Blockchain Technologies

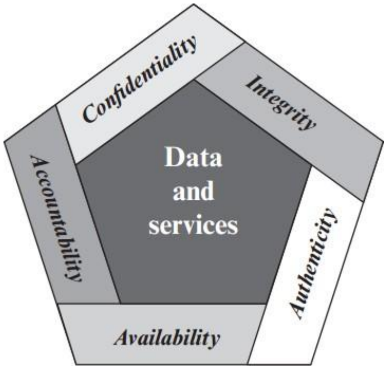
Cryptosystems

CIA TRIAD

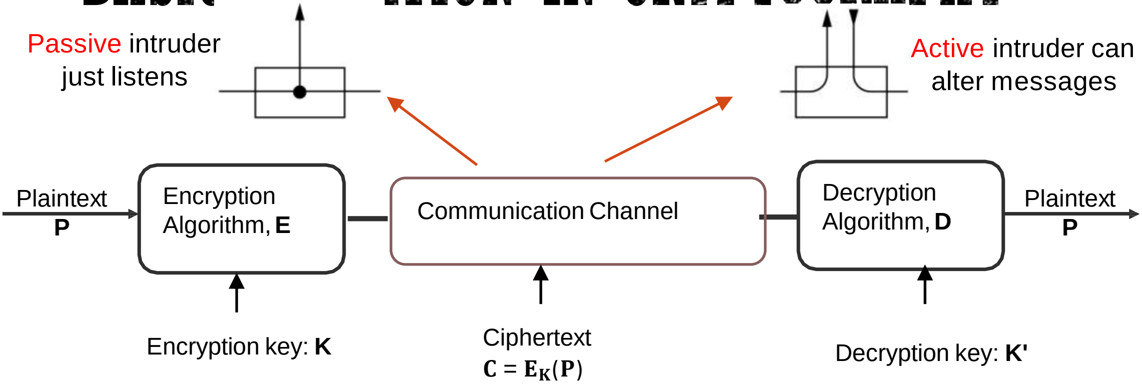
- **Confidentiality**: Preserving authorized restrictions on information access and disclosure.
  - A loss of confidentiality is the unauthorized disclosure of information.
- **Integrity**: Guarding against improper information modification or destruction.
  - A loss of integrity is the unauthorized modification or destruction of information.
- **Availability**: Ensuring timely and reliable access to information.
  - A loss of availability is the disruption of access to or use of information or an information system.

OTHER SECURITY REQUIREMENTS

- **Authenticity:** The property of being genuine and being able to be verified and trusted.
  - This means verifying that users are who they say they are and that each input arriving at the system came from a trusted source.
- **Accountability:** The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity.
  - We must be able to trace a security breach to a responsible party.
  - Systems must keep records of their activities to permit later forensic analysis to trace security breaches or to aid in transaction disputes.

