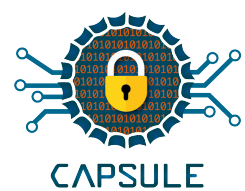
Introduction to Cryptography
Part V: Symmetric Cryptography

André Schrottenloher

Inria Rennes
Team CAPSULE



[0]

Primitives

Primitives

- A symmetric crypto **primitive** is a **small-scale function** (block cipher, compression function) providing **basic security guarantees**.
- It cannot be used "as is" for encryption, authentication, hashing, etc.

- Primitives are combined using **modes of operation** to offer high-level functionalities.

Compression functions

A compression function:

$$H : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n$$

should "behave" as a random function.

- Collision resistance $\implies$ $n/2$ bits
- Preimage resistance $\implies$ n bits
- Second-preimage resistance $\implies$ n bits

The complexities depend on the **size of the output**.

Block ciphers

Let $\mathcal{K} = \{0,1\}^k$ be a key space and $X = \{0,1\}^n$ be a plaintext space.

A **block cipher** E is a family of **permutations** $X \to X$, indexed by $\mathcal{K}$, such that: for any $K \in \mathcal{K}$, E_K and E_K^{-1} can be computed in polynomial time.

- k is the **key size** and n is the **block length**
- Typically n ranges from 64 (small) to 256 (large) and k from 80 (small) to 256
- AES: $n = 128$, $k \in \{128, 192, 256\}$

Block cipher security: PRP

A secure block cipher should be a **pseudo-random permutation** (PRP).

The **PRP security game** is played between a **distinguisher** $\mathcal{D}$ and a **challenger** $\mathcal{C}$:

- **Initialization:** $\mathcal{C}$ samples $K \hookleftarrow U(\mathcal{K})$ and $b \hookleftarrow U(0,1)$, and samples a random permutation Π over $\{0,1\}^n$
- **Queries:** $\mathcal{D}$ can ask **encryption** queries.
  If $b = 0$ (RAND) : $\mathcal{C}$ replies using Π
  If $b = 1$ (PRP) : $\mathcal{C}$ replies using E_K
- $\mathcal{D}$ computes a bit b', returns, and wins if $b' = b$.

The advantage of $\mathcal{D}$ is:

$$\mathrm{Adv}^{PRP}(\mathcal{D}) = |\Pr[\mathcal{D} \text{ Win}] - 1/2| \ .$$

Block cipher security: strong PRP

A secure block cipher should be a **strong pseudo-random permutation** (PRP).

The **SPRP security game** is played between a **distinguisher** $\mathcal{D}$ and a **challenger** $\mathcal{C}$:

- **Initialization:** $\mathcal{C}$ samples $K \hookleftarrow U(\mathcal{K})$ and $b \hookleftarrow U(0,1)$, and samples a random permutation Π over $\{0,1\}^n$
- **Queries:** $\mathcal{D}$ can ask **encryption or decryption** queries.
  If $b = 0$ (RAND) : $\mathcal{C}$ replies using Π or Π^{-1}
  If $b = 1$ (PRP) : $\mathcal{C}$ replies using E_K or E_K^{-1}
- $\mathcal{D}$ computes a bit b', returns, and wins if $b' = b$.

The advantage of $\mathcal{D}$ is:

$$\mathrm{Adv}^{PRP}(\mathcal{D}) = |\Pr[\mathcal{D} \text{ Win}] - 1/2| \ .$$

Summary

PRP = cannot distinguish E_K (with random K) from a random permutation.

SPRP = cannot distinguish E_K, E_K^{-1} (with random K) from a random permutation.

Generic security

A strategy that always work for the (S)PRP game: try all possible keys. This takes a time 2^k.

Block cipher cryptanalysis mostly studies **key-recovery attacks**, i.e., solve the following problem:

Given access to E_K (and E_K^{-1}) for some unknown key K, find K.

