Crypto Overview

APRICOT 2019 / Daejeon, Korea

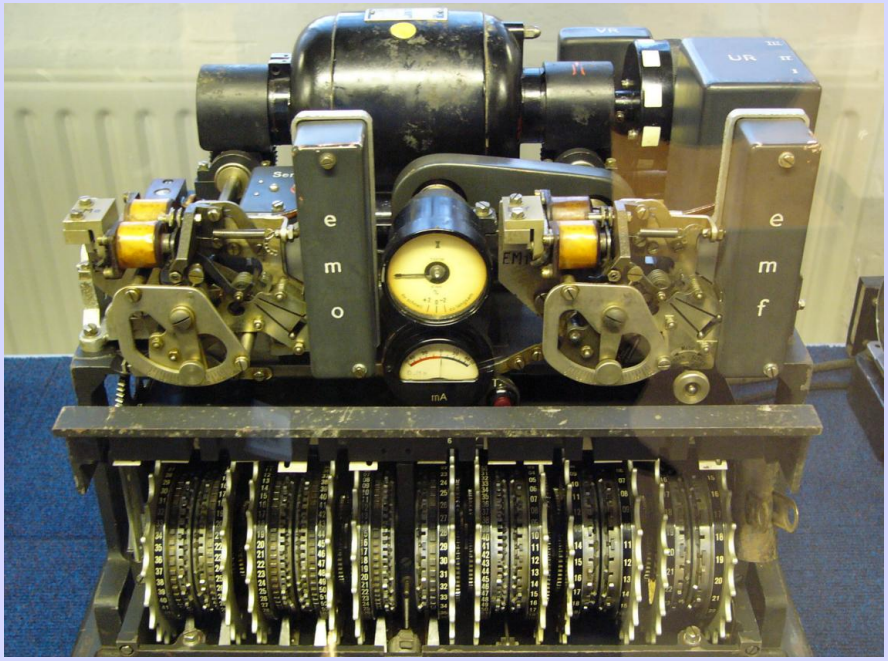
2019.02.18-22

Thanks to Steve Bellovin & Sheryl Hermoso

Overview

- What is Cryptography?
- Symmetric Key Cryptography
- Asymmetric Key Cryptography
- Block and Stream Ciphers
- Digital Signatures and Message Digests

Cryptography is Old



German Lorenz cipher machine

No, Older

Try Roman Times

Cryptography

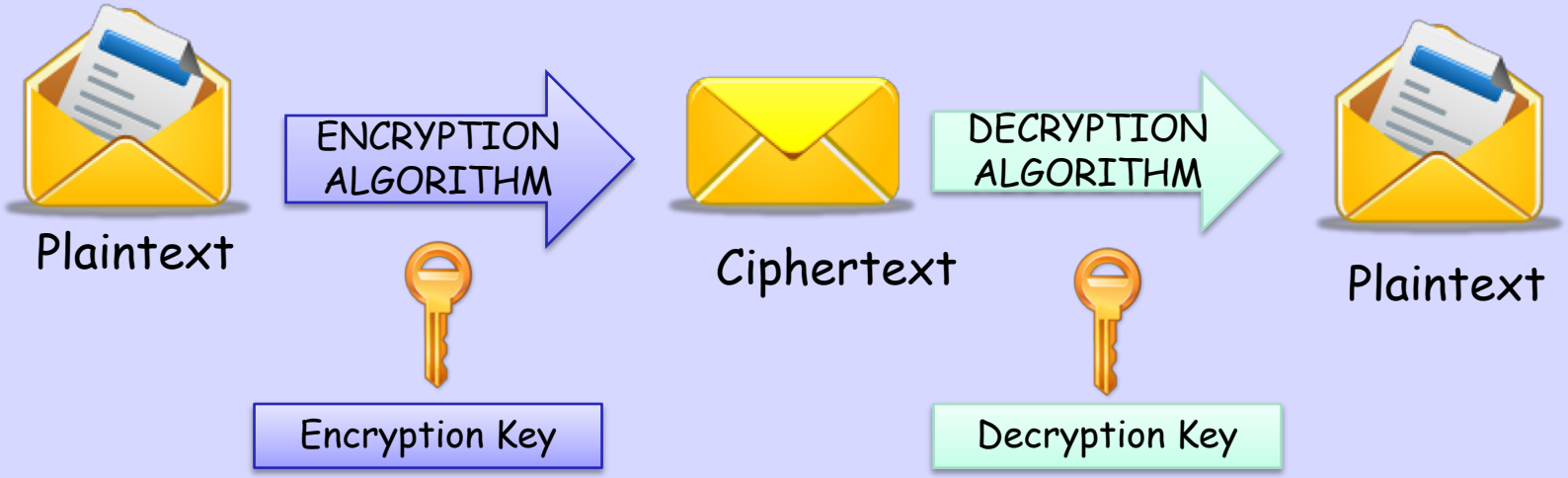
- Cryptography is the art of sending private messages secretly over public communication channels
- Cryptanalysis is the art of code breaking
- Encryption is a function of plaintext and a cryptographic key

$$C = F(P,k)$$

Notation:

Ciphertext (C)
CryptoFunction (F)
Plaintext (P)
Cryptographic Key (k)

Encryption & Decryption



Codes

- Not really encryption, but compression
- Telegraph used to charge by the word
- So you and I would have a codebook

aabbd – shipment has been sent

ljhtc – i arrive tomorrow by car

ljhdw – i will arrive wednesday

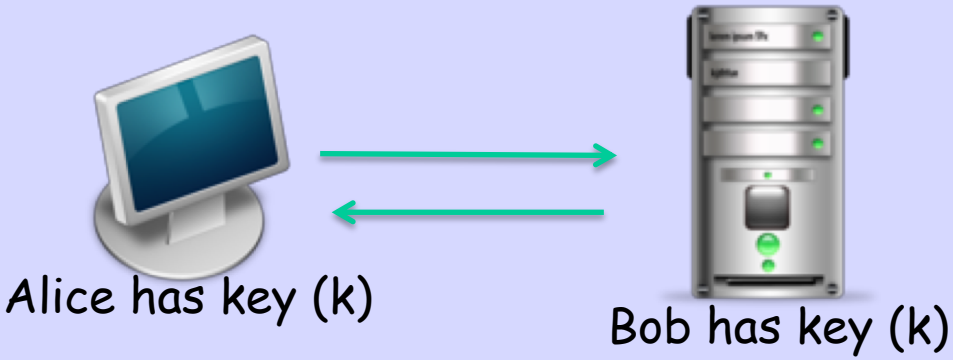
adfhd – when you receive the shipment, call head office

