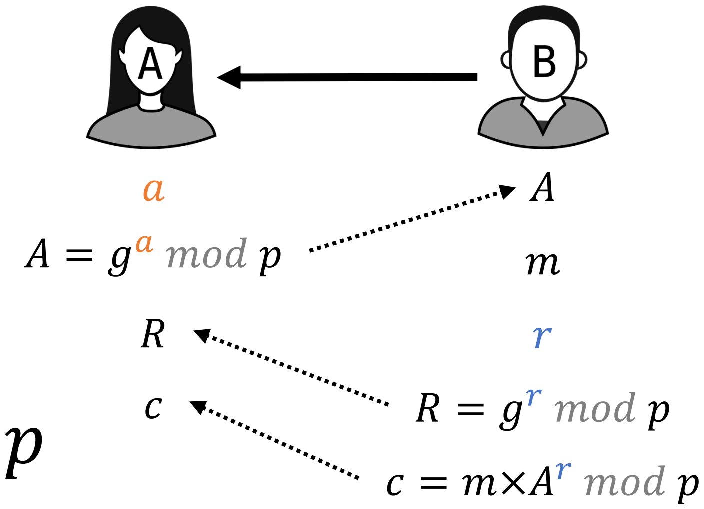


ElGamal encryption

- Alice chooses a secret key a
- Alice generates a public key $A = g^a \bmod p$
- Bob wants to encrypt m for Alice
  - Bob picks a random r and computes $R = g^r \bmod p$
  - Bob sends $\mathbf{c} = m \times A^r \bmod p$ and R to Alice



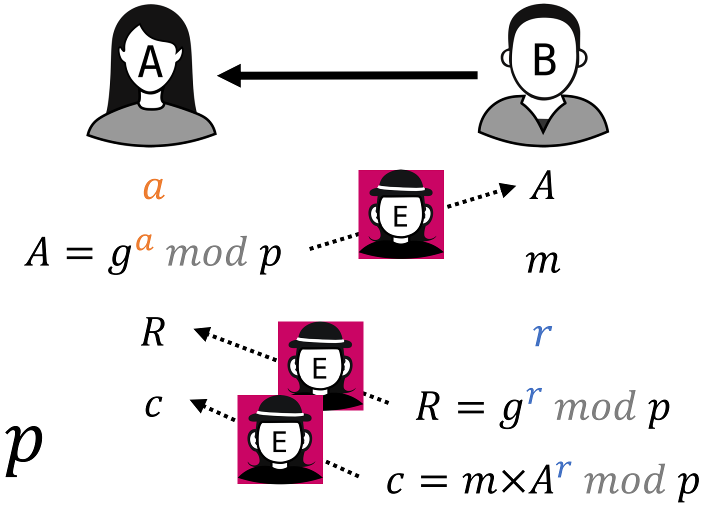
ElGamal encryption



- Alice chooses a secret key a
- Alice generates a public key $A = g^{a} \ mod \ p$
- Bob wants to encrypt m for Alice
  - Bob picks a random r and computes $R = g^{r} \ mod \ p$
  - Bob sends $\mathbf{c} = m \times A^{r} \ mod \ p$ and R to Alice
- Alice can decrypt c by:
  - $$c \times (R^{a})^{-1} = m \times A^{r} \times R^{-a} \ mod \ p = m \times (g^{a})^{r} \times (g^{r})^{-a} \ mod \ p$$
    $$= m \ mod \ p = m$$

ElGamal encryption

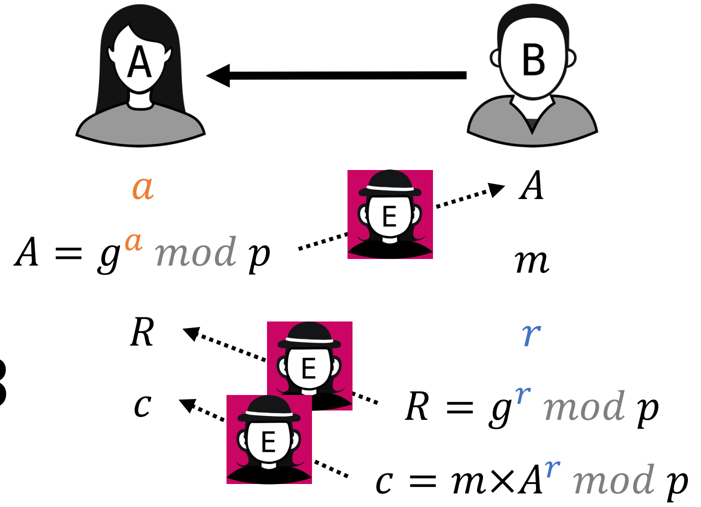
- Alice chooses a secret key a
- Alice generates a public key $A = g^a \mod p$
- Bob wants to encrypt m for Alice
  - Bob picks a random r and computes $R = g^r \mod p$
  - Bob sends $\mathbf{c} = m \times A^r \mod p$ and R to Alice
- Alice can decrypt c by:
  - $c \times (R^a)^{-1} = m \times A^r \times R^{-a} \mod p = m \times (g^a)^r \times (g^r)^{-a} \mod p$
    $= m \mod p = m$



Security: Given A, R, and c, Eve cannot recover m (DLP!)

