# Asymmetric Cryptography

Network Security (NETSEC)

Yuhong Li

2022-01-28

Stockholm University

# Outline

Stockholm University

- Principles of asymmetric cryptography
  - Model
  - Applications
  - Trapdoor function
- Asymmetric encryption algorithms
  - RSA
  - Diffie-Hellman key exchange
  - DSS, ECC
- Digital Signatures

# Asymmetric Cryptography Basics

2022-01-28

Stockholm University

# Asymmetric Cryptography -Terms

Stockholm University

- Plaintext (P)
- Encryption algorithm
- Ciphertext(C)
- Decryption algorithm
- **Public key and Private key**

- Also called Public Key Cryptography

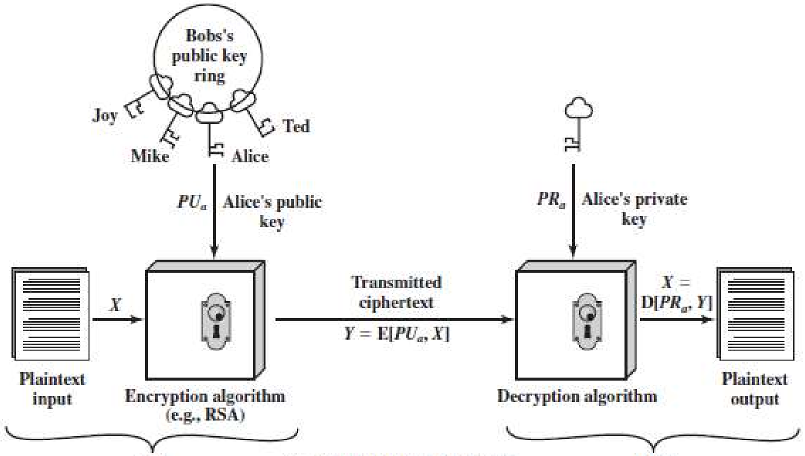
- Public key systems rely on a trapdoor one way function for their security.

# Asymmetric Cryptography -Model

- Sender encrypts a message with an encryption key (e.g., the receiver’s public key)
- Receiver decrypt the message (ciphertext) using a decryption key (e.g., the receiver’s private key)
- Encryption key/Public key
  - Can be known to everyone
- Decryption Key/Private key
  - Must be known and used only by its owner

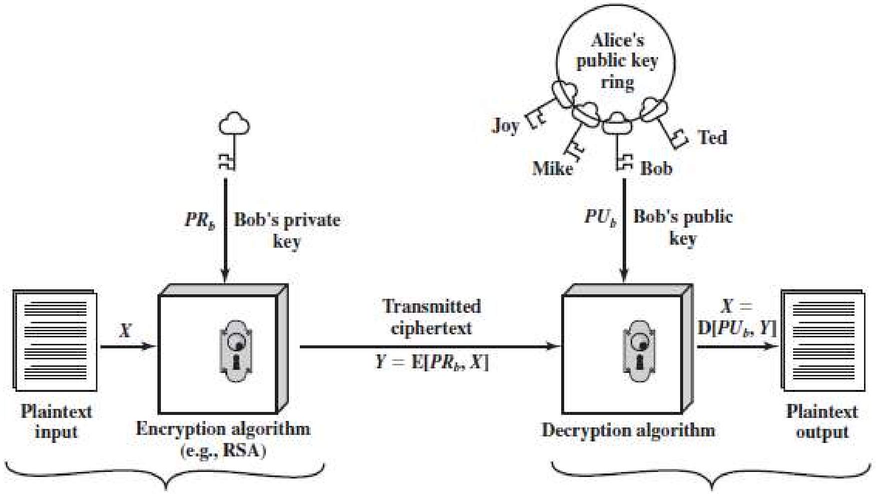
# Applications for Asymmetric Cryptography

- **Encryption/decryption:** The sender encrypts a message with the recipient's public key.



(a) Encryption with public key

- **Key exchange:** Two sides cooperate to establish a shared symmetric key (e.g., a session key). Several approaches are possible, involving the private key(s) of one or both parties.



