Part I

Block Ciphers

Block Ciphers

Perform operations on strings of bits of fixed size.

$$\text{Enc}: \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$$
$$\text{Dec}: \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$$

Example

Cipher	block size (bit)	key size (bit)
DES	64	56
3DES	64	168
AES	128	128, 192, 256

Generally slower than stream ciphers, but the security properties are generally better understood.

1. Block Ciphers

Keyed Functions

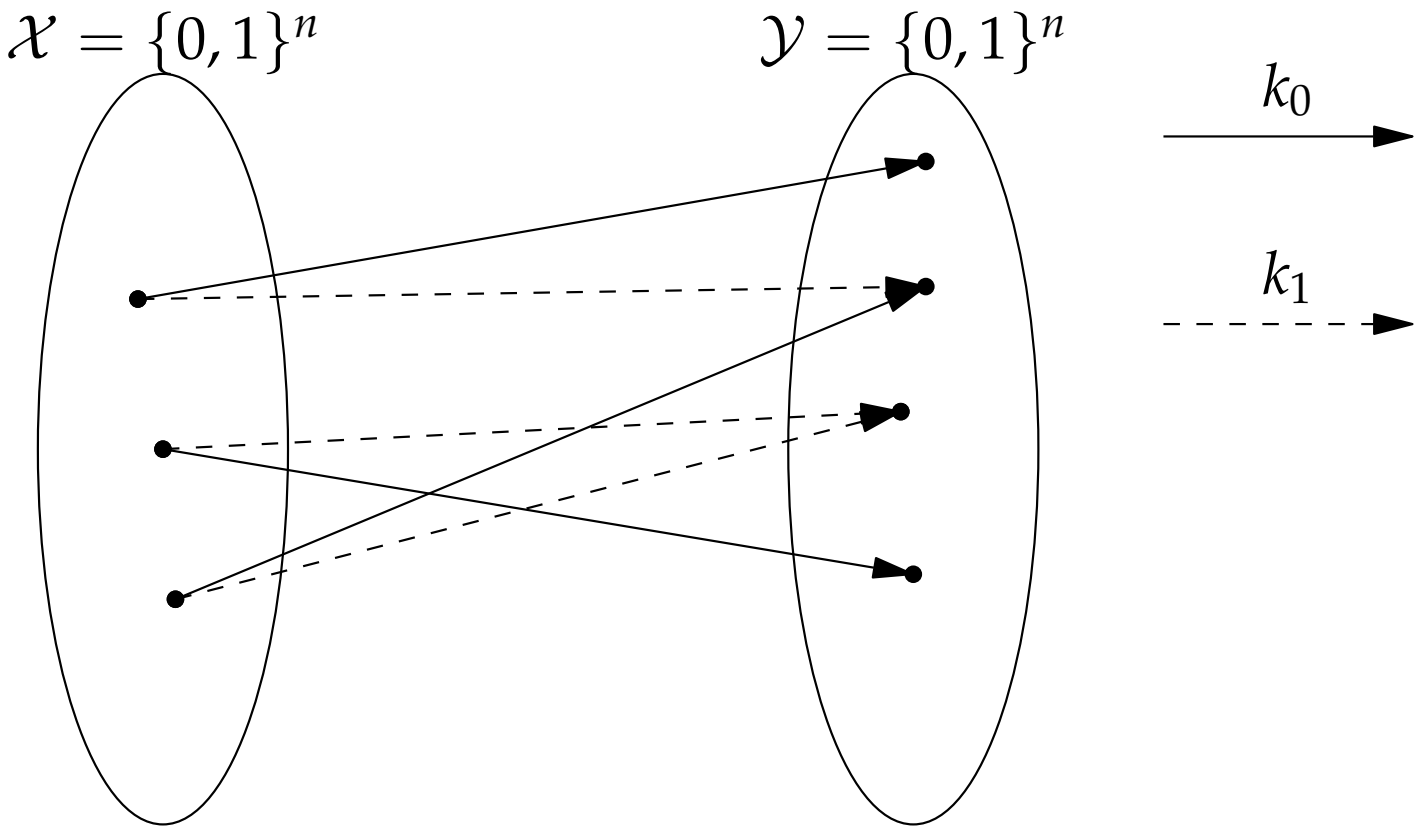
Definition (Keyed Function)

A family of functions

$$F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$$

such that there exists an efficient algorithm to evaluate $y = F(k, x)$.

Keyed Functions



Keyed Permutation

Definition (Keyed Permutation)

