Introduction to symmetric cryptography

Christina Boura

École de printemps en codage et cryptographie
May 17, 2016
Overview

- Introduction to symmetric-key cryptography
- Block ciphers
- Boolean functions and cryptographic Sboxes
- Attacks against block ciphers exploiting a low algebraic degree
  - Algebraic attacks
  - Higher-order differential attacks
  - Integral attacks
- Estimating the algebraic degree of iterated constructions
Overview

- Introduction to symmetric-key cryptography
- Block ciphers
- Boolean functions and cryptographic Sboxes
- Attacks against block ciphers exploiting a low algebraic degree
  - Algebraic attacks
  - Higher-order differential attacks
  - Integral attacks
- Estimating the algebraic degree of iterated constructions
Bibliography

- *The Block Cipher Companion*, Lars Knudsen and Matt Robshaw
- *Lecture Notes on Cryptographic Boolean Functions*, Anne Canteaut
- *Analyse de Fonctions de Hachage Cryptographiques*, Thèse, Christina Boura
Outline

1 Introduction to symmetric-key cryptography
Alice and Bob exchange the secret key through a secure channel.
Introduction to symmetric-key cryptography

Symmetric-key encryption

Alice and Bob exchange the secret key through a secure channel.

Key-exchange problem $\Rightarrow$ birth of the public-key cryptography.
Public-key encryption
Advantages and disadvantages of each system

<table>
<thead>
<tr>
<th></th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secret-key</td>
<td>Fast systems</td>
<td>Need secure key-exchange</td>
</tr>
<tr>
<td></td>
<td>Relatively short-keys</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$n$ users: $\frac{n(n - 1)}{2}$ keys</td>
</tr>
<tr>
<td>Public-key</td>
<td>No key-exchange needed</td>
<td>Slow systems</td>
</tr>
<tr>
<td></td>
<td>$n$ users: $2n$ keys</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Relatively long-keys</td>
</tr>
</tbody>
</table>
Hybrid encryption

Idea: Use a combination of asymmetric and symmetric encryption to benefit from the strengths of every system.
Hybrid encryption

- Use a public-key cryptosystem to exchange a key (session key).
- Use the exchanged key to encrypt data by using a symmetric-key cryptosystem.

**Advantages:**

- Slow public-key cryptosystem is used to encrypt a short string only.
- Fast symmetric-key cryptosystem is used to encrypt the longer communication session.
Introduction to symmetric-key cryptography

Symmetric-key authentication

Message authentication code (MAC)
Public-key authentication

Digital signatures

Sign

Alice’s secret key

Alice’s public key

Verify

Y/N
If the message to sign is long, the signing process becomes heavy...

Idea: Use a cryptographic hash function.

\[ H : \{0, 1\}^* \rightarrow \{0, 1\}^n \]

- A good hash function should be preimage, second-preimage and collision resistant.
- In recent hash proposals: \( n = 256, 512 \)

Hash functions are considered as symmetric-key functions because they use similar building blocks with block-ciphers.
Hash and sign
The best of the two worlds

- **Secrecy**: Hybrid encryption
- **Authentication**: Digital signatures with hashing

There is a need for both public and symmetric-key cryptosystems.
Symmetric-key cryptosystems

A **cryptosystem** is a five-tuple \((\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})\)

- \(\mathcal{P}\): set of possible **plaintexts**
- \(\mathcal{C}\): set of possible **ciphertexts**
- \(\mathcal{K}\): set of possible **keys**
- For each \(k \in \mathcal{K}\), there is an encryption rule \(e_k \in \mathcal{E}\) and a decryption rule \(d_k \in \mathcal{D}\).

For each \(k \in \mathcal{K}\): 
\[
d_k(e_k(m)) = m, \text{ for every } m \in \mathcal{P}.
\]

\[\begin{array}{c}
m \rightarrow e_k \\
c \rightarrow d_k \\
m
\end{array}\]
Kerckhoffs’s principle (1883)

In 1883 August Kerckhoffs stated 6 design principles for military ciphers. The 2nd principle states:

A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.

Reformulated by Claude Shannon as

“"The enemy knows the system.""

i.e., “One ought design systems under the assumption that the enemy will immediately gain full familiarity with them.”
Claude Shannon’s theory


Many fundamental ideas of modern cryptography are introduced there:

- Provable security.
- Confusion and diffusion.
- Product ciphers.
Shannon’s idea of perfect secrecy

“No information about the plaintext can be obtained by observing the ciphertext”.