- With access to E_K: **chosen-plaintext attack**
- With access to E_K and E_K^{-1}: **chosen-ciphertext attack**

A strategy that always works:

- Ask for a few plaintext-ciphertext pairs (P, C)
- Guess the key K'
- Check if the plaintext-ciphertext pairs match the expected values $(E_{K'}(P) = C)$

Security requirements in symmetric crypto

When we design a new block cipher, the main security requirement is that:

There should not exist a key-recovery attack better than brute force.

- For example, AES-128 has key length 128: any key-recovery attack in time $< 2^{128}$ would be considered a **break**.
- Even an attack infeasible in practice, like 2^{120} time and / or 2^{100} memory.

Why such drastic requirements?

- Because we can afford it;
- Because we need a **good understanding** of security levels;
- Because "attacks only get better".

Security requirements (ctd.)

More generally, we want symmetric crypto primitives to behave "as ideal".

But we have no way to prove it unconditionally, so at the primitive level, security is only guaranteed by **cryptanalysis**.

Pseudo-random function

A PRF is a **family of functions** $\{0,1\}^n \to \{0,1\}^n$ indexed by $K \in \mathcal{K}$ such that the advantage of any PPT adversary in the **PRF distinguishing game** is negligible.

PRF security game

The **PRF security game** is played between a **distinguisher** $\mathcal{D}$ and a **challenger** $\mathcal{C}$:

- **Initialization:** $\mathcal{C}$ samples $K \hookleftarrow U(\mathcal{K})$ and $b \hookleftarrow U(0,1)$, and samples a random function $F : \{0,1\}^n \to \{0,1\}^n$
- **Queries:** $\mathcal{D}$ can ask **encryption** queries.

  If $b = 0$ (RAND) : $\mathcal{C}$ replies using F
  If $b = 1$ (PRP) : $\mathcal{C}$ replies using F_K

- $\mathcal{D}$ computes a bit b', returns, and wins if $b' = b$.

The advantage of $\mathcal{D}$ is:

$$\mathrm{Adv}^{PRF}(\mathcal{D}) = |\Pr[\mathcal{D} \text{ Win}] - 1/2| \ .$$

PRF-PRP switching

Let $\mathcal{A}$ be an adversary that interacts with an oracle. How can $\mathcal{A}$ make the difference between a PRF and a PRP?
$\Longrightarrow$ by finding collisions!

PRF-PRP switching

Let $\mathcal{A}$ be an adversary making q queries, and F_K a family of pseudo-random functions :

$$\left|\mathrm{Adv}^{PRF}(\mathcal{A}) - \mathrm{Adv}^{PRP}(\mathcal{A})\right| \leq \frac{q^2}{2^{n+1}} \ .$$

This collision-like bound appears in many security proofs where we essentially use a PRP "as a PRF".

Proof (sketch)

We run $\mathcal{A}$:

- in the PRP security game (cases: F_K or random permutation Π)
- in the PRF security game (cases: F_K or random function F)

Let E be the event "$\mathcal{C}$ returns the same value for two distinct queries". Then:

$$\begin{aligned}
\Pr[\mathcal{A} \text{ wins PRF}] &= \Pr[\mathcal{A} \text{ wins PRF}|E]\Pr[E] \\
&\quad + \Pr[\mathcal{A} \text{ wins PRF}|\neg E]\Pr[\neg E] \\
&\leq \Pr[E] + \Pr[\mathcal{A} \text{ wins PRF}|\neg E] \\
&= \Pr[E] + \Pr[\mathcal{A} \text{ wins PRP}|\neg E] \\
&\leq \Pr[E] + \Pr[\mathcal{A} \text{ wins PRP}] \quad .
\end{aligned}$$

If E does not happen, both games are equivalent from $\mathcal{A}$'s point of view!

The probability that E occurs is $q^2/(2^{n+1})$ due to the birthday bound, giving the result.

(Proof not required for the exam).