A family of functions

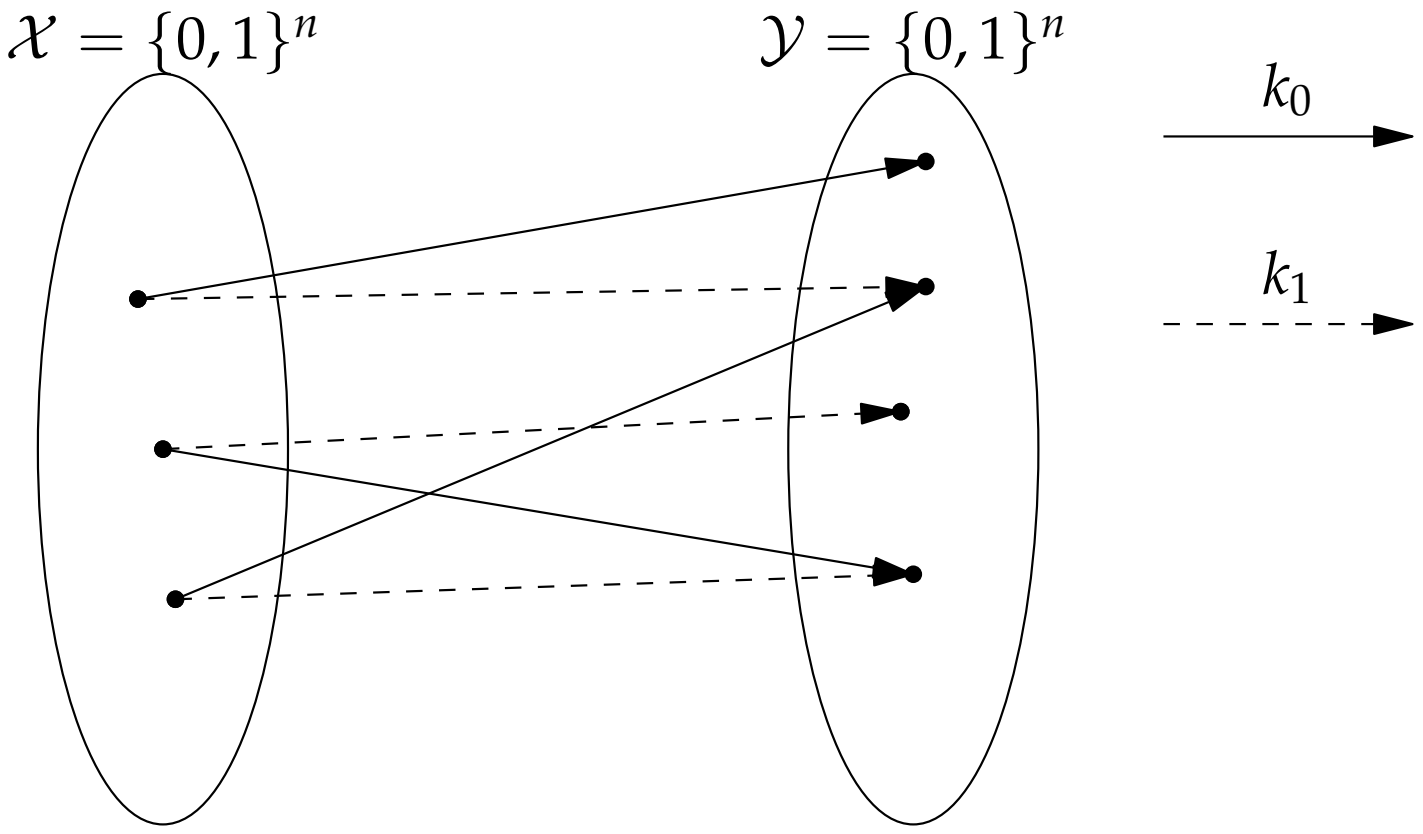
$$E : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$$

such that

1. there exists an efficient deterministic algorithm to evaluate $y = E(k, x)$
2. the function $E(k, x)$ is one-to-one
3. there exists an efficient inversion algorithm $x = D(k, y)$

Note that a keyed permutation is a special case of a function. Block ciphers are permutations.

Keyed Permutations



Random Functions

A random function is a function chosen randomly from the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$. In practice, it is the same as a random oracle:

- the first time the function sees a given input x, it chooses a random output y and stores it
- the following times that x is seen, the function yields the same y

A random permutation is a permutation chosen randomly from the set of all permutations from $\mathcal{X}$ to $\mathcal{Y}$. In practice, it is similar to a random function, with the additional constraint that two distinct inputs map onto two distinct outputs.

Secure Block Cipher

A secure block cipher is a keyed permutation that looks like a random permutation.

A block cipher $E(k, m)$ with a key having λ bits and a plaintext/ciphertext of n bits, can produce 2^λ possible permutations. The total number of random permutations is $(2^n!)$.

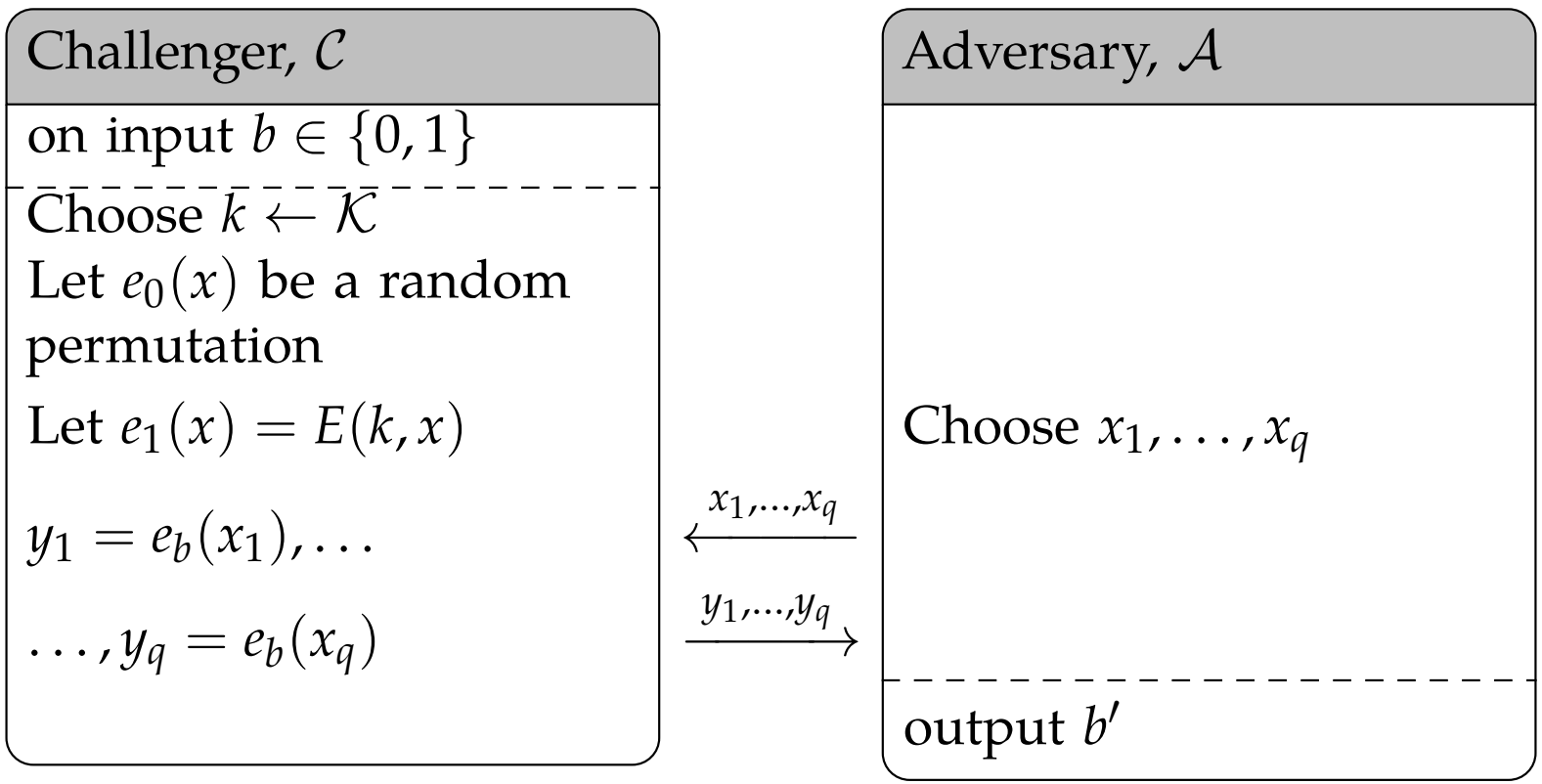
Usually, $\lambda \geq n$, so the number of permutations that can be obtained from E is a tiny fraction of the total number of permutations.

