CPR E 431

BASICS OF INFORMATION SYSTEM SECURITY

Introduction to Cryptography Tools

Public Key Encryption

Video Summary

- What is Public/Asymmetric Key Encryption
- Principles of Public-key Encryption
- Key Generation
- Public-key Encryption Assumptions
- Public-key Encryption Requirements

Public-Key Encryption Structure

Publicly proposed by Diffie and Hellman in 1976

Based on mathematical functions

Asymmetric

- Uses two separate keys
- Public key and private key
- Public key is made public for others to use

Some form of protocol is needed for distribution

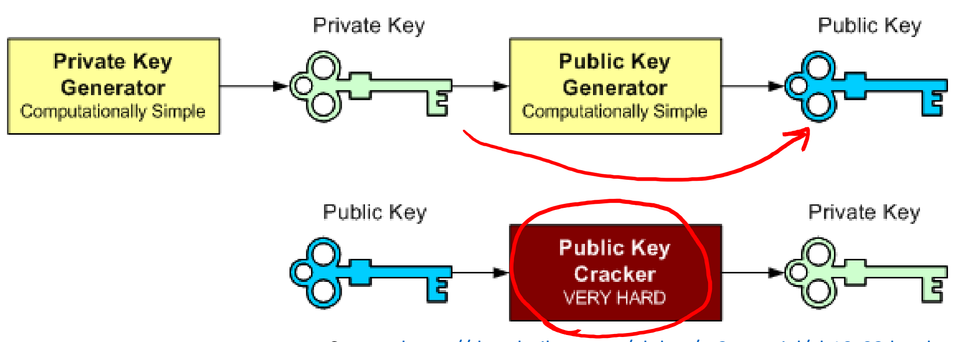
Public-Key Encryption Structure

- Two different keys are used interchangeably to encrypt/decrypt the data
- The keys always come in pairs

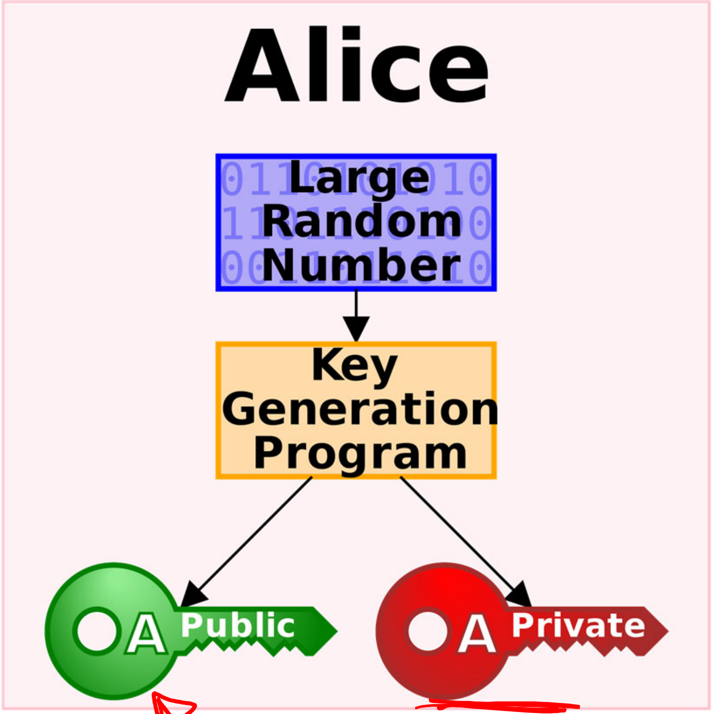
PU_A PR_A

- Each user is having two keys (one public and one private)

