Chair for Network Architectures and Services
Department of Informatics
TU München – Prof. Carle

Network Security

Chapter 2
Basic Building Blocks
2.1 Symmetric Cryptography

Acknowledgments

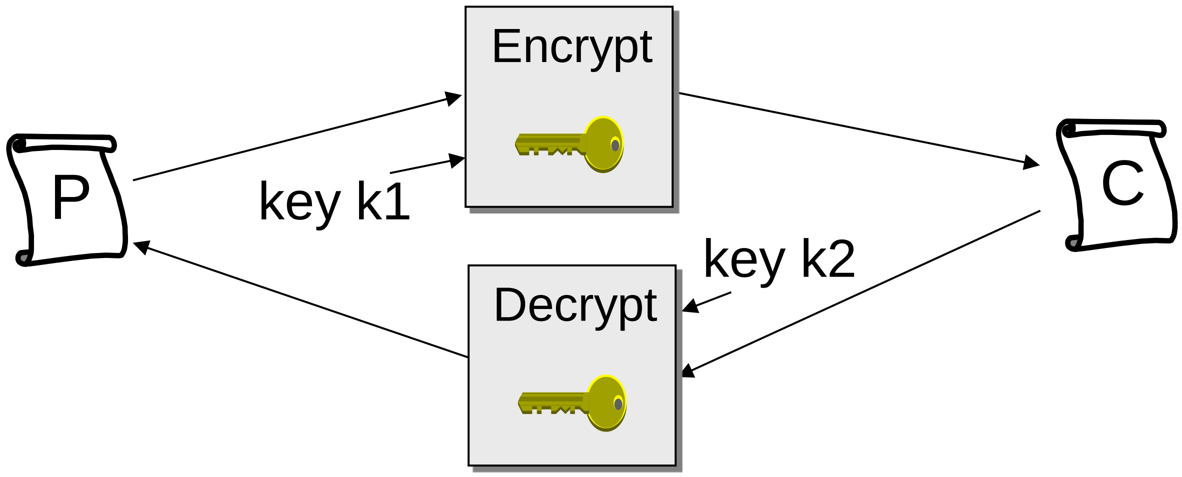
This course is based to a significant extend on slides provided by Günter Schäfer, author of the **book "Netzsicherheit - Algorithmische Grundlagen und Protokolle"**, available in German from **dpunkt Verlag**. The English version of the book is entitled “Security in Fixed and Wireless Networks: An Introduction to Securing Data Communications” and is published by Wiley is also available. We gratefully acknowledge his support.

Some slides on cryptography have been contributed by Pavel Laskov. We gratefully acknowledge his support.

The slides by Günter Schäfer have been partially reworked by Cornelius Diekmann, Heiko Niedermayer, Ali Fessi, Ralph Holz and Georg Carle.

Basic Terms: Plaintext and Ciphertext

- Plaintext P
  - The original readable content of a message (or data).
  - P = „This is network security“
- Ciphertext C
  - The encrypted version of the plaintext.
  - C = „ad 5c 66 d3 55 be 00 88 8c 82 41 d2 75 3d 93 da fe d0 12 20 ac c1 2c e6 64 60 b4 82 2c 87 03 b2 “



In case of symmetric cryptography, k1 = k2.

Cryptographic algorithms: overview

- *Encryption*: transforms plaintext data into ciphertext
  - $C = Enc_K(P)$
  - $P = Dec_K(C)$
  - The identity: $P = Dec_K(Enc_K(P))$

- *Signing*: computes a *check value* or *digital signature*
  - Verifies that the signed data was not tampered with
  - Integrity

- Categories of cryptographic algorithms:
  - *Symmetric cryptography* using 1 key for en-/decryption
  - *Asymmetric cryptography* using 2 different keys for en-/decryption
  - *Cryptographic hash functions* using 0
  - *Message Authentication Codes* using 1 key for signing and verification

Basic Terms: Block cipher and Stream cipher

- Both ciphers require a symmetric key *k*

- Block cipher
  - A cipher that encrypts / decrypts inputs of length *n* to outputs of length *n*
  - Block length *n*
- Stream cipher
  - Generates a random bitstream, called *key stream*
  - Ciphertext = key stream $\oplus$ plaintext

Basic Terms: Block cipher and Stream cipher

Ponies indicate that this slide is intended for your personal postprocessing at home.

- Both ciphers require a symmetric key *k*
- Block cipher
  - A cipher that encrypts / decrypts inputs of length *n* to outputs of length *n*
  - Block length *n*
  - Many modern symmetric ciphers are block ciphers
    e.g. AES, DES, Twofish, …
  - For example, AES 128 uses a block length of 128 bit
- Stream cipher
  - Generates a random bitstream, called *key stream* from the key *k*
  - Ciphertext = key stream ⊕ plaintext
  - ⊕ denotes the XOR operation
  - Popular stream cipher: RC4 (which is no longer considered secure)

Attacking cryptography (1): brute force attack

- Given: *C*
- Unknown: *P, K*

- *brute force attack:* try all keys until an intelligible plaintext is found
- 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

1 encryption / µs: 100 Clock cycles of a 100 MHz processor
10^6 encryptions / µs: Clock cycles using 500 parallel 2GHz processors

Attacking cryptography (2): 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 (3 GHz computer)	$\approx 1 * 10^{17}$
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	$\approx 8.37 * 10^{77}$

Attacking cryptography (3): Cryptanalysis

- Definition: *Cryptanalysis* is the process of attempting to discover the plaintext
  and / or the key

- Types of cryptanalysis:
  - *Ciphertext only:* hope that specific patterns of the plaintext have remained in the ciphertext (frequencies of letters, digraphs, etc.)
  - *Known ciphertext / plaintext pairs*
  - *Chosen plaintext* or *chosen ciphertext*
  - *Differential cryptanalysis*, *linear cryptanalysis*

2.1 Symmetric Cryptography

A perfect symmetric cipher: One-Time-Pad

- Assumption: Alice and Bob share a perfectly random bitstream *otp*.

- Key = *otp*

- Encryption:
  - $C = P \oplus otp$
- Decryption:
  - $P = C \oplus otp$

- Requirement
  - *Key must have same size as message.*
  - *Key must only be used once.*

- Cryptanalysis for One-Time-Pad
  - Ciphertext only: No attack possible as *any possible plaintext* can be generated with the ciphertext.
  - Pairs of ciphertext and plaintext don’t help
  - The ciphertext is perfectly random

Real-world ciphers

- Strengths of otp
  - C of length *n* can be decrypted to any P of length *n*
  - Only knowledge of *k* reveals the right P
  - otp is a **perfect cipher**
- Drawbacks of otp
  - length(key) = length(message)
  - Usually length(key) << length(message)
  - Key must only be used once
- Real- world ciphers
  - Key *k* of fixed length
  - Key *k* is reused for several messages
- Implications
  - The number of possible decryptions of C is smaller

Brute Force attacks on non-perfect ciphers

- Message of length m
  - → 2^m possible messages
- Key of length k
  - → 2^k possible keys
  - → 2^k possible decryptions of message
- Usually: $k \ll m$

- Brute Force: Ciphertext only
  - → if the decryption is intelligible, with high probability *k* is found

- Further advantages for the attacker
  - no perfect randomness of C

- The attacker might be able to break non-perfect ciphers
  - or at least find the most likely plaintext and key

Basic cryptographic Principles

- Substitution
  - Individual characters are exchanged by other characters

  Types of substitution
  - simple substitution: operates on single letters
  - polygraphic substitution: operates on larger groups of letters
  - monoalphabetic substitution: uses fixed substitution over the entire message
  - polyalphabetic substitution: uses different substitutions at different sections of a message

- Transposition
  - The position of individual characters changes (Permutation)

Transposition: scytale

- Known as early as 7th century BC
- Principle:
  - Wrap parchment strip over a wooden rod of a fixed diameter and write letters along the rod.
  - Unwrap a strip and “transmit”
  - To decrypt, wrap a received over a wooden rod of the same diameter and read off the text.

