MONASH University
Information Technology

FIT2093 INTRODUCTION TO CYBER SECURITY

Week 4 Lecture

Cryptography III:

Public Key Encryption:

Part 2 – Encryption Algorithms

Principles for CONFIDENTIALITY


Outline

Public-Key Cryptography

Last week: concept and maths of PKE

This week: how public-key algorithms work

- Sharing secrets over public channels
  - Diffie-Hellman Key Exchange
- Public-key encryption
  - ElGamal
  - RSA
- Hybrid Encryption

MONASH University

Diffie-Hellman Key Exchange

Symmetric Key Encryption (SKE): Limitation

Public Key Crypto

- both sides share same secret key
- Q: how to share securely?
  - key distribution problem

Solving the Key Distribution Problem

Public Key Crypto


- share secret using public channel?
- jointly compute secret key based on public values?
- Diffie-Hellman Key Exchange (1976): 1st idea that suggests PKC



note: Malcolm Williamson (UK GCHQ) **secretly** found it in 1974!

Diffie-Hellman Key Exchange

Public Key Crypto

- Used almost everywhere
  - SSL/TLS (https): web
  - IPsec: IP packets
  - Bluetooth: personal area networks
  - 5G: mobile comms
  - IoT: internet of things/sensors
  - ...

New directions in cryptography

W Diffie, M Hellman - IEEE transactions on Information Theory, 1976 - ieeexplore.ieee.org

Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the …

☆ 99 Cited by 20315 Related articles All 156 versions

☆ 99 Cited by 18999 Related articles All 153 versions

Diffie-Hellman KE: Design Strategy

Public Key Crypto

- **Easy** for **good guys** (Tx, Rx) to:
  - generate own public key pk
  - compute K

  &
- **Infeasible** for **attacker** to:
  - get other's sk from pk
  - compute K

Diffie-Hellman KE: What

Public Key Crypto

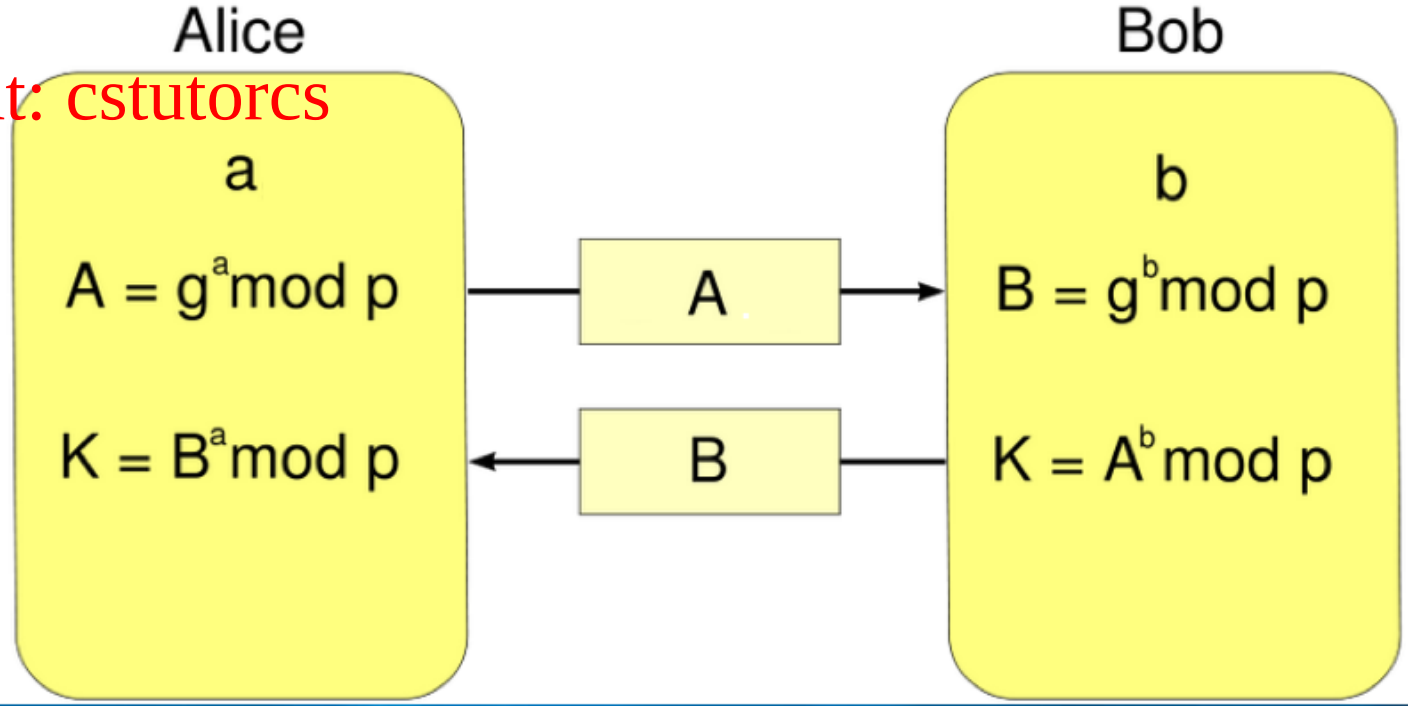
- p and g are public parameters, p is a prime number

- a is private key of Alice, b is private key of Bob
- A is public key of Alice, B is public key of Bob

- A = f(a), B = f(b): private key & public key have special relationship

- K = shared secret key between Alice and Bob = g(A, b) = g(B, a)

