Secure Channels
Summer Term 2018

Problem Set 2

Prof. Stefan Lucks, Eik List

Bauhaus-Universität Weimar

May 4, 2018

Agenda

In this problem set, you should learn/deepen your understanding in...

- ...security notions for encryption,
- ...their relations, and
- ...reductionist proofs (simulator proofs).

Simulator Proofs

Relations among Notions

How can we show:

$$\text{Notion } X \implies \text{Notion } Y?$$

- **Means:** Every scheme Π that is secure against X-adversaries is also secure against Y-adversaries

Simulator Proofs

Relations among Notions

How can we show:

$$\text{Notion } X \implies \text{Notion } Y?$$

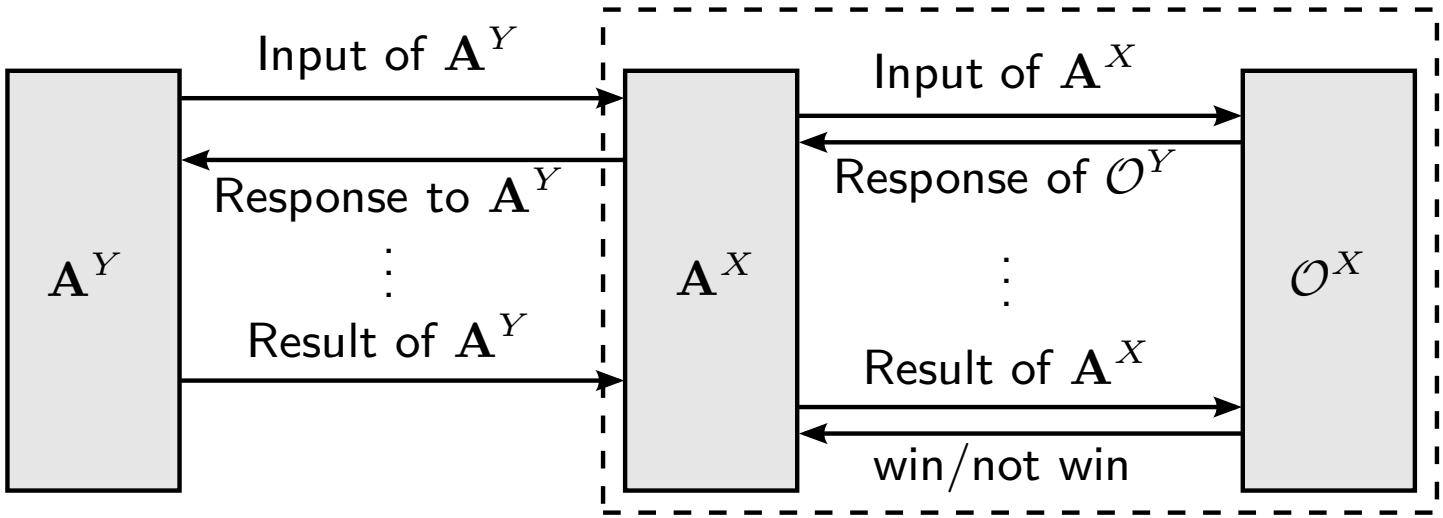
- **Means:** Every scheme Π that is secure against X-adversaries is also secure against Y-adversaries

By **contradiction!**

- If an efficient Y-adversary $\mathbf{A}^Y$ that wins the Y security game **would exist**, then we could use ($=$ **simulate**) it to win the X security game
- $\implies$ There **exists no** efficient Y-adversary with significant advantage on Π

