INF3510 Information Security
University of Oslo
Spring 2011

Lecture 5

Cryptography

Audun Jøsang

Cryptography

- Cryptography is the science and study of secret writing.
- Cryptanalysis is the science and study of methods of breaking ciphers.
- Cryptology: cryptography and cryptanalysis.
- Today: Cryptography is the study of mathematical techniques related to aspects of information security, such as confidentiality, data integrity, entity authentication, and data origin authentication. [Handbook of Applied Cryptography]

When is cryptography used?

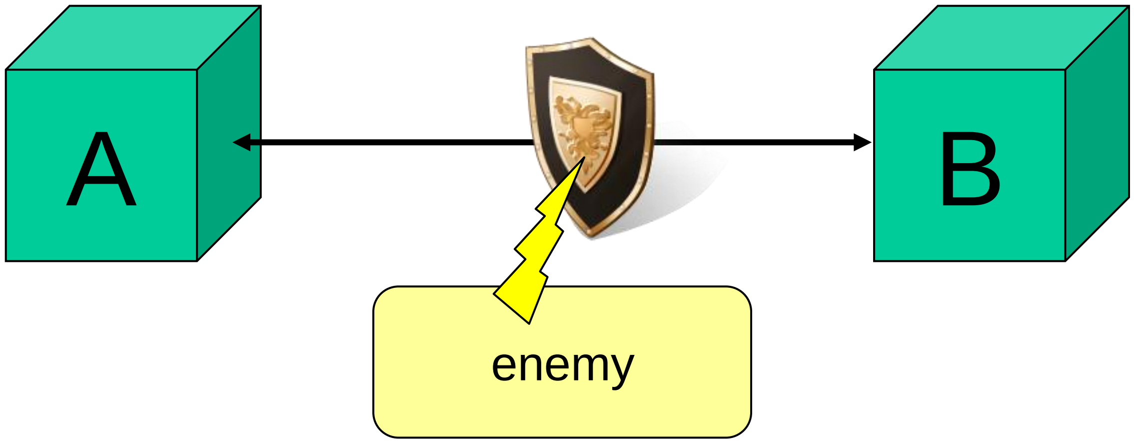
- If you require
  - **Confidentiality:**
    - so that your data is not made available to anyone who shouldn't have access.
    - That is, protection against snoops or eavesdroppers
  - **Integrity:**
    - So you know that the message content is correct, and has not been altered, either deliberately or accidentally
  - **Authentication:**
    - So you can be sure that the message is from the place or sender it claims to be from
- Cryptography can provide these security services.

When is cryptography used?

- Some example situations:
  - **Historically**, the military and spy agencies were the main users of cryptology
    - Situation: transmitting messages over insecure channels
  - **Now**, it is used in many other areas, especially in electronic information processing and communications technologies:
    - Banking: your financial transactions, such as EFTPOS
    - Communications: your mobile phone conversations
    - Info stored in databases: hospitals, universities, etc.
- Cryptography can be used to protect information in storage or during transmission

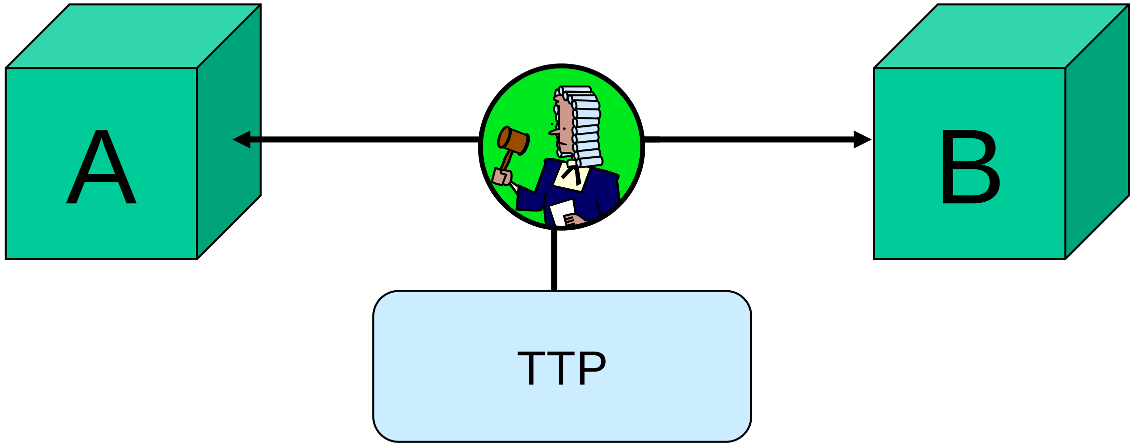
Traditional paradigm

- *A* and *B* communicate over an insecure channel.
- *A* and *B* trust each other.
- Intruder can read, delete, and insert messages.
- With cryptography, *A* and *B* construct a secure logical channel over an insecure network.



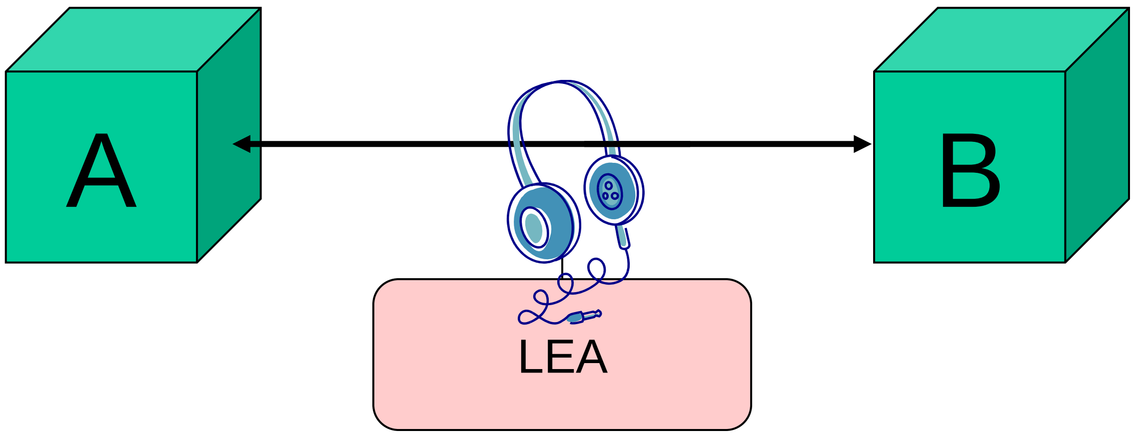
Trust paradigm

- Electronic commerce: *A* and *B* have a business relationship, but do not fully “trust” each other.
- We want protection against fraud and deception as much as possible.
- Trusted Third Parties help settle disputes.



Law enforcement paradigm

- In many countries laws regulate how a law enforcement agency (LEA) can intercept traffic.
- Key recovery makes cryptographic keys available to their owner.
- Key escrow makes keys available to a LEA.

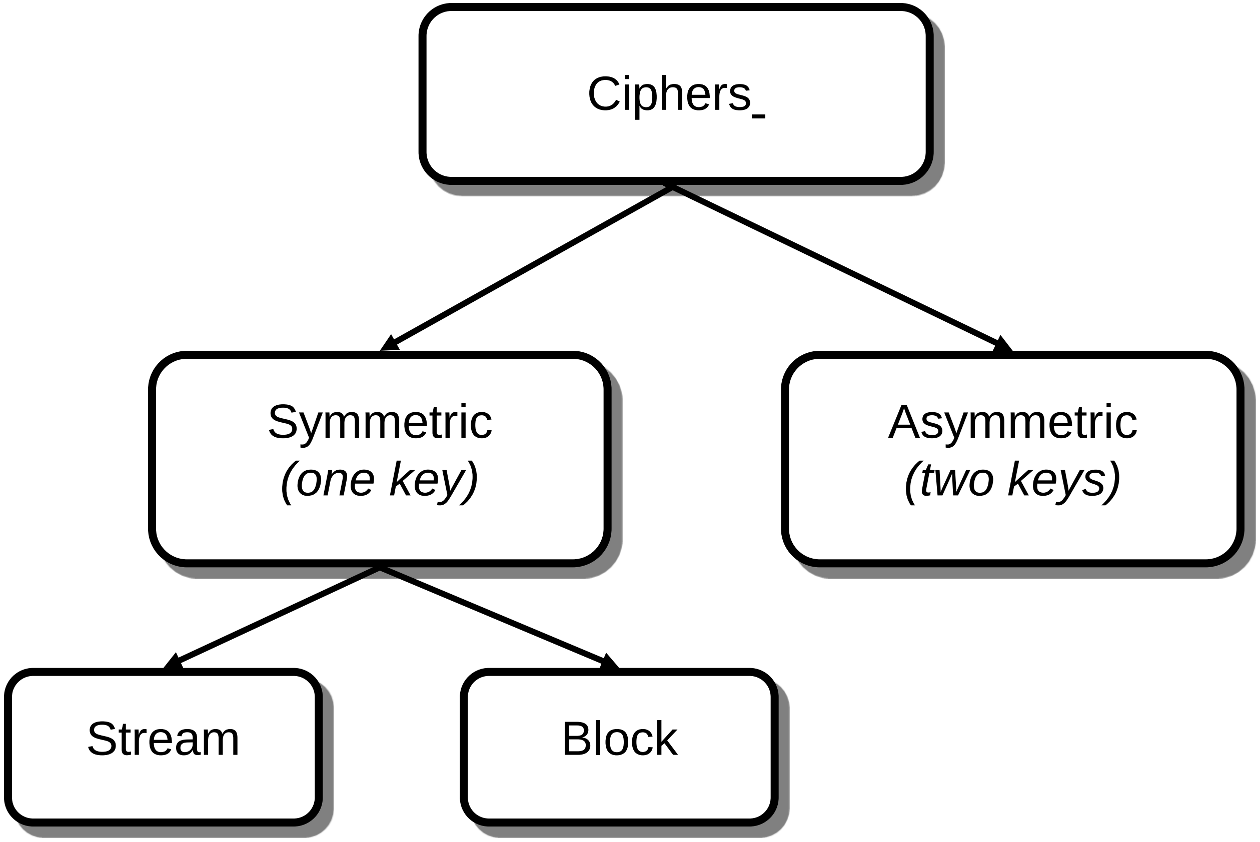


Alternatives to cryptography

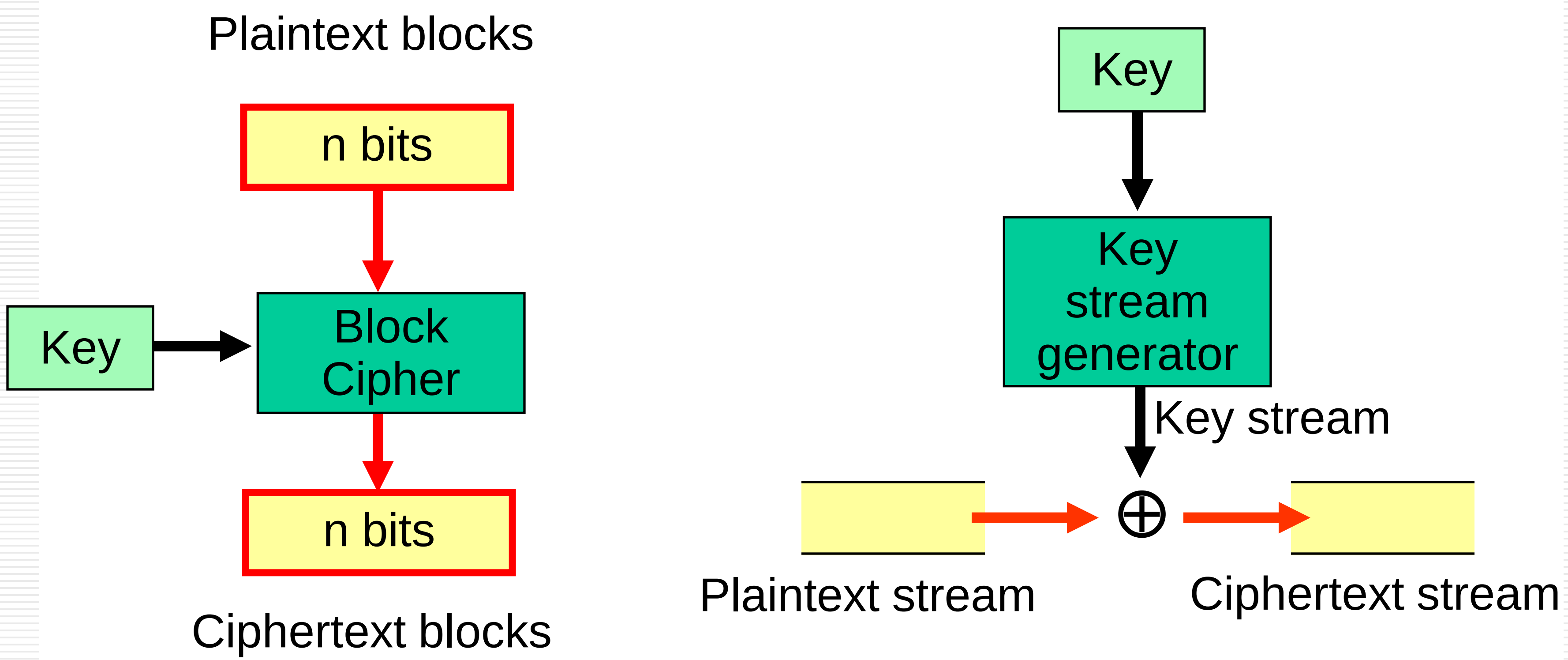
- **Steganography:** – used to hide the existence of a message
  – Hide the information within a document or image, so that the presence of the message is not detected
- **Steganographic techniques include**
  – Using invisible ink (try writing in lemon juice)
  – Microdots
  – Character arrangement and selection
  – Hiding information, e.g. in graphics and sound files
- **Steganographic techniques do not use a secret key**