Keys Generation



Source: https://docs.huihoo.com/globus/gt3-tutorial/ch10s03.html

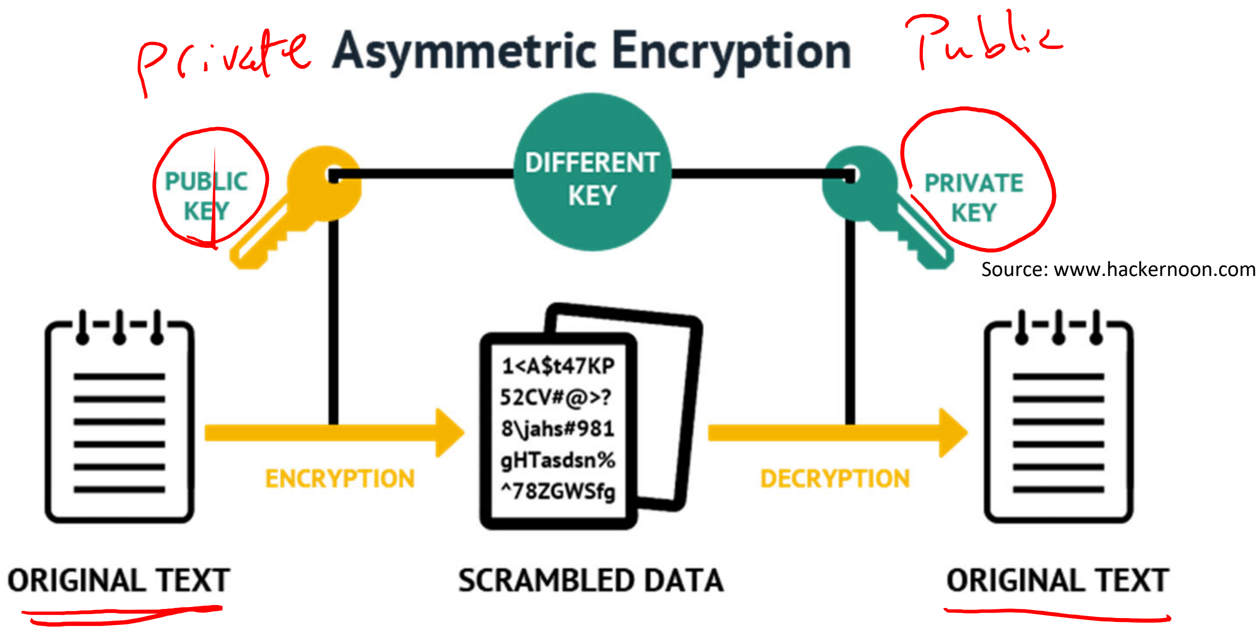


Source: www.wikipedia.org

Public and Private Keys

If you encrypted with public key
