Symmetric Encryption

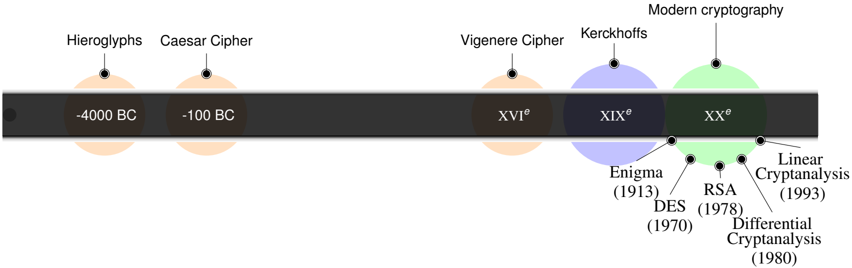
Vincent Migliore

vincent.migliroe@insa-toulouse.fr

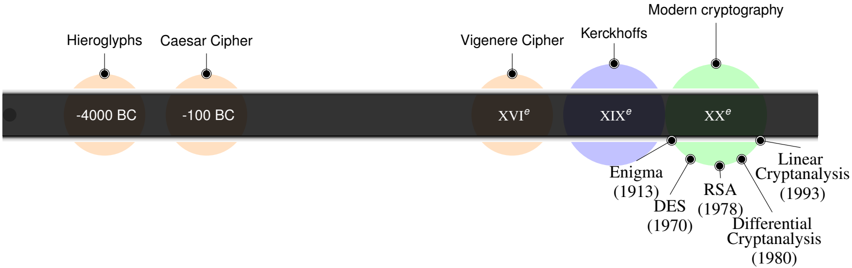
INSA-TOULOUSE / LAAS-CNRS

Summary of previous lesson

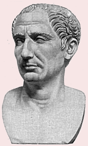
Brief History of Cryptography

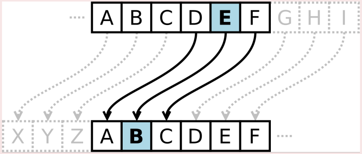


Brief History of Cryptography



Caesar Cipher

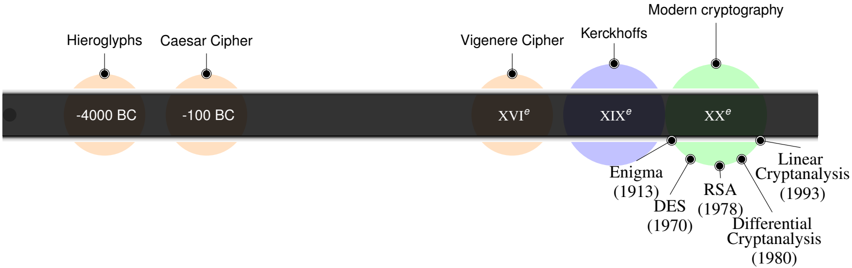




$$\mathrm{Enc}(k, m_i) = m_i + k \ [26]$$
$$\mathrm{Dec}(k, c_i) = c_i - k \ [26]$$

Vulnerable to frequency analysis.

Brief History of Cryptography



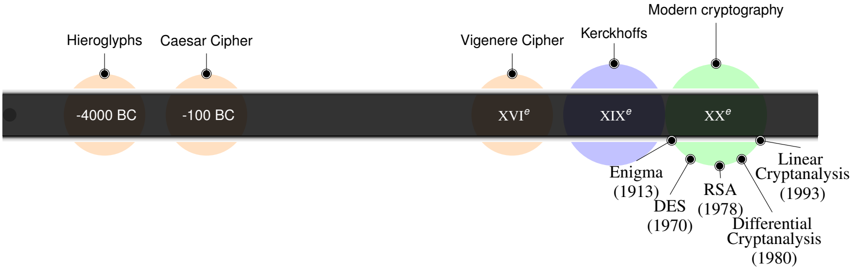
(Blaise de) Vigenère Cipher

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

$$\begin{aligned} \mathrm{Enc}(k_i, m_i) &= m_i + k_i \ [26] \\ \mathrm{Dec}(k_i, c_i) &= c_i - k_i \ [26] \end{aligned}$$

Still vulnerable to frequency analysis when $|K| < |M|$

Brief History of Cryptography

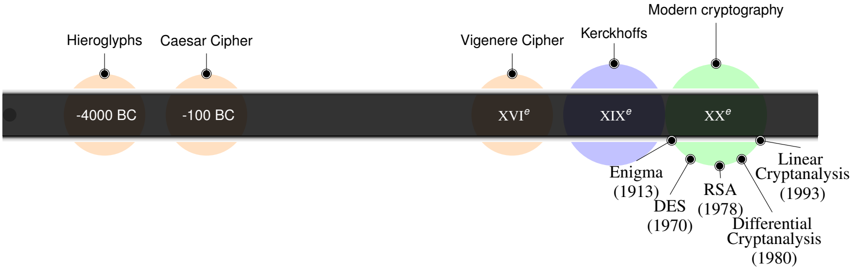


(Auguste) Kerckhoffs principle

Military cryptographier. Provided several principles that influenced modern cryptography:

- The system should be, if not theoretically unbreakable, unbreakable in practice.
- The design of a system should not require secrecy, and compromise of the system should not break security.

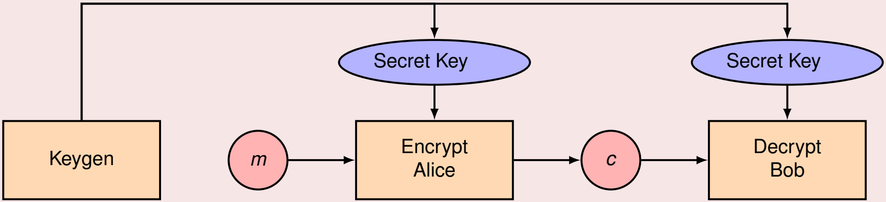
Brief History of Cryptography



Modern cryptography

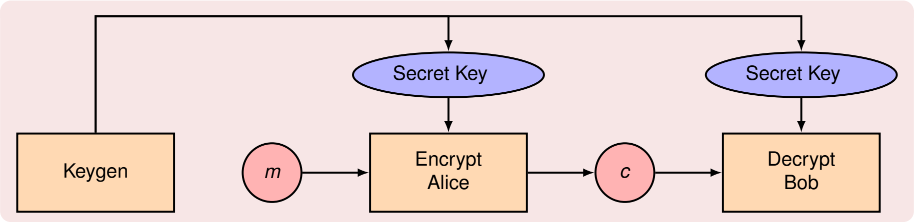
- Major improvements in terms of mathematical background.
- Industrialization of calculators $\Longrightarrow$ security based on computational complexity.
- Highly standardized (mostly by Americans): NIST, IETF, ISO.

Standard constructions



Symmetric Encryption

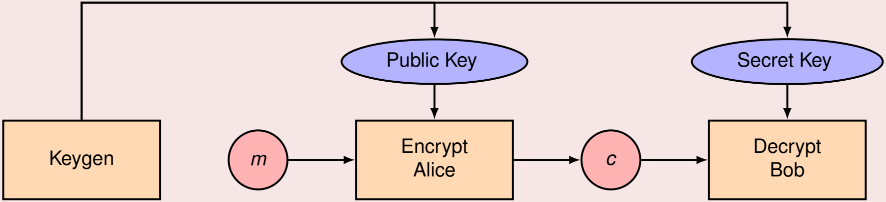
Privacy
Integrity
Authentication
Non-repudiation



Symmetric Encryption

- ✓ Privacy
- ✗ Integrity
- ✓ Authentication
- ✗ Non-repudiation (both Alice and Bob can Encrypt)