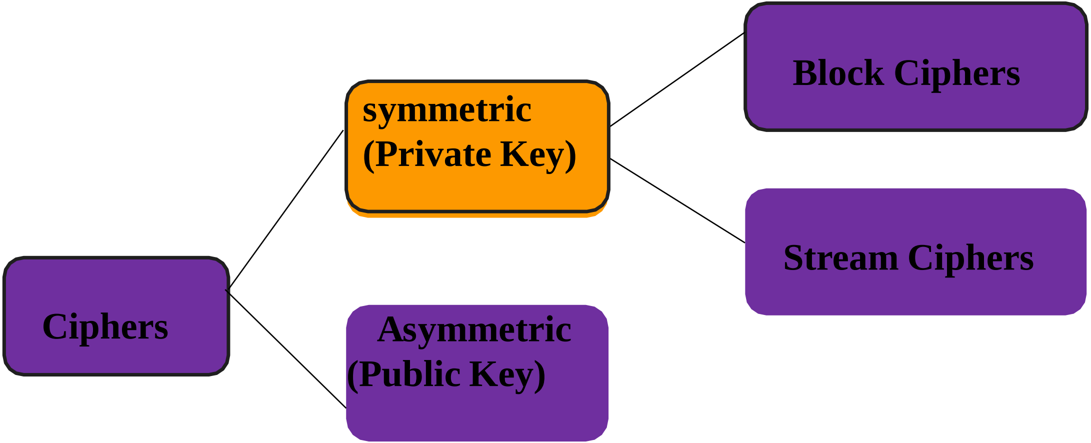


BASIC SITUATION IN CRYPTOGRAPHY



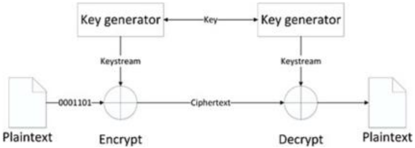
- **Passive attack**: the attacker only monitors the traffic attacking the confidentiality of the data
- **Active attack**: the adversary attempts to alter the transmission attacking data integrity, confidentiality, and authentication, system resources or affect their operations

CLASSIFICATION OF CRYPTOSYSTEMS

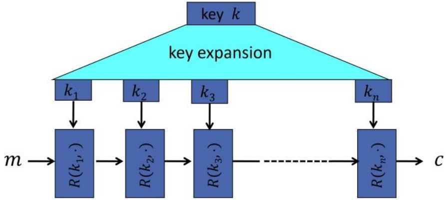


SYMMETRIC CIPHERS

- **Stream cipher** is one that encrypts a digital data stream one bit (or byte) at a time
  - Example: autokey Vigenère system

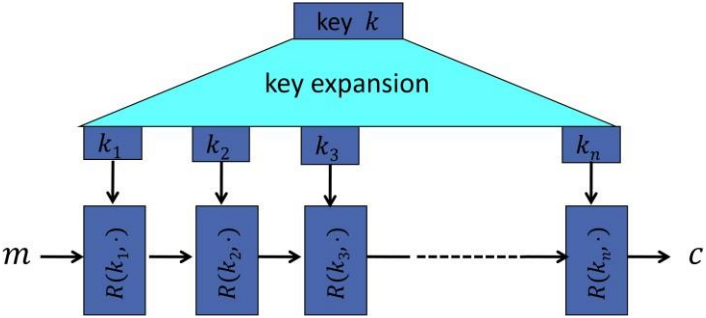


- **Block cipher** is one in which the plaintext is divided in blocks and one block is encrypted at one time producing a ciphertext of equal length
  - 64 bits or 128 bits are typical block lengths
  - Many modern ciphers are block ciphers



BLOCK CIPHERS STRUCTURE

- Block ciphers are built by iteration:



- $R(k, m)$ is called a round function.

ADVANCED ENCRYPTION STANDARD

- AES competition
  - Started in January 1997 by NIST
  - 4-year cooperation between
  - U.S. Government
  - Private Industry
  - Academia

- Why?
  - Replace 3DES
  - Provide a publicly disclosed encryption algorithm, available royalty-free, worldwide

THE FINALISTS

- **MARS**
  - IBM
- **RC6**
  - RSA Laboratories
- **Rijndael**
  - Joan Daemen (Proton World International) and Vincent Rijmen (Katholieke Universiteit Leuven)
- **Serpent**
  - Ross Anderson (University of Cambridge), Eli Biham (Technion), and Lars Knudsen (University of California San Diego)
- **Twofish**
  - Bruce Schneier, John Kelsey, and Niels Ferguson (Counterpane, Inc.), Doug Whiting (Hi/fn, Inc.), David Wagner (University of California Berkeley), and Chris Hall (Princeton University)

VERSIONS OF AES

- Rijndael supports block sizes and key sizes of 128, 160, 192, 224 and 256 bits.

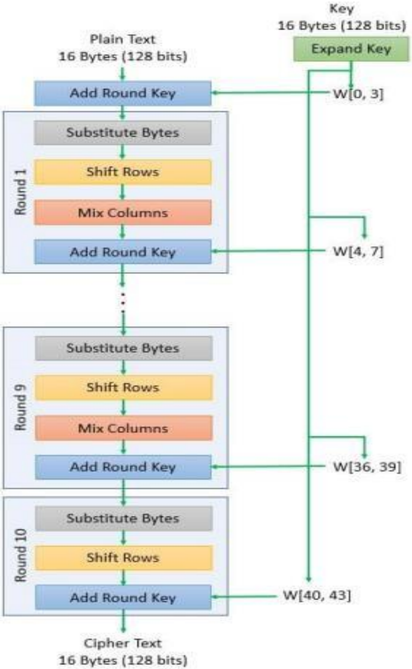
- Only 128-bit block size, and 128, 192, and 256 key sizes are specified in the AES.

