Introduction to Modern Cryptography

Michele Ciampi

(Slides courtesy of Prof. Jonathan Katz)

Lecture 9, part 1

Modes of Operation
Block Ciphers and Stream Ciphers*

CPA-secure Encryption (Recall)

Practical CPA-secure Scheme

We have shown a CPA-secure encryption scheme based on any PRF:

$$\mathsf{Enc}_{k}(m) = \langle r, F_k(r) \oplus m \rangle$$

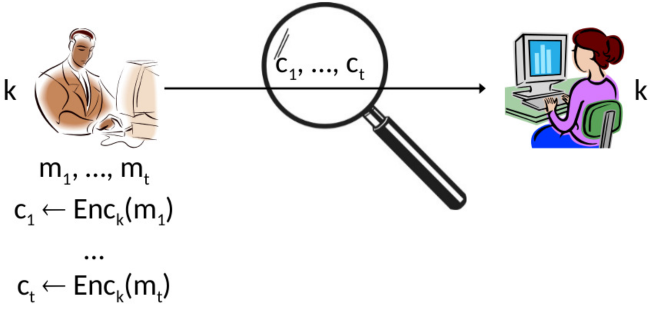
Drawbacks?

- A **1**-block plaintext results in a **2**-block ciphertext
- Only defined for encryption of n-bit messages
- (Both key and message of length n i.e. OTP limitation 1)
- Solution: **Modes of Operation**

Encrypting Long Messages?

- CPA-security $\implies$ security for the encryption of multiple messages
- So, we can encrypt the message $\boldsymbol{m_1 \ldots m_t}$ as $\mathsf{Enc}_{\boldsymbol{k}}(\boldsymbol{m_1}), \mathsf{Enc}_{\boldsymbol{k}}(\boldsymbol{m_2}) \ldots \mathsf{Enc}_{\boldsymbol{k}}(\boldsymbol{m_t})$
- This is also CPA-secure!

Encrypting Long Messages?



- $\forall i: \; c_i = E_k(m_i)$ is CPA-secure
- $\implies \; c = (c_1, c_2, \ldots) = (E_k(m_1), E_k(m_2) \ldots)$ is CPA-secure

Drawback

- The ciphertext is twice the length of the plaintext:

$$\boldsymbol{E_k(m_i) = \langle r_i,\ F_k(r_i) \oplus m_i \rangle}$$

- i.e. ciphertext expansion by a factor of **2**

Question

Can we do better?

Mode of operation (MO)

Modes of operation

Efficient mechanisms for encrypting arbitrary length messages:

1. Block cipher MO
2. Stream cipher MO

Recall

- PRP/PRF $\implies$ block cipher
- PRG $\implies$ stream cipher

Block Ciphers

- Block ciphers are practical constructions of PRPs
- No asymptotics: $\boldsymbol{F : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^m}$
  - $\boldsymbol{n}$ = key length
  - $\boldsymbol{m}$ = block length
- Hard to distinguish $\boldsymbol{F_k}$ from uniform $\boldsymbol{f} \in \boldsymbol{\mathcal{P}_m}$ **even for attackers running in time $\boldsymbol{2^{n-c}}$**

The Advanced Encryption Standard (AES)

- Designed by Belgian cryptographers **Vincent Rijmen** and **Joan Daemen**
- Original proposal **Rijndael**
- **Standardized as AES in 2001** by US NIST
  - after **4** year competition, **15** candidates

The Advanced Encryption Standard (AES)

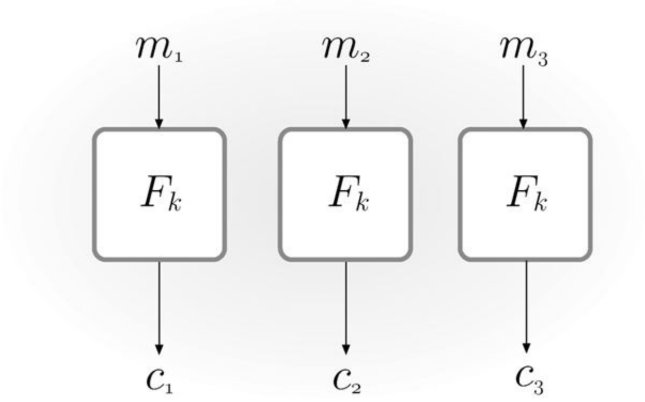
- Technical details
  - Key length = **128**, **192**, or **256** bits
  - Block length = **128** bits
- **Rijndael vs. AES**:
  - Rijndael block size **128/192/256** bits
  - AES block size **128** bits

The Advanced Encryption Standard (AES)