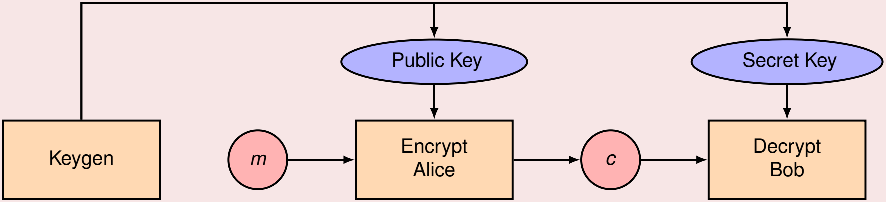
Standard constructions



Asymmetric Encryption

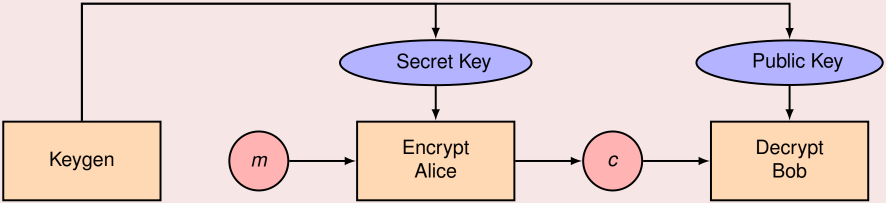
- Privacy
- Integrity
- Authentication
- Non-repudiation

Standard constructions



Asymmetric Encryption

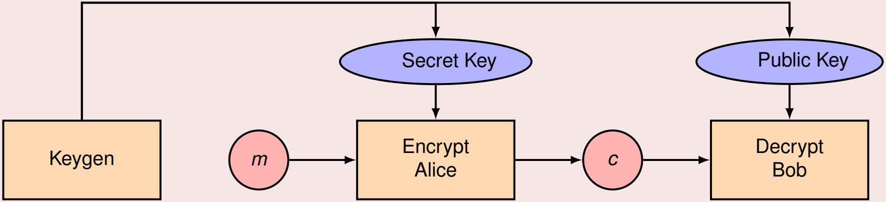
- ✓ Privacy
- ✗ Integrity
- ✗ Authentication
- ✗ Non-repudiation



Signature

- Privacy
- Integrity
- Authentication
- Non-repudiation

Standard constructions



Signature

- ✗ Privacy
- ✗ Integrity
- ✓ Authentication
- ✓ Non-repudiation

Symmetric encryption achieving perfect secrecy

Perfect secrecy definition

Perfect Secrecy (or information-theoretic secure) means that the ciphertext conveys no information about the content of the plaintext.

One Time Pad (Vernam, 1917)

message $\oplus$ key = cipher   cipher $\oplus$ key = message

```
message :  0 1 0 1 1 1 1 0 0 0 0 1 0 0 1
clé     :  1 1 0 0 0 1 0 1 0 0 0 1 1 1 0
=========================================
chiffré :  1 0 0 1 1 0 1 1 0 0 0 1 1 1 1
```

Highly secure

Uniform output + for a given ciphertext, any plaintext is possible.

Symmetric encryption achieving perfect secrecy

Perfect secrecy definition

Perfect Secrecy (or information-theoretic secure) means that the ciphertext conveys no information about the content of the plaintext.

One Time Pad (Vernam, 1917)

message $\oplus$ key = cipher   cipher $\oplus$ key = message

```
message :  0 1 0 1 1 1 1 0 0 0 0 1 0 0 1
clé     :  1 1 0 0 0 1 0 1 0 0 0 1 1 1 0
=========================================
chiffré :  1 0 0 1 1 0 1 1 0 0 0 1 1 1 1
```

But limited

- Shannon: $|K| \geq |M| \implies$ unpracticable (+ key must not be used twice)
- Maleable: Any partial knowledge on the plaintext leads to devastating attack.

Symmetric encryption achieving perfect secrecy

Perfect secrecy definition

Perfect Secrecy (or information-theoretic secure) means that the ciphertext conveys no information about the content of the plaintext.

One Time Pad (Vernam, 1917)

message $\oplus$ key = cipher   cipher $\oplus$ key = message

```
message :  0 1 0 1 1 1 1 0 0 0 0 1 0 0 1
clé     :  1 1 0 0 0 1 0 1 0 0 0 1 1 1 0
=========================================
chiffré :  1 0 0 1 1 0 1 1 0 0 0 1 1 1 1
```

Remark

OTP can be viewed as a Vigenère cipher with 1-bit symbols with key as long as the message.

Perfect secrecy definition

Perfect Secrecy (or information-theoretic secure) means that the ciphertext conveys no information about the content of the plaintext.

One Time Pad (Vernam, 1917)

message $\oplus$ key = cipher   cipher $\oplus$ key = message

```
message :  0 1 0 1 1 1 1 0 0 0 0 1 0 0 1
clé     :  1 1 0 0 0 1 0 1 0 0 0 1 1 1 0
=========================================
chiffré :  1 0 0 1 1 0 1 1 0 0 0 1 1 1 1
```

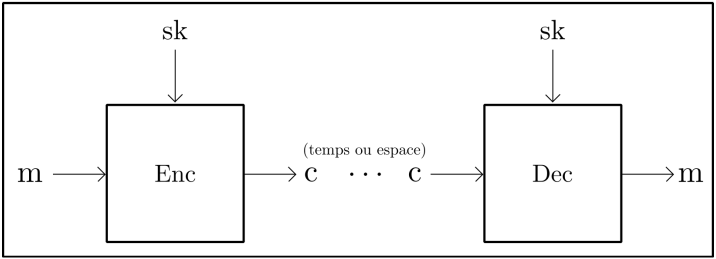
Remark [2]

In one specific case, OTP may be practical:

- We generate offline an incredible amount of random bits.
- We physically store these bits into at least 2 mass storages.
- We distribute to some recipients a mass storage.
- Afterword, OTP communication can be started using random bits

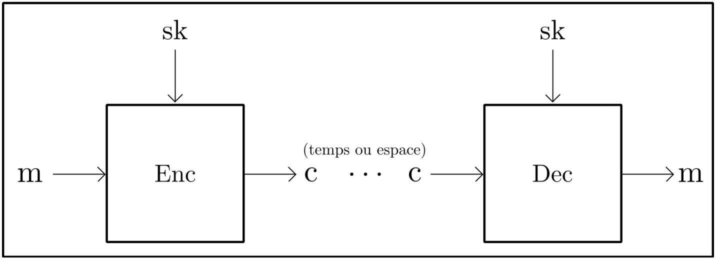
Practical symmetric encryption

Symmetric encryption - beyond OTP



Limitations of OTP

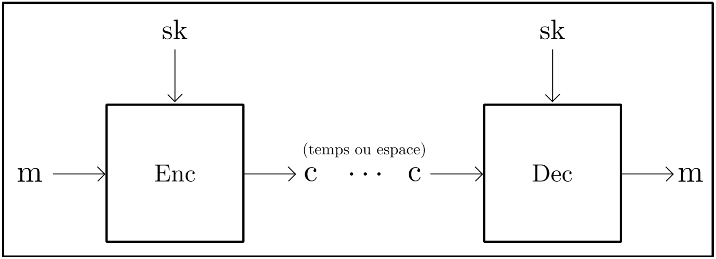
- Key length equals to message length;
- maleable;
- Cannot use key twice.



Desirable property and consequences

- We would like to use a bounded key for large messages;
- At some point, we must reduce security on perfect secrecy to allow such property;
- Now, we consider that attacker may break cryptosystem, but we want that such attack demands unpractical power.

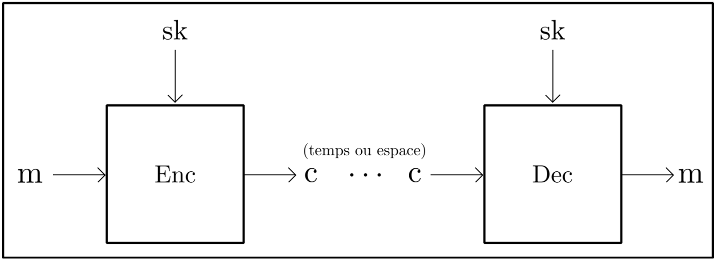
Symmetric encryption - Block cipher



Definition of a block cipher

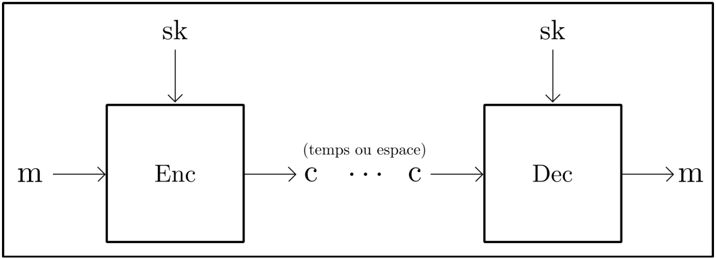
- Message is split into blocks of size n;
- Key is selected as random string of size k;
- Each block of message is encrypted with the key and produces ciphertext of size n;
- decryption is the invert operation of encryption, using the same key and the same blocksize.