  → Key is the diameter of the rod

- Example:

```
troops
headii
nthewe             thnsm predd opoah nrlod eeeis iedus
stneed      ⇨
moresu
pplies
```

- Weakness:
  - Easy to break by finding a suitable matrix transposition.

Monoalphabetic substitution: Atbash

Jeremiah 25:25

And all the kings of the north, far and near, one with another, and all the kingdoms of the world, which are upon the face of the earth: and the king of Sheshach shall drink after them.

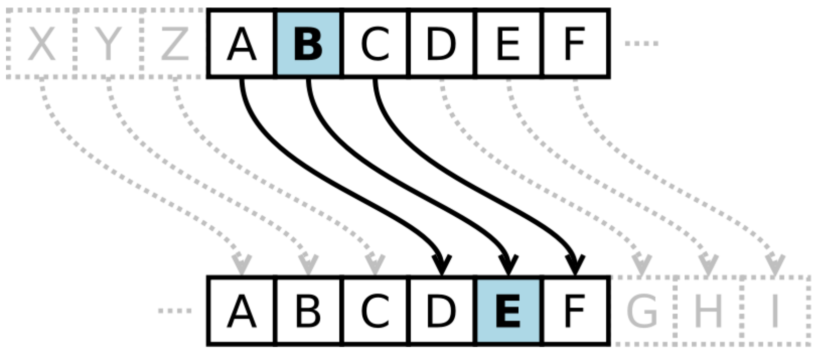
Atbash code: reversed Hebrew alphabet.

A Aleph א	B Beth ב	G Gimel ג	D Daleth ד	H He ה	WVFY Waw ו	Z Zajin ז	H Chet ח	T Tet ט	IJ Jod י	K Kaph כ ך	L Lamed ל	M Mem מ ם	N Nun נ ן	X Samech ס	O Ajin ע	P Pe פ ף	Z Sade צ ץ	Q Koph ק	R Resch ר	S Sin ש	T Taw ת
T Taw ת	S Sin ש	R Resch ר	Q Koph ק	Z Sade צ ץ	P Pe פ ף	O Ajin ע	X Samech ס	N Nun נ ן	M Mem מ ם	L Lamed ל	K Kaph כ ך	IJ Jod י	T Tet ט	H Chet ח	Z Zajin ז	WVFY Waw ו	H He ה	D Daleth ד	G Gimel ג	B Beth ב	A Aleph א

Sheshach ⇨ כ ך ש ש ⇨ ל ב ב ⇨ Babel

Monoalphabetic substitution: Caesar cipher

- Caesar code: left shift of alphabet by 3 positions.



- Example (letter of Cicero to Caesar):

  MDEHV RSNQNRQNV PHDH XHVXNPRQNZP

  HABES OPINIONIS MEAE TESTIMONIUM

- Weakness: a limited number of possible substitutions. Easy to break by brute force!

Now on the bit level

Plaintext	P = 1 1 0 1 1 0 0 1
Key	k = 1 0 0 1
Ciphertext	C = ?

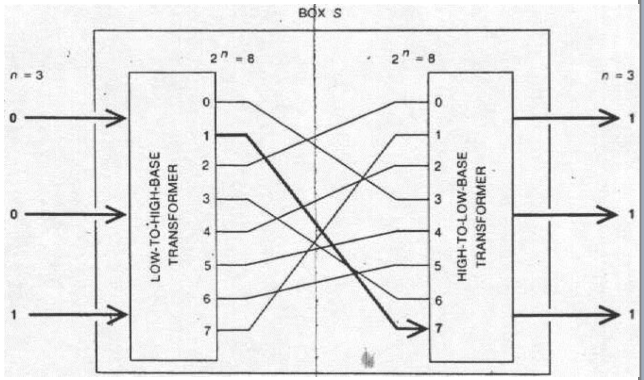
$C = P \oplus k = 11011001 \oplus 10011001 = 01000000$

Why is this not as secure as the OTP?

Modern cryptography: S and P-boxes

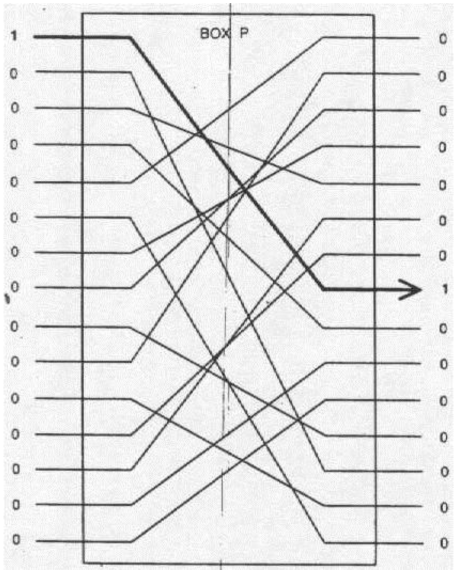
S-box:

- Block-wise **substitution** of binary digits.
  - Can be static or depend on key
  - Input and output size can be different
  - Can be implemented as a large table with all inputs and their predefined outputs
- Resistant to attacks for sufficiently large block size; e.g. for n=128 it provides 2^{128} possible mappings.



P-box:

- Block-wise **permutation** of binary digits.
- Realizes a simple **transposition** cipher with maximal entropy.
- Problem: straightforward attacks exist.

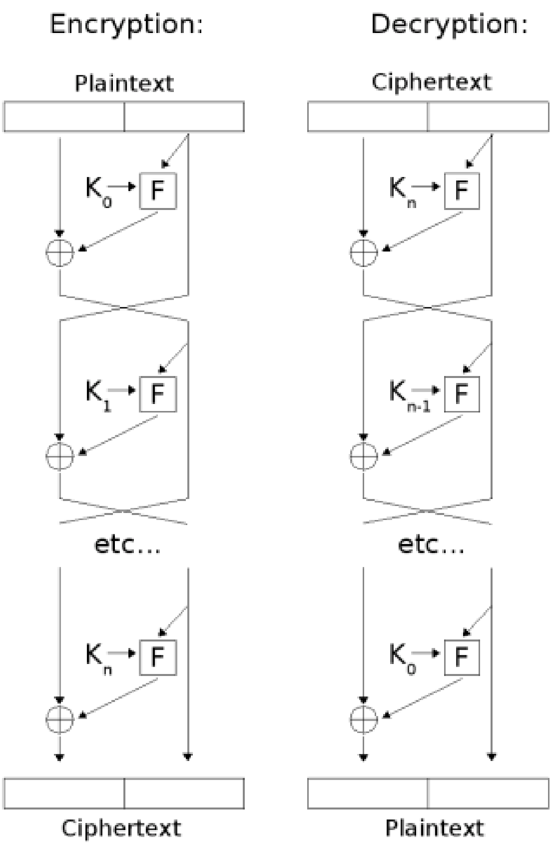


A product cipher of S and P-boxes

- A product cipher is a combination of simple ciphers (e.g. S-box and P-box).
- Rounds: This combination may be applied multiple times.
- Multiple rounds provide a cryptographically strong polyalphabetic substitution.
- Combination of substitution with transposition provides protection against specific attacks (frequency analysis).
- Follows the theoretical principles outlined by C. Shannon in 1949: combines “confusion” with “diffusion” to attain maximal entropy of a cipher text.
  - *Confusion*: cipher text statistics depend in a very complex way on plaintext statistics (approach: substitution in different rounds)
    - e.g. make the number of 1s and 0s in ciphertext seem independent of their numbers in plaintext
  - *Diffusion*: each digit in plaintext and in key influence many digits of cipher text (approach: many rounds with transposition)

Feistel ciphers (Feistel network)

- A multiple-round scheme with separate keys per round.
- Goal: Encrypt plaintext block $P = L_0 \mid R_0$
- Function $f(K_i,R_{i-1})$ is algorithm-specific, usually a combination of permutations and substitutions.
- Invertible via a reverse order of rounds.
- 3 rounds suffice to achieve a pseudorandom permutation.
- 4 rounds suffice to achieve a strong pseudorandom permutation (i.e. it remains pseudorandom to an attacker with an oracle access to its inverse permutation).
- A foundation for a large number of modern symmetric ciphers: DES, Lucifer, Blowfish, RC5, Twofish, etc.