- Functions f and g are modular exponentiation – efficiently computable



Diffie-Hellman KE: Why It works

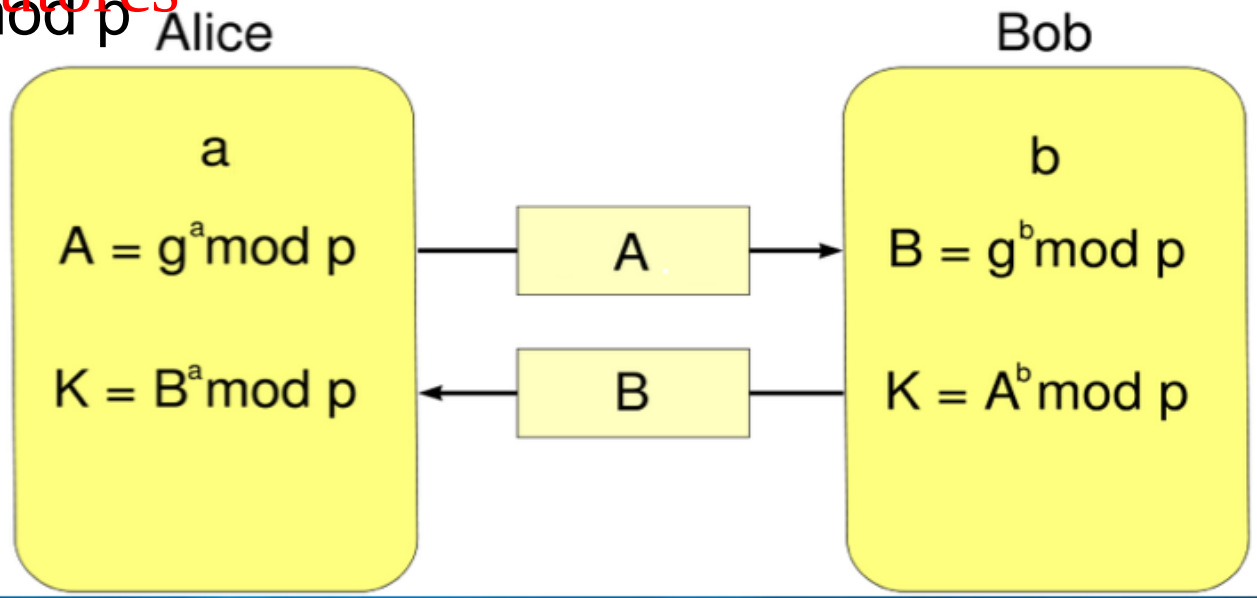
Public Key Crypto

- A = f(a) = g^a mod p; B = f(b) = g^b mod p: sk and pk have special relationship
- K = shared secret key between Alice and Bob
- **Q:** Why do Alice and Bob compute the same K when they use different formulas?

For Alice: K = B^a mod p = $(g^b)^a$ mod p = g^{ba} mod p

For Bob: K = A^b mod p = $(g^a)^b$ mod p = g^{ab} mod p

Both can easily compute K, using
modulo exponentiation & their own private key

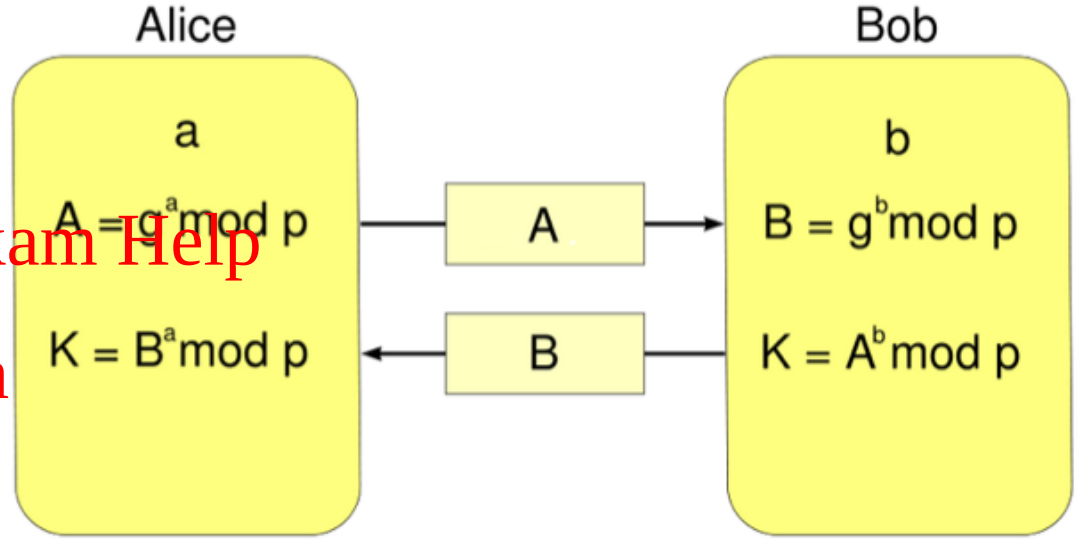


Diffie-Hellman KE: Why it Solves the Problem

Public Key Crypto

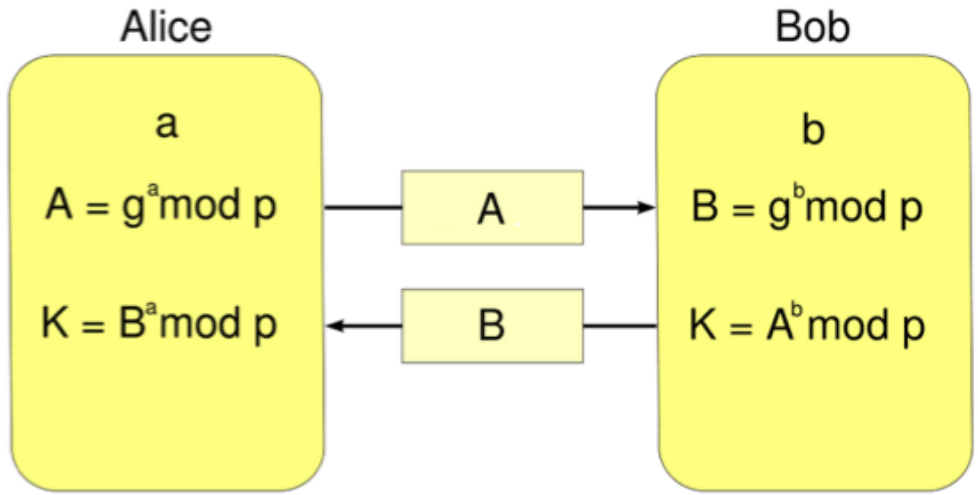
- The only **secret** of Alice is **a**
- while **p** and **g** are **public**