Version	Key Size	Number of rounds
AES-128	128 bits	10
AES-192	192 bits	12
AES-256	256 bits	14

AES Key Size

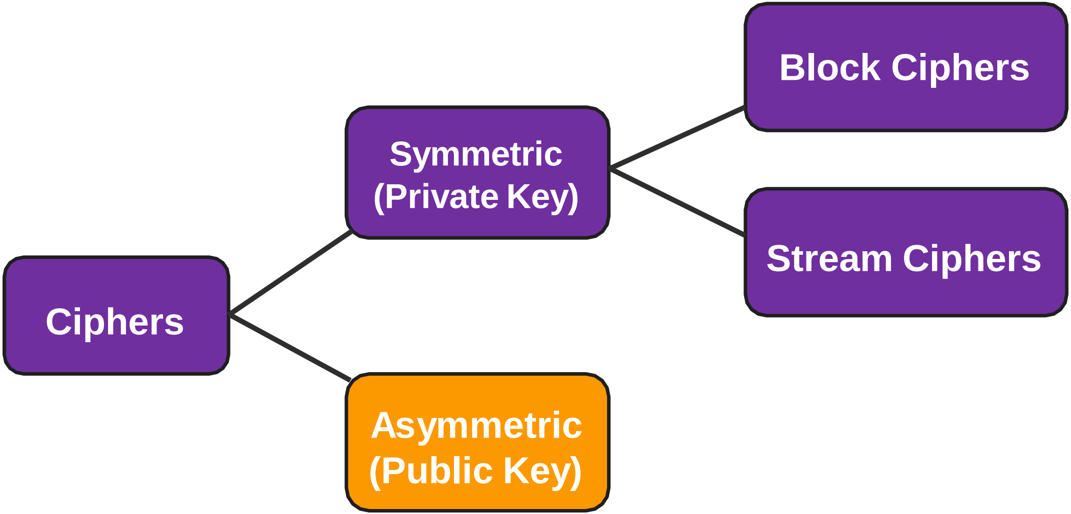
- Uses really big numbers
  - 1 in 2^{61} odds of winning the lotto and being hit by lightning on the same day
  - 2^{92} atoms in the average human body
  - 2^{128} possible keys in AES-128
  - 2^{170} atoms in the planet
  - 2^{190} atoms in the sun
  - 2^{192} possible keys in AES-192
  - 2^{233} atoms in the galaxy
  - 2^{256} possible keys in AES-256

AES



Key
16 Bytes (128 bits)
Expand Key
Plain Text
16 Bytes (128 bits)
Add Round Key
W[0, 3]
Round 1
Substitute Bytes
Shift Rows
Mix Columns
Add Round Key
W[4, 7]
Round 9
Substitute Bytes
Shift Rows
Mix Columns
Add Round Key
W[36, 39]
Round 10
Substitute Bytes
Shift Rows
Add Round Key
W[40, 43]
Cipher Text
16 Bytes (128 bits)

CLASSIFICATION OF CRYPTOSYSTEMS



PROBLEMS WITH SYMMETRIC CIPHERS

- **Key management**: changing the secret key or establishing one is nontrivial.
  - Change the keys two users share (should be done reasonably often)
  - Establish a secret key with somebody you do not know and cannot meet in person: (e.g., visiting secure websites such as e-shops)
  - This could be done via a trusted Key Distribution Center (KDC)
  - Can (or should) we really trust the KDC?
  - "What good would it do after all to develop impenetrable cryptosystems, if their users were forced to share their keys with a KDC that could be compromised by either burglary or subpoena?" – Diffie, 1988

- **Digital signatures**: a mathematical scheme for demonstrating the authenticity of digital messages or documents

A BREAKTHROUGH IDEA

- Rather than having a secret key that the two users must share, each users has **two keys**
- **One key is secret** and he is the only one who knows it
- **The other key is public** and anyone who wishes to send him a message uses that key to encrypt the message
- Diffie and Hellman's groundbreaking 1976 paper, "New Directions in Cryptography," introduced the ideas of public-key cryptography
- NSA claims to have known it since mid-1960s!
- Communications-Electronic Security Group (British counterpart of NSA) documented the idea in a classified report in 1970.

