SWINBURNE UNIVERSITY OF TECHNOLOGY

Theory of Blockchain

Session 3:

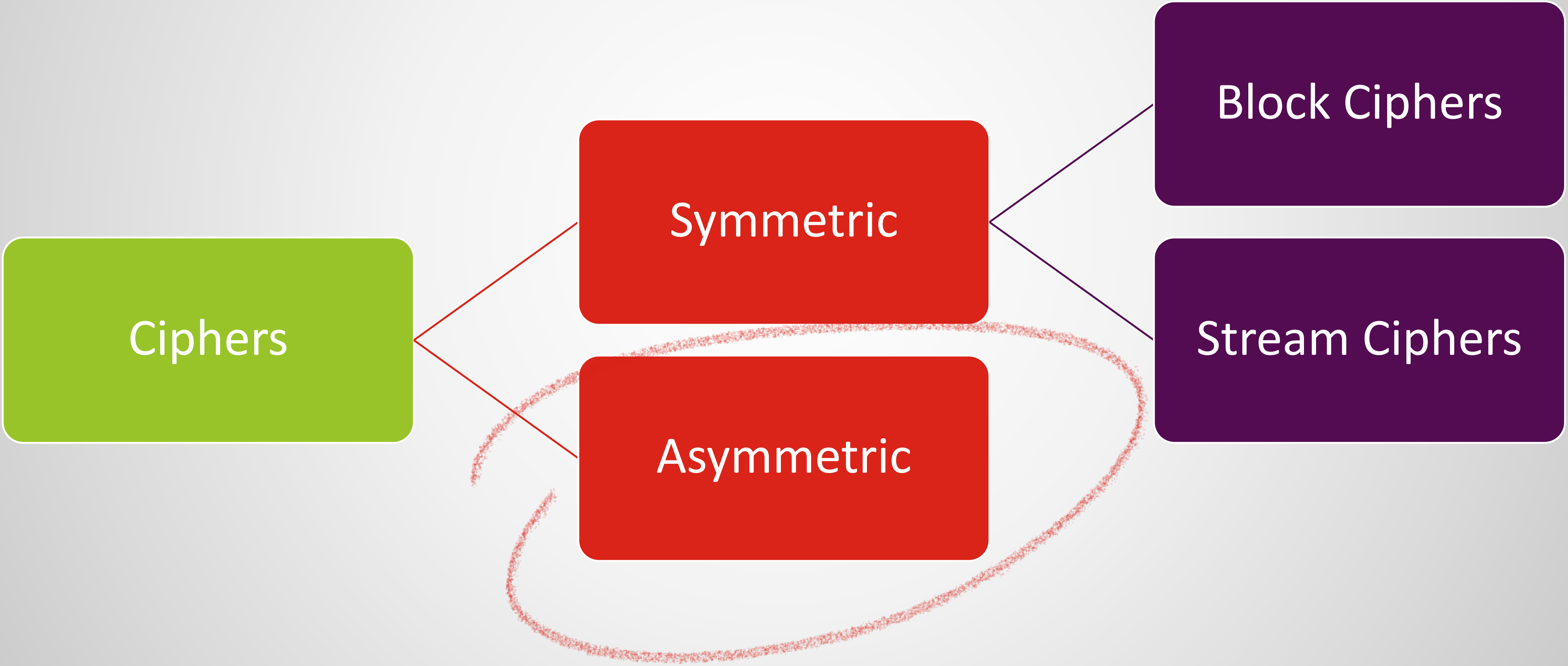
Asymmetric Cryptography - Part 1

Module 1 – Hard Mathematical Problems and RSA

Sayad Haghighi

Classification of Ciphers

Asymmetric Cryptography (Public Key Cryptography)



General Form of Asymmetric Cryptography

Asymmetric Encryption (Public Key Encryption)

$$E = D^{-1}\ ,\ D = E^{-1}$$

Public Key
Everybody can send encrypted messages to the owner of the public key

Private Key
Only the owner of the private key (corresponding to the public key) can decrypt the message

Samples: RSA ،Elgamal …

Security of Asymmetric Algorithms

m → **E** → c ⇢ c → **D** → m

K_{pu} → **E** K_{pr} → **D**

$$E = D^{-1}\ ,\ D = E^{-1}$$

K_{pu} and K_{pr} are mathematically related.