Encryption

Taxonomy of modern ciphers



Block Cipher vs. Stream Cipher



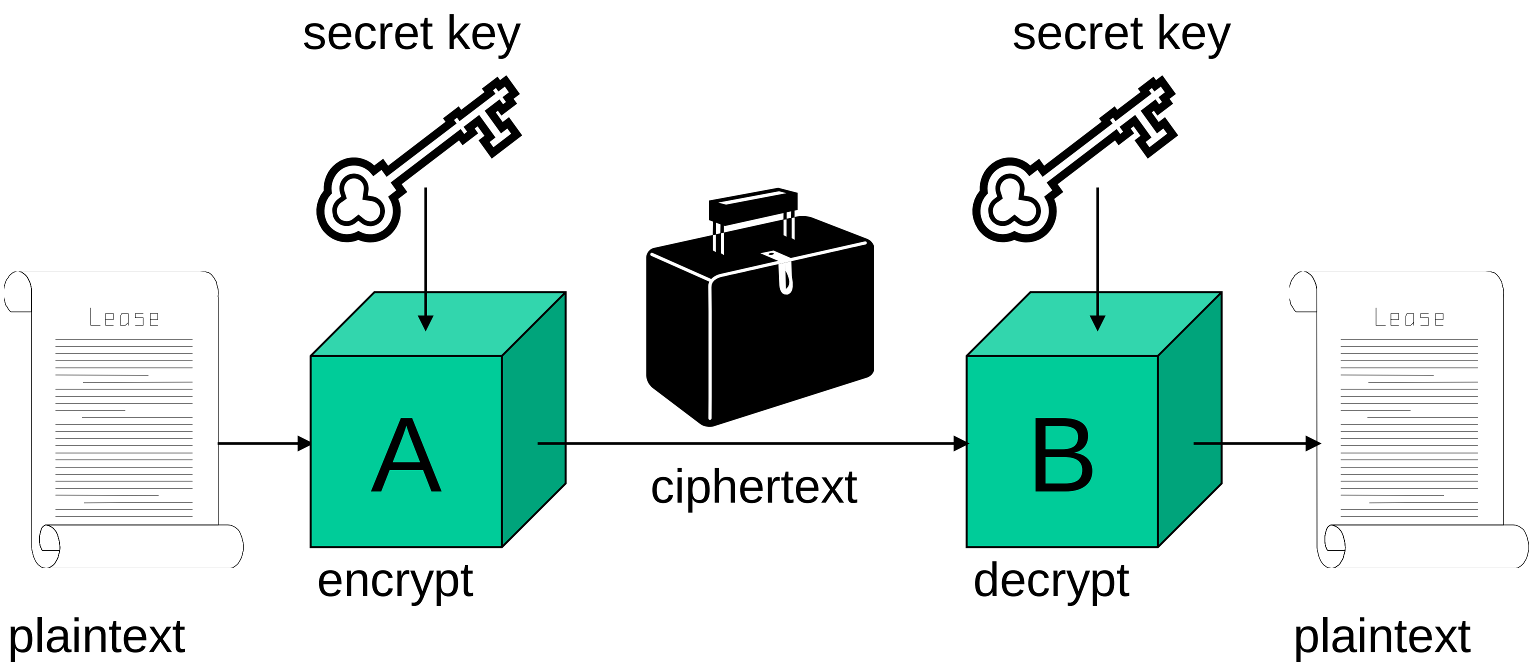
Block cipher

Stream cipher

Terminology

- Encryption: plaintext (clear text) M is converted into a ciphertext C under the control of a key k.
  - We write $C = E(M, k)$.
- Decryption with key k recovers the plaintext M from the ciphertext C.
  - We write $M = D(C, k)$.
- Symmetric ciphers: the secret key is used for both encryption and decryption.
- Asymmetric ciphers: Pair of private and public keys where it is computationally infeasible to derive the private decryption key from the corresponding public encryption key.

Symmetric Key Encryption

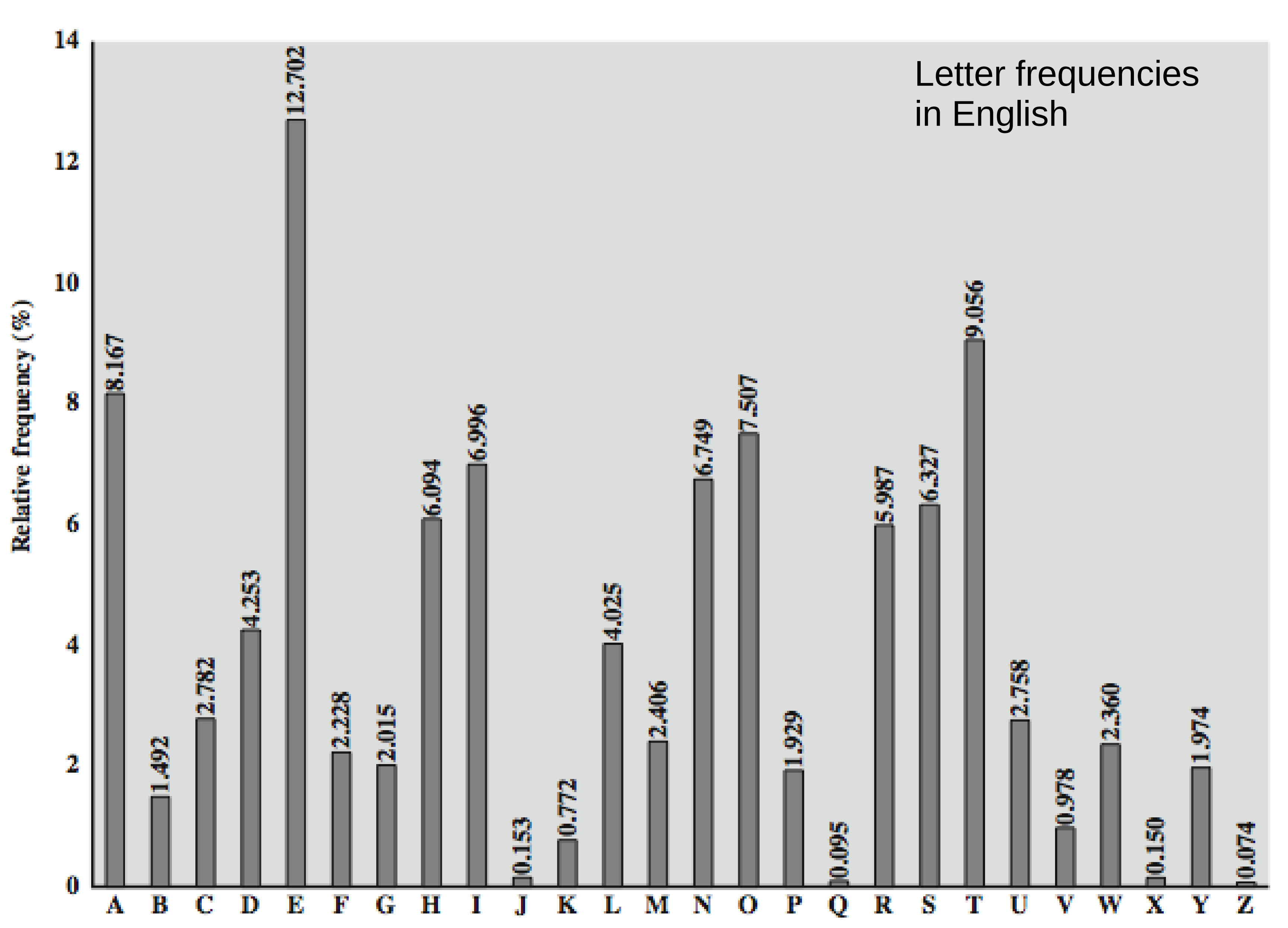


Classical Ciphers

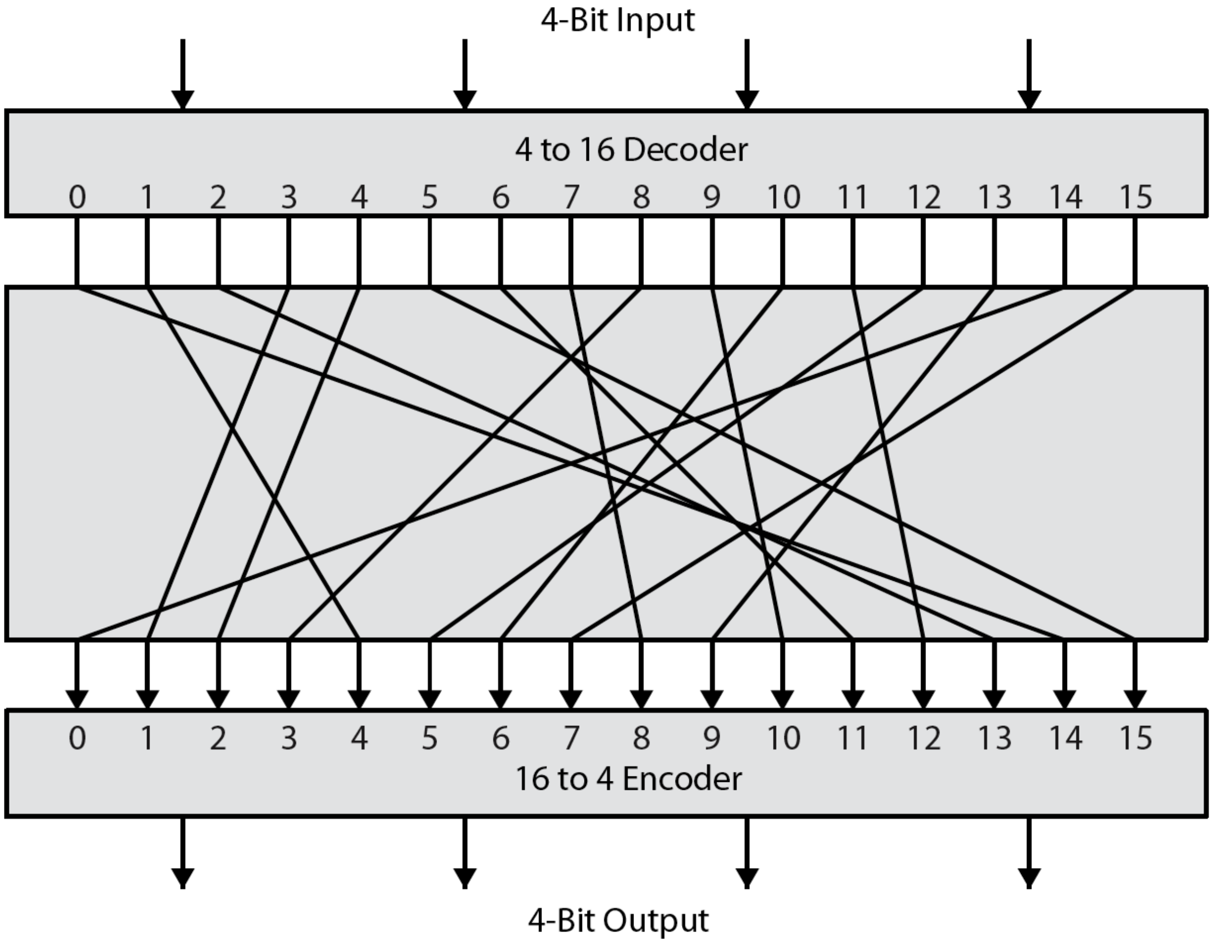
- Caesar cipher
  - Shift alphabet a fixed number of places
- Vigenère cipher
  - Multiple Caesar ciphers
- Mono/poly-alphabetic ciphers
  - Substitute alphabet(s)
- Transposition ciphers
  - Reorder characters within a block
- Product ciphers
  - Serial combination of substitution and transposition
- Vernam ciphere / one-time pad
  - Modular character addition (key size = message size)

Should remove statistical regularities as much as possible

Letter Frequencies→ statistical attacks



Ideal 4-bit Block Cipher



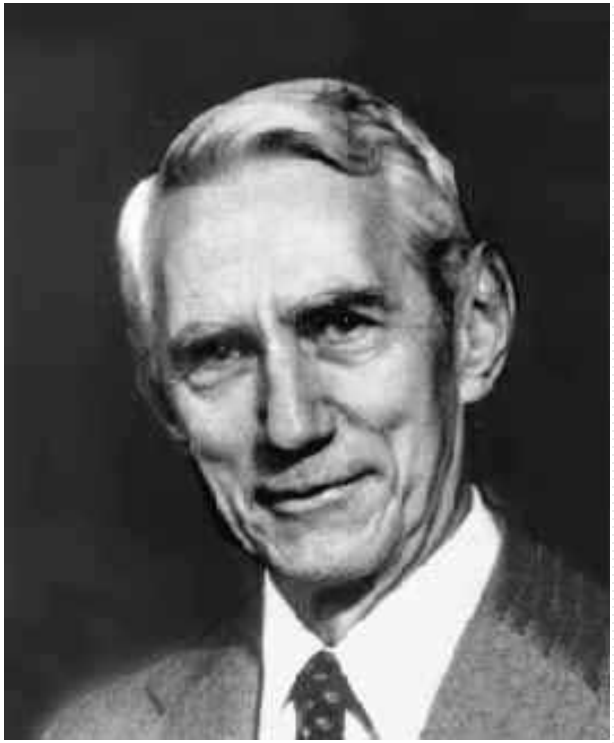
- Diagram shows a single substitution function from the message space to the ciphertext space
- 16! different substitution functions possible with 4-bit blocks

Modern ciphers have a 128-bit block, but only have e.g. 128 or 256 bit keys (2^{128} or 2^{256} substitutions), not $(2^{128})!$ substitutions

Claude Shannon (1916 – 2001)

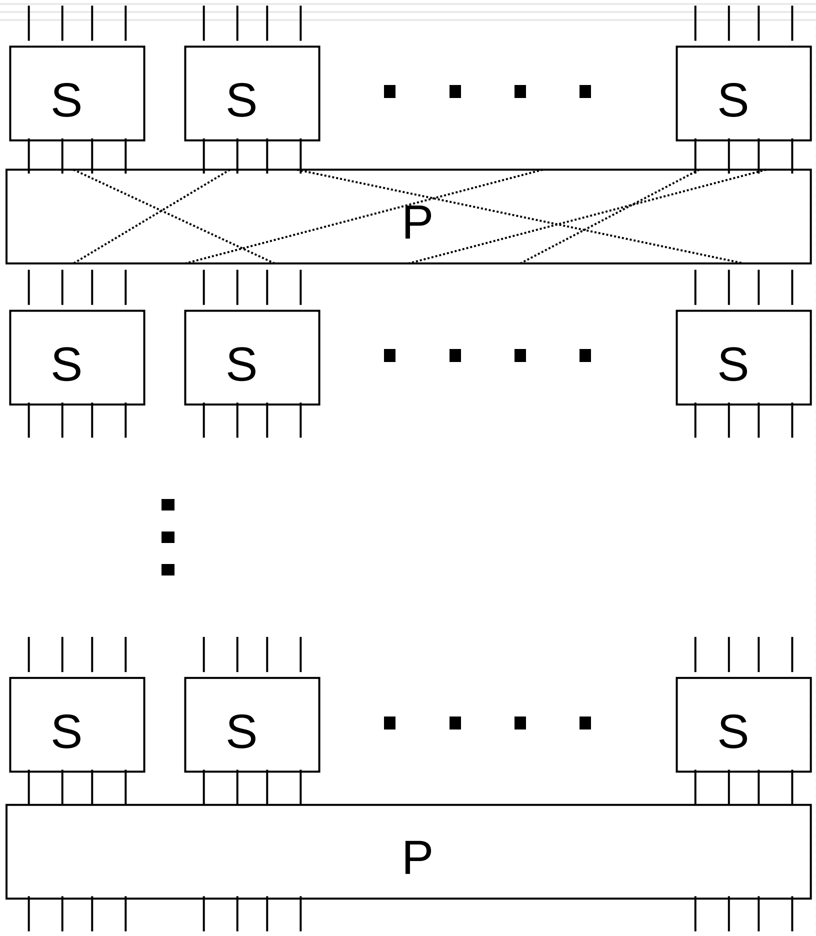
The Father of Information Theory – MIT / Bell Labs