Modes of Encryption

General remarks

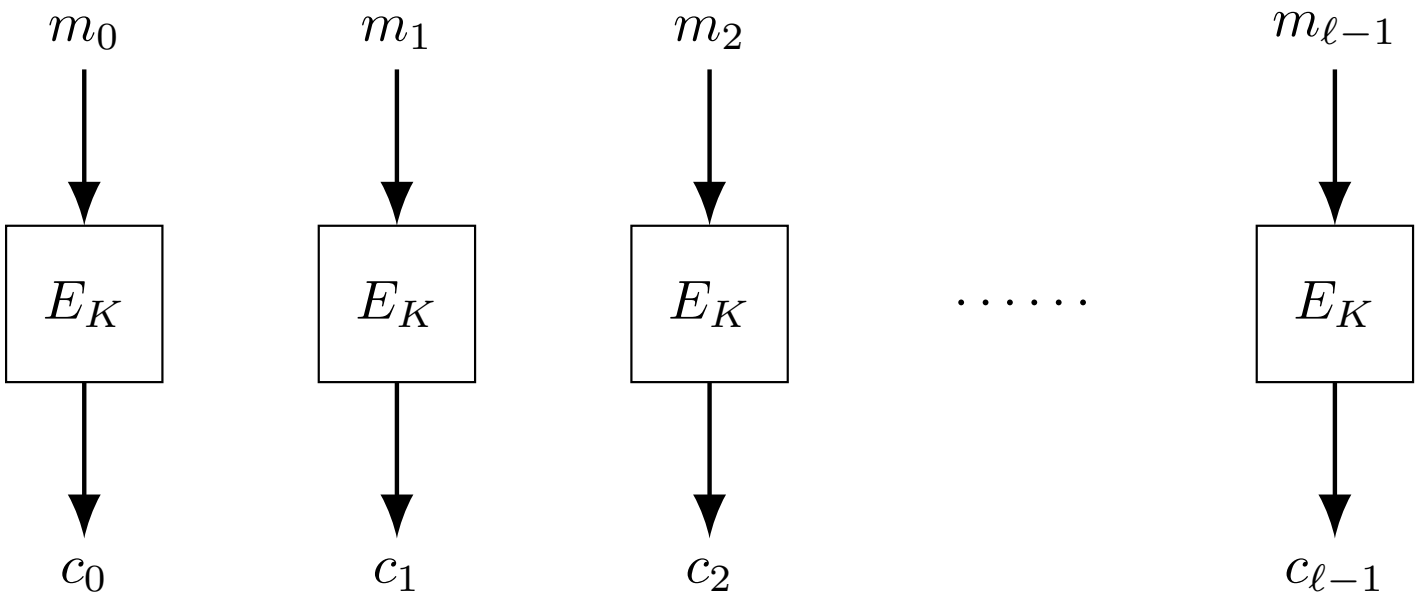
Modes of operation are generic constructions which use **primitives** to obtain some functionality. For example, symmetric encryption:

$$\text{Enc} : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^*$$

A **proof of security** will reduce the security of the mode to the primitive(s): "if the block cipher is a PRP, then Enc is IND-CPA secure up to... queries ".

Modes will accept messages of **any length**, but usually need some **secure padding scheme** to transform the message into fixed-length blocks.