Shannon’s definition:

A cryptosystem has **perfect secrecy** if

\[ Pr(m|c) = Pr(m) \text{ for all } m \in \mathcal{P}, c \in \mathcal{C}. \]

An equivalent formulation:

\[ Pr(c|m) = Pr(c) \text{ for all } m \in \mathcal{P}, c \in \mathcal{C}. \]
Shannon’s theorem

A cryptosystem where $|\mathcal{P}| = |\mathcal{C}| = |\mathcal{K}|$ provides perfect secrecy iff

1. $Pr_K(k) = 1/|\mathcal{K}|$, $\forall k \in \mathcal{K}$
2. $\forall m \in \mathcal{P}, c \in \mathcal{C}$, exists unique $k$ such that $e_k(m) = c$.

Fact:

If $|\mathcal{P}| > |\mathcal{K}|$ then no scheme is perfectly secure.
The Vernam Cipher or One-time Pad

One-time Pad

Let $n \geq 1$ and $\mathcal{P}, \mathcal{C}, \mathcal{K} = \{0, 1\}^n$. If $m = (m_1, \ldots, m_n) \in \mathcal{P}$ and $k = (k_1, \ldots, k_n) \in \mathcal{K}$ then

$$c = e_k(m) = (m_1 \oplus k_1, \ldots, m_n \oplus k_n).$$

Decryption: $d_k(c) = c \oplus k = m \oplus k \oplus k = m$

The One-time Pad provides perfect secrecy if used correctly:

- All keys are equally likely.
- Each key is used only once.

Two-time Pad

$$c \oplus c' = (m \oplus k) \oplus (m' \oplus k) = m \oplus m'.$$
The One-time Pad is perfectly secure but...

- The secret key must be as long as the message.
- A new key has to be generated for each communication.
- These long keys have to be exchanged in a secure way.
- Problem of generating truly random sequences for the key.
Confusion and diffusion

**Diffusion:** Each digit of the plaintext and each digit of the secret key should influence many digits of the ciphertext.

**Confusion:** The ciphertext statistics should depend on the plaintext statistics in a manner too complicated to be exploited by the cryptanalyst.

**Idea:** Use permutations to attain diffusion and substitutions to attain confusion.

→ **Product Ciphers**
Security notions

- Perfectly secret system: the key has to be at least as long as the message.

All cryptosystems used in practice can theoretically be broken.

Symmetric-key approach:

- Try to make the system secure against all known attacks.

- No attack should be faster than exhaustive search on the key.
Exhaustive search

**Expected time** to recover a $\kappa$-bit key: $2^{\kappa-1}$ operations.

<table>
<thead>
<tr>
<th>$\kappa$ (bits)</th>
<th>Time complexity (operations)</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>$2^{40}$</td>
<td>easy to break</td>
</tr>
<tr>
<td>64</td>
<td>$2^{64}$</td>
<td>practical to break</td>
</tr>
<tr>
<td>80</td>
<td>$2^{80}$</td>
<td>not currently feasible</td>
</tr>
<tr>
<td>128</td>
<td>$2^{128}$</td>
<td>very strong</td>
</tr>
<tr>
<td>256</td>
<td>$2^{256}$</td>
<td>exceptionally strong</td>
</tr>
</tbody>
</table>

Table from [Knudsen, Robshaw, “The Block Cipher Companion”, 2011.]

- The universe is less than $2^{80}$ microseconds old!
- The number of the protons in the universe is $\approx 2^{265}$. 

Cryptanalysis of an encryption scheme

Different attack models:
- Ciphertext-only attack.
- Known-plaintext attack.
- Chosen-plaintext/ciphertext attack.
- Adaptively chosen-plaintext/ciphertext attack.

The performance of an attack is measured by its:
- time complexity.
- data complexity.
- memory complexity.
Symmetric encryption schemes

Stream ciphers
- Combine (XOR) plaintext bits with a keystream generated by a pseudo-number generator.
- Keystream should have good statistical properties.
- **Advantages**: Performance and low hardware complexity.

