Lecture 2. Block ciphers

Passive adversaries, small messages and small shared secret

Bruno Grenet

https://membres-ljk.imag.fr/Bruno.Grenet/IntroCrypto.html

Introduction to cryptology
Université Grenoble Alpes – IM²AG
M1 INFO, MOSIG & AM

Block ciphers: what do we want to achieve?

Goal: Symmetric Encryption

- Encryption: from a message and a key → ciphertexts
- Decryption: from a ciphertext and the key → message
- Security: a ciphertext alone should not give much information

non-determinism

Objects

- Message: any string $\in \{0,1\}^*$.
- Ciphertext: string $\in \{0,1\}^*$, not much larger than the message
- Key: string $\in \{0,1\}^*$ not too large, not too small

efficiency

Block cipher

- Message / ciphertext: fixed-length
- One-to-one mapping for each key → deterministic!

block size

Block ciphers are (mainly) a tool to build higher-level schemes

Contents

Block cipher: definition

Definition

A block cipher is a mapping $E : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ such that for all $k \in \mathcal{K}$, $E(k, \cdot)$ is one-to-one, with

- $\mathcal{K} = \{0,1\}^{\kappa}$: the *key space* $\kappa \in \{\cancel{64}, \cancel{80}, \cancel{96}, \cancel{112}, 128, 192, 256\}$
- $\mathcal{M} = \{0,1\}^{n}$: the *message space* $n \in \{64, 128, 256\}$

- A block cipher is a family of permutations, indexed by the keys
- Examples:
  - One-time pad: $\mathrm{OTP}(k, m) = k \oplus m$
  - Identity: $\mathrm{ID}(k, m) = m$

Notation

- We write $E_k(m)$ for $E(k, m)$
- For a fixed k, $E_k : \mathcal{M} \to \mathcal{M}$

What are *good* block ciphers?

Efficiency

- Fast: e.g. *few cycles per byte* on modern CPUs
- Compact: small code / small circuit size
- Easy to implement → avoid side-channel attacks, etc.
- …

Security

- Given $c = E_k(m)$, *hard* to find m without knowing k
- Given m, *hard* to compute c without knowing k
- Given *oracle access* to E_k, *hard* to find k (*)
- Given *oracle access* to $E_k^{\pm}$, *hard* to find k (**)

$E_k^{\pm}$: both E_k and E_k^{-1}

OTP$(k, m) = k \oplus m$ is not secure for (**)

What are *good* block ciphers?

Efficiency

- Fast: e.g. *few cycles per byte* on modern CPUs
- Compact: small code / small circuit size
- Easy to implement → avoid side-channel attacks, etc.
- …

Security

- Given $c = E_k(m)$, *hard* to find m without knowing k
- Given m, *hard* to compute c without knowing k
- Given *oracle access* to E_k, *hard* to find k
- Given *oracle access* to $E_k^{\pm}$, *hard* to find k

$E_k^{\pm}$: both E_k and E_k^{-1}

→ Not enough! Ex.: given E, define $E'_k(m_L \| m_R) = m_L \| E_k(m_R)$

Need a *more general* security definition, that encompasses all of the above (and other)

In an ideal world

Definition

- Let Perm_n the set of all the $(2^n)!$ permutations of $\mathcal{M} = \{0,1\}^n$
- $E : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ is an ideal block cipher if for all $k \in \mathcal{K}$, $E_k \twoheadleftarrow \mathsf{Perm}_n$.

- All keys provide perfectly random and independent permutations
- Non-realistic world:
  - $(2^n)^{2^{n-1}} < (2^n)! < (2^n)^{2^n}$
  - Key size $\simeq \log(2^n!) \simeq n \cdot 2^n$ bits

$n = 32 \Rightarrow 2^{37}$-bit keys!

Why *ideal*?

Fix a key k and a subset $\mathcal{S} \subset \mathcal{M}$ of messages

Assume an adversary knows:

- $E_{k'}(m)$ for all $k' \in \mathcal{K} \setminus k$ and $m \in \mathcal{M}$
- $E_k(m)$ for all $m \in \mathcal{M} \setminus \mathcal{S}$

Perfect secrecy: The adversary has no information about $E_k(m)$ for m in $\mathcal{S}$

(Strong) PRP security: informal presentation

Informally, a block cipher is secure if its behavior is *close enough* to the ideal world