(b) Encryption with private key

- **Digital signature:** 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.

# Asymmetric Cryptography Properties

- Confidentiality
  - Transmitting data over an insecure channel
  - Secure storage on insecure media
- Authentication protocols
- Digital signature
  - Provides integrity and non-repudiation
  - No non-repudiation with symmetric keys
- Strengths
  - Can provide integrity, authentication and non-repudiation
- Weaknesses
  - Slower than symmetric cryptography
  - Mathematically intensive tasks

# Trapdoor Functions

Stockholm University

- A special kind of one-way function is known as a ”trapdoor one way function”
  - Easy to compute in one direction Y = F (X) , difficult to inverse, unless parameter D(decryption key) is known.
  - If F is a trapdoor function, then there exists some secret information D, such that given F(X) and D, it is easy to compute X.
- Uses three algorithms (G, F, $F^{-1}$):
  - A trapdoor function is a function that goes from set X to set Y, and is defined by a set of three algorithms (G, F, $F^{-1}$). Trapdoor permutation maps X onto itself, the trapdoor function maps X to some arbitrary set Y (X->Y)
    - G = key generation algorithm (outputs in public key and private key)
    - F = function (without $F^{-1}$, it is a one-way function)
    - $F^{-1}$ = inverse of function F
- With knowledge of D (Y), decryption (finding $F^{-1}$) is easy; otherwise it is difficult.

# Asymmetric Cryptography Algorithms

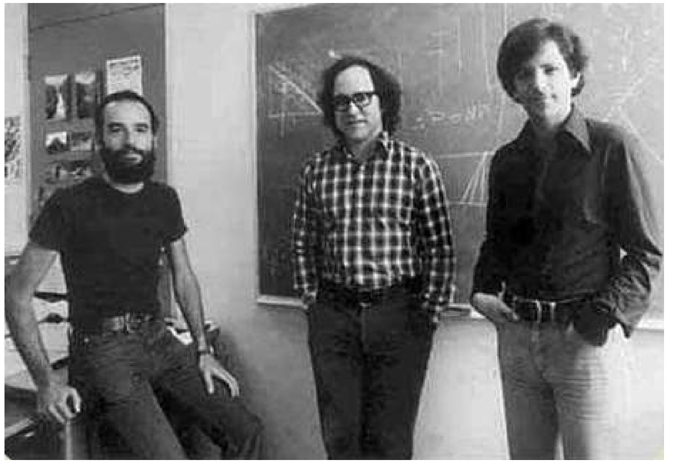
RSA
Diffie-Hellman
DSS and ECC

2022-01-28

Stockholm University

# RSA

- Rivest, Shamir, and Adleman (at MIT 1978)
- Public key cryptographic algorithm
  - Two keys (Public/Private)
  - Key length varies: 2048, 4096, 8192 bits
- Block cipher
  - Plaintext block must be smaller than key
  - Ciphertext block = key length



# Several Concepts…

Stockholm University

- Prime: a natural number greater than 1 that is not a product of two smaller natural numbers, e.g.,5. (vs. composite number)
- Coprime (relatively prime or mutually prime): two integers ***a*** and ***b***, if the only positive integer that evenly divides both of them is 1, we say ***a*** is coprime to ***b***. (i.e., their greatest common divisor (gcd) is 1)
- Euler totient function φ (n): counts the positive integers up to a given integer ***n*** that are relatively prime to ***n*** (it is the number of intergers ***k*** in the range of $1 \le k \le n$, for which the greatest common divisor gcd(***n***,***k***) is equal to 1)
  - If ***n*** is prime, then $\varphi(n) = n-1$
- mod: modulo operation, returns the remainder or signed remainder of a division, after a number is divided by another, e.g., given two positive number ***a*** and ***n***, if ***b***= ***a*** mod ***n***, then there exist an integer ***k***, such that $a = kn+b$

# RSA Algorithm – Generating the Keys

1. Select ”large” prime numbers p=11 q=3
2. Then n=p*q = 11*3=33 and
3. Calculate φ (n) = (p-1) (q-1) =10 x 2=20
4. Select e =3 (relatively prime to 20)
5. Determine d such that d*e mod 20 =1