We say that the block cipher is secure is the adversary cannot tell whether a given permutation is taken from E for some key, or it is random.

Secure Block Cipher

Non-Adaptive q-query Adversary

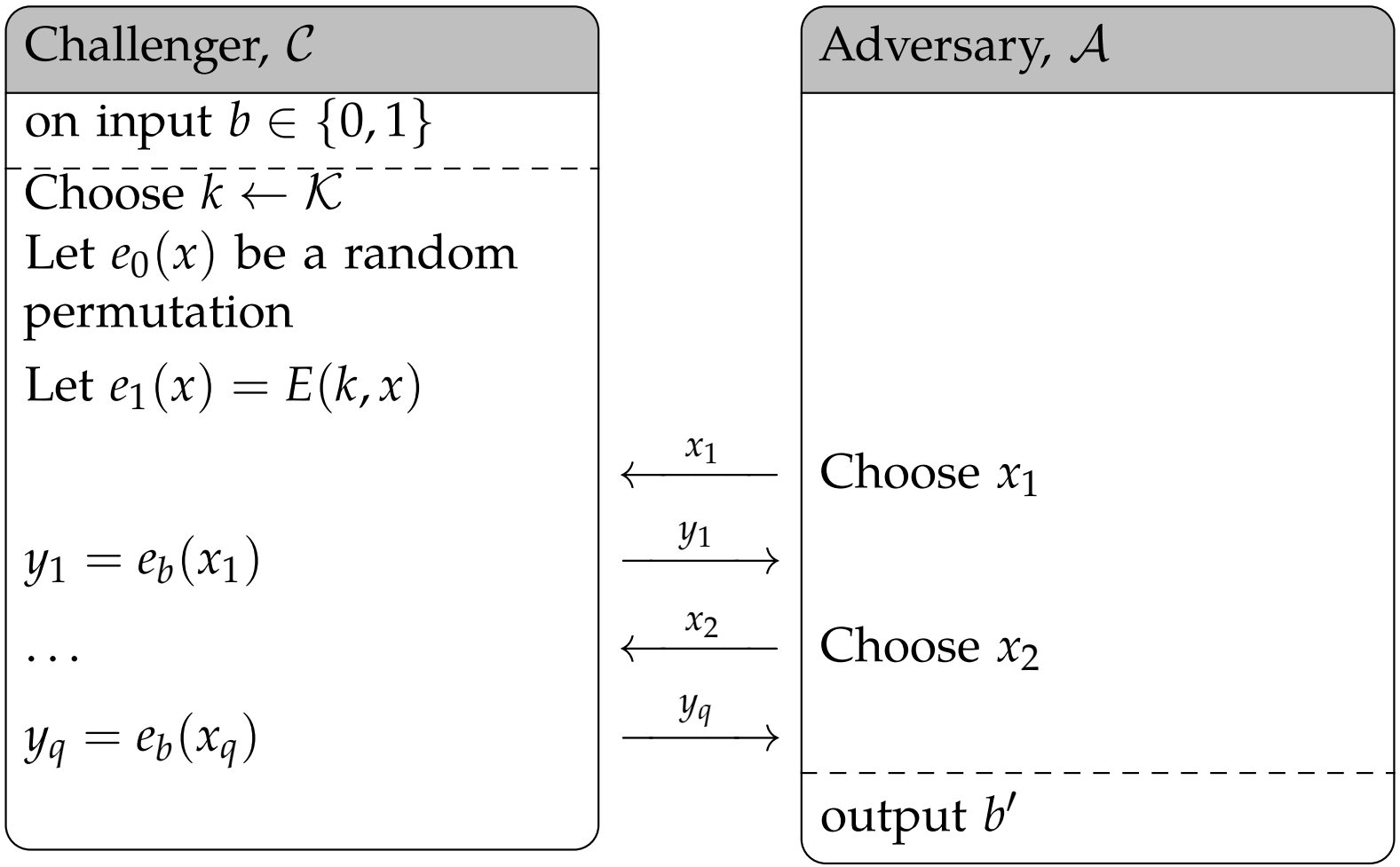
The non-adaptive adversary chooses q queries before interacting with the challenger.



Secure Block Cipher

Adaptive q-query Adversary

The adaptive adversary chooses a new query after seeing the answer to the previous one.



Secure Block Cipher

$E(k, x)$ is secure if it is computationally indistinguishable from a random permutation, i.e., for all efficient Adversaries, the advantage

$$\text{Adv} = |\Pr[\text{Exp}(0) = 1] - \Pr[\text{Exp}(1) = 1]|$$

is negligible.

Exhaustive Search with DES

1997: Internet search – 3 months

1998: EFF machine (deep crack) – 3 days (250K$)

1999: combined search – 22 hours

2006: COPACOBANA (120 FPGAs) – 7 days (10K$)

56-bit ciphers should not be used !

1. Block Ciphers

General Principles

A good block cipher has the following properties (Shannon):

Confusion: each bit of the ciphertext depends on all the bits of the key.

Diffusion: each bit of the ciphertext depends on all the bits of the plaintext. We expect that if we change one bit of the plaintext, half of the bits of the ciphertext change.

The basic bulding blocks of block ciphers are:

Substitution takes as input a group of bits and outputs a new group of bits.

Permutation shuffles the bits in a group.

