F2

Symmetric Encryption

Cryptographic Algorithms: Overview

- During this course two main applications of cryptographic algorithms are of principal interest:
  - *Encryption* of data: transforms plaintext data into ciphertext in order to conceal its' meaning
  - *Signing* of data: computes a *check value* or *digital signature* to a given plain- or ciphertext, that can be verified by some or all entities being able to access the signed data
- Some cryptographic algorithms can be used for both purposes, some are only secure and / or efficient for one of them.
- Principal categories of cryptographic algorithms:
  - *Symmetric cryptography* using 1 key for en-/decryption or signing/checking
  - *Asymmetric cryptography* using 2 different keys for en-/decryption or signing/checking
  - *Cryptographic hash functions* using 0 keys (the "key" is not a separate input but "appended" to or "mixed" with the data).

Attacking Cryptography (1): Cryptanalysis

- *Cryptanalysis* is the process of attempting to discover the plaintext and / or the key
- Types of cryptanalysis:
  - *Ciphertext only:* specific patterns of the plaintext may remain in the ciphertext (frequencies of letters, digraphs, etc.)
  - *Known ciphertext / plaintext pairs*
  - *Chosen plaintext or chosen ciphertext*
  - Newer developments: *differential cryptanalysis, linear cryptanalysis*
- Cryptanalysis of public key cryptography:
  - The fact that one key is publicly exposed may be exploited
  - Public key cryptanalysis is more aimed at breaking the cryptosystem itself and is closer to pure mathematical research than to classical cryptanalysis
  - Important directions:
    - Computation of discrete logarithms
    - Factorization of large integers

Attacking Cryptography (2): Brute Force Attack

- The *brute force attack* tries every possible key until it finds an intelligible plaintext:
  - Every cryptographic algorithm can in theory be attacked by brute force
  - On average, half of all possible keys will have to be tried

Average Time Required for Exhaustive Key Search

Key Size [bit]	Number of keys	Time required at 1 encryption / µs	Time required at 10^6 encryption / µs
32	$2^{32} = 4.3 * 10^9$	2^{31} µs = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 * 10^{16}$	2^{55} µs = 1142 years	10.01 hours
128	$2^{128} = 3.4 * 10^{38}$	2^{127} µs = $5.4 * 10^{24}$ years	$5.4 * 10^{18}$ years

Attacking Cryptography (3): How large is large?

Reference Numbers Comparing Relative Magnitudes		
Reference		Magnitude
Seconds in a year		$\approx 3 * 10^{7}$
Seconds since creation of solar system		$\approx 2 * 10^{17}$
Clock cycles per year (50 MHz computer)		$\approx 1.6 * 10^{15}$
Binary strings of length 64	2^{64}	$\approx 1.8 * 10^{19}$
Binary strings of length 128	2^{128}	$\approx 3.4 * 10^{38}$
Binary strings of length 256	2^{256}	$\approx 1.2 * 10^{77}$
Number of 75-digit prime numbers		$\approx 5.2 * 10^{72}$
Electrons in the universe		$\approx 8.37 * 10^{77}$

Important Properties of Encryption Algorithms

Consider, a sender is encrypting plaintext messages P_1, P_2, ... to ciphertext messages C_1, C_2, ...

Then the following properties of the encryption algorithm are of special interest:

- *Error propagation* characterizes the effects of bit-errors during transmission of ciphertext to reconstructed plaintext $P_1´$, $P_2´$, ...
  - Depending on the encryption algorithm there may be one or more erroneous bits in the reconstructed plaintext per erroneous ciphertext bit

- *Synchronization* characterizes the effects of lost ciphertext data units to the reconstructed plaintext
  - Some encryption algorithms can not recover from lost ciphertext and need therefore explicit re-synchronization in case of lost messages
  - Other algorithms do automatically re-synchronize after 0 to n (n depending on the algorithm) ciphertext bits

Classification of Encryption Algorithms: Three Dimensions

- The type of operations used for transforming plaintext to ciphertext:
  - *Substitution*, which maps each element in the plaintext (bit, letter, group of bits or letters) into another element
  - *Transposition,* which re-arranges elements in the plaintext
- The number of keys used:
  - *Symmetric ciphers,* which use the same key for en- / decryption
  - *Asymmetric ciphers,* which use different keys for en- / decryption
- The way in which the plaintext is processed:
  - *Stream ciphers* work on bit streams and encrypt one bit after another:
    - Many stream ciphers are based on the idea of linear feedback shift registers, and there have been detected vulnerabilities of a lot of algorithms of this class, as there exists a profound mathematical theory on this subject.
    - Most stream ciphers do not propagate errors but are sensible to loss of synchronization.
  - *Block ciphers* work on blocks of width *b* with *b* depending on the specific algorithm.

Symmetric Encryption

- General description:
  - The same key $K_{A,B}$ is used for enciphering and deciphering of messages:

- Notation:
  - If P denotes the plaintext message $E(K_{A,B}, P)$ denotes the ciphertext and it holds $D(K_{A,B}, E(K_{A,B}, P)) = P$
  - Alternatively we sometimes write $\{P\}_{K_{A,B}}$ or $E_{K_{A,B}}(P)$ for $E(K_{A,B}, P)$
- Examples: DES, 3DES, AES, ...

Two types of ciphers

- Substitution Cipher.
  - In the simplest form: Each letter is replaced by another letter.
  - More realistically: Each block of letters is replaced by another block of letters.
- Transposition Cipher (Permutation).
  - The letters in a block are permuted. The same permutation is used for every block.

Confusion and Diffusion

Concepts defined by Claude Shannon

- Diffusion
  - Dependency between output and input .
  - Ideally, flipping one input bit should flip each output bit with probability of one half .
  - Knowledge of many cipher texts doesn't give information about the plaintext.
  - A small change in the plaintext X gives a big change in the cipher text C.

- Confusion
  - Complex relationship between key and cipher text .
  - Knowledge of a many cipher texts doesn't give information about the keys.
  - A small change in the key K gives a big change in the cipher text C.

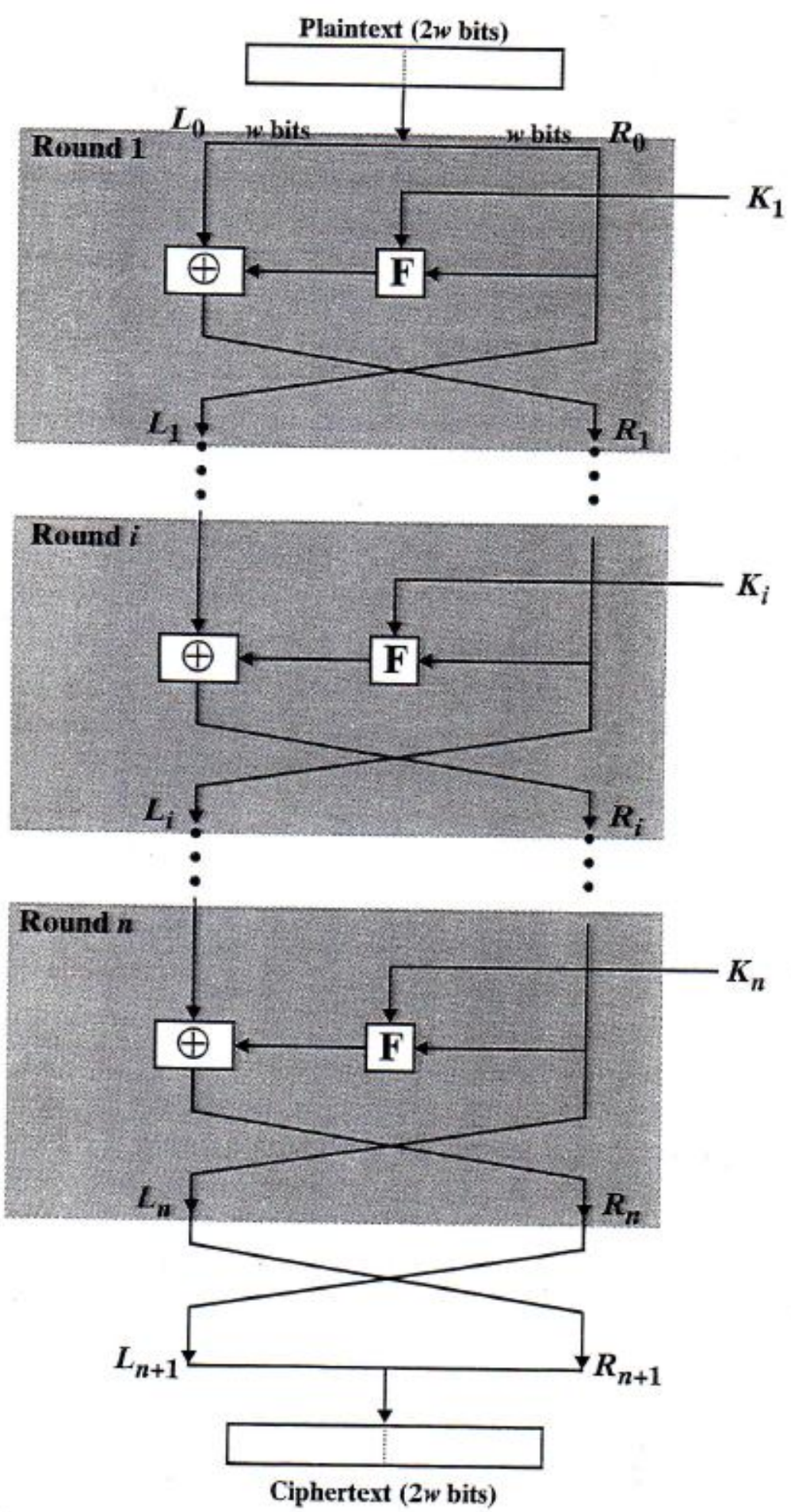
Symmetric Block Ciphers - Algorithm Overview