Block ciphers
- Operate on blocks of data.
- Probably the best understood symmetric primitives.
- Can be used to build hash functions, stream ciphers, MACs, authenticated encryption algorithms, PRNGs...
Block ciphers

Encrypt a block of message $m$ into a block of ciphertext $c$ under the action of the key $k$.

$$E : \{0, 1\}^n \times \{0, 1\}^\kappa \rightarrow \{0, 1\}^n$$

$$(m, k) \mapsto E(m, k) = c$$

- Given $k$, it must be easy to compute $c$ from $m$.
- Given $m, c$ it must be hard to compute $k$ such that $E(m, k) = c$. 
Two important parameters:

- **block size**, \( n \)
- **key size**, \( \kappa \)

A block cipher generates a family of permutations indexed by a key \( k \).

**Ideal design:** \( 2^\kappa \) permutations chosen uniformly at random from all \( 2^n! \approx 2^{(n-1)2^n} \) permutations.
Iterated block ciphers

Idea: Iterate a round function $f$ several times. The function $f^r$ is waited to be strong for large $r$.

Advantages:
- Compact implementation.
- Easier analysis.

Use a key schedule to extend the user-supplied (or master) key to a sequence of $r$ subkeys.
How to build the round function?

Two major approaches:

- Feistel network.
- Substitution-Permutation Network (SPN).
Feistel Network

Introduced by Horst Feistel in the early 70’s.

- Split plaintext block: \( m = (L_0, R_0) \)
- For each round \( i = 0, \ldots, r \) do:
  - \( L_{i+1} = R_i \)
  - \( R_{i+1} = L_i \oplus F(R_i \oplus k_{i+1}) \)
- Ciphertext block \( c = (R_{r+1}, L_{r+1}) \)
Feistel Network

Introduced by Horst Feistel in the early 70’s.

- Split ciphertext block: \( c = (R_{r+1}, L_{r+1}) \)
- For each round \( i = r, \ldots, 0 \) do:
  - \( R_i = L_{i+1} \)
  - \( L_i = R_{i+1} \oplus F(L_{i+1} \oplus k_{i+1}) \)
- Plaintext block \( m = (L_0, R_0) \)

Decryption with \( K = (k_1, \ldots, k_r) \) equals encryption with \( K' = (k_r, \ldots, k_1) \).

\( \rightarrow F \) has not to be invertible.
Data Encryption Standard (DES)

The first and probably most famous Feistel cipher.

Designed by IBM and published in 1975.

- Based on an earlier internal design called *Lucifer*.
- 1977: DES is published as a FIPS standard [FIPS 46].
Introduction to symmetric-key cryptography

DES

- **Block size**: 64 bits
- **Key size**: 56 bits
- **16 rounds**

![Diagram of DES algorithm]

**Key Points**:
- Input block size: 64 bits
- Key size: 56 bits
- 16 rounds of the algorithm
- Each round involves two S-boxes and a Feistel function
- The last round is simplified with a single S-box

**Algorithm Steps**:
1. **IP**: Initial Permutation
2. **P**: Bit-reversal permutation
3. **E**: Expansion permutation
4. **F**: Feistel function
   - S-boxes: S1, S2, S3, S4, S5, S6, S7, S8
5. **IP^-1**: Inverse Initial Permutation
Generalized Feistel Networks

Classical Feistel

Unbalanced Feistel

Alternating Feistel

Type-1 Feistel

Type-2 Feistel
Structural properties of DES

The Complementation Property

\[ \overline{\text{DES}}_k(m) = \text{DES}_{\overline{k}}(\overline{m}) \]

where \( \overline{x} := \text{bitwise complement of } x \)

- Limited impact to the security in the classical model.
- Halves the cost of the exhaustive key search.

Encrypt \( m \) and \( \overline{m} \): \( c = \text{DES}_k(m) \) and \( c' = \text{DES}_k(\overline{m}) \)

For each candidate \( t \), compute \( d = \text{DES}_t(m) \).

- Check if \( d = c \) \( \rightarrow \) \( t \) candidate for \( k \).
- Check if \( \overline{d} = c' \) (\( \overline{d} = \text{DES}_{\overline{t}}(\overline{m}) \)) \( \rightarrow \) \( \overline{t} \) candidate for \( k \).
Structural properties of DES

Weak keys

The weak keys are defined as:

\[
\text{Weak keys: } \text{DES}_k(\text{DES}_k(m)) = m.
\]

- 4 weak keys were found for DES.