Symmetric encryption - Block cipher



Construction of a block cipher

- Assumption: Block ciphers are secured if they can be modeled as pseudo-random permutations (PRPs).
- Formally: an *n*-bit blockcipher under a randomly-chosen key is computationally indistinguishable from a randomly-chosen *n*-bit permutation.
- Challenge: Find a computationally efficient algorithm that meet the assumption.



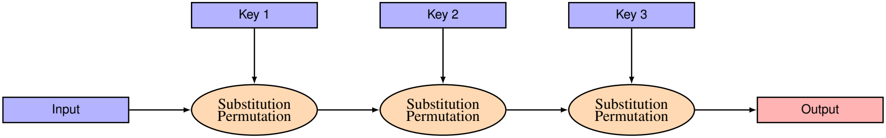
Practical block cipher - Shannon properties (1949)

Two main properties for block ciphers:

- Diffusion: If 1 bit of plaintext is changed, statistically half of output bits must be changed (avalanch effect).
- Confusion: 1 bit of ciphertext must be linked with several bits of the key.

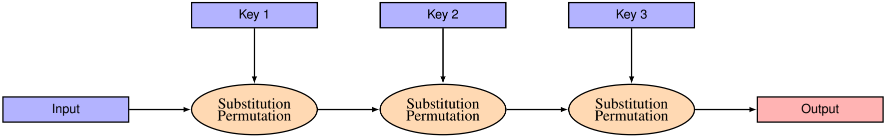
Question: Does it apply to OTP?

SP-Network



Construction

- SP-network is a succession of Substitution/permutation functions parametrized with a key.
- Substitution/permutation functions must be invertible.
- Each iteration of Substitution/permutation function is called a round.
- The more rounds implemented, the more outputs looks uniform and independant from message/key (if properly implemented).
- Security: finding information about plaintext must be as hard as an exhaustive search on the key $\Longrightarrow$ security level $\approx 2^{\text{key length}}$.

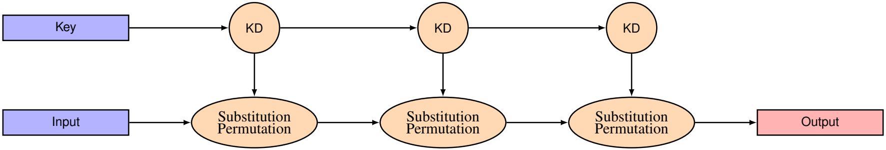


Design considerations

Two main approaches exist:

- Making Substitution/permutation pseudo-random with a unique key:
  - Requires the implementation of many Substitution/permutation functions.
- Making Key pseudo-random with a fixed Substitution/permutation function:
  - Requires the generation of many keys, as many as the number of rounds.

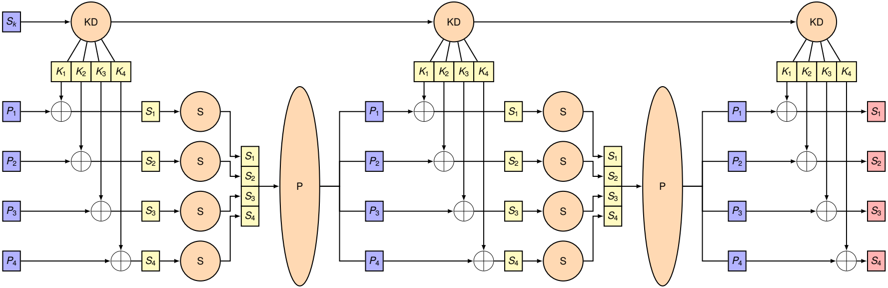
SP-Network



Most practical approach

- Second choice: Key is pseudo-random with a fixed Substitution/permutation.
- Round keys are generated with a Key Derivation function.

SP-Network - in details



Definitions

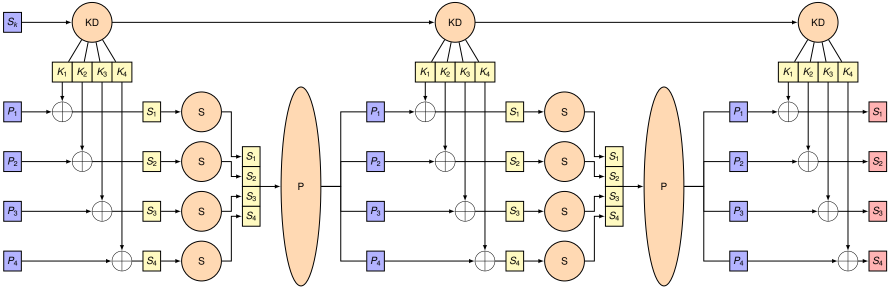
Let:

- n be the length in bits of a block.
- k be the length in bits of the key.

Construction

A SP-Network is constructed with the execution of a given number N of rounds. A round consists in 1 round key addition, 1 Substitution and 1 Permutation. Each function is invertible to provide symmetric encryption.

SP-Network - in details



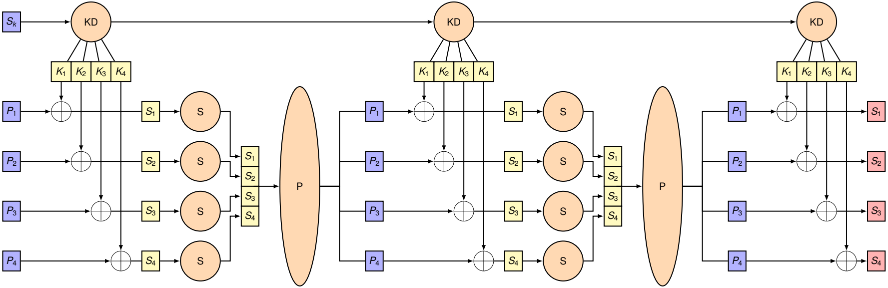
Substitution → S-BOX

Substitutes 1 symbol to another. It contributes to confusion because it makes output non-intelligible. It also contributes to non-linearity, i.e.:
S-BOX($v_1 \oplus v_2$) $\neq$ S-BOX(v_1) $\oplus$ S-BOX(v_2).

Permutation → P-BOX

Switch symbols. It contributes to diffusion because it dispatches bits all over the internal state. By construction, it is linear, i.e.:
P-BOX($v_1 \oplus v_2$) = P-BOX(v_1) $\oplus$ P-BOX(v_2).

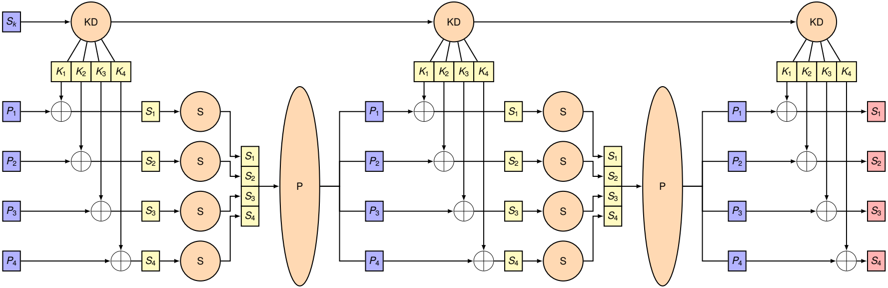
SP-Network - in details



Important note

S-BOX and P-BOX are basically permutations, that is why sometimes we prefer define S-BOX and D-BOX (*Diffusion*-BOX), where both are permutations but first one is non-linear.

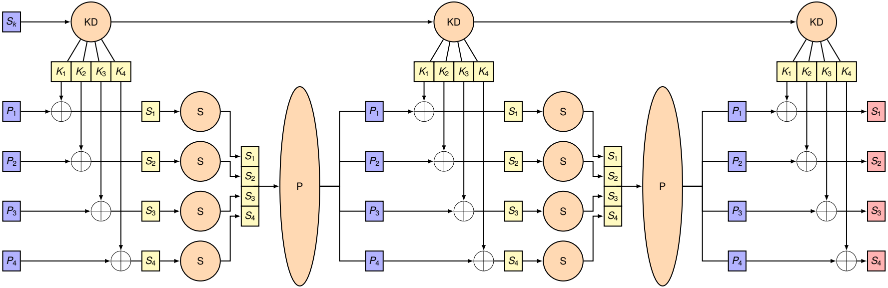
SP-Network - in details



KD

- Key derivation function. For N rounds and a k-bit key, generates $(N+1)$ n-bit subkeys.
- Like OTP, make input uniform before each round.