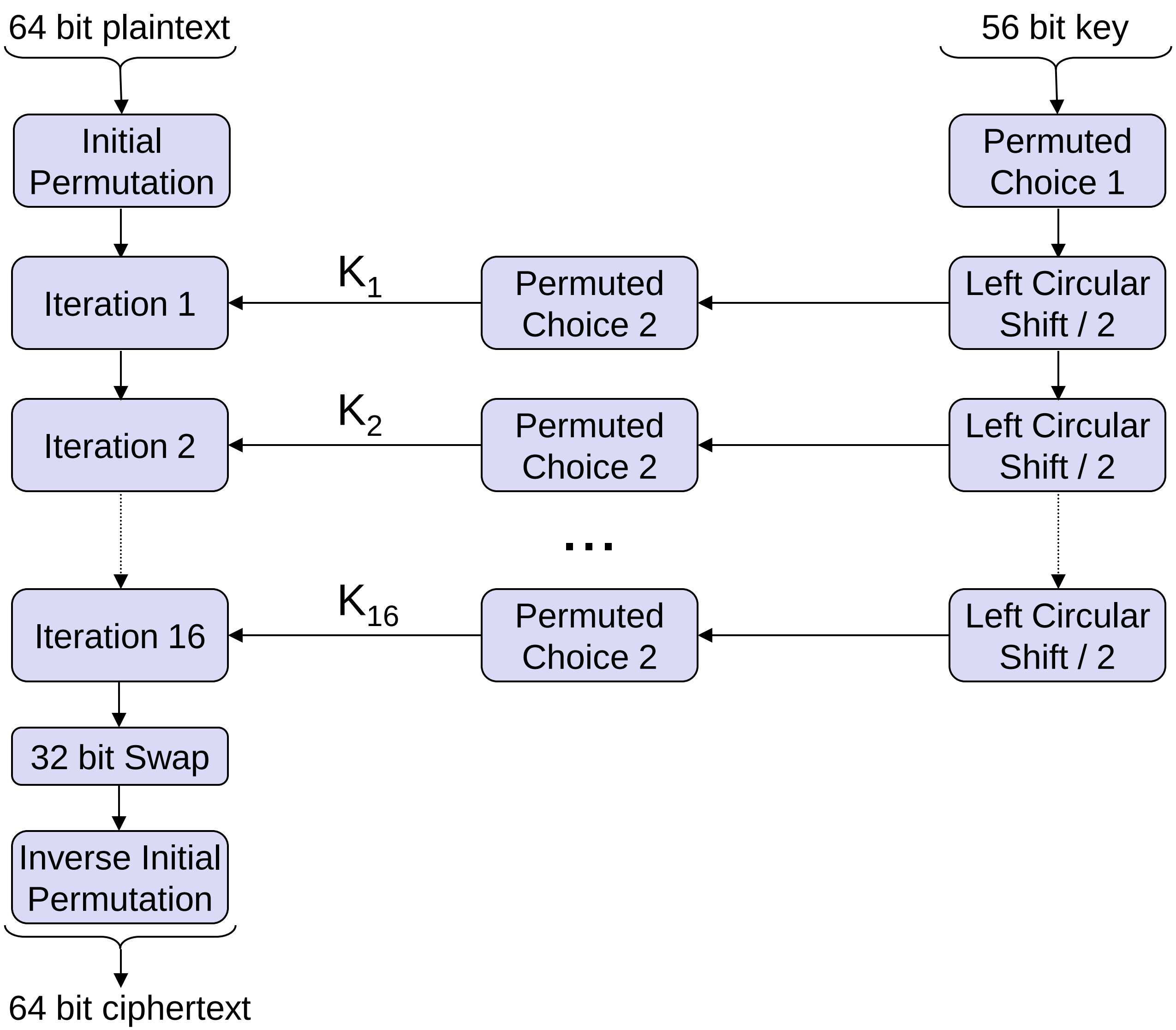
Feistel Cipher

Data Encryption Standard (DES)

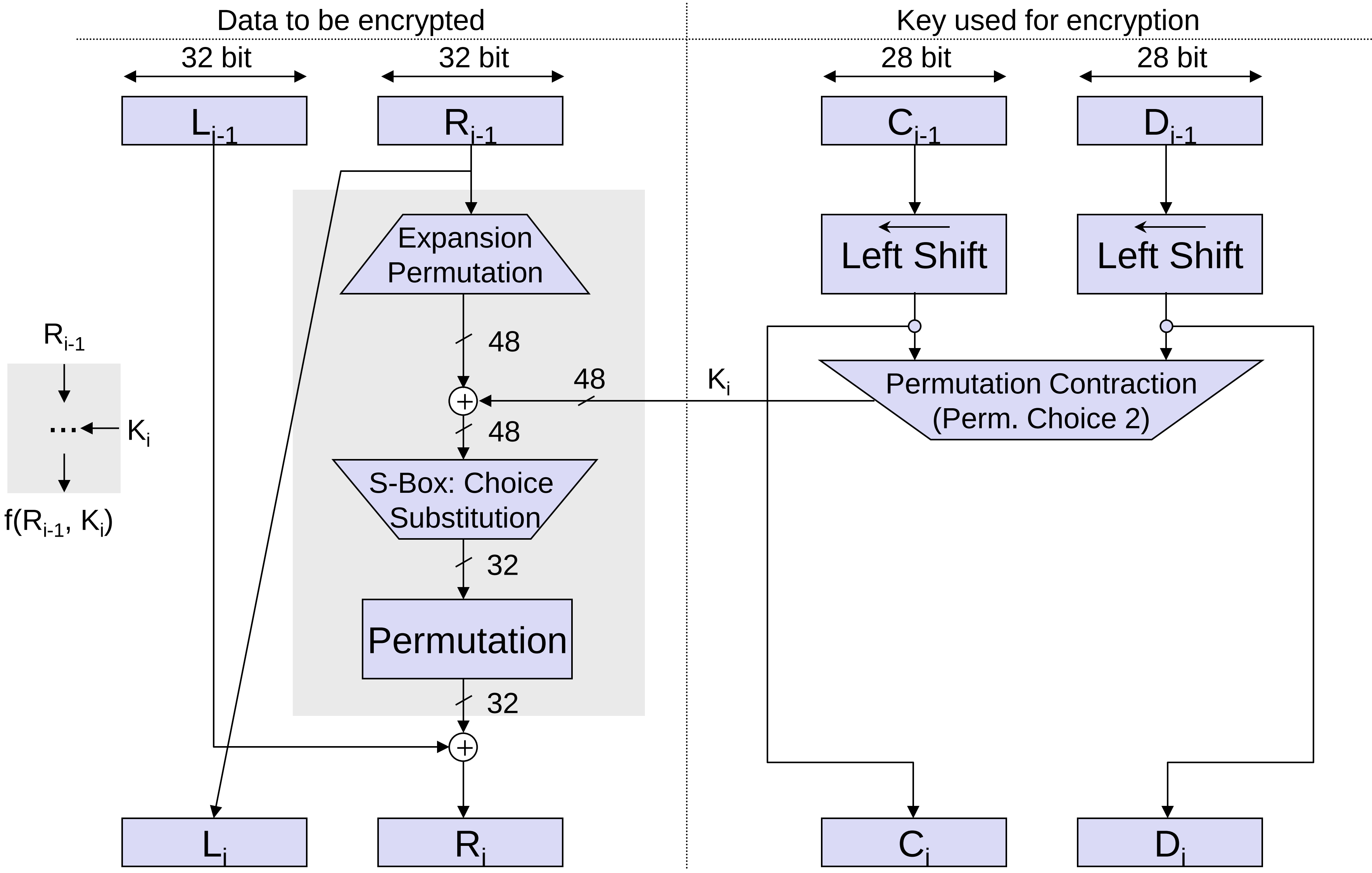
- Data Encryption Standard - DES
- Standardized 1977
- We will look at this cipher as example

