Cryptography for Cryptocurrency

Behnam Bahrak

PUBLIC KEY CRYPTOGRAPHY

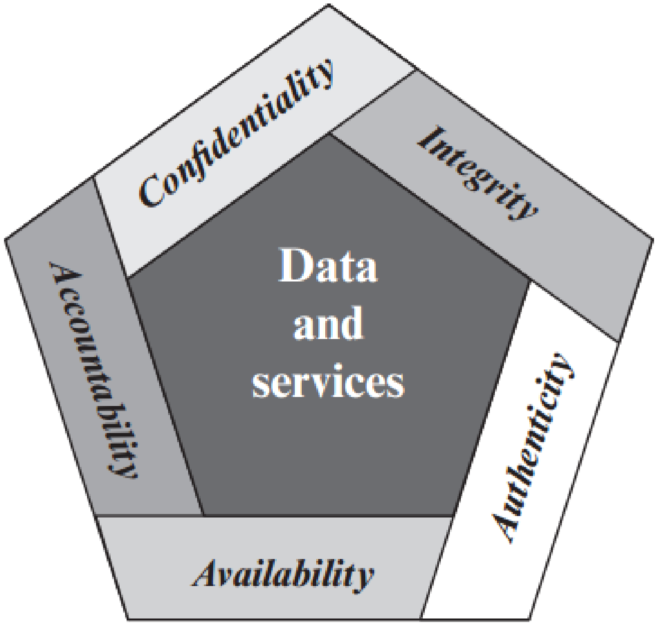
CIA Triad

- **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.

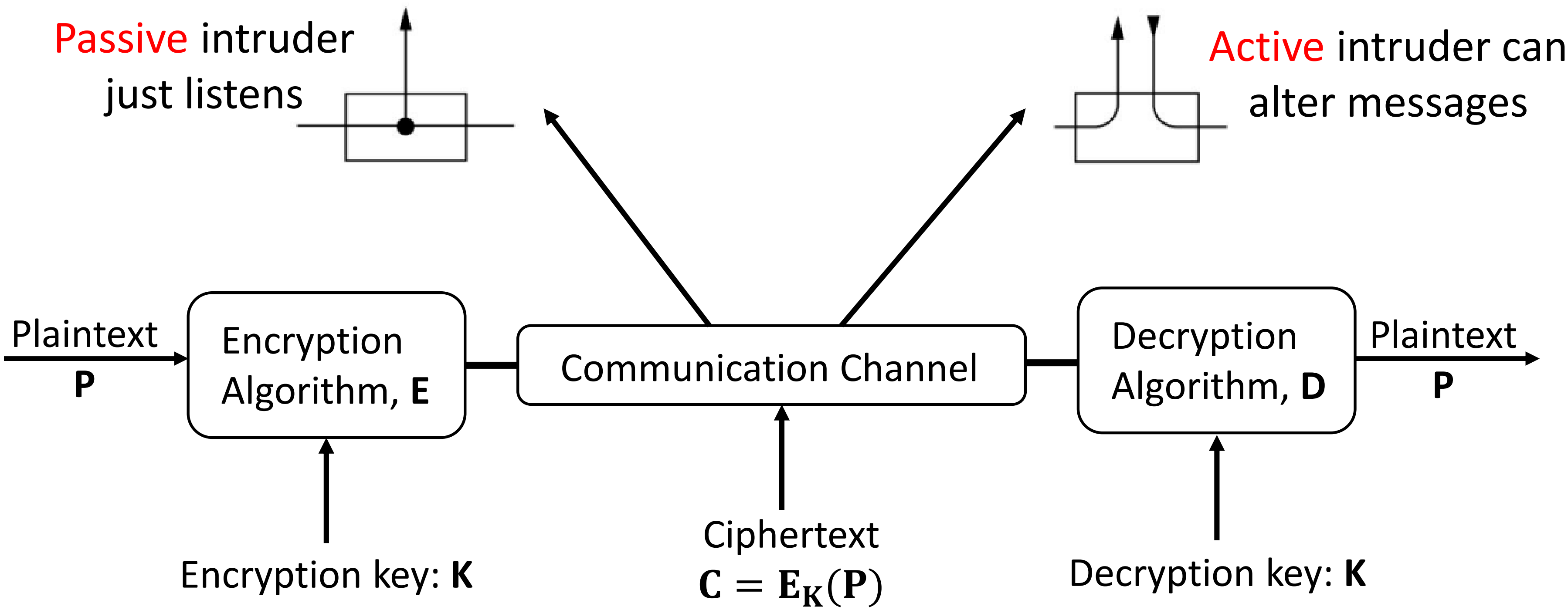
Other Security Requirements

- **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.

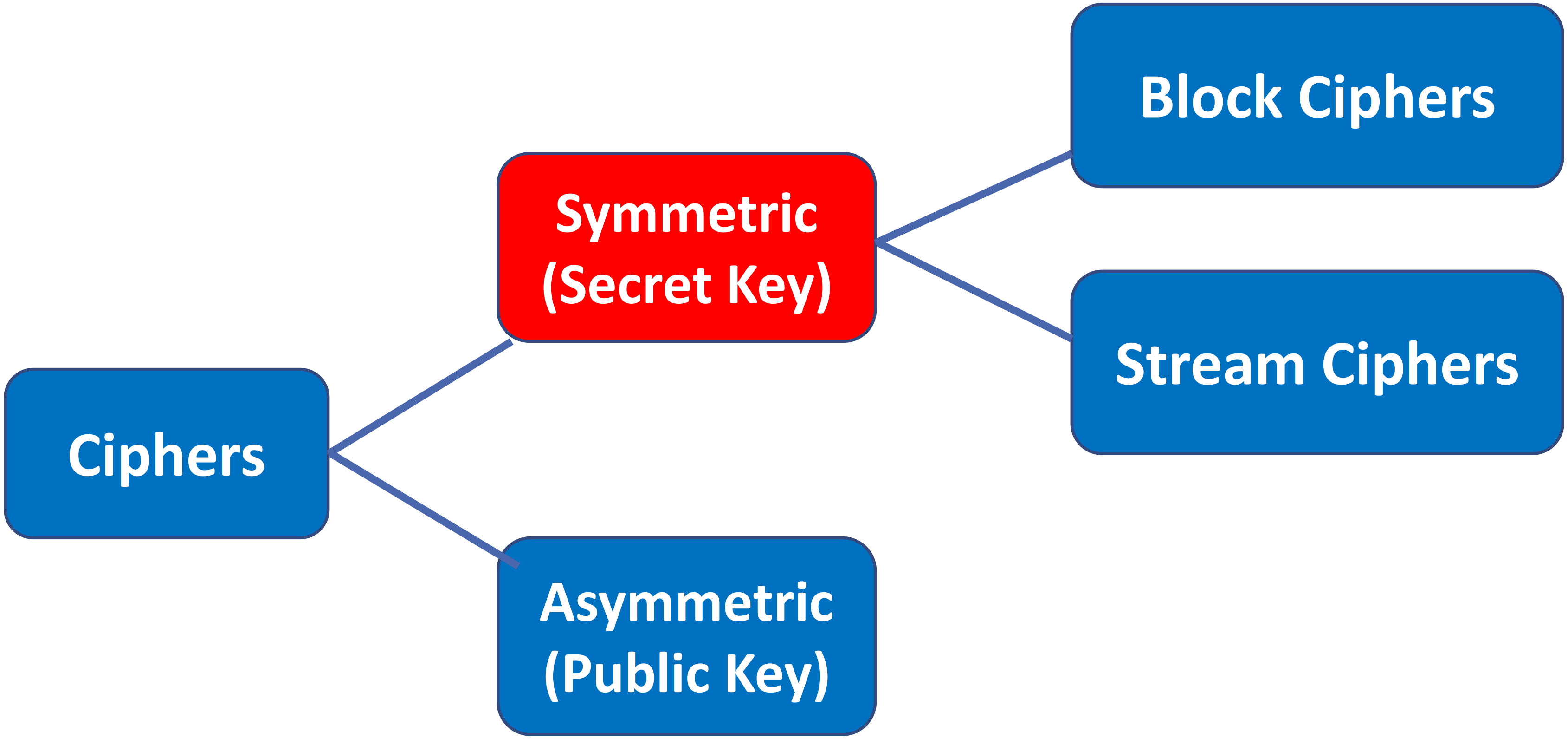
- **Accountability**: The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity.
  - We must be able to trace a security breach to a responsible party.
  - Systems must keep records of their activities to permit later forensic analysis to trace security breaches or to aid in transaction disputes.