Typical Scenario

- Alice wants to send a "secret" message to Bob
- What are the possible problems?
  - Data can be intercepted
- What are the ways to intercept this message?
- How to conceal the message?
  - Encryption

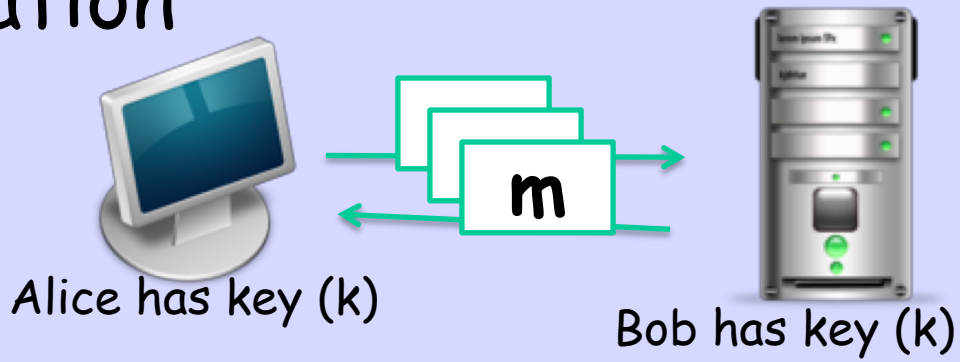
Crypto Core

Secure key establishment



Secure communication

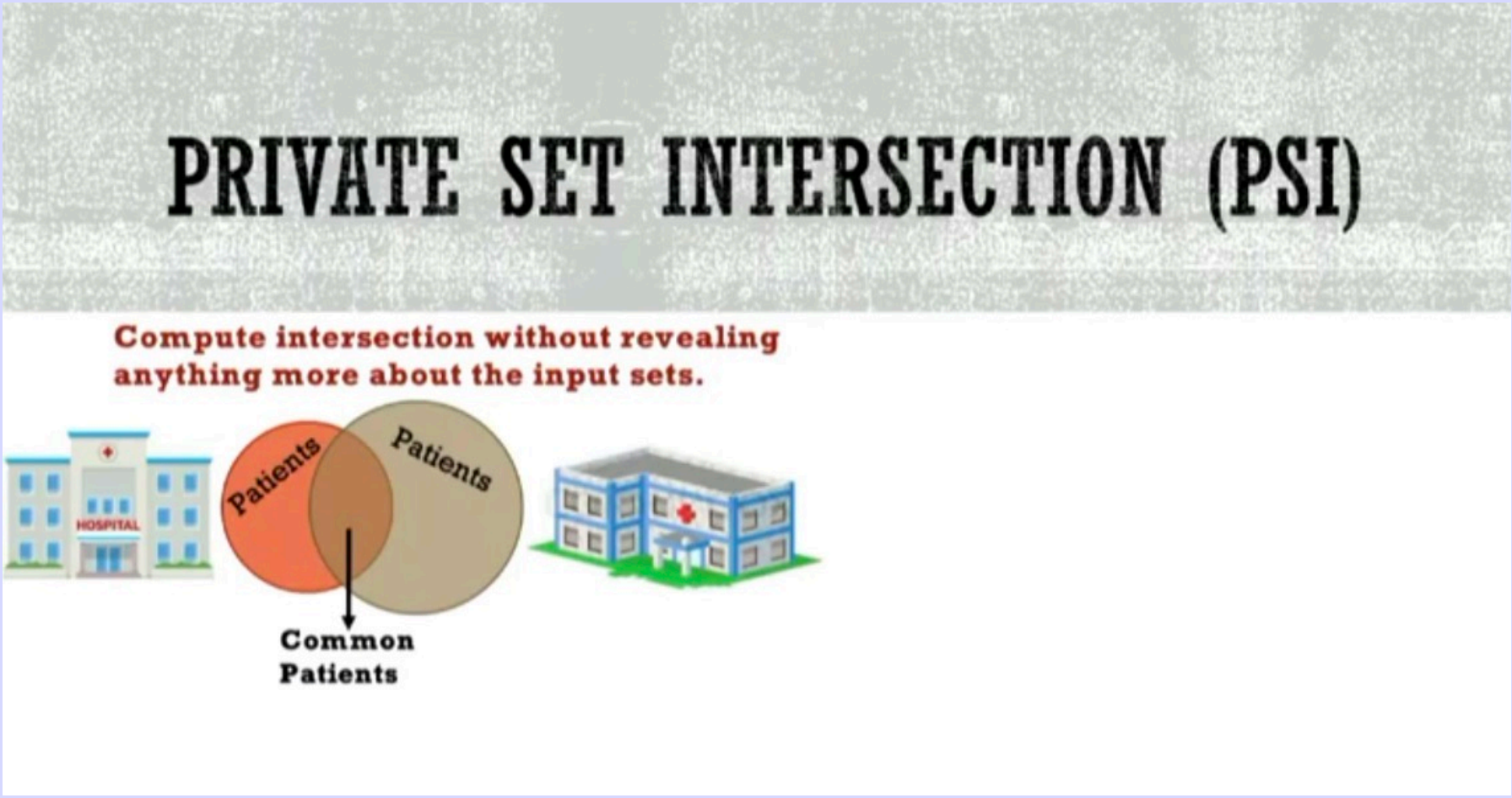
Confidentiality and integrity



Crypto Can Do More

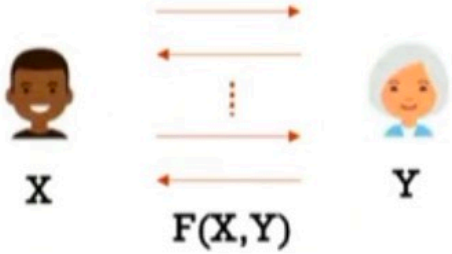
- Digital Signatures
- Anonymous communication
- Anonymous digital cash
  - Spending a digital coin without anyone knowing my identity
  - Buy online anonymously?
- Elections and private auctions
  - Finding the winner without actually knowing individual votes (privacy)