Simulator Proofs

Relations among Notions

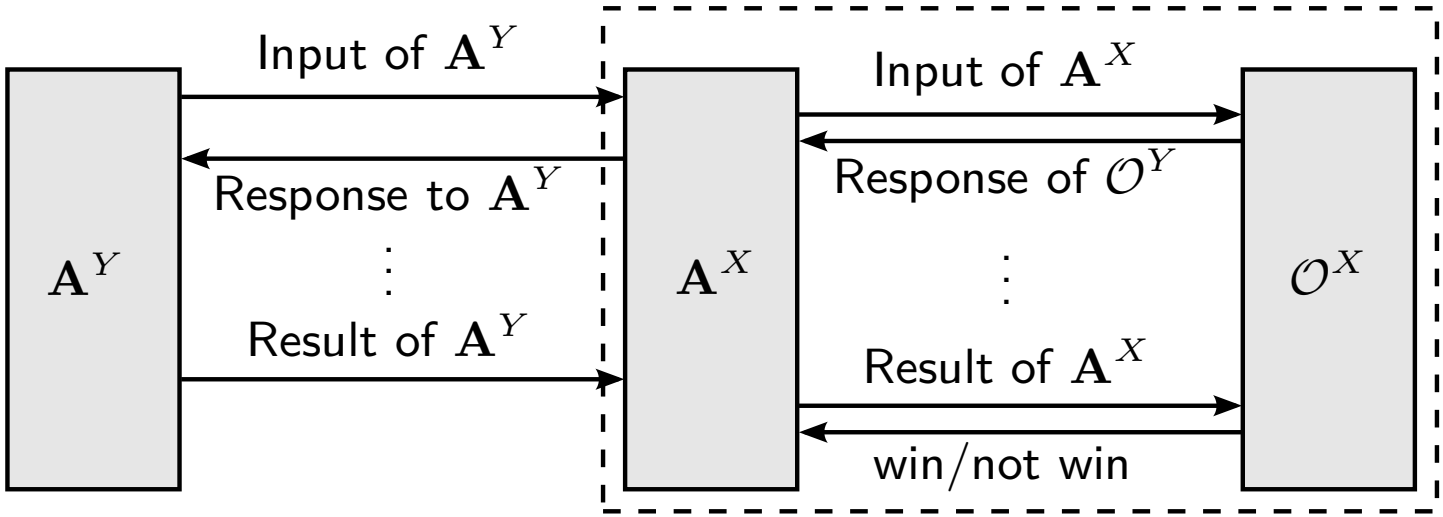


Task 1: Simulator Proofs – Relations among Notions

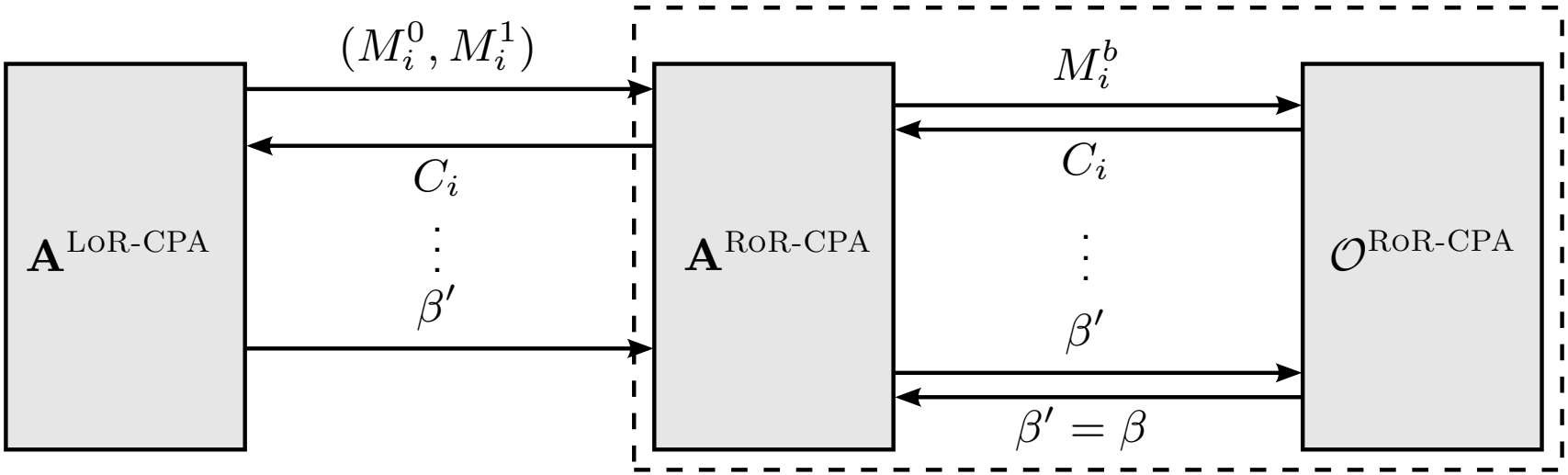
a) RoR-CPA security $\Longrightarrow$ LoR-CPA security