- Some popular algorithms:
  - Data Encryption Standard (DES)
  - International Data Encryption Algorithm (IDEA)
  - Triple encryption with a block cipher, e.g. Triple-DES

- New standard: Advanced Encryption Standard (AES)
  - Open standardization process with international participation
  - Started in 1997 by call for algorithms
  - In October 2000, one algorithm called *Rijndael* has been proposed for AES
  - AES standard announced in November 2001
  - See also http://www.nist.gov/aes/

The Feistel Cipher

- A sort of a first version of DES.
- Not actually a cipher but more like a prototype of cipher.
- Developed by Horst Feistel (IBM) 1973.
- Subdivides a 64-bit block in two 32-bit blocks.
- The blocks are permuted, added to different keys and put together.


Plaintext (2w bits)
L0
w bits
w bits
R0
Round 1
K1
F
L1
R1
Round i
Ki
F
Li
Ri
Round n
Kn
F
Ln
Rn
Ln+1
Rn+1
Ciphertext (2w bits)

Feistel's recommendation

- The size of the block should be 64 bits.
- The size of the key should be 128 bits.
- The number of rounds should be 16.
- The function F should be sufficiently complicated.

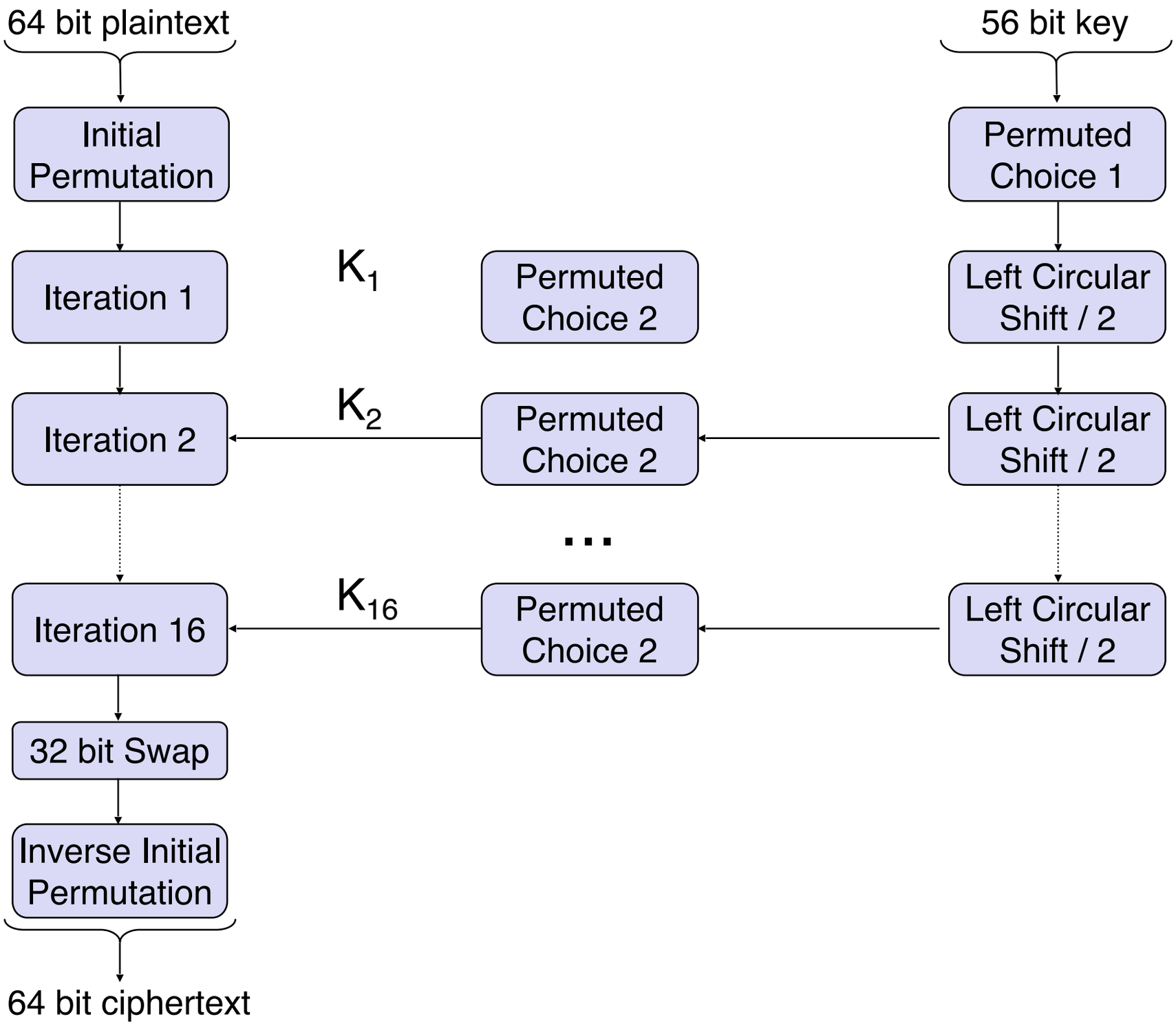
The Data Encryption Standard (DES) – History

- 1973 the National Bureau of Standards (NBS, now National Institute of Standards and Technology, NIST) issued a request for proposals for a national cipher standard, demanding the algorithm to:
  - provide a high level of security,
  - be completely specified and easy to understand,
  - provide security only by its' key and not by its' own secrecy,
  - be available to all users,
  - be adaptable for use in diverse applications,
  - be economically implementable in electronic devices,
  - be efficient to use,
  - be able to be validated, and
  - be exportable.
- None of the submissions to this first call came close to these criteria.
- In response to a second call, IBM submitted its' algorithm LUCIFER, a symmetric block cipher, which works on blocks of length 128 bit using keys of length 128 bit and that was the only promising candidate

DES – History continued

- The NBS requested the help of the National Security Agency (NSA) in evaluating the algorithm's security:
  - The NSA reduced the block size to 64 bit, the size of the key to 56 bit and changed details in the algorithm's *substitution boxes.*
  - Many of the NSA's reasoning for these modifications became clear in the early 1990's, but raised great concern in the late 1970's.
- Despite all criticism the algorithm was adopted as "Data Encryption Standard" in the series of Federal Information Processing Standards in 1977 (FIPS PUB 46) and authorized for use on all unclassified government communications.
- DES has been widely adopted in the years to follow

DES – Algorithm Outline



DES – Single Iteration (1)

