Introduction to Cryptology

5.1 - Modes of Operation

Federico Pintore
Mathematical Institute, University of Oxford (UK)

UNIVERSITY OF OXFORD

Michaelmas term 2020

Modes of operation

Stream and block ciphers are used to obtain computationally-indistinguishable and CPA-secure encryption, respectively.

Both the constructions have some drawbacks.

They are addressed by different modes of operation of block and stream ciphers.

Modes of Operation of Stream Ciphers

Computational Indistinguishability using a PRG

A stream cipher (Init,GetBits) can be used to construct PRGs.

Construction of a PRG $G_{\ell(n)}$:

$$
\begin{aligned}
&\mathrm{st}_0 \leftarrow \mathrm{Init}(s, IV) \\
&\text{for } i = 1, \cdots, \ell(n) \\
&\quad (y_i, \mathrm{st}_i) \leftarrow \mathrm{GetBits}(\mathrm{st}_{i-1}) \\
&\text{return } y_1, \cdots, y_{\ell(n)}
\end{aligned}
$$

Computational Indistinguishability using a PRG

A stream cipher (Init,GetBits) can be used to construct PRGs.

Construction of a PRG $G_{\ell(n)}$:

$$
\begin{aligned}
&\mathrm{st}_0 \leftarrow \mathrm{Init}(s, IV) \\
&\text{for } i = 1, \cdots, \ell(n) \\
&\quad (y_i, \mathrm{st}_i) \leftarrow \mathrm{GetBits}(\mathrm{st}_{i-1}) \\
&\text{return } y_1, \cdots, y_{\ell(n)}
\end{aligned}
$$

A stream cipher is secure if:

- it takes no IV,
- for any expansion factor $\ell(n)$, $G_{\ell(n)}$ is a PRG.

Computational Indistinguishability using a PRG

Let G be a pseudorandom generator with expansion factor $\ell(n)$. Define a fixed-length encryption scheme

$$E = (\text{KeyGen}, \text{Enc}, \text{Dec})$$

with $\mathcal{M} = \{0,1\}^{\ell(n)}$, as follows:

- $k \leftarrow \text{KeyGen}(n)$: it uniformly samples $k \in \{0,1\}^n$.
- $c \leftarrow \text{Enc}(k, m)$: on input a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^{\ell(n)}$, it outputs $c = G(k) \oplus m$.
- $m \leftarrow \text{Dec}(k, c)$: on input a key $k \in \{0,1\}^n$ and a ciphertext $c \in \{0,1\}^{\ell(n)}$, it outputs $m = G(k) \oplus c$.

Computational Indistinguishability using a PRG

Let G be a pseudorandom generator with expansion factor $\ell(n)$. Define a fixed-length encryption scheme

$$E = (\text{KeyGen}, \text{Enc}, \text{Dec})$$

with $\mathcal{M} = \{0,1\}^{\ell(n)}$, as follows:

- $k \leftarrow \text{KeyGen}(n)$: it uniformly samples $k \in \{0,1\}^n$.
- $c \leftarrow \text{Enc}(k, m)$: on input a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^{\ell(n)}$, it outputs $c = G(k) \oplus m$.
- $m \leftarrow \text{Dec}(k, c)$: on input a key $k \in \{0,1\}^n$ and a ciphertext $c \in \{0,1\}^{\ell(n)}$, it outputs $m = G(k) \oplus c$.

Theorem

If G is a PRG, then the encryption scheme E derived from G is computationally indistinguishable.

Computational Indistinguishability using a PRG

Drawbacks:

- message length is fixed;
- Enc is deterministic, hence E is not CPA secure.

Computational Indistinguishability using a PRG

Drawbacks:

- message length is fixed;
- Enc is deterministic, hence E is not CPA secure.

Are there alternative uses of stream ciphers
which address these drawbacks?

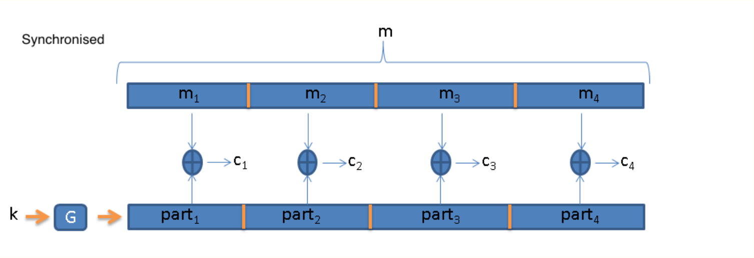
Synchronised mode of operation

Since every stream cipher gives rise to a family of PRGs (one for each $\ell(n)$), an arbitrary-length E can be defined.