- Information Theory
  - Defined the „binary digit“ (bit) as information unit
  - Definition of „entropy“ as a measure of information
- Cryptography
  - Model of a secrecy system
  - Definition of perfect secrecy
  - Defined cryptographic „confusion“ and „diffusion“
  - Designed S-P networks (Substitution & Permutation)



Shannon's S-P Network

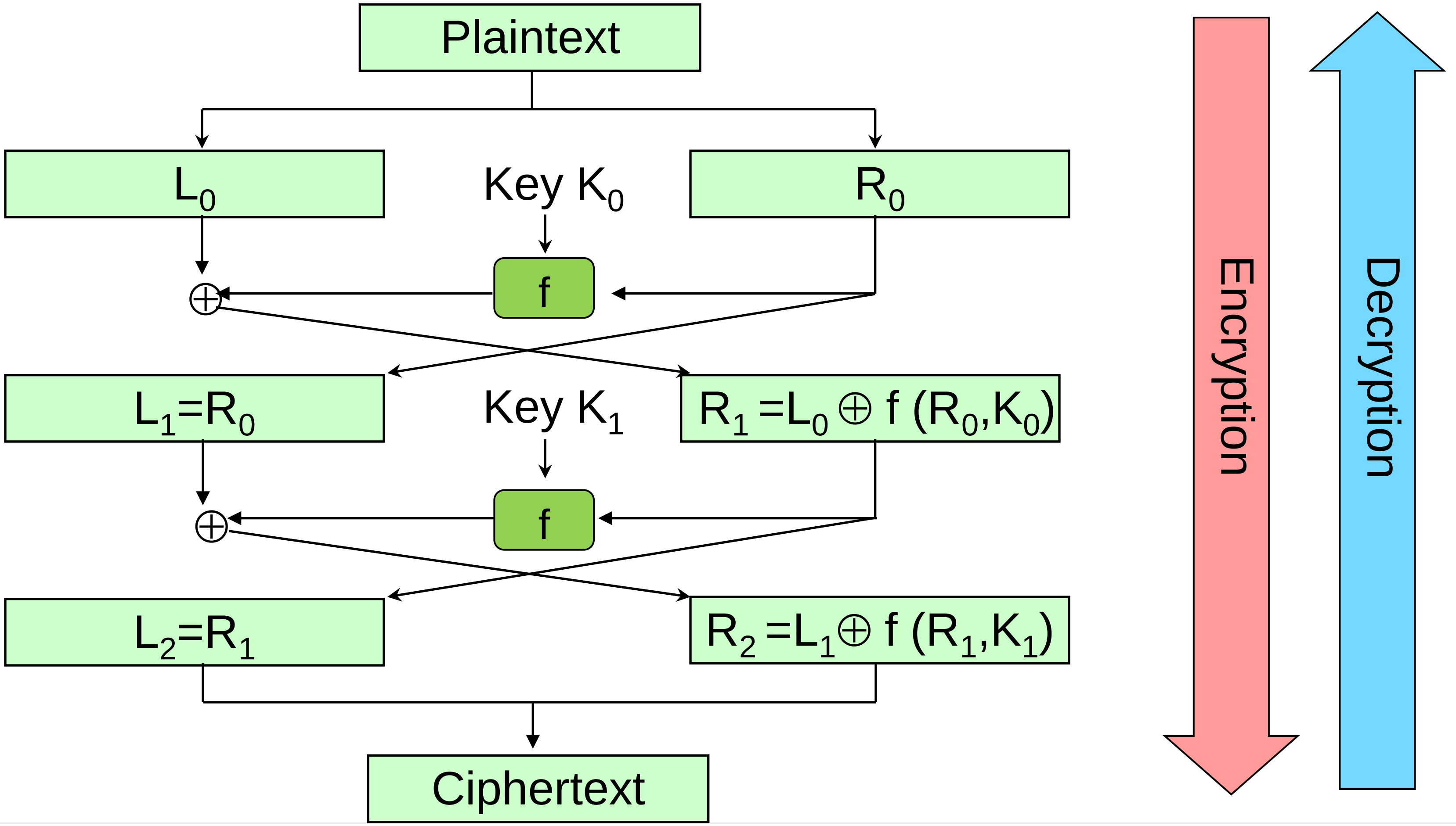
- "S-P Networks" (1949)
  - Sequence of many substitutions & permutations
  - small, carefully designed substitution boxes ("confusion")
  - their output mixed by a permutation box ("diffusion")
  - iterated a certain number of times
  - Functions **must be invertible**



Horst Feistel's (1915 – 1990) and his revolutionary cipher design

- The **feistel cipher** is a general and elegant architecture for designing product ciphers
- Split input block in two halves
  - Perform S-P transformation on one half
  - XOR output with other half
  - Swop Halves
  - Repeat for multiple rounds
- The S-P transformation does **not** have to be invertible

2-round Feistel Network



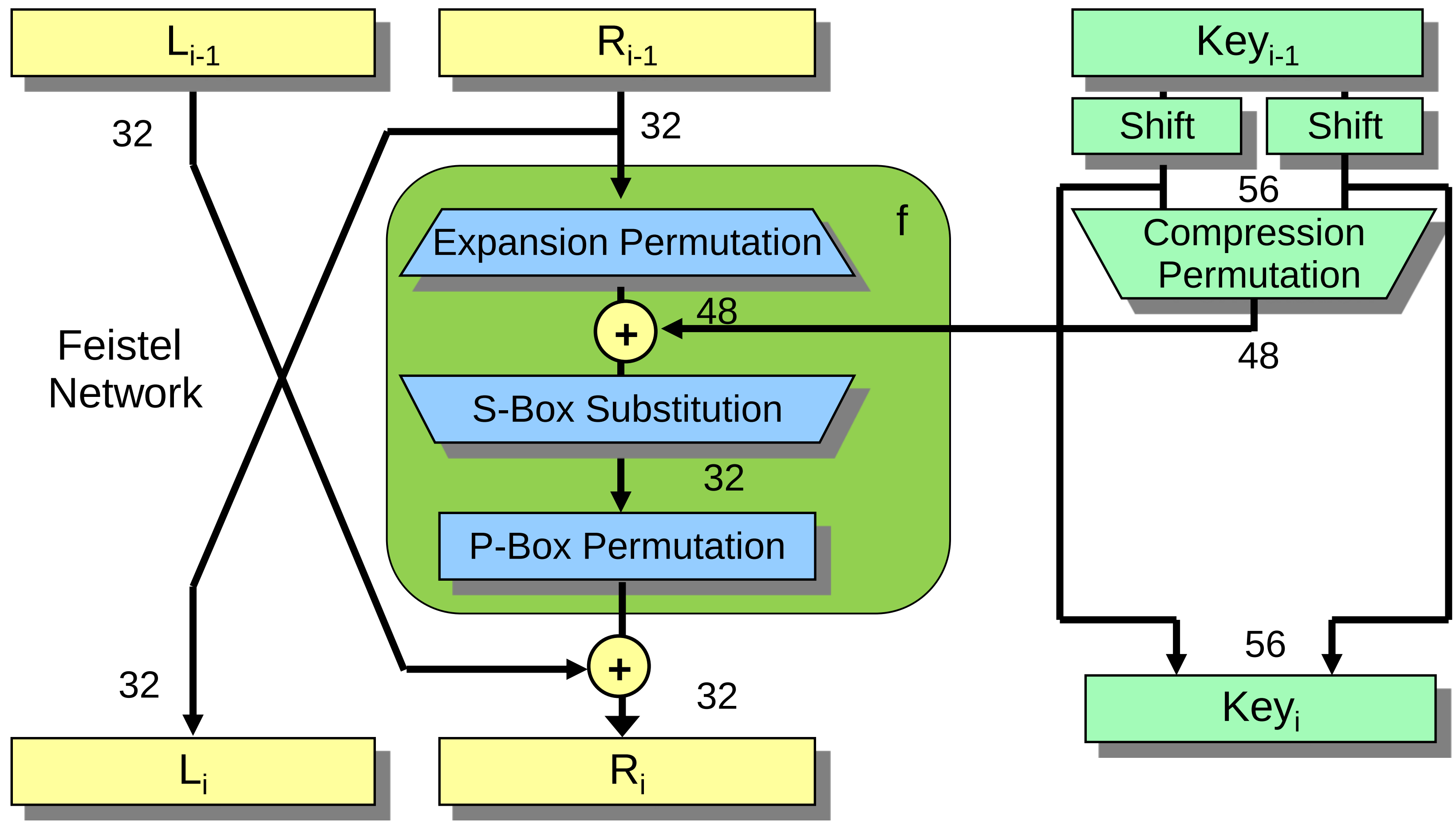
Data Encryption Standard - History

- In May 1973, and again in Aug 1974 the NBS (now NIST) called for possible encryption algorithms for use in unclassified government applications.
- Response was mostly disappointing
- IBM submitted their Feistel-network based Lucifer cipher as a candidate for DES. After some redesign (a reduction to a 56-bit key and 64-bit block, it became the Data Encryption Standard in 1977.

Data Encryption Standard

- Published in 1977 by the US National Bureau of Standards for use in unclassified government applications with a 15 year life time.
- 16 round Feistel cipher with 64-bit data blocks, 56-bit keys.
- 56-bit keys were controversial in 1977; today, exhaustive search on 56-bit keys is very feasible.
- Controversial because of classified design criteria, however no loop hole has yet emerged.
- DES designed to resist differential cryptanalysis.

One Round of DES



Advanced Encryption Standard

- Public competition to replace DES: because 56-bit keys and 64-bit data blocks no longer adequate.
- Rijndael nominated as the new Advanced Encryption Standard (AES) in 2001 [FIPS-197].
- Rijndael (pronounce as “Rhine-doll”) designed by Vincent Rijmen and Joan Daemen.
- Versions for 128-bit, 196-bit, and 256-bit data and key blocks (all combinations of block length and key length are possible).
- Rijndael is not a Feistel cipher.

Advanced Encryption Standard (AES) Contest (1997-2001)

January 1997 — **Call for cipher proposals**

Key sizes: 128, 192 or 256 bit, **block size**: 128 bits

June 1998 — **Round 1**

15 Candidates
from USA, Canada, Belgium, France, Germany, Norway, UK, Israel, Korea, Japan, Australia, Costa Rica

Security, Software efficiency, Flexibility

August 1999 — **Round 2**

5 final candidates

Mars, RC6, Rijndael, Serpent, Twofish

Security, Hardware efficiency

October 2000

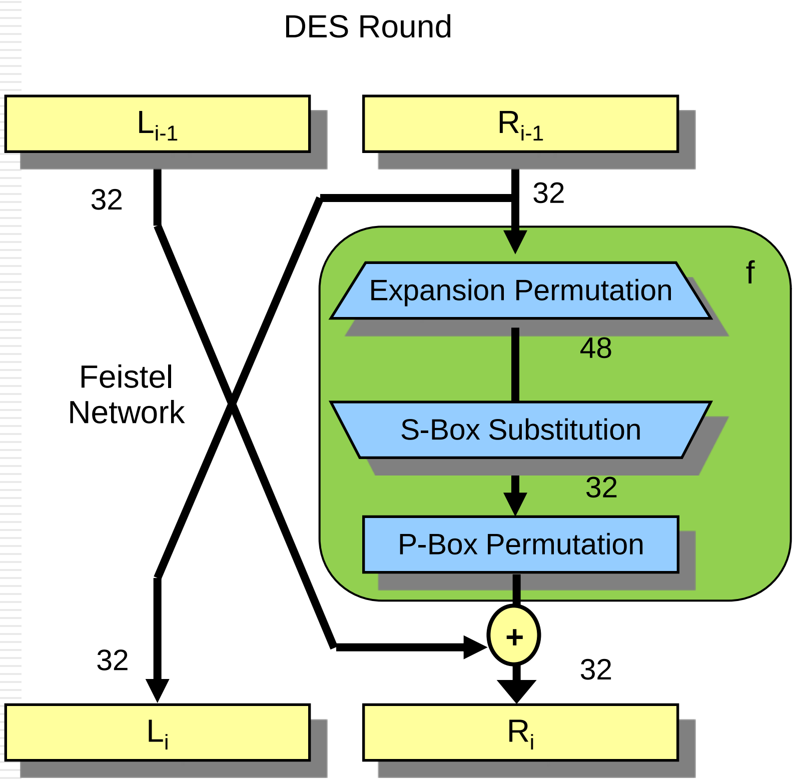
1 winner: Rijndael Belgium

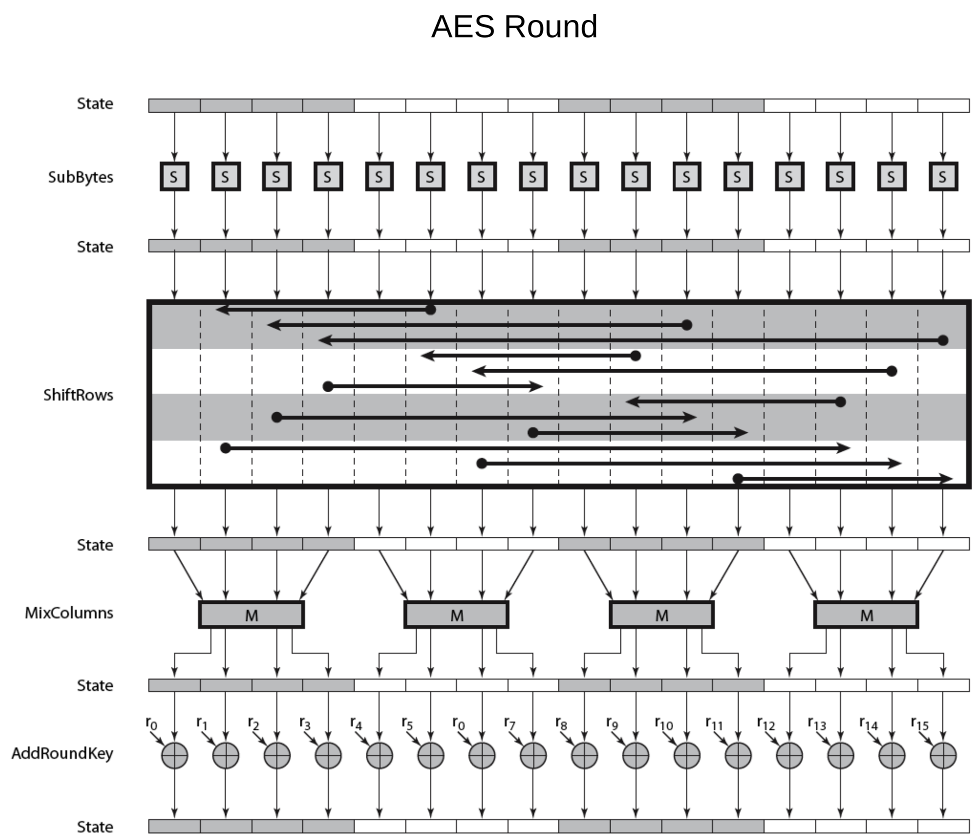
November 2001 — **AES FIPS PUB 197 standard**