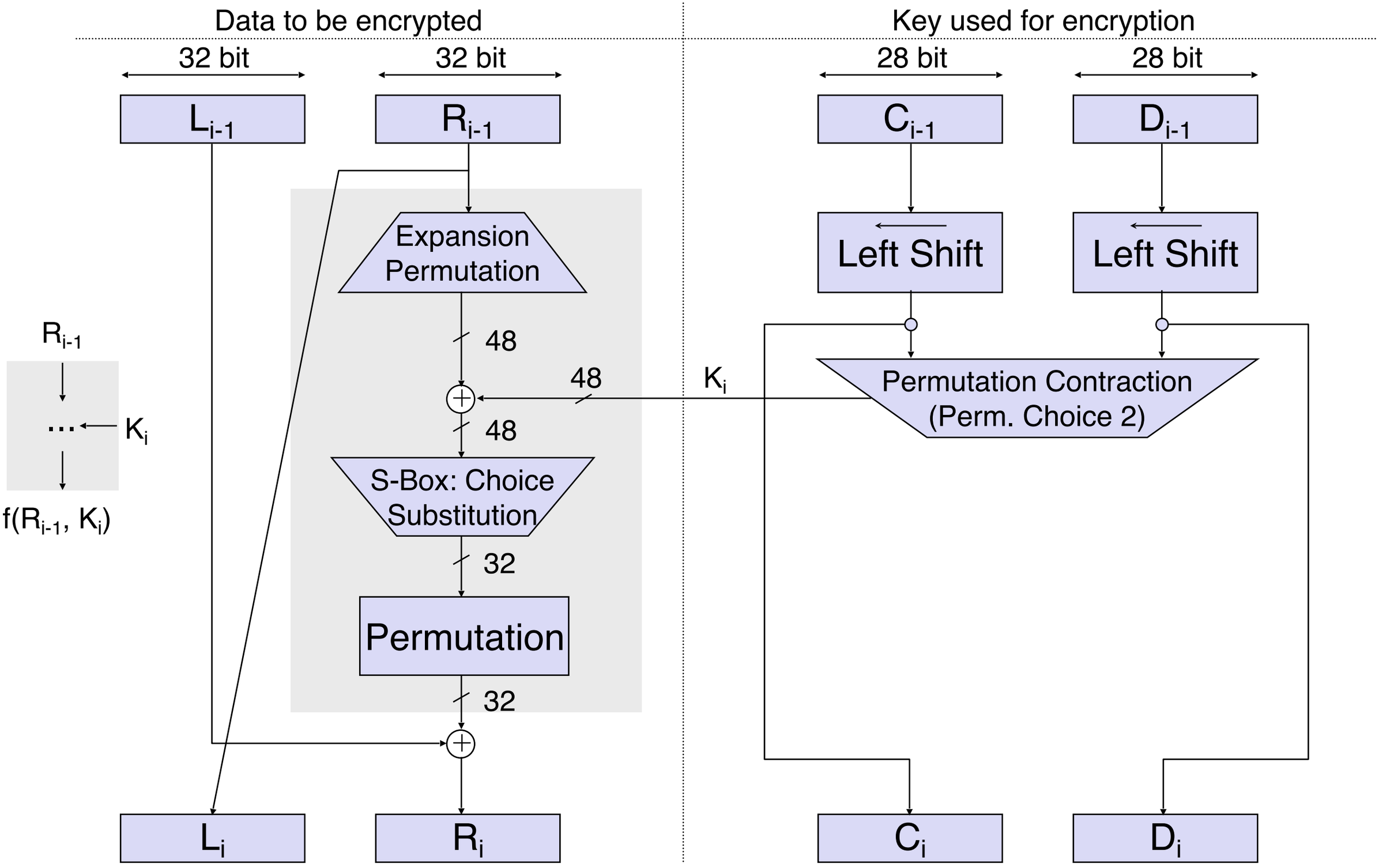
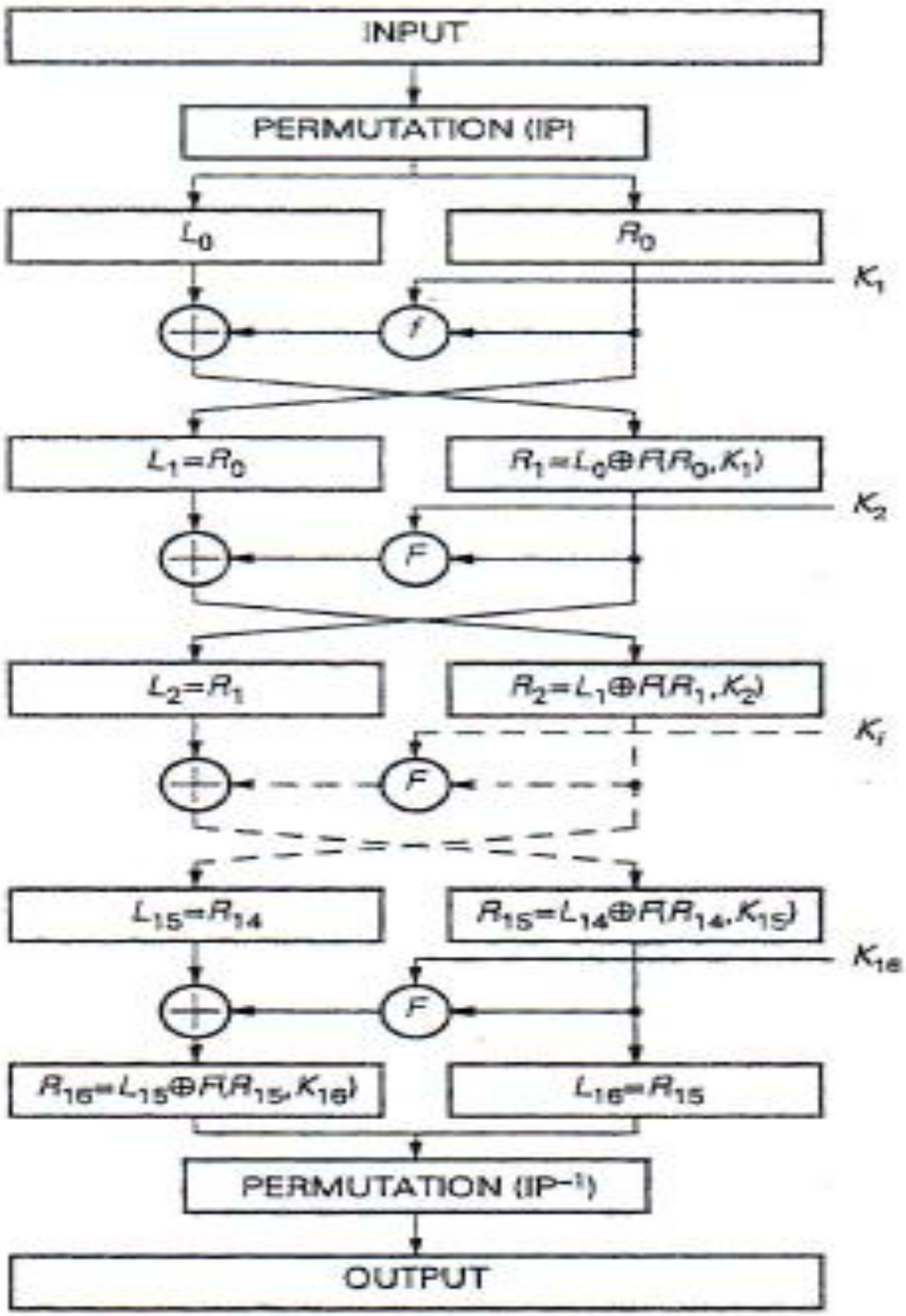


Figure 4.1. The general diagram of the DES algorithm.



R (32 bits)
E
48 bits
K (48 bits)
S1
S2
S3
S4
S5
S6
S7
S8
P
32 bits

Table 4.1. Table for S-box 4.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

DES – Single Iteration (2)

- The right-hand 32 bit of the data to be encrypted are expanded to 48 bit by the use of an expansion / permutation table
- Both the left- and the right-hand 28 bit of the key (also called *subkeys*) are circularly left-shifted and the resulting value is contracted to 48 bit by the use of a permutation / contraction table
- The above two values are XORed and fed into a choice and substitution box:
  - Internally this operation is realized by 8 so-called *s-boxes,* each of them mapping a six bit value to a four bit value according to a box-specific table, altogether leading to a 32 bit output
  - The design of these s-boxes was strengthened by the NSA, which led to intense discussion in the 1970's and was understood in the 1990's after the discovery of *differential cryptanalysis*
- The output of the above step is permuted again and XORed with the left-hand 32 bit of data leading to the new right-hand 32 bit of data
- The new left-hand 32 bit of data are the right-hand value of the previous iteration

DES – Decryption (1)

- Using the abbreviation *f(R, K)* the encryption process can be written as:
  - $L_i = R_{i-1}$
  - $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$
  - This design idea (splitting the data into two halfs and organize encryption according to the above equations) is used in many block ciphers and is the essentila part of a *Feistel*
- The DES decryption process is essentially the same as encryption. It uses the ciphertext as input to the encryption algorithm, but applies the subkeys in reverse order
- So, the initial values are:
  - $L'_0 \,||\, R'_0$ = InitialPermutation(ciphertext)
  - ciphertext = InverseInitialPermutation($R_{16} \,||\, L_{16}$)
  - $L'_0 \,||\, R'_0$ = InitialPermuation(InverseInitialPermutation($R_{16} \,||\, L_{16}$)) = $R_{16} \,||\, L_{16}$
- After one step of decryption:
  - $L'_1 = R'_0 = L_{16} = R_{15}$
  - $R'_1 = L'_0 \oplus f(R'_0, K_{16}) = R_{16} \oplus f(R_{15}, K_{16}) = [L_{15} \oplus f(R_{15}, K_{16})] \oplus f(R_{15}, K_{16}) = L_{15}$

DES – Decryption (2)

- This relationship holds through all the process as:
  - $R_{i-1} = L_i$
  - $L_{i-1} = R_i \oplus f(R_{i-1}, K_i) = R_i \oplus f(L_i, K_i)$
- Finally, the output of the last round is:
  - $L'_{16} \,||\, R'_{16} = R_0 \,||\, L_0$
- After the last round, DES performs a 32-bit swap and the inverse initial permutation:
  - *InverseInitialPermutation*($L_0 \,||\, R_0$) = *InverseInitialPermutation*(*InitialPermutation*(*plaintext*)) = *plaintext*