b) Sem-CPA security $\Longrightarrow$ FtG-CPA security

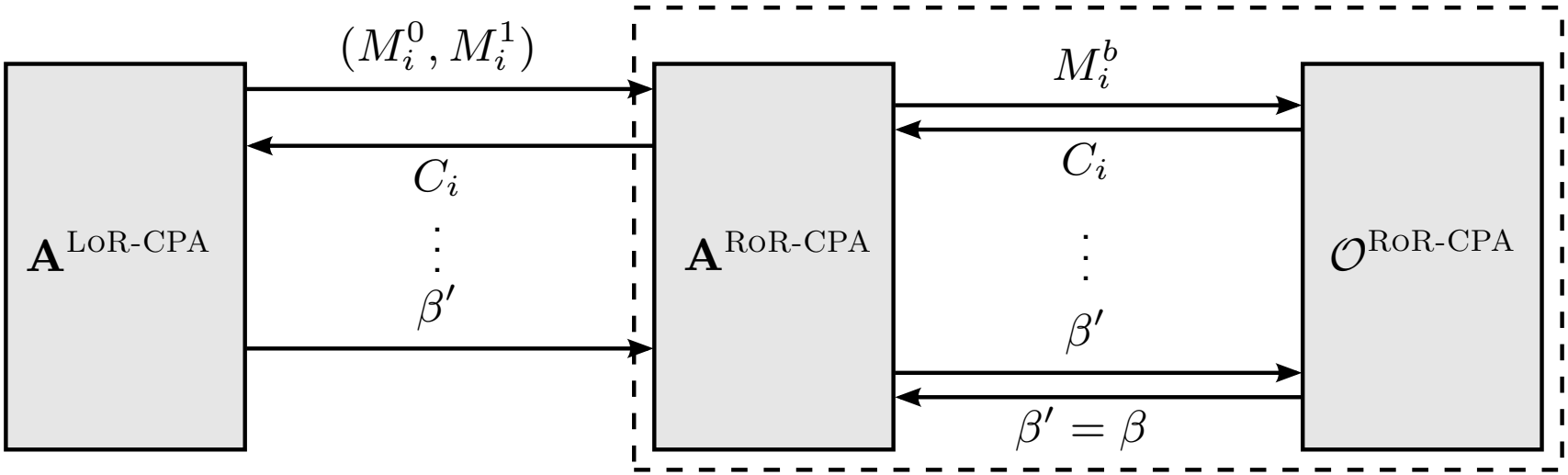
c) LoR-CPA security $\Longrightarrow$ FtG-CPA security