Martin Hellman & Whitfield Diffie

INVENTION OF PUBLIC KEY CRYPTOGRAPHY

- Diffie and Hellman's invention of public-key cryptography and digital signatures revolutionized computer security

They received the 2015 ACM A.M. **Turing Award** for critical contributions to modern cryptography

THE IDEA OF PUBLIC-KEY CRYPTOGRAPHY

- Although the concept was proposed by Diffie and Hellman, no practical way to design such a system was suggested.
- Each user has two keys: one encryption key that he makes public and one decryption key that he keeps secret.
  - Clearly, it should be computationally infeasible to determine the decryption key given only the encryption key and the cryptographic algorithm.
- Some algorithms (such as RSA) satisfy also the following useful characteristic:
  - Either one of the two keys can be used for encryption – the other one should then be used to decrypt the message.

STEPS IN PUBLIC-KEY SCHEME

- Each user generates a pair of keys to be used for encryption and decryption.
- Each user places one of the two keys in a public register and the other key is kept private.
- If B wants to send a confidential message to A, B encrypts the message using A's public key.
- When A receives the message, she decrypts it using her private key
  - Nobody else can decrypt the message because that can only be done using A's private key.
  - Deducing a private key should be infeasible.
- If a user wishes to change his keys – generate another pair of keys and publish the public one: no interaction with other users is needed.

SOME NOTATION

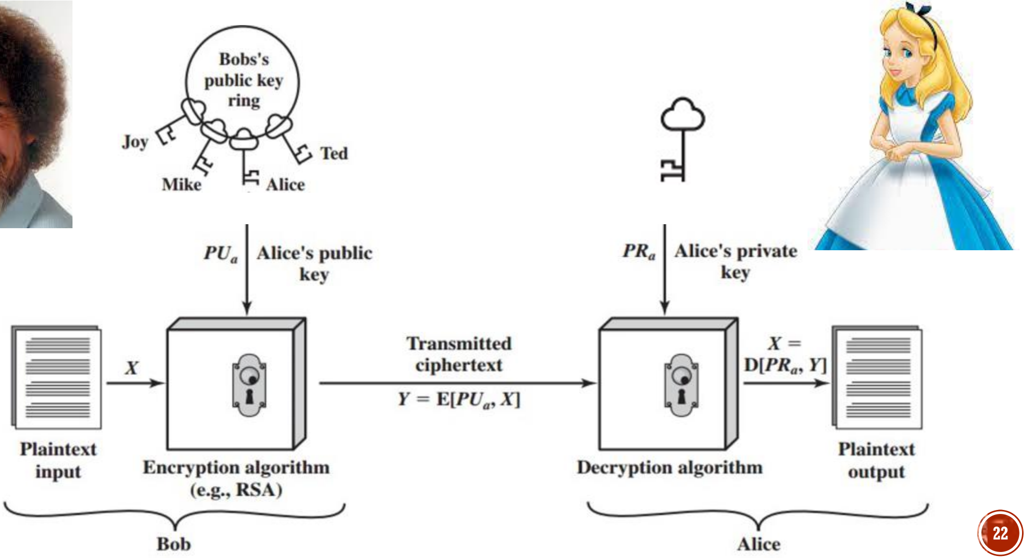
- The public key of user A will be denoted PU_A
- The private key of user A will be denoted PR_A
- Encryption method will be a function E
- Decryption method will be a function D
- If B wishes to send a plain message X to A, then he sends the ciphertext:

$$Y = E(PU_A, X)$$

- The intended receiver A will decrypt the message:

$$D(PR_A, Y) = X$$

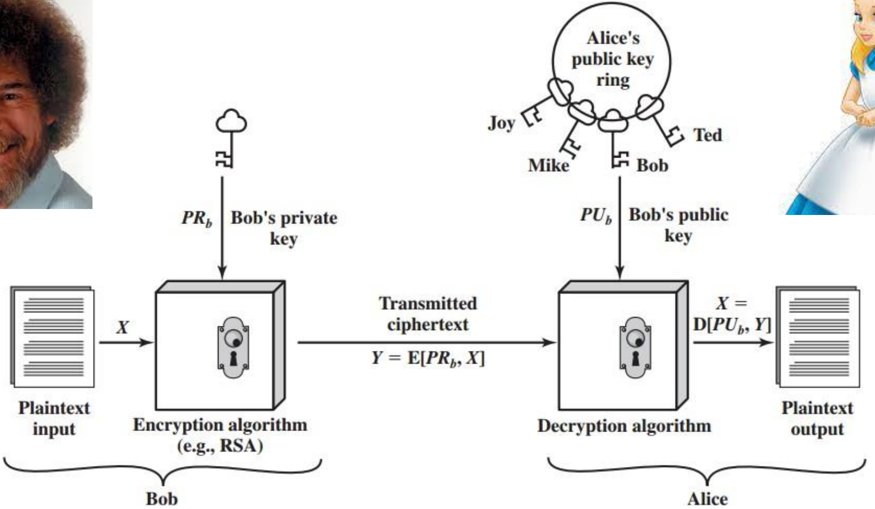
PUBLIC KEY SCHEME FOR CONFIDENTIALITY



ATTACK ON THE PUBLIC-KEY SCHEME

- **Immediate attack on this scheme**:
  - **An attacker may impersonate user B**: he sends a message $E(PU_A, X)$ and claims in the message to be B
  - This was guaranteed in symmetric cryptosystems through knowing the key (only A and B are supposed to know the symmetric key)
- The **authenticity** of user B can be established as follows:
  - B will encrypt the message using his private key: $Y = E(PR_B, X)$
  - This shows the authenticity of the sender because he is the only one who knows the private key
  - The entire encrypted message serves as a digital signature
  - *Note*: **this may not be the best possible solution**: ideally, digital signatures should be rather small so that one can preserve many of them over a long period of time

PUBLIC KEY SCHEME FOR AUTHENTICATION

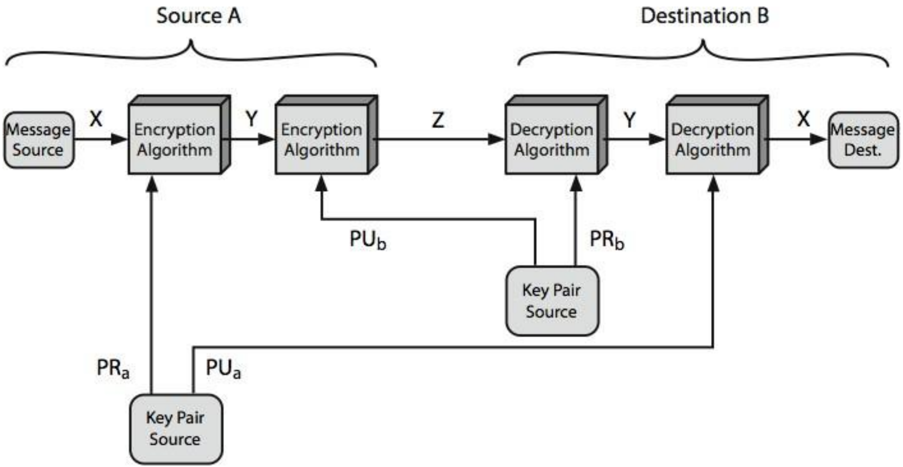


CONFIDENTIALITY AND AUTHENTICATION

- **Still a drawback**: the scheme on the previous slide authenticates but does not ensure confidentiality of the message
  - Anybody can decrypt the message using B's public key