DES – Security (1)

- Key weaknesses:
  - *Weak keys:* four keys are weak as they generate subkeys with either all 0's or all 1's
  - *Semiweak keys:* there are six pairs of keys, which encrypt plaintext to identical ciphertext as they generate only two different subkeys
  - *Possibly weak keys:* there are 48 keys, which generate only four different subkeys
  - As a whole 64 keys out of 72,057,594,037,927,936 are considered weak
- Algebraic structure:
  - If DES were *closed,* then for every K_1, K_2 there would be a K_3 such that: $E(K_2, E(K_1,M)) = E(K_3, M)$, thus double encryption would be useless
  - DES is not *closed* , thus a multiple encryption scheme might be used to increase the key length (see also below)

DES – Security (2)

- *Differential cryptanalysis:*
  - In 1990 E. Biham and A. Shamir published this method of analysis
  - It looks specifically for differences in ciphertexts whose plaintexts have particular differences and tries to guess the correct key from this
  - The basic approach needs chosen plaintext together with its ciphertext
  - DES with 16 rounds is immune against this attack, as the attack needs 2^{47} chosen plaintexts or (when "converted" to a known plaintext attack) 2^{55} known plaintexts.
  - The designers of DES told in the 1990's that they knew about this kind of attacks in the 1970's and that the s-boxes were designed accordingly
- Key length:
  - As a 56 bit key can be searched in 10.01 hours when being able to perform 10^6 encryptions / µs (which is feasible today), DES can no longer be considered as sufficiently secure

Extending the Key-Length of DES by Multiple Encryption (1)

- Double DES: as DES is not closed, double encryption results in a cipher that uses 112 bit keys:
  - Unfortunately, it can be attacked with an effort of 2^{56}
  - As $C = E(K_2, E(K_1, P))$ we have $X := E(K_1, P) = D(K_2, C)$
  - If an attacker can get one known plaintext / ciphertext pair then he can construct two tables (*meet-in-the-middle-attack*):
    - Table 1 holds the values of X when P is encrypted with all possible values of K
    - Table 2 holds the values of X when C is decrypted with all possible values of K
    - Sort the two tables and construct keys $K_{T1} \,||\, K_{T2}$ for all combinations of entries that yield to the same value
  - As there are 2^{64} possible ciphertext values for any given plaintext that could be produced by Double-DES, there will be on the average $2^{112}/2^{64} = 2^{48}$ false alarms on the first known plaintext / ciphertext pair.
  - Every additional plaintext / ciphertext pair reduces the chance of getting a wrong key by a factor of $1 / 2^{64}$, so with two known blocks the chance is 2^{-16}

Extending the Key-Length of DES by Multiple Encryption (2)

- So, the effort required to break Double DES is on the magnitude of 2^{56}, which is only slightly better than the effort of 2^{55} required to break Single DES with a known plaintext attack and far from the 2^{111} we would expect from cipher with a key length of 112 bit!
- This kind of attack can be circumvented by using a triple encryption scheme, as proposed by W. Tuchman in 1979:
  - $C = E(K_3, D(K_2, E(K_1, P)))$
  - The use of the decryption function *D* in the middle allows to use triple encryption devices with peers that only own single encryption devices by setting $K_1 = K_2 = K_3$
  - Triple encryption can be used with two (set $K_1 = K_3$) or three different keys
  - There are no known practical attacks against this scheme up to now
  - Drawback: the performance is only 1/3 of that of single encryption, so it might be a better idea to use a different cipher, which offers a bigger key-length right away

The Advanced Encryption Standard AES (1)

- Jan. 1997: the *National Institute of Standards and Technology (NIST)* of the USA announces *the AES development* effort.
  - The overall goal is to develop a Federal Information Processing Standard (FIPS) that specifies an encryption algorithm(s) capable of protecting sensitive government information well into the next century.
  - The algorithm(s) is expected to be used by the U.S. Government and, on a voluntary basis, by the private sector.
- Sep. 1997: formal *call for algorithms*, open to everyone on earth
  - AES would specify an unclassified, publicly disclosed encryption algorithm(s), available royalty-free, worldwide.
  - The algorithm(s) must implement symmetric key cryptography as a block cipher and (at a minimum) support block sizes of 128-bits and key sizes of 128-, 192-, and 256-bits.
- Aug. 1998: first AES candidate conference
  - NIST announces the selection of 15 candidate algorithms
  - Demand for public comments

The Advanced Encryption Standard AES (2)

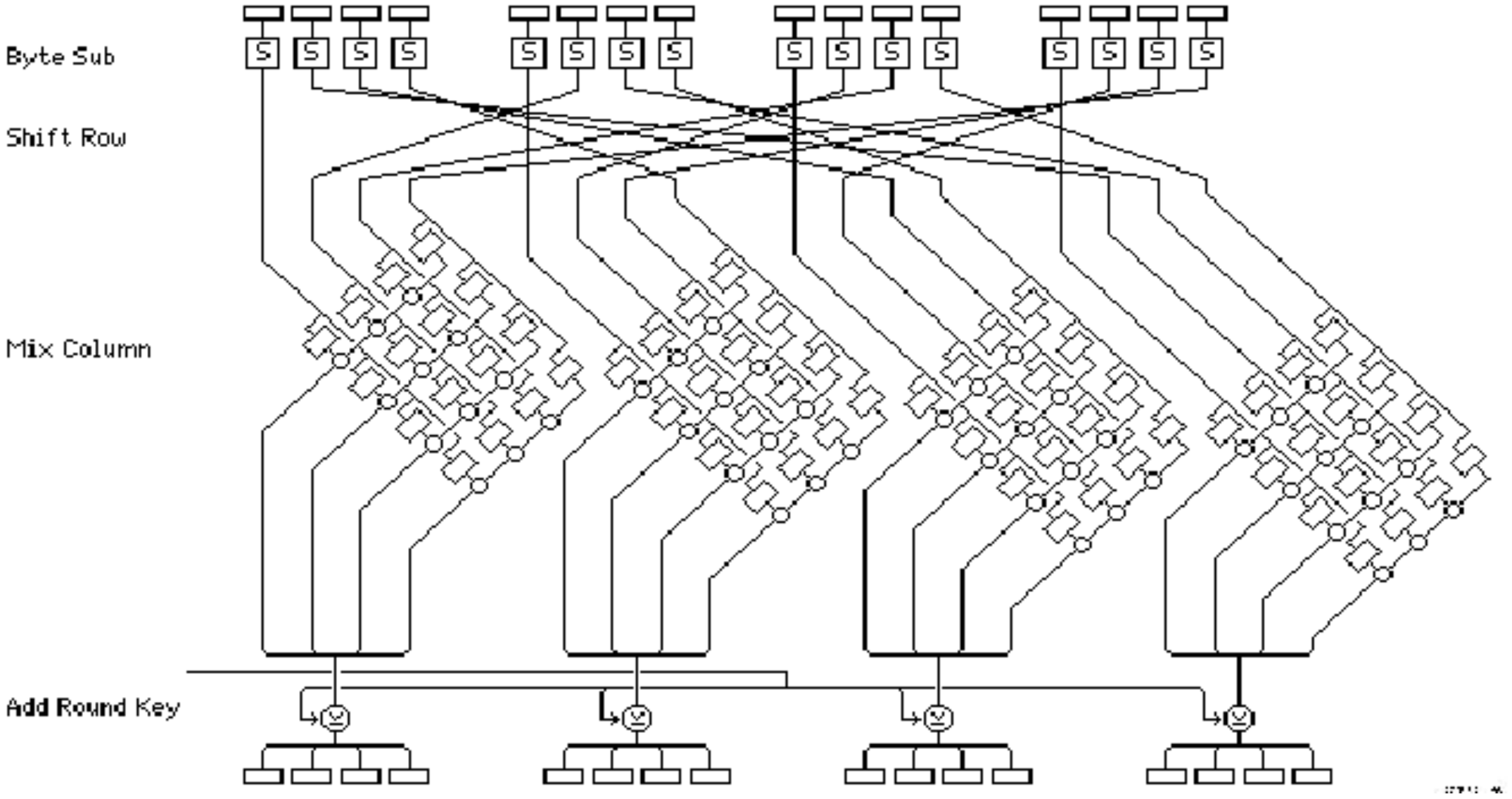
- Mar. 1999: second AES candidate conference
  - Discussion of results of the analysis conducted by the global cryptographic community on the candidate algorithms.
- April 1999:
  - Using the analyses and comments received, NIST selects five algorithms as finalist candidates: *MARS, RC6, Rijndael, Serpent,* and *Twofish*
  - Demand for public comments on any aspect of the finalists:
    - Cryptanalysis
    - Implementation issues
    - Intellectual property & Overall recommendations
- May 2000: third AES candidate conference
- October 2000: Rijndael is announced as NIST's proposal for AES
- 28. February 2001: draft FIPS standard is published [AES01a]
- 29. May 2001: comment period ends
- 26. November 2001: official announcement of the AES standard

