Lecture 2. Block ciphers

Introduction to cryptology

Bruno Grenet

M1 INFO, MOSIG & AM

Université Grenoble Alpes – IM²AG

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

Block ciphers: what do we want to achieve?

Goal: Symmetric Encryption

- Encryption: from a plaintext and a key $\rightarrow$ ciphertext**s** — *non-determinism*
- Decryption: from a ciphertext and the key $\rightarrow$ plaintext
- Security: a ciphertext alone should not give much information

Objects

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

Block cipher

- Plaintext / ciphertext: fixed-length — *block size*
- One-to-one mapping for each key $\rightarrow$ deterministic!

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

1. Definitions and security

2. Construction of block ciphers

3. Another (generic) attack: *Meet in the middle*

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

$\rightarrow$ a block cipher is a family of permutations, indexed by the keys

Notation

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

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 $\to$ 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 $E(k, \cdot)$, *hard* to find k
- Given *oracle access* to $E^{\pm}(k, \cdot)$, *hard* to find k

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

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 $\to$ 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 $E(k, \cdot)$, *hard* to find k
- Given *oracle access* to $E^{\pm}(k, \cdot)$, *hard* to find k — $E^{\pm}$: both E and E^{-1}

$\to$ 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 $k \in \mathcal{K}$, $E_k \twoheadleftarrow \text{Perm}_n$.

- 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 $\qquad n = 32 \Rightarrow 2^{37}$-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

Experiment

- Challenger gives the Adversary access to an *oracle* $\mathcal{O}$
  - The adversary can *query* $\mathcal{O}(m)$ for any $m \in \mathcal{M}$
  - $\mathcal{O}$ is either a random permutation, or a 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 experiment

PRP experiment for a block cipher E: $\mathrm{Exp}_E^{\mathrm{PRP}}(A)$

Challenger chooses a bit $b \in \{0, 1\}$
Challenger defines an oracle $\mathcal{O}$:
- if $b = 0$: $\mathcal{O} \twoheadleftarrow \mathrm{Perm}_n$
- if $b = 1$: $\mathcal{O} \leftarrow E_k$ where $k \twoheadleftarrow \mathcal{K}$

Adversary submits queries m_i and gets $c_i = \mathcal{O}(m_i)$
Adversary outputs a bit $\hat{b}$

Strong PRP experiment for E: $\mathrm{Exp}_E^{\mathrm{SPRP}}(A)$

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 A

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

- PRP advantage of A is closely related to $\Pr[\text{success of } A]$ *exercise*

(Strong) PRP advantage

PRP advantage of A

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

- PRP advantage of A is closely related to $\Pr\left[\text{success of } A\right]$ *exercise*

PRP advantage of the block cipher E

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

where $A_{q,t}$ 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: replace $\mathsf{Exp}_E^{\mathsf{PRP}}$ by $\mathsf{Exp}_E^{\mathsf{SPRP}}$

The generic attack

Generic adversary $A_{\textsc{gen}}$:

Input: Oracle access to either $\mathcal{O} \twoheadleftarrow \mathrm{Perm}_n$ 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: $O(t)$

Probability analysis for the generic attack

Random permutation world

$$\Pr\left[\mathrm{Exp}_E^{\mathrm{PRP}}(A_{\mathrm{GEN}}) = 1 : \mathcal{O} \twoheadleftarrow \mathrm{Perm}_n\right] = \Pr\left[\exists k_i, \forall m_j, \mathcal{O}(m_j) = E(k_i, m_j)\right] \le t/q \cdot 2^{(n-2)q}$$

Proof. $\Pr[A_{GEN} \text{ returns } 1] = \Pr[\exists i, c_i = c] = \Pr[\exists i\ \forall j\ E(k_i, m_j) = \mathcal{O}(m_j)]$

($\le$) Fix k_i → this fixes every $E_{k_i}(m_j)$

↳ for a fixed c, what is the prob. that $\mathcal{O}(m_j) = c$?