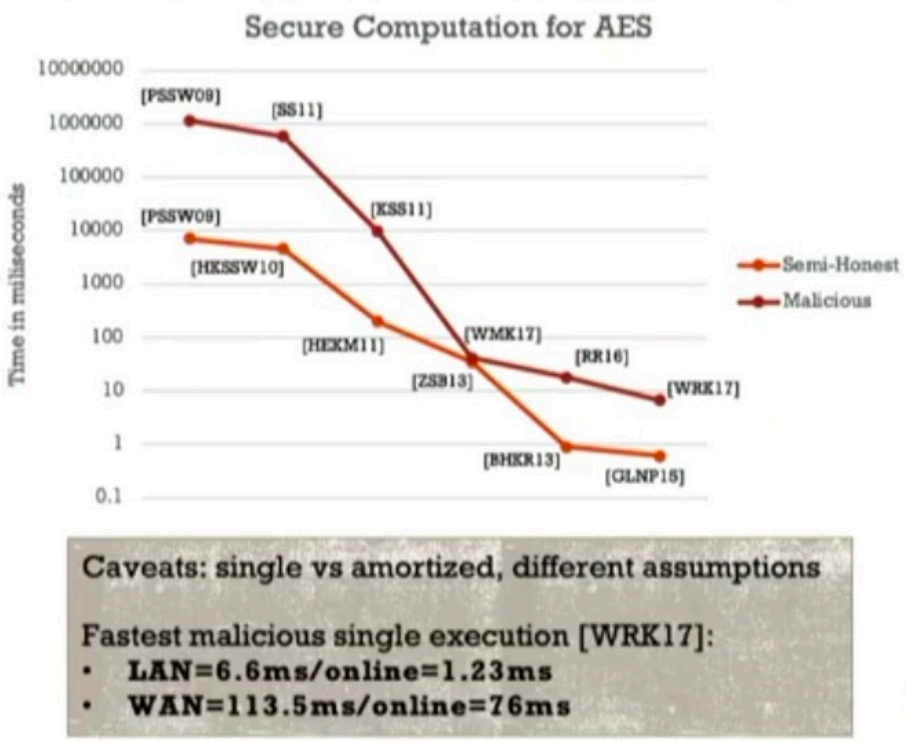
Private Set Intersection



Two Party Computation

Compute F(X, Y) without revealing anything more about X and Y





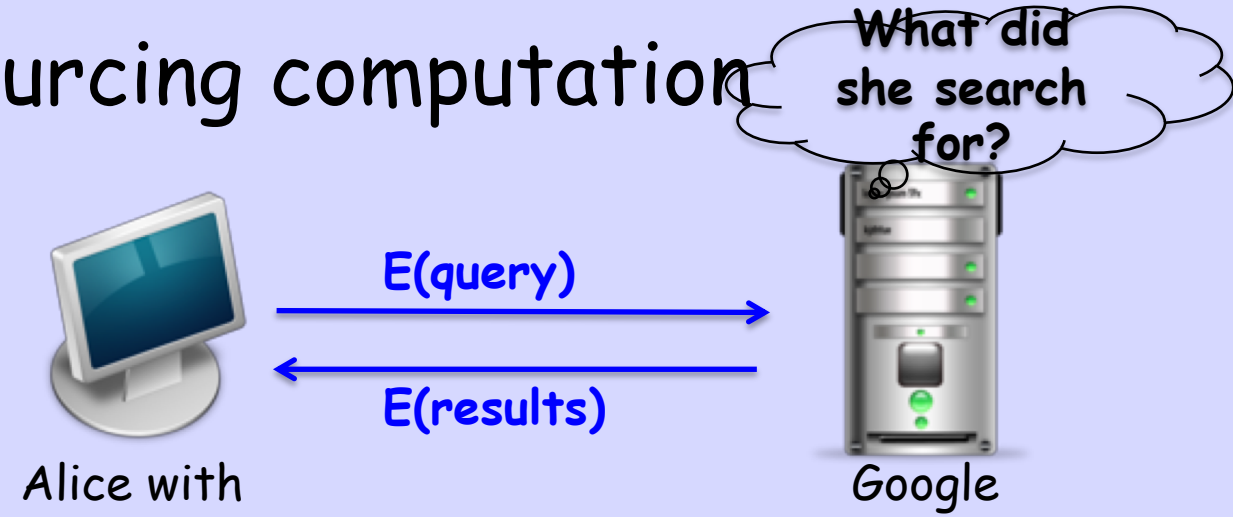
Caveats: single vs amortized, different assumptions

Fastest malicious single execution [WRK17]:

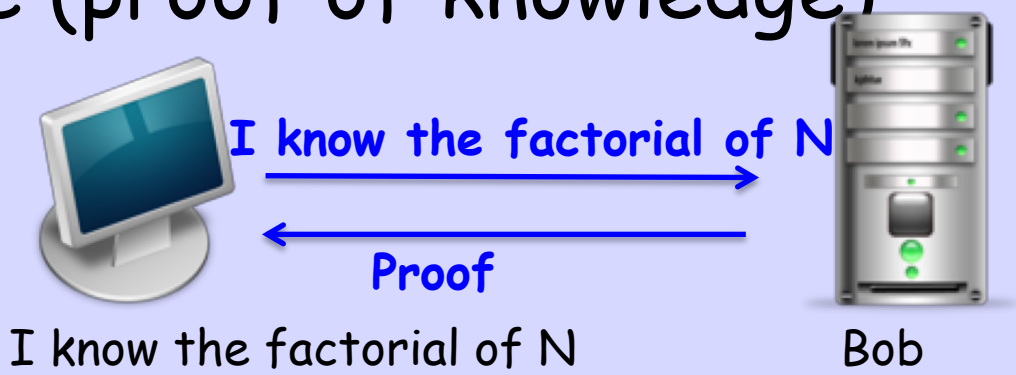
- **LAN=6.6ms/online=1.23ms**
- **WAN=113.5ms/online=76ms**

Magic Tricks

Privately outsourcing computation



Zero knowledge (proof of knowledge)