The encryption of a message m is $G_{\ell(n)}(k) \oplus m$, where $\ell(n) = |m|$.

It can be proven that the resulting encryption scheme is computationally indistinguishable.

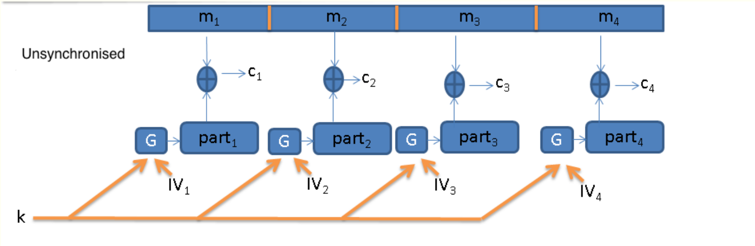
Synchronised mode of operation



Multiple messages can be treated as a single, long message.

- Encrypted blocks can be sent gradually.
- Sender and receiver have to maintain synchronised state.

Unsynchronised mode of operation



- Initialisation vectors are used.
- Stateless CPA-secure encryption is obtained, provided that the stream cipher enjoys extra properties.

Modes of Operation of Block Ciphers

CPA-Security using a PRP

Let F be a PRP. We define the following fixed-lenght encryption scheme $E = (\text{KeyGen}, \text{Enc}, \text{Dec})$:

- $k \leftarrow \text{KeyGen}(n)$: on input n, it outputs a uniformly random key $k \in \{0,1\}^{\ell_{key}(n)}$.
- $c \leftarrow \text{Enc}(k, m)$: given a key k and a message $m \in \{0,1\}^{\ell_{in}(n)}$, it uniformly samples $r \leftarrow \{0,1\}^{\ell_{in}(n)}$, and outputs

$$(c_0, c_1) \leftarrow (r, F_k(r) \oplus m).$$

- $m \leftarrow \text{Dec}(k, (c_0, c_1))$: on input a key k and a ciphertext $c = (c_0, c_1)$, it returns

$$m \leftarrow (F_k(c_0) \oplus c_1).$$

CPA-Security using a PRP

Let F be a PRP. We define the following fixed-lenght encryption scheme $E = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$:

- $k \leftarrow \mathrm{KeyGen}(n)$: on input n, it outputs a uniformly random key $k \in \{0,1\}^{\ell_{key}(n)}$.
- $c \leftarrow \mathrm{Enc}(k, m)$: given a key k and a message $m \in \{0,1\}^{\ell_{in}(n)}$, it uniformly samples $r \leftarrow \{0,1\}^{\ell_{in}(n)}$, and outputs

$$(c_0, c_1) \leftarrow (r, F_k(r) \oplus m).$$

- $m \leftarrow \mathrm{Dec}(k, (c_0, c_1))$: on input a key k and a ciphertext $c = (c_0, c_1)$, it returns

$$m \leftarrow (F_k(c_0) \oplus c_1).$$

Theorem

If F is a PRP with $\ell_{in}(n) \geq n$, then the encryption scheme E is CPA-secure.

CPA-Security using a PRP

Drawbacks:

- message length is fixed;
- the length of the ciphertext is double the length of the message.

CPA-Security using a PRP

Drawbacks:

- message length is fixed;
- the length of the ciphertext is double the length of the message.

Are there alternative uses of block ciphers
to address these drawbacks?

CPA-Security using a PRP

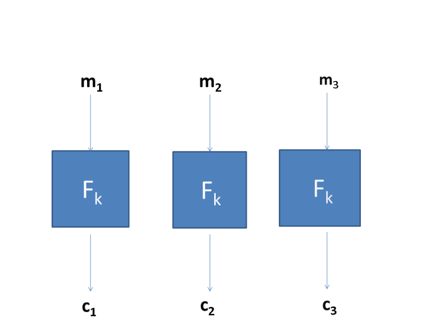
Drawbacks:

- message length is fixed;
- the length of the ciphertext is double the length of the message.

