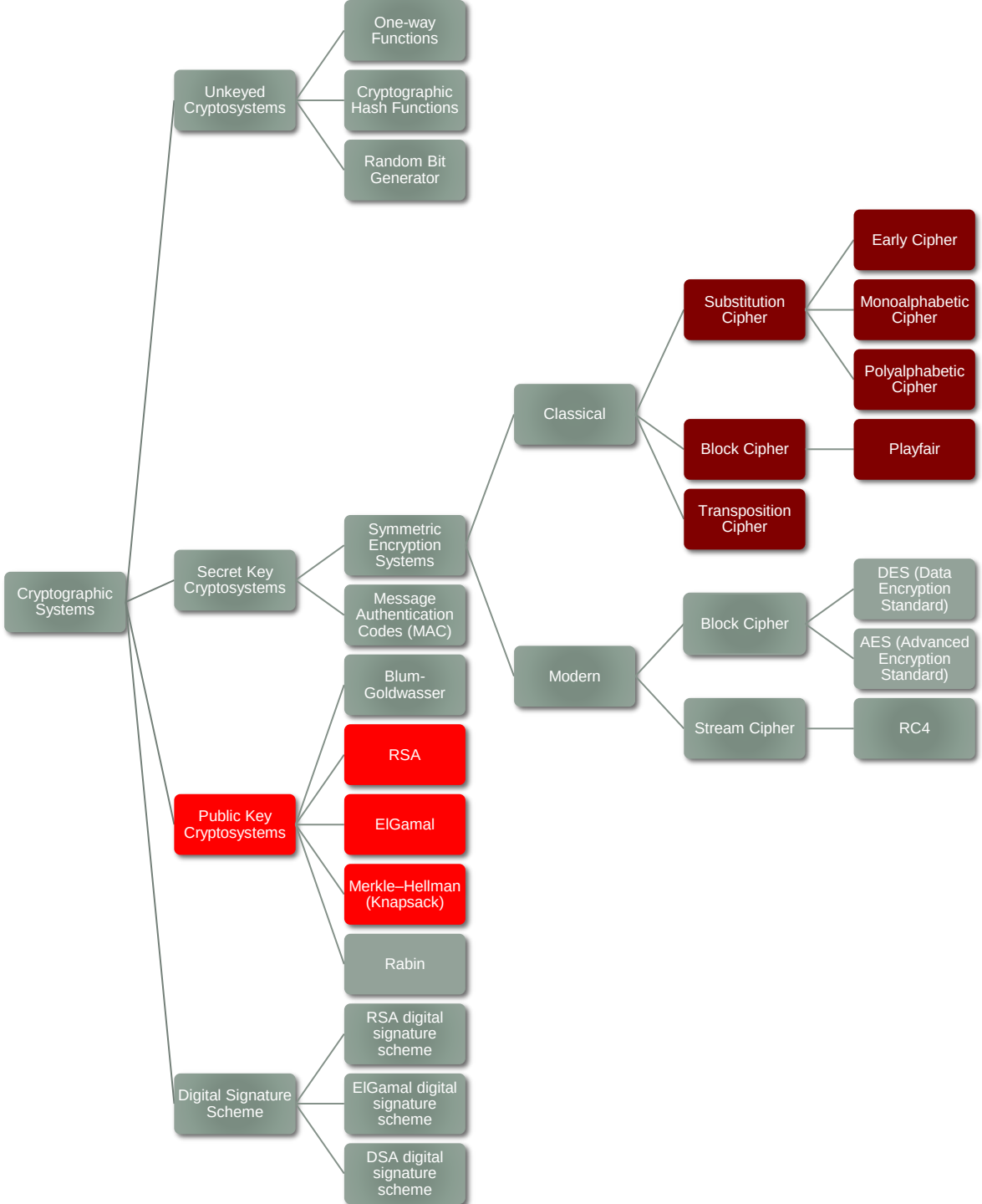
TUTORIAL 2

CSCI361 – Computer Security

Sionggo Japit

sjapit@uow.edu.au

12 February 2024


Cryptographic Systems
Unkeyed Cryptosystems
One-way Functions
Cryptographic Hash Functions
Random Bit Generator
Secret Key Cryptosystems
Symmetric Encryption Systems
Classical
Substitution Cipher
Early Cipher
Monoalphabetic Cipher
Polyalphabetic Cipher
Block Cipher
Playfair
Transposition Cipher
Modern
Block Cipher
DES (Data Encryption Standard)
AES (Advanced Encryption Standard)
Stream Cipher
RC4
Message Authentication Codes (MAC)
Public Key Cryptosystems
Blum-Goldwasser
RSA
ElGamal
Merkle–Hellman (Knapsack)
Rabin
Digital Signature Scheme
RSA digital signature scheme
ElGamal digital signature scheme
DSA digital signature scheme

Public key Cryptosystems

- Introduced in 1976 by Diffie and Hellman at Stanford University.
- The idea:
  - Two keys, a private key and public key are used.
  - A recipient (let's say Bob) can announce (make public) his public key, and all senders (public) who wants to send Bob, the recipient, secret messages, can send the messages by encrypting the messages using Bob's public key.