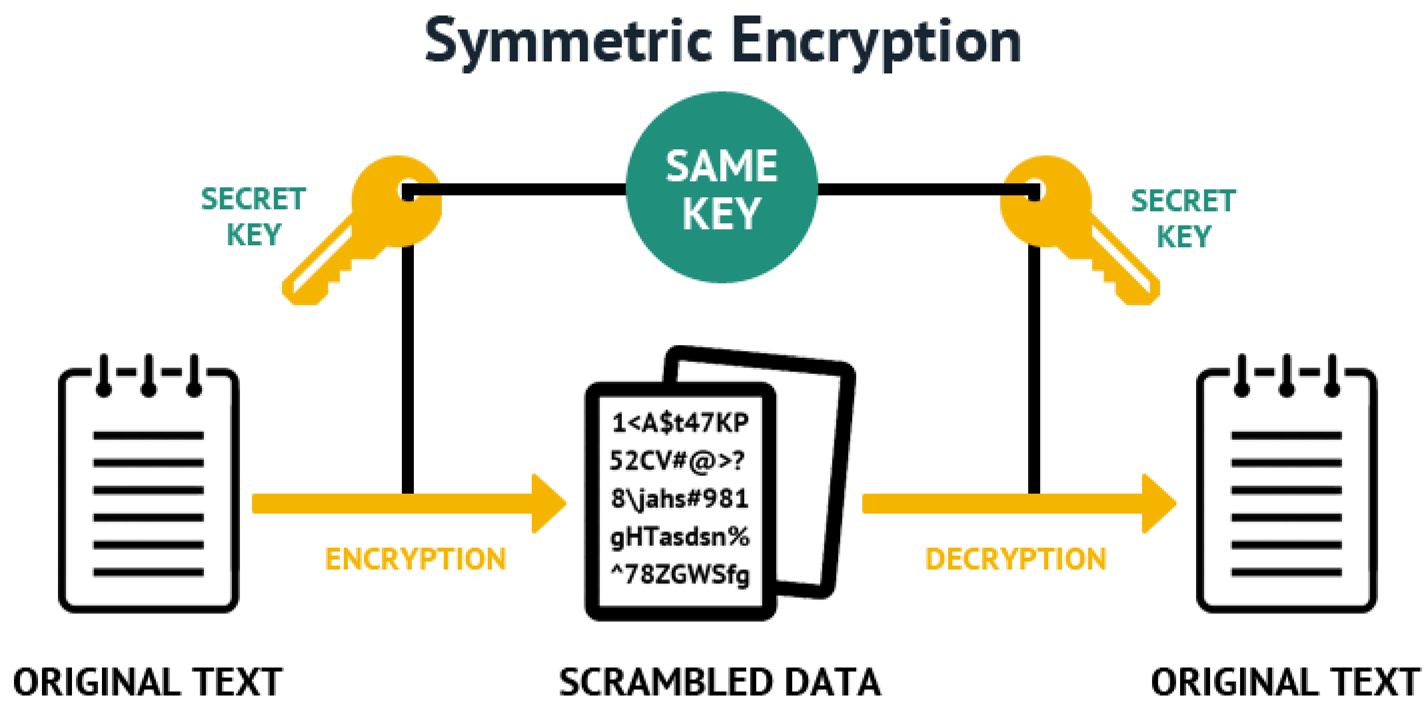
Basic Situation in Cryptography



Classification of Cryptosystems



Symmetric Cryptosystems

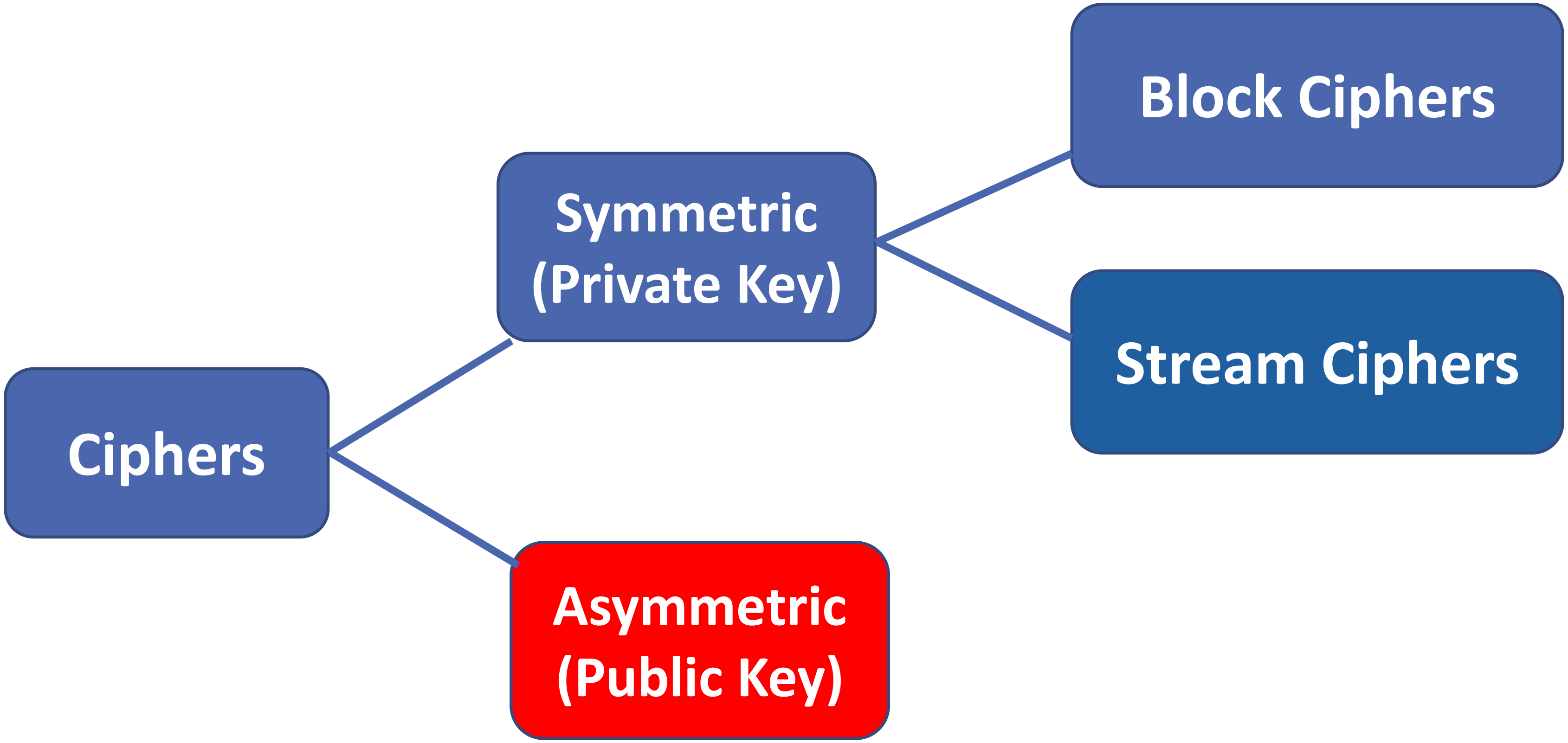


- Advanced Encryption Standard (AES) is one of the most used symmetric cryptosystems that uses keys of size 128, 192, or 256 bits.

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^{68} grains of sand on earth
  - 2^{92} atoms in the average human body
  - 2^{128} possible keys in AES-128
  - 2^{170} atoms in the earth
  - 2^{190} atoms in the sun
  - 2^{192} possible keys in AES-192
  - 2^{233} atoms in the Milky Way galaxy
  - 2^{256} possible keys in AES-256

Classification of Cryptosystems



Disadvantage of Symmetric Ciphers

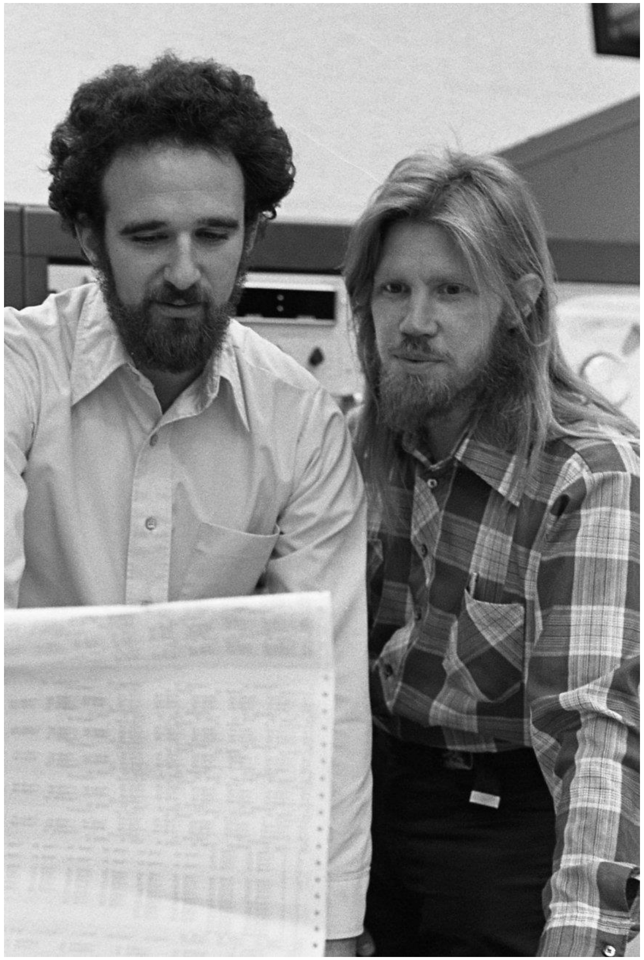
- **Key management**: How to transfer the secret key

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 the owner 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