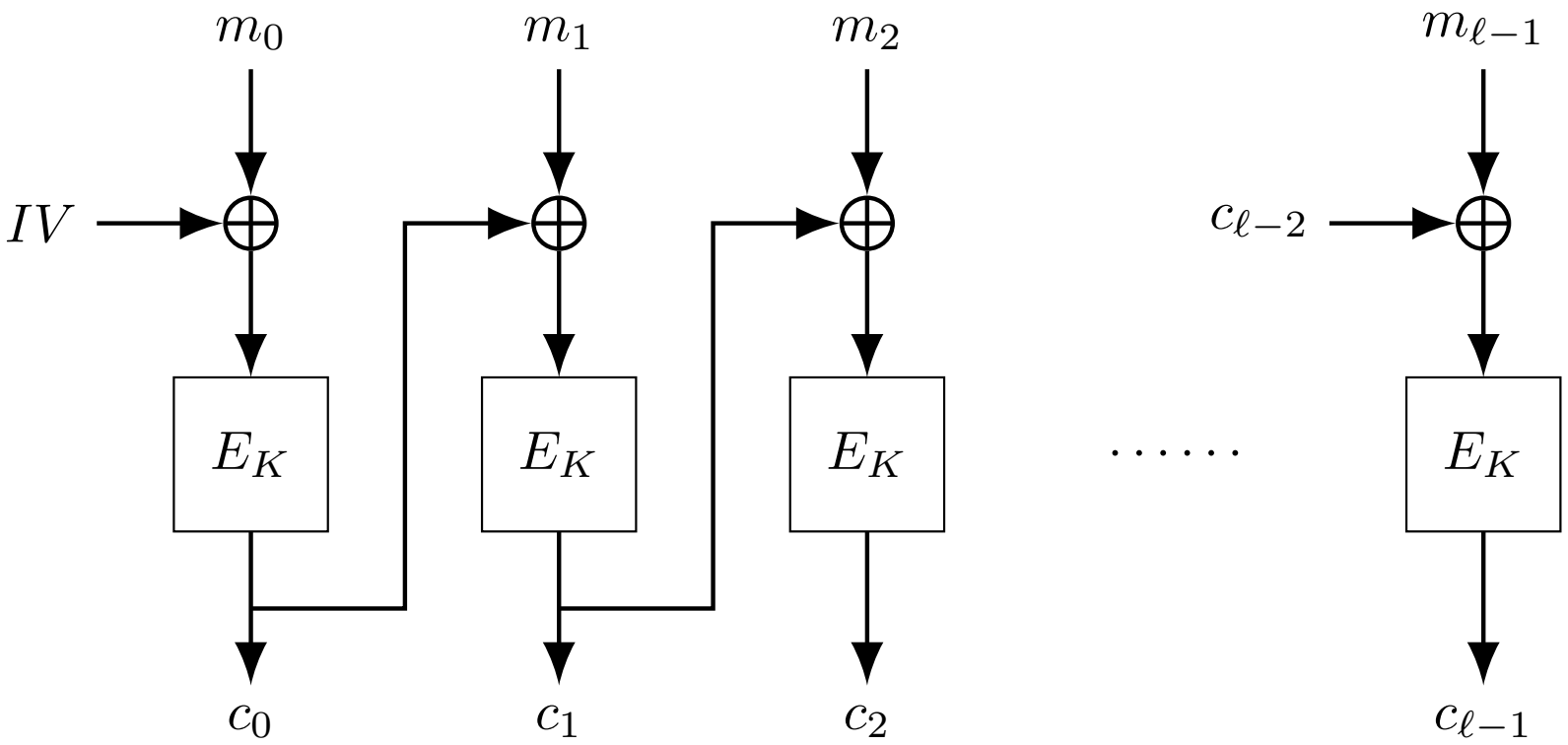
Not a mode: ECB



It's bad.

Fun fact: Zoom "end-to-end encryption" used AES in ECB mode at some point.

Cipher Block Chaining



CBC with random IV

Enc(K, m):

- Cut m in blocks $m_0, \ldots, m_{\ell-1}$ of size n (using padding)
- $IV \hookleftarrow U(\{0,1\}^n)$
- $c_0 \leftarrow E_K(m_0 \oplus IV)$
- For $i = 1$ to $t-1$: $c_i \leftarrow E_K(c_{i-1} \oplus m_i)$
- Return $IV, c_0, \ldots, c_{\ell-1}$

Dec(K, IV, c):

- $m_0 \leftarrow E_K^{-1}(c_0) \oplus IV$
- For $i = 1$ to $\ell - 1$: $m_i \leftarrow E_K^{-1}(c_i) \oplus c_{i-1}$

CBC Security

CBC using a random IV is IND-CPA if the block cipher is a PRP: for any IND-CPA adversary $\mathcal{A}$ there is a PRP adversary $\mathcal{B}$ such that:

$$\mathrm{Adv}^{CPA}(\mathcal{A}) \leq \mathrm{Adv}^{PRP}(\mathcal{B}) + \sigma^2/2^{n-1}$$

where σ is the total number of blocks encrypted.