- **A** is sent over channel, so **public** too:
  - **no secret is sent** over channel,
  - so no private key distribution problem



Diffie-Hellman KE: Example I

Public Key Crypto



- **p** = 23, **g** = 5
- *Alice*

choose **a** = 6 (private)

compute $A = g^a \bmod p$

$= 5^6 \bmod 23$

$= (5^2 x 5)^2$ **mod 23**

= $(2x5)^2 \bmod 23$

= $8 \pmod{23}$

Bob

choose **b** = 15 (private)

compute $B = g^b \bmod p$

$= 5^{15} \bmod 23$

$= ((5^2\mathbf{x5})^2\mathbf{x5})^2 x5$ **mod 23**

= $(2x5)^2\mathbf{x5})^2 x5$ **mod 23**

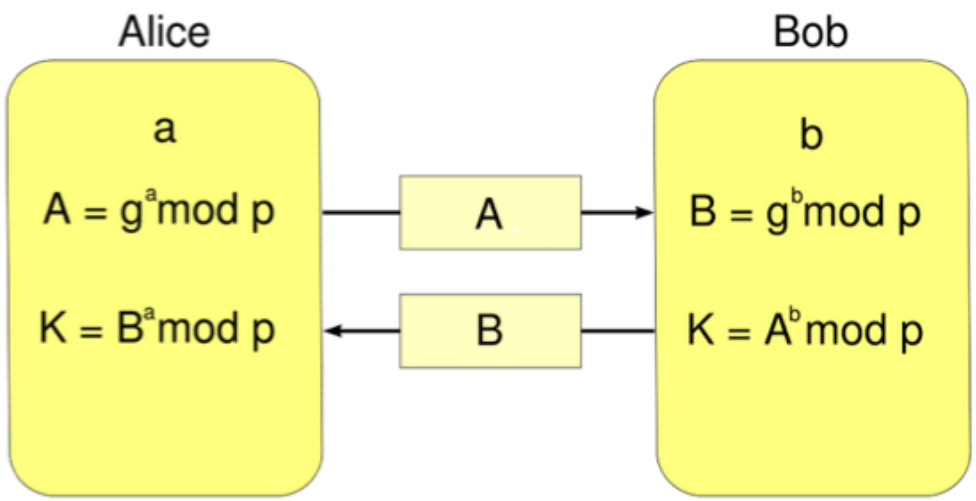
$= (8\mathbf{x5})^2 x5$

= $13 \times 5 \pmod{23}$

= 19

Diffie-Hellman KE: Example I

Public Key Crypto



Alice (**a** = 6) Bob (**b** = 15)

A = 8

B = 19

compute **K** = B^a mod p

= 19^6 mod 23

= $(19^2 x 19)^2$ (mod 23)

= $((-7) \times (-4))^2$ (mod 23)

= 5^2 (mod 23)

= 2 (mod 23)

compute **K** = A^b mod p

= $((8^2 \mathbf{x8})^2 \mathbf{x8})^2 x8$ mod 23

= $(((-5)\mathbf{x8})^2 \mathbf{x8})^2 x8$ mod 23

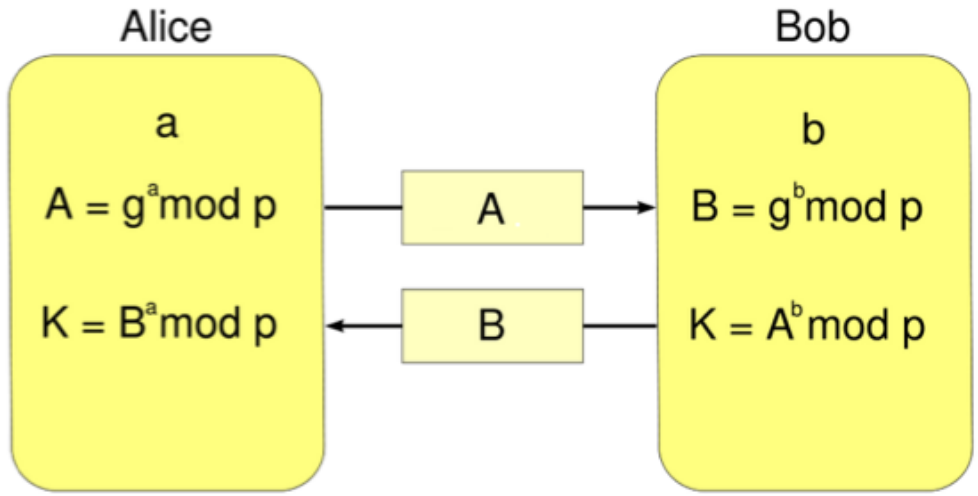
= $(13\mathbf{x8})^2 x8$ mod 23

= 6x8 mod 23

= 2 (mod 23)

Diffie-Hellman KE: Example II

Public Key Crypto



Suppose **p** = 17, **g** = 7, Alice's private key a = 4.