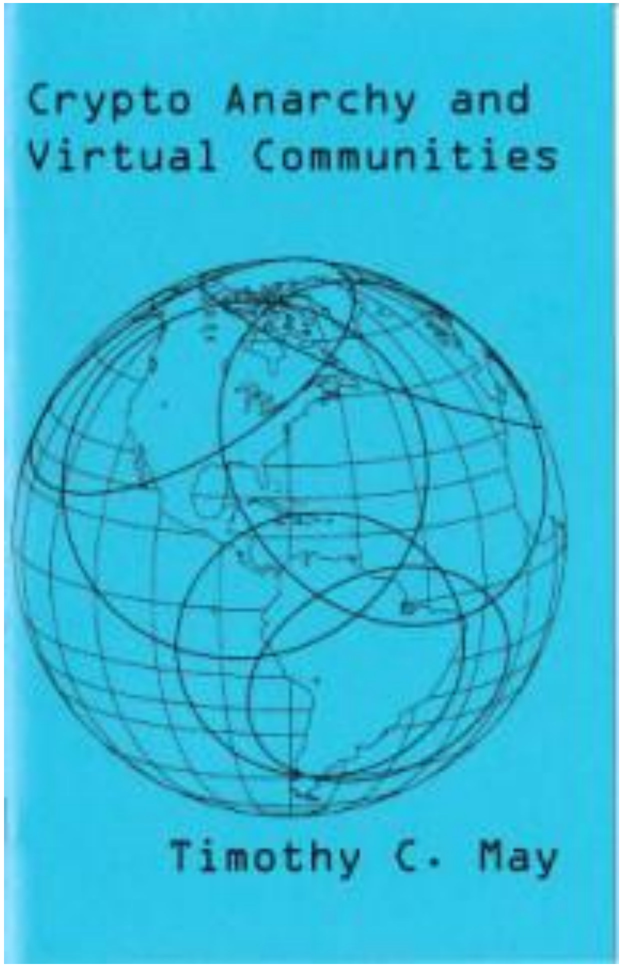
Martin Hellman & Whitfield Diffie

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

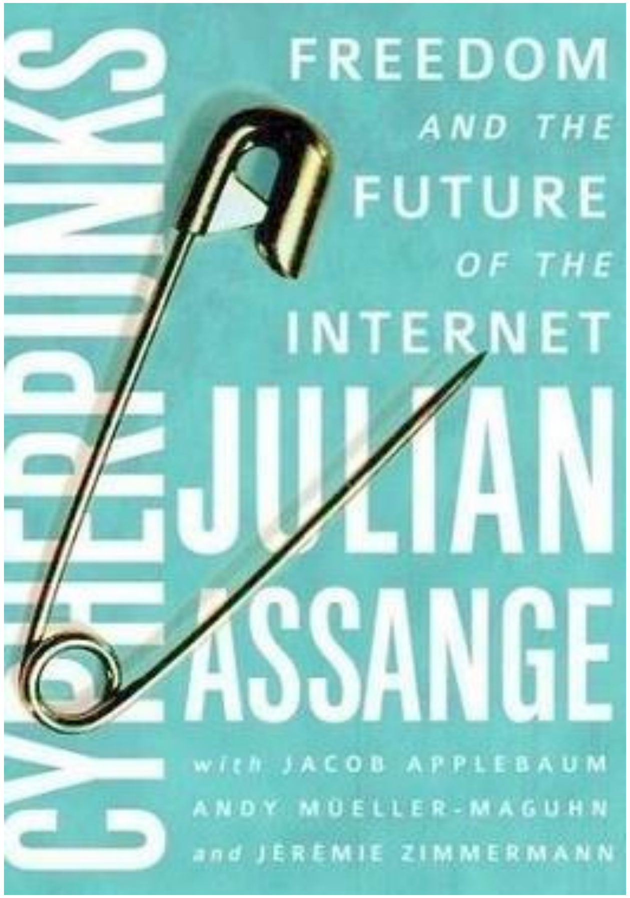
Cypherpunks and Crypto-Anarchists



A group of libertarians formed the "Cypherpunk Mailing List" to exchange information on privacy, cryptography and online liberty.

Noteworthy Cypherpunks

- **Jacob Appelbaum**: A core member of Tor project
- **Julian Assange**: WikiLeaks founder
- **Adam Back**: inventor of Hashcash
- **Philip Zimmermann**: original creator of PGP
- **Nick Szabo**: inventor of Bitgold
- **Bruce Schneier**: well-known security author
- **Hal Finney**: cryptographer, main author of PGP 2.0
- **Satoshi Nakamoto**



Some Notation

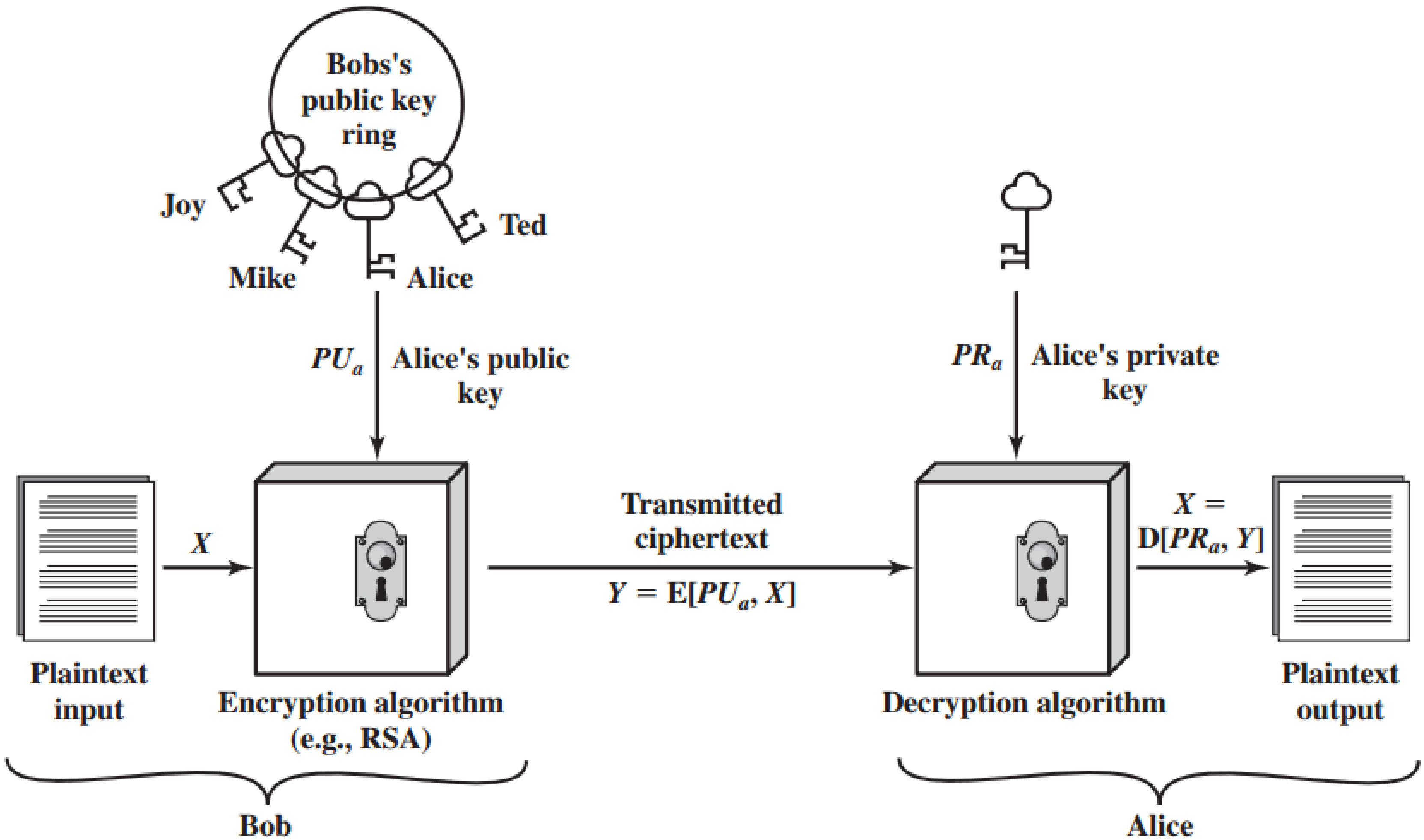
- The public key of user A will be denoted PU_A
- The private key of user A will be denoted PR_A
- Encryption method will be a function E
- Decryption method will be a function D
- If B wishes to send a plain message X to A, then he sends the ciphertext:

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

- The intended receiver A will decrypt the message:

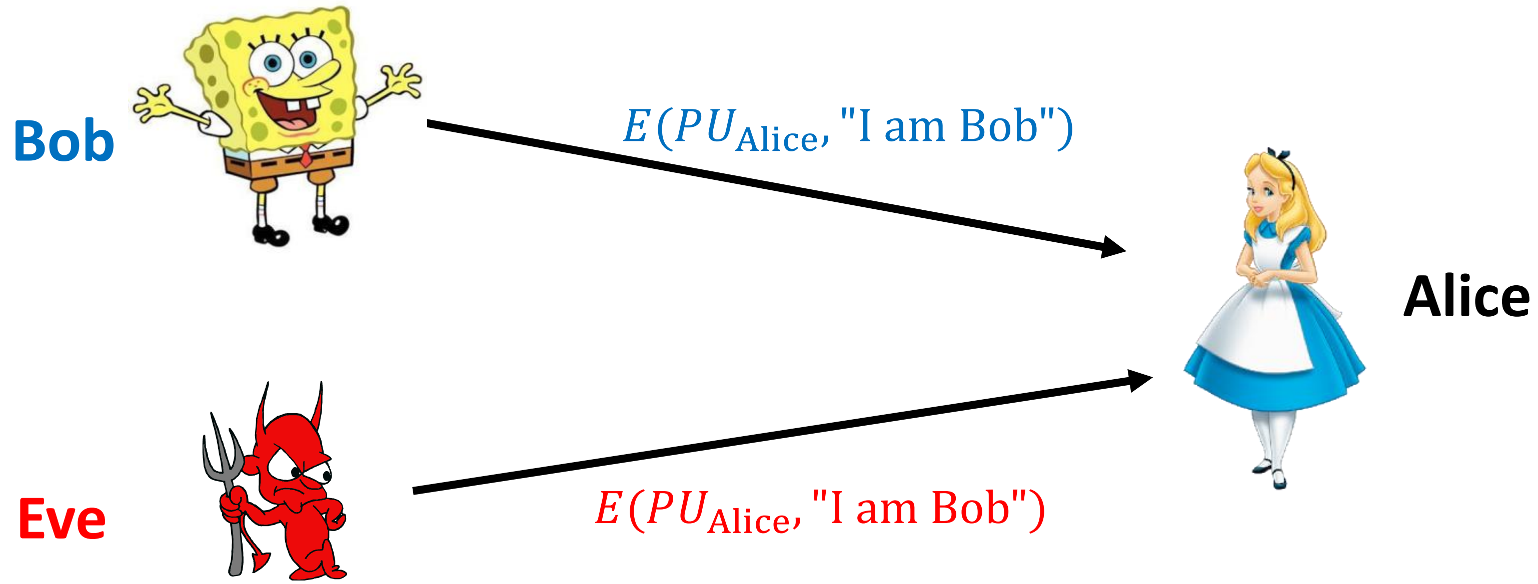
$$D(PR_A, Y) = X$$

Public Key Scheme for Confidentiality

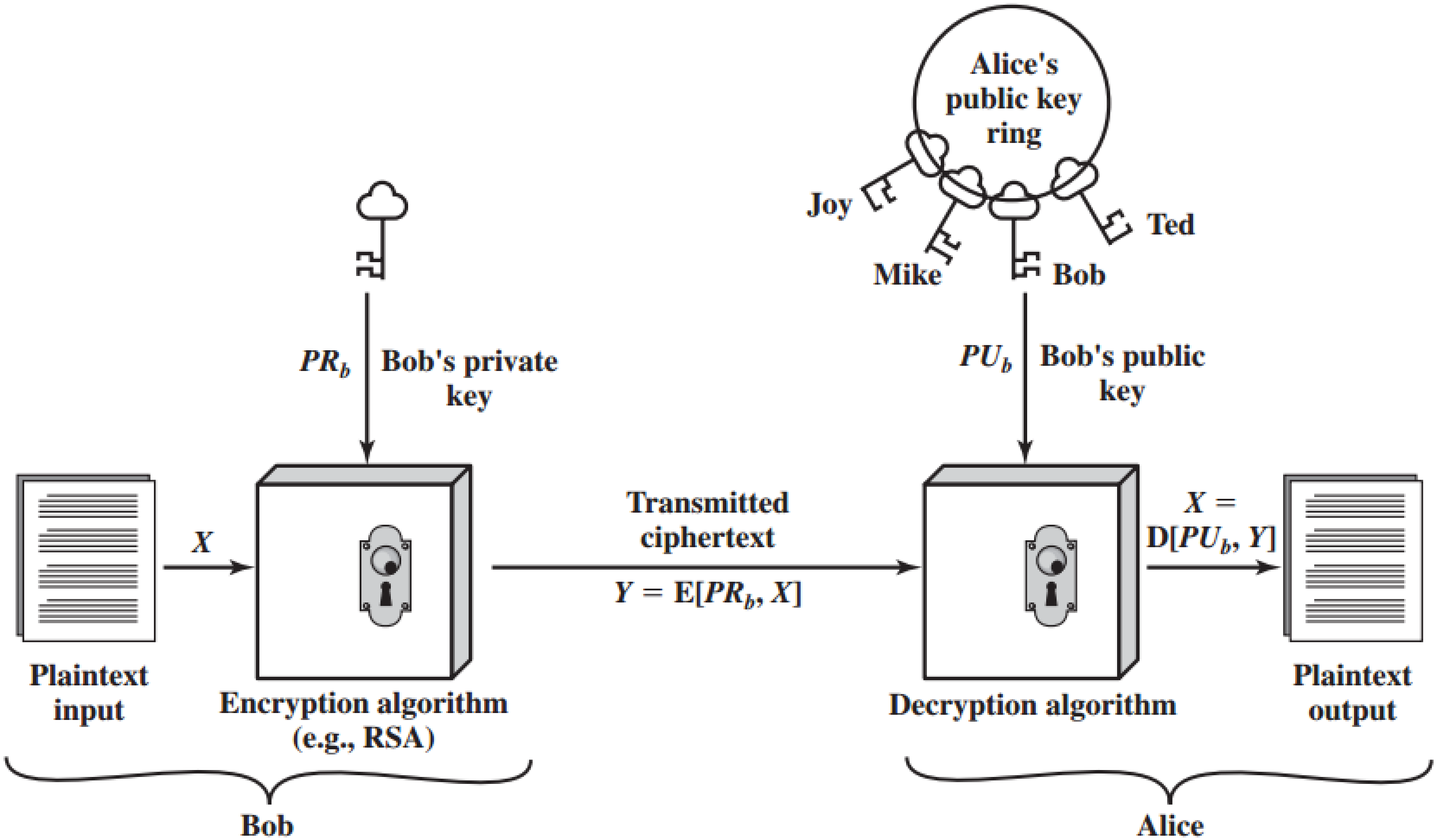


A first attack on the public-key scheme

- **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



Public Key Scheme for Authentication



Confidentiality and Authentication

Bob

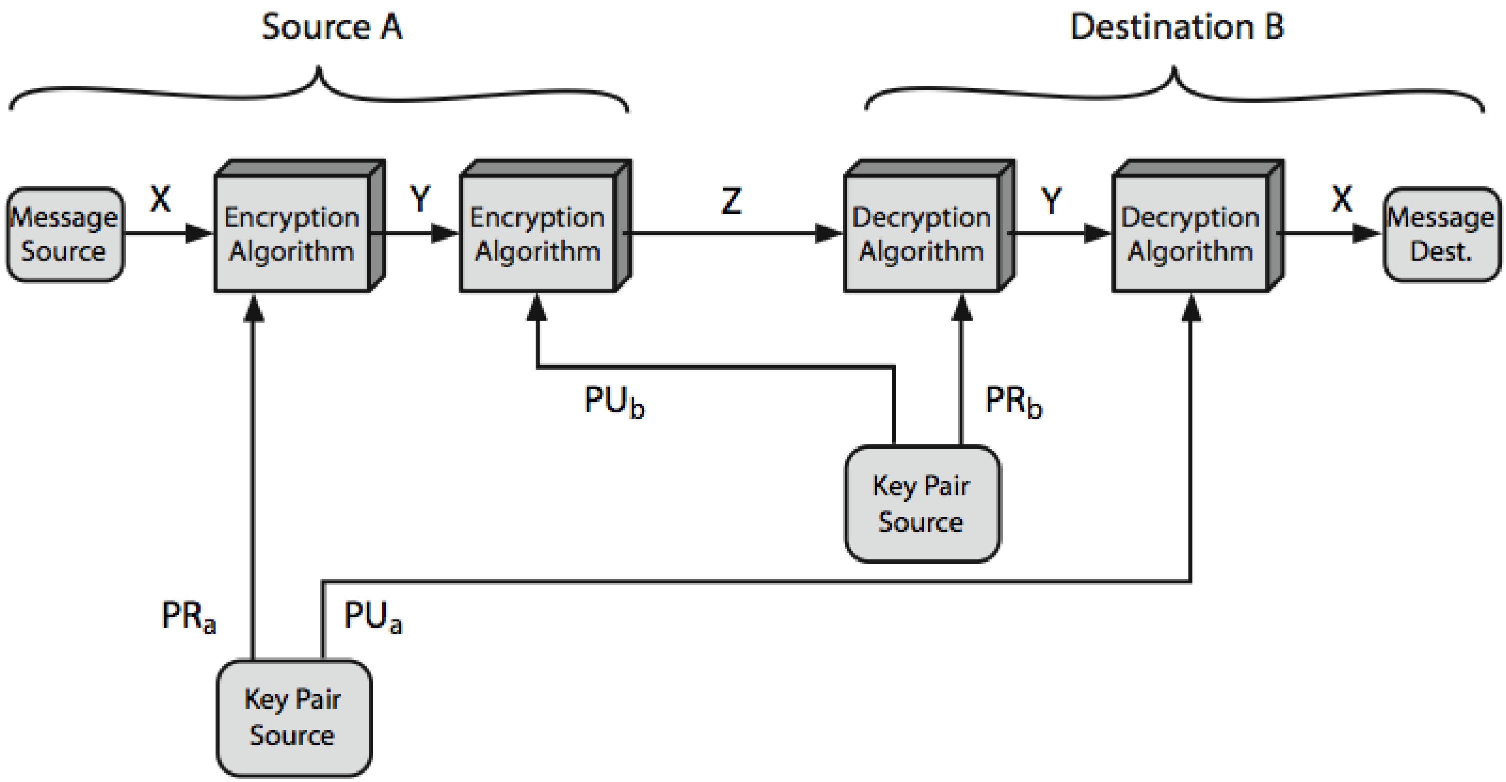
$E(PU_{\text{Alice}}, E(PR_{\text{Bob}}, \text{"I am Bob"}))$

Alice

- ➢Alice decrypts $E(PU_{\text{Alice}}, E(PR_{\text{Bob}}, \text{"I am Bob"}))$ using her private key PR_{Alice} and obtains $E(PR_{\text{Bob}}, \text{"I am Bob"})$.

- ➢Alice decrypts $E(PR_{\text{Bob}}, \text{"I am Bob"})$ using Bob's public key PU_{Bob} to get the plaintext and ensure that it comes from Bob.

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

RSA

- One of the first public-key cryptosystems by Rivest, Shamir, Adleman was introduced in 1977: RSA
- In RSA the plaintext and the ciphertext are integers between 0 and $n-1$ for some fixed n
- 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**: number M with $M < n$
  - **Encryption**: $C = M^e \, mod \, n$
  - **Decryption**: $C^d \, mod \, n = M^{de} \, mod \, n = M$
  - **Public key**: $PU = \{e, n\}$ and **Private key**: $PR = \{d\}$

Number Theory

- 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 $n = pq$.
- For any integers k, m, 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, $d \equiv e^{-1} \bmod \varphi(n)$
- To calculate the modular inverse of an integer: the extended Euclid's algorithm

DSA: Digital Signature Algorithm

- **What Is DSA (Digital Signature Algorithm)?**
- DSA is a United States Federal Government standard for digital signatures.
- It was proposed by the National Institute of Standards and Technology (NIST) in August 1991
- DSA is based on ElGamal public-key cryptosystem
- Elliptic Curve Digital Signature Algorithm (ECDSA) is an update of DSA algorithm adapted to use elliptic curves.
  - Bitcoin uses ECDSA for signing transactions.