then
decrypt with private key

If you encrypted with private key
then
decrypt with public key



Source: www.hackernoon.com

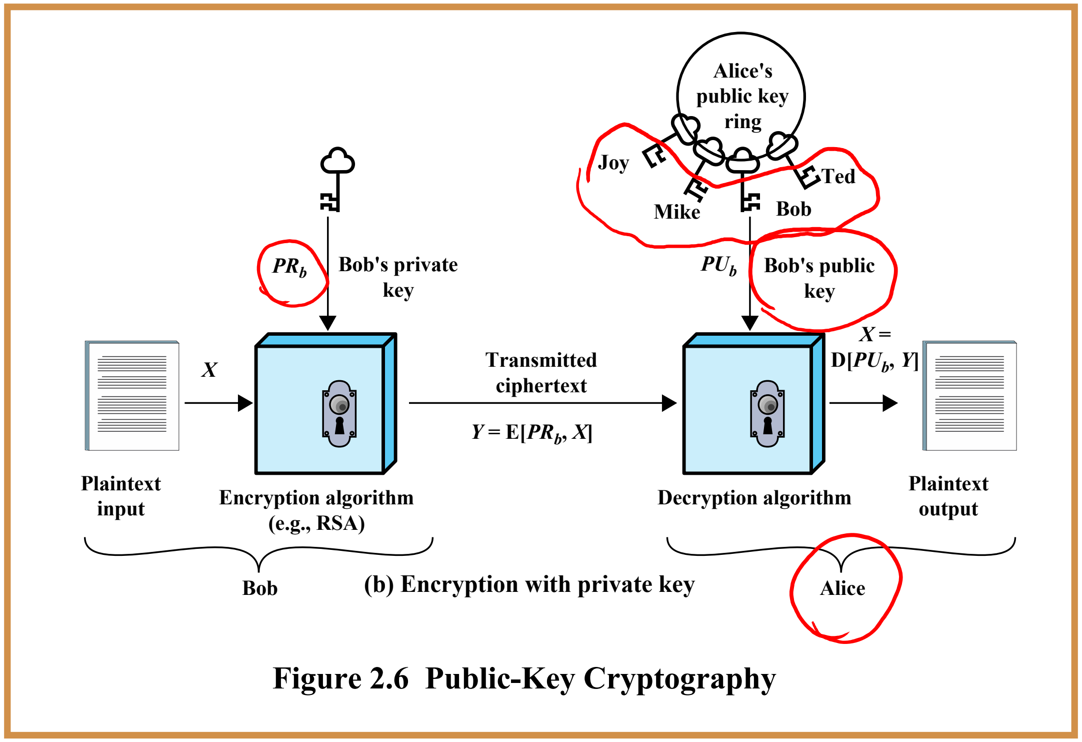
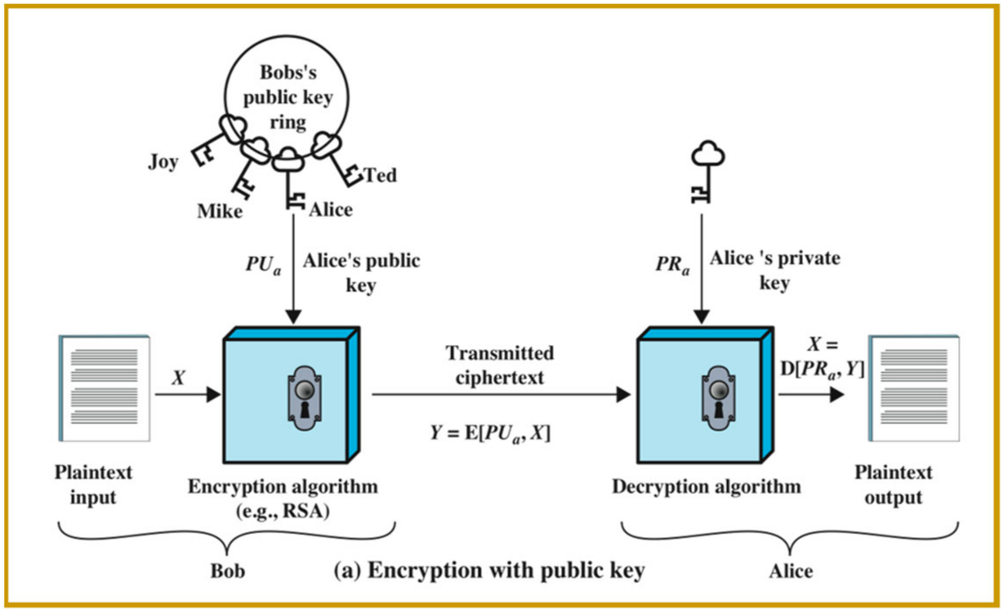