- One can provide both authentication and confidentiality using the public-key scheme twice:
  - B encrypts X with his private key: $Y = E(PR_B, X)$
  - B encrypts Y with A's public key: $Z = E(PU_A, Y)$
  - A will decrypt Z (and she is the only one capable of doing it):
    $$Y = D(PR_A, Z)$$
  - A can now get the plaintext and ensure that it comes from B (B is the only one who knows his private key): decrypt Y using B's public key:
    $$X = E(PU_B, Y)$$

CONFIDENTIALITY AND AUTHENTICATION



APPLICATIONS FOR PUBLIC-KEY CRYPTOSYSTEMS

1. **Encryption/decryption**: sender encrypts the message with the receiver’s public key.

2. **Digital signature**: sender “signs” the message (or a representative part of the message) using his private key.

3. **Key exchange**: two sides cooperate to exchange a secret key for later use in a secret-key (symmetric) cryptosystem.

REQUIREMENTS FOR PUBLIC-KEY CRYPTOSYSTEMS

- Generating a key pair (public key, private key) is computationally easy.
- Encrypting a message using a known key (his own private or somebody else's public) is computationally easy.
- Decrypting a message using a known key (his own private or somebody else's public) is computationally easy.
- Knowing the public key, it is computationally infeasible for an opponent to deduce the private key.
- Knowing the public key and a ciphertext, it is computationally infeasible for an opponent to deduce the private key.

DESIGNING A PUBLIC-KEY CRYPTOSYSTEM

- *Computationally easy* usually means polynomial-time algorithm
- *Computationally infeasible* more difficult to define:
  - Usually means super-polynomial-time algorithms, e.g., exponential-time algorithms
  - Classical complexity analysis (worst-case complexity or average-case complexity) are worthless in cryptography: we must make sure a problem is difficult for almost all inputs and not just in the worse or in the average case

- Public-key cryptosystems usually rely on difficult math functions rather than S-P networks as classical cryptosystems:
  - **Aim**: find a trap-door one-way function for encryption – decryption will be the inverse

DIFFICULT MATH FUNCTIONS

- **One-way function**: easy to calculate in one direction, infeasible to calculate in the other direction (i.e., the inverse is infeasible to compute)

- One-way function has
  - Computing $Y = f(X)$ is easy
  - Computing $X = f^{-1}(Y)$ is infeasible

DIFFICULT MATH FUNCTIONS

- **Trap-door function**: difficult function that becomes easy if some extra information is known
  - A trap-door one-way function has
    - Computing $Y = f_k(X)$ is easy, if k and X are known
    - Computing $X = {f_k}^{-1}(Y)$ is easy, if k and Y are known
    - Computing $X = {f_k}^{-1}(Y)$ is infeasible, if Y is known but k is not known

ONE-WAY AND TRAP-DOOR FUNCTIONS

- One-way function has
  - Computing $Y = f(X)$ is easy
  - Computing $X = f^{-1}(Y)$ is infeasible
- A trap-door one-way function has
  - Computing $Y = f_k(X)$ is easy, if k and X are known
  - Computing $X = f_k^{-1}(Y)$ is easy, if k and Y are known
  - Computing $X = f_k^{-1}(Y)$ is infeasible, if Y is known but k is not known
- A practical public-key scheme depends on a suitable trap-door one-way function.

RSA

- One of the first proposals on implementing the concept of public-key cryptography was that of Rivest, Shamir, Adleman – 1977: RSA
- The RSA scheme works like a block cipher in which the plaintext and the ciphertext are integers between 0 and n – 1 for some fixed n
  - Typical size for n is 1024 bits (or 309 decimal digits)
  - To be secure with today's technology size should between 1024 and 2048 bits
- Idea of RSA: it is a difficult math problem to factorize (large) integers
  - **Choose p and q odd primes, and compute $n = pq$**
  - **Choose integers d, e such that $M^{ed} = M \bmod n$, for all $M < n$**
  - **Plaintext**: block of k bits, where $2^{k\text{-}1} \leq n < 2^k$ – can be considered a number M with $M < n$
  - **Encryption**: $C = M^e \, mod \, n$
  - **Decryption**: $C^d \, mod \, n = M^{ed} \, mod \, n = M$
  - **Public key**: $PU = \{e, n\}$ and **Private key**: $PR = \{d, n\}$