Task 1a) LoR-CPA $\Longrightarrow$ RoR-CPA

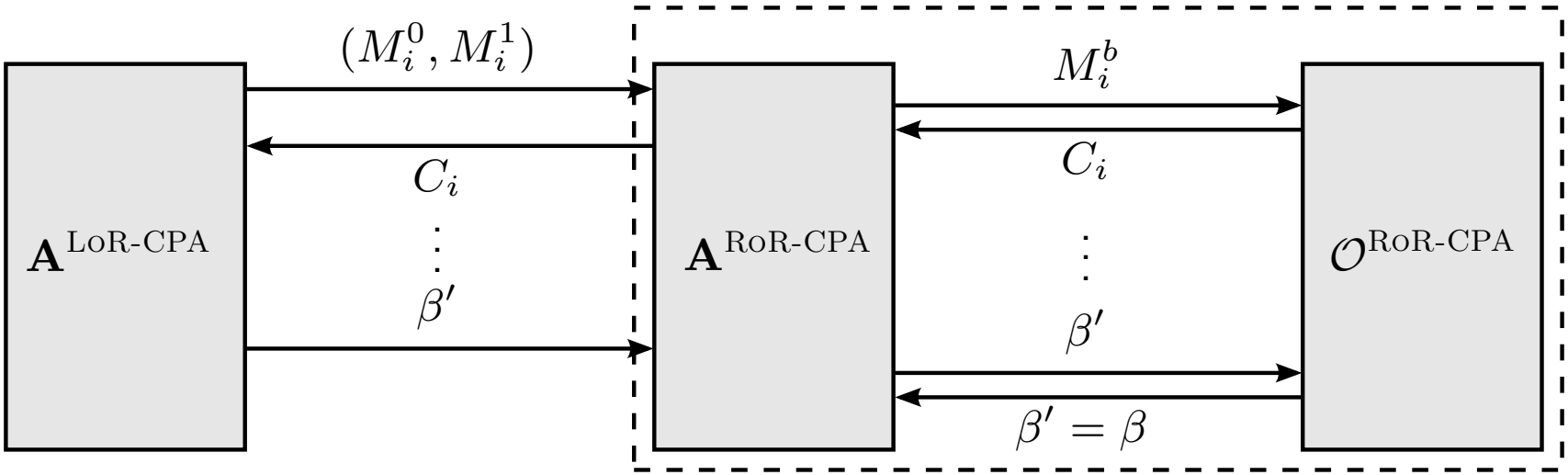


Task 1a) LoR-CPA $\Longrightarrow$ RoR-CPA



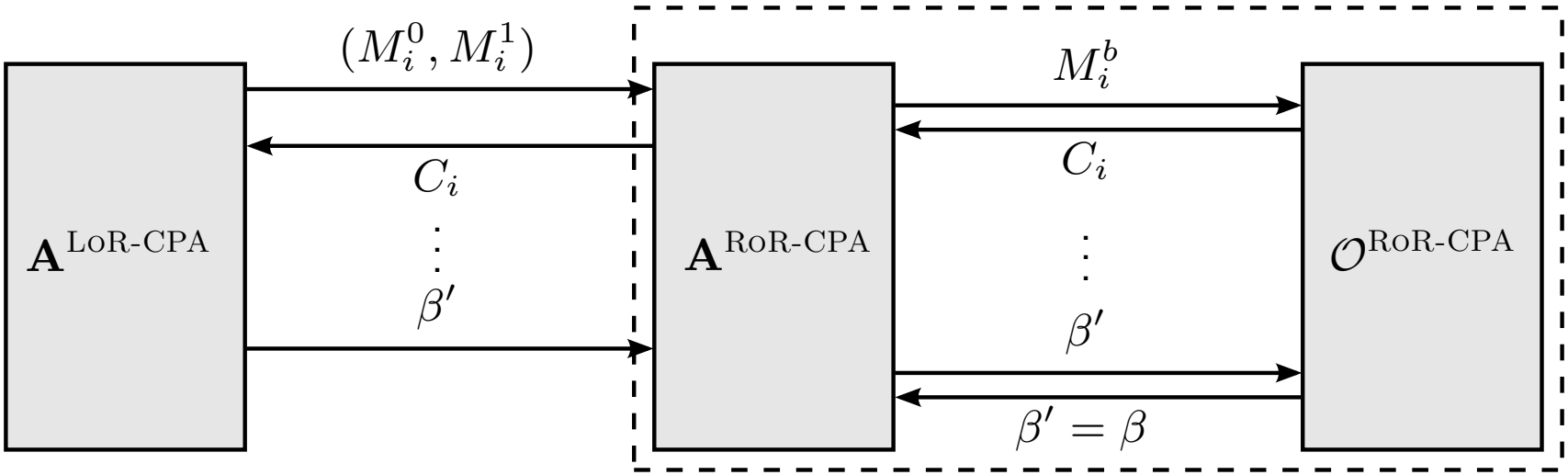
- **Initialization:** $\mathbf{A}^{\text{RoR-CPA}}$ chooses $b \xleftarrow{\$} \{0,1\}$
- **Querying:** $\mathbf{A}^{\text{RoR-CPA}}$ forwards messages M_i^b to its oracle and the responses C_i to $\mathbf{A}^{\text{LoR-CPA}}$, for $1 \leq i \leq q$
- **Guessing:** $\mathbf{A}^{\text{RoR-CPA}}$ forwards the bit β' to the oracle