The AES Block Cipher

Advanced Encryption Standard (Rijman, Daemen 1998) is a substitution-permutation network. It comprises multiple rounds composed of substitution layers and permutation layers.

AES Variants

Name	Block size	Key Size	Rounds
AES-128	128	128	10
AES-192	128	192	12
AES-256	128	256	14

AES Rounds

Each round is made of

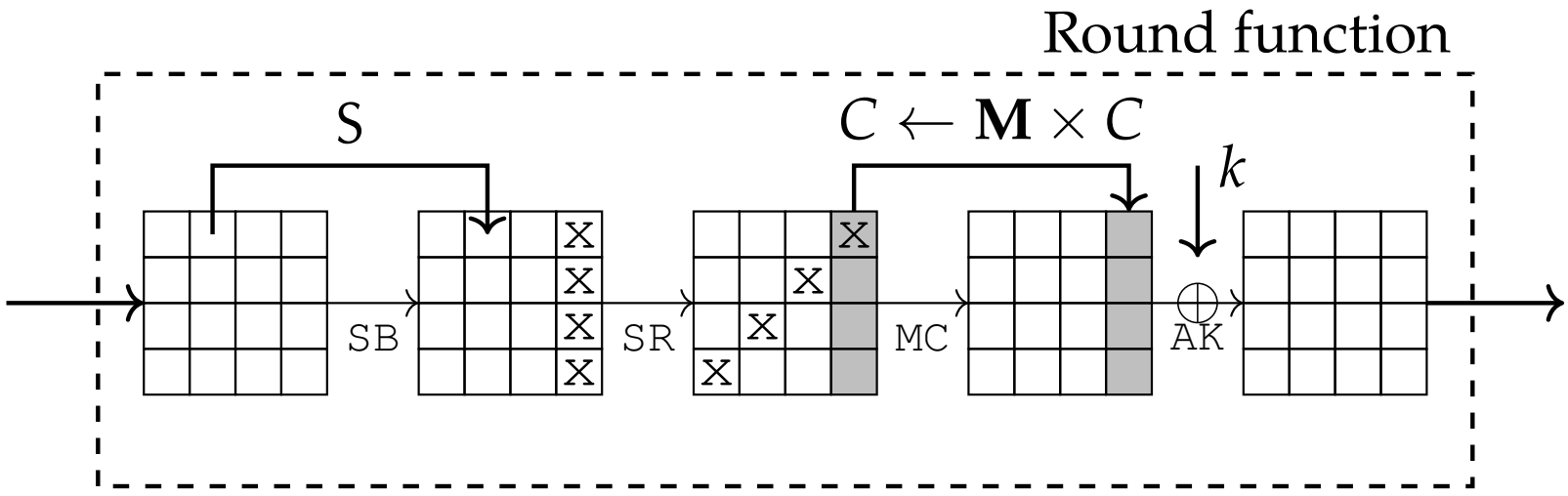
- mixing the input block with the key (diffusion)
- non-linear substutition of octects with other octects by means of a fixed nonlinear function
- permutation of the bits by rows and columns (confusion)

The AES is parallelizable in hardware and permits different space-time tradeoffs.

AES-128 Rounds

function AES-128(k, m)
  $(k_0, \ldots, k_{10}) = \text{ExpandKey}(k)$
  $s = m$ ▷ s is the state
  $s = s \oplus k_0$ ▷ AddRoundKey
  for $i = 1$ **to** 10 **do** ▷ Round function
    $s = \text{SubBytes}(s)$
    $s = \text{ShiftRows}(s)$
    if $r \leq 9$ **then** ▷ No MC in the final round
      $s = \text{MixColumns}(s)$
    end if
    $s = s \oplus k_i$ ▷ AddRoundKey
  end for
  return s
end function

AES Round Function



Some of the operations on the state can be described by using finite field algebra.

Finite Fields from Polynomial Quotient Rings

Let p be a prime and $r(x)$ be an irreducible polynomial with coefficients in $\mathbb{Z}_p$ and degree k.

$\mathbb{Z}_p[X]/r(X)$ is the set of all polynomials of degree less than k along with the addition and multiplication defined modulo $r(X)$.

$\mathbb{Z}_p[X]/r(X)$ is a representation of the finite field $GF(p^k)$. All elements $a(x) \neq 0$ have an inverse $a(x)^{-1}$.

$$a(x)a(x)^{-1} \bmod r(x) = 1$$

SubBytes

SubBytes operates independently over each byte of the State matrix. The substitution function is the same for all elements. The substitution table is always precomputed. However it is interesting to discuss how it is obtained.

1. Consider the finite field $GF(2^8)$ represented as $\mathbb{Z}_2[x]/x^8 + x^4 + x^3 + x + 1$.
2. The input byte is interpreted as an element $a(x) \in GF(2^8)$.
3. We calculate

$$y(x) = \begin{cases} 0 & \text{if } a(x) = 0 \\ a^{-1}(x) & \text{otherwise} \end{cases}$$

4. $y(x)$ is not a good substitution because it has fixed points ($a = 0, a = 1$). Therefore, we further apply an affine transformation to $y(x)$.

ShiftRows and MixColumns

ShiftRows makes the State rows rotate with different speeds depending on the row.

$$(c_{i,j}) = (b_{i,(j+i) \bmod 4})$$

In MixColumns the matrix bytes are again interpreted as elements $c_{i,j}(x) \in GF(2^8)$. The output is obtained with a dot-product.

$$(d_{i,j}) \begin{pmatrix} x & x+1 & 1 & 1 \\ 1 & x & x+1 & 1 \\ 1 & 1 & x & x+1 \\ x+1 & 1 & 1 & x \end{pmatrix} (c_{i,j}) \bmod x^8 + x^4 + x^3 + x + 1$$

Round Key

The round key is obtained by expanding the key. We describe the process in AES-128.
Let $W(n)$ be a sequence of column vectors of 4 bytes each.

- $W(n)$ for $n = 0, \ldots, 3$ is the AES key.
- For $n \geq 4$,

$$W(n) = \begin{cases} W(n-4) \oplus W(n-1) & \text{if 4 does not divide } n \\ W(n-4) \oplus T(W(n-1)) & \text{if 4 divides } n \end{cases}$$

The function T performs a rotation, a substitution with SubBytes, and adds a round constant.