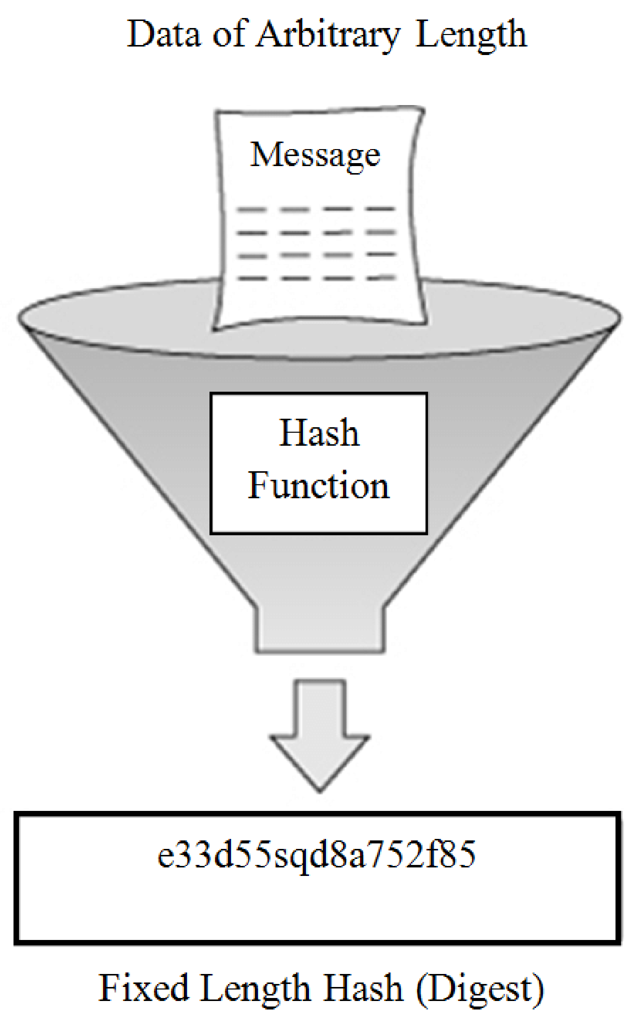
CRYPTOGRAPHIC HASH FUNCTIONS

Hash Functions

- A fixed-length hash value h is generated by a hash function H that takes as input a message M of arbitrary length: $h = H(M)$

- A simple hash function:
  - Bit-by-bit XOR of plaintext blocks:

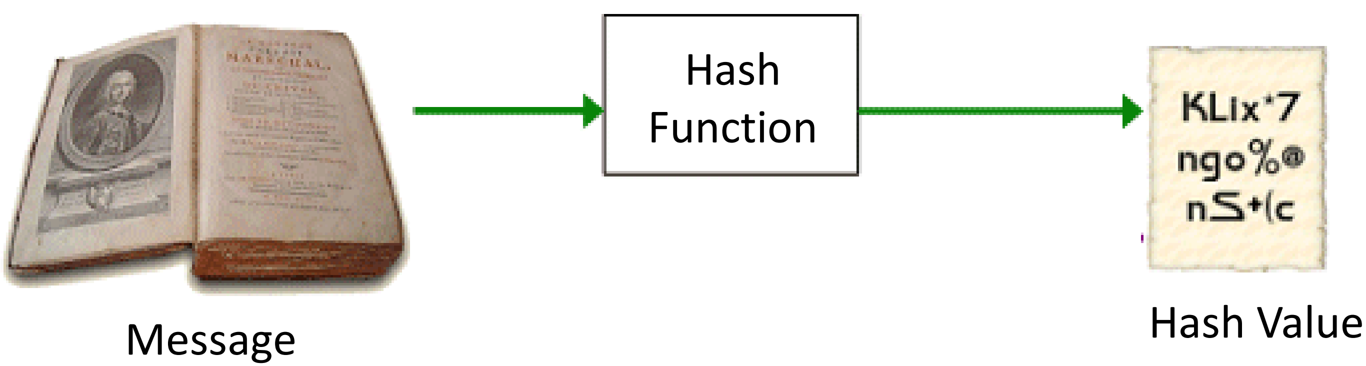
    $h = M_1 \oplus M_2 \oplus \cdots \oplus M_N$



Cryptographic Hash Function

- Requirements for a cryptographic hash function:
  - H can be applied to a message of any size
  - H produces fixed-length output
  - It is easy to compute $H(M)$
  - Preimage resistance property (one-wayness):

for a given h, it is computationally infeasible to find M such that $H(M) = h$



Cryptographic Hash Function

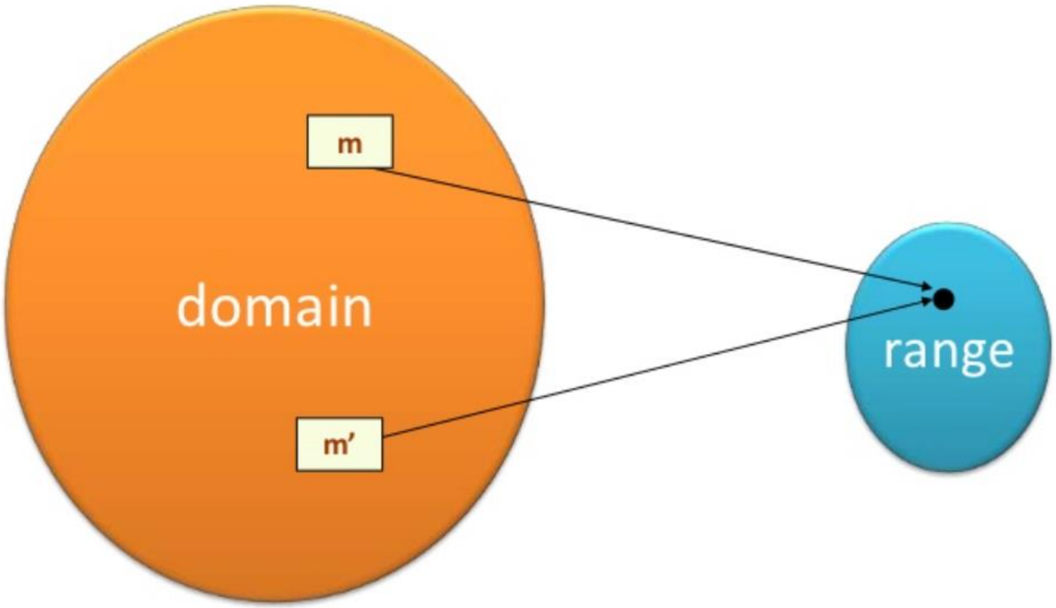
- **Second-preimage resistance property** (weak collision resistance):

  for a given M, it is computationally infeasible to find M' such that

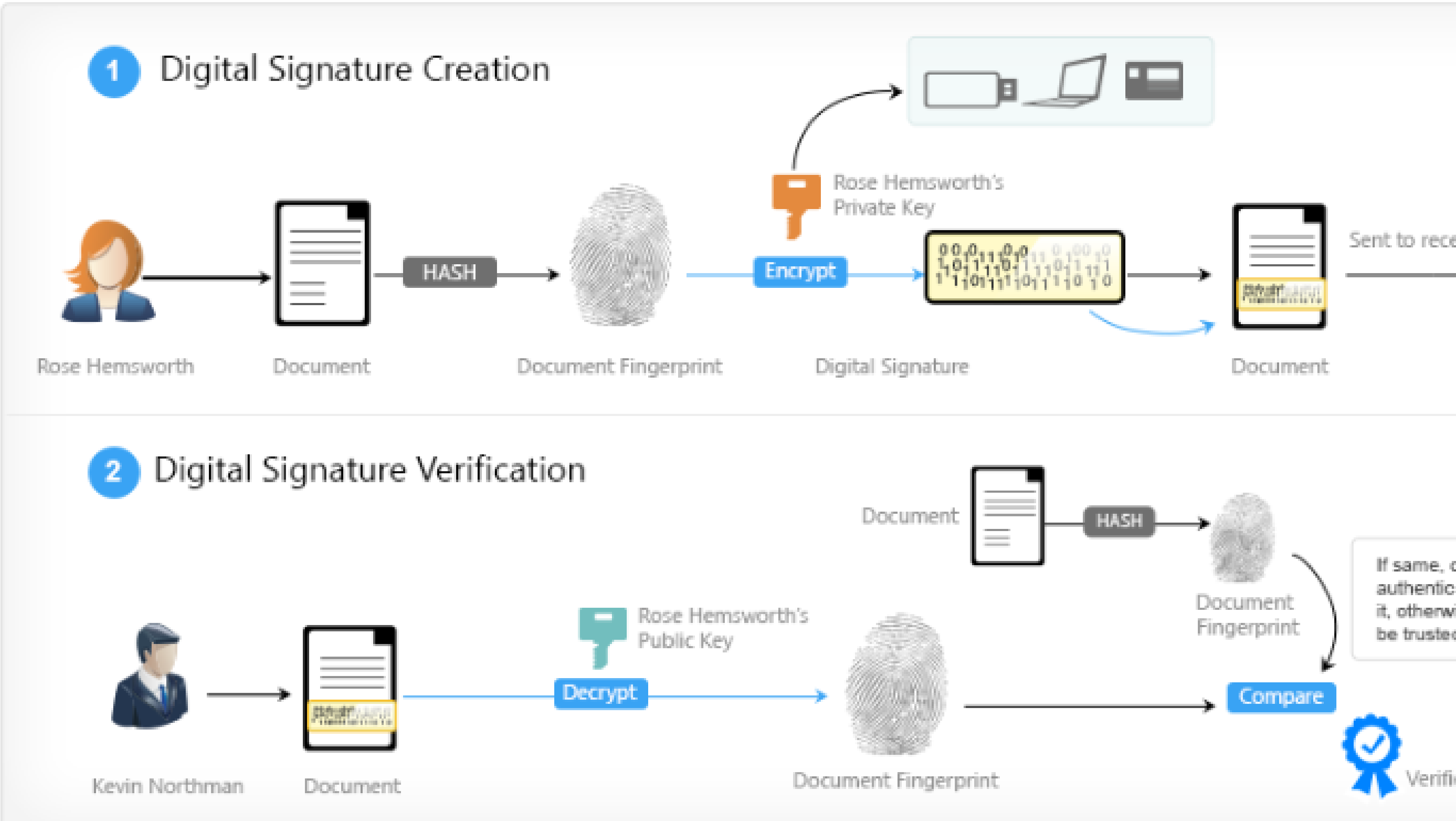
  $$H(M') = H(M)$$

- **Collision-resistance property** (strong collision resistance):