Task 1a) LoR-CPA $\Longrightarrow$ RoR-CPA – Advantage



- **2 Cases:**
  1. $\mathcal{O}^{\text{RoR-CPA}}$ returns real ciphertexts: Exactly the LoR-CPA setting $\Longrightarrow$ $\mathbf{Adv}(\mathbf{A}^{\text{RoR-CPA}}) = \mathbf{Adv}(\mathbf{A}^{\text{LoR-CPA}})$
  2. $\mathcal{O}^{\text{RoR-CPA}}$ returns random ciphertexts: $\mathbf{A}^{\text{LoR-CPA}}$ has no advantage in general $\Longrightarrow$ $\mathbf{Adv}(\mathbf{A}^{\text{RoR-CPA}}) \geq 0$.

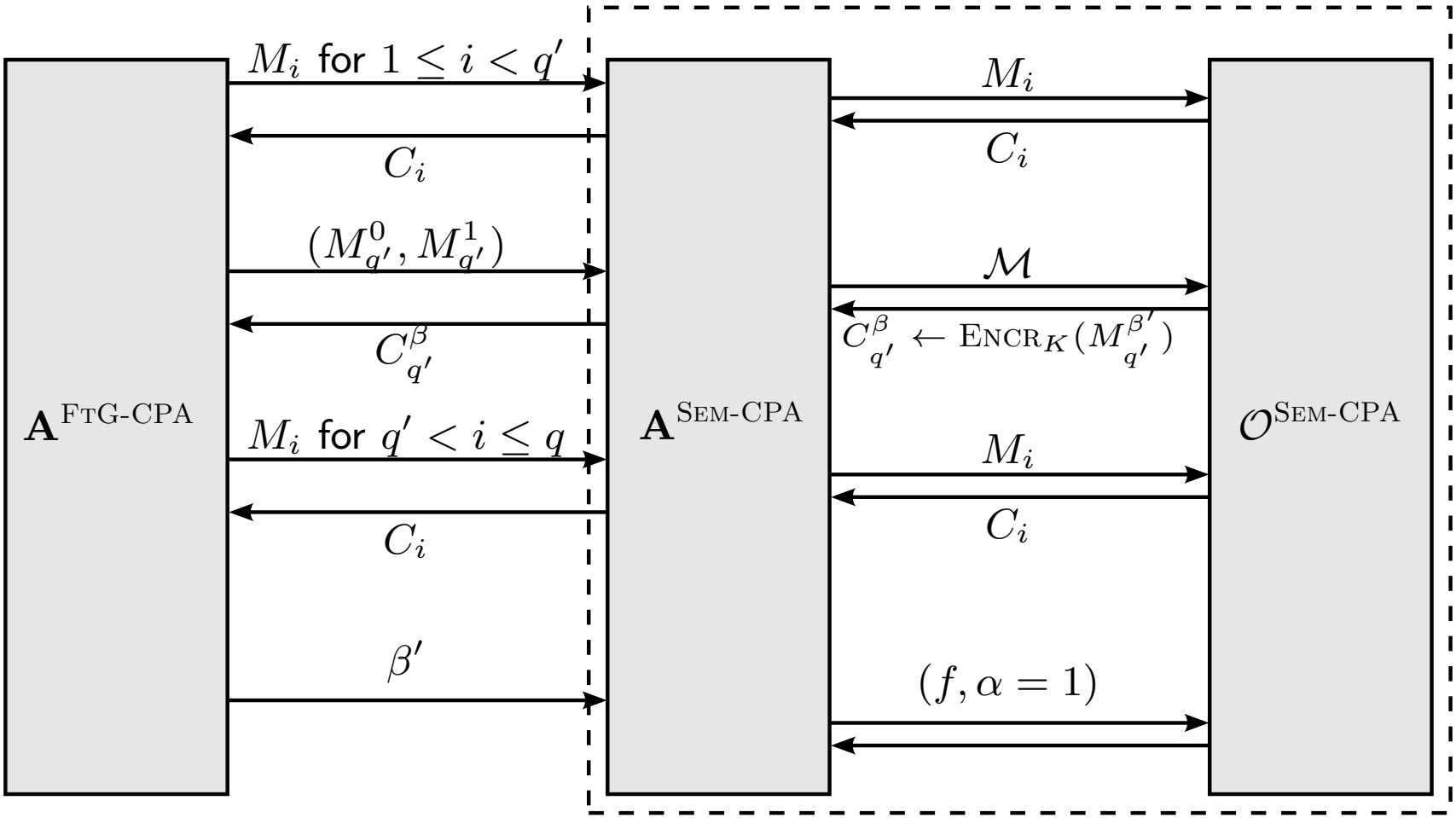
Task 1a) LoR-CPA $\Longrightarrow$ RoR-CPA – Advantage