The Advanced Encryption Standard AES (3)

- Key and block lengths:
  - Key Length: 128, 192, or 256 bit
  - Block Length: 128, 192, or 256 bit
  - In the following only 128 bit is considered
- The algorithm operates on:
  - state[4, 4]: a byte-array of 4 rows and 4 columns (for 128 bit block size)
  - key[4, 4]: an array of 4 rows and 4 columns (for 128 bit key size)
- Number of rounds: 10 (for block and key size of 128 bit)
  - Rounds 1 - 9 make use of four different operations:
    - ByteSub: a non-linear byte substitution (basically an s-box)
    - ShiftRow: the rows of the state are cyclically shifted by various offsets
    - MixColumn: the columns of state[] are considered as polynomials over GF(2^8) and multiplied modulo $x^4 + 1$ with a fixed polynomial c(x), given by c(x) = ‘03’ x^3 + ‘01’ x^2 + ‘01’ x + ‘02’
    - RoundKey: a round-key is XORed with the state
  - Round 10 does not make use of the MixColumn operation

The Advanced Encryption Standard AES (4)

Structure of one Round in Rijndael



(source: “Rijndael”, a presentation by J. Daemen and V. Rijmen)

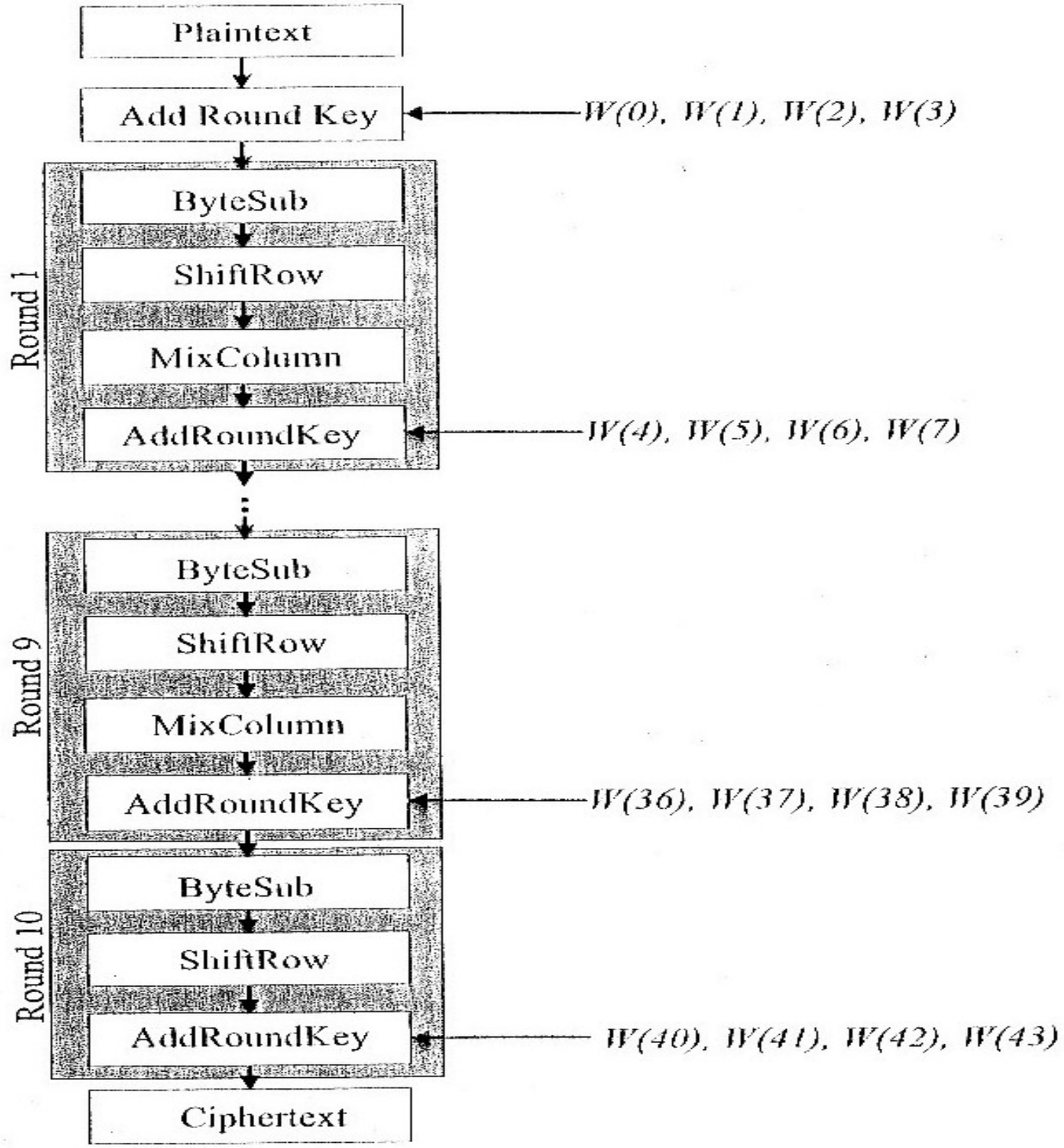


Figure 5.1: The AES-Rijndael Algorithm

The Stream Cipher Algorithm RC4 (1)

- RC4 is a stream cipher that has been invented by Ron Rivest in 1987
- It was proprietary until 1994 when someone posted it anonymously to a mailing list
- RC4 is operated in the output feedback mode (OFB):
  - The encryption algorithm generates a pseudo-random sequence RC4*(IV, K)*, that depends only on the key *K* and an initialization vector *IV*
  - The plaintext P_i is then XORed with the pseudo-random sequence to obtain the ciphertext and vice versa:
    - $C_1 = P_1 \oplus$ RC4(IV_1, K)
    - $P_1 = C_1 \oplus$ RC4(IV_1, K)
  - The pseudo-random sequence is often also called *keystream*
  - It is crucial to the security that keystream is never re-used!!!
    - If keystream is re-used (that is $IV_1 = IV_2$ with the same *K*), then the XOR of two plaintexts can be obtained:

      $C_1 \oplus C_2 = P_1 \oplus$ RC4$(IV, K) \oplus P_2 \oplus$ RC4$(IV, K) = P_1 \oplus P_2$

The Stream Cipher Algorithm RC4 (2)

- RC4 uses a variable length key up to 2048 bit
  - Actually, the key serves as the seed for a pseudo-random-bit-generator
- RC4 works with two 256 byte arrays: S[0,255], K[0,255]
- Step 1: Initialize the arrays

  for (i = 0; i < 256; i++) S[i] = i;// fill array S[] with 0 to 255
  // fill array K[] with the key and IV by repeating them until K[] is filled
  n = 0;
  for (i =0; i < 256; i++) { n = (n + S[i] + K[i]) MOD 256; swap(S[i], S[n]); }
- Step 2: Generate the keystream (after initializing i = 0; n = 0;)

  i = (i + 1) MOD 256; n = (n + S[i]) MOD 256;
  swap(S[i], S[n]);
  t = (S[i] + S[n]) MOD 256;
  Z = S[t]; // Z contains 8 bit of keystream produced by one iteration
- Step 3: XOR the keystream with the plaintext or ciphertext

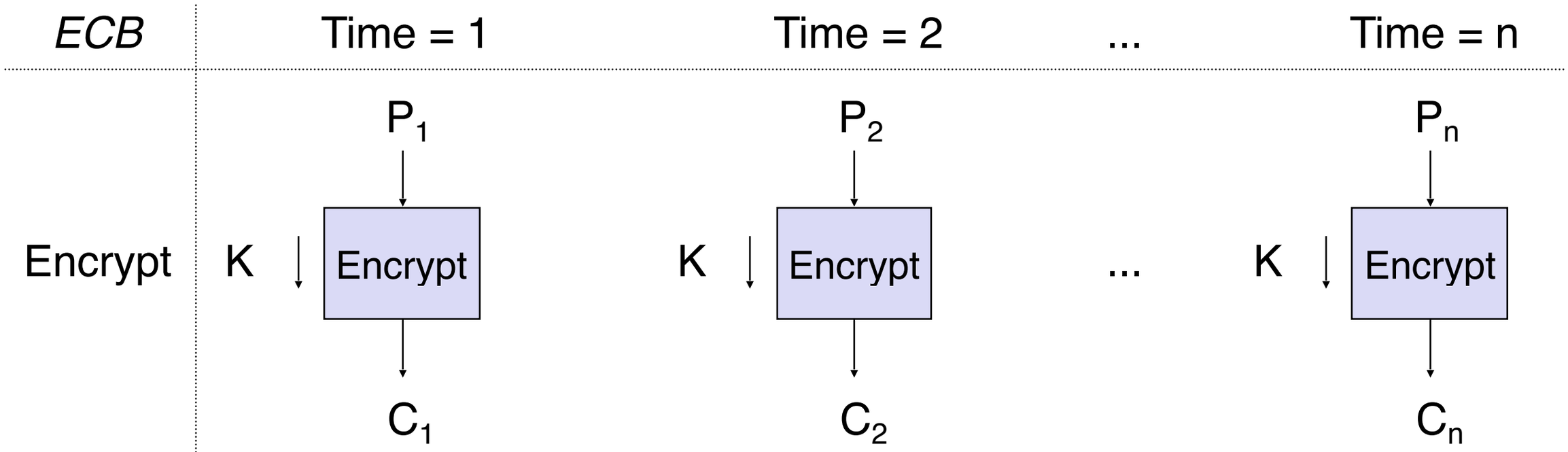
The Stream Cipher Algorithm RC4 (3)