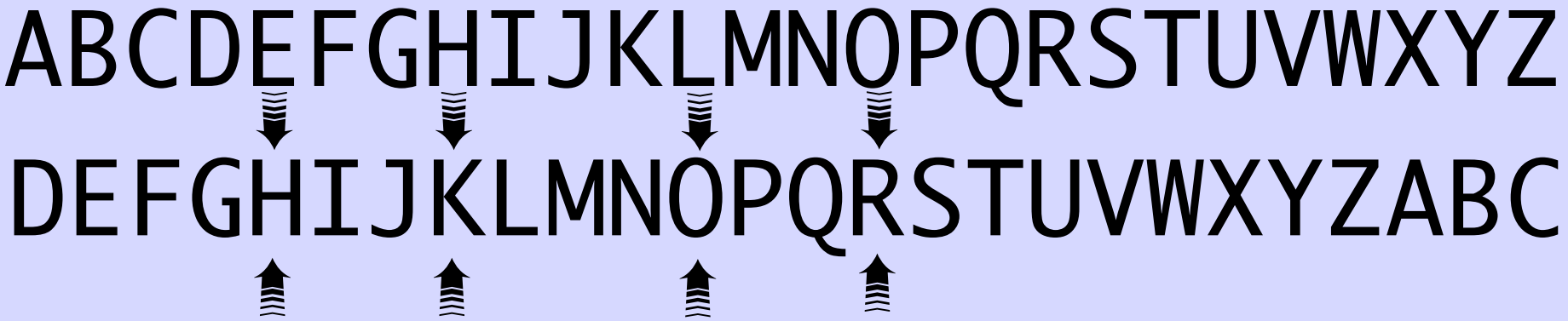


Source: Dan Boneh, Stanford

History: Ciphers

- Substitution ciphers
  - involve replacing an alphabet with another character of the same alphabet set
  - Can be mono-alphabetic (single set for substitution) or poly-alphabetic system (multiple alphabetic sets)
- Example:
  - Caesar cipher, a mono-alphabetic system in which each character is replaced by the third character in succession
  - Vigenere cipher, a poly-alphabetic cipher that uses a 26x26 table of characters

Caesar Cypher

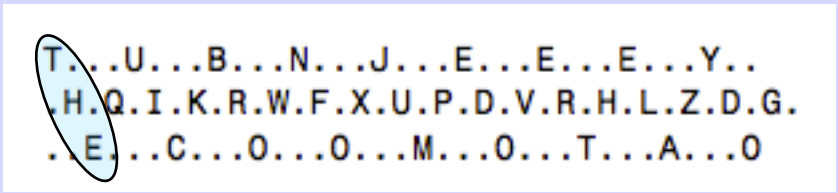


HELLO Plaintext or Cleartext

KHOOR Encrypted Text

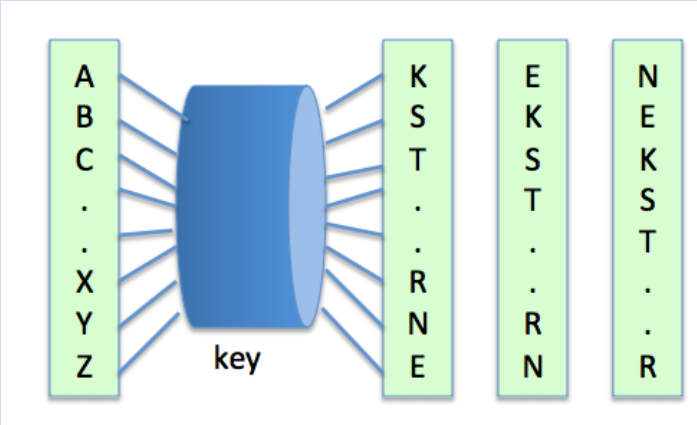
Transposition Cipher

- No letters are replaced, they are just rearranged.
- Rail Fence Cipher – a transposition cipher in which the words are spelled out as if they were a rail fence.

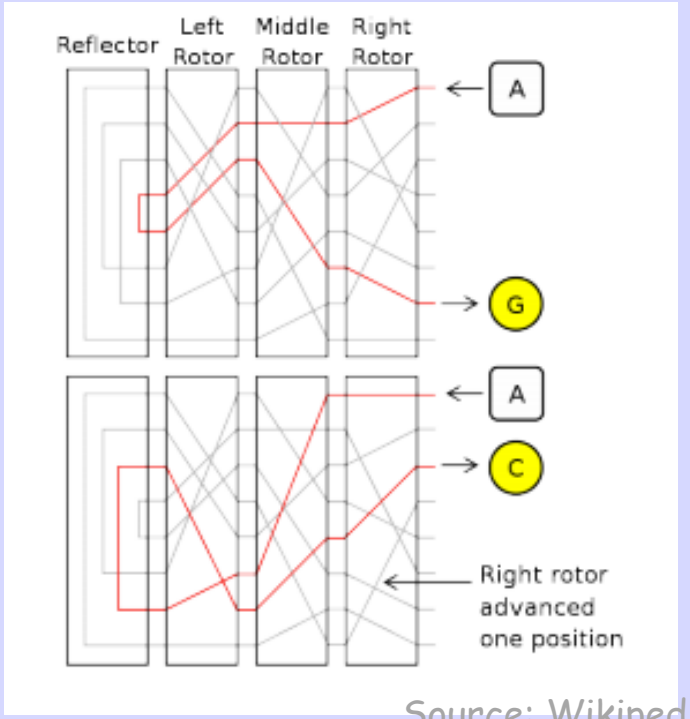


History: Rotor Machines (1870-1943)

Hebern machine – single rotor



Enigma - 3-5 rotors



Source: Wikipedia (image)

Kerckhoff's Law (1883)

- The system must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience.

This is Critical

- In other words, the security of the system must rest entirely on the secrecy of the key.

Modern Crypto Algorithms

- Specify the mathematical transformation that is performed on data to encrypt/decrypt
- Crypto algorithm is NOT secret, keys are
- Analyzed by public community to show that there are no serious weaknesses
- Explicitly designed for encryption

Properties of Good Crypto

- There should be no way short of enumerating all possible keys to find the key from any amount of ciphertext and plaintext, nor any way to produce plaintext from ciphertext without the key.
- Enumerating all possible keys must be infeasible.
- The ciphertext must be indistinguishable from true random values.