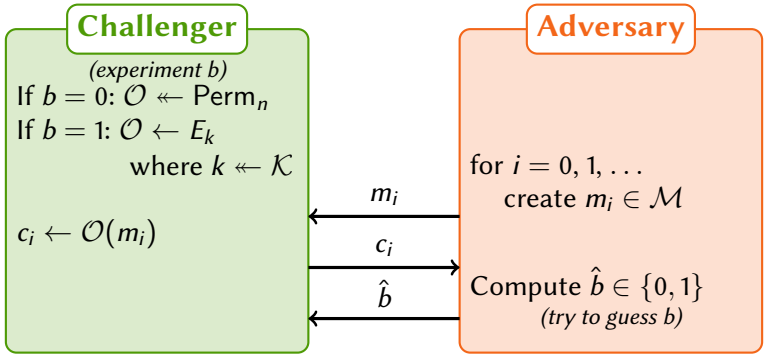
PRP Game

- The adversary plays against a challenger that encrypts using an *oracle* $\mathcal{O}$:
  - either a random permutation
  - or the block cipher E_k with $k \twoheadleftarrow \mathcal{K}$
- The adversary must distinguish between the two *worlds*
- Strong version: access to $\mathcal{O}^{\pm}$

Why does it encompass previous tentative definitions?

- If m can be found from $c = E_k(m)$ without k
  - Take any c and compute the corresponding m
  - Query the oracle on m and compare the result with c
- *Other definitions: exercise!*

(Strong) PRP game



Strong PRP game for E

Adversary also submits queries c_j and gets $m_j = \mathcal{O}^{-1}(c_j)$

Remark

- The adversary *knows* $E \rightarrow$ can compute $E_{k'}(m)$, given k' and m

(Strong) PRP advantage

PRP advantage of $\mathcal{A}$

$$\mathsf{Adv}_E^{\mathsf{PRP}}(\mathcal{A}) = \left|\Pr\left[\mathcal{A}^{\mathcal{O}} \to 1 : \mathcal{O} = E_k, k \twoheadleftarrow \mathcal{K}\right] - \Pr\left[\mathcal{A}^{\mathcal{O}} \to 1 : \mathcal{O} \twoheadleftarrow \mathsf{Perm}_n\right]\right|$$

- Advantage strongly related to the probability of success of $\mathcal{A}$ *exercise*

(Strong) PRP advantage

PRP advantage of $\mathcal{A}$

$$\mathsf{Adv}_E^{\mathsf{PRP}}(\mathcal{A}) = \left|\Pr\left[\mathcal{A}^{\mathcal{O}} \to 1 : \mathcal{O} = E_k, k \twoheadleftarrow \mathcal{K}\right] - \Pr\left[\mathcal{A}^{\mathcal{O}} \to 1 : \mathcal{O} \twoheadleftarrow \mathsf{Perm}_n\right]\right|$$

- Advantage strongly related to the probability of success of $\mathcal{A}$ *exercise*

PRP advantage function of the block cipher E

$$\mathsf{Adv}_E^{\mathsf{PRP}}(q, t) = \max_{\mathcal{A}_{q,t}} \mathsf{Adv}_E^{\mathsf{PRP}}(\mathcal{A}_{q,t})$$

where $\mathcal{A}_{q,t}$ denotes an algorithm that runs in time $\leq t$ and makes $\leq q$ queries to $\mathcal{O}$

- The PRP advantage provides a *measure* on the quality of a PRP, hence a block cipher
- The PRP advantage does *not* define when it is *good*
- Strong PRP advantage: same with the *strong* PRP game queries to $\mathcal{O}^{\pm}$

The generic attack

Generic adversary $\mathcal{A}_{\text{GEN}}$:

Input: Oracle access to either $\mathcal{O} \twoheadleftarrow \text{Perm}_n$ or $\mathcal{O} = E_k$ with $k \twoheadleftarrow \mathcal{K}$

1. $m_1, \ldots, m_q \twoheadleftarrow \mathcal{M}$
2. $k_1, \ldots, k_{t/q} \twoheadleftarrow \mathcal{K}$
3. $C_i \leftarrow [E_{k_i}(m_1), \ldots, E_{k_i}(m_q)]$ for $1 \leq i \leq t/q$ — *computations*
4. $C \leftarrow [\mathcal{O}(m_1), \ldots, \mathcal{O}(m_q)]$ — *oracle queries*
5. Return 1 if there exists i s.t. $C = C_i$, 0 otherwise

Complexity analysis

- Number of queries: q
- Running time: $O(t)$ calls to E.

Probability analysis for the generic attack

Random permutation world

$$\Pr\left[\mathcal{A}_{\mathsf{GEN}}^{\mathcal{O}} \to 1 : \mathcal{O} \twoheadleftarrow \mathsf{Perm}_n\right] = \Pr\left[\exists k_i, \forall m_j, \mathcal{O}(m_j) = E_{k_i}(m_j)\right] \leq t/\left(q \cdot 2^{(n-2)q}\right)$$

Proof.