In 2003 US NSA approves the use of AES for **secret** (128 bit key) and **top secret** (256 bit key) documents

- The **most widely used cipher today**:
  - IPSec, SSL/TLS, WiFi IEEE 802.11, SSH, PGP/GPG, ...
- Available in standard crypto libraries
- Best attack only slightly better than brute-force:
  - Bogdanov, Khovratovich, Rechberger, 2011: $\mathbf{2^{126.1}}$

ECB Mode



IMC Textbook 2nd ed. CRC Press 2015

ECB Mode

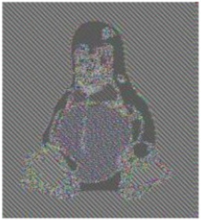
Electronic Codebook Mode

$$\mathsf{Enc}_{\boldsymbol{k}}(\boldsymbol{m_1 \ldots m_t}) = \boldsymbol{F_k(m_1) \ldots F_k(m_t)}$$

- Standartized in 1977 (!)
- Deterministic $\implies$ not CPA-secure
- Can tell from the ciphertext whether $\boldsymbol{m_i = m_j}$
- $\implies$ not even EAV-secure
- **Should not be used in practise!**

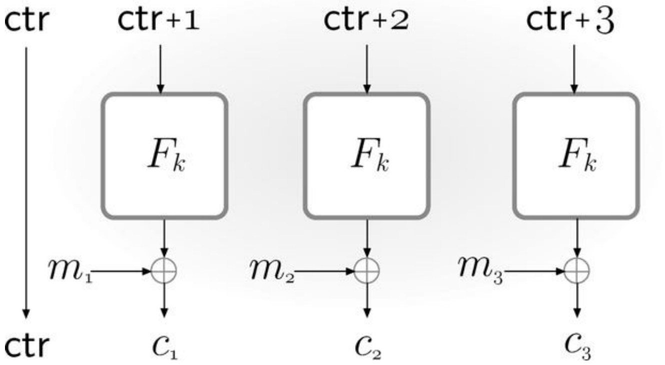
Not just a theoretical problem!

original



encrypted using ECB mode

CTR Mode



IMC Textbook 2nd ed. CRC Press 2015

CTR Mode

Counter Mode

- $\mathsf{Enc}_k(m_1 \ldots m_t)$ (arbitrary t):
  1. Choose $\mathtt{ctr} \leftarrow \{0,1\}^n$, set $c_0 = \mathtt{ctr}$
  2. For $i = 1$ to t:
     - $c_i = m_i \oplus F_k(\mathtt{ctr} + i)$
  3. Output $c_0, c_1, \ldots, c_t$
- Decryption?

Note

Ciphertext expansion is just 1 block (the `ctr` value)

CTR Mode

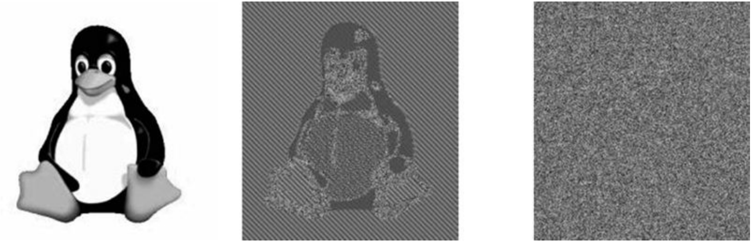
Theorem

If $\boldsymbol{F}$ is a PRF, then CTR mode is CPA-secure

Proof sketch

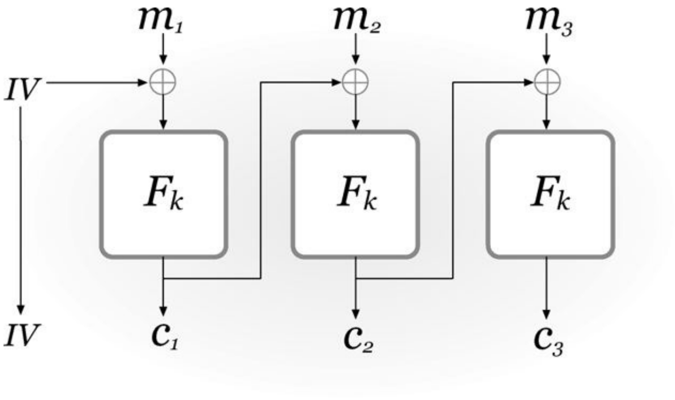
- $\texttt{ctr}_i$ supports up to 2^n values, while message length $t \ll 2^n \implies$ no wraparound
- So the sequence $F_k(\texttt{ctr}_i + 1) \ldots F_k(\texttt{ctr}_i + t)$ used to encrypt the i-th message is pseudorandom
- Moreover, it is independent of every other such sequence unless $\texttt{ctr}_i + j = \texttt{ctr}_{i'} + j'$ for some i, j, i', j'
- It can be shown that the probability of such a collision is $\mathbf{negl}(n)$