Rijndael, the selected AES cipher

Designed by Vincent Rijmen and Joan Daemen from Belgium

Comparison DES – AES Round





Using encryption for real

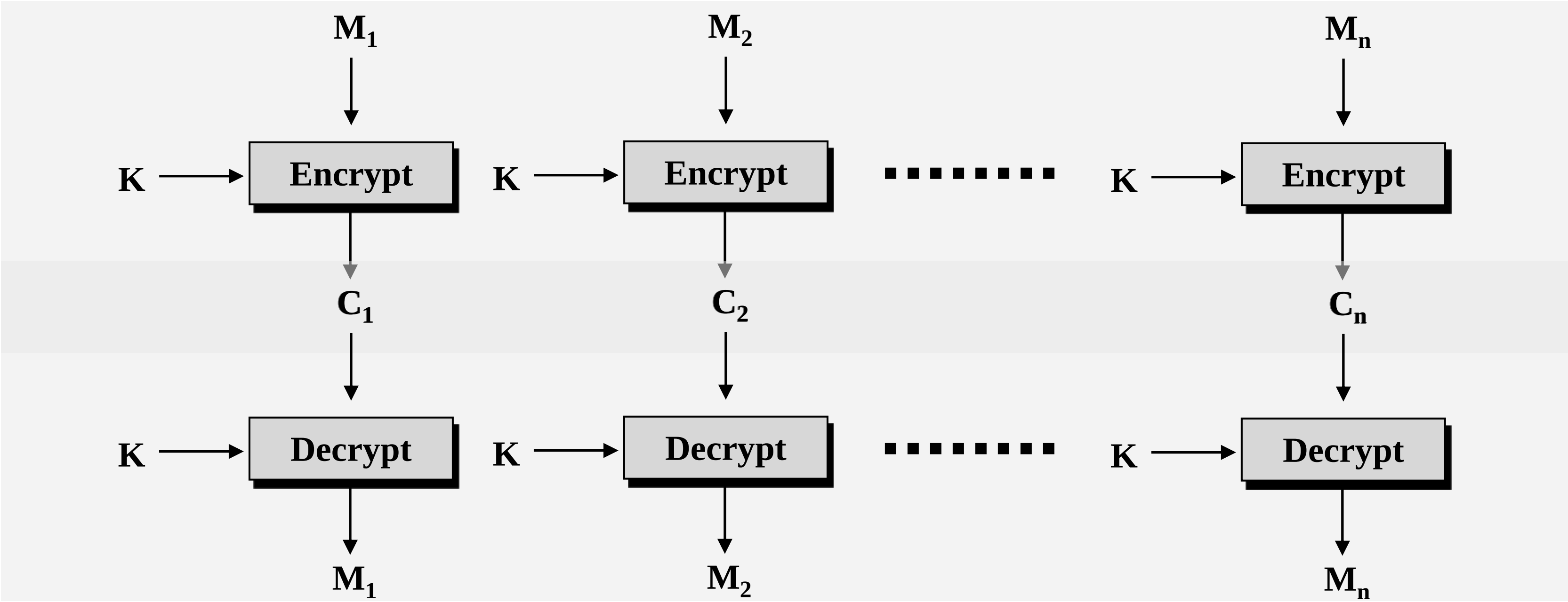
- With a block cipher, encrypting a *n*-bit block *M* with a key *k* gives a ciphertext block $C = E(M,k)$.
- Given a well designed block cipher, observing *C* would tell an adversary nothing about *M* or *k*.
- What happens if the adversary observes traffic over a longer period of time?
  - The adversary can detect if the same message had been sent before; if there are only two likely messages “buy” and “sell” it may be possible to guess the plaintext without breaking the cipher.

Block Ciphers: Modes of Operation

- Block ciphers can be used in different modes in order to provide different security services.
- Common modes include:
  - **E**lectronic **C**ode **B**ook (ECB)
  - **C**ipher **B**lock **C**haining (CBC)
  - **O**utput **F**eed**b**ack (OFB)
  - **C**ipher **F**eed**b**ack (CFB)
  - **C**oun**t**e**r** Mode (CTR)

Electronic Code Book

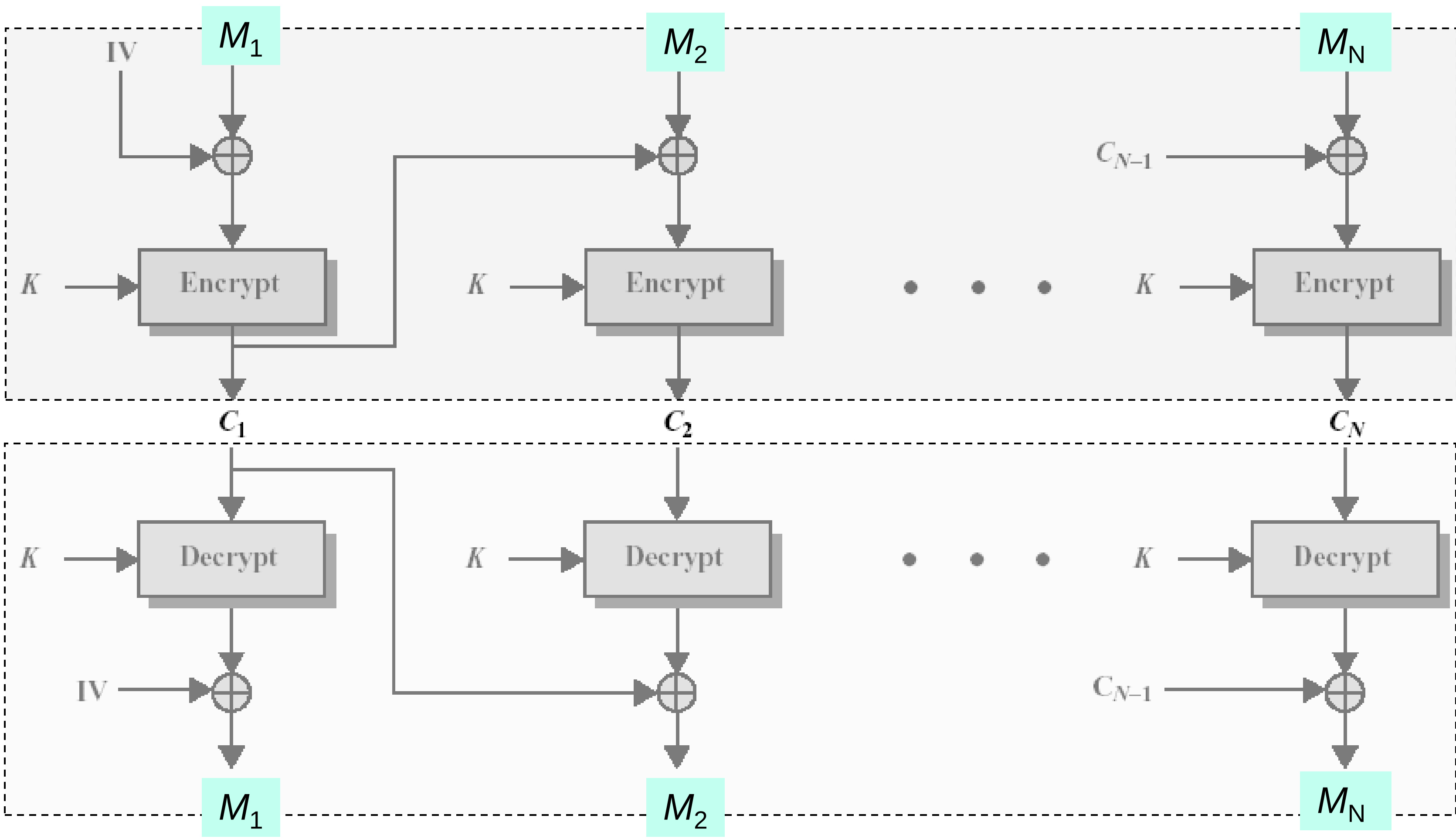
- ECB Mode encryption
  - Simplest mode of operation
  - Plaintext data is divided into blocks $M_1, M_2, \ldots, M_n$
  - Each block is then processed separately
    - Plaintext block and key used as inputs to the encryption algorithm



ECB Mode

- ECB Mode Issues
  - Problem: For a given key, the same plaintext block always encrypts to the same ciphertext block.
    - This may allow an attacker to construct a code book of known plaintext/ciphertext blocks.
    - The attacker could use this codebook to insert, delete, reorder or replay data blocks within the data stream without detection
  - Other modes of operation can prevent this, by not encrypting blocks independently
    - For example, using the output of one block encryption as input to the next (chaining)

Cipher Block Chaining Mode



CBC Mode

- CBC Mode Issues
  - Chaining guards against the construction of a code book
    - The same plaintext block encrypts to different ciphertext blocks each time.
  - May assist in detecting integrity breaches
    - Such as the insertion, deletion or reordering of data blocks into the ciphertext.
- What happens when there is an error?
  - If there is a bitflip error (0 to 1 or vice versa) that block and the following block will be decrypted incorrectly
  - If a ciphertext bit, or even a character is inserted or deleted this will be detected because of the incorrect ciphertext length
    - Not multiples of block size
  - Inserting or deleting a block will cause incorrect decryption

OFB Output Feedback Mode

A bit error in the ciphertext affects exactly the same bit in the plaintext.