CBC attack: birthday bound

Encrypt $2^{n/2}$ times the same block 0, then $2^{n/2}$ times some secret block m. What happens?

$\Longrightarrow$ we can expect a collision between two ciphertext blocks: one from the first part (c_i), one from the second part (c'_j).

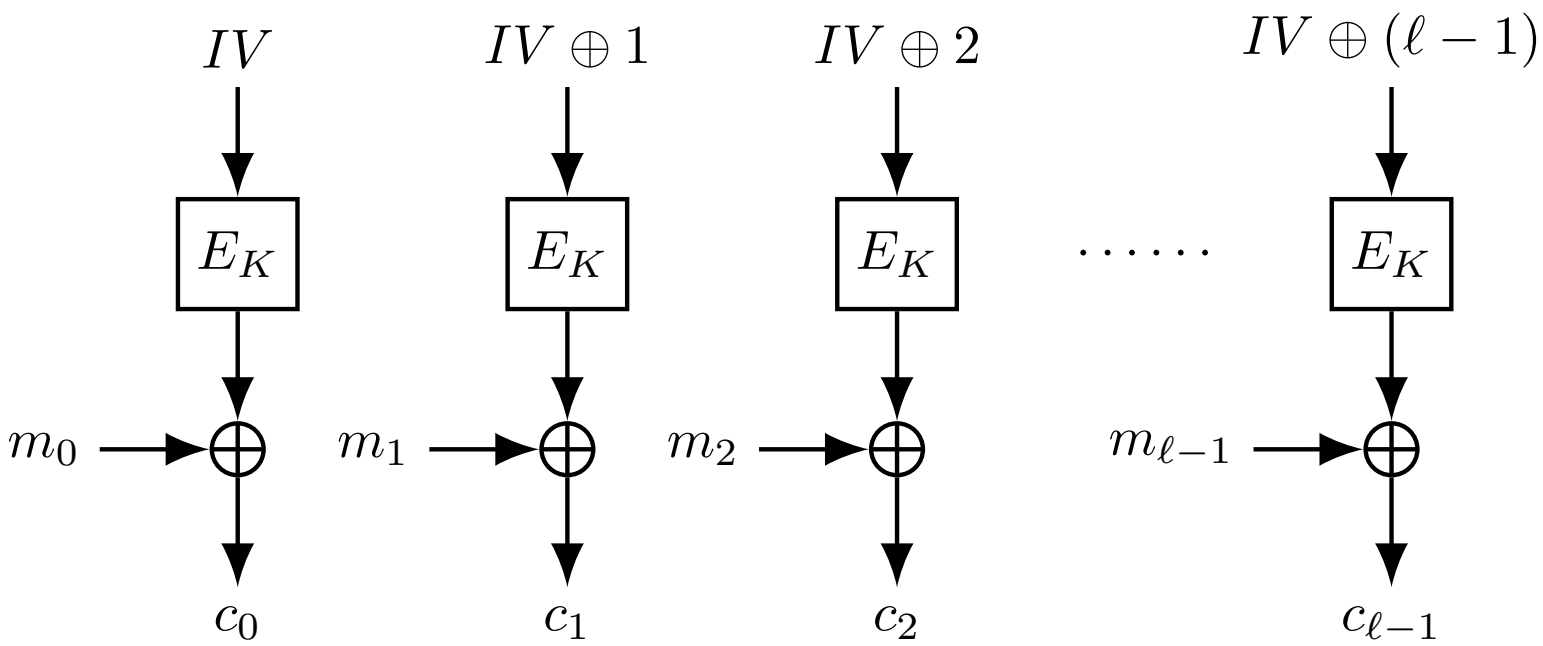
We have:

$$\begin{cases} c_i = 0 \oplus E_K(c_{i-1}) \text{ (encryption of 0 block)} \\ c'_j = m \oplus E_K(c'_{j-1}) \text{ (encryption of } m \text{ block)} \\ c_i = c'_j \end{cases}$$

$\Longrightarrow E_K(c_{i-1}) \oplus E_K(c'_{j-1}) = m \Longrightarrow$ get m.