Q:

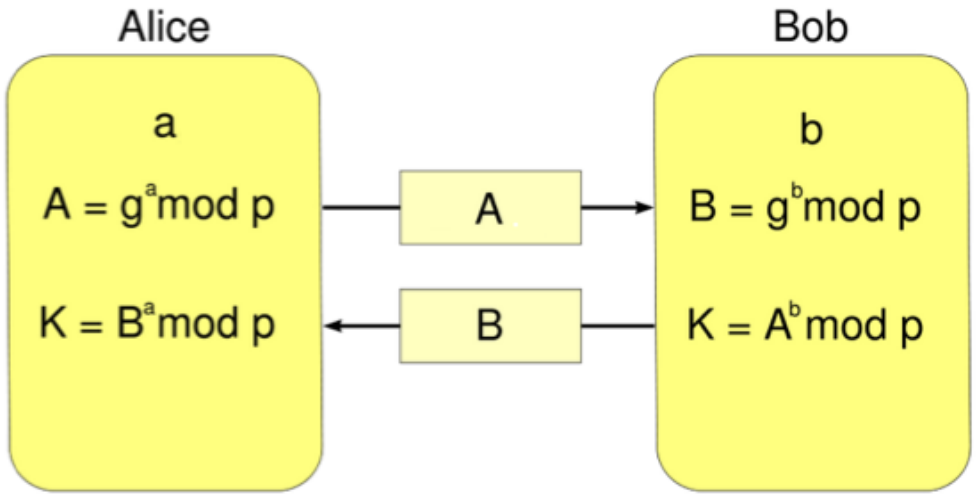
1) What is Alice's public key A?
2) If Alice receives Bob's public key B = 5, what is the shared key K between Alice and Bob?

Activity (2 mins)

1) Click the latest link in the Zoom chat
2) Add your question response to the Ed forum

Diffie-Hellman KE: Security

Public Key Crypto

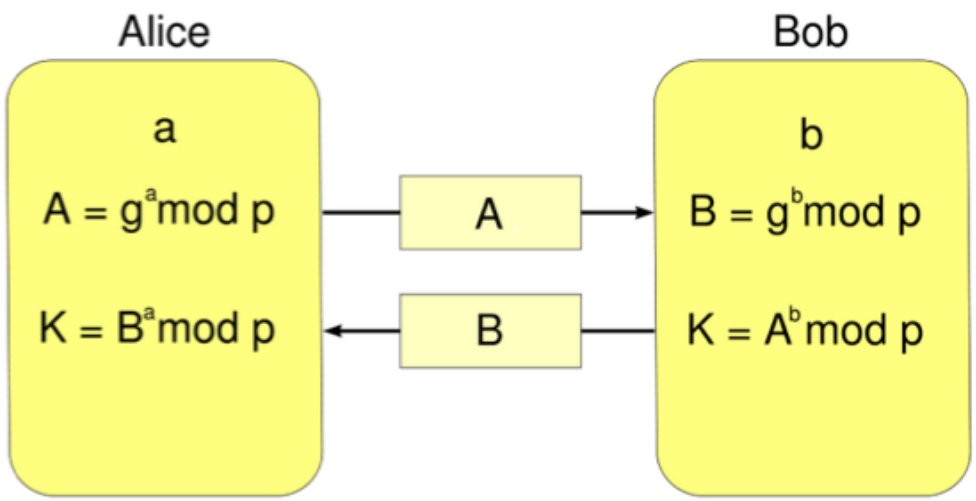


- the **secrets**:
  - shared key: **K**
  - each person's private key: **a** , **b**
- to attack (compute **K**), attacker needs to know **a** or **b**

- **Q**: can attacker get **a**?
  - can get/compute **a** from **p**, **g**, **A**?
  - **Discrete Log Problem (DLP) - computationally infeasible (large a,b,p)**

- **Q**: can attacker get **K**?
  - can get/compute **K** without knowing **a** , **b** but only knowing **p**, **g**, **A**?
  - **Diffie-Hellman Problem (DHP) – computationally infeasible (large a,b,p)**

Diffie-Hellman KE: Security



Public Key Crypto

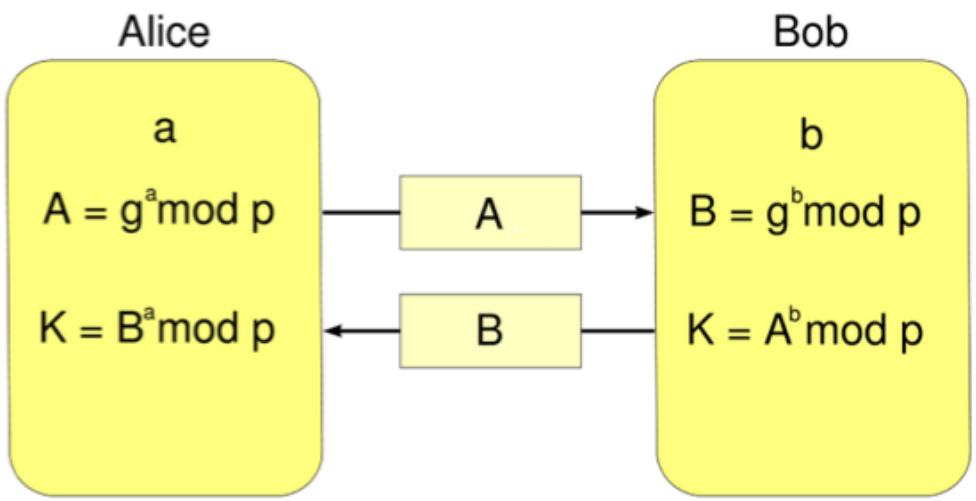
- The secrets:
  - shared key: **K**
  - each person's private key: **a** , **b**