ElGamal encryption

- Example
  - Public parameters: $p = 13$, $g = 2$
  - Alice's secret key $a = 3$ // randomly chosen
  - Alices' public key $A = g^a \bmod p = 2^3 \bmod 13 = 8$
  - Bob's message $\boxed{m = 11}$
  - Bob's random $r = 5$
  - Bob computes $R = g^r \bmod p = 2^5 \bmod 13 = 6$
  - Bob encrypts m: $c = m \times A^r \bmod p = 11 \times 8^5 \bmod 13 = 10$
  - Alice receives R and c from Bob and decrypts c to obtain m
    - $m = c \times (R^a)^{-1} \bmod p = 10 \times 6^{-3} \bmod 13 = \boxed{11}$ Correctly decrypted!



Using Python: `10 * pow(6, -3, 13) % 13`

Summary of ElGamal encryption

- ElGamal encryption provides confidentiality
  - Discrete logarithm problem
- ElGamal encryption does not provide integrity
  - Mallory can tamper with the ciphertext without decrypting it
  - e.g.,
    - Mallory (MitM) receives R and c from Bob
    - Mallory sends R and $\mathrm{c}' = c \times 2$ to Alice
    - Alice decrypts c' and retrieves $m \times 2 \ mod\ 13$

Cryptography roadmap

Goal \ Scheme	Symmetric Key	Asymmetric Key
Confidentiality	✅ One Time Pad (OTP) ✅ Block ciphers (DES, AES) ✅ Stream ciphers	✅ DH secure key exchange ✅ ElGamal encryption • RSA encryption
Integrity & Authentication	• Message Authentication Code (MAC)	• Digital signature

RSA Encryption

- Idea: Prime factorization of large numbers is hard
  - Q) What are the prime factors of 10403?

RSA Encryption

- Idea: Prime factorization of large numbers is hard
  - Q) What are the prime factors of 10403?
    - Naïve algorithm:

```
# N = pq where p and q are primes
def factorize(N):
    for i in range(2, sqrt(N)):
        if N mod i == 0:
            p = i
            q = N / i
            return (p, q)
```

This algorithm works, but takes time $O(\sqrt{N})$

e.g., using a 2048-bit N, naïve factorization takes $O(\sqrt{2^{2048}})$

RSA Encryption

- Choose two large primes p and q
- Compute public $N = pq$
- Compute the totient, $T = (p-1)(q-1)$
- Select public key e, such that e is relatively prime to T
- Compute private key $d = e^{-1}\ mod\ T$ // modular inverse of e
  - $ed = 1\ mod\ T$

RSA Encryption

- Encryption function:
  - $E(e, m) = m^e \ mod \ N = c$ // Anyone can encrypt using the public key e
- Decryption function:
  - $D(d, c) = c^d \ mod \ N$ // Only the receiver can decrypt using the private key d
  - Magically, $m = c^d \ mod \ N$ holds:
    - $$\begin{aligned} c^d \ mod \ N &= (m^e)^d \ mod \ N \\ &= m^{ed} \ mod \ N && \cdots \ ed = kT + 1 \text{ because } ed = 1 \ mod \ T \\ &= m^{kT} m^1 \ mod \ N \\ &= m \ mod \ N && \cdots \ m^T = 1 \ mod \ N \text{ by Euler's theorem*} \end{aligned}$$

* If m and $N = pq$ are relatively prime, then $m^T = 1 \ mod \ N$ where $T = (p-1)(q-1)$

RSA example

- $p = 7, q = 11$
- $N = 77$
- $T = (p-1)(q-1) = 6 \times 10 = 60$
- Select public key e that is coprime to 60 → $e = 7$
- Private key $d = e^{-1} \; mod \; T = 7^{-1} \; mod \; 60 = 43$
  - Problem: Find e such that $7 \times e \; mod \; 60 = 1$
    - Can be obtained by the Extended Euclid's algorithm
  - In Python: `pow(7, -1, 60)`

RSA example

- Given
  - Secret: $p = 7, q = 11, d = 43$
  - Public: $N = 77, e = 7$
- Plaintext $m = 8$
- Encryption
  - $c = m^e \; mod \; N = 8^7 \; mod \; 77 = 57$
- Decryption
  - $m = c^d \; mod \; N = 57^{43} \; mod \; 77 = 8$

RSA example

- Given
  - Secret: $p = 7, q = 11, d = 43$
  - Public: $N = 77, e = 7$
- Plaintext $m = 8$
- Encryption
  - $c = m^e \ mod \ N = 8^7 \ mod \ 77 = 57$
- Decryption
  - $m = c^d \ mod \ N = 57^{43} \ mod \ 77 = 8$ ← Correctly decrypted!