But:

K_{pu}

Bob

E

Computationally Easy

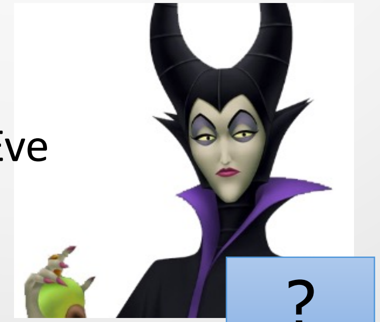
Alice

K_{pr}

D

Computationally Easy

Eve



?

D

Computationally Hard

Public Key Systems

- Merkle-Hellman knapsack
- Diffie-Hellman key exchange
- RSA
- Rabin cipher
- NTRU cipher
- ElGamal
- …

Public Key Crypto

- Some public key systems provide it all: encryption, digital signatures, etc.
  - For example: RSA
- Some are only for key exchange
  - For example: Diffie-Hellman
- Some are used for signatures more
  - For example: ElGamal
- **All of these are public-key systems !**

Modular Arithmetic

"**mod**" gives the residue of a division operation:

example :

8 mod 4 = 0

6 mod 4 = 2

1 mod 4 = 1

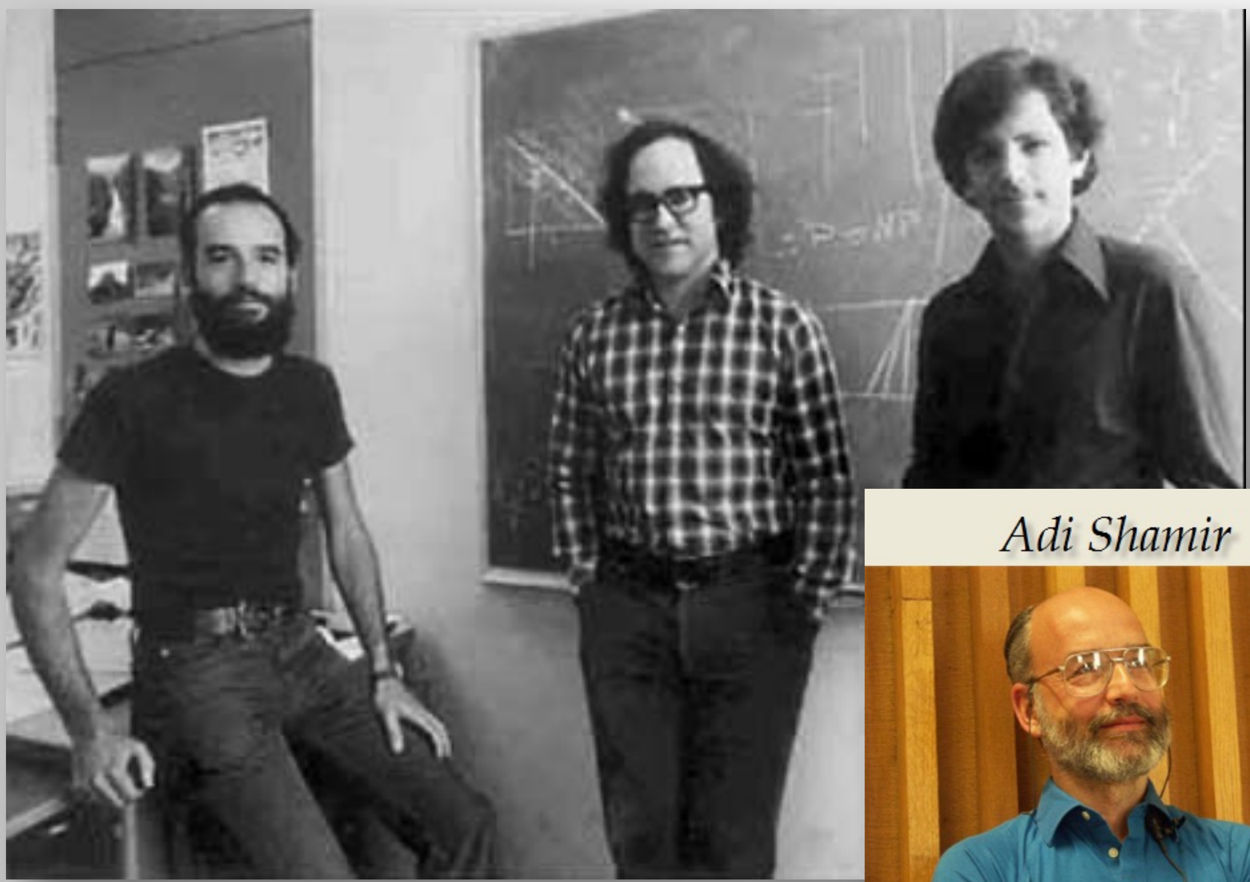
13 mod 4 =1

"**a**" and "**b**" are called **congruent modulo n** if they have the same residue in division over "**n**".

$$a \overset{n}{\equiv} b \quad \text{or} \quad a = b \mod n$$

RSA Asymmetric Encryption Algorithm

- Introduced by Rivest, Shamir & Adleman at MIT in 1978.



RSA Asymmetric Encryption Algorithm

How to make an RSA cryptosystem? :

1- choose two large prime numbers p & q.

2- calculate n=p*q

3- calculate Euler's Phi function as $\varphi(n) = (p-1)(q-1)$

$\varphi(n)$ is an arithmetic function that counts the positive integers less than or equal to *n* that are relatively prime to n (i.e. their GCD with *n* is 1).

RSA is called a block cipher in Stallings' book.