Each weak key has \(2^{32}\) fixed points \(m : \text{DES}_k(m) = m\).
Breaking DES

- **1992**: Differential cryptanalysis (theoretical attack, $2^{47}$ chosen plaintexts).
- **1994**: Linear cryptanalysis (practical attack, a DES key is recovered).
- **1997**: DESCHALL Project (brute-force project over the net). A message encrypted with DES is broken for the first time.
- **1999**: Deep Crack and distributed.net break a DES key in less than 23 hours.
- **2004**: The standard is withdrawn.

Key-length too short!!!

DES still survives via its Triple-DES form.
Substitution Permutation Network (SPN)
Introduction to symmetric-key cryptography

Substitution Permutation Network (SPN)

- On January 2, 1997 the NIST announced that they wished a successor to DES (to be known as AES).
- Public competition, inputs from the cryptographic community.
- **Requirements**: Block size of 128 bits, key size of 128, 192, 256 bits, security of 2-key triple-DES as minimum.
- 21 submissions (15 accepted for the 1st round)
- 5 finalists (Rijndael, Serpent, Twofish, RC6, MARS)
- On October 2, 2000, Rijndael becomes the AES.
- 2001: Standardization [FIPS 197]
AES

Designed by Joan Daemen and Vincent Rijmen.

Structure: Byte-oriented Substitution-Permutation Network.

- **State**: 128 bits, seen as a $4 \times 4$ matrix of bytes.
- **3 key-lengths**: 128, 192, 256 bits
- **Number of rounds**: 10, 12, 14 rounds resp.
AES Representation

Each byte is viewed in two different ways:

- string of 8 bits \( (b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0) \) \((8^{th}\)-dim vector over \( \mathbb{F}_2 \))
- An element of the finite field with \( 2^8 \) elements \( \mathbb{F}_{2^8} \)

\[
b_7X^7 + b_6X^6 + b_5X^5 + b_4X^4 + b_3X^3 + b_2X^2 + b_1X^1 + b_0
\]

Irreducible polynomial \( R_P \)

\[
R_P = X^8 + X^4 + X^3 + X + 1
\]
An AES round

Four byte-oriented transformations.

- SubBytes
- ShiftRows
- MixColumns
- AddRoundKey
SubBytes
The AES Sbox

\[ S : \mathbb{F}_{2^8} \rightarrow \mathbb{F}_{2^8} \]
\[ x \mapsto x^{-1} \]

followed by an affine transformation on \( \mathbb{F}_{2^8}^8 \):

\[
\begin{pmatrix}
  y_0 \\
  y_1 \\
  y_2 \\
  y_3 \\
  y_4 \\
  y_5 \\
  y_6 \\
  y_7
\end{pmatrix} = \begin{pmatrix}
  1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
  1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
  1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
  1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
  0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
  0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
  0 & 0 & 0 & 1 & 1 & 1 & 1 & 1
\end{pmatrix}
\begin{pmatrix}
  x_0 \\
  x_1 \\
  x_2 \\
  x_3 \\
  x_4 \\
  x_5 \\
  x_6 \\
  x_7
\end{pmatrix} + \begin{pmatrix}
  1 \\
  1 \\
  0 \\
  0 \\
  0 \\
  0 \\
  1 \\
  0
\end{pmatrix}
\]

- Good resistance against differential and linear cryptanalysis.
ShiftRows
MixColumns

\[
\begin{bmatrix}
02 & 03 & 01 & 01 \\
01 & 02 & 03 & 01 \\
01 & 01 & 02 & 03 \\
03 & 01 & 01 & 02
\end{bmatrix}
\begin{bmatrix}
x_0 \\
x_1 \\
x_2 \\
x_3
\end{bmatrix}
=
\begin{bmatrix}
y_0 \\
y_1 \\
y_2 \\
y_3
\end{bmatrix}
\]
MixColumns

- MDS matrix.
- Branch number $= \min_{x \in \mathbb{F}_2^8} \left( HW(x) + HW(M(x)) \right) = 5.$
AddRoundKey

- Lightweight non-linear key-schedule (memory, performance)
Cryptanalysis of AES

- **2000** Integral attacks
- **2002** Algebraic attacks: AES is claimed to be broken. Proved to be not realistic.
- **2009** Related-key attacks: AES-192 and AES-256 are broken under this model. Should we care?
- **2010-2013** Meet-in-the-middle attacks
- **2011** Biclique attacks: First theoretical attacks on full AES. Complexity is quite marginal (see them as accelerated exhaustive search).