Encryption

- Process of transforming plaintext to ciphertext using a cryptographic key
- Used all around us
  - In Application Layer – used in secure email, database sessions, and messaging
  - In session layer – using Secure Socket Layer (SSL) or Transport Layer Security (TLS)
  - In the Network Layer – using protocols such as SSH, VPNs, …

If they outlawed encryption

You could not shop or bank over the Internet

Backdoor or Golden Key

Giving a key to the the NSA who lost their tools which were used in the major ransomware attack of May 2017

Backdoor or Golden Key

Giving a key to the government who allowed the OPM hack, the biggest intelligence failure in human history

Good Algorithms

- Resistant to cryptographic attack
- Support variable and long key lengths and scalability
- Create an avalanche effect
- No export or import restrictions

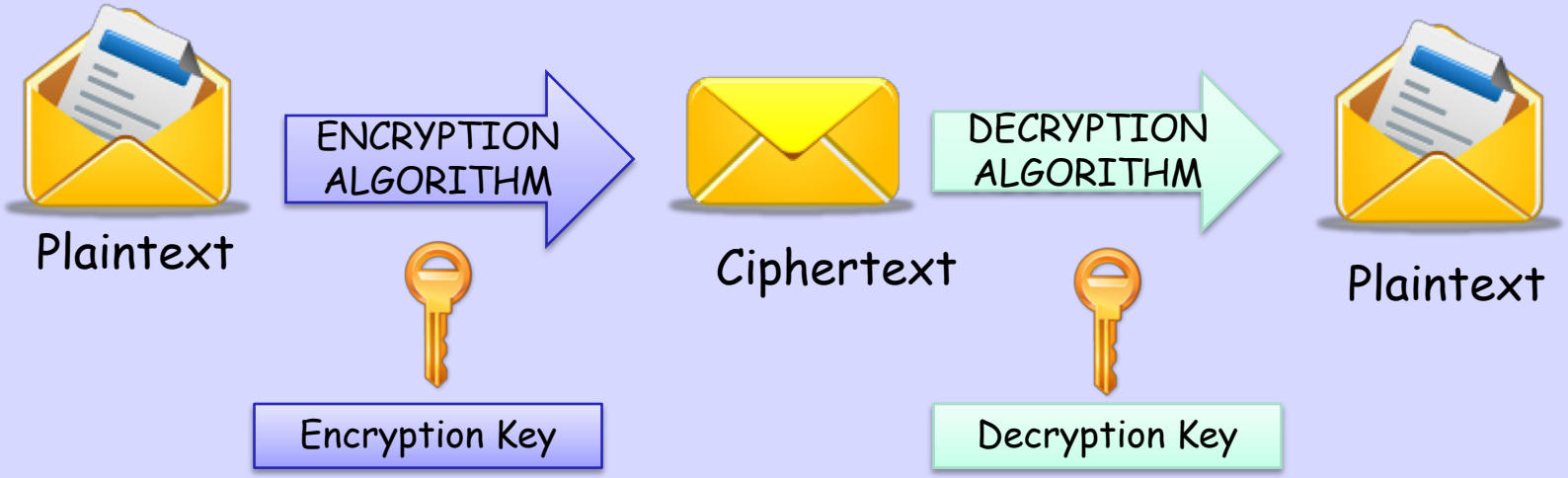
Avalanche Effect

If the cleartext changes the smallest amount, the ciphertext changes a lot

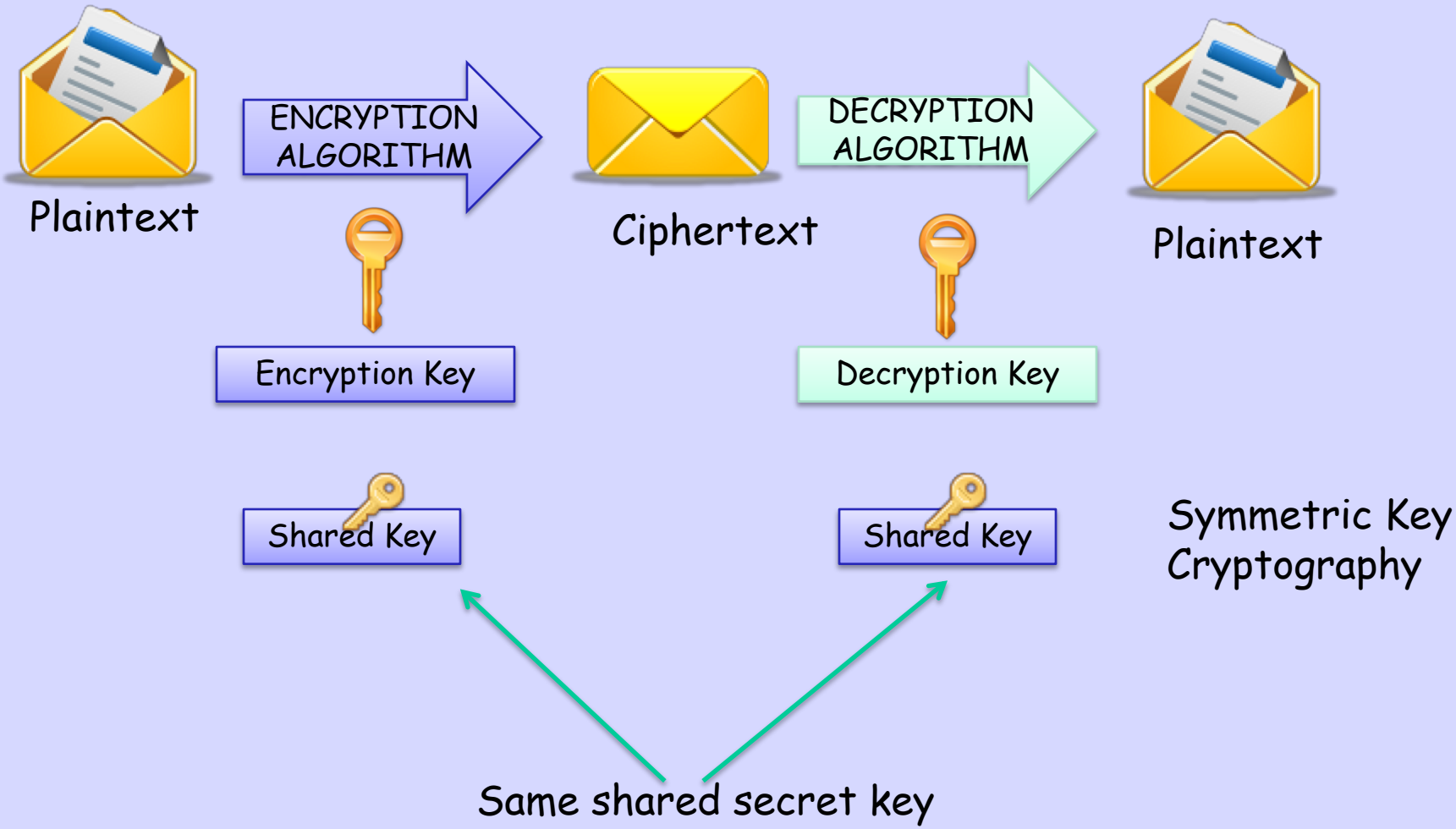
Wikipedia, the source of all knowledge ☺