   The correct value of d =7, because 7 x 3 =21 = 10 x 2=20

- Public key: (e, n) =(3, 33)
- Private key: (d, n)= (7, 33)

**Key Generation**

| | |
|---|---|
| Select $p, q$ | $p$ and $q$ both prime, $p \neq q$ |
| Calculate $n = p \times q$ | |
| Calculate $\phi(n) = (p-1)(q-1)$ | |
| Select integer $e$ | $\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$ |
| Calculate $d$ | $de \bmod \phi(n) = 1$ |
| Public key | $KU = \{e, n\}$ |
| Private key | $KR = \{d, n\}$ |

# RSA Algorithm – Encryption

- You know e and n
- Public key: (e, n)= (3, 33)
- Encrypt: C= $M^e$ (mod n)
- Suppose message M= 8

| Encryption | |
|---|---|
| Plaintext: | $M < n$ |
| Ciphertext: | $C = M^e \pmod{n}$ |

- Ciphertext C is computed as

  $C = M^e$ mod n

  $= 8^3$ mod 33

  = 512 mod 33 = 17

# RSA Algorithm – Decryption

- You know d and n
- Private key: (d, n) =(7, 33)
- Decrypt: $M = C^d \pmod{n}$

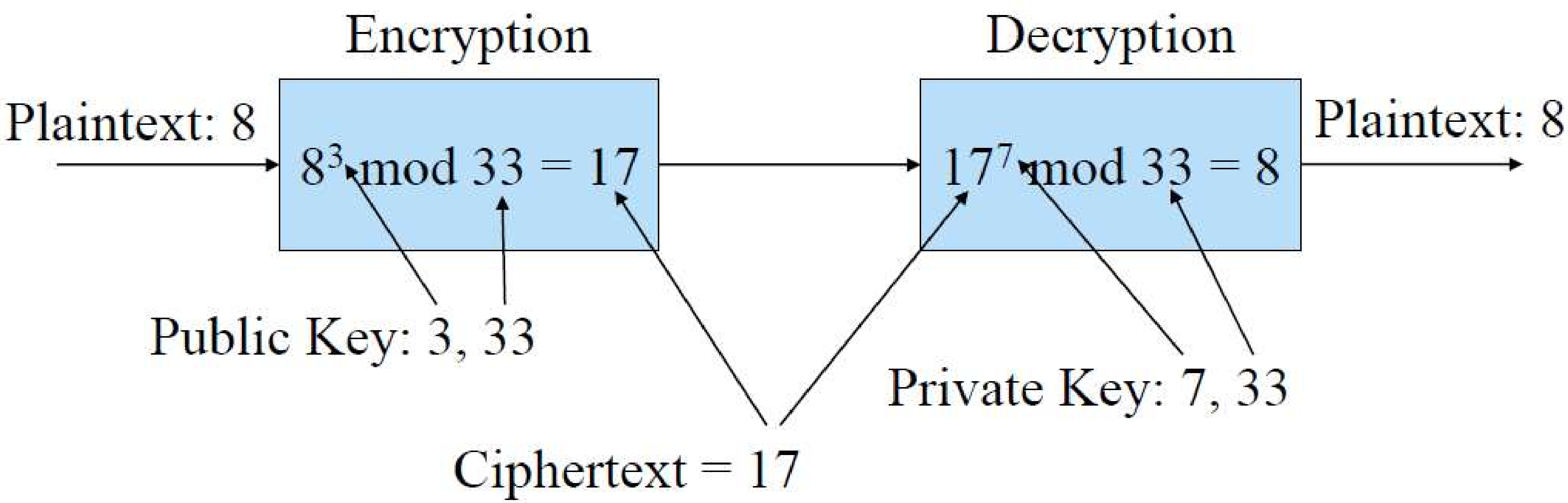
  $M = C^d \bmod n$

  $= 17^7 \bmod 33$

  $= 410338673 \bmod 33 = 8$

| Decryption | |
|---|---|
| Ciphertext: | $C$ |
| Plaintext: | $M = C^d \pmod{n}$ |

# RSA - Example



Encryption
Decryption
Plaintext: 8
$8^3$ mod 33 = 17
$17^7$ mod 33 = 8
Plaintext: 8
Public Key: 3, 33
Private Key: 7, 33
Ciphertext = 17

Stockholm University

# RSA is used in….

- SSL/TLS certificates
- IPSec
- E-mail systems
- File systems
- Etc.