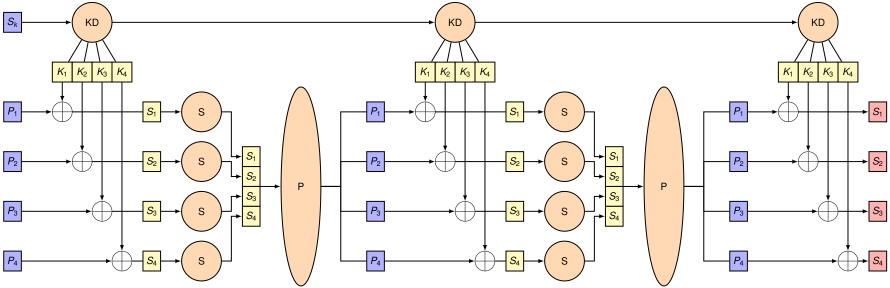
SP-Network - in details



Why non-linearity so important? Application

We note (X_1, X_2) two messages and (Y_1, Y_2) associated ciphertexts encrypted with same key.
We consider a P-Network (i.e. SP-Network without S-BOX), and $N = 2$ rounds. Evaluates $(\Delta Y = Y_1 \oplus Y_2)$

SP-Network - in details



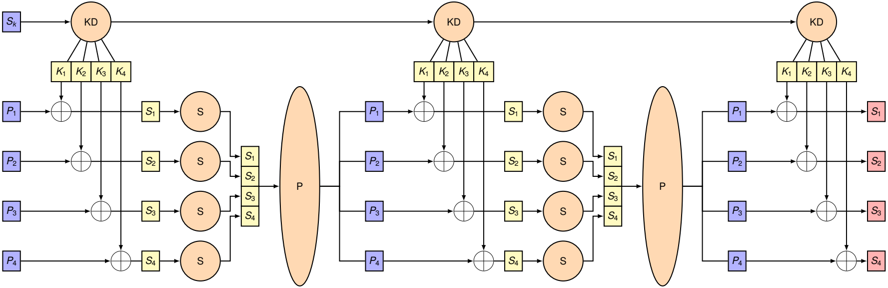
Why non-linearity so important? Application

We note (X_1,X_2) two messages and (Y_1, Y_2) associated ciphertexts encrypted with same key.
We consider a P-Network (i.e. SP-Network without S-BOX), and $N = 2$ rounds. Evaluates ($\Delta Y = Y_1 \oplus Y_2$)

Answer

Due to linearity, $\Delta Y = \text{P-BOX}(\text{P-BOX}(X_1 \oplus X_2))$ independent from the key $\Longrightarrow$ differential attack.

SP-Network - in details



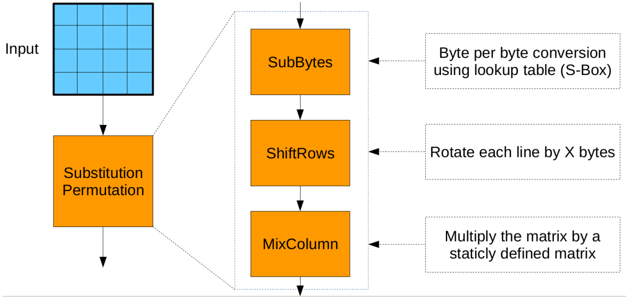
Why non-linearity so important? Application

We note (X_1, X_2) two messages and (Y_1, Y_2) associated ciphertexts encrypted with same key.
We consider a P-Network (i.e. SP-Network without S-BOX), and $N = 2$ rounds. Evaluates $(\Delta Y = Y_1 \oplus Y_2)$

Note

More advanced attack tries to find some linearity inside S-BOX, in order to partially remove key bits. It is so called linear cryptanalysis.

Symmetric encryption - case of AES (Rijndael -

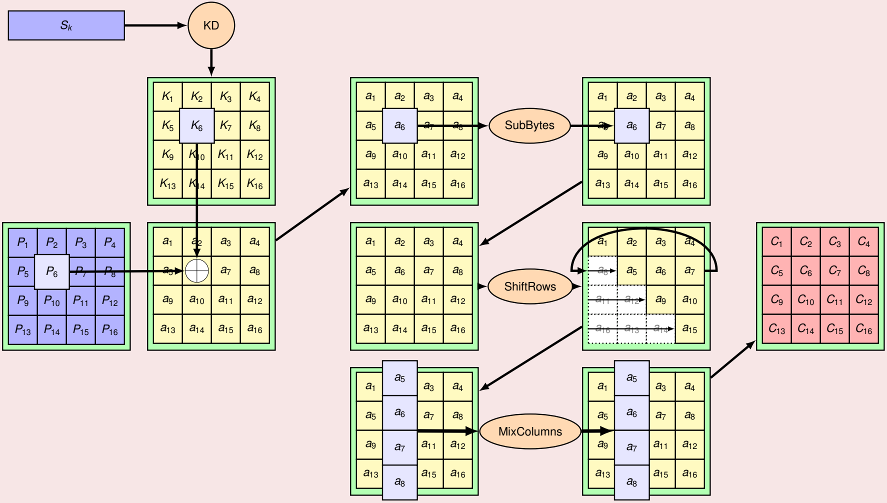


History

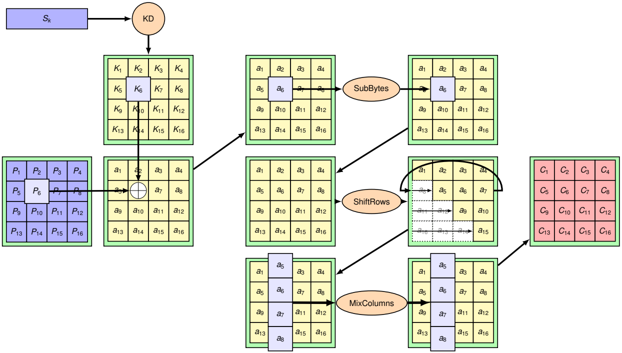
- Designed by Joan Daemen et Vincent Rijmen (Belgium).
- Winner in 2000 of the NIST “AES” competition.
- Based on SP-NETWORK.
- Interesting construction: Both security AND implementation have been studied during design process.

Symmetric encryption - Round of AES

Description of 1 round of AES:



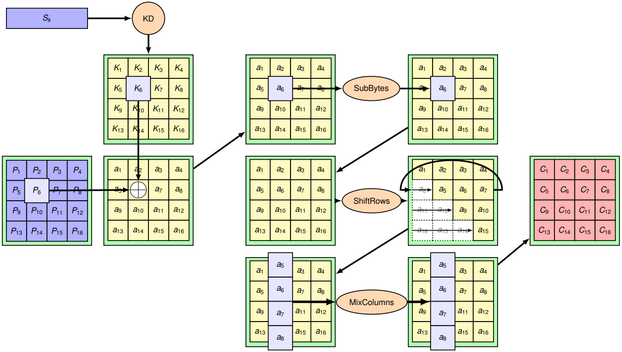
Symmetric encryption - Round of AES



Structure

Internal state is composed of a 4x4 matrix of bytes. 4 operations are executed over internal state each round:

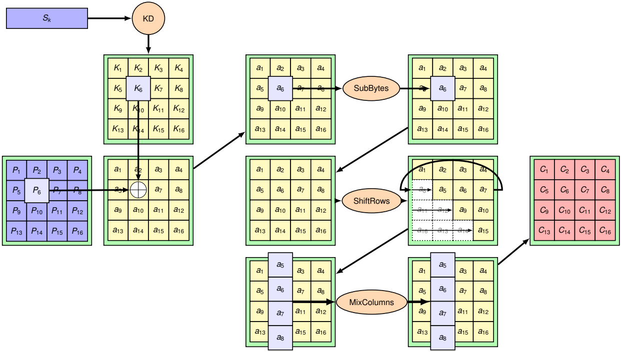
1. AddRoundKey
2. SubBytes (S-BOX)
3. ShiftRows (D-BOX)
4. MixColumns (D-BOX)



1 - AddRoundKey

- xor between state and round-key.
- if message independant from key, and key uniform, then the new state looks uniform.