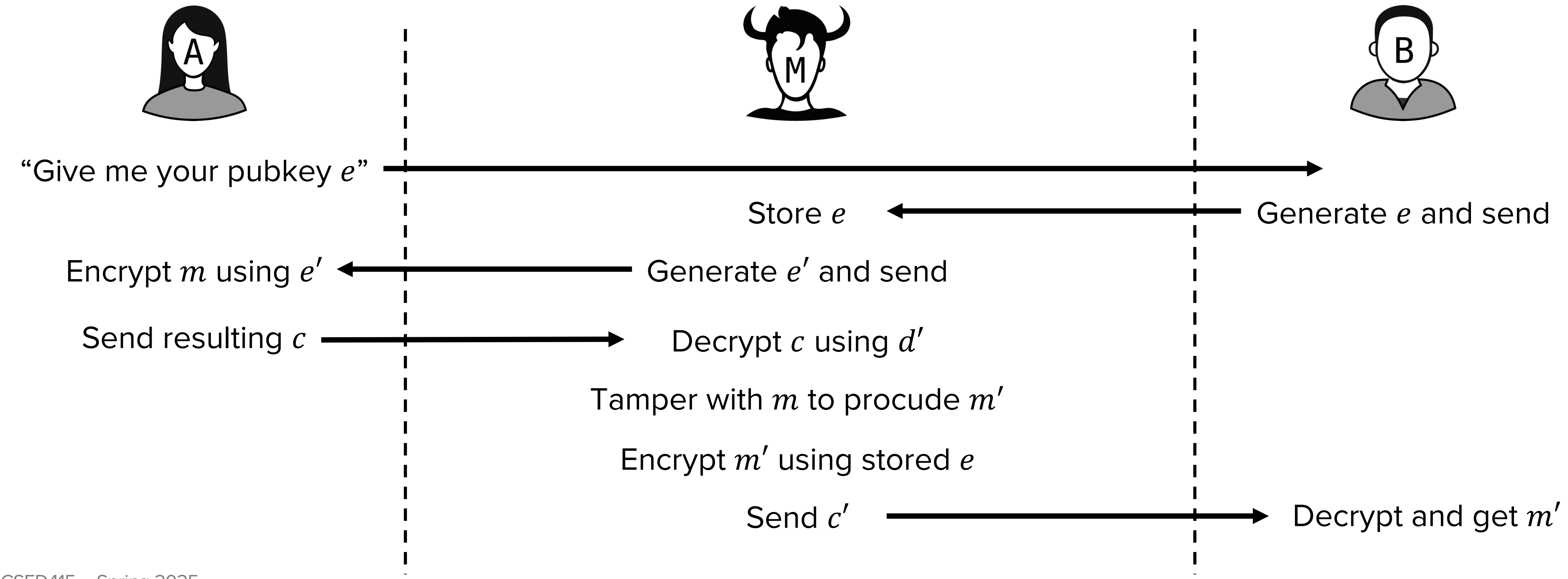
(Use modular exponentiation
for computation)

RSA security – Confidentiality

- RSA provides confidentiality based on the hardness of integer factorization problem
  - Steps for Eve to decipher c given public N and public key e,
    - To compute $m = c^d \ mod \ N$, Eve needs to find the secret key d
    - To derive $d = e^{-1} \ mod \ T$, Eve needs to find T
    - To find $T = (p - 1)(q - 1)$, Eve needs to find p and q
    - To find p and q such that $N = pq$, Eve needs to prime factorize N
    - However, there is no polynomial time algorithm that can factor a large integer N to find its prime factors p and q

RSA security – Integrity

- RSA does not guarantee integrity
  - Still susceptible to MitM attacks



Cryptography roadmap

Scheme / Goal	Symmetric Key	Asymmetric Key
Confidentiality	✅ One Time Pad (OTP) ✅ Block ciphers (DES, AES) ✅ Stream ciphers	✅ DH secure key exchange ✅ ElGamal encryption ✅ RSA encryption
Integrity & Authentication	• Message Authentication Code (MAC)	• Digital signature

Questions?