- **2 Cases:**
  1. $\mathcal{O}^{\text{RoR-CPA}}$ returns real ciphertexts: Exactly the LoR-CPA setting $\Longrightarrow$ $\mathbf{Adv}(\mathbf{A}^{\text{RoR-CPA}}) = \mathbf{Adv}(\mathbf{A}^{\text{LoR-CPA}})$
  2. $\mathcal{O}^{\text{RoR-CPA}}$ returns random ciphertexts: $\mathbf{A}^{\text{LoR-CPA}}$ has no advantage in general $\Longrightarrow$ $\mathbf{Adv}(\mathbf{A}^{\text{RoR-CPA}}) \geq 0$.
- Both cases occur with probability $1/2$:

$$\mathbf{Adv}(\mathbf{A}^{\text{RoR-CPA}}) = 1/2 \cdot \mathbf{Adv}(\mathbf{A}^{\text{LoR-CPA}}) + 0 \cdot 1/2$$

Task 1b) SEM-CPA $\Longrightarrow$ FTG-CPA



Task 1b) SEM-CPA $\Longrightarrow$ FTG-CPA

- **Initialization:** As in usual SEM-CPA game
- **Querying:** $\mathbf{A}^{\text{SEM-CPA}}$ simply forwards queries from and to $\mathbf{A}^{\text{FTG-CPA}}$
- **Challenge:** After $\mathbf{A}^{\text{FTG-CPA}}$ chooses the challenge query, $(M^0_{q'}, M^1_{q'})$, $\mathbf{A}^{\text{SEM-CPA}}$ derives the distribution $\mathcal{M}$:

$$\mathcal{M}(M) := \begin{cases} 1/2 & \text{if } M = M^0_{q'}, \\ 1/2 & \text{if } M = M^1_{q'}, \\ 0 & \text{otherwise.} \end{cases}$$

  $\Longrightarrow$ The oracle chooses $M_{q'}$ as either $M^0_{q'}$ or $M^1_{q'}$ at random with pr. $1/2$ each

- **Guessing:** $\mathbf{A}^{\text{FTG-CPA}}$ outputs β'.
  - $\mathbf{A}^{\text{SEM-CPA}}$ chooses f to model exactly the FTG-CPA response:

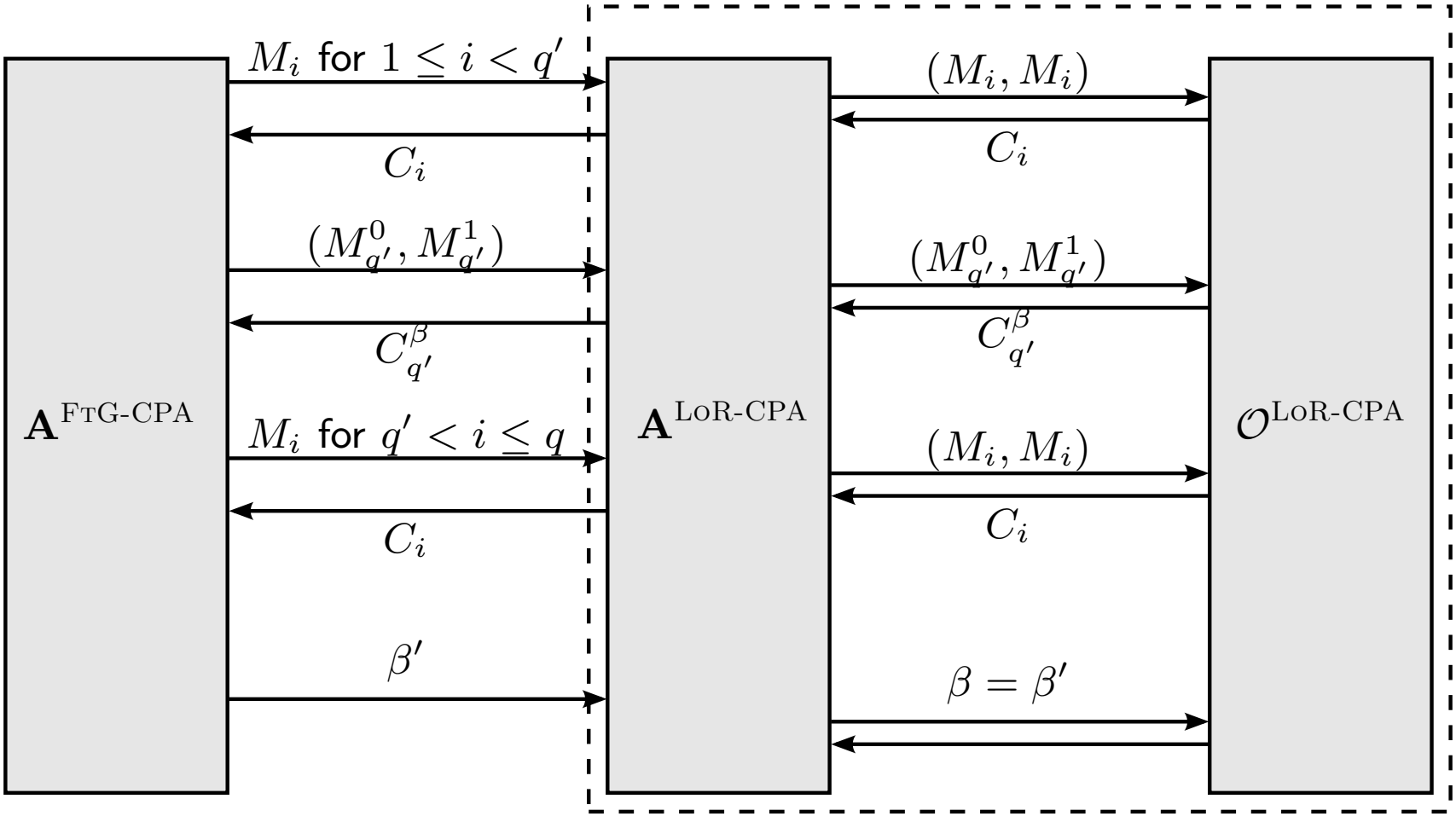
$$f(M) := \begin{cases} 1 & \text{if } M = M^{\beta'}_{q'} \\ 0 & \text{otherwise.} \end{cases}$$