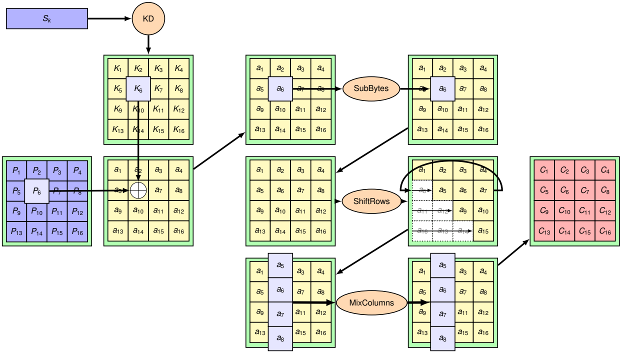
Symmetric encryption - Round of AES



2 - SubBytes

- Non-linearity: Minimization of input-output correlation.
- Complexity: Complex expression in GF(2^8).
- Simple implementation: Look-up table (and must be since litteral expression complex).

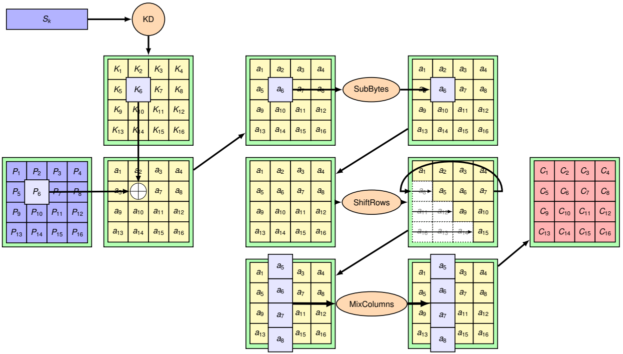
Symmetric encryption - Round of AES



3 - ShiftRows

- Variable byte rotation of each line depending on line index.
- First line: no rotation.
- Second row: 1 byte rotation.
- Third row: 2 bytes rotation.
- Fourth row: 3 bytes rotation.

Symmetric encryption - Round of AES

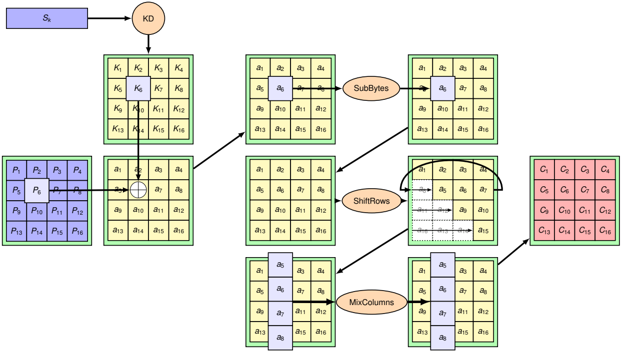


4 - MixColumns

Column per column scrambling of coefficients. Equivalent to multiplying each column by following matrix:

$$\begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \end{pmatrix}$$

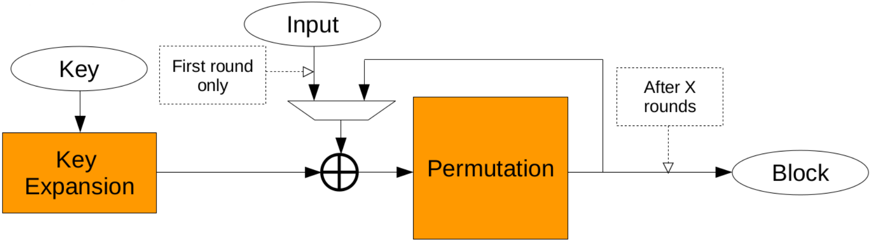
Symmetric encryption - Round of AES



High level consideration

MixColumns of last round is skipped to make Encryption/decryption symmetric, i.e.:

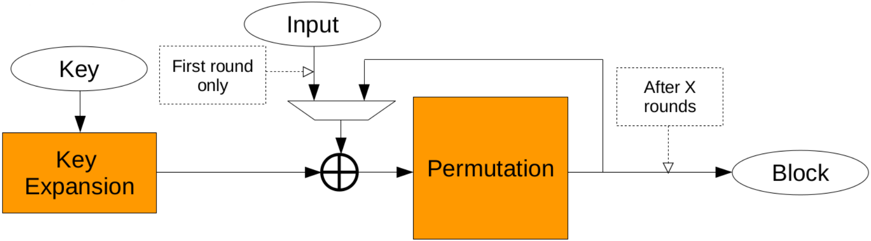
- Encryption: $\oplus \rightarrow$ S-BOX $\rightarrow$ D-BOX $\rightarrow \cdots \rightarrow \oplus \rightarrow$ S-BOX $\rightarrow \oplus$
- Decryption: $\oplus \rightarrow$ S-BOX $\rightarrow$ D-BOX $\rightarrow \cdots \rightarrow \oplus \rightarrow$ S-BOX $\rightarrow \oplus$



Security

- AES is considered as a good PRP if implemented properly.
- Security depends on the number of rounds executed:

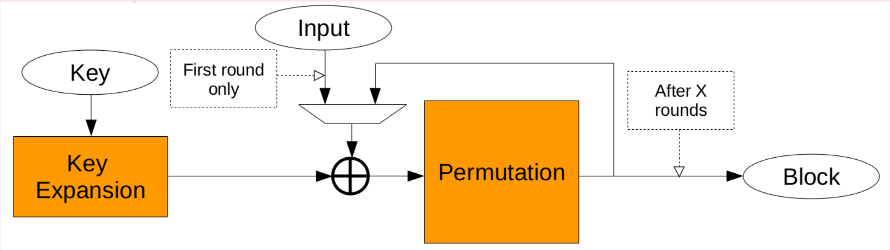
Name	Key length (bits)	Security	rounds
AES-128	128	128	10
AES-196	196	192	12
AES-256	256	256	14



Security

- Best known attack: biclique attack on full AES-128 reducing security by 2 bits (i.e. 4 times faster than exhaustive search).
- Variant of Meet-In-The-Middle (MITM) attack (Diffie and Hellman 1977)

Numerical application

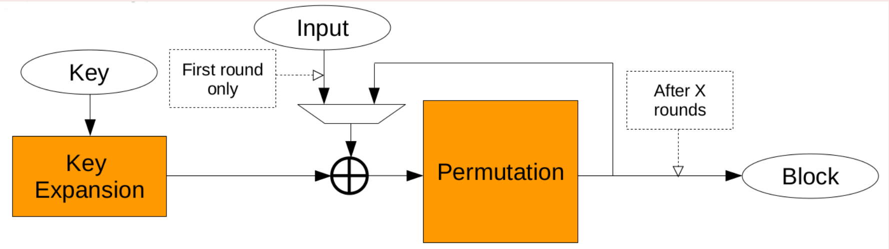


Question

We consider AES-256 (i.e. blocks of 4x4 bytes, 12 rounds). I can encrypt:

- 16 bytes of data.
- 12x16 bytes of data.
- No limitation.

Numerical application

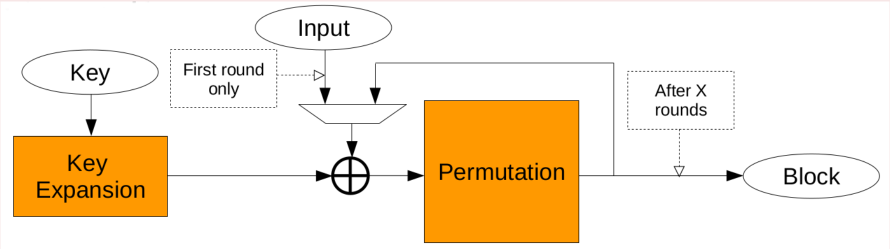


Question

We consider AES-256 (i.e. blocks of 4x4 bytes, 12 rounds). I can encrypt:

- 16 bytes of data.
- 12x16 bytes of data.
- No limitation.