NUMBER THEORY

- Questions: How do we find d, e? How do we find large primes?
  - Answer: Number Theory!
- **Fermat's little theorem**: if p is prime and a is positive integer not divisible by p, then $a^{p-1} \equiv 1 \mod p$
- **Corollary**: For any positive integer a and prime p, $a^p \equiv a \mod p$
- Fermat's little theorem provides a necessary condition for an integer p to be prime – the condition is not sufficient
  - We will turn this theorem into a (probabilistic) test for primality
- Fermat's theorem, as useful as it will turn out to be, it does not provide us with integers d, e we are looking for
  - Euler's theorem (a refinement of Fermat's) does.

EULER'S TOTIENT FUNCTION

- Euler's function associates to any positive integer n a number $\phi(n)$: the number of positive integers smaller than n and relatively prime to n
  - Obviously for a prime number p: $\phi(p) = p - 1$
- It is easy to show that if $n = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k}$ be the prime factorization of n, then: $\phi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right) \dots \left(1 - \frac{1}{p_k}\right)$
- For prime numbers p and q : $\phi(pq) = (p-1)(q-1)$
- **Euler's theorem**: for any relatively prime integers a, n we have:

$$a^{\phi(n)} \equiv 1 \; mod \; n$$

- **Corollary**: For any integers a, k, n we have $a^{k\phi(n)+1} \equiv a \; mod \; n$

BACK TO RSA

- Let p, q be two odd primes and $n = pq$. Then for any integers k, m with $0 < m < n$, we have $m^{k(p-1)(q-1)+1} \equiv m \bmod n$
- Euler's theorem provides us the numbers d, e such that

$$M^{ed} = M \bmod n$$

- We have to choose d, e such that $ed = k\phi(n) + 1$ for some k
- Equivalently, $d \equiv e^{-1} \bmod \varphi(n)$

RSA EXAMPLE

- **Key generation**
- Select primes $p = 17, q = 11$
- Compute $n = pq = 187$
- Compute $\phi(n) = (p-1)(q-1) = 160$
- Select $e = 7$: $\gcd(7,160) = 1$
- Compute $d = e^{-1} \bmod \phi(n)$ using the *extended Euclid's algorithm*:

$$160 = 22 \times 7 + 6 \quad \rightarrow \quad 7 = 1 \times 6 + 1$$
$$1 = 7 - 1 \times 6 = 7 - 1 \times (160 - 22 \times 7) = 23 \times 7 - 1 \times 160$$
$$\rightarrow \; 7^{-1}\, mod\; 160 = 23$$

- $PU = \{7,187\}$ is the public key
- $PR = \{23,187\}$ is the private key

RSA EXAMPLE

- **Encrypt** $M = 88$:

  $88^7 \bmod 187 = [(88^4 \bmod 187)(88^2 \bmod 187)\ (88 \bmod 187)] = 11$

- **Decrypt** $C = 11$:

  $M = 11^{23} \bmod 187 = [\ (11^{16} \bmod 187)(11^4 \bmod 187)\ (11^2 \bmod 187)\ (11 \bmod 187)]$

  - $11^2 \bmod 187 = 121$
  - $11^4 \bmod 187 = 121^2 \bmod 187 = 55$
  - $11^8 \bmod 187 = 55^2 \bmod 187 = 33$
  - $11^{16} \bmod 187 = 33^2 \bmod 187 = 154$

- $M = 154 \times 55 \times 121 \times 11 \bmod 187 = 88$

- The above algorithm is called **square-and-multiply algorithm** and often used for fast modular exponentiation.

EQUIVALENT SECURITY OF RSA

- Public-key cryptography complements rather than replaces symmetric cryptography

- There is nothing in principle to make public-key crypto more secure than symmetric crypto

- Public-key crypto does not make symmetric crypto obsolete: it has its advantages but also its (major) drawbacks such as speed

- Due to its low speed, it is mostly confined to key management and digital signatures