- Security of RC4:
  - Security against brute force attacks (trying every possible key):
    - The variable key length of up to 2048 bit allows to make them impractical (at least with the resources available in our universe)
    - However, by reducing the key length RC4 can also be made arbitrarily insecure!
  - RSA Data Security, Inc. claims that RC4 is immune to differential and linear cryptanalysis, and no small cycles are known
- RC4 with 40 bit keys had special export status, even when other ciphers were not allowed to be exported from the USA
  - Secure Socket Layer (SSL), which has been designed to secure HTTP transfers uses RC4 with 40 bit key length as the default algorithm
  - 40 bit key length is not immune against brute-force attacks
- However, recent results show weaknesses that, depending on the details of the key scheduling method, lead to severe vulnerabilities! [FMS01a, Riv01a, SIR01a]

- General Remarks & Notation:
  - A plaintext *p* is segmented in blocks p_1, p_2, ... each of length *b* or *j*, respectively, where *b* denotes the block size of the encryption algorithm and $j < b$
  - The ciphertext *c* is the combination of c_1, c_2, ... where c_i denotes the result of the encryption of the i^{th} block of the plaintext message
  - The entities encrypting and decrypting a message have agreed upon a key *K.*

Symmetric Block Ciphers - Modes of Encryption 2

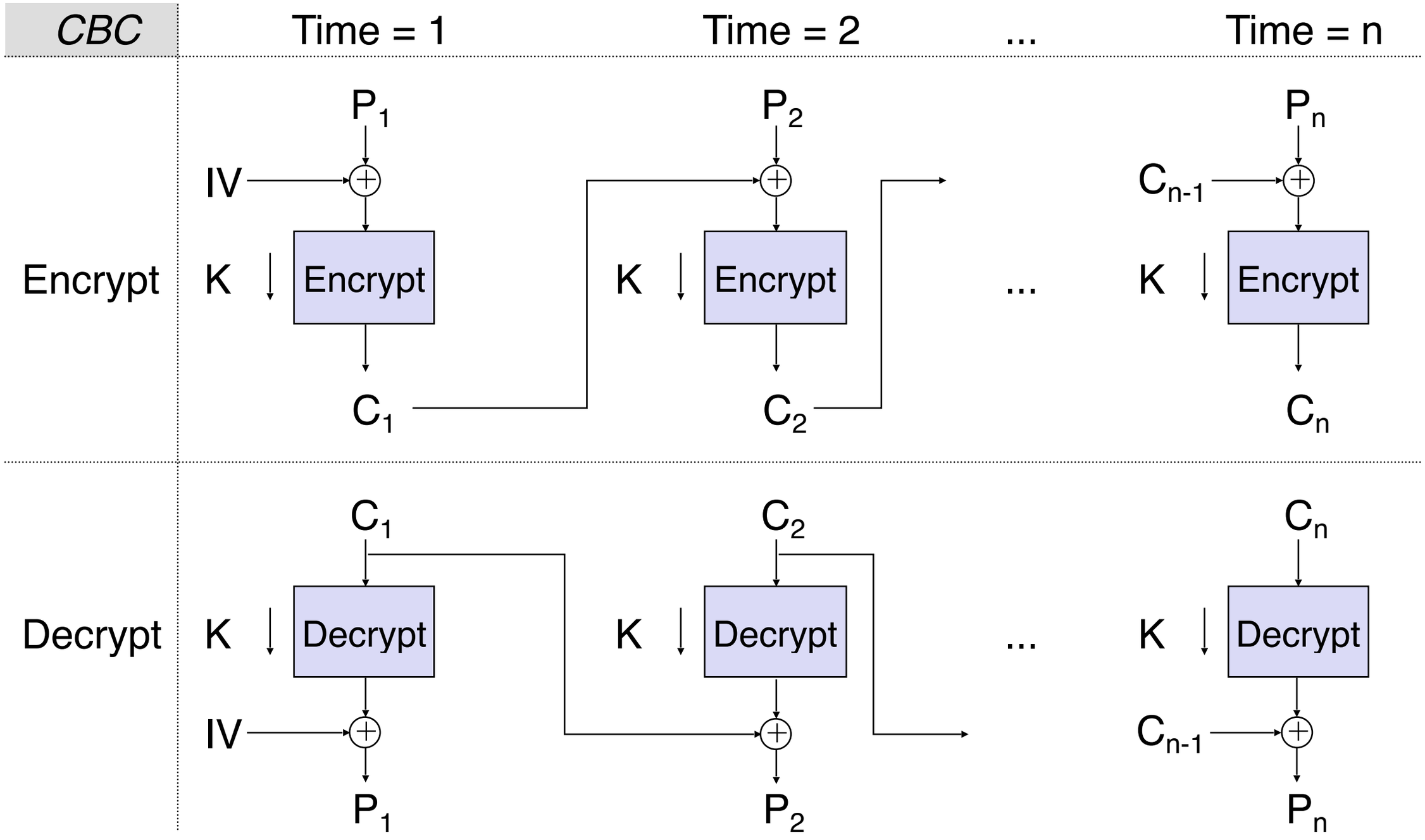
- *Electronic Code Book Mode (ECB):*
  - Every block p_i of length b is encrypted independently: $c_i = E(K, p_i)$
  - A bit error in one ciphertext block c_i results in a completely wrongly recovered plaintext block $p_i´$
  - Loss of synchronization does not have any effect if integer multiples of the block size b are lost.
    If any other number of bits are lost, explicit re-synchronization is needed.
  - Drawback: identical plaintext blocks are encrypted to identical ciphertext!



Symmetric Block Ciphers - Modes of Encryption 3

- *Cipher Block Chaining Mode (CBC):*
  - Before encrypting a plaintext block p_i it is XORed ($\oplus$) with the preceding ciphertext block c_{i-1}:
    - $c_i = E(K, c_{i-1} \oplus p_i)$
    - $p_i' = c_{i-1} \oplus D(K, c_i)$
  - In order to compute c_1 both parties agree on an *initial value (IV)* for c_0
- Properties:
  - Error propagation:
    - A distorted ciphertext block results in two distorted plaintext blocks, as p_i' is computed using c_{i-1} and c_i
  - Synchronisation:
    - If the number of lost bits is a multiple integer of *b,* one additional block p_{i+1} is distorted before synchronization is re-established.
      If any other number of bits are lost explicit re-synchronization is needed.
  - Advantage: identical plaintext blocks are encrypted to non-identical ciphertext.

Symmetric Block Ciphers - Modes of Encryption 4



- *Ciphertext Feedback Mode (CFB):*
  - A block encryption algorithm working on blocks of size *b* can be converted to an algorithm working on blocks of size *j (j<b):*
    - Let: *S(j, x)* denote the *j* higher significant bits of *x*
      P_i, C_i denote the i^{th} block of plain- and ciphertext of length *j*
      IV be an initial value both parties have agreed upon

    then :

$$R_1 = IV$$

$$R_n = \left(R_{n-1} \cdot 2^j \bmod 2^b\right) \oplus C_{n-1} \qquad \text{// j-bit left shift and XOR with old ciphertext}$$

$$C_n = S\left(j, E_K\left(R_n\right)\right) \oplus P_n$$

$$S\left(j, E_K\left(R_n\right)\right) \oplus C_n = S\left(j, E_K\left(R_n\right)\right) \oplus S\left(j, E_K\left(R_n\right)\right) \oplus P_n$$