- **Discrete Log Problem (DLP)- infeasible**:
  - given **A** = g^a mod p is known, **g** and **p** known,
  - compute the exponent/discrete log **a**

- **Diffie-Hellman Problem (DHP) - infeasible**:
  - given **A** = g^a mod p is known, **B** = g^b mod p, **g** and **p** also known,
  - compute K = g^{ab} mod p

Diffie-Hellman KE: Security

Public Key Crypto



Tx
<A, a>

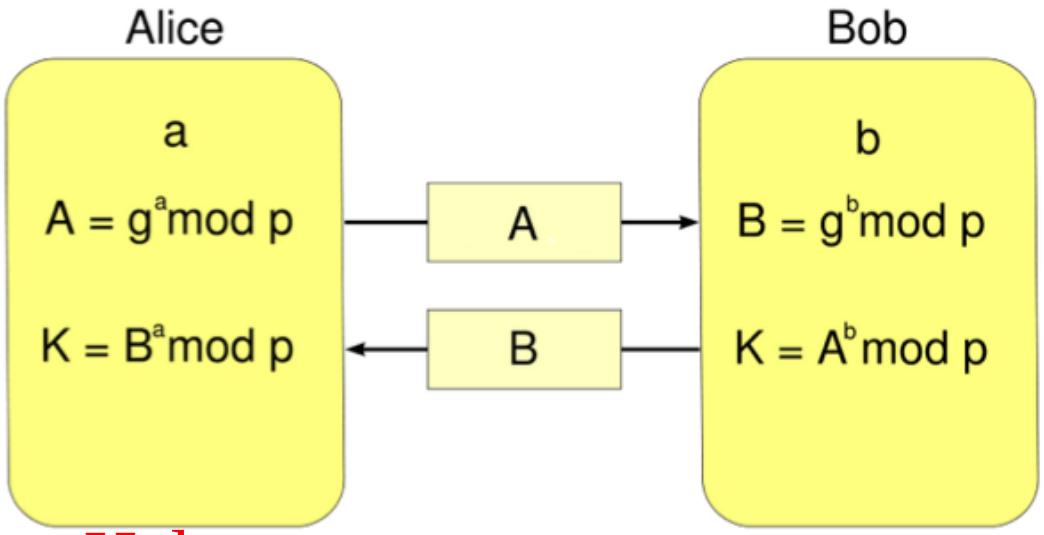
Rx
<B, b>

- to share key with Rx,
  - use Rx' public key B

- **Q**: what if public key B replaced with attacker's pub key M?
- **Integrity attack (also known as Man-in-the-Middle attack)** on pk value is devastating

Diffie-Hellman KE: MIM Attack

Public Key Crypto



Alice

choose a (private)

compute A = g^a mod p

Attacker

choose m

compute M = g^m mod p

Bob

choose b (private)

compute B = g^b mod p

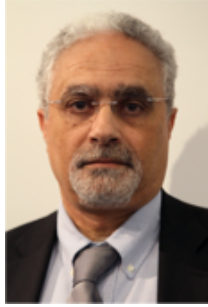
Attacker shares a key with Alice and a key with Bob - can read their communications – we'll return to this problem later (use authenticated certificates).

MONASH University

El Gamal PKE

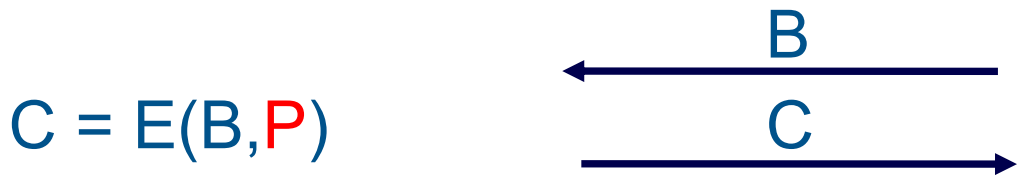
From Diffie-Hellman to ElGamal

Public Key Crypto



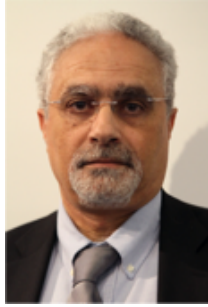
- In DHKE, **both** Alice/Bob **publish** their public key **A** and **B**

- In public-key encryption, Bob (Rx) **publishes** his public key **B**, Alice generates a ciphertext based on Bob's public key **B**



From Diffie-Hellman to ElGamal

Public Key Crypto



- But can convert DH **key exchange** to public-key **encryption** (PKE) [ElGamal PKE, 1985]

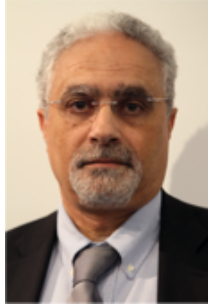
- Idea: Alice **includes** "**ephemeral**" public key "**A** **in** her **ciphertext** to Bob

Alice (Tx) Bob (Rx)

C = E(B,P) ⟶ A, C

From Diffie-Hellman to ElGamal

Public Key Crypto



Alice

Bob

choose g, p, **b** & **B** = g^{b} mod p

(private key: **b**, public key: **B**)

lookup public key of Bob: **B**

choose random ephemeral (one-time) **a**

compute **A** = g^{a} mod p

(***ephemeral private key***: **a**, ***ephemeral public key***: **A**)

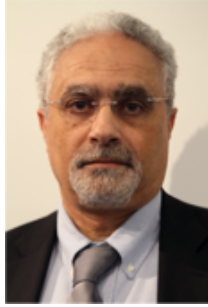
compute ***ephemeral shared key*** **K** = B^{a} mod p = g^{ab} mod p

A, C = Enc(K,P)

(Enc: symmetric key encryption)

From Diffie-Hellman to ElGamal

Public Key Crypto



Alice

Bob

A, C = Enc(K,P)

compute **K** = $\mathbf{A}^{\mathbf{b}}$ mod p = $g^{\mathbf{ab}}$ mod p

decrypt C using K to get P, i.e.