- Warning: DES is no longer used as of today
  1. The key length is too small
  2. DES is comparably slow

DES – Algorithm Outline



DES – Single Iteration



DES – Security

- Main weakness: 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
- *Differential cryptanalysis:*
  - In 1990 E. Biham and A. Shamir published a cryptoanalysis method for DES
  - It looks specifically for differences in ciphertexts whose plaintexts have particular differences and tries to guess the correct key
  - 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 1990s that they knew about this kind of attacks in the 1970’s and that the S-boxes were designed accordingly

Extending the Key-Length of DES by Multiple Encryption

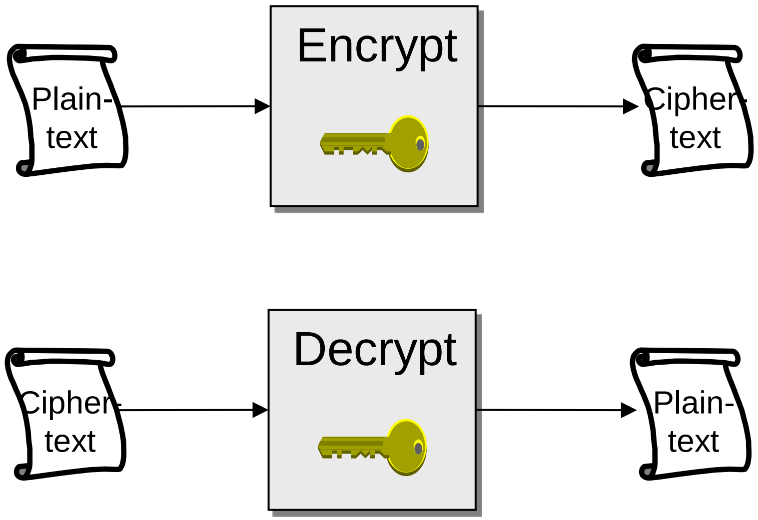
- Triple encryption scheme, as proposed by W. Tuchman in 1979:
  - $C = Enc_{K3}\, Dec_{K2}\, Enc_{K1}\, (P)$
  - The use of the decryption function *Dec* in the middle allows to use triple encryption devices with peers that only own single encryption devices by setting K1 = K2 = K3 (backwards compatibility with DES)
  - Triple encryption can be used with two or three different keys
    - Two keys: set K1 = K3
    - Three pairwise distinct 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 should be a better idea to use a different cipher, which offers a bigger key-length right away
- Double encryption is not a feasible option – there is an attack against it (Meet-in-the-middle-attack)

State-of-the-art symmetric cryptography

- Advanced Encryption Standard (AES)
- Standardized 2001
- Key and block lengths
  - Key Length: 128, 192, or 256 bit
  - Block Length: 128, 192, or 256 bit
- Fast
  - Roughly 3 times the speed of DES (200 MBit/s vs. 80 MBit/s)
  - Hardware support in modern CPUs (Intel AES-NI)
  - Modern CPUs: > 2GB/s
- Hardware implementations for embedded devices available
- Secure
  - There are attacks, but AES is still practically secure
  - AES seems to be the best we have, and it is among the most researched algorithms

Symmetric Encryption (revisited)

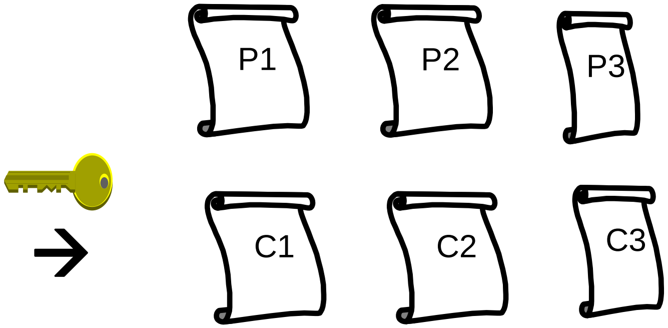
- General description:
  - The same key $K_{A,B}$ is used for enciphering and deciphering of messages:



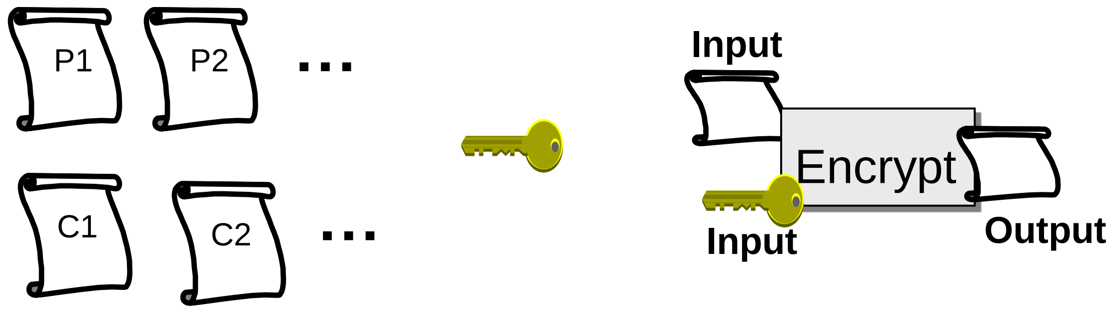
- Notation
  - If *P* denotes the plaintext message, $Enc_{K_{A,B}}(P)$ denotes the cipher text. The following holds: $Dec_{K_{A,B}}(Enc_{K_{A,B}}(P)) = P$
- Symmetric encryption
  - $Enc_{K_{A,B}}$ is at least an injective, often a bijective function
  - $Dec_{K_{A,B}}$ is the inverse function of $Enc_{K_{A,B}}$ is $Dec_{K_{A,B}} = (Enc_{K_{A,B}})^{-1}$
- Examples: DES, 3DES, AES, Twofish, RC4

Using Block Ciphers

- Problem
  - Block ciphers operate on a block size *b*. For example, b=128bit
  - We want to encrypt and decrypt a plaintext of larger length
- Solution
  - A plaintext *P* is segmented in blocks $p_1, p_2, \ldots p_n$ each of length *b.*
  - The last block may need padding to be of length *b*
  - The ciphertext *c* is the combination of $c_1, c_2, \ldots p_n$ where c_i denotes the result of the encryption of the i^{th} block of the plaintext message

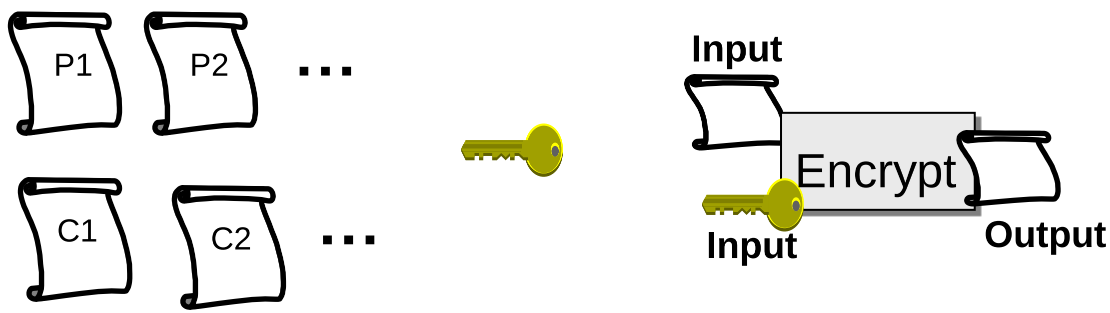


Modes of Encryption



- Modes of Encryption
  - Ways to combine encryption algorithm with the plaintext blocks and key and maybe additional input to generate the ciphertext blocks