$$S\left(j, E_K\left(R_n\right)\right) \oplus C_n = P_n$$

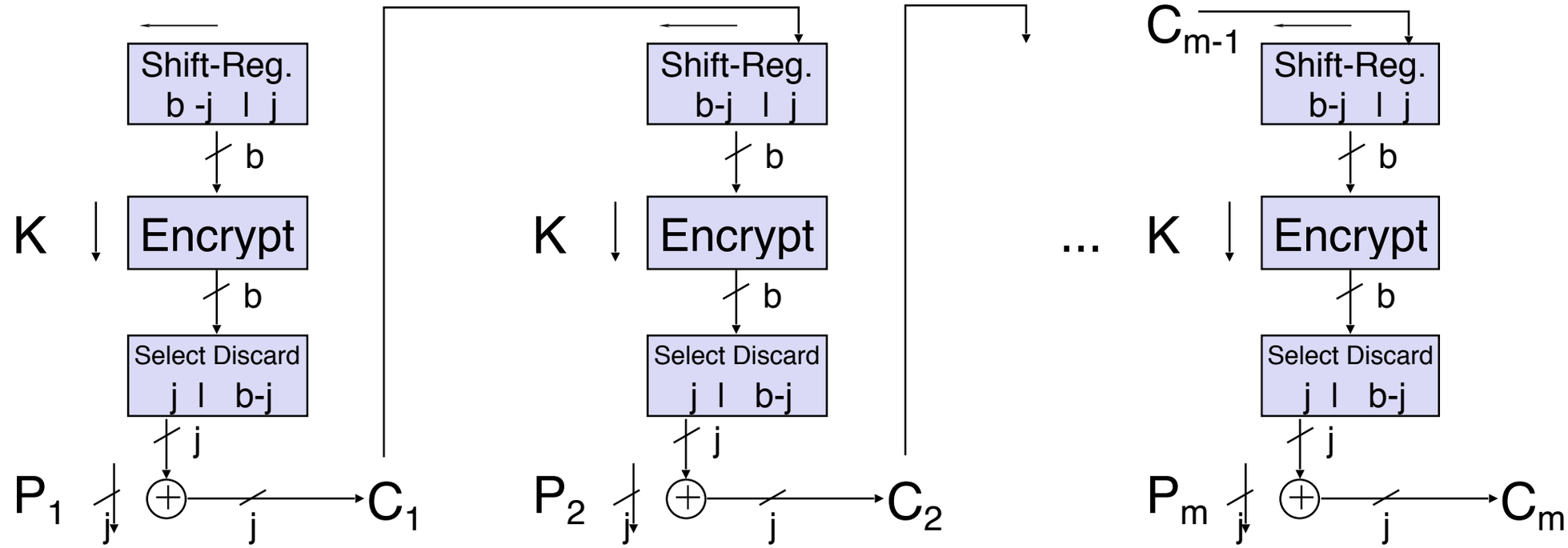
  - A current value of *j* is 8 for encryption of one character per step

Symmetric Block Ciphers - Modes of Encryption 6

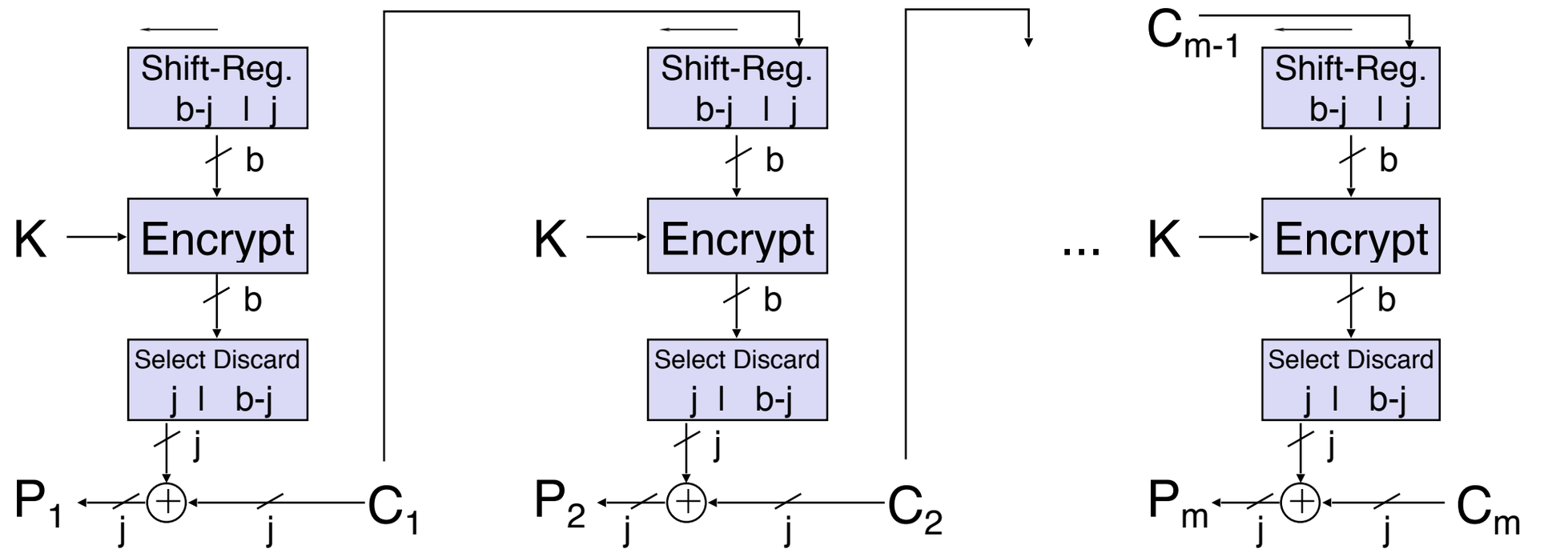
CFB

Time = 1 | Time = 2 | ... | Time = m

Encrypt



Decrypt



- Properties of CFB:
  - Error propagation:
    - As the ciphertext blocks are shifted through the register step by step, an erroneous block c_i distorts the recovered plaintext block p_i´ as well as the following $\lceil b / j \rceil$ blocks
  - Synchronisation:
    - If the number of lost bits is a multiple integer of j then $\lceil b / j \rceil$ additional blocks are distorted before synchronization is re-established.
      If any other number of bits are lost explicit re-synchronization is needed.

- Drawback:
  - The encryption function E needs to be computed more often, as one encryption of b bit has to be performed to conceal j bit of plaintext
  - Example: Use of DES with encryption of one character at a time:
    ⇒ encryption has to be performed 8 times more often

- *Output Feedback Mode (OFB):*
  - The block encryption algorithm is used to generate a pseudo-random sequence R_i, that depends only on *K* and *IV:*
    - Let: *S(j, x)* denote the *j* higher significant bits of *x*
      P_i, C_i denote the i^{th} block of plain- and ciphertext of length *j*
      IV be an initial value both parties have agreed upon
      then :

$$R_1 = IV$$

$$R_n = \left(R_{n-1} \cdot 2^j \bmod 2^b\right) \oplus S\left(j, E_K\left(R_{n-1}\right)\right)$$ // j-bit left shift + encrypted old value

$$C_n = S\left(j, E_K\left(R_n\right)\right) \oplus P_n$$

$$S\left(j, E_K\left(R_n\right)\right) \oplus C_n = S\left(j, E_K\left(R_n\right)\right) \oplus S\left(j, E_K\left(R_n\right)\right) \oplus P_n$$

$$S\left(j, E_K\left(R_n\right)\right) \oplus C_n = P_n$$

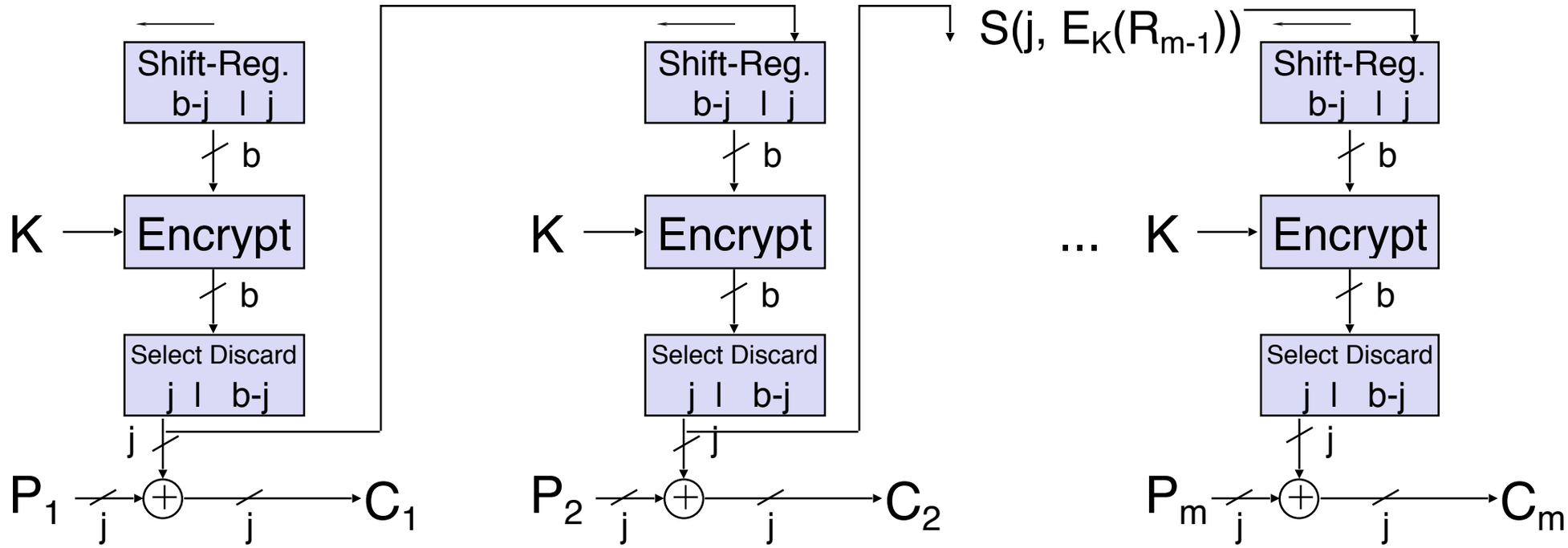
- The plaintext is XORed with the pseudo-random sequence to obtain the ciphertext and vice versa