The key for the rth round is

$$(k_{i,j}) = \begin{pmatrix} W(4r) & W(4r+1) & W(4r+2) & W(4r+3) \end{pmatrix}$$

Best Attacks

- Best key recovery attack: four times better than exhaustive search (2011)
- Best related key attack on AES-256 (2009): Given 2^{99} PT/CT pairs from four related keys in AES-256 can recover keys in time $\approx 2^{99}$

1. Block Ciphers

Attack Scenario

Adversary's capabilities: access to a single ciphertext (one time key)

Adversary's goal: obtain information about the plaintext (semantic security)

Note that a single pair of PT/CT can be made of multiple blocks.

Example: email with session key.

Notation

In the following we will use the following notation.

- Let P be the plaintext.
- Let $E(k, m)$ a secure block cipher with block size n.
- P is split in L blocks of size n, $P = P_1 \| P_2 \| \ldots \| P_L$. We assume that there is no need to perform padding.
- The encryption of the full PT with key k is $\mathrm{Enc}(k, P)$. The algorithm used to calulate $\mathrm{Enc}(k, P)$ using multiple times the block cipher $E(k, m)$ is called *a mode of operation*.

Incorrect use of a block cipher

The ECB mode of operation

In Electronic Codebook (ECB), the message is divided in blocks. Each block is encrypted independently.

Definition (ECB encryption)

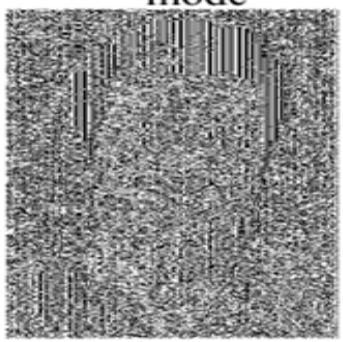
$$C_i = E(k, P_i) \quad 1 \leq i \leq L$$
$$C = C_1 \| C_2 \| \ldots \| C_L = \mathsf{Enc}(k, P)$$

Problem: $P_i = P_j \iff C_i = C_j$.

ECB fail

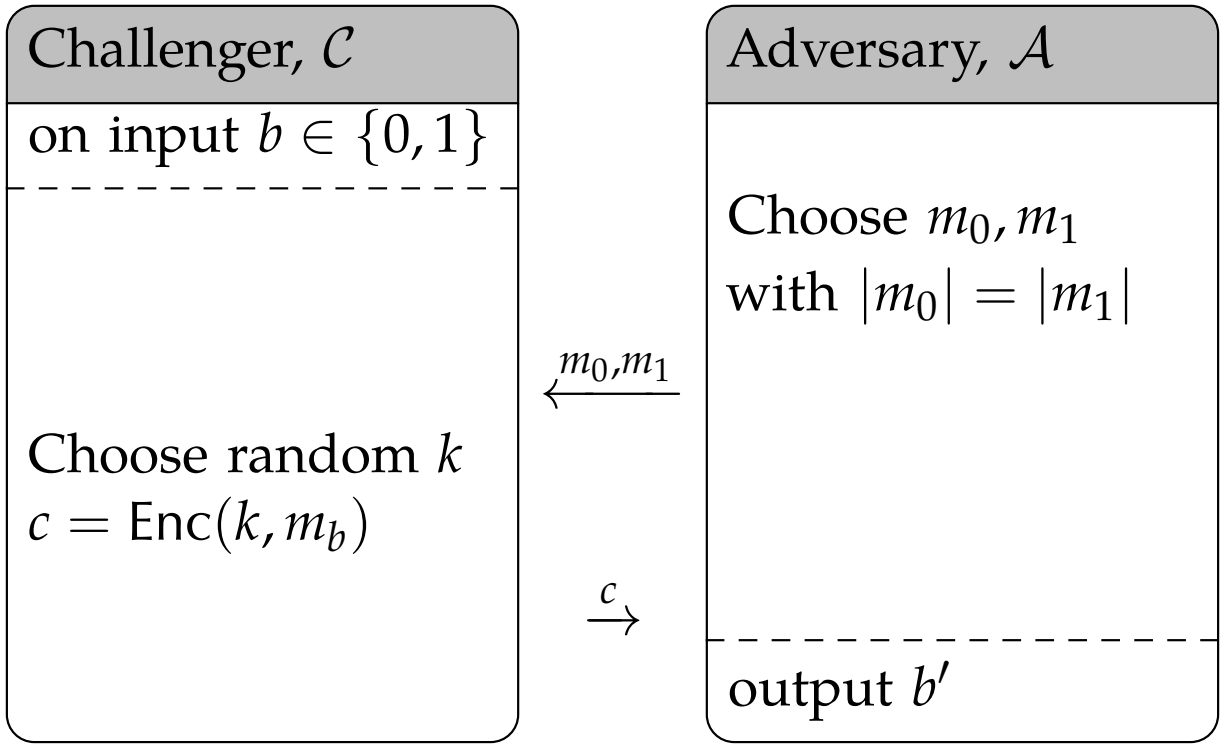
An example plaintext

Encrypted with AES in ECB mode



Semantic Security (one-time key)

The definition of Semantic Security in the one-time key case is identical to the Stream Cipher case.



$$\mathrm{Adv}[\mathcal{A}] = |\Pr[\mathrm{Exp}(0) = 1] - \Pr[\mathrm{Exp}(1) = 1]|$$

must be negligible for all efficient $\mathcal{A}$.

ECB is not secure

If we find at least one algorithm $\mathcal{A}$ that has non negligible advantage $\mathrm{Adv}[\mathcal{A}]$, then we have proved that ECB is not secure.