P = Dec(K,C)

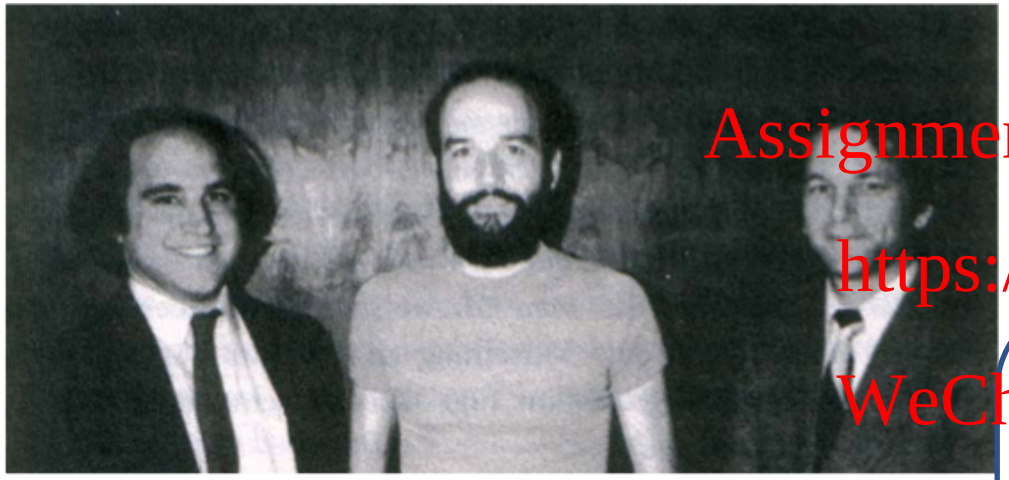
MONASH University

RSA PKE

RSA Public-key Encryption (PKE)

Public Key Crypto

1st Public-Key Encryption/Cipher [1977]



- Ron **R**ivest
- Adi **S**hamir
- Leonard **A**dleman

Note: Secretly discovered by Clifford Cocks (UK GCHQ) in 1973 following PKC concept discovery by James Ellis (UK GCHQ) in 1969

RSA Public-key Encryption (PKE)

Public Key Crypto

- **KeyGen**
  - choose two distinct large primes **p** and **q**
  - compute the **modulus** **n** = pq
  - compute the Euler's totient function **ø**(n) = (p–1)(q–1)
  - choose an integer e coprime to ø: **e** is the **public key**
  - compute d = e^{-1} mod ø(n) as e's inverse: **d** is the **private key**
    - *Note*: **e×d ≡ 1 mod ø(n)** since d is the multiplicative inverse of e
- **Recall (last week): All of the above operations are efficiently computable even for large numbers.**

RSA Public-key Encryption (PKE)

Public Key Crypto

- **KeyGen**
  - outputs key pair: public key (**e, n)**, private key **d**

- **Encrypt**
  - $c = m^e \bmod n$ (*modular exponentiation*)
- **Decrypt**
  - $m = c^d \bmod n$ (*modular exponentiation*)

RSA PKE: KeyGen Example

Public Key Crypto

- choose two primes: p = 5, q = 11
- compute the modulus n = p x q = 55
- compute ø(n) = (p–1) x (q–1) = 4 x 10 = 40
- find out two numbers e = 3 & **d = 27** which satisfy

  (3 * **27**) mod 40 = 1

- Bob's public key: (e, n) = (3, 55)
  - encryption: modular exponentiation
- Bob's **private** key: (**d**, n) = (**27**,55)
  - decryption: modular exponentiation

RSA PKE: Encrypt Example

Public Key Crypto

- Alice has a **message** m=13 to be sent to Bob

- Compute the **ciphertext** c as:

$$c = m^e \pmod{n}$$
$$= 13^3 \pmod{55}$$
$$= (169 \pmod{55} \times 13 \bmod 55) \bmod 55$$
$$= 4 \times 13 \bmod 55$$
$$= 52 \pmod{55}$$

RSA PKE: Decrypt Example

Public Key Crypto

- Bob receives c=52 from Alice

- Compute the **plaintext** as: (use $27 = 11011_2$)

$$m = 52^{27} \pmod{55}$$
$$= ((((-3)^2 x (-3))^2)^2 x (-3))^2 x (-3) \pmod{55}$$
$$= ((14)^2 x (-3))^2 x (-3) \pmod{55}$$
$$= 17^2 \times (-3) \pmod{55}$$
$$= 13$$

RSA Encryption – Example 2

Public Key Cryptography

Suppose Bob's public key is (e = 7, n = 33).

Q:

1) What is Bob's private key d?
2) If Bob receives a ciphertext c = 8 from Alice, what message m does Bob decrypt?

Activity (2 mins)

1) Click the latest link in the Zoom chat
2) Add your question response to the Ed forum

RSA PKE: Why Works?

Public Key Crypto

- Recall: e x d ≡ 1 mod ø(n)

- **But recall from last week:**
- **Implication of Euler's Theorem**: $M^y \bmod n = M$ if $y \bmod ø(n) = 1$

- **Here, $y = e \times d$: so $M^y \bmod n = M^{e \times d} \bmod n = M$.**

- Encrypt: $c = m^e \bmod n$
- Decrypt (why it works?)
  - $c^d \bmod n = m^{e \times d} \bmod n = m \pmod n$
  - by implication of Euler's Theorem!