- when an input is changed slightly (for example, flipping a single bit) the output changes significantly (e.g., half the output bits flip)

Encryption & Decryption



Symmetric Encryption



Symmetric Key Algorithms

- Same key to both encrypt and decrypt
- Also known as a secret-key algorithm
  - The key must be kept a "secret" to maintain security
  - This key is sometimes known as a private key
- Follows the more traditional form of cryptography with key lengths ranging from 40 to 256 bits.
- Examples:
  - DES, 3DES, AES, RC4, RC6, Blowfish

Block Cipher

- Transforms a fixed-length block of plain text into a block of ciphertext
  - operates on a pre-determined block of bits (one byte, one word, 512 bytes, so forth), mixing key data in with the message data in a variety of different ways
- Common block ciphers:
  - DES and 3DES (in ECB and CBC mode)
  - Skipjack
  - Blowfish
  - RSA
  - AES
  - IDEA
  - SAFER

Stream Cipher

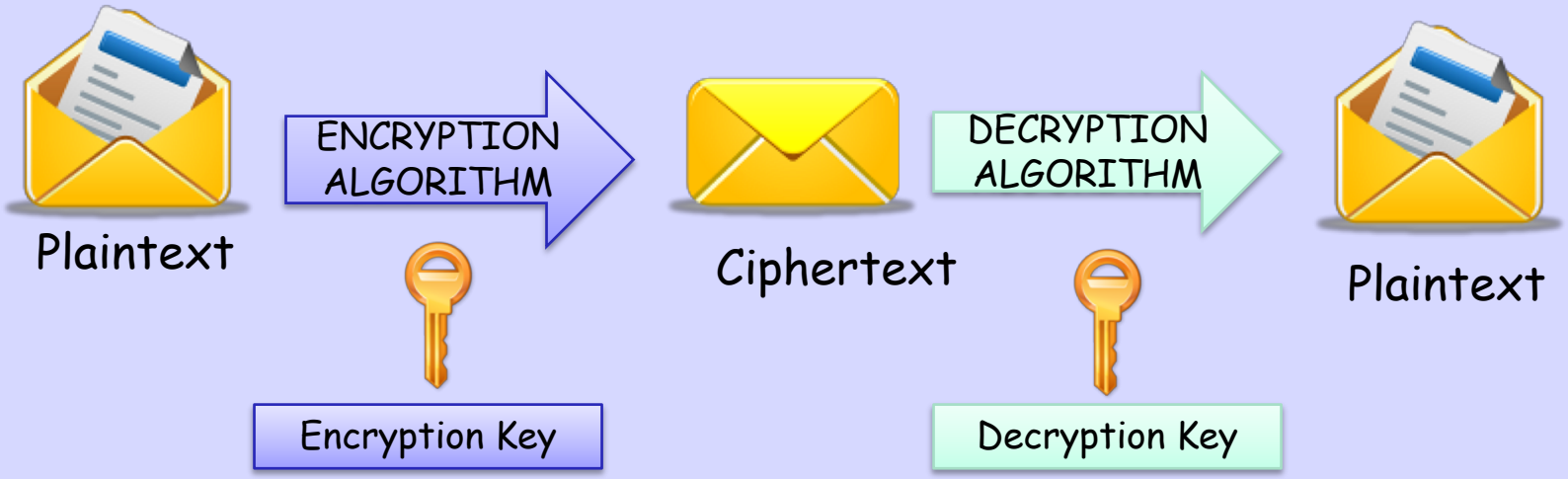
- Smaller units of plaintext than block ciphers.
  - Typically bit-wise
  - Performs some operation (typically an exclusive OR) with one of these key bits and one of the message bits
- Either has a very long key (that eventually repeats) or a reusable key that generates a repeatable but seemingly random string of bits.
- Common stream ciphers:
  - ChaCha
  - DES and 3DES (running OFB or CFB mode)
  - SEAL

Data Encryption Standard (DES)

- Developed by IBM for the US government in 1973-1974, and approved in Nov 1976.
- Based on Horst Feistel's Lucifer cipher
- Block cipher using shared key encryption, 56-bit key length
- An archetypal block cipher with block size of 64 bits
- Now considered as insecure

DES: Illustration

64-bit blocks of input text



56-bit keys +
8 bits parity

DES is Broken

- Developed in the early 1970s
- The 56-bit key length is too short for modern computation
- Broken in 22 hours in 1999, but known to be vulnerable earlier
- So it lasted 25 years, OK for early days
- But messages encrypted with DES in 1990 can be read now trivially!!!

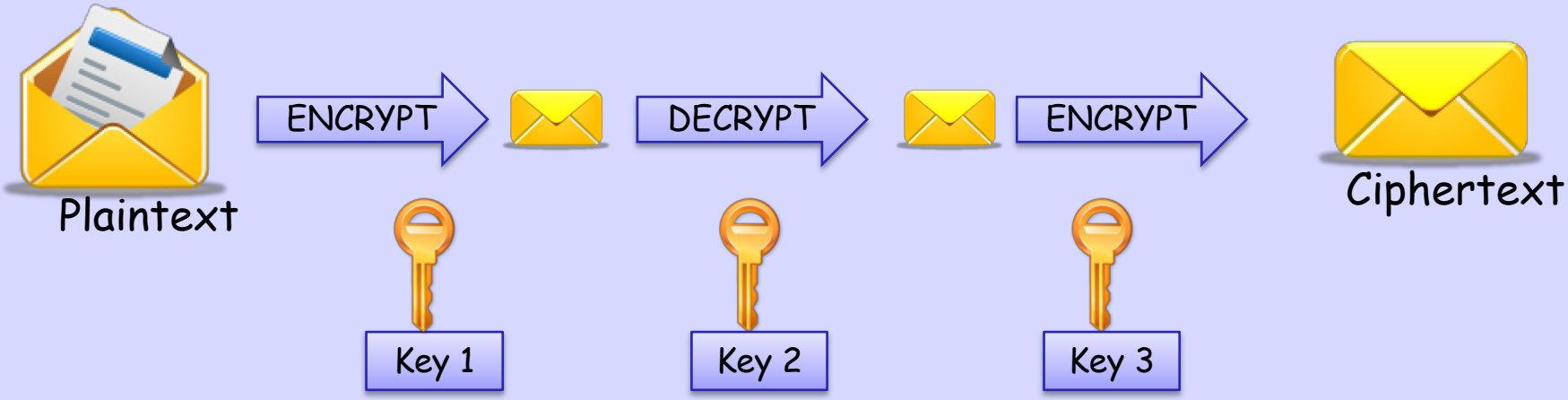
Triple DES-peration

- 3DES (Triple DES) – a block cipher that applies DES three times to each data block
- Uses a key bundle comprising of three DES keys (K1, K2, K3), each with 56 bits excluding parity
- DES encrypts with K1, decrypts with K2, then encrypts with K3

$$C_i = E_{K3}(D_{K2}(E_{K1}(P_i)))$$

- Disadvantage: very slow

3DES: Illustration



Note:

If Key1 = Key2 = Key3, this is similar to DES

Usually, Key1 = Key3

Advanced Encryption Standard (AES)

- NIST/NSA held a contest
- Published in November 2001
- Symmetric block cipher
- Has a fixed block size of 128 bits
- Has a key size of 128, 192, or 256 bits
- Based on Rijndael cipher which was developed by Joan Daemen and Vincent Rijmen
- Better suited for high-throughput, low latency environments

Rivest Cipher

Chosen for speed and variable-key length capabilities

Designed mostly by Ronald Rivest

Each of the algorithms have different uses

RC Algorithm	Description
RC2	Variable key-sized cipher used as a drop in replacement for DES
RC4	Variable key sized stream cipher; Often used in file encryption and secure communications (SSL)
RC5	Variable block size and variable key length; uses 64-bit block size; Fast, replacement for DES
RC6	Block cipher based on RC5, meets AES requirement

Symmetric Key Algs

Symmetric Algorithm	Key Size
DES	56-bit keys
Triple DES (3DES)	112-bit and 168-bit keys
AES	128, 192, and 256-bit keys
IDEA	128-bit keys
RC2	40 and 64-bit keys
RC4	1 to 256-bit keys
RC5	0 to 2040-bit keys
RC6	128, 192, and 256-bit keys
Blowfish	32 to 448-bit keys

Note:
Longer keys are more difficult to crack, but more computationally expensive.

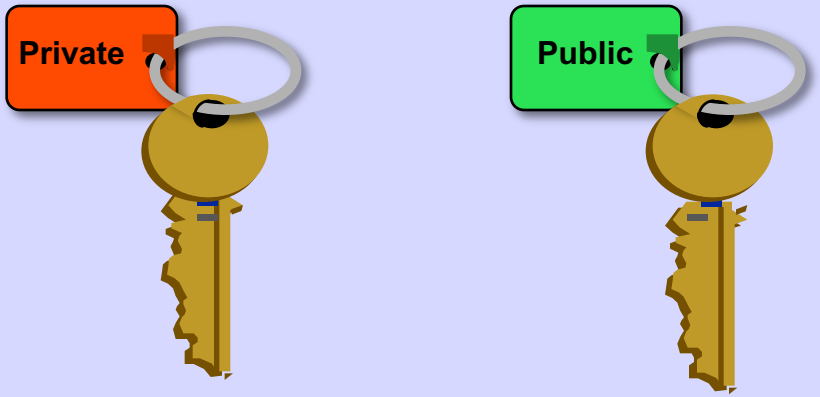
Asymmetric Key Algs

- Also called **public-key cryptography**
  - Keep private key to yourself and protected
  - Send/share the public key to anyone
- Separate keys for encryption and decryption (public and private key pairs)
- Examples:
  - RSA, DSA, Diffie-Hellman, ElGamal, PKCS

Public Key Encryption

Uses public/private keys

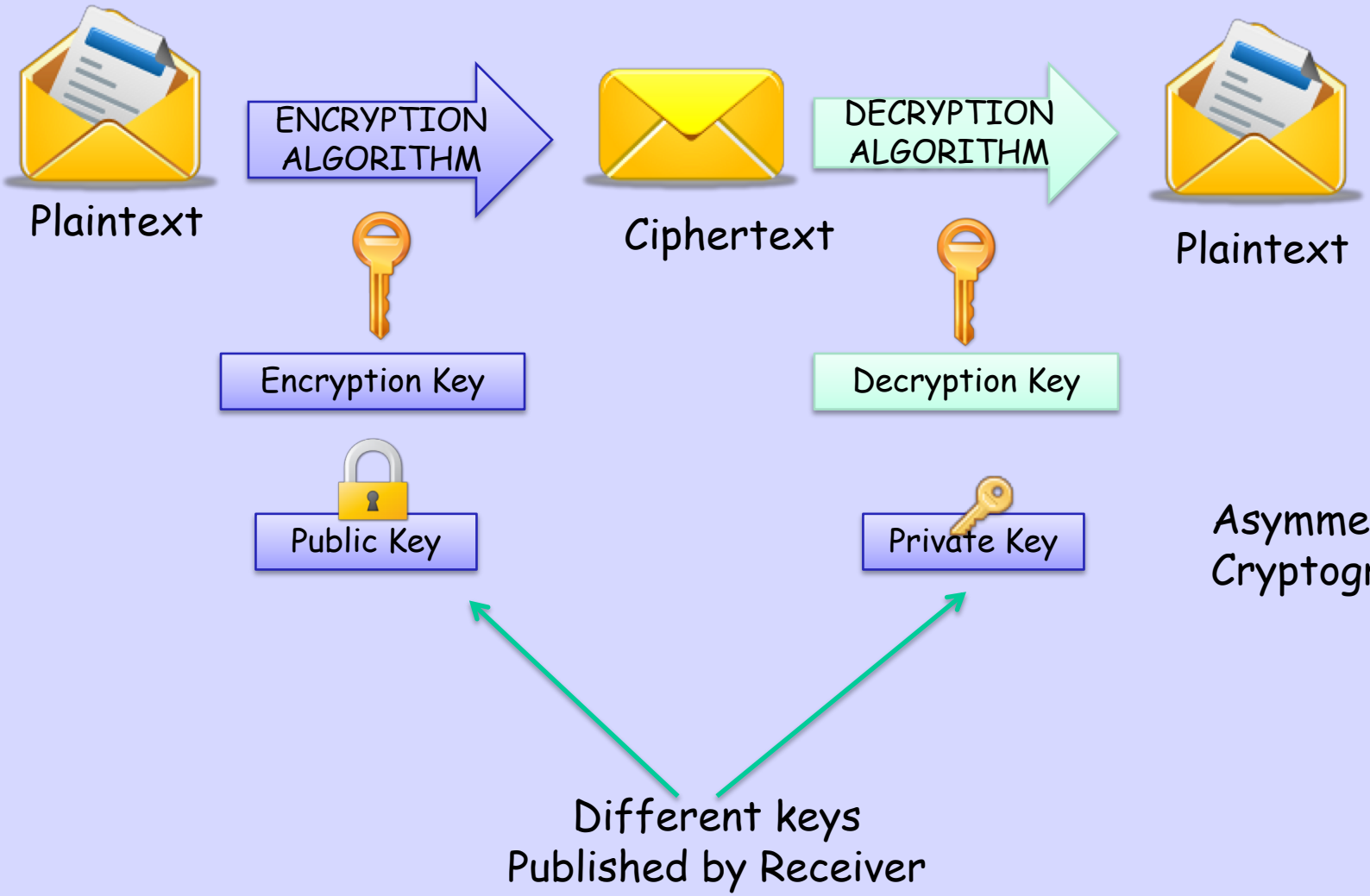
- One key is mathematical inverse of the other
- Private key is only known by owner of the pair
- Public keys are stored in public servers