Challenger, $\mathcal{C}$		Adversary, $\mathcal{A}$
Choose random k		Choose any P_1, P_2
		Choose $m_0 = P_1 \| P_2$
		and $m_1 = P_1 \| P_1$
	$\xleftarrow{m_0, m_1}$	
$c = \mathrm{Enc}(k, m_b)$		
	$\xrightarrow{c}$	Let $C_1 \| C_2 = c$
		if $C_1 = C_2$ output 1
		otherwise output 0

This algorithm always guesses right!

$$\mathrm{Adv}[\mathcal{A}] = |0 - 1| = 1$$

A secure construction: Deterministic Counter Mode

Definition (DetCTR encryption)

With P be a plaintext and $F(k, m)$ a secure pseudorandom function.

$$C_i = P_i \oplus F(k, i) \quad 1 \leq i \leq L$$
$$C = C_1 \| C_2 \| \ldots \| C_L = \mathsf{Enc}(k, P)$$

Theorem

If F is a secure pseudorandom function against a L-query adversary, then Enc *is a semantically secure cipher.*

Switching lemma. A secure block cipher can be used as a secure pseudorandom function if $L^2/2^{n+1}$ is negligible (think of the birthday attack).

1. Block Ciphers

Attack Scenario

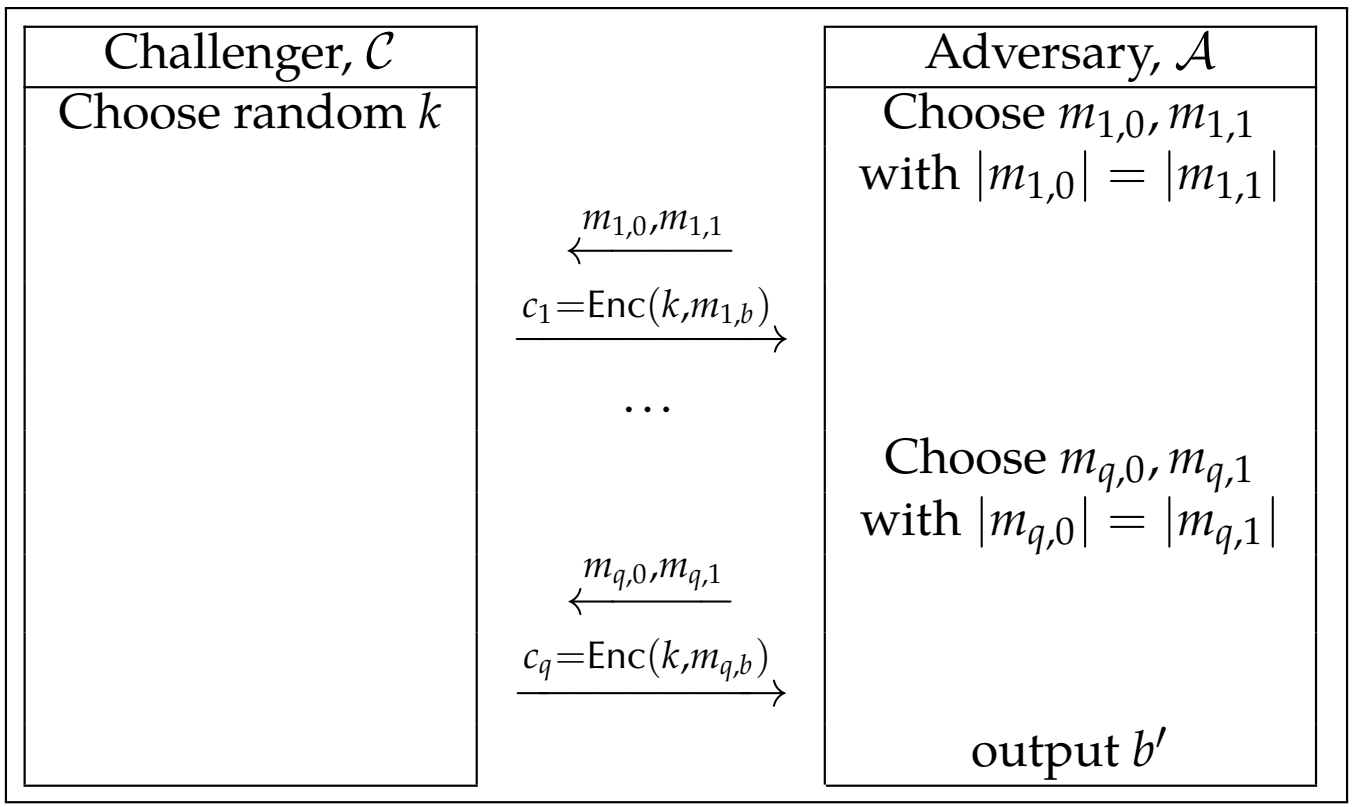
Adversary's capabilities: can obtain encryption of several PT (Chosen Plaintext Attack, CPA)

Adversary's goal: obtain information about a PT (semantic security)

Example: files on disk, packets in WiFi

Semantic Security (many-time key)

The definition of Semantic Security in the one-time key case is slightly different. We assume that the Adversary can obtain the encryptions of q messages before answering the challenge.