  it is computationally infeasible to find M, M' with $H(M) = H(M')$

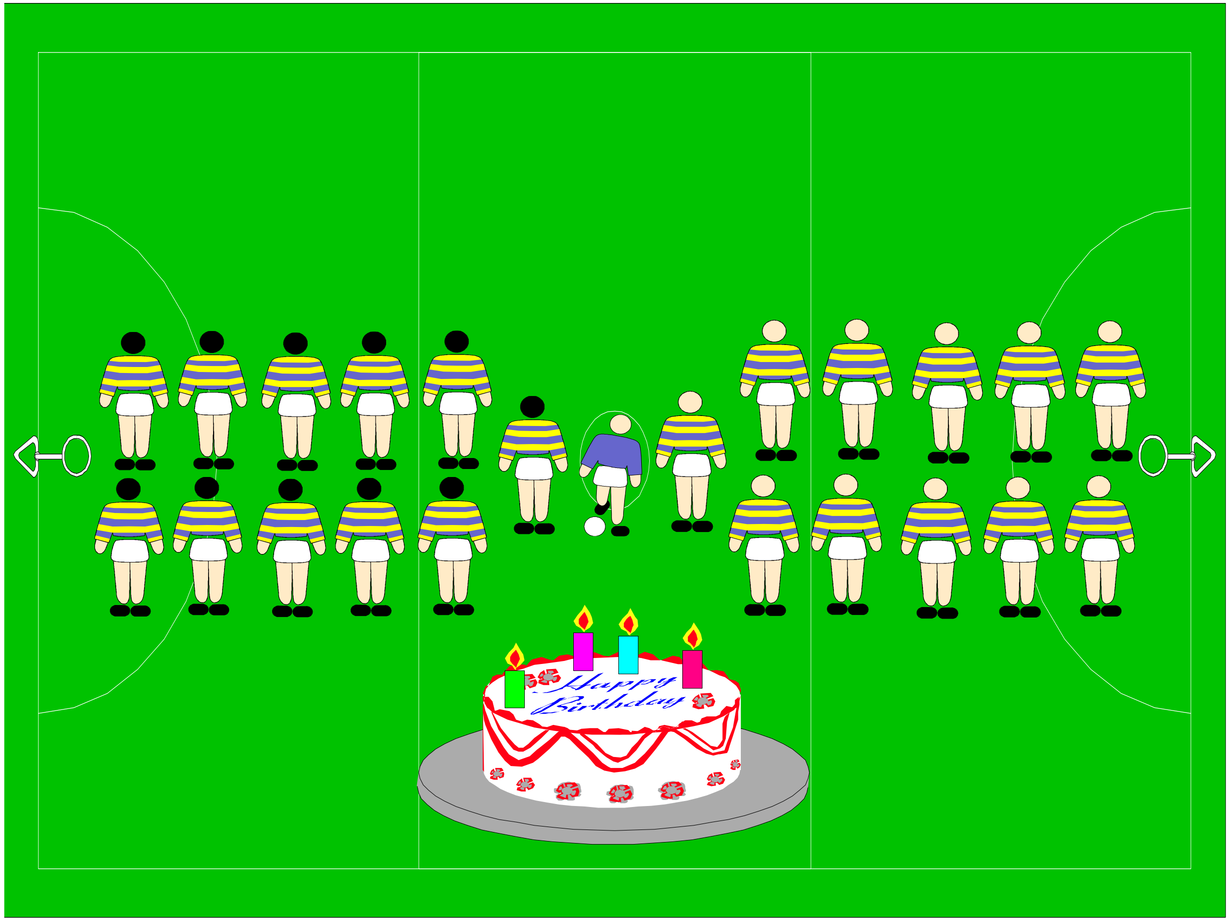


Digital Signature Scheme



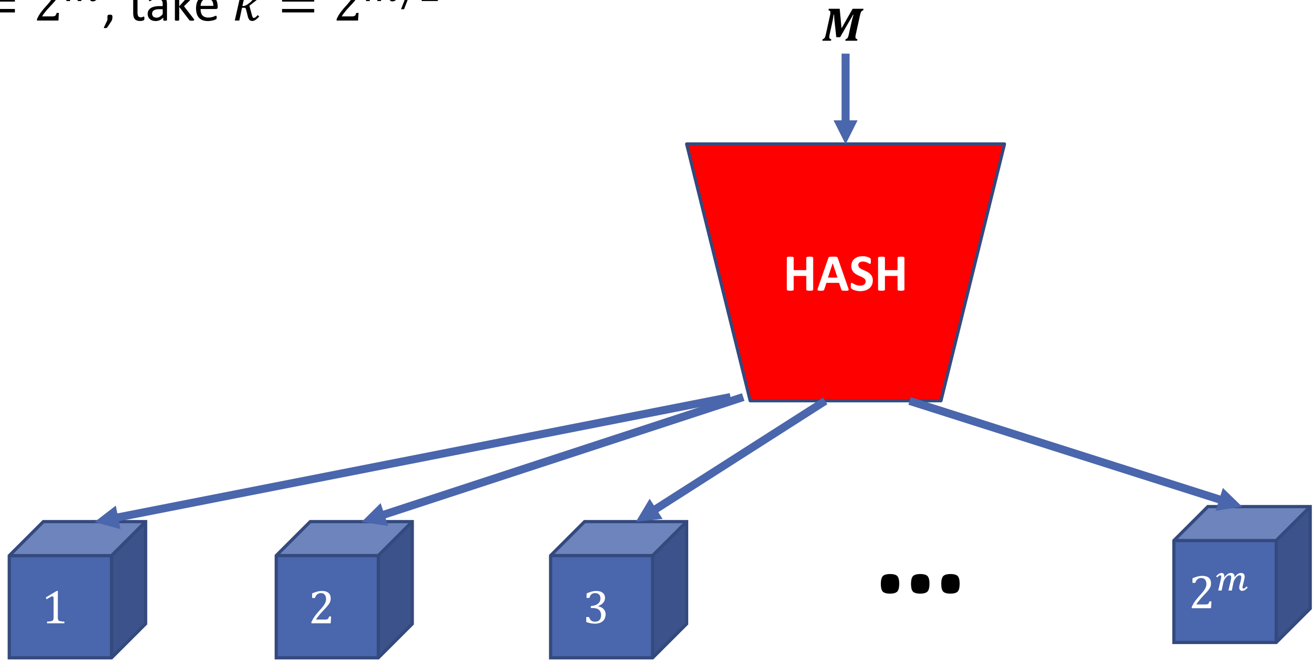
Birthday Attack

- Birthday paradox: Given at least 23 people, the probability of having two people with the same birthday is more then 0.5



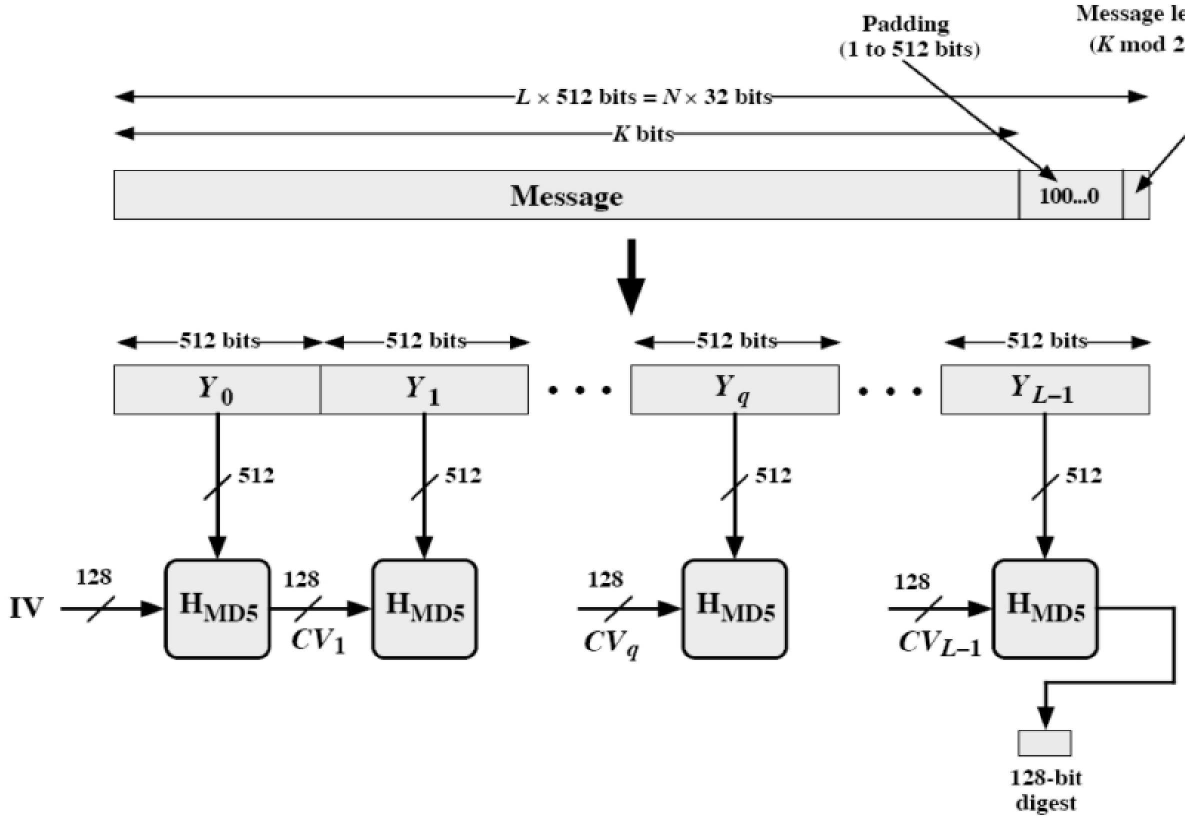
Birthday Attack

- **General case**: Given two sets X, Y each having k elements from the set $\{1,2,\dots,N\}$, how large should k be so that the probability that X and Y have a common element is more than 0.5?
  - Answer: k should be larger $\sqrt{N}$
  - If $N = 2^m$, take $k = 2^{m/2}$