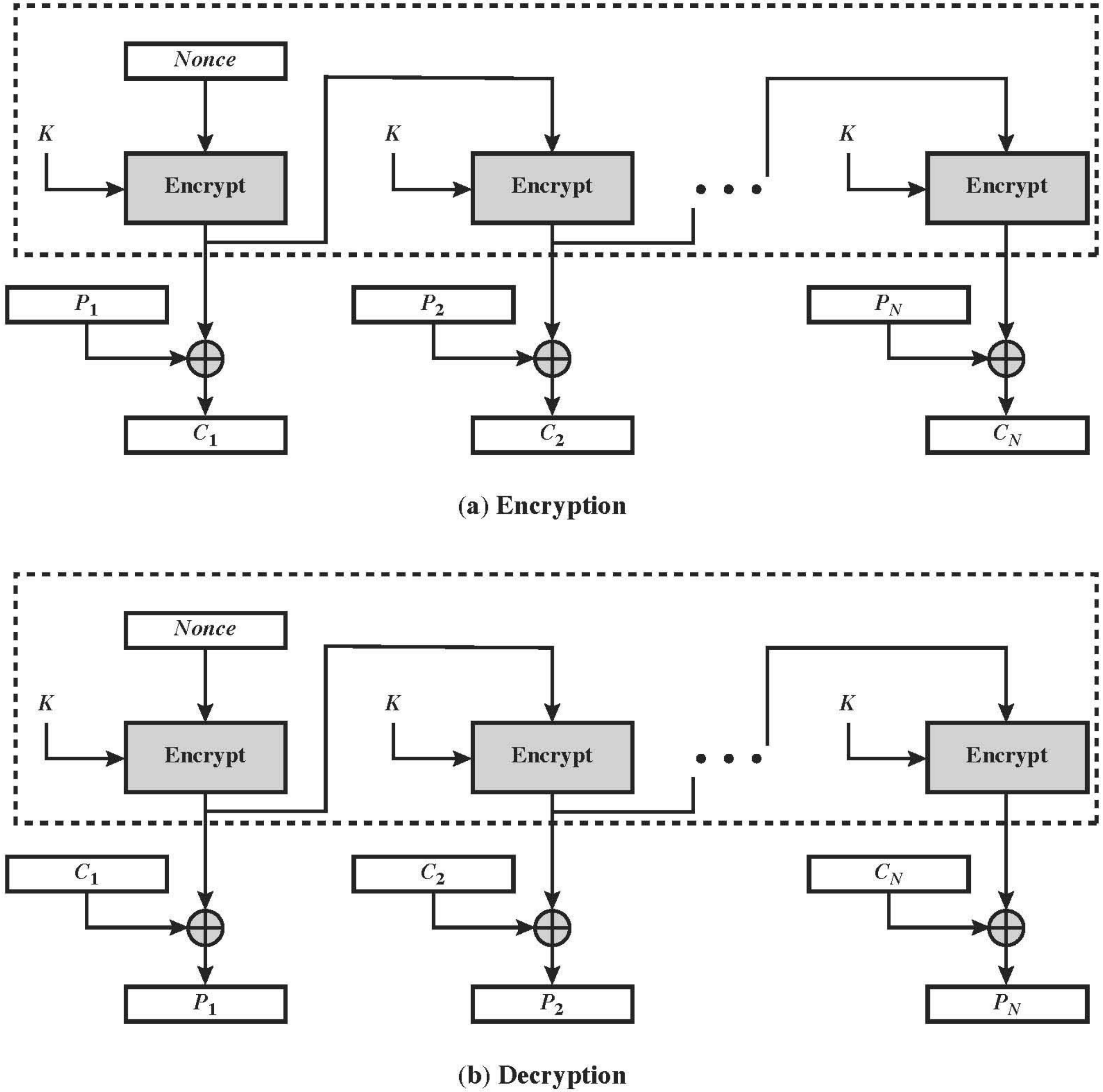
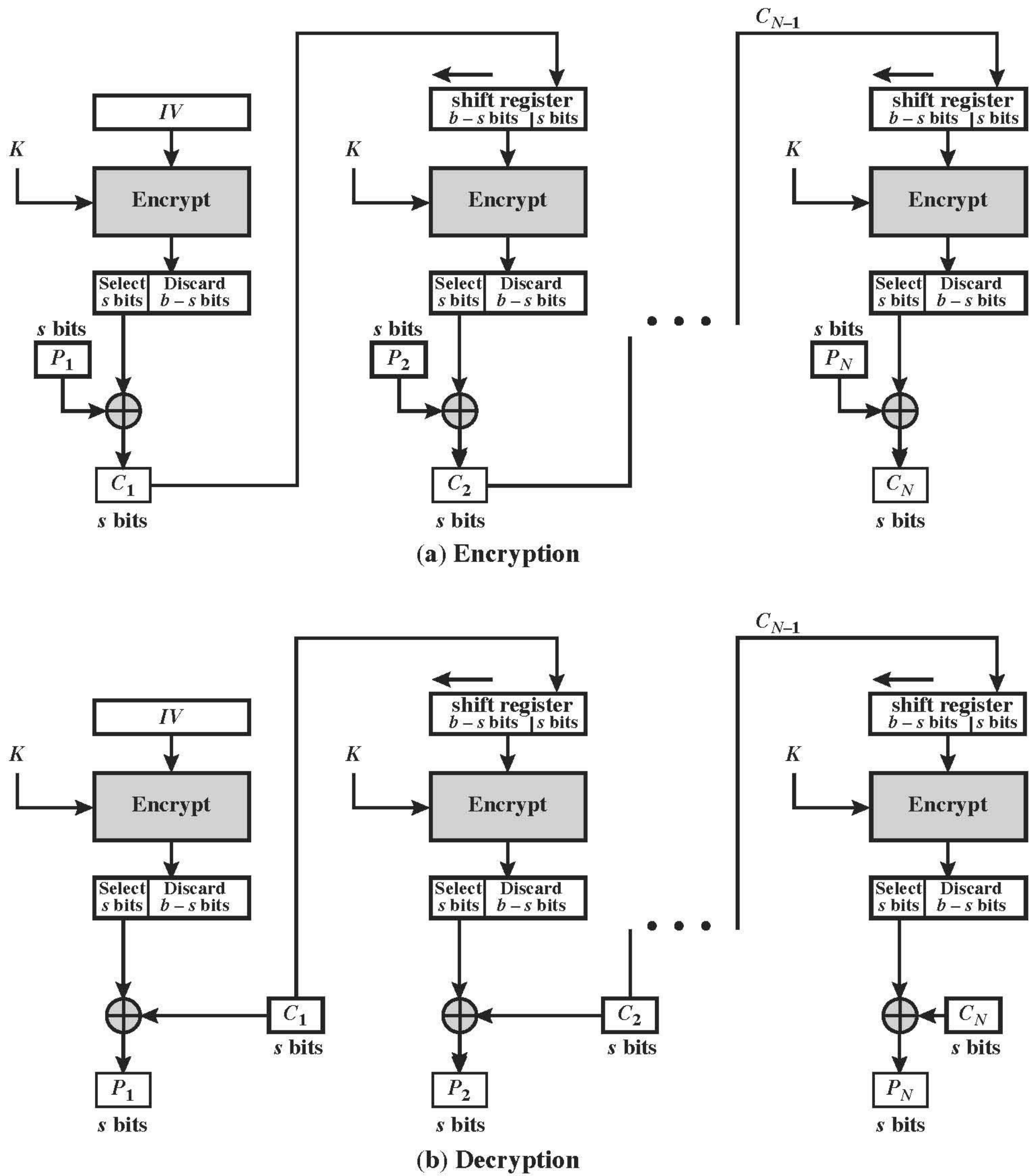


Figure 6.6 Output Feedback (OFB) Mode

Output Feedback Mode (OFB)

- Repeated plaintext blocks do not show up as repeated blocks in the ciphertext.
- Different encryptions of the same plaintext with the same key and IV give the same ciphertext.
- Encryption of different plaintexts with the same key and IV reveals information about the plaintexts.

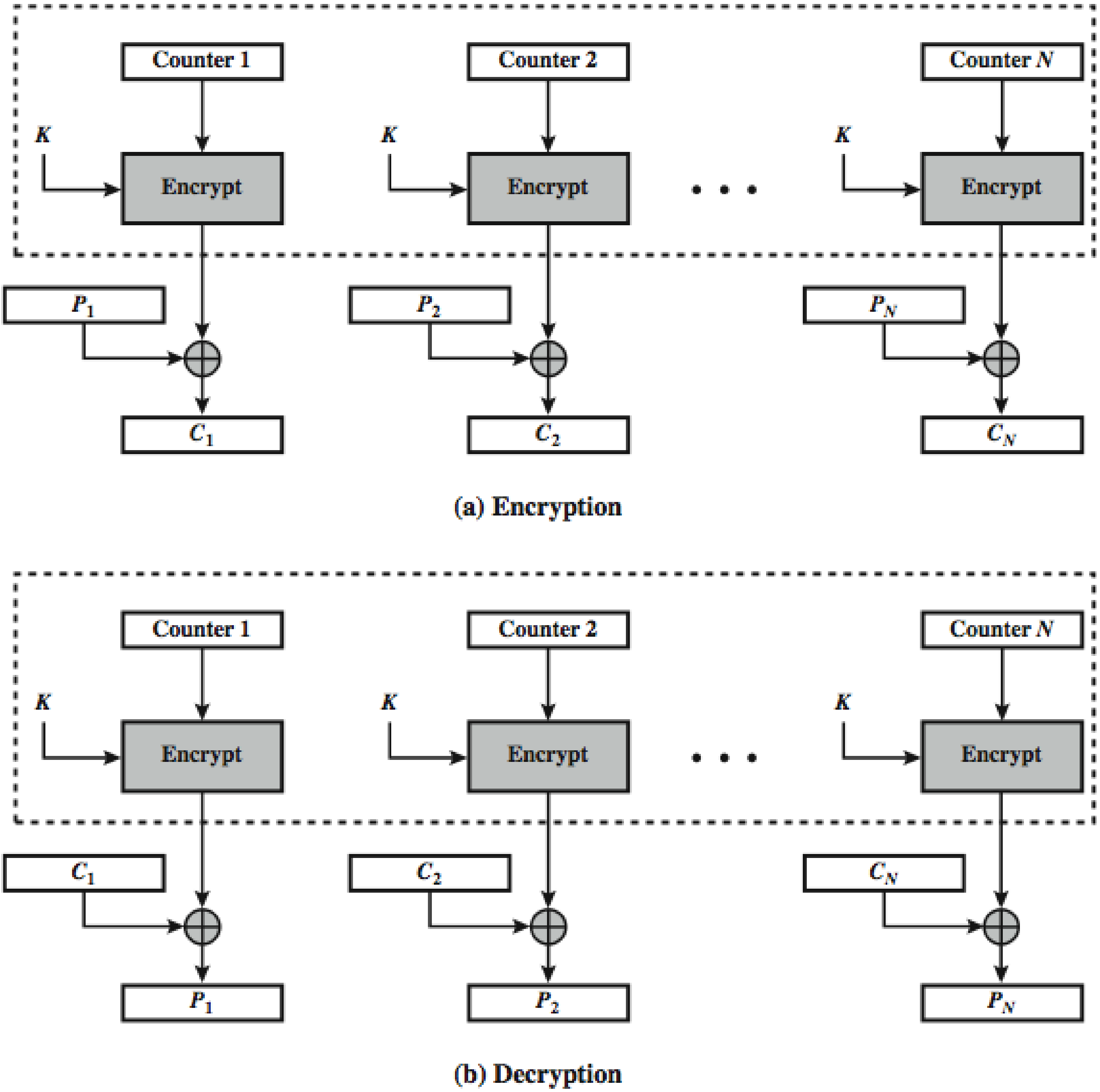
CFB Cipher Feedback Mode



Cipher Feedback Mode (CFB)

- Repeated plaintext blocks do not show up as repeated blocks in the ciphertext.
- Different encryptions of the same plaintext with the same key and IV give the same ciphertext.
- Encryption of different plaintexts with the same key and IV is not a security problem.
- A single bit error in a ciphertext block affects decryption until this block is shifted out of the register of the key generator.

CTR
Counter Mode



Counter (CTR)

- a “new” mode, though proposed early on
- similar to OFB but encrypts counter value rather than any feedback value
- must have a different key & counter value for every plaintext block (never reused)

$$O_i = E_K(i)$$

$$C_i = P_i \text{ XOR } O_i$$

- uses: high-speed network encryptions

Advantages and Limitations of CTR

- Efficiency
  - can do parallel encryptions in h/w or s/w
  - can preprocess in advance of need
  - good for bursty high speed links
- Random access to encrypted data blocks
- Provable security (good as other modes)
- But must ensure never reuse key/counter values, otherwise could break (cf OFB)

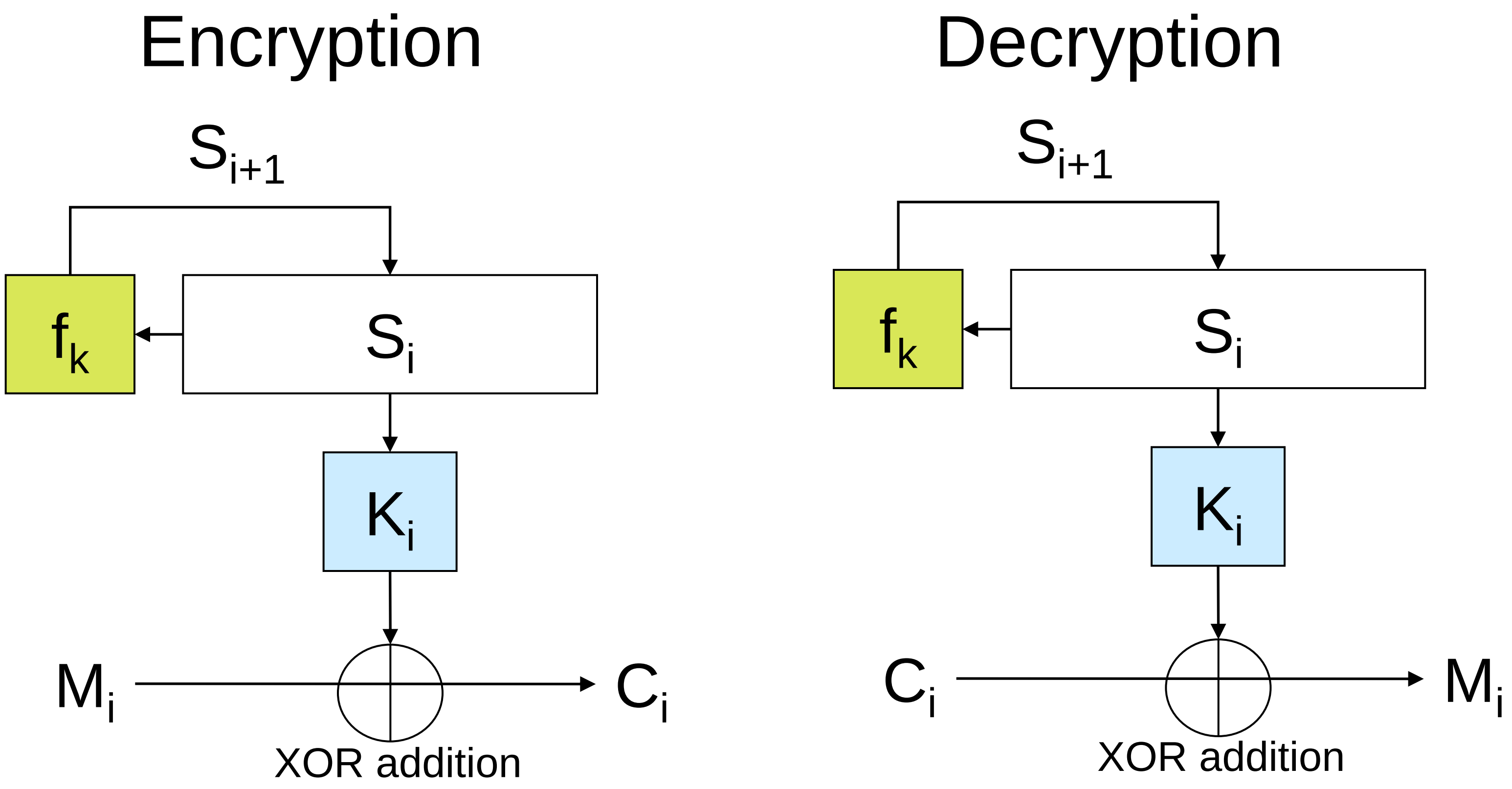
Block cipher: Applications

- Block ciphers are often used for providing confidentiality services
- They are used for applications involving processing large volumes of data, where time delays are not critical.
  - Examples:
    - Computer files
    - Databases
    - Email messages
- Block ciphers can also be used to provide integrity services, i.e. for message authentication

Stream Ciphers

- Consist of a key stream generator and a function for combining key stream and data.
- The combing function tends to be simple, exclusive-or is a typical example.
- The key stream generator takes as its input a key k seed S_0 and updates its state with a state transition function f_k, $S_{i+1} = f_k(S_i)$.
- The output at step i is the bitstream key K_i derived from S_i

Stream Ciphers



Encryption and decryption are usually identical operations.

Stream Ciphers

- In such a cipher, a bit error in ciphertext bit *i* causes a single bit error in plaintext bit *i*.
- Wireless networks use stream ciphers to protect data confidentiality.
- An adversary can make precise relative changes to the plaintext by modifying the corresponding ciphertext bits.
- Stream ciphers therefore cannot be used for integrity protection.

Is there a ‘perfect’ cipher?

- Yes - if you require confidentiality, the One Time Pad is provably secure.
- BUT we don’t use it much.

- To understand
  - why the OTP is not widely used, and
  - how to provide other security services like integrity or authentication

  you need to know a bit more about ciphers.

- Basically, there are two types: symmetric and asymmetric.