Are there alternative uses of block ciphers
to address these drawbacks?

We assume F is a length-preserving PRP (block cipher), with $\ell_{in}(n) = \ell_{out}(n) = n$, and messages have length multiple of n.

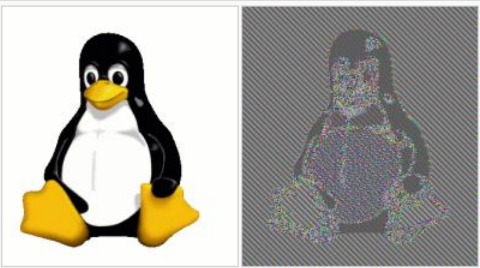
Electronic Code Book (ECB) mode



- It is deterministic, so it cannot be CPA-secure;
- it is not even computationally indistinguishable.

Electronic Code Book (ECB) mode

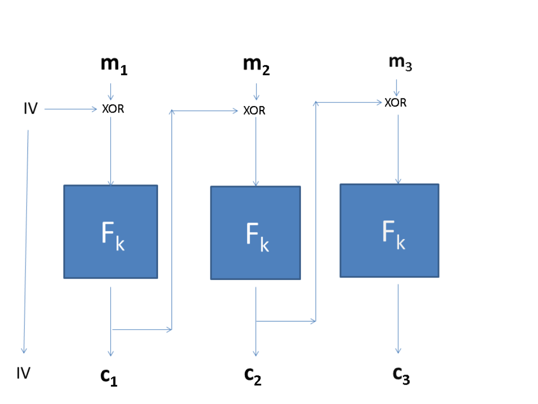
The ECB mode may reveal information about the message:



Original image — Encrypted using ECB mode

Source: Wikipedia

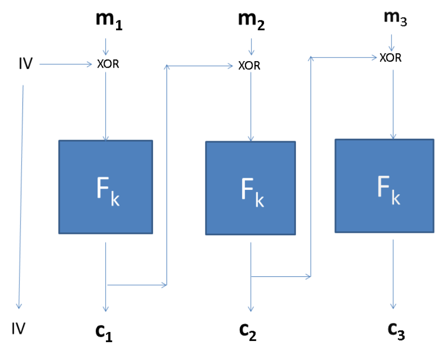
Cipher Block Chaining (CBC) mode



$c \leftarrow \text{Enc}(k, m)$: given a message $m = (m_1, m_2, \ldots, m_t)$ and a key k, it outputs $c = (c_0, c_1, \ldots, c_t)$, where $c_0 = IV$ and

$$c_i = F_k(c_{i-1} \oplus m_i) \quad \text{for} \quad i = 1 \ldots t.$$

Cipher Block Chaining (CBC) mode



$m \leftarrow \mathrm{Dec}(k, c)$: given a ciphertext $c = (c_0, c_1, \ldots, c_t)$ and a key k, it outputs $m = (m_1, \ldots, m_t)$, where

$$m_i \leftarrow F_k^{-1}(c_i) \oplus c_{i-1} \qquad \text{for} \quad i = 1 \cdots t.$$

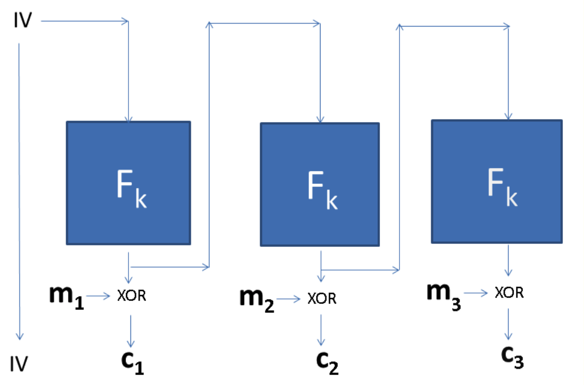
Cipher Block Chaining (CBC) mode

Security:

- If F is a pseudorandom permutation, than the CBC-mode encryption is CPA-secure.
- Chained CBC mode: stateful variant of CBC mode, where the last block of the previous ciphertext repleaces IV in the encryption of the new message. It is not CPA-secure.

Efficiency: no parallel processing (encryption is sequential).