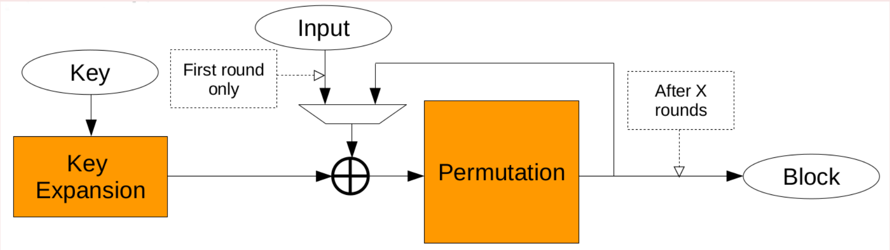
Numerical application



Question

We consider AES-256 (i.e. blocks of 4x4 bytes, 12 rounds). Compared to OTP:

- I have a smaller secret key.
- I have a larger secret key.
- I have a comparable key length.

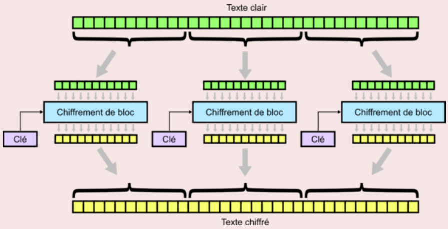


Question

We consider AES-256 (i.e. blocks of 4x4 bytes, 12 rounds). Compared to OTP:

- I have a smaller secret key.
- I have a larger secret key.
- I have a comparable key length.

Electronic Code Book (ECB)



Construction

The message is split into blocks matching the size of Block-Cipher's block length. Each block is encrypted with the same key.
Pros:

- Simplest construction.
- Destination can decrypt a specific block without extra computations.
- Vulnerabilities?

How to evaluate security?

Security property: Semantic security

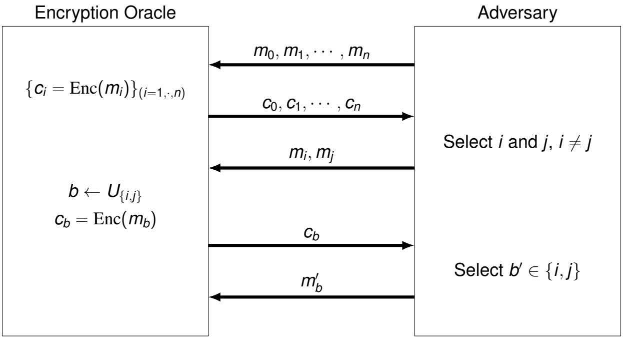
Without information about the key, ciphertext does not leak information about the message.

Adversary capability

Adversary capabilities are defined as indistinguishability games:

- IND-KPA (known plaintext-attack): adversary sees pairs $(m_i, Enc(m_i))$.
- IND-CPA (chosen plaintext-attack): adversary SELECTS messages m_i and ASKS an entity to encrypt m_i.
- IND-CCA: More information during asymmetric encryption lesson.

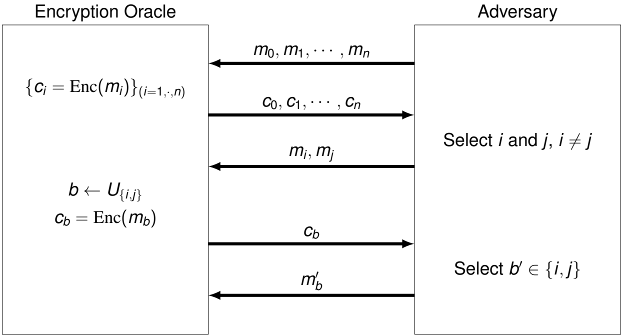
IND-CPA game



Win condition

- Adversary wins the game if: $\Pr[b = b'] > 1/2$.
- If $\Pr[b = b'] = 1/2$, then adversary can only guess randomly which message has been encrypted.
- Advantage: $\mathcal{A}_{CPA} = |\Pr[b = b'] - 1/2| = \epsilon$

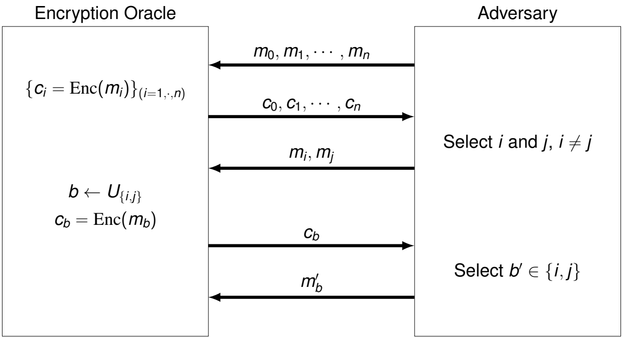
IND-CPA game



Notion of negligible advantage

- For key length k;
- For Advantage $\mathcal{A}_{CPA} = |\Pr[b = b'] - 1/2| = \epsilon(k)$;
- Adversary has negligible advantage if $e(k) < \frac{1}{2^k}$ for all k after given k_0.

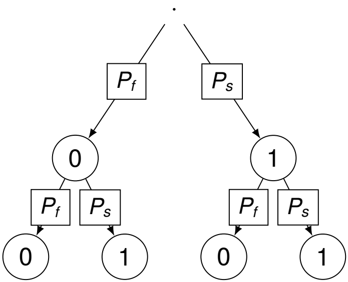
IND-CPA game



Question

If I have an algorithm that provides a very small (say 1/10000) advantage, does this lead to a real distinguability?

INSA TOULOUSE

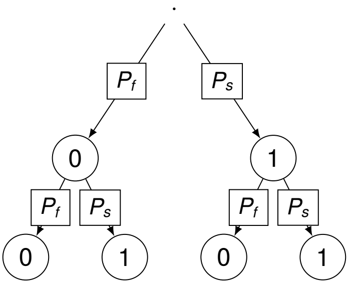


Success probability

P_s = probability of success, P_f = probability of a fail.

Algorithm

If algorithm output the same value twice, I select this value. If values are different, I flip a coin to select one.
By doing so, I can double my success rate. True?

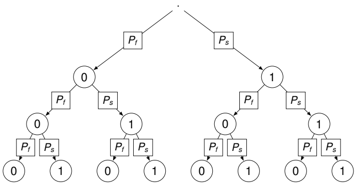


Success probability

$P_s = 0.5 + \epsilon$, $P_f = 0.5 - \epsilon$.

$$P_{success} = P_s^2 + 0.5 \times P_s P_e + 0.5 \times P_e P_s = (0.5 + \epsilon)^2 + (0.5 + \epsilon)(0.5 - \epsilon)$$
$$= 0.5 + \epsilon \quad \text{(fail...)}$$

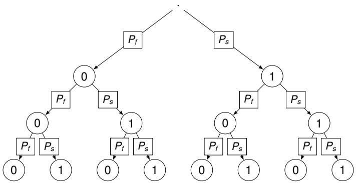
Second try - I run my algorithm three times and I make a vote



Success probability

Better advantage this time?

Second try - I run my algorithm three times and I make a vote

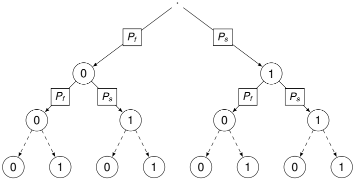


Success probability

$P_s = 0.5 + \epsilon$, $P_f = 0.5 - \epsilon$.

$$\begin{aligned} P_{success} = P_s^3 + 3 \times P_s^2 P_e &= P_s^2 \times (P_s + 3P_e) \\ &= (0.5 + \epsilon)^2 \times (0.5 + \epsilon + 1.5 - 3\epsilon) \\ &= (0.5 + 2\epsilon + 2\epsilon^2) \times (1 - \epsilon) \\ &= 0.5 + 1.5\epsilon - 2\epsilon^3 \ \ \text{(ouf...)} \end{aligned}$$

I run my algorithm N times and I make a vote



Success probability

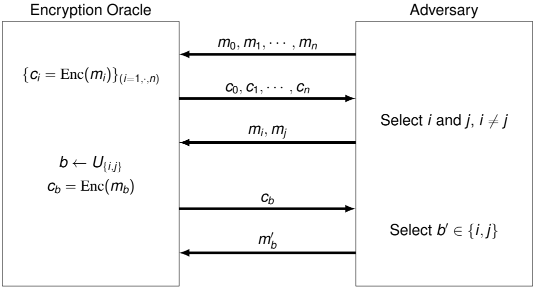
$P_s = 0.5 + \epsilon$, $P_f = 0.5 - \epsilon$.