Public key Cryptosystems

- Bob and all senders need not have to have a predefined (or pre-agreed) secret key; i.e., no secret channel is required.
- A required condition is that the private key (known only to the recipient, in this case Bob) is not computationally feasible to be computed or derived from the public key.

Public key Cryptosystems

- Public key cryptosystems use trapdoor functions to ensure that the private key cannot be derived or computed easily.
- A public key cryptosystem can provide:
  - **Confidentiality** – because only the receiver can decrypt and find the plaintext, and
  - **Authenticity** – because only the sender can create such a cryptogram (encrypted data).

Public key Cryptosystems

Three broad category of applications:

- Public-key cryptosystem can be used as:
  - Encryption/decryption system
    - When used as an encryption/decryption system, the sender encrypts a message with the recipient's public key and sends the encrypted message to the intended recipient. Upon receiving the encrypted message, the recipient will then use his/her private key to decrypt the encrypted message to retrieve the plaintext. Only the intended recipient can retrieve the plaintext because only he/she knows the private key.

Public key Cryptosystems

- Digital signature
  - When used as a digital system, the sender "signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message. When the intended recipient received the encrypted message, the recipient use the sender’s public key to decrypt and retrieve the signed message.
- Key exchange
  - When used as a key exchange system, two sides cooperate to exchange a session key. Several different approaches are possible, involving the private key(s) of one or both parties

Public key Cryptosystems

Two public parameters:

- A prime number p, and
- A generator g such that

$$\forall y \in [1; p-1] : \exists k; y = g^k \bmod p$$

An example of a generator in z_p^*

Generator in Z^*_{11}: $\forall y \in [1; p-1] : \exists k; y = g^k \bmod p$

k

g

$\forall y$

	1	2	3	4	5	6	7	8	9	10
1	1	1								
2	**2**	**4**	**8**	**5**	**10**	**9**	**7**	**3**	**6**	**1**
3	3	9	5	4	1	3				
4	4									
5	5	3	4	9	1	5				
6	**6**	**3**	**7**	**9**	**10**	**5**	**8**	**4**	**2**	**1**
7	**7**	**5**	**2**	**3**	**10**	**4**	**6**	**9**	**8**	**1**
8	**8**	**9**	**6**	**4**	**10**	**3**	**2**	**5**	**7**	**1**
9	9	4	3	5	1	9				
10	10	1	10							

2, 6, 7 and 8 are generators in Z^*_{11}.

Public key Cryptosystems

How it works?

1. Alice generates a private random integer a
   Bob generates a private random integer b
2. Alice generates her public value $g^a \mod p$
   Bob generates his public value $g^b \mod p$
3. Alice computes $g^{ab} = (g^a)^b \mod p$
   Bob computes $g^{ba} = (g^b)^a \mod p$
4. Both now have a shared secret k since $k = g^{ab} = g^{ba}$

This protocol is also known as Diffie-Hellman key exchange.

Public key Cryptosystems

Some popular public key cryptosystems are:

- RSA (Rivest, Shamir and Adleman (1978)) - based on factorization problem
- ElGamal (1984) - based on discrete logarithm problem
- Merkle-Hellman (1977) - based on knapsack problem

Public key Cryptosystems

Other public key cryptosystems are:

- Rabin – based on factorization
- Blum-Goldwasser – based on factorization problem

Public key Cryptosystems

Weakness

- Vulnerable to the man-in-the-middle attack

Public key Cryptosystems

How does Man-in-the-Middle attack work against public-key based key distribution?

Man-in-the-Middle attack can be carried out as follow:

- An attacker Eve computes her own public key as $y_E = g^{k_E} \bmod p$.
- Eve then intercepts the public keys sent by the sender Adam ($y_A = g^{k_A} \bmod p$), and receiver Barbie ($y_B = g^{k_B} \bmod p$).

Public key Cryptosystems

- Eve then replaces both the public key y_A and y_B with y_E and sends it to both Adam and Barbie. As a result, Adam receives y_E instead of y_B and Barbie also receives y_E instead of y_A.

- Not knowing they are receiving what they are not suppose to receive, both Adam and Barbie will use y_E as the key to encrypt. The encrypted message by Adam to Barbie will be $m_A^{y_E} \bmod p$ and the encrypted message by Barbie to Adam will be $m_B^{y_E} \bmod p$.

Public key Cryptosystems

- Eve can decrypt both $m_A^{y_E} \bmod p$ and $m_B^{y_E} \bmod p$ using her own private key to obtain m_A and m_B.

- Eve can also encrypt $m_A^{y_B} \bmod p$ and forward it to Barbie, and encrypt $m_B^{y_A} \bmod p$ and foward it to Adam.

- In this way, Eve can read all messages send by Adam and Barbie without Adam and Barbie knowing it.