RSA (cont’d)

4- choose an encryption key (“**e**”) so that it’s relatively prime to $\varphi(n)$.

5- calculate its inverse congruent modulo $\varphi(n)$ and call it “**d**”.

- This is done by the Extended Euclidean Algorithm which we will see later

$$e.d \overset{\varphi(n)}{\equiv} 1$$

Public Parameters :PU={e, n}

Private Parameters : PR={d}

Encryption :

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

Decryption :

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

The only condition is that $M<n$

Why does RSA work?

Encryption: $C \overset{n}{\equiv} M^{e}$

Decryption: $C^{d} \overset{n}{\equiv} (M^{e})^{d} \overset{n}{\equiv} M^{ed}$

Euler's Theorem (Fermat's Little Theorem): $a^{\varphi(n)} \overset{n}{\equiv} 1$ if $(a,n)=1$

Assignment #1 $\longrightarrow$ $M^{ed} \overset{n}{\equiv} M^{ed \bmod \varphi(n)} \overset{n}{\equiv} M^{1}$

Even if gcd(M,n)$\neq 1$, it's possible to prove that $M^{ed} \equiv M \bmod n$

RSA (cont'd)

Factoring is known to be a hard mathematical problem:

187=11*17

Mathematically hard (no polynomial time algorithm for classic computers)

Based on this, it has been proven that knowing ***n* & *e*,** *and* without knowing **p & q,** calculation of **d** is mathematically hard and is equivalent to factoring ***n*** (which is a big number!).

While, if one has **p & q**, he can easily compute $\varphi(n)$ and find the inverse of **e** (i.e. **d**) .