Fun fact: this can become practical if you use a 64-bit block cipher and rekeying is not implemented (Sweet32 attack).

Counter Mode (CTR)



Can be used with a random IV, or with an internal counter

CTR with counter

Internal counter: $ctr \leftarrow 0$
Enc(K, m):

- Cut m in blocks $m_0, \ldots, m_{t-1}$ of size n
- For $i = 0$ to $\ell - 1$: $c_i = m_i \oplus E_K(ctr + i)$
- $ctr \leftarrow ctr + \ell$
- Return $ctr - \ell, c_0, \ldots, c_{t-1}$

Dec(K, ctr, c):

- Return $m_i = c_i \oplus E_K(ctr + i)$ for $i = 0$ to $\ell - 1$

CTR with a counter is IND-CPA if the block cipher is a PRP: for any IND-CPA adversary $\mathcal{A}$ there is a PRP adversary $\mathcal{B}$ such that:

$$\mathrm{Adv}^{IND-CPA}(\mathcal{A}) \leq 2\mathrm{Adv}^{PRP}(\mathcal{B}) + \sigma^2/2^{n+1}$$

where σ is the total number of encrypted blocks.

Proof (sketch)

1. If E_K is replaced by a random function, CTR is IND-CPA (the advantage of the adversary is 0)
2. Using PRP-PRF switching, we replace E_K by a PRF (term $\sigma^2/2^{n+1}$)
3. We transform an adversary $\mathcal{A}$ for the IND-CPA game into an adversary $\mathcal{B}$ for the PRF game.

Proof not required for exam.

Authenticated Encryption

Authenticated Encryption

- Modes like CBC or CTR offer no protection against tampering
- They offer no IND-CCA security (with decryption queries) & no authenticity

- Use an **authenticated encryption mode** with built-in authenticity (e.g., GCM)
- Or use a **message authentication code** in addition to encryption

Message authentication code

$$\begin{cases} \text{KeyGen}: & 1^n & \mapsto & k \\ MAC: & k, m & \mapsto & t \\ \text{Verify}: & k, m, t & \mapsto & \{0,1\} \end{cases}$$

Correctness: $\forall m$, $\text{Verify}(k, m, MAC(k, m)) = 1$.

It's like a "symmetric signature":

- Guarantees authenticity
- Guarantees integrity
- Security based on an unforgeability game

MAC security: EUF and SUF-CMA

EUF-CMA (**SUF-CMA**) is defined by a security game played by $\mathcal{C}$ and $\mathcal{A}$.

- **Initialization**: $\mathcal{C}$ samples a key k
- **Queries:** at any point, $\mathcal{A}$ can choose m_i and obtain the MAC $t_i = MAC(k, m_i)$
- **Forgery:** $\mathcal{A}$ sends a pair (m, t) to $\mathcal{C}$ and wins if:

  EUF-CMA: (m, t) is valid and m is a new message

  SUF-CMA: (m, t) is valid and (m, t) is a new pair (but m could be one of the previous messages)

The EUF/SUF-CMA advantage of $\mathcal{A}$ is defined as:

$$\mathrm{Adv}(\mathcal{A}) = |\Pr[\mathcal{A} \text{ wins}]| \ .$$

The MAC is EUF/SUF-CMA secure iff any PPT adversary has a negligible advantage.

Encrypt-then-MAC

Use a **pair** of keys (k, k').

On input m:

- Compute $c = \mathrm{Enc}(k, m)$
- Compute $MAC(k', c)$
- Return $c, MAC(k', c)$

Theorem If Enc is IND-CPA and MAC is SUF-CMA, then encrypt-then-MAC is secure (in particular, IND-CCA).

Example: ECBC-MAC