The perfect cipher: One-Time-Pad

- Famous example: *Vernam one-time pad*
  - One-time pad is the only provably secure cipher
  - Vernam OTP:
    - Plaintext is a stream of bits
    - Key is a <u>truly random binary sequence same length as message</u>
    - The cipher is a binary additive stream cipher, so
      - Ciphertext is obtained by binary addition (XOR) of plaintext and key
      - Plaintext is recovered by binary addition (XOR) of ciphertext and key
  - NOTE: Key can be used once only (hence the name), so each message requires a new, truly random key

Integrity Check Functions

Integrity protection

- Protection against modification of data can be done with integrity check values
  - CRC (cyclic redundancy code), message digest, hash functions etc.
- Unintentional modification (accidental errors) poses no threat to the integrity check values.
- Protecting integrity check values against intentional modifications relies on security
  - By access control, can be used for stored data
  - By cryptography, used in data communications

Cryptographic data integrity

- Data origin authentication includes data integrity: a message that has been modified in transit no longer comes from the original source.
- Data integrity includes data origin authentication: when the sender's address is part of the message, you have to verify the source of a message when verifying its integrity.
- Under the assumptions made, data integrity and data origin authentication are equivalent.
- In other applications a separate notion of data integrity makes sense, e.g. for file protection by anti-virus software.

Hash functions (message digest functions)

Requirements on a one-way hash function h:

- Ease of computation: given x, it is easy to compute $h(x)$.
- Compression: h maps inputs x of arbitrary bitlength to outputs $h(x)$ of a fixed bitlength n.
- Pre-image resistance (one-way): given a value y, it is computationally infeasible to find an input x so that $h(x)=y$.

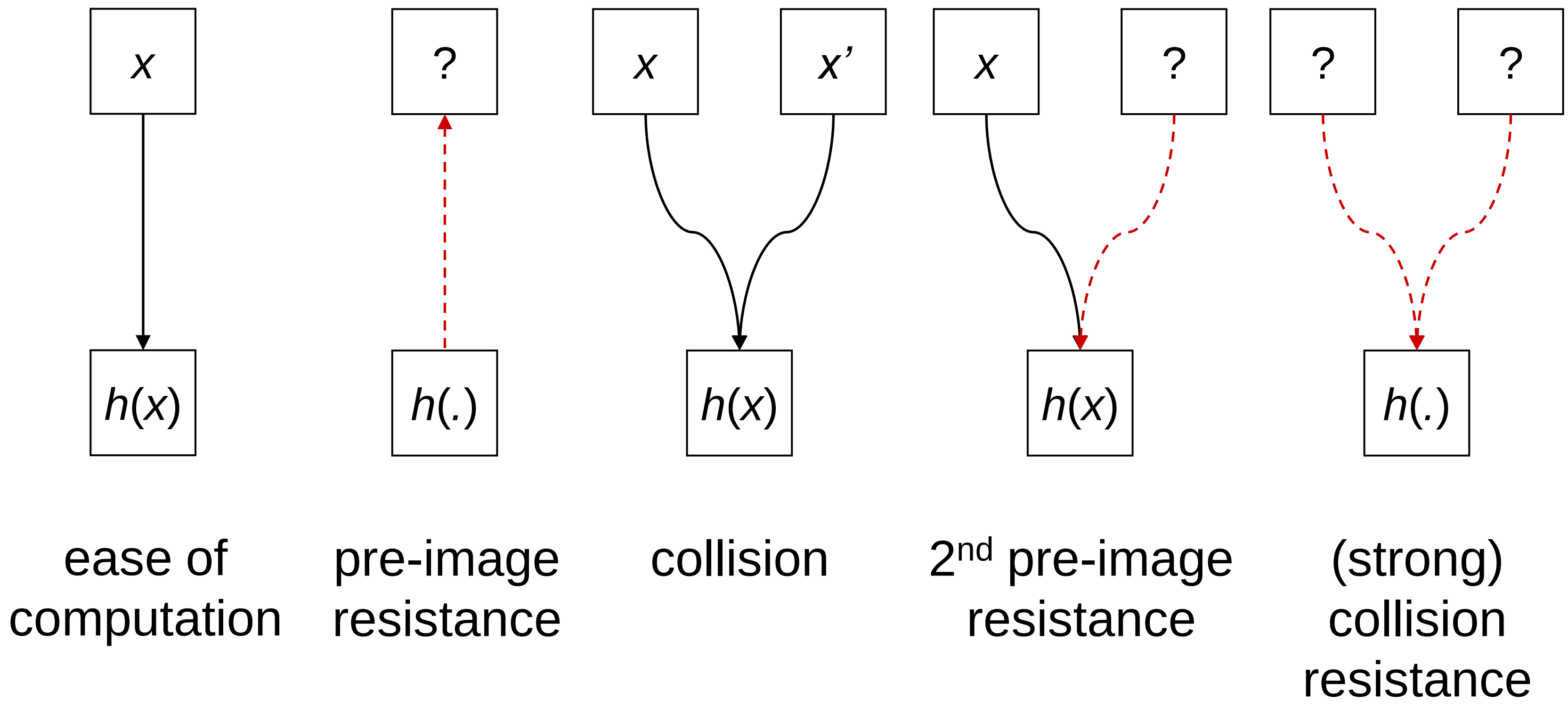
Hash collisions

- The application just described needs more than the one-way property of *h*.
- We are not concerned about an attacker reconstructing the message from the hash.
- We are concerned about attackers who change message *x* to *x'* so that *h*(*x'*)= *h*(*x*).
- Then, our integrity protection mechanism would fail to detect the change.
- We say there is a collision when two inputs *x* and *x'* map to the same hash.

Collision Resistance

- Integrity protection requires collision-resistant hash functions; we distinguish between:
- 2nd pre-image resistance (weak collision resistance): given an input x and $h(x)$, it is computationally infeasible to find another input x', $x \neq x'$, with $h(x)=h(x')$.
- Collision resistance (strong collision resistance): it is computationally infeasible to find any two inputs x and x', $x \neq x'$, with $h(x)=h(x')$.

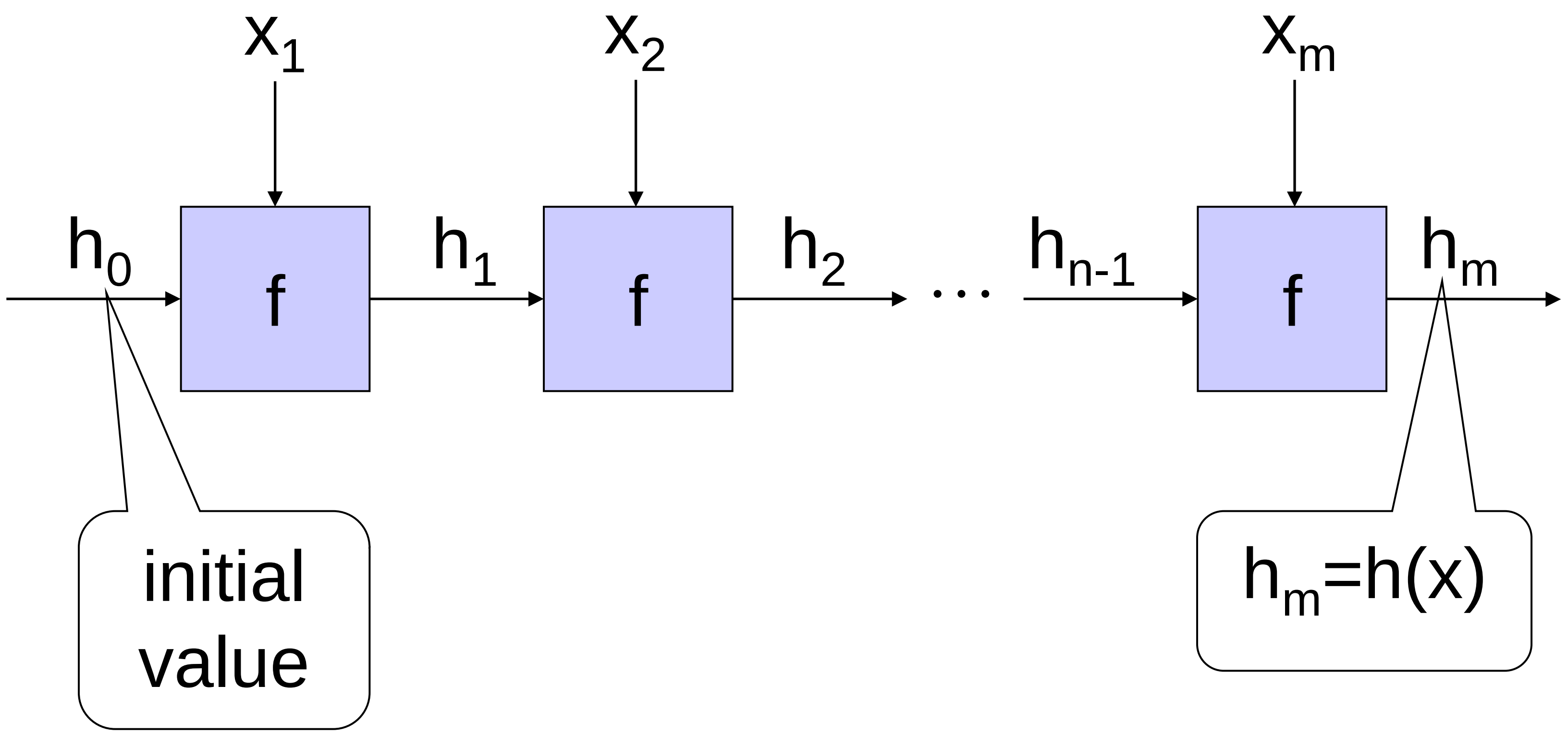
Properties of hash functions



Hash function construction

- Pattern for the design of fast hash functions:
- The core of the hash function is a compression function f that works on fixed size input blocks.
- An input x of arbitrary length is broken up into blocks $x_1, \ldots, x_m$ of the given block size; the last block has to be padded.
- Compute the hash of x by repeatedly applying the compression function: with a (fixed) initial value h_0, compute $h_i = f(x_i, h_{i-1})$ for $i=1, \ldots, m$ and take h_m as the hash value of x.

Hash function principle



Frequently used hash functions

- MD5: 128 bit digest. Broken. Often used in Internet protocols but no longer recommended.
- SHA-1 (Secure Hash Algorithm):160 bit digest. Potential attacks exist. Designed to operate with the US Digital Signature Standard (DSA);
- SHA-256, 384, 512 bit digest. Still secure. Replacement for SHA-1
- RIPEMD-160: 160 bit digest. Still secure. Hash function frequently used by European cryptographic service providers.
- NIST competition for new secure hash algorithm, announcement in 2011.

Message Authentication Codes

- A message *M* with a simple message hash *h*(*M*) can be changed by attacker.
- In communications, we need to verify the origin of data, i.e. we need message authentication.
- MAC (message authentication code) computed as *h*(*M, k*) from message *M* and a secret key *k*.
- To validate and authenticate a message, the receiver has to share the same secret key used to compute the MAC with the sender.
- A third party who does not know the key cannot validate the MAC.

MAC and MAC algorithms

- MAC means two things:
  1. The computed message authentication code $h(M, k)$
  2. General name for algorithms used to compute a MAC
- In practice, the MAC algorithm is e.g.
  – HMAC (Hash-based MAC algorithm))
  – CBC-MAC (CBC based MAC algorithm)
  – CMAC (Cipher-based MAC algorithm)
- MAC algorithms, a.k.a. keyed hash functions, support data origin authentication services.

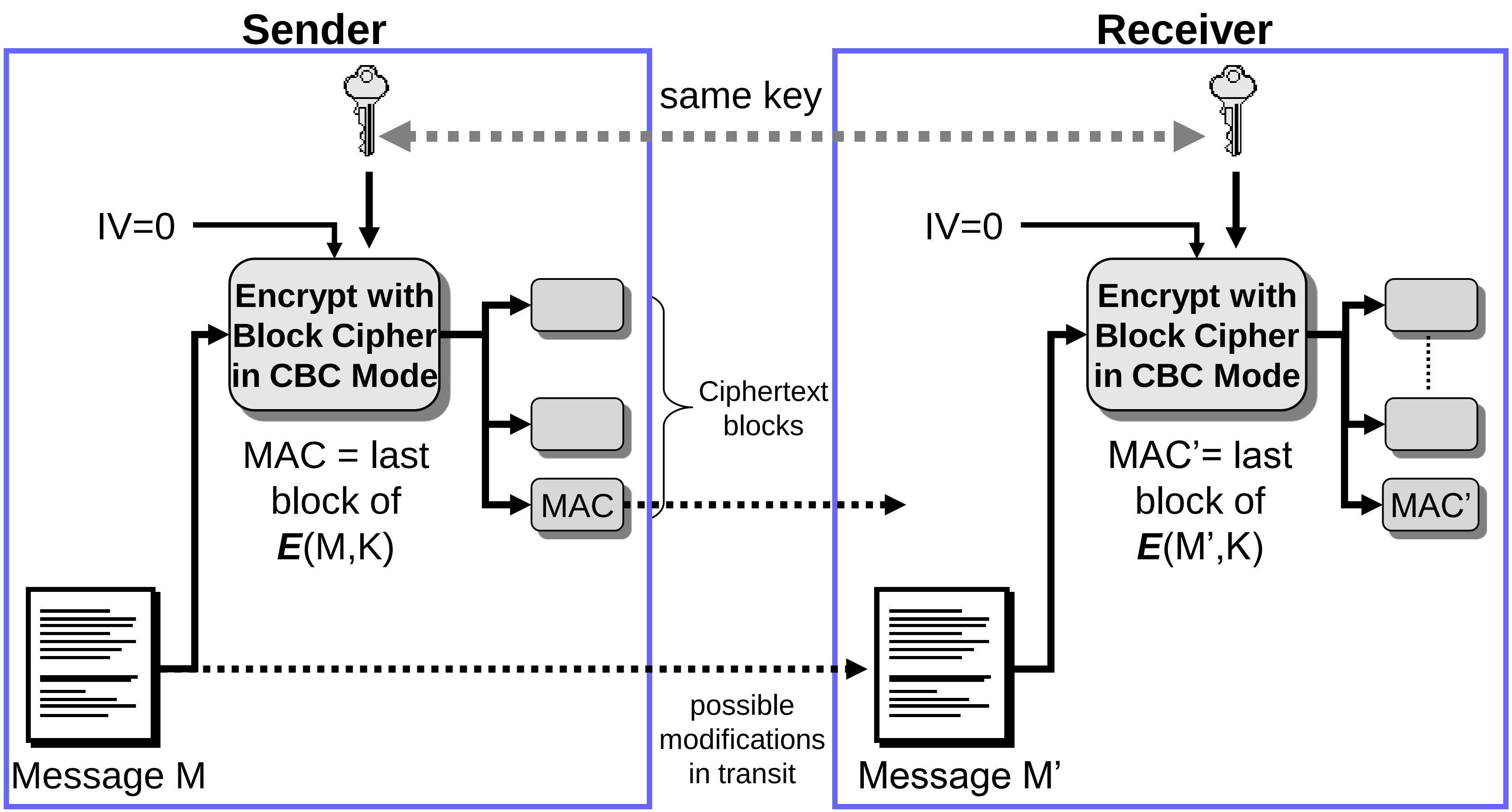
HMAC

- A MAC algorithm can be derived from a hash algorithm *h* using the HMAC construction:
- For a given key *k* and message *M*, compute

  $HMAC(x) = h(k||p_1||h(k||p_2||M))$

  where p_1 and p_2 are bit strings (padding) that extend *k* to a full block length of the compression function used in *h*.
- HMAC is specified in Internet RFC 2104.