Figure 2.6 Public-Key Cryptography

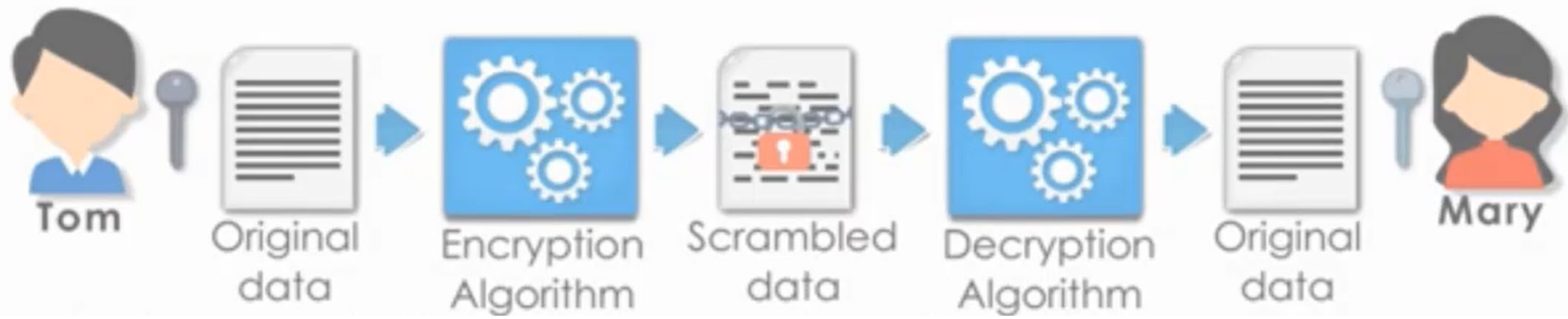
- User encrypts data using his or her own private key
- Anyone who knows the corresponding public key will be able to decrypt the message



- User encrypts data using a certain public key

- Anyone who knows the corresponding private key will be able to decrypt the message

Tom wants to send an encrypted message to Mary


Tom
Original data
Encryption Algorithm
Scrambled data
Decryption Algorithm
Original data
Mary

Assumptions: Public Key Encryption

- There is a pair of keys, public (PU) and private (PR). One key from the pair is used for encryption, the other is used for decryption. Each entity has their own pair, e.g. (PU_A, PR_A).
- Encrypting a plaintext message, M, with a key, produces ciphertext C, e.g. $C = \mathrm{E}(PU_A, M)$.
- Decrypting ciphertext with the correct key will produce the original plaintext. The decrypter will be able to recognise that the plaintext is correct (and therefore the key is correct). E.g. $M = \mathrm{D}(PR_A, C)$.

Requirements of Public-Key Cryptography

1. Computationally easy for B to generate pair (PU_b, PR_b)
2. Computationally easy for A, knowing PU_b and message M, to generate ciphertext:

$$\boxed{C = \mathrm{E}(PU_b, M)}$$

3. Computationally easy for B to decrypt ciphertext using PR_b:

$$M = \mathrm{D}(PR_b, C) = \boxed{\mathrm{D}[PR_b, \mathrm{E}(PU_b, M)]}$$

4. Computationally infeasible for attacker, knowing PU_b and C, to determine PR_b
5. Computationally infeasible for attacker, knowing PU_b and C, to determine M
6. (Optional) Two keys can be applied in either order:

$$M = \mathrm{D}[PU_b, \mathrm{E}(PR_b, M)] = \mathrm{D}[PR_b, \mathrm{E}(PU_b, M)]$$

Asymmetric Encryption Algorithms

RSA (Rivest, Shamir, Adleman)	Developed in 1977	Most widely accepted and implemented approach to public-key encryption	Block cipher in which the plaintext and ciphertext are integers between 0 and n-1 for some n.
Diffie-Hellman key exchange algorithm	Enables two users to securely reach agreement about a shared secret that can be used as a secret key for subsequent symmetric encryption of messages	Limited to the exchange of the keys	
Digital Signature Standard (DSS)	Provides only a digital signature function with SHA-1	Cannot be used for encryption or key exchange	
Elliptic curve cryptography (ECC)	Security like RSA, but with much smaller keys		

Video Summary

- What is Public/Asymmetric Key Encryption
- Principles of Public-key Encryption
- Key Generation
- Public-key Encryption Assumptions
- Public-key Encryption Requirements