Modes of Encryption



- Modes of Encryption
  - Ways to combine encryption algorithm with the plaintext blocks and key and maybe additional input to generate the ciphertext blocks
- Modes where the plaintext is input to the block cipher. Examples:
  - Electronic Code Book Mode (ECB), Cipher Block Chaining Mode (CBC)
- Modes where the plaintext is XORed with the output of a block cipher
  - A pseudorandom stream of bits, called *key stream,* is generated from the symmetric key *K* and a specific input per block,
    e.g. Enc_K(“Block 1”), Enc_K(“Block 2”), Enc_K(“Block 3”), …
  - Examples
    - Output Feedback Mode (OFB), Counter Mode (CTR)

Properties of modes of encryption

- *Error propagation*
  - characterizes the effects of bit-errors during transmission of ciphertext

- *Synchronization*
  - characterizes the effects of lost ciphertext data units

Properties of modes of encryption

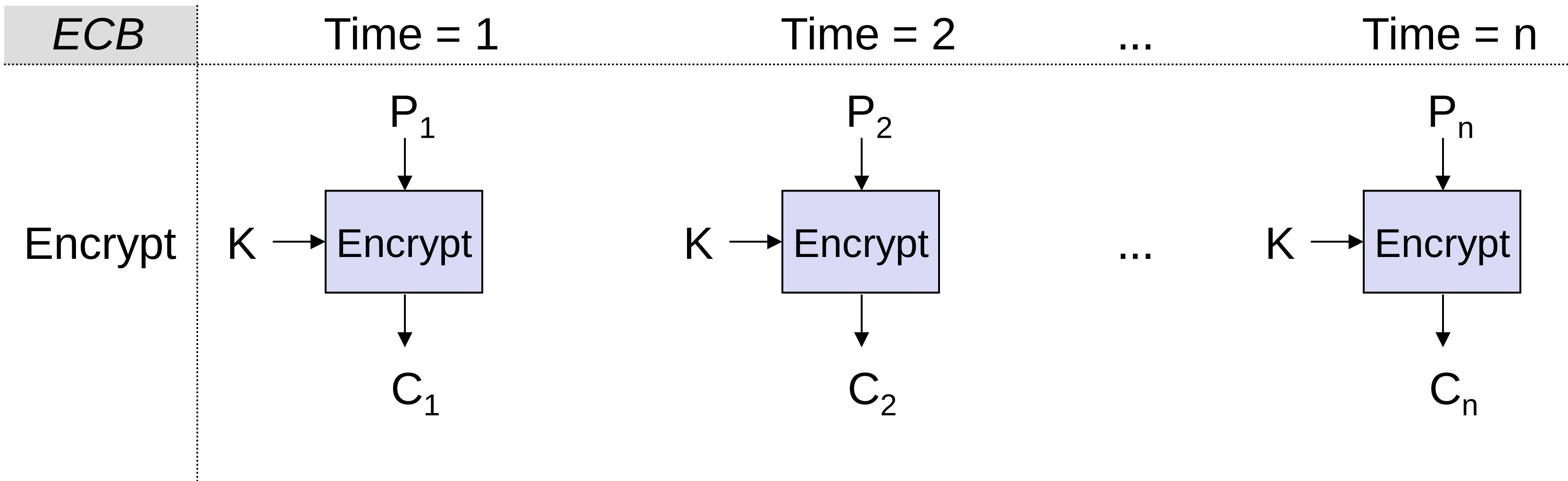
- Plaintext messages p_1, p_2, ... and ciphertext messages c_1, c_2, ...
- The following properties of the mode of encryption and cipher are of interest

- *Error propagation* characterizes the effects of bit-errors during transmission of ciphertext
  - Affects reconstructed plaintext $p_1´$, $p_2´$, ...
  - Depending on the mode of encryption, 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
  - Affects reconstructing the plaintext
  - Some modes of encryption cannot recover from lost ciphertext and need therefore explicit re-synchronization in case of lost messages
  - Other modes of encryption do automatically re-synchronize after 0 to n (n depending on the algorithm) ciphertext bits

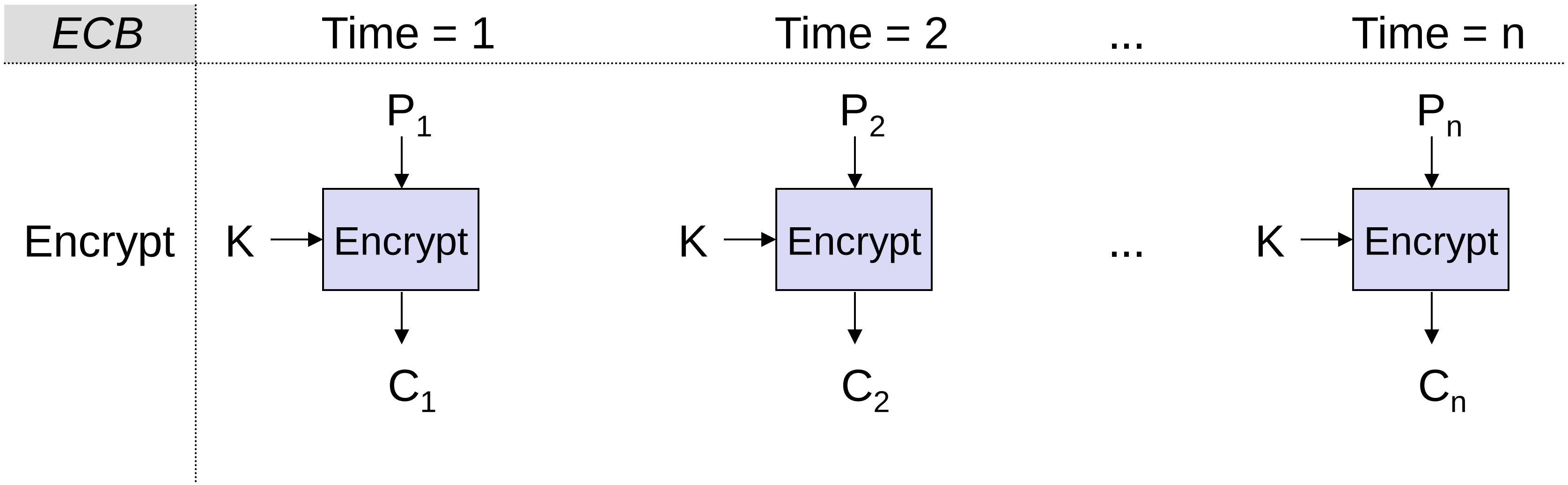
Modes of Encryption – ECB

- *Electronic Code Book Mode (ECB):*
  - Every block p_i is encrypted independently
    $c_i = Enc_K(p_i)$



Modes of Encryption – ECB

- *Electronic Code Book Mode (ECB):*
  - A bit error in one ciphertext block c_i results in a completely wrongly recovered plaintext block p_i' (subsequent blocks are not affected)
  - 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!