. Fix a key k_i. Then E_{k_i} is a fixed mapping and each $E_{k_i}(m_j)$ is a fixed value

. Let us fix $c \in \{0,1\}^n$. Then $\forall j \ \Pr\left[\mathcal{O}(m_j) = c\right] = 1/2^n$.

More generally $\Pr\left[\mathcal{O}(m_1) = c_1 \wedge \mathcal{O}(m_2) = c_2 \wedge \dots \wedge \mathcal{O}(m_q) = c_q\right]$

$$= \frac{1}{2^n} \times \frac{1}{2^n - 1} \times \dots \times \frac{1}{2^n - q + 1} = \frac{(2^n - q)!}{2^n!} \leq \frac{1}{2^{(n-2)q}} \quad \text{(loose)}$$

. $\Pr\left[\exists i, \forall j \ \mathcal{O}(m_j) = E_{k_i}(m_j)\right] = \Pr\left[\bigvee_i \forall j \ \mathcal{O}(m_j) = E_{k_i}(m_j)\right] \leq \frac{t}{q} \cdot \frac{1}{2^{(n-2)q}}$

Probability analysis for the generic attack

Random permutation world

$$\Pr\left[\mathcal{A}^{\mathcal{O}}_{\mathsf{GEN}} \to 1 : \mathcal{O} \twoheadleftarrow \mathsf{Perm}_n\right] = \Pr\left[\exists k_i, \forall m_j, \mathcal{O}(m_j) = E_{k_i}(m_j)\right] \leq t/\left(q \cdot 2^{(n-2)q}\right)$$

Block cipher world

$$\Pr\left[\mathcal{A}^{\mathcal{O}}_{\mathsf{GEN}} \to 1 : \mathcal{O} = E_k, k \twoheadleftarrow \mathcal{K}\right] \geq \Pr\left[\exists k_i, k = k_i\right] = t/\left(q \cdot 2^{\kappa}\right)$$

Proof.

We ignore cases where $k \neq k_i$ for all i but it happens that $C = C_i$ for some i.

$$\Pr\left[\exists k_i, k = k_i\right] = \Pr\left[\bigvee_i k = k_i\right] = \frac{\#\{k_i\}}{\#\mathcal{K}} = \frac{t/q}{2^{\kappa}}.$$

Probability analysis for the generic attack

Random permutation world

$$\Pr\left[\mathcal{A}_{\mathsf{GEN}}^{\mathcal{O}} \to 1 : \mathcal{O} \twoheadleftarrow \mathsf{Perm}_n\right] = \Pr\left[\exists k_i, \forall m_j, \mathcal{O}(m_j) = E_{k_i}(m_j)\right] \leq t/q \cdot 2^{(n-2)q}$$

Block cipher world

$$\Pr\left[\mathcal{A}_{\mathsf{GEN}}^{\mathcal{O}} \to 1 : \mathcal{O} = E_k, k \twoheadleftarrow \mathcal{K}\right] \geq \Pr\left[\exists k_i, k = k_i\right] = t/q \cdot 2^{\kappa}$$

Conclusion

$$\mathrm{Adv}_E^{\mathrm{PRP}}(q,t) \geq \mathrm{Adv}_E^{\mathrm{PRP}}(\mathcal{A}_{\mathsf{GEN}}) \geq \frac{t}{q \cdot 2^{\kappa}} - \frac{t}{q \cdot 2^{(n-2)q}} \simeq \frac{t}{q \cdot 2^{\kappa}}$$

for standard parameters

So, what are *good* PRPs or block ciphers?

In this course, no formal definition of a good PRP

Informal (equivalent) definitions

- The advantage is the same as for an ideal block cipher
- The generic attack is almost the best possible
- $\mathrm{Adv}_E^{\mathrm{PRP}}(q,t) \simeq t/q \cdot 2^{\kappa}$

Remarks

- A good PRP is useless if κ is small — *brute force attack*
  - $\kappa \simeq 40$ on a laptop, $\kappa \simeq 60$ on a CPU/GPU cluster, $\kappa \simeq 80$ on an ASIC cluster
- In *asymptotic security*, good $\simeq \mathrm{Adv}_E^{\mathrm{PRP}}(\mathrm{poly}(n), \mathrm{poly}(n)) \ll 1/\mathrm{poly}(n)$

Some final remarks

Block cipher: $E : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ s.t. for all $k \in \mathcal{K}$, E_k is a permutation

- functional definition *what does it do?*

Pseudo-random permutation: $\sigma : \mathcal{M} \to \mathcal{M}$ *indistinguishable* from a random permutation

- security definition *how does it behave?*

Models of security *to use a block cipher E in a more general construction*

Random oracle model: Consider E as a *random permutation*

- Shows resistance against *generic* attacks
- Not sufficient!