RSA PKE: Security

Public Key Crypto

- the **secrets**: private key **d** , ø , p , q , plaintext **m**
- The known info to attacker: known values pub key (**e**, **n**), ciphertext **c**

- Q1: Can attacker efficiently compute **p, q** from public key (**e**,**n**)? Why/Why not?
- Q2: What about computing ø? Why/Why not?

- Q3: What about computing **d**? Why/Why not?

- Q4: What about computing **m**? Why/Why not?

RSA PKE: Security

Public Key Crypto

- **Attack approaches**
  - since knows m in range [1, (n–1)], brute force through all n possible values (exp time, infeasible),

  or
  - compute **d** based on **d** = e^{-1} mod ϕ(n)
    - but need to compute ϕ(n)=(p–1)x(q–1) which needs to know p and q,
    - i.e. **need to factor** n - Integer Factorisation Problem – infeasible (last week).
    - *(vs honest users just do efficient exponentiation)*

Q: Is above RSA encryption method secure if m is chosen from a small set of possible messages (e.g. m = 10 (“yes”), m = 20 (“no”))? Why/Why not?

PKE Key Lengths vs Security

Public Key Crypto

Date	Minimum of Strength	Symmetric Algorithms	Factoring Modulus	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve
(Legacy)	80	2TDEA*	1024	160	1024	160
2016 - 2030	112	3TDEA	2048	224	2048	224
2016 - 2030 & beyond	128	AES-128	3072	256	3072	256
2016 - 2030 & beyond	192	AES-192	7680	384	7680	384
2016 - 2030 & beyond	256	AES-256	15360	512	15360	512

All key sizes are provided in bits. These are the minimal sizes for security.

Fig. source: https://www.keylength.com/

RSA PKE vs ElGamal PKE: Security Summary

Public Key Crypto

	RSA	ElGamal
pk & sk relations		
Problem for attacker	Integer Factorisation Problem	Discrete Logarithm Problem

Integrity Attack on PKE Key Directory

Public Key Crypto

- to encrypt for Riley (Rx):
  - Tania (Tx) uses Riley's (Rx) public key pk_R
- Q: Why is RSA PKE insecure if attacker Marvin can replace pk_R received by Tania with pk_M (pub key of Marvin)?

Solving the PKE Key Integrity Problem

Public Key Crypto

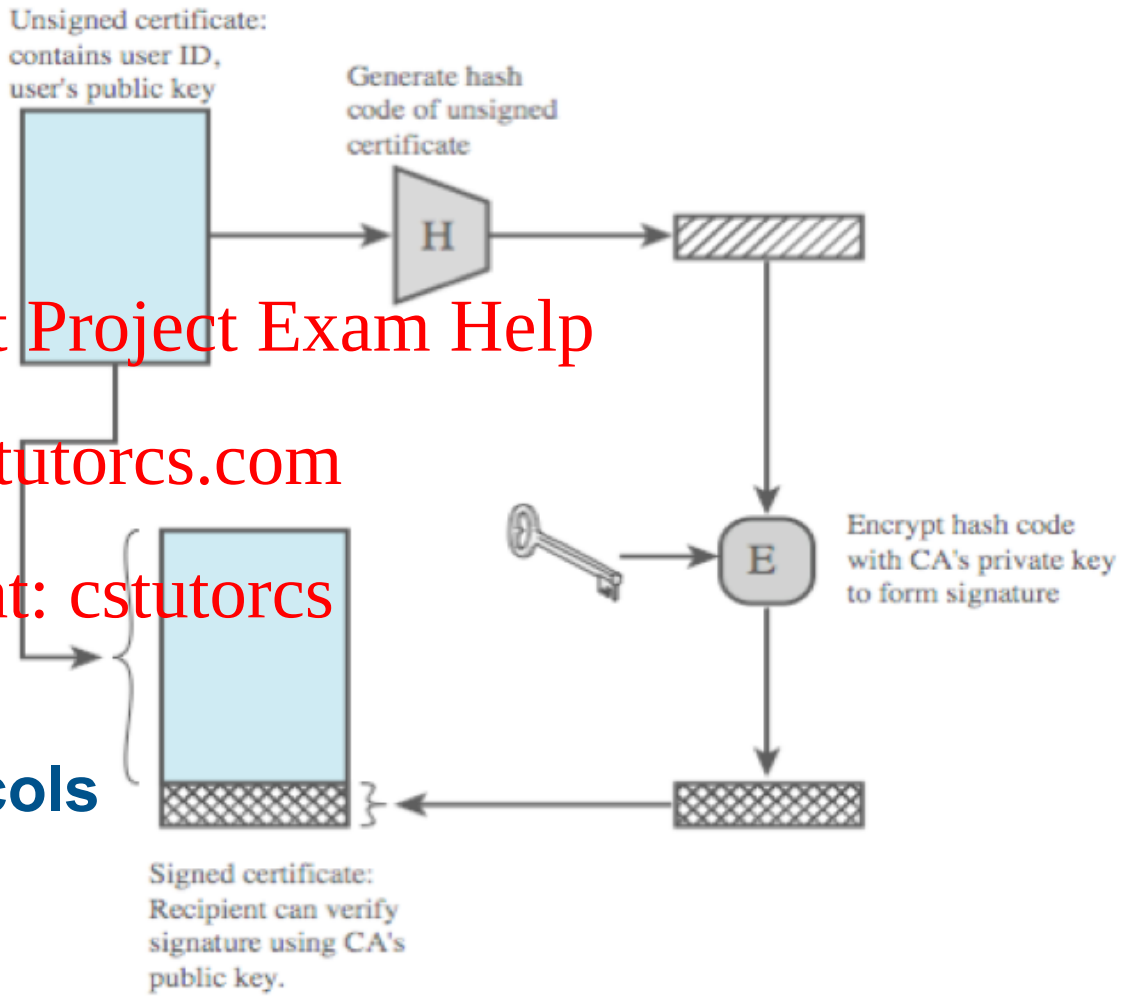
- **digital certificates**

Idea:

- trusted certification authority authenticates public keys of user
- Anyone can verify user's certificate to check authenticity of user's pk

→ prevent MIM/Integrity attacks.