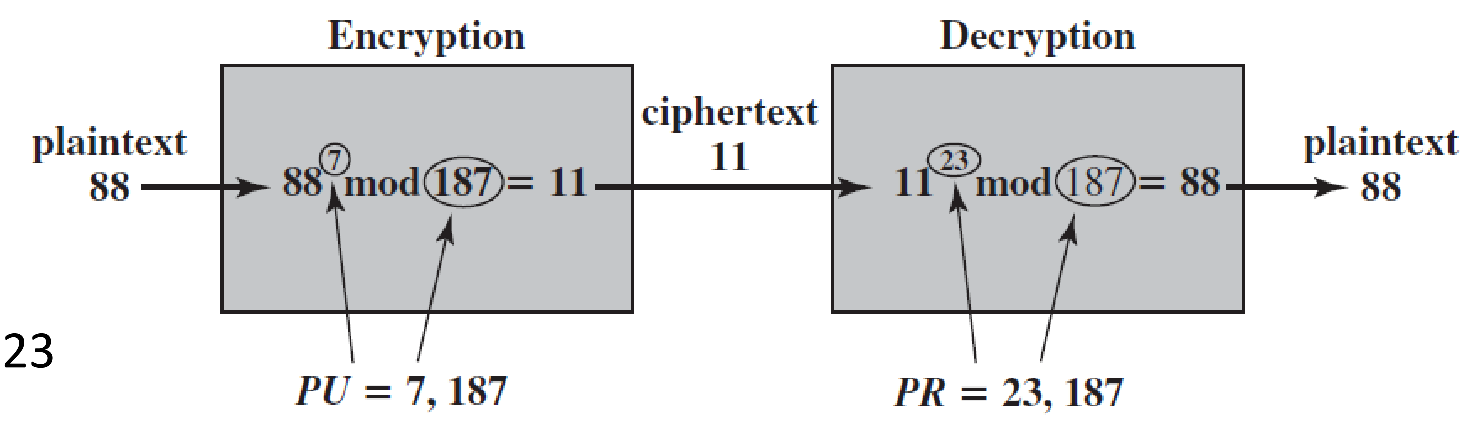
Example:

p=11, q=17

n=11*17=187

$\varphi(n)$=(p-1)(q-1)=160

e=7 (notice that gcd(e, $\varphi(n)$)=1 => d=23



Encryption Example

$$88^7 \bmod 187 = [(88^4 \bmod 187) \times (88^2 \bmod 187) \times (88^1 \bmod 187)] \bmod 187$$
$$88^1 \bmod 187 = 88$$
$$88^2 \bmod 187 = 7744 \bmod 187 = 77$$
$$88^4 \bmod 187 = 132$$
$$88^7 \bmod 187 = (88 \times 77 \times 132) \bmod 187 = 894{,}432 \bmod 187 = 11$$

(Stallings)

Decryption Example

$$
\begin{aligned}
11^{23} \bmod 187 &= [(11^1 \bmod 187) \times (11^2 \bmod 187) \times (11^4 \bmod 187) \times \\
&\qquad (11^8 \bmod 187) \times (11^8 \bmod 187)] \bmod 187 \\
11^1 \bmod 187 &= 11 \\
11^2 \bmod 187 &= 121 \\
11^4 \bmod 187 &= 14{,}641 \bmod 187 = 55 \\
11^8 \bmod 187 &= 214{,}358{,}881 \bmod 187 = 33 \\
11^{23} \bmod 187 &= (11 \times 121 \times 55 \times 33 \times 33) \bmod 187 \\
&= 79{,}720{,}245 \bmod 187 = 88
\end{aligned}
$$

Extended Euclidean Algorithm

How do we find the inverse of a number modulo $\varphi(n)$:

$$e \times d \ \ mod\ \phi(n) = 1$$
$$7 \times d \ \ mod \ \ 40 \ \ = \ 1$$

Step 1: Euclidean Algorithm

$7d = 40k + 1$

$7d + 40 \times (-k) = 1$

$40x + 7d = 1$

$40 = 5(7) + 5$

$7 = 1(5) + 2$

$5 = 2(2) + 1$

Step 2: Back Substitution

$1 = 5 - 2(2)$

$1 = 5 - 2(7 - 1(5))$

$1 = 3(5) - 2(7)$

$1 = 3(40 - 5(7)) - 2(7)$

$1 = 3(40) - 17(7)$

$d = -17\ mod\ 40$

$= -17 + 40\ mod\ 40$

$= 23\ mod\ 40$

$p = 11$

$q = 5$

$n = 55$

$\phi(n) = 40$

$e = 7$

$d = ?$

This is a polynomial time algorithm.

What Comes Next …

- We learned about the difficulty of factoring problem.

- We learned how RSA encrypts and decrypts mathematically and why it is hard to break it.

- In the next video, we introduce another hard mathematical problem and explain an asymmetric method for symmetric key agreement.

See you in the next video …