# Attacks on RSA

- Most attacks on RSA are based on the assumption that Alice or Bob (or both) have been careless in their implementation of the RSA cryptosystem
- Factoring?
  - We have n
  - We want p & q
  - n = p*q
  - Precompute lots of prime factors (similar to rainbow tables) so that if given an n, we can look up what p & q are for any given n

## Summary of RSA

- RSA keys are generated using 2 large prime numbers.
- RSA key security is based on the difficult level of factoring prime numbers. Given p and q to calculate n = p*q is easy. But given n to find p and q is very difficult.
- RSA encryption operation requires calculation of "$C = M^e \bmod n$", which can be done by a loop.
- Most RSA tools are using public keys generated from large probable prime numbers, because generating large prime numbers is very expensive.

# The Diffie-Hellman-Merkle Key Exchange -1

- A key exchange mechanism which is used to establish a shared symmetric/secrete key
- Invented by Whitfield Diffie and Martin Edward Hellman (Stanford), based on a concept developed by Ralph Merkle (Merkle often not included in the name)

## Diffie-Hellman-Merkle Key Exchange -2

Stockholm University

- Uses some public key encryption techniques
  - Use case: Alice and Bob have never met, but want to exchange a shared secret/symmetric key over an insecure line, so that they can communicate securely
- Based on the discrete log problem:
  - Given numbers: a, p, and $a^k \pmod p$
  - Find exponent k

# The Diffie-Hellman Key Exchange Algorithm -1

Stockholm University

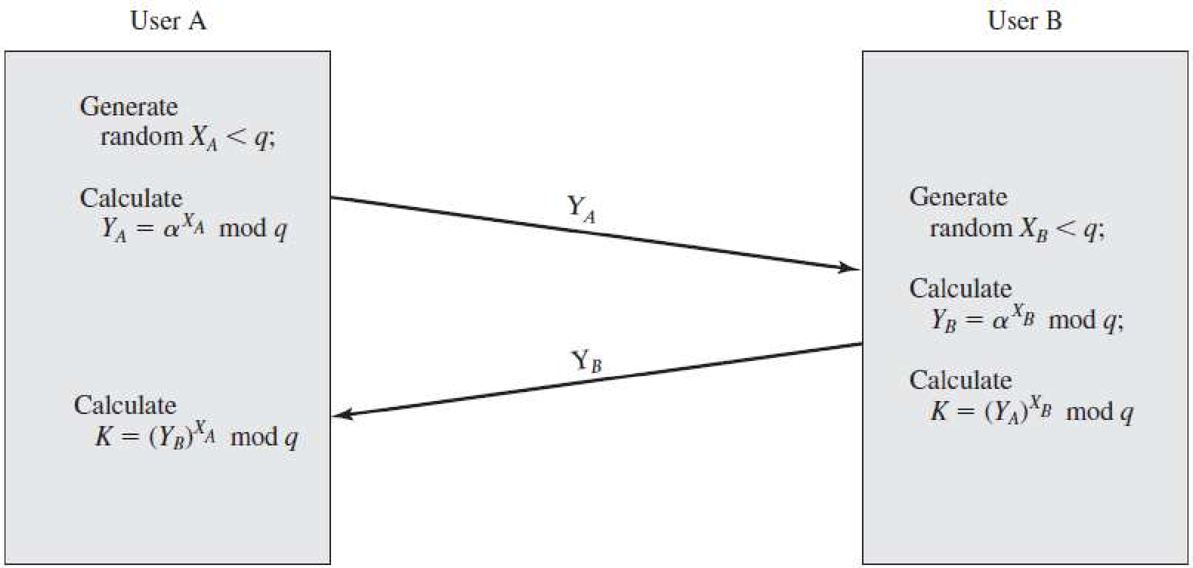
- Values q and a are predetermined constants already agreed upon by User A and User B.