Modes of Encryption – ECB

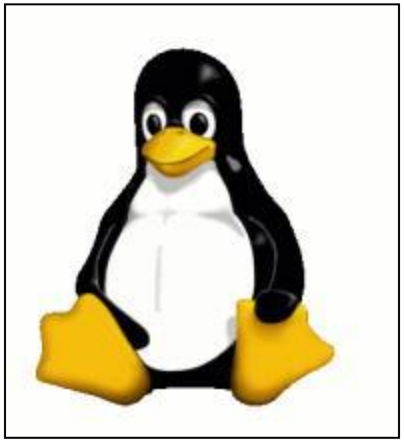
- If $p_i = p_j$ then $c_i = c_j$

- The string “This is network.This is network.Security” with AES-128 key = "AliceBob“

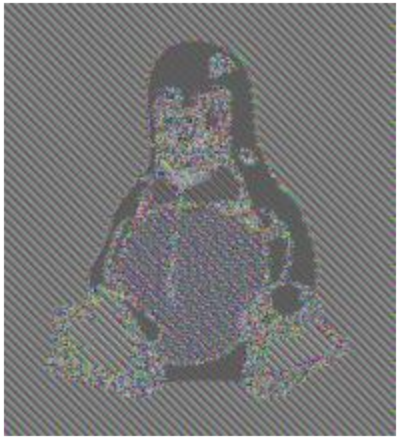
2d	3c	ab	1b	a0	80	77	ec	e8	1d	56	0d	09	2b	f6	77
2d	3c	ab	1b	a0	80	77	ec	e8	1d	56	0d	09	2b	f6	77
16	ea	2c	19	97	e7	40	db	06	a0	35	93	49	5c	37	0b

- Why is it important for this example to use AES with blocklength 128?