Output Feedback (OFB) mode



- $IV \in \{0,1\}^n$ is chosen uniformly at random.
- $y_0 := IV$ and $y_i := F_k(y_{i-1})$.
- Given IV, a message $m = (m_1, \ldots, m_t)$ and a key k, Enc returns $(c_0, c_1, \ldots, c_t)$ where $c_0 := y_0$, $c_i := y_i \oplus m_i$.
- To decrypt, $m_i := y_i \oplus c_i$ are computed.

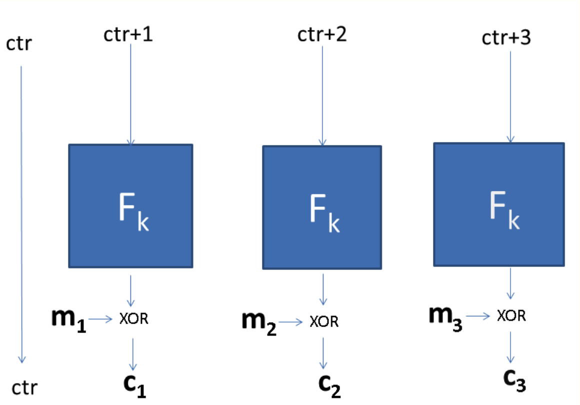
Output Feedback (OFB) mode

Security:

- F_k does not have to be invertible.
- If F is a pseudorandom function, then the OFB mode is CPA-secure.
- Its stateful variant is secure.

Efficiency: most of the computation can be done before encrypting/decrypting.

Counter (CTR) mode



- $\text{ctr} \in \{0,1\}^n$ is chosen uniformly at random.
- $y_i := F_k(\text{ctr} + i \pmod{2^n})$.
- Given ctr, a message $m = (m_1, \ldots, m_t)$ and a key k, Enc returns $(c_1, \ldots, c_t)$ where $c_i := y_i \oplus m_i$.
- To decrypt, $m_i := y_i \oplus c_i$ are computed.

Counter (CTR) mode

Security:

- F_k does not have to be invertible.
- If F is a pseudorandom function, then the CTR mode is CPA-secure.
- Its stateful version is secure.

Efficiency: parallel processing is possible.

Initialisation Vector IV

CBC, OFB and CTR modes use a random IV (or ctr).

[1] (For the birthday paradox - we will cover it.)

Initialisation Vector *IV*

CBC, OFB and CTR modes use a random *IV* (or ctr).

A repeated *IV* could jeopardise security:

- OFB or CTR: the attacker can xor the two resulting ciphertexts to learn about the encrypted plaintexts.
- CBC: after few blocks the inputs to F_k will “diverge”.

[1] (For the birthday paradox - we will cover it.)

Initialisation Vector IV

CBC, OFB and CTR modes use a random IV (or ctr).

A repeated IV could jeopardise security:

- OFB or CTR: the attacker can xor the two resulting ciphertexts to learn about the encrypted plaintexts.
- CBC: after few blocks the inputs to F_k will "diverge".

The block length for DES is $\ell = 64$. After the encryption of data of size 2^{32} bits $\approx$ 34 gigabytes, a repeated IV is expected[1].

[1] (For the birthday paradox - we will cover it.)

Further Reading I

Don Coppersmith.
The data encryption standard (DES) and its strength against attacks.
IBM journal of research and development, 38(3):243–250, 1994.

Itai Dinur, Orr Dunkelman, Masha Gutman, and Adi Shamir.
Improved top-down techniques in differential cryptanalysis.
Cryptology ePrint Archive, Report 2015/268, 2015. http://eprint.iacr.org/.

Further Reading II

Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir.
Efficient dissection of composite problems, with applications to cryptanalysis, knapsacks, and combinatorial search problems.
Cryptology ePrint Archive, Report 2012/217, 2012. http://eprint.iacr.org/.

Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir.
New attacks on feistel structures with improved memory complexities.
In Rosario Gennaro and Matthew Robshaw, editors, Advances in Cryptology – CRYPTO 2015, volume 9215 of Lecture Notes in Computer Science, pages 433–454. Springer Berlin Heidelberg, 2015.

Further Reading III

Lov K Grover.
A fast quantum mechanical algorithm for database search.
In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, pages 212–219. ACM, 1996.

Howard M Heys.
A tutorial on linear and differential cryptanalysis.
Cryptologia, 26(3):189–221, 2002.