Computing Key pair is computationally expensive!!

Common Algorithms: RSA, El Gamal, DSS, ECC

Asymmetric Encryption



Asymmetric Key Algorithms

- RSA – the first and still most common implementation (Rivest, Shamir, Adelman)
- DSA – specified in NIST's Digital Signature Standard (DSS), provides digital signature capability for authentication of messages
- Diffie-Hellman – used for secret key exchange only, and not for authentication or digital signature
- ElGamal – similar to Diffie-Hellman and used for key exchange
- PKCS – an overly complex set of interoperable standards and guidelines

Symmetric vs. Asymmetric Key

Symmetric	Asymmetric
Generally fast Same key for both encryption and decryption	Can be 1000 times slower Uses two different keys (public and private) Decryption key cannot be calculated from the encryption key Key lengths: 512 to 4096 bits Used in low-volume

Mostly used in combination

Hash Functions

- Produce a condensed representation of a text (hashing)
- The fixed-length output called the hash or message digest
- Hash function takes an input message of arbitrary length and outputs fixed-length code.
  - Given x, we can compute the value f(x).
  - Given f(x), it is extremely hard to get the value of x.
- If it uniquely represents the data, it is Collision-free
- Uses:
  - Verifying file integrity - if the hash changes, it means the data is either compromised or altered in transit.
  - Digitally signing documents
  - Hashing passwords

Hash Functions

- Message Digest (MD) Algorithm
  - Outputs a 128-bit fingerprint of an arbitrary-length input
  - MD4 is obsolete, MD5 is widely-used but deprecated
- Secure Hash Algorithm (SHA)
  - SHA-1 produces a 160-bit message digest similar to MD5
  - Widely-used on security applications (TLS, SSL, PGP, SSH, S/MIME, IPsec)
  - SHA-256, SHA-384, SHA-512 are also commonly used, which can produce hash values that are 256, 384, and 512-bits respectively
- RIPEMD
  - Derived from MD4, but performs like SHA
  - RIPEMD-160 is the most popular version

Digital Signature

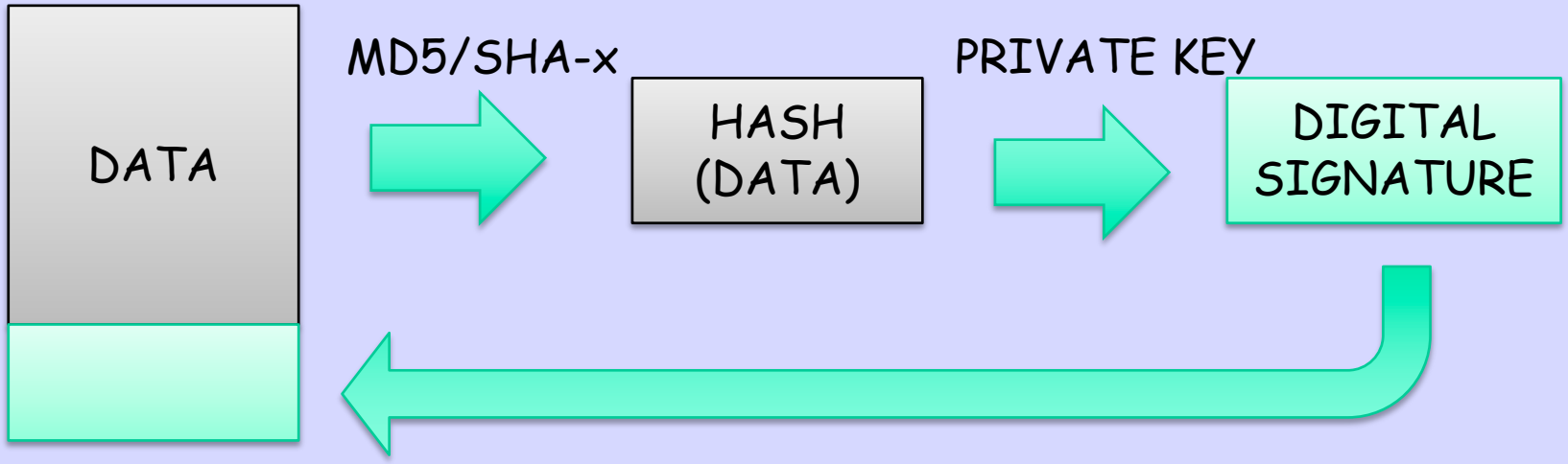
- A digital signature is a message appended to a packet
- The sender encrypts message with their private key instead of encrypting with intended receiver's public key
- The receiver of the packet uses the sender's public key to verify the signature.
- Used to prove the identity of the sender and the integrity of the message

Digital Signature

- Two common public-key digital signature techniques:
  - RSA (Rivest, Shamir, Adelman)
  - DSS (Digital Signature Standard) (deprecated)
- Used in a lot of things:
  - Email, software distribution, electronic funds transfer, etc
- A common way to implement is to use a hashing algorithm to get the message digest of the data, then use an algorithm to sign the message

Digital Signing Process

1. Hash the data using one of the supported hashing algorithms (MD5, SHA-1, SHA-256) (first two deprecated)
2. Encrypt the hashed data using the sender's private key
3. Append the signature (and a copy of the sender's public key) to the end of the data that was signed)



Signature Verification

1. Hash the original data using the same hashing algorithm
2. Decrypt the digital signature using the sender's public key. All digital signatures contain a copy of the signer's public key
3. Compare the results of the hashing and the decryption. If the values match then the signature is verified. If the values do not match, then the data or signature was probably modified.