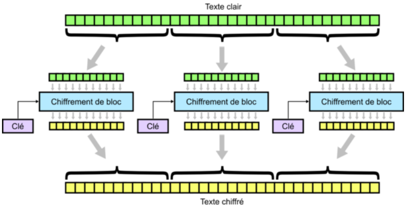
$$P_{success} = \sum_{i=0}^{N/2} \binom{N}{i} P_s^{N-i} P_e^i = P_s^N \times \sum_{i=0}^{N/2} \binom{N}{i} \left(\frac{P_e}{P_s}\right)^i > P_s^N$$

$$P_{success} > (0.5 + \epsilon)^N \sim 0.5 + N\epsilon$$

Conclusion: If I run my algorithm $1/(\epsilon)$, I can distinghuish with probability close to 1.

Go back to ECB mode of operation

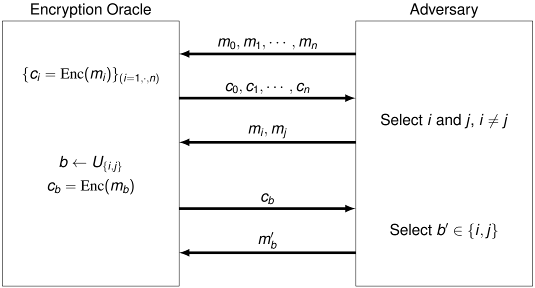


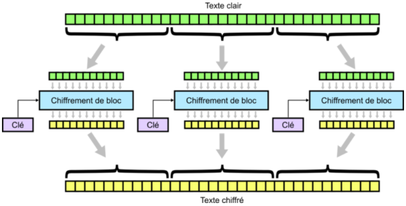


How to win the game?

Which m_i and m_j adversary can select to win?

Go back to ECB mode of operation



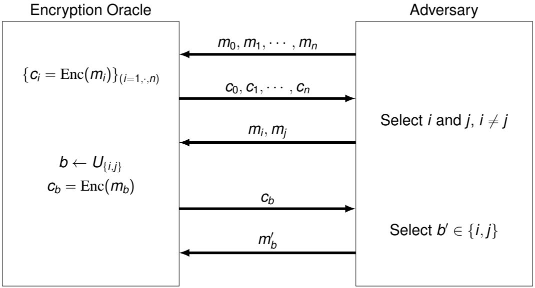


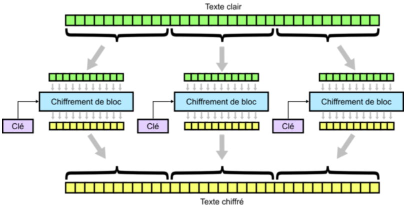
How to win the game?

- $m_i = [\text{Hello }][\text{World }]$
- $m_j = [\text{Hello }][\text{Hello }]$
- $\mathrm{Enc}(m_i) = [c_0][c_1]$,$\mathrm{Enc}(m_j) = [c_0][c_0]$

If encrypted block 0 = encrypted block 1,return *j* else *i*.

Go back to ECB mode of operation

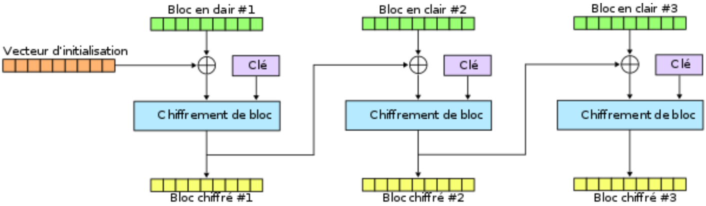




Conclusion

$\mathcal{A}_{CPA} = 1/2$, i.e. adversary always wins!
$\implies$ ECB mode is trivially insecure under IND-CPA game and should not be used in practice.

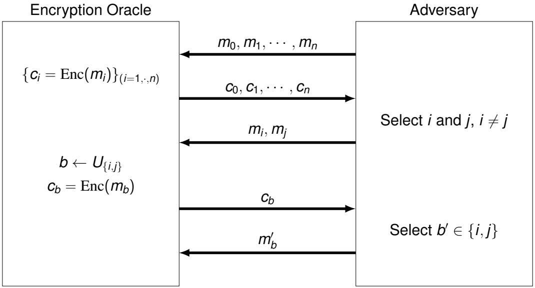
Cipher Block Chaining (CBC)

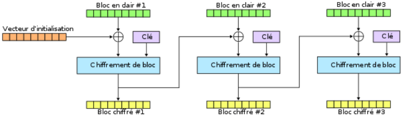


Construction

- Also called nonce-based encryption;
- Initialization Vector (IV = nonce) is XORed with input massage block, and chained with next input massage block;
- How I select a secure nonce?

Back to IND-CPA game - Case of CBC mode

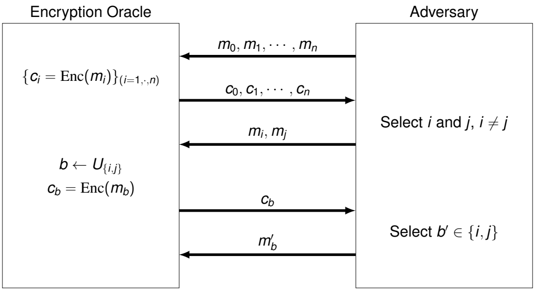


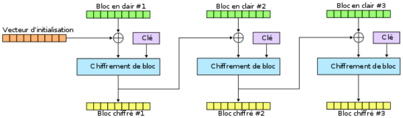


Under free nonce, how to win the game?

Which m_i and m_j adversary can select to win?

Back to IND-CPA game - Case of CBC mode





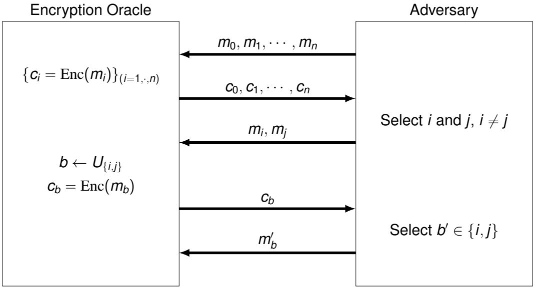
Under free nonce, how to win the game?

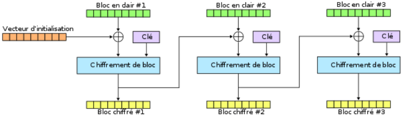
Adversary ask for encryption of two plaintexts differents, say:

- $m_i = [\mathrm{Hello}\ ]$, $m_j = [\mathrm{World}\ ]$
- $\mathrm{Enc}(m_i) = [c_i]$,$\mathrm{Enc}(m_j) = [c_j]$

then choose $[\mathrm{Hello}\ ]$ and $[\mathrm{World}\ ]$ as challenges.

Back to IND-CPA game - Case of CBC mode

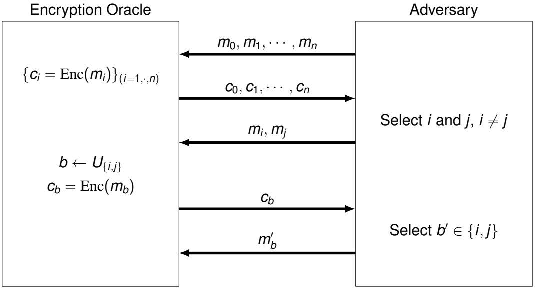


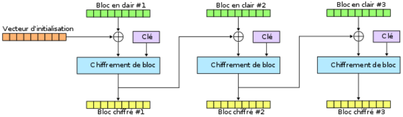


Conclusion

Which nonce may I choose?

Back to IND-CPA game - Case of CBC mode

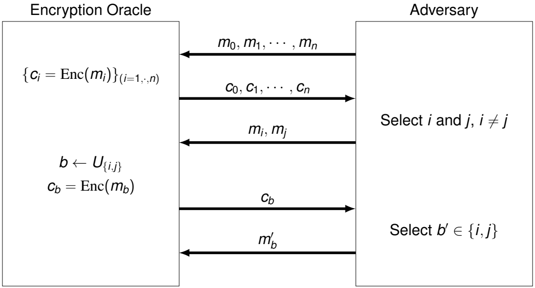


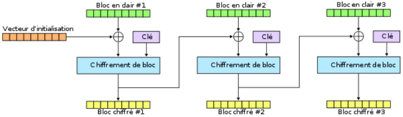


Case 1 - random, secret but repeated nonce

Nonce is selected at random at the start of communication and kept secret from adversary. Secure?