CBC-MAC principle



CBC-MAC and CMAC

- A block cipher used in CBC mode can be used as a MAC algorithm
  - For a given message or data file,
    - The file is encrypted using the block cipher in CBC mode
    - The last ciphertext block is used as a Message Authentication Code (MAC) value
    - Both the message and the MAC are sent to the receiver
  - Described in ISO/IEC 9797-1:1999
  - Security depends on block-size of cipher. The typical block size of 128 bit is too short.
- CMAC uses a block cipher in a way to output hash blocks that are larger than the cipher block.

Security of hash functions

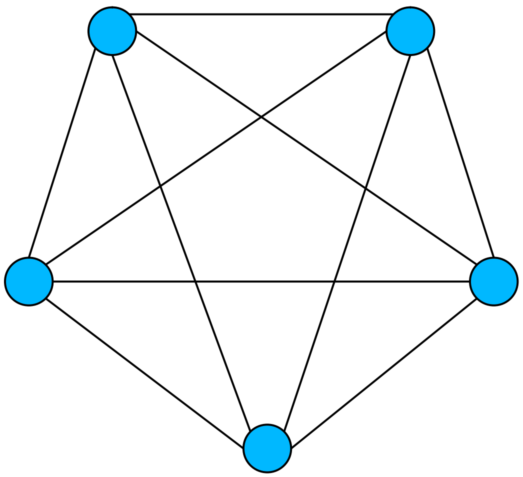
- Large block size necessary to resist birthday attacks.
- Birthday paradox:
  - A group of 253 persons is needed to have p = 0.5 that any person has birthday on a specific date. Seems reasonable.
  - A group of only 23 persons is needed to have p = 0.5 that any two persons have birthday on the same date. Seems strange.
- Finding any two hashes that are equal (collision) in a large table of hash values is therefore relatively easy.
- A block size of n bits is considered to provide only $n/2$ bit complexity.
- To provide strong collision resistance, large blocks are needed. 160 bit hash block is currently a minimum.

Hash functions and Message Authentication

- Shared secret key is used for HMAC, CBC-MAC and CMAC
- When used during message transmission, this provides an additional security service of Message Authentication:
  - A correct MAC value confirms the sender of the message is in possession of the shared secret key
  - Hence, much like a password, it confirms the authenticity of the message sender to the receiver.
- Indeed, message integrity is meaningless without knowing who sent the message.

Symmetric key distribution

- Shared key between each pair
- Each participant needs n-1 keys.
- Total number of exchanged keys:
  $= (n-1) + (n-2) + \ldots + 2 + 1$
  $= n(n-1)/2$
- Grows exponentially, which is problematic.
- Is there a better way?



Community of 5 nodes

Public-Key Cryptography

James H. Ellis (1924 – 1997)

- British engineer and mathematician
- Worked at GCHQ (Government Communications Headquarters)
- Idea of non-secret encryption to solve key distribution problem
- Encrypt with non-secret information in a way which makes it impossible to decrypt without related secret information
- Never found a practical method

Clifford Cocks (1950 –)

- British mathematician and cryptographer
- Silver medal at the International Mathematical Olympiad, 1968
- Works at GCHQ
- Heard from James Ellis the idea of non-secret encryption in 1973
- Spent 30 minutes in 1973 to invent a practical method
- Equivalent to the RSA algorithm
- Was classified TOP SECRET
- Revealed in 1998

Malcolm J. Williamson

- British mathematician and cryptographer
- Gold medal at the International Mathematical Olympiad, 1968
- Worked at GCHQ until 1982
- Heard from James Ellis the idea of non-secret encryption, and from Clifford Cocks the practical method.
- Intrigued, spent 1 day in 1974 to invent a method for secret key exchange without secret channel
- Equivalent to the Diffie-Hellmann key exchange algorithm

Public Key Encryption

- Proposed in the open literature by Diffie & Hellman in 1976.
- Each party has a public encryption key and a private decryption key.
- Reduces total number of exchanged keys to *n*
- Computing the private key from the public key should be computationally infeasible.
- The public key need not be kept secret but it is not necessarily known to everyone.
- There can be applications where even access to public keys is restricted.

Ralph Merkle, Martin Hellman and Whitfield Diffie

- Merkle invented (1974) and published (1978) Merkle's puzzle, a key exchange protocol which was unpractical

- Diffie & Hellman invented (influenced by Merkle) a practical key exchange algorithm using discrete logarithm.

- D&H defined public-key encryption (equiv. to non-secret encryption)
- Defined digital signature
- Published 1976 in "*New directions in cryptography*"

Diffie-Hellman key agreement

Alice picks random integer a

$g^a \bmod p$ →

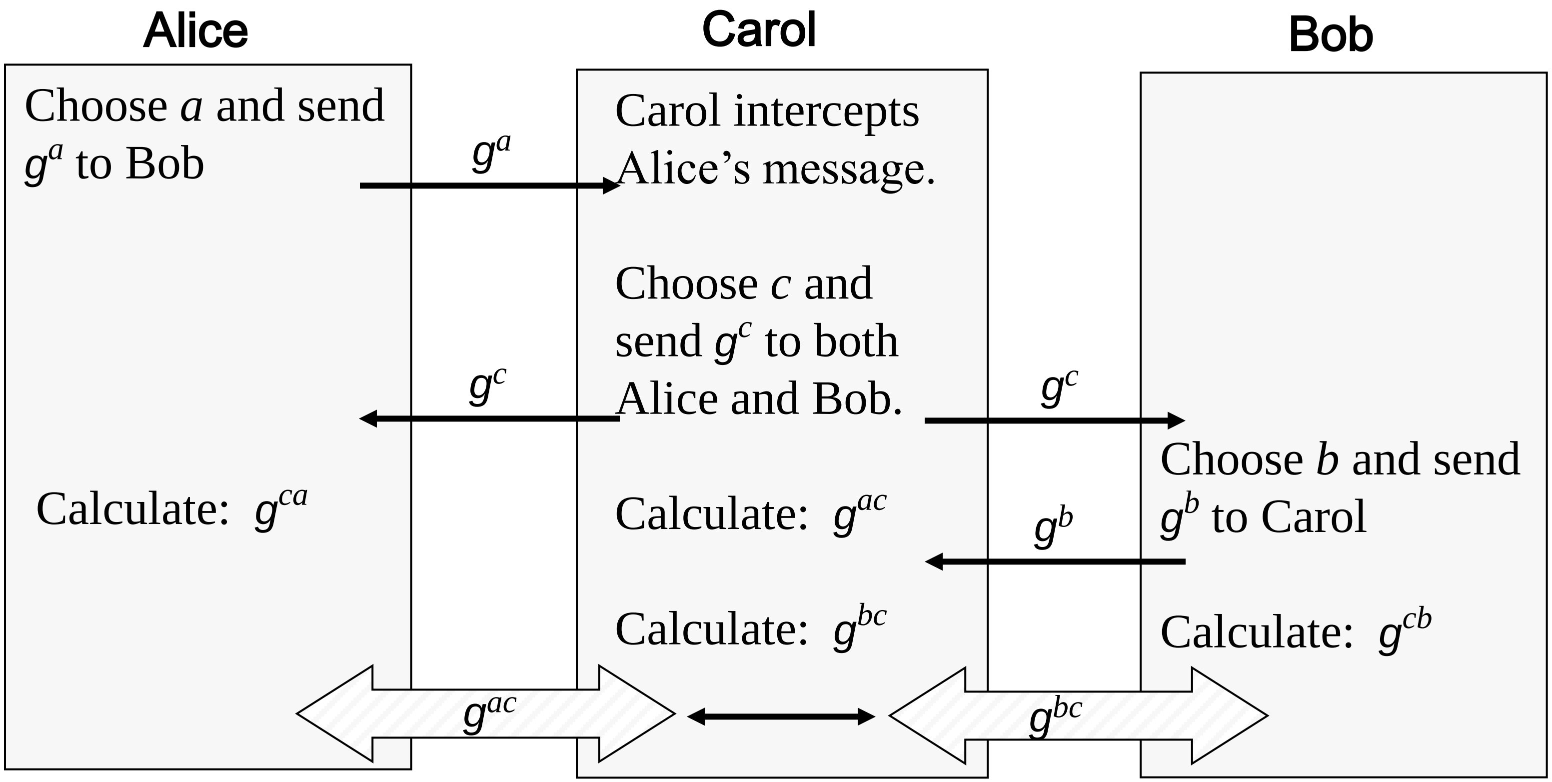
Bob picks random integer b

← $g^b \bmod p$

Alice computes the shared secret
$(g^b)^a = g^{ab} \bmod p$

Bob computes the same secret
$(g^a)^b = g^{ab} \bmod p$.

Man-in-the-middle attack in Diffie-Hellman



Carol now shares keys with both Alice and Bob. Alice sends confidential messages believing that she is communicating with Bob. Carol can decrypt these messages, read them and re-encrypt the message for Bob and send it on. Likewise, Bob sends messages to Alice unaware of Carol.

Diffie-Hellman Applications

- IPSec (IP Security)
  - IKE (Internet Key Exchange) is part of the IPSec protocol suite
  - IKE is based on Diffie-Hellman Key Agreement
- SSL/TLS
  - Several variations of SSL/TLS protocol including
    - Fixed Diffie-Hellman
    - Ephemeral Diffie-Hellman
    - Anonymous Diffie-Hellman

Public key encryption algorithms

- Each party *B* has a public encryption key and a private decryption key.
- A method is required for each communicating party *A* to get an authentic copy of *B*'s public key (hopefully easier than getting a shared secret key).
- For *n* parties, only *n* key pairs are needed
  - as opposed to n(n-1)/2 symmetric keys.

Ron Rivest, Adi Shamir and Len Adleman

- Read about public-key cryptography in 1976 article by Diffie & Hellman: *“New directions in cryptography”*
- Intrigued, they worked on finding a practical algorithm
- Spent several months in 1976 to re-invent the method for non-secret/public-key encryption discovered by Clifford Cocks 3 years earlier
- Named RSA algorithm

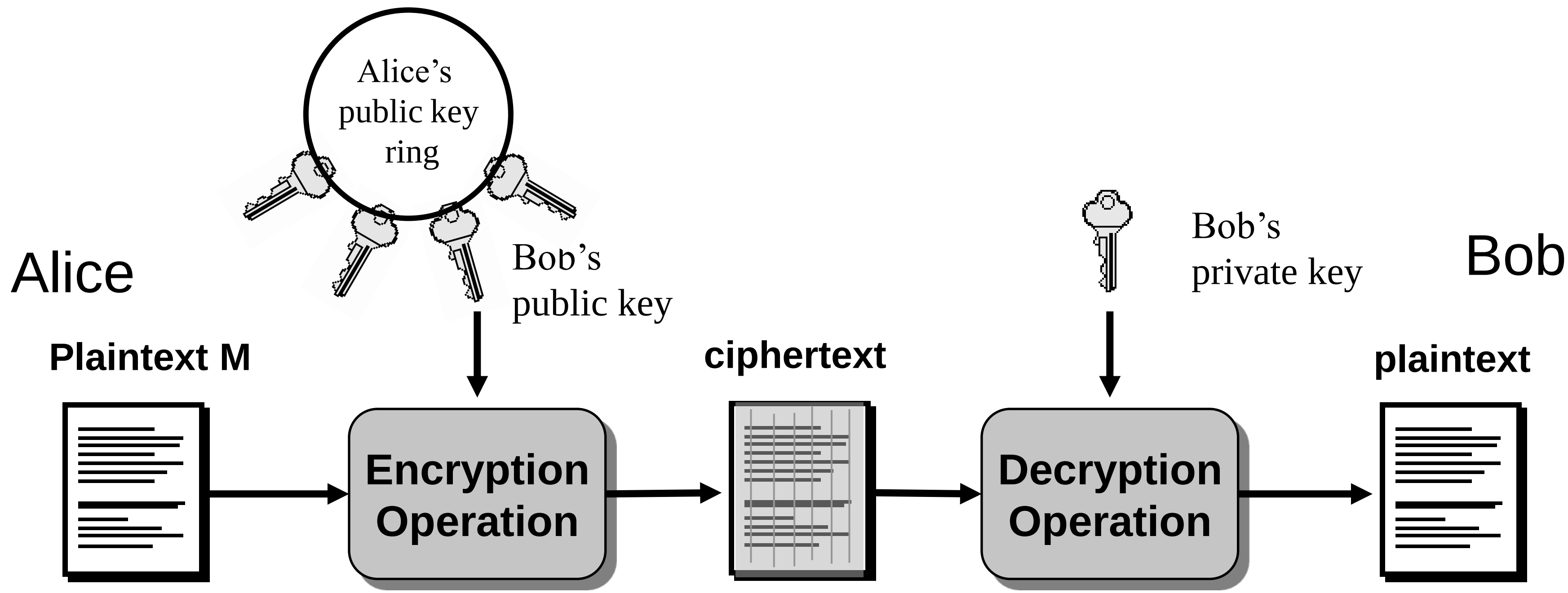
RSA Algorithm

- $n = pq$ which is made public (but not p and q)
- Calculate secret: $z = (p-1)(q-1)$
- Choose a public key e
- Compute private key d such that $ed = 1 \bmod (z)$
- Encryption of message m where $(1 < m < n)$.
  - Compute: $c = m^e \bmod n$
- Decryption of ciphertext c
  - Compute: $m = c^d \bmod n$
- Security depends on the difficulty of factorizing n
  - so the prime factors p and q must be LARGE

Asymmetric Ciphers: Examples of Cryptosystems

- RSA: best known asymmetric algorithm.
  - RSA = Rivest, Shamir, and Adleman (published 1977)
  - Historical Note: U.K. cryptographer Clifford Cocks invented the same algorithm in 1973, but didn’t publish.
- ElGamal Cryptosystem
  - Based on the difficulty of solving the discrete log problem.
- Elliptic Curve Cryptography
  - Based on the difficulty of solving the EC discrete log problem.
  - Provides same level of security with smaller key sizes.