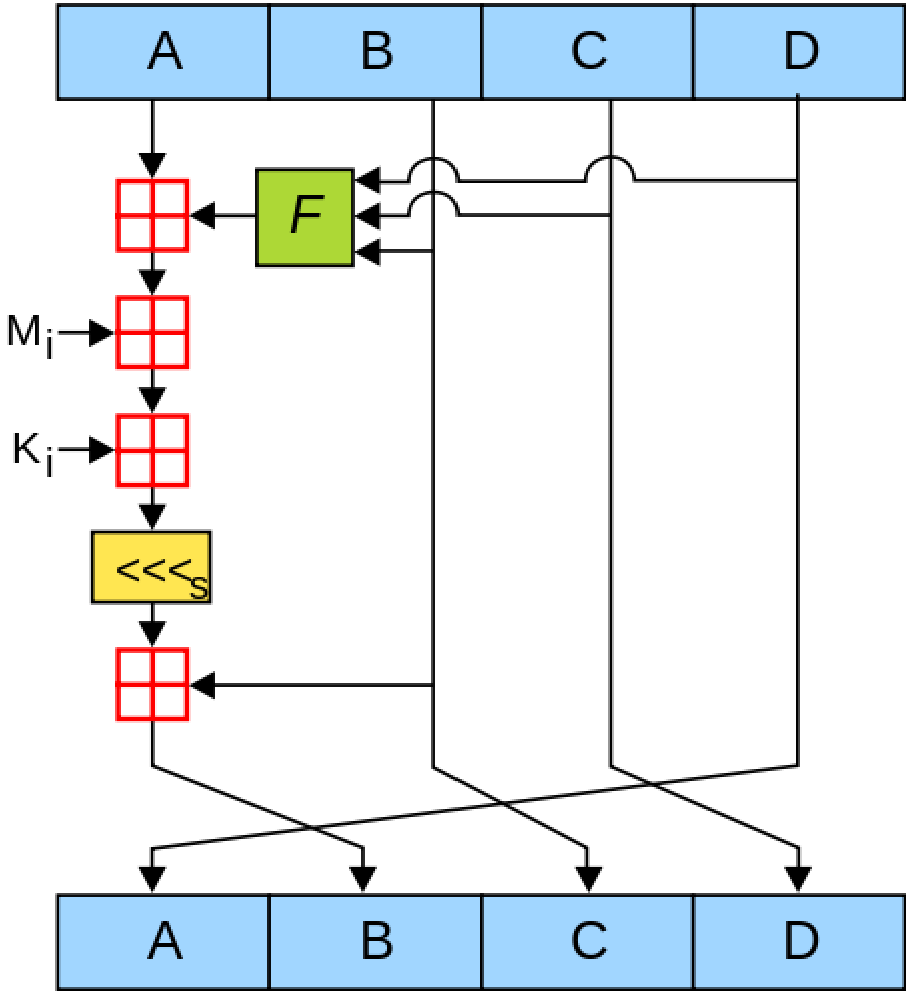
MD5

- Most popular hash algorithm until recently
  - Developed by Ron Rivest at MIT in 1991.
  - For a message of arbitrary length, it produces an output of 128 bits



MD5 Operations

- MD5 consists of 64 of these operations, grouped in four rounds of 16 operations.
- F is a nonlinear function; one function is used in each round.
- M_i denotes a 32-bit block of the message input, and K_i denotes a 32-bit constant, different for each operation.
- $\lll_s$ denotes a left bit rotation by s places; s varies for each operation.
- $\boxplus$ denotes addition modulo 2^{32}.

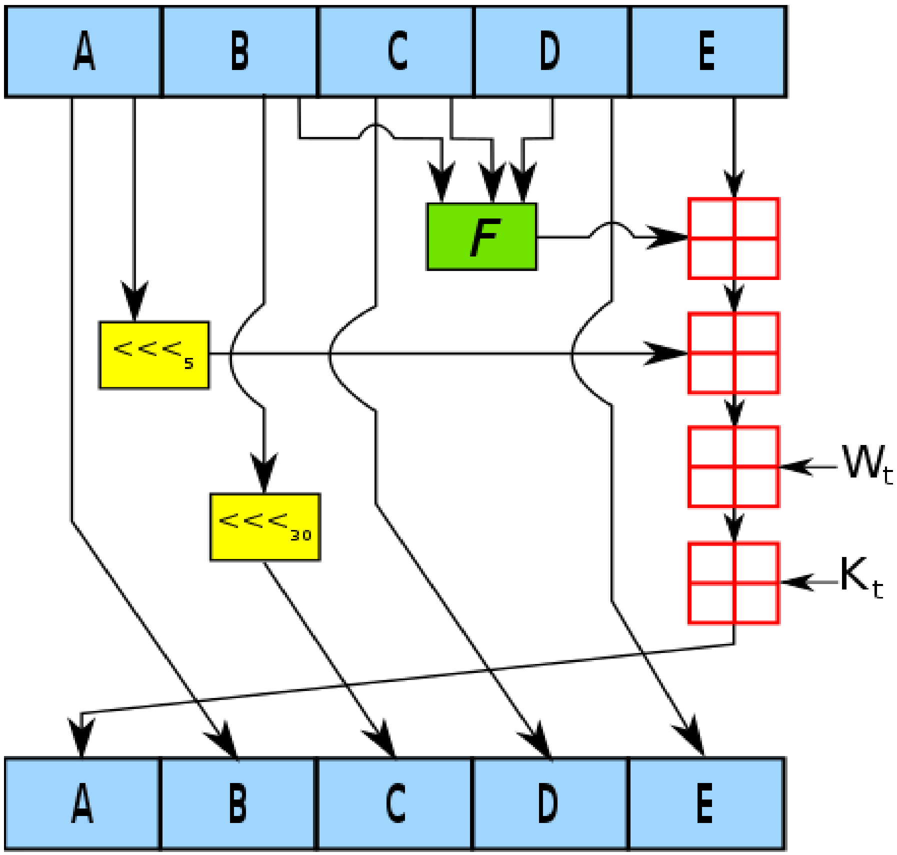


SHA-1

- Developed by NSA and adopted by NIST in FIPS 180-1 (1993)
- Part of a family of 3 hashes: SHA-0, SHA-1, SHA-2
  - SHA-1 most widely used
- Design based on MD4 (previous version of MD5)
- Takes as input any message of length up to 2^{64} bits and gives a 160-bit message digest
- Microsoft, Google, Apple and Mozilla have all announced that their respective browsers will stop accepting SHA-1 SSL certificates by 2017.
- On February 23, 2017 CWI Amsterdam and Google announced they had performed a collision attack against SHA-1, publishing two dissimilar PDF files which produce the same SHA-1 hash as proof of concept.

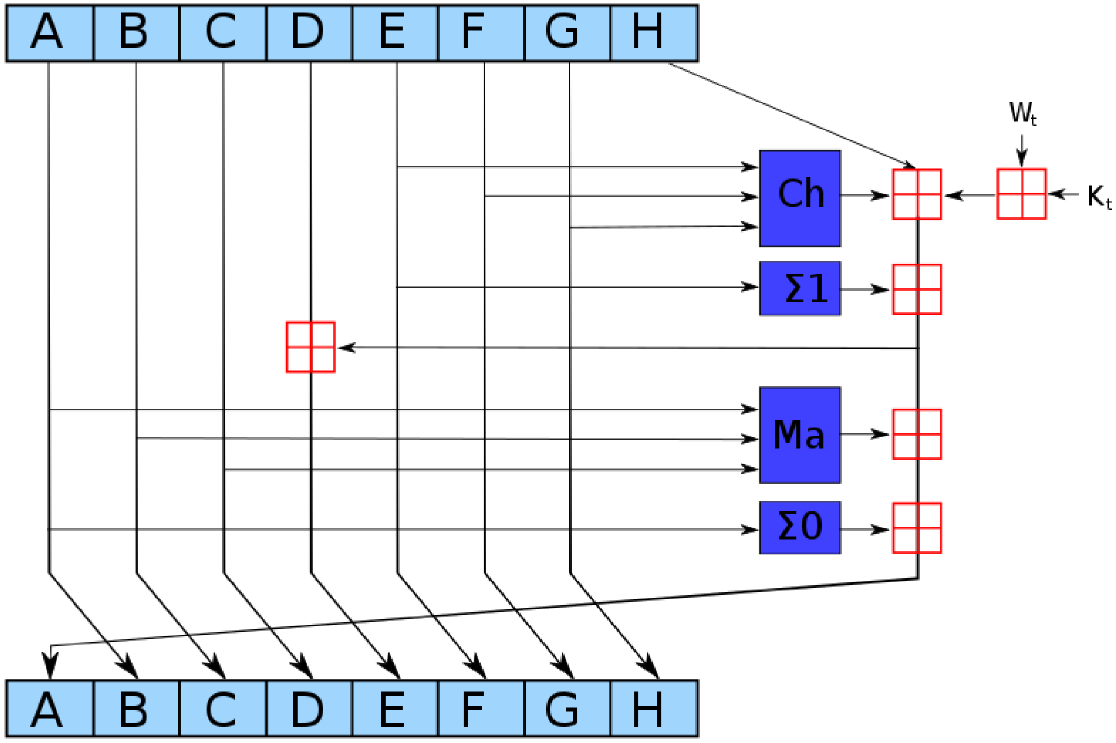
SHA-1 Operation

- Structure very similar to MD4 and MD5.
  - Secret design criteria
- Stronger than MD5 because of longer message digest
- Slower than MD5 because of more rounds
- Best known attacks:
  - 2015: SHAppening
  - 2017: SHAttered
    - Can be broken in 2^{61} iteration



SHA-2

- SHA-2 similar to SHA-1 , but with different input-output length
- The algorithms are collectively known as SHA-2, named after their digest lengths: SHA-256, SHA-384, and SHA-512.
- There is no known attack against SHA-2.