ECB vs. CTR



IMC Textbook 2nd ed. CRC Press 2015

CBC Mode



IMC Textbook 2nd ed. CRC Press 2015

CBC Mode

Cipher Block Chaining

- $\mathsf{Enc}_k(m_1 \ldots m_t)$ (arbitrary t):
  1. Choose random $c_0 \leftarrow \{0,1\}^n$ (also called the IV)
  2. For $i = 1$ to t:
     - $c_i = F_k(m_i \oplus c_{i-1})$
  3. Output $c_0, c_1, \ldots, c_t$
- Decryption?
  - Requires F to be invertible i.e. a permutation
  - Hence F – block cipher

Note

Ciphertext expansion is just **1** block (the IV value)

CBC Mode

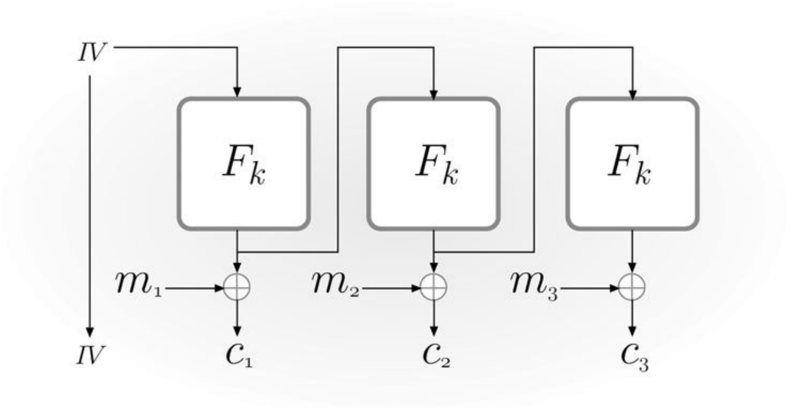
Theorem

If $\boldsymbol{F}$ is a PRP, then CBC mode is CPA-secure

Proof

Proof is more complicated than for CTR mode and is omitted

OFB Mode



IMC Textbook 2nd ed. CRC Press 2015

OFB Mode

Output Feedback Mode

- $\mathsf{Enc}_k(m_1 \ldots m_t)$ (arbitrary t):
  1. Choose random $c_0 \leftarrow \{0,1\}^n$ (the IV); set $y_0 \leftarrow c_0$
  2. For $i = 1$ to t:
     - $y_i = F_k(y_{i-1})$
     - $c_i = m_i \oplus y_i$
  3. Output $c_0, c_1, \ldots, c_t$
- Decryption?
  - F not required to be invertible

Note

OFB mimics OTP/POTP/Stream cipher

OFB Mode

Theorem

If $\boldsymbol{F}$ is a PRF, then OFB mode is CPA-secure

Proof

Omitted

Stream Ciphers*

Stream Ciphers

PRGs

- As we defined them, PRGs are limited
- Have fixed-length output: expand $\boldsymbol{n}$ to $\boldsymbol{p(n)}$
- Produce all output **at once**

Stream Ciphers

- A practical realization of PRGs
- Can be viewed as producing an **infinite stream of pseudorandom bits**, on demand
- More flexible, more efficient

Popular Stream Cipher Standards

- **The A5 family** (A5/1 broken!, A5/2 broken!, A5/3)
  - GSM cellular communications standard
- **RC4** (broken!)
  - TLS/SSL, wireless (WEP/WPA)
- **E0** (broken!)
  - Bluetooth
- **Salsa20**
  - eSTREAM finalist, used in TLS
- **Sosemanuk, HC-128, Rabbit, Trivium, Grain, MICKEY**
  - Other eSTREAM finalists
  - `http://www.ecrypt.eu.org/stream/`

Stream Ciphers

Stream Cipher

Pair of efficient, deterministic algorithms $(\mathsf{Init}, \mathsf{GetBits})$:

- Init: takes a seed $\boldsymbol{s}$ (and optional IV), and outputs initial state $\mathbf{st_0}$
- $\mathsf{GetBits}$: takes the current state $\mathbf{st}$ and outputs a bit $\boldsymbol{y}$ along with updated state $\mathbf{st'}$
  - In practice, $\boldsymbol{y}$ would be a block rather than a bit

End

Reference: Section 3.6.2