(we'll revisit in the Security protocols Week – after we study digital signatures)



PKE or SKE?

Public Key Crypto

	PKE	SKE
Keys	Each user has a public key pk & a private key sk; pk & sk are mathematically related	Each user shares a secret key k with another
Do, Undo	Do with pk, undo with sk	Do and undo with k
Problem	No Private Key Distribution Problem	Private Key Distribution Problem
Speed	Slower	Faster
Q	*better?*	*better?*

Hybrid Encryption (a.k.a. Digital Envelope)

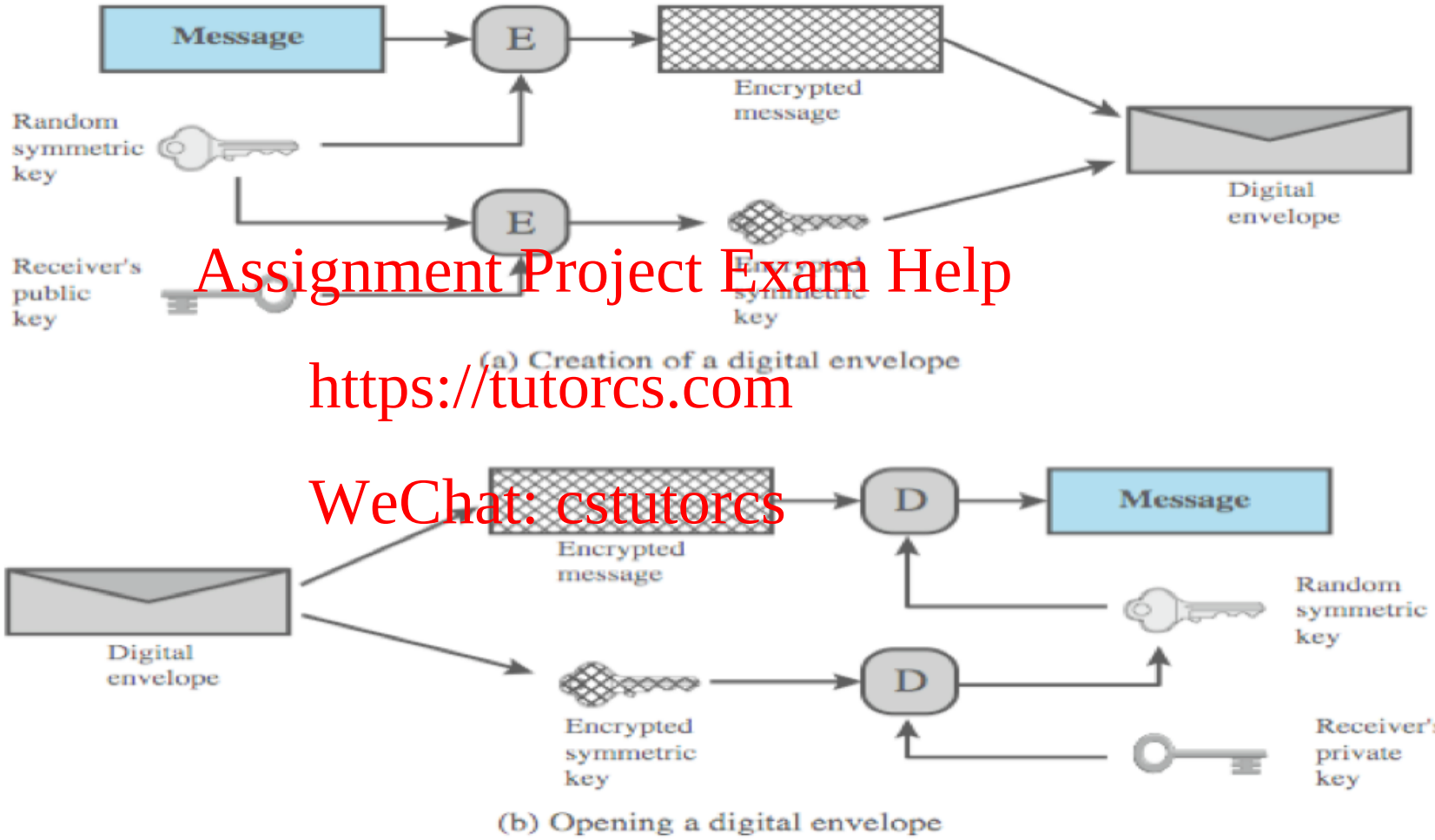
Public Key Crypto

In practice, we can combine best features of PKE and SKE

- use a PKE
  - to distribute/transport the key used for SKE
    - key small so does not matter much that it is slow
    - solves private key distribution problem
- use an SKE
  - to encrypt and decrypt messages:
    - can encrypt large messages fast

Hybrid Encryption: Digital Envelope

Public Key Crypto



Further Reading

Public Key Crypto

- Chapter 21 of the reference book: Computer Security: Principles and Practice" by William Stallings & Lawrie Brown, Prentice Hall, 2015
- Optional deeper reading:
  - Chapter 10 of the reference book: Introduction to Modern Cryptography, by Jonathan Katz and Yehuda Lindell, Chapman & Hall/CRC, 2008
  - PKCS #1 v2.2: RSA Cryptography Standard (EMC Corporation), available at https://www.emc.com/collateral/white-papers/h11300-pkcs-1v2-2-rsa-cryptography-standard-wp.pdf