| Global Public Elements | |
|---|---|
| $q$ | prime number |
| $\alpha$ | $\alpha < q$ and $\alpha$ a primitive root of $q$ |

| User A Key Generation | |
|---|---|
| Select private $X_A$ | $X_A < q$ |
| Calculate public $Y_A$ | $Y_A = \alpha^{X_A} \bmod q$ |

| User B Key Generation | |
|---|---|
| Select private $X_B$ | $X_B < q$ |
| Calculate public $Y_B$ | $Y_B = \alpha^{X_B} \bmod q$ |

# The Diffie-Hellman Key Exchange Algorithm -2




$$
\begin{aligned}
K &= \boxed{(Y_B)^{X_A} \bmod q} \\
&= (\alpha^{X_B} \bmod q)^{X_A} \bmod q \\
&= (\alpha^{X_B})^{X_A} \bmod q \\
&= \alpha^{X_B X_A} \bmod q \\
&= (\alpha^{X_A})^{X_B} \bmod q \\
&= (\alpha^{X_A} \bmod q)^{X_B} \bmod q \\
&= \boxed{(Y_A)^{X_B} \bmod q}
\end{aligned}
$$

Prerequisite:
- q is a prime number,
- a<q, a is a primitive root of q

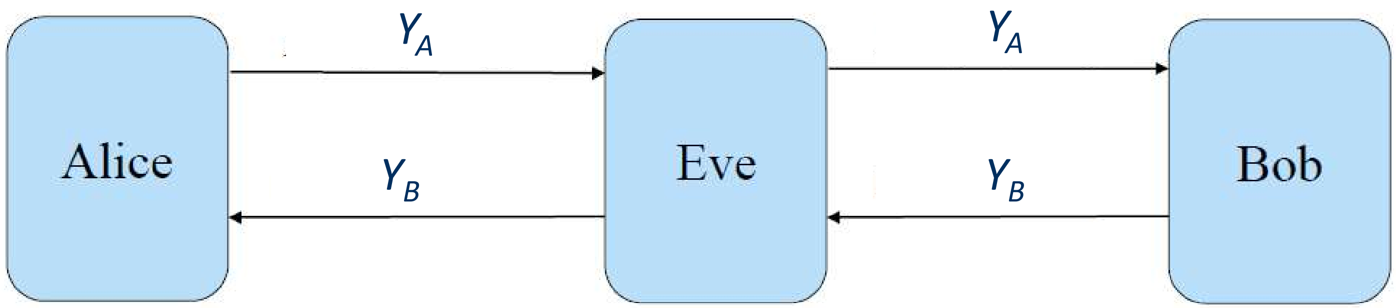
- A third person never sees $X_A$ or $X_B$ , so cannot calculate the $K$

# Attacking Diffie-Hellman-Merkle -1

Stockholm University

- Suppose Eve can see $q$, $a$, $Y_A$ and $Y_B$, but Alice's exponent $X_A$ and Bob's exponent $X_B$ are secret
- Can Eve compute $K$?
  - If Eve can find $X_A$ or $X_B$ , she gets $K$

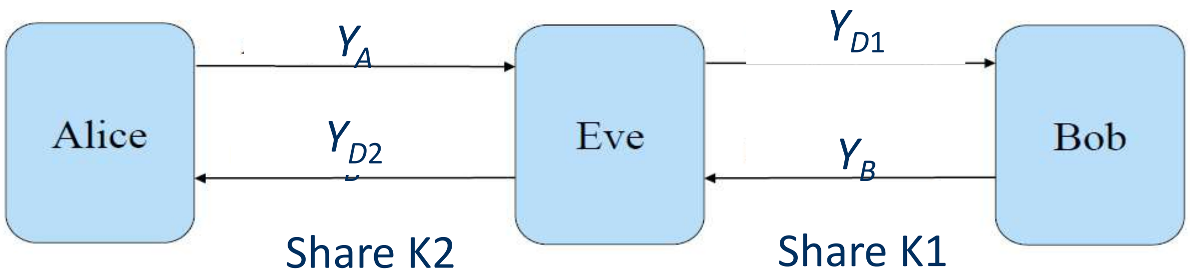
- This protocol is vulnerable to a man-in-the-middle-attack:



# Attacking Diffie-Hellman-Merkle -2

Stockholm University

1. Eve prepares for the attack by generating two random private keys $X_{D1}$ and $X_{D2}$, and then computing the corresponding public keys $Y_{D1}$ and $Y_{D2}$.
2. Alice sends $Y_A$ to Bob.
3. Eve intercepts $Y_A$ and transmits $Y_{D1}$ to Bob. Eve also calculates $K2 = (Y_A)^{XD2} \bmod q$.
4. Bob receives $Y_{D1}$ and calculates $K1 = (Y_{D1})^{XB} \bmod q$.
5. Bob transmits $Y_B$ to Alice.
6. Eve intercepts $Y_B$ and transmits $Y_{D2}$ to Alice. Eve calculates $K1 = (Y_B)^{XD1} \bmod q$
7. Alice receives $Y_{D2}$ and calculates $K2 = (Y_{D2})^{XA} \bmod q$



# Preventing a MITM Attack

Stockholm University

- Authenticate each other using either a previously shared secret/symmetric key or verified public keys
- Sign DH values with secret/symmetric or private key
- Ephemeral Diffie-Hellman-Merkle
  - A temporary (ephemeral) DH key is generated for every message, thus the same key is never used twice.
  - Enables Forward Secrecy (FS), which means that if the long-term private key of the server gets leaked, past communication is still secure

# Digital Signature Standard (DSS)

Stockholm University

- Published by NIST (FIPS 186-4)
- Originally proposed in 1991 and revised in 1993, 1996, 2000, 2009, and 2013.
- Uses an algorithm that is designed to provide only the digital signature function
  - Digital Signature Algorithm (DSA)
  - Unlike RSA, it cannot be used for encryption or key exchange

# Elliptic Curve Cryptography (ECC)

Stockholm University

- A new kind of mathematical problem, elliptic curves, can be used as the basis for a one-way trapdoor function instead of the prime factorization problem used in RSA.
- Does not require as many bits
  - a 224-bit ECC key is approximately equivalent to a 2048-bit RSA key
  - a 512-bit ECC key is approximately equivalent to a 256-bit AES key (or a 15360-bit RSA key)
- Key generation is faster than RSA
- Most cryptographic operations are faster than RSA
- RSA is still very common and entrenched due to historical reasons, but ECC is slowly gaining in popularity.

# Digital Signatures

2022-01-28

# Digital Signatures



- Use protocols to mimic real signatures
  - It must be unforgeable
  - It must be authentic
  - Is not alterable, and not reusable
- Two functions:
  - Used to detect unauthorized modifications to data
  - Provide non-repudiation
- A signature is generated by using a private key: the private key is known only to the user.
- The signature is verified: makes use of the public key which corresponds to the private key
- Used in e-mails, electronic funds transfer and any application that needs to assure the integrity and originality of data

# Example - without Signature

- Alice orders products from Bob.
  - Alice computes a MAC using a symmetric key. So Alice and Bob share a key.
  - The price drops, and Alice claims that she didn’t place the order.
- Can Bob prove that Alice placed the order?
  - No, since Bob also knows symmetric key, he could have forged the message.

⇨ Bob knows that Allice placed the order but he can’t prove it.

## Example - with Signature

- Alice orders products from Bob.
  - Alice signs her order with her private key.
  - The price drops and Alice regrets her order, so she claims that she didn't place it.
- Since Alice signed the order with her private key, Bob can prove that Alice in fact placed the order.

⇨ Non-repudiation by using signature!

# Thank you.

2022-01-28

Stockholm University

# References

- Stallings. Network Security Essentials
  - Chapter 3
- OpenSSL Wikipage on Elliptic Curve Cryptography
  - https://wiki.openssl.org/index.php/Elliptic_Curve_Cryptography