(S)PRP model: Consider E as a *good* (S)PRP

- Stronger guarantee
- Still need to be careful

Contents

Generalities

How to build a block cipher?

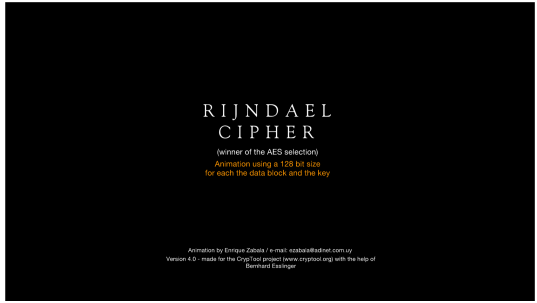
- Several families of construction
  - Substitution-permutation network (SPN) — *e.g.* AES
  - Feistel network — *e.g.* DES
- (Non exhaustive) security goals: prevent the known attacks
  - Brute force
  - Linear cryptanalysis
  - Differential cryptanalysis

Some known block cipher(s families)

- Lucifer / DES: — *Data Encryption Standard*
  - 56-bit key; 64-bit block length — broken using brute force
  - Variants (3-DES & DES-X) with larger key length — quite slow
- Rijndael / AES — *Advanced Encryption Standard*
  - 128, 192 or 256-bit key; 128-bit block length
  - Current standard
- Others: Blowfish, Twofish, Camellia, TEA, …

Example : AES

- NIST Competition (1997-2000)
- Winner: Rijndael, due to V. Rijmen & J. Daemen
- 128-bit block length; Key length 128, 192 or 256 (3 versions)
- Substitution-Permutation Network



Source : https://formaestudio.com/portfolio/aes-animation/

Some algebraic considerations

Bit strings, bytes and finite field

- Input: 128-bit string $\to$ 16-byte string
- One byte $\simeq$ element of $\mathbb{F}_{2^8}$
- $\mathbb{F}_{2^8} \simeq \mathbb{F}_2[x]/\langle x^8 + x^4 + x^3 + x + 1 \rangle$

$10110010 \rightsquigarrow x^7 + x^5 + x^4 + x$

$01000001 \rightsquigarrow x^6 + 1$

$x^7+x^5+x^4+x + x^7+x^{10}+x^{11}+x^{13}$

finite field with 2^8 elements

Degree-7 polynomials

$x^{13} + x^{11} + x^{10} + x^5 + x^4 + x$

$x^5(x^4+x^3+x+1) + x^3(x^4+x^3+x+1)$

$+ x^2(x^4+x^3+x+1) + x^5+x^4+x$

$= \ldots =$ $\deg \leq 7$

SubBytes

- Inverse in $\mathbb{F}_{2^8}$ (with $0 \mapsto 0$)
- Composed with an invertible affine transformation

MixColumns

- Column $\to$ vector in $\mathbb{F}_{2^8}^4$
- Matrix multiplication by an *MDS circulant matrix*

coding theory

$\to$ Algebraic considerations to avoid known attacks

Contents

The context

Increase key length

Given a block cipher with (small) key length κ *e.g.* DES

Build a block cipher with larger key length $\lambda = 2\kappa$ or 3κ, etc.

Rationale: a block cipher can be *very good* except its key length

The simple idea

- Double encryption: $EE_{k_1\|k_2}(m) = E_{k_2}(E_{k_1}(m))$
- Triple encryption:
  - $EEE^3_{k_1\|k_2\|k_3}(m) = E_{k_3}(E_{k_2}(E_{k_1}(m)))$
  - $EEE^2_{k_1\|k_2}(m) = E_{k_1}(E_{k_2}(E_{k_1}(m)))$
  - $EDE^3_{k_1\|k_2\|k_3}(m) = E_{k_3}(E^{-1}_{k_2}(E_{k_1}(m)))$
  - $EDE^2_{k_1\|k_2}(m) = E_{k_1}(E^{-1}_{k_2}(E_{k_1}(m)))$ 3-DES

Are these constructions safe?

- For instance, 3-DES *is* safe
- Exhaustive search: $O(2^{2\kappa})$ or $O(2^{3\kappa})$

Attack on double encryption

$EE_{k_1\|k_2}(m) = E_{k_2}(E_{k_1}(m))$, with $k_1, k_2 \in \{0,1\}^\kappa$.