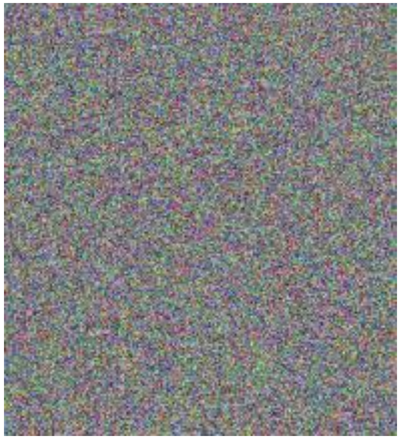
Modes of Encryption – ECB



Original

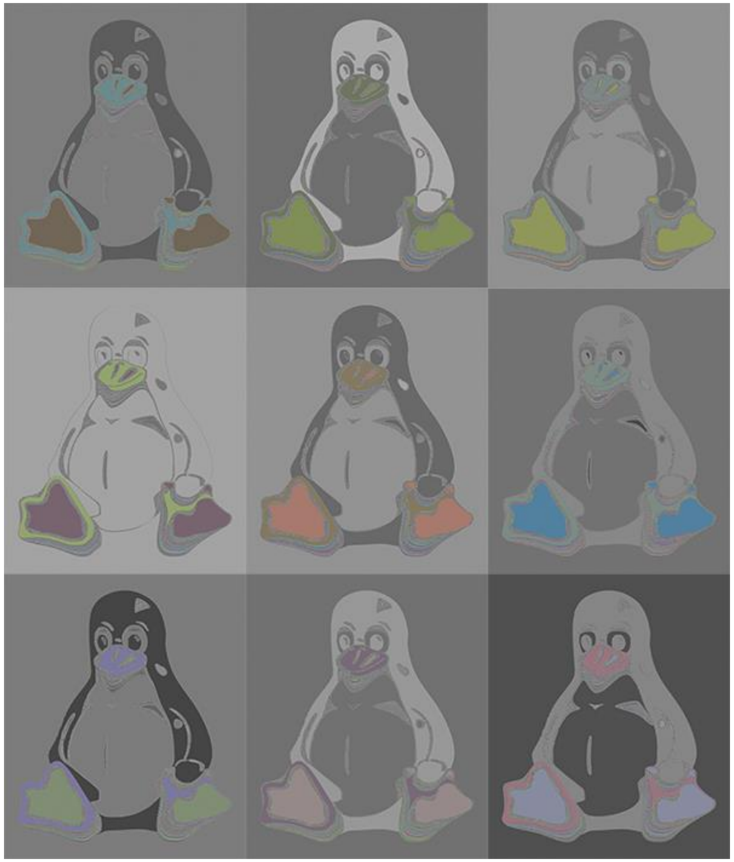


Encrypted using ECB mode



Encrypted using other modes

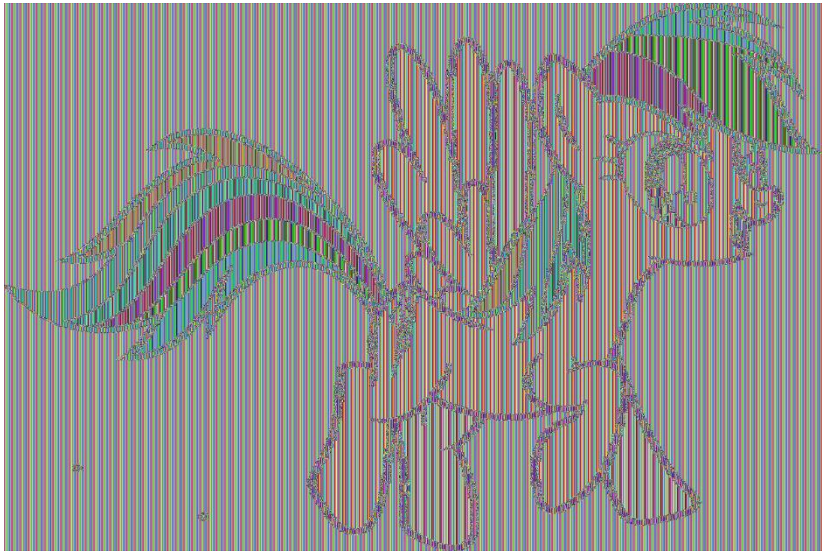
Source: http://www.wikipedia.org/



Fun with fixed *P* and varying keys

Source: https://filippo.io/the-ecb-penguin/

Modes of Encryption – ECB



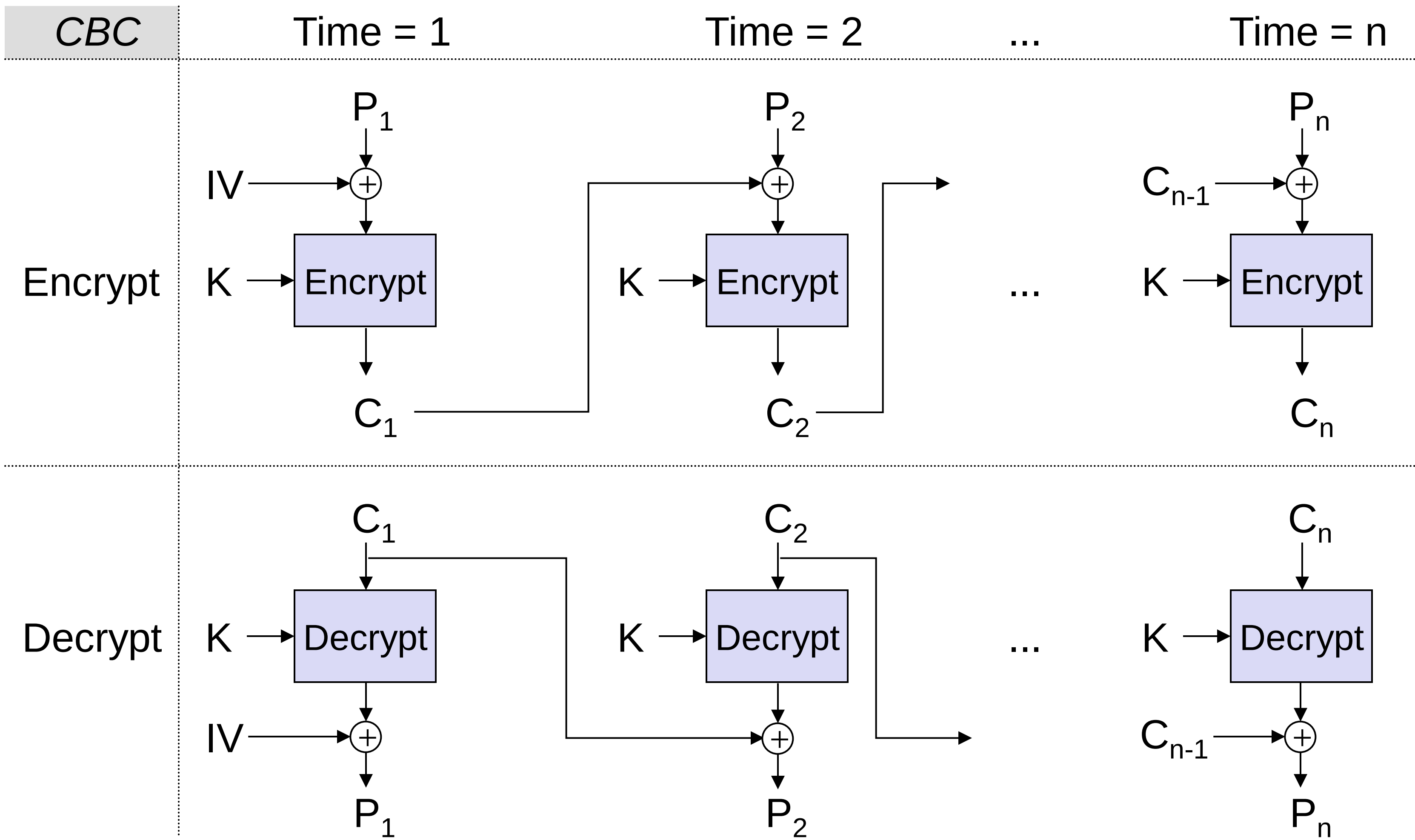
Does not only work with penguins

Modes of Encryption – CBC

- *Cipher Block Chaining Mode (CBC)*

- Before encrypting a plaintext block, it is ⊕ with the preceding ciphertext block
- An initial value, called *Initialization Vector (IV)* is required

Modes of Encryption – CBC



- *Cipher Block Chaining Mode (CBC):*

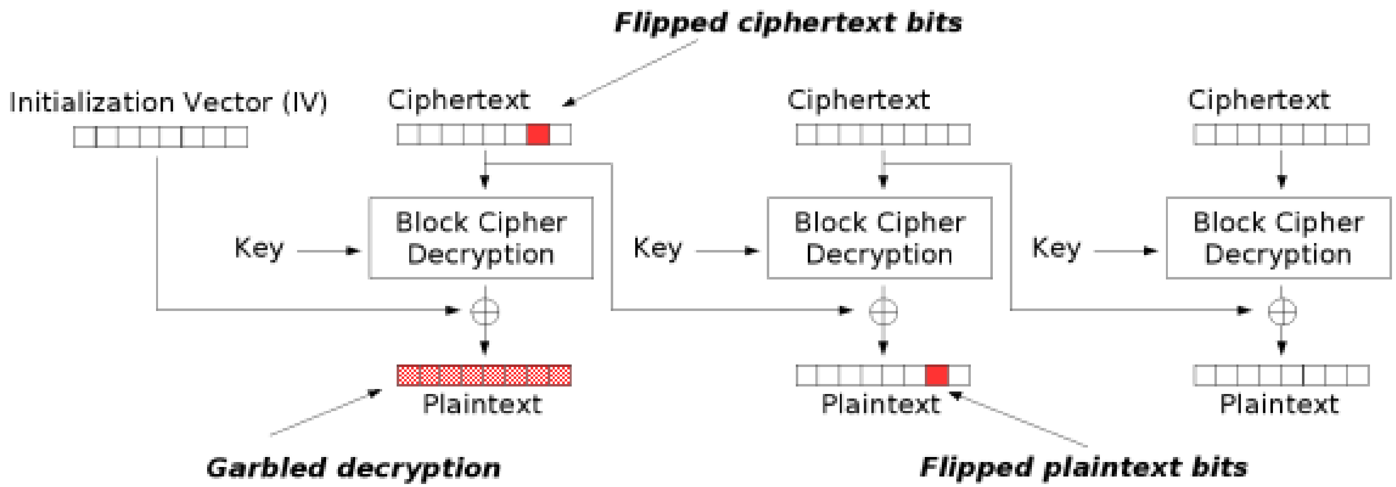
  - $c_i = Enc_K(c_{i-1} \oplus p_i)$
  - $p_i' = c_{i-1} \oplus Dec_K(K, c_i)$
  - $c_0 = \text{IV}$

  Both parties need to agree on an *Initialization Vector (IV)*
  IV may be public

- Identical plaintext blocks are encrypted to non-identical ciphertext

CBC Error Propagation and Synchronization

- Error Propagation
  - p_i' depends on c_{i-1} and c_i
  - One distorted ciphertext block results in two distorted plaintext blocks



Modification attack or transmission error for CBC

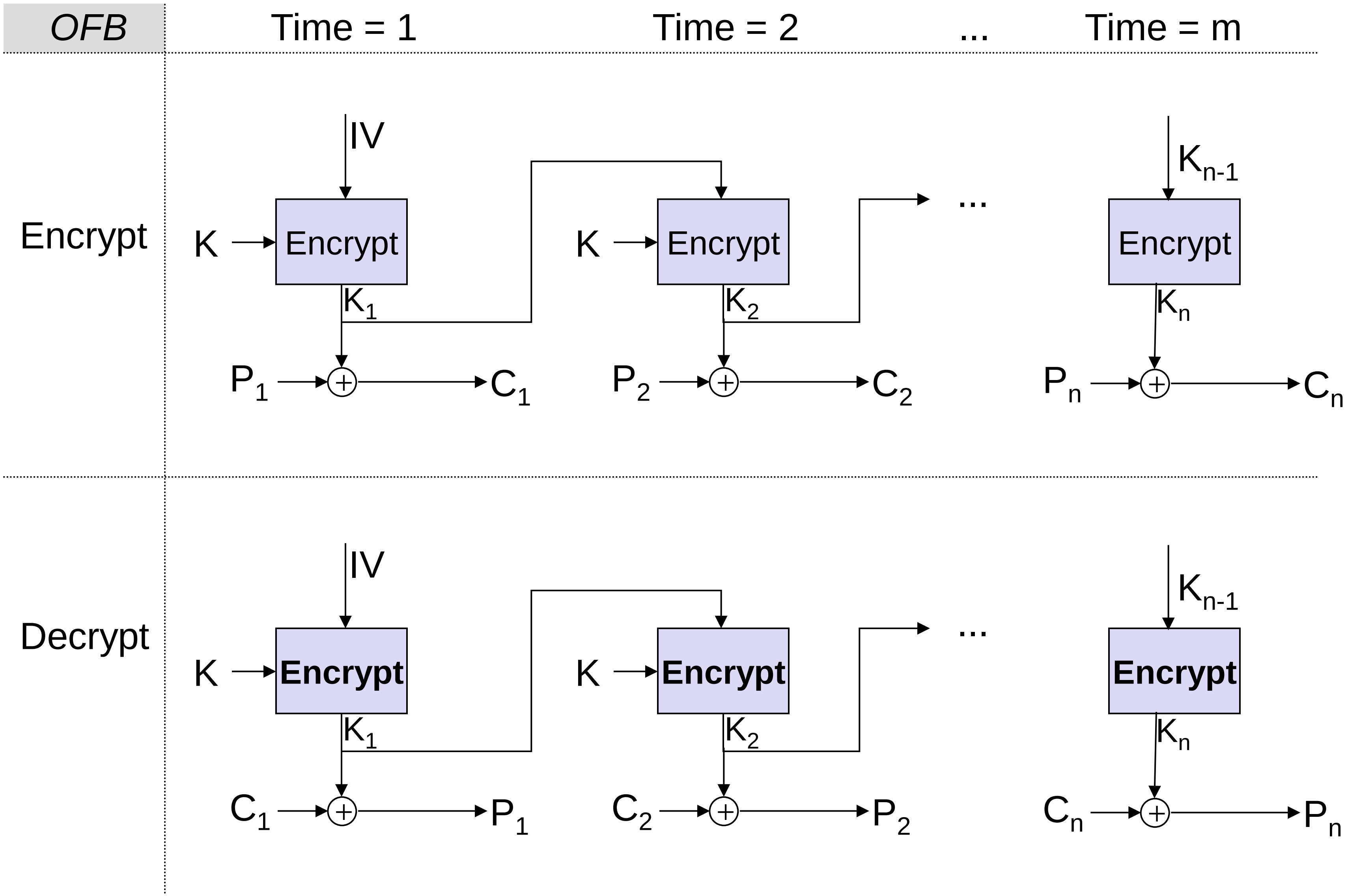
Source: http://www.wikipedia.org/

- Synchronization
  - If the number of lost bits is a multiple integer of *b*, one additional block p_{i+1} is misrepresented before synchronization is re-established.
  - If any other number of bits are lost explicit re-synchronization is needed.