Asymmetric Encryption: Basic encryption operation



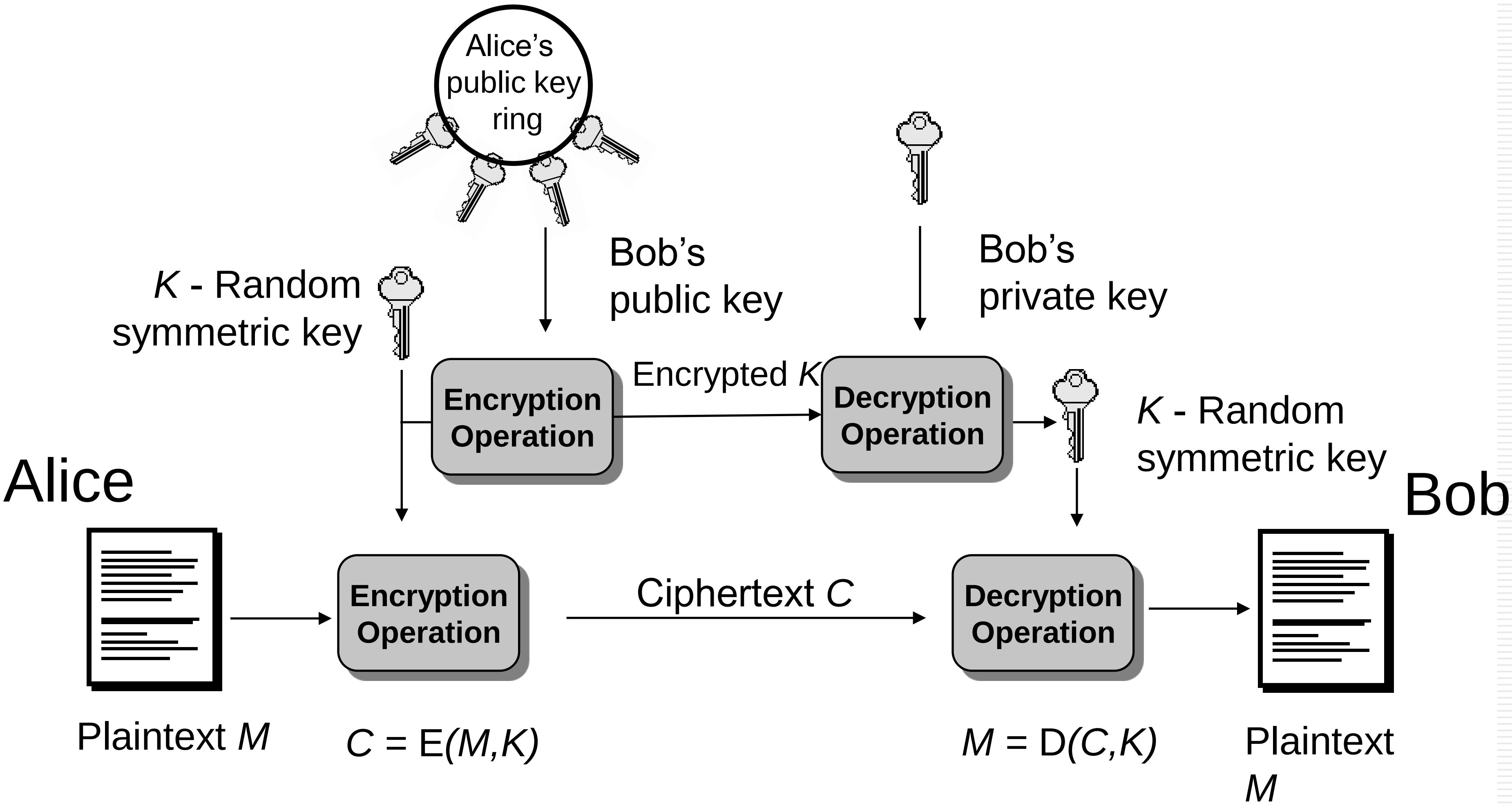
$C = E(M, K_{pub})$ $M = D(C, K_{priv})$

- In practical application, large messages are not encrypted directly with asymmetric algorithms. Hybrid systems are used.

Hybrid Cryptosystems

- Symmetric ciphers are faster than asymmetric ciphers (because they are less computationally expensive), but ...
- Asymmetric ciphers simplify key distribution, therefore ...
- a combination of both symmetric and asymmetric ciphers can be used – a hybrid system:
  - The asymmetric cipher is used to distribute a randomly chosen symmetric key.
  - The symmetric cipher is used for encrypting bulk data.

Confidentiality Services: Hybrid Cryptosystems

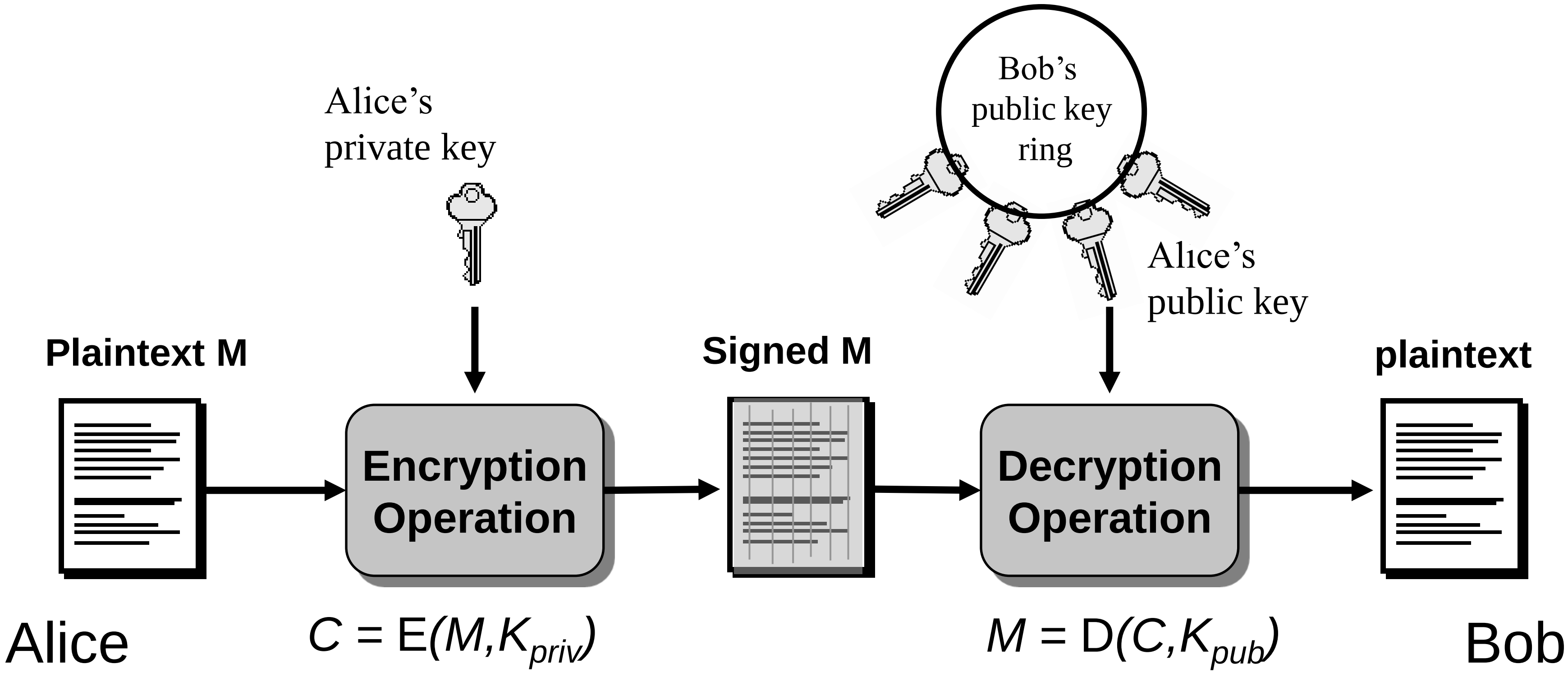


Digital Signatures

Digital Signature Mechanisms

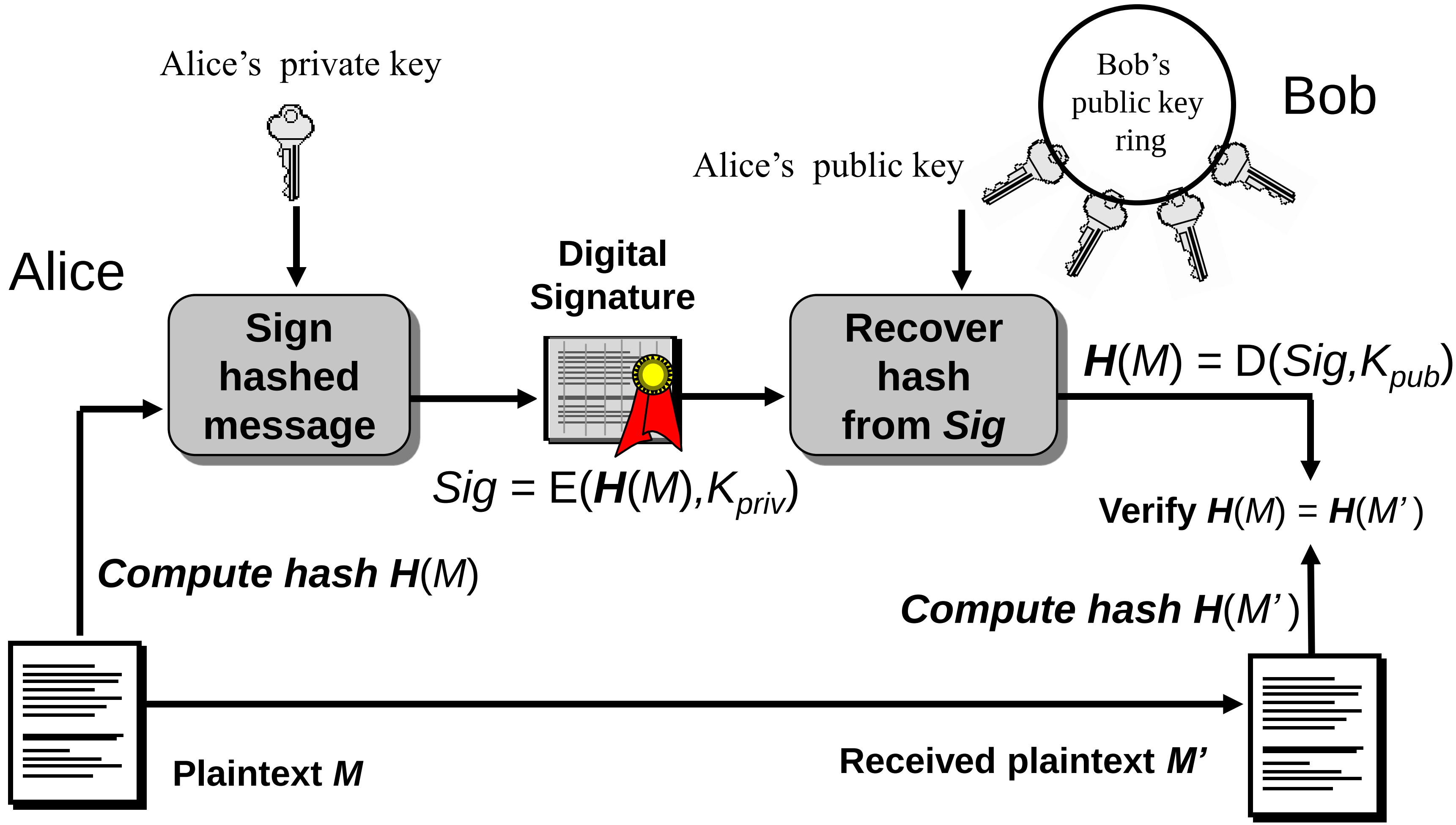
- A MAC cannot be used as evidence that should be verified by a third party.
- Digital signatures used for non-repudiation, data origin authentication and data integrity services, and in some authentication exchange mechanisms.
- Digital signature mechanisms have three components:
  - key generation
  - signing procedure (private)
  - verification procedure (public)

Digital signature: Basic operation



- In practical applications, message M is not signed directly, only a hash value h(M) is signed.

Digital signature based on hash value



Digital Signatures

- *A* has a public verification key and a private signature key ($\rightarrow$ public key cryptography).
- *A* uses her private key to compute her signature on document *m*.
- *B* uses a public verification key to check the signature on a document *m* he receives.
- At this technical level, digital signatures are a cryptographic mechanism for associating documents with verification keys.

Digital Signatures

- To get an authentication service that links a document to *A*’s name (identity) and not just a verification key, we require a procedure for *B* to get an authentic copy of *A*’s public key.
- Only then do we have a service that proves the authenticity of documents ‘signed by *A*’.
- This can be provided by a PKI (Public Key Infrastructure)
- Yet even such a service does not provide non-repudiation at the level of persons.

Difference between MACs & Dig. Sig.

- MACs and digital signatures are both authentication mechanisms.

- MAC: the verifier needs the secret that was used to compute the MAC; thus a MAC is unsuitable as evidence with a third party.
  - The third party does not have the secret.
  - The third party cannot distinguish between the parties knowing the secret.

- Digital signatures can be validated by third parties, and can in theory thereby support both non-repudiation and authentication.

Key length comparison:

Symmetric and Asymmetric ciphers offering comparable security

AES Key Size	RSA Key Size	Elliptic curve Key Size
-	1024	163
128	3072	256
192	7680	384
256	15360	512

Ciphers and security

- A cipher must
  - be hard to cryptanalyse
  - use a sufficiently large key
- Algorithm secrecy makes cryptanalysis harder, but
  - can give false assurance, i.e. “security by obscurity”
  - challenging to keep cipher design confidential
  - safest to assume that attacker knows cipher

- Auguste Kerckhoffs proposed in 1883 that communication secrecy should only be based on the secrecy of the key
- Still, many organisations use secret algorithms.

Hopefully, you now know:

- What cryptography is
- When cryptography is used
- That there is a ‘perfect’ cipher, but that it has serious limitations
- That there are practical ciphers and how to use them
  - Symmetric ciphers (stream and block)
  - Asymmetric ciphers
- Hash and MAC functions and how to use them
- Hybrid cryptosystems and digital signatures