Symmetric Block Ciphers - Modes of Encryption 9

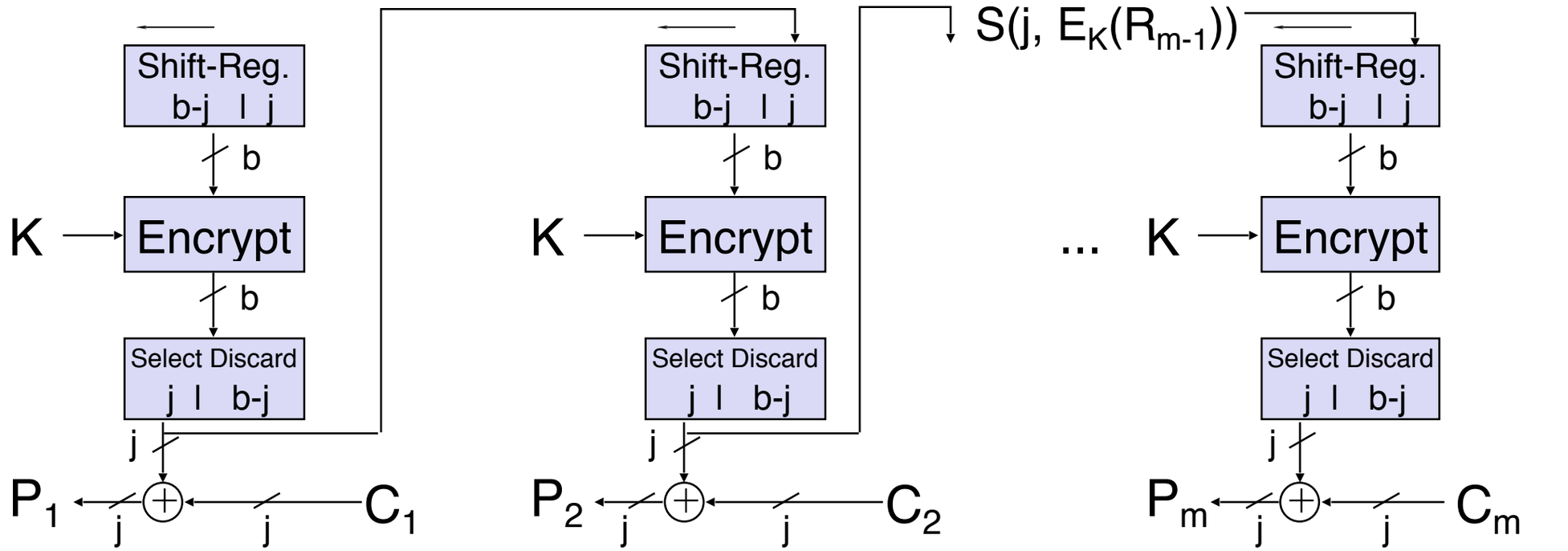
OFB

Time = 1 Time = 2 ... Time = m

Encrypt



Decrypt



- Properties of OFB:
  - Error propagation:
    - Single bit errors result only in single bit errors ⇒ no error multiplication
  - Synchronisation:
    - If some bits are lost explicit re-synchronization is needed
  - Advantage:
    - The pseudo-random sequence can be pre-computed in order to keep the impact of encryption to the end-to-end delay low
  - Drawbacks:
    - Like with CFB the encryption function *E* needs to be computed more often, as one encryption of *b* bit has to be performed to conceal *j* bit of plaintext
    - It is possible for an attacker to manipulate specific bits of the plaintext

Cryptology – Some Historic Remarks 1

- 400 BC: The Spartans employ a cipher device called *scytale* for communications between military commanders.
  - The scytale consisted of a tapered baton, around which was spirally wrapped a strip of parchment or leather on which the message was written
  - When unwrapped, the letters were scrambled in order and formed the cipher
  - When the strip was wrapped around another baton of identical proportions to the original, the plaintext reappeared
- During 4. century BC:
  - Aeneas Tacticus (Greek) writes *"On the defense of fortifications"*, with one chapter devoted to cryptography
  - Polybius (Greek) invents a means of encoding letters into pairs of symbols by a device called the *Polybius Checkerboard* which realizes a bi-literal substitution and presages many elements of later cryptosystems

Cryptology – Some Historic Remarks 2

- The Romans used monoalphabetic substitution with simple cyclic displacement of the alphabet:
  - *Julius Caesar* employed a shift of three letters (A giving D, ..., Z giving C)
  - *Augustus Caesar* employed a single shift (A giving B, ...)
- The Arabs were the first people to understand the principles of cryptography and to discover the beginnings of cryptanalysis:
  - Design and use of substitution and transposition ciphers
  - Discovery of the use of letter frequency distributions and probable plaintext in cryptanalysis
  - By 1412 AD *Al-Kalka-Shandi* includes an elementary and respectable treatment of several cryptographic systems and their cryptanalysis in his encyclopaedia *Subh al-a'sha*
- European Cryptography:
  - Development started in the Papal States and the Italian city-states in the middle age
  - First ciphers used only vowel substitution

Cryptology – Some Historic Remarks 3

- European Cryptography: (cont.)
  - 1397: *Gabriele de Lavinde* of Parma writes first European manual on cryptography, containing a compilation of ciphers as well as a set of keys for 24 correspondents and embracing symbols for letters, numbers and several two-character code equivalents for words and names
  - Code vocabularies, called *Nomenclators* became the mainstay for several centuries for diplomatic communications of most European governments
  - 1470: *Leon Battista Alberti* publishes *Trattati In Cifra*, which describes the first cipher disk and already prescribes to regularly reset the disk, conceiving the notion of polyalphabeticity
  - 1563: *Giambattista della Porta* provides a modified form of a square table and the earliest example of a digraphic cipher (2-letter-substitution)
  - 1586: *Blaise de Vigenère* publishes *Traicté des chiffres*
  - By 1860 large codes were used for diplomatic communications and ciphers were only used in military communications (except high command level) because of the difficulty of protecting codebooks in the field

Cryptology – Some Historic Remarks 4

- Developments during World Wars 1 and 2:
  - During World War 1: cipher systems were mostly used for tactical communications and high level communication was protected using codes
  - 1920: The communication needs of telecommunications and the maturing of electromechanical technology bring about a true revolution in cryptodevices - the development of *rotor cipher machines:*
    - The rotor principle is discovered independently by *E. E. Hebern* (USA), *H. A. Koch* (Netherlands) and *A. Scherbius* (Germany)
    - Rotor cipher machines cascade a collection of cipher disks to realize polyalphabetic substitution of high complexity
    - Cryptanalysis of tactical communications plays a very important role during World War 2 with the greatest triumphs being the British and Polish solution of the German *Enigma* and two teleprinter ciphers and the American cryptanalysis of Japanese ciphers

Cryptology – Some Historic Remarks 5

- Developments after World War 2:
  - Modern electronics allow even more complex ciphers, initially following the rotor principles (and including their weaknesses)
  - Most information about electronic cipher machines used by various national cryptologic services is not publicly available
  - By the end of the 1960's commercially available cryptography was poorly understood and strong cryptography was reserved for national agencies
  - 1973-1977: Development of the *Data Encryption Standard (DES)*
  - 1976-1978: Discovery of Public Key Cryptography
    - 1976: *W. Diffie* and *M. Hellman* publish "New Directions in Cryptography" introducing the concepts of public key cryptography and describing a scheme of exchanging keys over insecure channels
    - *R. Merkle* independently discovers the public key principle, but his first publications appear 1978, due to a slow publishing process
    - 1978: *R. L. Rivest*, *A. Shamir* and *A. M. Adleman* publish "A Method for Obtaining Digital Signatures and Public Key Cryptosystems", containing the first working and secure public key algorithm *RSA*