Modes of Encryption – OFB

- *Output Feedback Mode (OFB):*
  - The block encryption algorithm is used to generate a key stream that depends only on K and IV
    - $K_0 = IV$
    - $K_i = Enc_K(K_{i-1})$
    - $C_i = P_i \oplus K_i$
  - The plaintext blocks are XORed with the pseudo-random sequence to obtain the ciphertext and vice versa

Modes of Encryption – OFB



Modes of Encryption – OFB

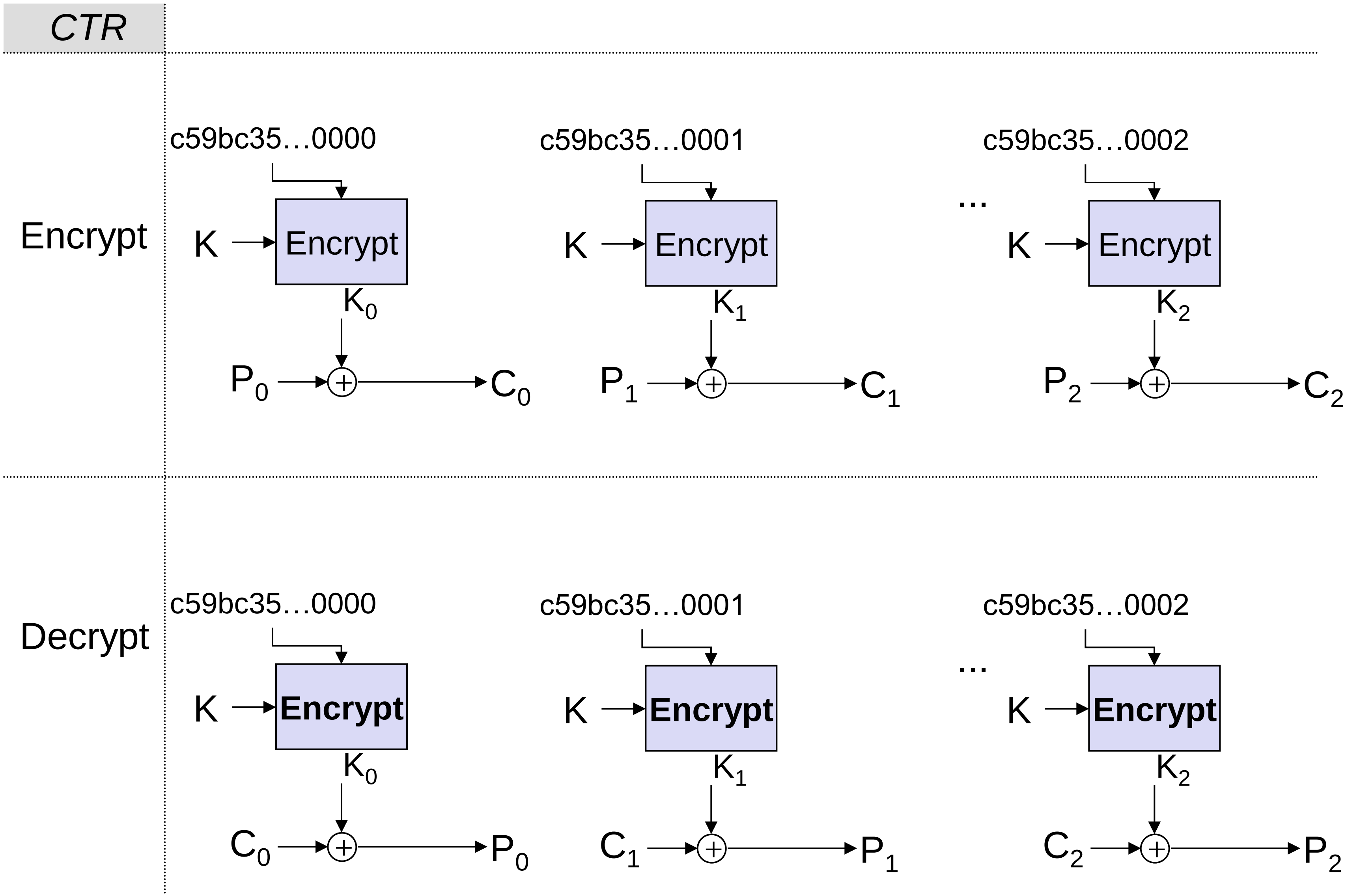
- Properties of OFB:
  - Error propagation:
    - Single bit errors result only in single bit errors ⇒ no error multiplication
  - Synchronisation:
    - If some bits are lost explicit re-synchronization is needed
  - Advantage:
    - The pseudo-random sequence can be pre-computed in order to keep the impact of encryption to the end-to-end delay low
  - Drawbacks:
    - It is easily possible for an attacker to manipulate specific bits of the plaintext

Modes of Encryption - CTR

- *Counter Mode (CTR)*
  - The block encryption algorithm is used to generate a key stream that depends on K and a counter function ctr_i .
  - The counter function can be simply an increment modulo 2^w, where w is a convenient register width, e.g.
    - $ctr_i = Nonce \mathbin{||} i$
  - The counter function does not provide any security other than the uniqueness of the input to the block cipher function E
  - The plaintext blocks are XORed with the pseudo-random sequence to obtain the ciphertext and vice versa
  - Putting everything together:
    - $K_i = Enc_K(Nonce \mathbin{||} i)$
    - $C_i = P_i \oplus K_i$

Modes of Encryption - CTR

CTR



Modes of Encryption - CTR

- Properties of CTR:
  - Error propagation:
    - Single bit errors result only in single bit errors $\Rightarrow$ no error multiplication
  - Synchronisation:
    - If some bits are lost explicit re-synchronization is needed.
  - Advantage:
    - The key stream can be pre-computed in order to keep the impact of encryption to the end-to-end delay low.
    - The computation of the key stream can be parallelized.
  - Drawbacks:
    - It is easily possible for an attacker to manipulate specific bits of the plaintext

Additional References

[AES01a] National Institute of Standards and Technology (NIST). *Specification for the Advanced Encryption Standard (AES).* Federal Information Processing Standards Publication, February 2001.

[DR97a] J. Daemen, V. Rijmen. *AES Proposal: Rijndael.* http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf, 1997.

[FMS01a] S. Fluhrer, I. Mantin, A. Shamir. *Weaknesses in the Key Scheduling Algorithm of RC4.* Eighth Annual Workshop on Selected Areas in Cryptography, August 2001.

[Riv01a] R. Rivest. *RSA Security Response to Weaknesses in Key Scheduling Algorithm of RC4*. http://www.rsa.com/rsalabs/technotes/wep.html, 2001.

[SIR01a] A. Stubblefield, J. Ioannidis, A. D. Rubin. *Using the Fluhrer, Mantin, and Shamir Attack to Break WEP.* AT&T Labs Technical Report TD-4ZCPZZ, August 2001.