→ for m_1, any n-bit string is equiprobable ⇝ $1/2^n$

m_2, $\mathcal{O}(m_2) \neq \mathcal{O}(m_1)$ ⇝ $1/(2^n - 1)$

⋮

m_j : $\mathcal{O}(m_j) \notin \{\mathcal{O}(m_1), \ldots, \mathcal{O}(m_{j-1})\}$ → $1/(2^n - j + 1)$

$$\Rightarrow \Pr[c_i = c] = \prod_{j=0}^{q-1} \frac{1}{2^n - j} \Rightarrow \Pr[\exists k_i, c_i = c] \le \frac{t}{q} \times \prod_j \frac{1}{2^n - j}$$

Probability analysis for the generic attack

Random permutation world

$$\Pr\left[\mathrm{Exp}_E^{\mathrm{PRP}}(A_{\mathrm{GEN}}) = 1 : \mathcal{O} \twoheadleftarrow \mathrm{Perm}_n\right] = \Pr\left[\exists k_i, \forall m_j, \mathcal{O}(m_j) = E(k_i, m_j)\right] \leq t/q \cdot 2^{(n-2)q}$$

Block cipher world

$$\Pr\left[\mathrm{Exp}_E^{\mathrm{PRP}}(A_{\mathrm{GEN}}) = 1 : \mathcal{O} = E_k, k \twoheadleftarrow \mathcal{K}\right] \geq \Pr\left[\exists k_i, k = k_i\right] = t/(q \cdot 2^{\kappa})$$

Proof.

$$\Pr[A_{\mathrm{GEN}} \text{ returns } 1] = \Pr[\exists i: c_i = c] \geq \Pr[\exists k_i, k = k_i]$$

ignoring the cases where $c = c_i$ though $k \neq k_i$

$$\Pr[\exists k_i ; k = k_i] = \frac{\#\{k_i\}}{\#\{\text{possible } k\}} = \frac{t/q}{2^{\kappa}}.$$

Probability analysis for the generic attack

Random permutation world

$$\Pr\left[\mathrm{Exp}_E^{\mathrm{PRP}}(A_{\mathrm{GEN}}) = 1 : \mathcal{O} \twoheadleftarrow \mathrm{Perm}_n\right] = \Pr\left[\exists k_i, \forall m_j, \mathcal{O}(m_j) = E(k_i, m_j)\right] \leq t/q \cdot 2^{(n-2)q}$$

Block cipher world

$$\Pr\left[\mathrm{Exp}_E^{\mathrm{PRP}}(A_{\mathrm{GEN}}) = 1 : \mathcal{O} = E_k, k \twoheadleftarrow \mathcal{K}\right] \geq \Pr\left[\exists k_i, k = k_i\right] = t/q \cdot 2^{\kappa}$$

Conclusion

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

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 is 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

Generalities

How to build a block cipher?

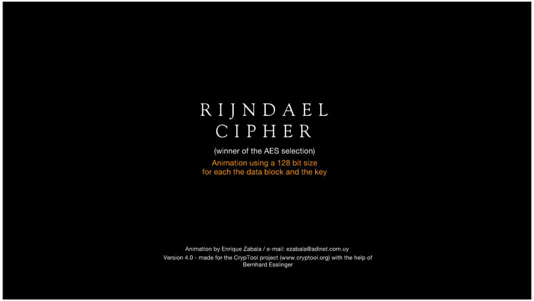
- Several families of construction
  - Substitution-permutation network (SPN) — *e.g.* AES
  - Feistel network — *e.g.* DES
- (Non exhaustive) security goals: prevent the known attacks
  - Brute force
  - Linear cryptanalysis
  - Differential cryptanalysis

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 $\rightarrow$ 16-byte string
- One byte $\simeq$ element of $\mathbb{F}_{2^8}$ — *finite field with 2^8 elements*
- $\mathbb{F}_{2^8} \simeq \mathbb{F}_2[x]/\langle x^8 + x^4 + x^3 + x + 1\rangle$ — *Degree-7 polynomials*

`SubBytes`

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

`MixColumns`

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

> $\rightarrow$ Algebraic considerations to avoid known attacks

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: $EE_2(k_1 \| k_2, m) = E(k_2, E(k_1, m))$
- 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

$EE_2(k_1 \| k_2, m) = E(k_2, E(k_1, m))$, with $k_1, k_2 \in \{0,1\}^\kappa$.

Meet-in-the-middle

Input: (m, c) where $c = EE_2(k_1^* \| k_2^*, m)$ 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 $z_{k_2} = E^{-1}(k_2, c)$ 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
   - $EE_2(k_1 \| k_2, m) = E(k_2, E(k_1, m)) = E(k_2, y_{k_1}) = E(k_2, z_{k_2}) = c$

Analysis

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

$\to$ 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 $E(k, \cdot)$ is a permutation

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

Ideal block cipher: each $E(k, \cdot)$ 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