Meet-in-the-middle

Input: (m, c) where $c = EE_{k_1^*\|k_2^*}(m)$ for *unknown* k_1^*, k_2^*

Output: a (small) set of keys that contains $k_1^*\|k_2^*$

1. Compute each $y_{k_1} = E_{k_1}(m)$ for $k_1 \in \{0,1\}^\kappa$
2. Compute each $z_{k_2} = E_{k_2}^{-1}(c)$ for $k_2 \in \{0,1\}^\kappa$
3. For each *match* $y_{k_1} = z_{k_2}$, add $k_1\|k_2$ to the set of keys
   - $EE_{k_1\|k_2}(m) = E_{k_2}(E_{k_1}(m)) = E_{k_2}(y_{k_1}) = E_{k_2}(z_{k_2}) = c$

Analysis

- Time: twice $O(2^\kappa)$ calls to $E^\pm$ + the matches $\rightarrow$ roughly $O(2^\kappa)$
- Space: two lists of 2^κ ciphertexts and keys $\rightarrow O((n+\kappa)\cdot 2^\kappa)$

$\rightarrow$ Same time as brute force attack with key length 2^κ!

Attack on double encryption

EE

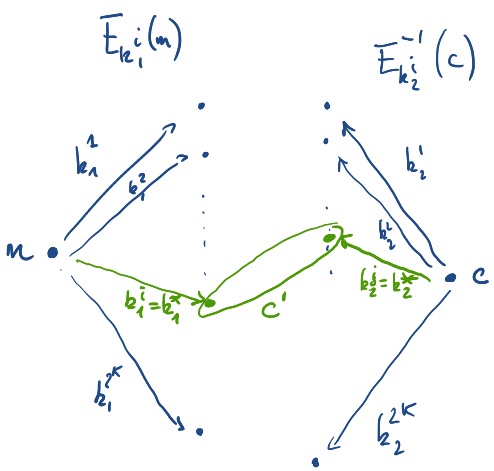
Me

1.
2.
3.

Ana

- ▶
- ▶

→ Same time as brute force attack with key length 2^n!



$c = E_{k_1 \| k_2}(m) = E_{k_2}(E_{k_1}(m))$

Let $c' = E_{k_1}(m)$

Then:

- $c = E_{k_2}(c')$
- $c' = E_{k_2}^{-1}(c)$

Attack on double encryption

$EE_{k_1\|k_2}(m) = E_{k_2}(E_{k_1}(m))$, with $k_1, k_2 \in \{0,1\}^\kappa$.

Meet-in-the-middle

Input: (m, c) where $c = EE_{k_1^*\|k_2^*}(m)$ for *unknown* k_1^*, k_2^*

Output: a (small) set of keys that contains $k_1^*\|k_2^*$

1. Compute each $y_{k_1} = E_{k_1}(m)$ for $k_1 \in \{0,1\}^\kappa$
2. Compute each $z_{k_2} = E_{k_2}^{-1}(c)$ for $k_2 \in \{0,1\}^\kappa$
3. For each *match* $y_{k_1} = z_{k_2}$, add $k_1\|k_2$ to the set of keys
   - $EE_{k_1\|k_2}(m) = E_{k_2}(E_{k_1}(m)) = E_{k_2}(y_{k_1}) = E_{k_2}(z_{k_2}) = c$

Analysis

- Time: twice $O(2^\kappa)$ calls to $E^\pm$ + the matches $\rightarrow$ roughly $O(2^\kappa)$
- Space: two lists of 2^κ ciphertexts and keys $\rightarrow O((n+\kappa)\cdot 2^\kappa)$

$\rightarrow$ Same time as brute force attack with key length 2^κ!

Conclusion

Definitions and security

Block cipher: $E : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ such that each E_k is a permutation

Pseudo-random permutation: $\sigma : \mathcal{M} \to \mathcal{M}$ *indistinguishable* from a random permutation using the (S)PRP game and advantage

Ideal block cipher: each E_k is a random permutation

In practice

- AES / Rijndael:
  - Most used block cipher nowadays, standardized by the NIST, replacement of DES
  - Block size $n = 128$ bits; Key size $\kappa = 128$, 196 or 256 bits
- Some other (less used) possibilities:
  - PRESENT: $n = 64$, $\kappa = 80$ or 128 — *lightweight*
  - SHACAL-2: $n = 256$, $\kappa = 512$ — *large parameters*
  - …

Next lecture

- Symmetric encryption: from fixed-length to variable-length encryption