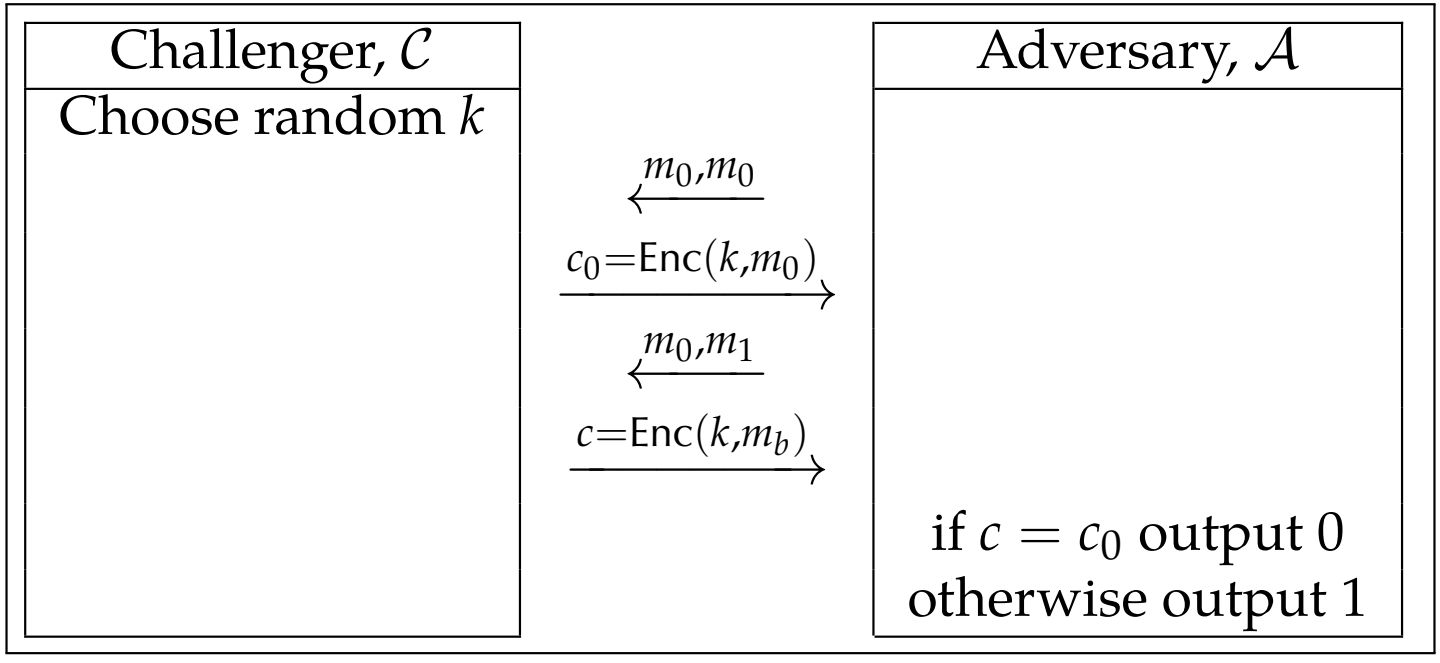


$$\mathrm{Adv}_{\mathrm{SS}}[\mathcal{A}, \mathcal{E}] = |\Pr[\mathrm{Exp}(0) = 1] - \Pr[\mathrm{Exp}(1) = 1]|$$

should be negligible for all efficient $\mathcal{A}$.

Deterministic Encryption is not SS under CPA

Suppose that $\mathsf{Enc}(k, m)$ always outputs the same CT for the same pair (k, m). Then



The attacker can learn that two packets are the same, etc. Big problems if the message space is small.

Nonce-based Encryption

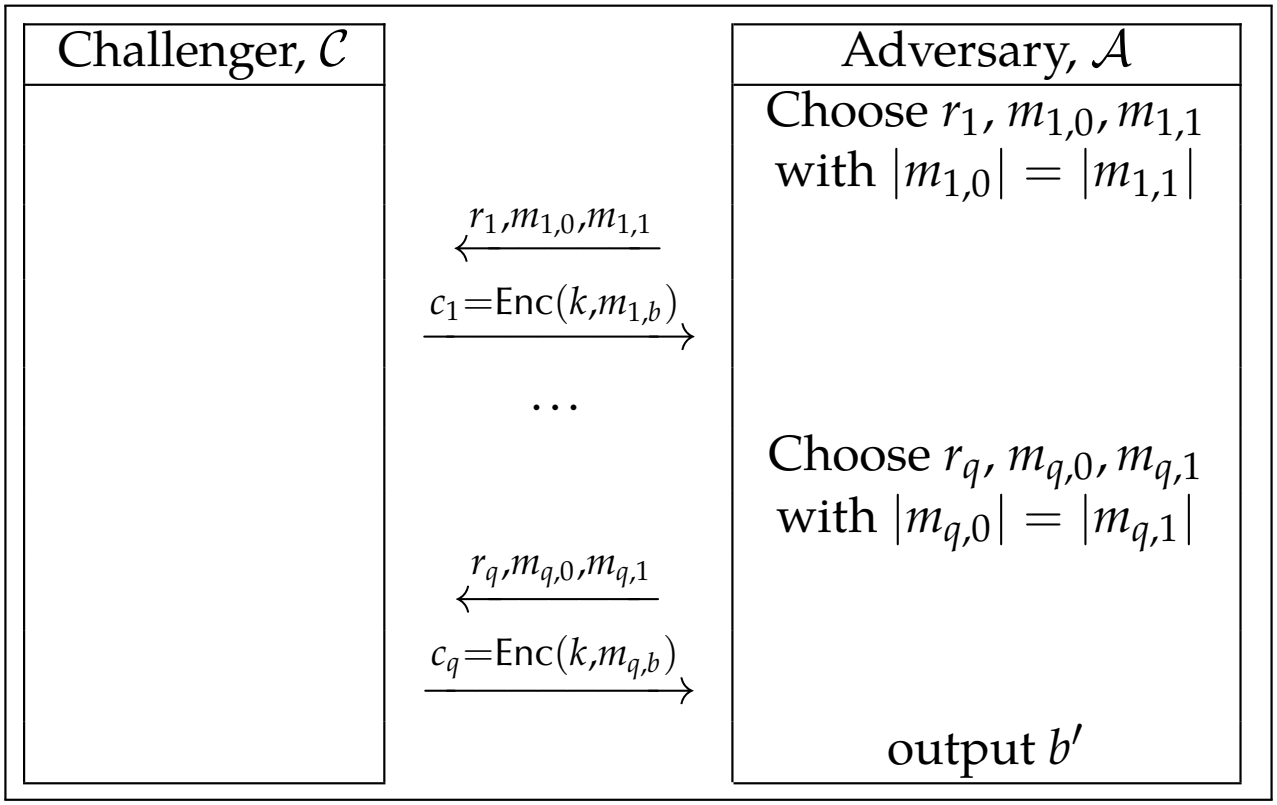
$\mathsf{Enc}(k, m)$ becomes $\mathsf{Enc}(k, m, r)$, where the nonce r is a value that changes from message to message and a given pair nonce is never used twice with the same key.

The nonce can be

1. a random value chosen by the sender and sent with the message
   - Con: increases the message size.
   - Pro: stateless and robust to losses.
2. a counter
   - Con: requires that sender and receiver keep a synchronized state.
   - Pro: does not increase the message size.

Semantic Security for nonce-based encryption

We let the adversary choose the nonce, which must be different for each encryption.



All nonces $r_1, \ldots, r_q$ are distinct.

Randomized CBC

Let $E(k, m)$ a secure block cipher and $D(k, c)$ its inverse.