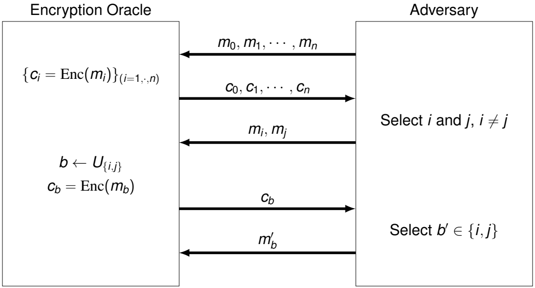
Back to IND-CPA game - Case of CBC mode

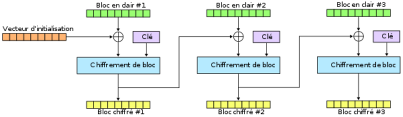




Case 1 - random, secret but repeated nonce

Still not CPA secure since adversary can select m_i and m_j before challenge and requests $c_i = \mathrm{Enc}(m_i)$ and $c_j = \mathrm{Enc}(m_j)$.

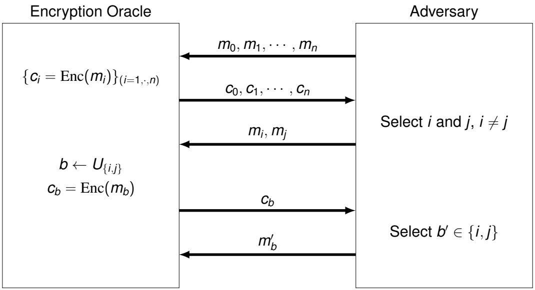


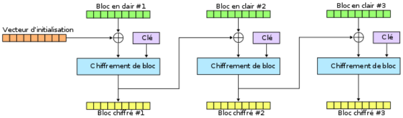


Case 1 - Conclusion

Nonce should not be used twice.

Back to IND-CPA game - Case of CBC mode

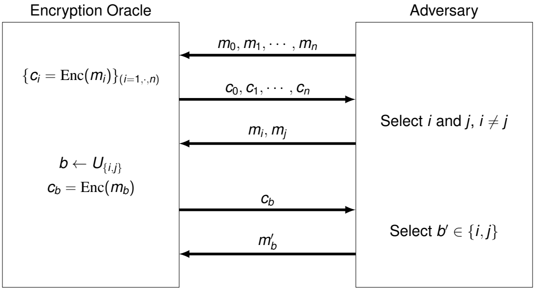


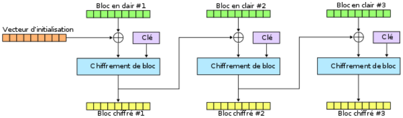


Case 2 - randomized, public but predictible

- Nonce is firstly selected at random.
- For next message, we just continue the chaining, i.e. last cipher block is taken as the new nonce. Secure? (case of TLSv1.0).

Back to IND-CPA game - Case of CBC mode

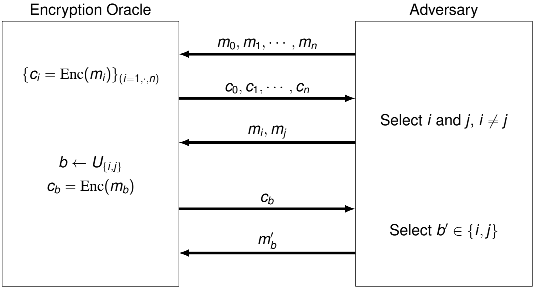


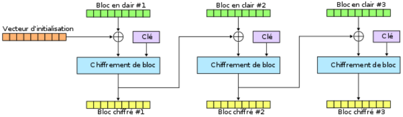


Case 2 - randomized, public but predictible

Select m_i such as $m_i = IV_{n-1}$ = last encrypted block
$\Longrightarrow$ first block is the encryption of 0 under a free nonce.
$\Longrightarrow$ deterministic.

Back to IND-CPA game - Case of CBC mode

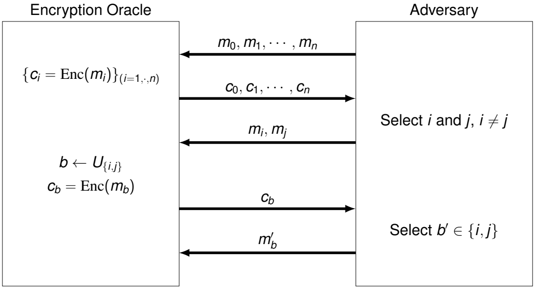


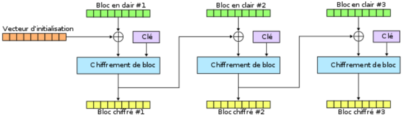


Case 2 - Conclusion

Nonce must not be predictible by adversary.

Back to IND-CPA game - Case of CBC mode

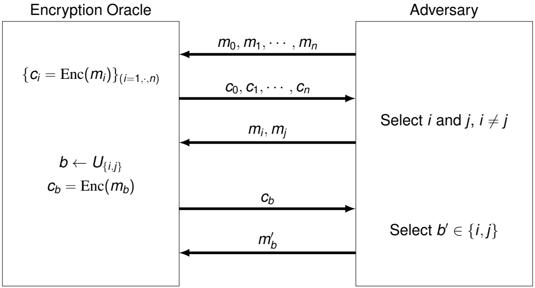


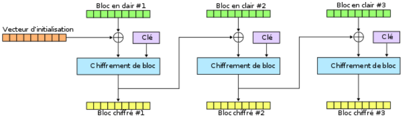


Case 3 - Random and unpredictible

Secure?

Back to IND-CPA game - Case of CBC mode

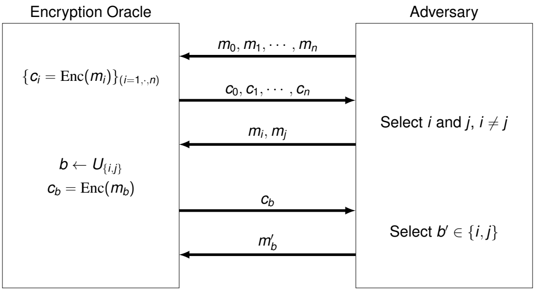




Case 3 - Random and unpredictible

Secure, but be carefull, you must send secretly to your corresponding the nounce used for next encryption and ensure integrity.

And what about the key? How often I must rene



CBC - theorem

For any length $L > 0$:
If PRP E is semantically secure over (K,X), then E used in CBC mode (E_{CBC}) is semantically secure under CPA over (K,X^L,X^{L+1}).
For adversary making q-query, then:

$$\mathcal{A}(E_{CBC}) \leq 2\mathcal{A}(E) + q^2L^2/|X|$$

Where $|X|$ is the number of outputs possible for the permutation and L the

Numerical application

Case of AES

- size of AES output: 128 bits;
- Target advantage: 2^{-80}.

Upper bound of encrypted blocks?

Case of AES

- size of AES output = 128 bits $\implies |X| = 2^{128}$;
- Target advantage = $2^{-80} \implies q^2L^2/|X| = 2^{-80}$;
- $qL = \sqrt{2^{-80+128}} = 2^{24}$ encrypted blocks.

Conclusion: We must renew the key before reaching 2^{28} bytes of encrypted data, i.e. 256 MB.

Numerical application

Case of AES

- size of AES output: 128 bits;
- Target advantage: 2^{-80}.

Upper bound of encrypted blocks?

Case of AES

- size of AES output = 128 bits $\implies |X| = 2^{128}$;
- Target advantage = $2^{-80} \implies q^2 L^2 / |X| = 2^{-80}$;
- $qL = \sqrt{2^{-80+128}} = 2^{24}$ encrypted blocks.

Conclusion: We must renew the key before reaching 2^{28} bytes of encrypted data, i.e. 256 MB.

Next lesson

How to ensure integrity?