Ch(E, F, G) = (E ∧ F) ⊕ (¬ E ∧ G)

Ma (A, B, C) = (A ∧ B) ⊕ (A ∧ C) ⊕ (B ∧ C)

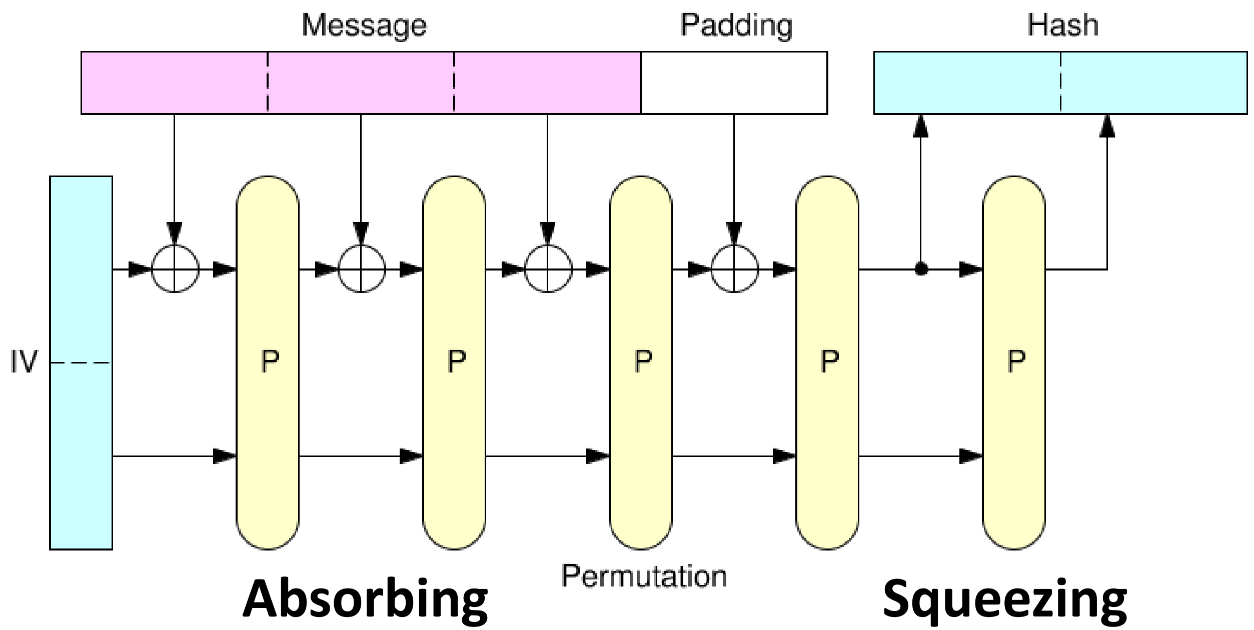
Σ0 (A) = (A ⋙ 2) ⊕ (A ⋙ 13) ⊕ (A ⋙ 22)

Σ1 (E) = (E ⋙ 6) ⊕ (E ⋙ 11) ⊕ (E ⋙ 25)

SHA-3

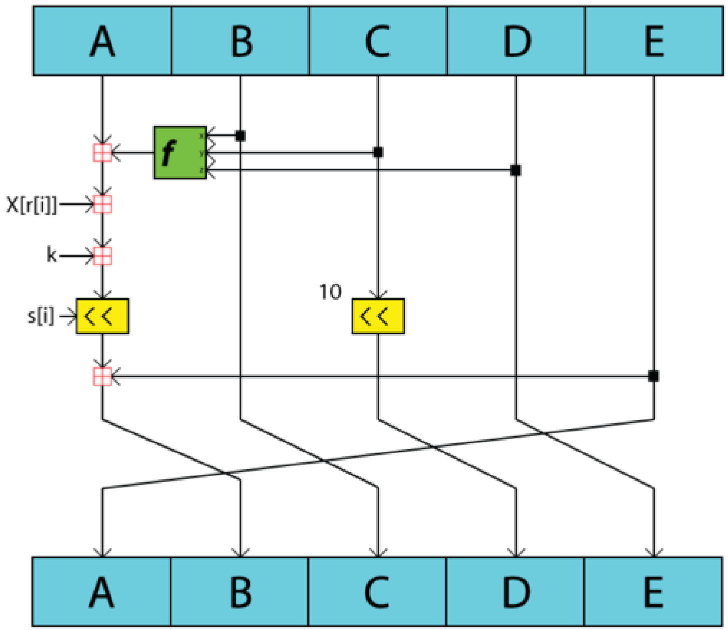
- SHA-3 is the latest member of the Secure Hash Algorithm family of standards, released by NIST on 2015 as FIPS 202.
- In 2006 NIST started to organize the NIST hash function competition to create a new hash standard, SHA-3.
  - On October 2, 2012, Keccak was selected as the winner of the competition.

Sponge construction of SHA-3:



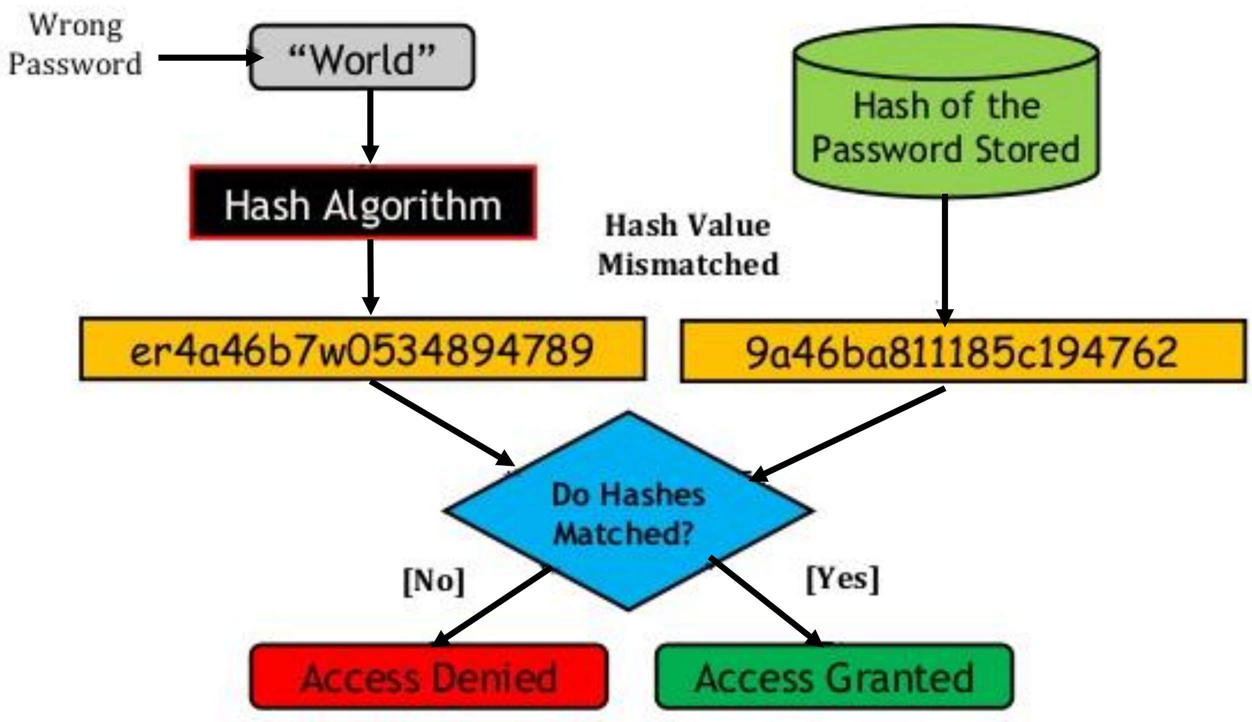
RIPEMD-160

- ➢RIPEMD (RACE Integrity Primitives Evaluation Message Digest) is a family of cryptographic hash functions developed in Katholieke Universiteit Leuven, and first published in 1996.
- ➢RIPEMD-160 is an improved, 160-bit version of the original RIPEMD, and the most common version in the family.
- ➢RIPEMD-160 was designed in the open academic community, in contrast to the NSA-designed SHA-1 and SHA-2 algorithms.
- ➢There is no known attack against RIPEMD-160.



Hashing Passwords

➢One way to reduce this danger is to only store the hash digest of each password. To authenticate a user, the password presented by the user is hashed and compared with the stored hash.



Simple Hash Commitment Scheme

- **Why are these hash properties useful?**

Consider a simple auction example:

1) Alice commits to pay a dollars for the item: she broadcasts $H(a)$
2) Bob is happy to pay b dollars for the item: he broadcasts $H(b)$

- Alice and Bob cannot be aware of each other's suggested price a and b.
  - Even the auction holder cannot reveal such a value to one of the parties.
- Alice cannot change her suggested price after knowing Bob's suggested price:
  - Impossible to find $a' \neq a$ such that $H(a') = H(a)$

Hash Functions in Bitcoin

1. Producing the public bitcoin address by hashing the public key.

2. Producing a transaction digest for use as the input in signing a transaction.

3. Producing the hash of the previous block to use in the block header in the Blockchain.

4. Producing the Merkle tree root for authenticating the transactions in a block (using hashes all the way up the tree).

5. Producing the double hash of the block (with nonces) to find a block that satisfies the difficult needed in mining.