Definition (CBC with Random Nonce)

Choose random nonce r

$$
\begin{aligned}
C_1 &= r \oplus E(k, P_1) \\
C_i &= C_{i-1} \oplus E(k, P_i) \quad 2 \leq i \leq L \\
C &= r \| C_1 \| C_2 \ldots \| C_L
\end{aligned}
$$

Decription is trivial: $P_i = D(k, C_i) \oplus C_{i-1}$.
Two equal blocks are encrypted differently with high probability.

Security of Randomized CBC

Theorem (CBC Theorem)

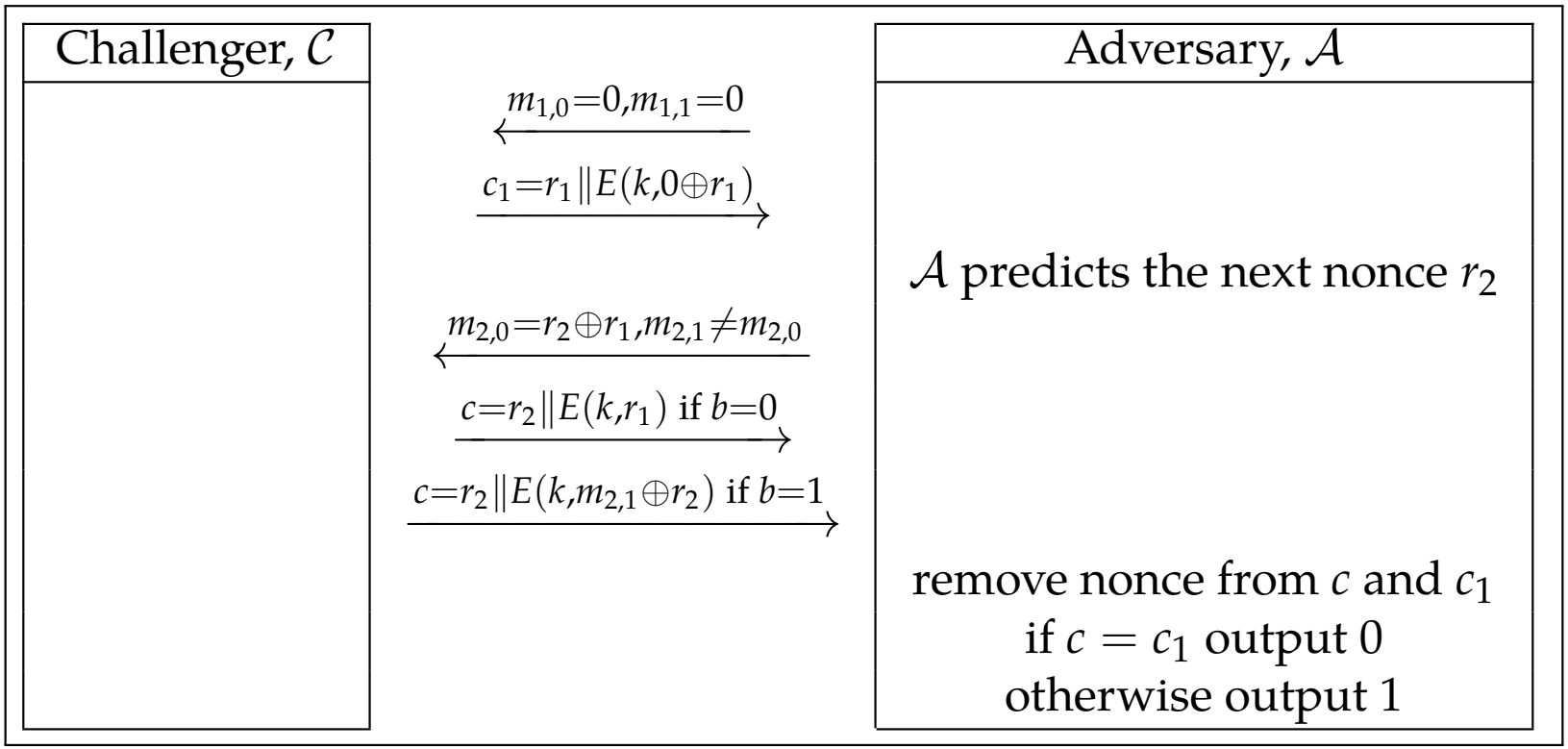
If E is a secure block cipher with block size n, then Randomized CBC is a semantically secure cipher against a q-query adversary for messages of size L as long as $(qL)^2/2^n$ is negligible.

The number qL is the number of encrypted blocks after which we must change the key.
Example: let's say that negligible is 2^{-32}. Then:

- with AES, $n = 128$, then $qL < 2^{48}$.
- with DES, $n = 64$, then $qL < 2^{16}$.

Warning: CBC with predictable nonce

CBC with predictable nonce is not secure.
In fact, we can build an attack:



CBC technicalities

To prevent predictability with non-random (e.g. user-supplied) nonce encrypt it *with a different key* before use.

Padding is necessary because the message length may not be an integer multiple of the block size.
Suppose $n > 0$ bytes of padding are necessary. Then, the encrypted message becomes:

$$P \| \underbrace{n \| \ldots \| n}_{n \text{ times}}$$

If $n = 0$, add a full padding block.

Nonce based CTR

Let $F(k, m)$ a secure pseudorandom function.

Definition (CTR with random nonce)

Choose random nonce r

$$C_i = P_i \oplus E(k, r + i - 1) \quad 1 \leq i \leq L$$
$$C = r \| C_1 \| C_2 \ldots \| C_L$$

Pros: parallelizable, no padding necessary (truncate last block).

It is not enough that nonces are not repeated. The intervals between the each nonce r_j and $r_j + L$ must not overlap. It can be guaranteed by multiplying the nonce by the maximum message size. (This is always necessary is nonce is user-supplied).

$$r = \underbrace{nonce}_{n - \log_2 L} \| \underbrace{0 \ldots 0}_{\log_2 L}$$

Security of Randomized CTR

Theorem (CTR Theorem)

If E is a secure block cipher, then Randomized counter mode is semantically secure for messages of size L as long as $q^2 L / 2^n$ is negligible.

Note that it is better than CBC.
Example: let's say that negligile is 2^{-32}. Then with AES, $|X| = 128$, then $q\sqrt{L} < 2^{48}$. For example 2^{32} messages each of size 2^{32}.