  - $\mathbf{A}^{\text{SEM-CPA}}$ sends $(f, \alpha = 1)$ to the oracle

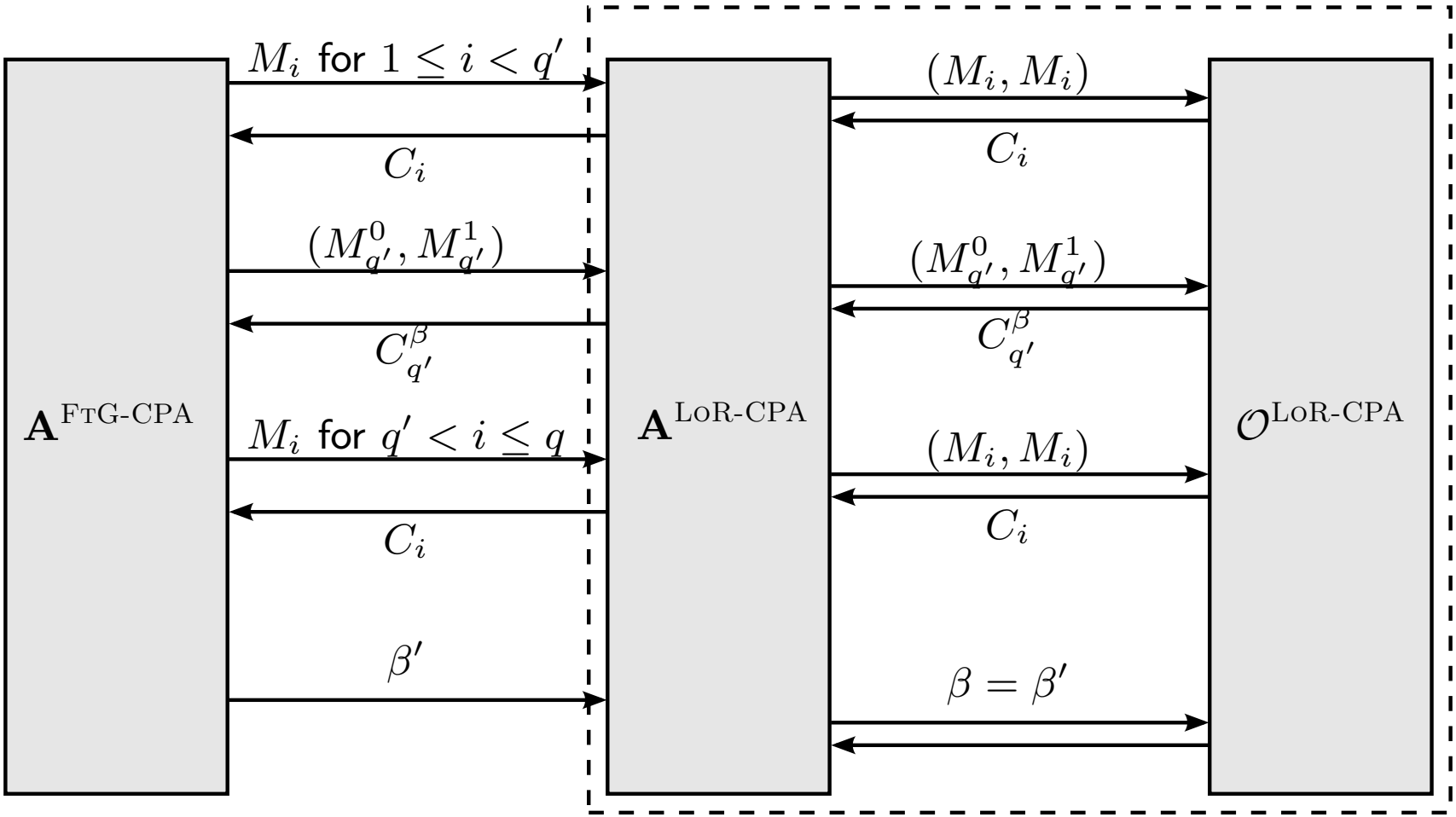
It holds:

$$\mathbf{Adv}(\mathbf{A}^{\text{SEM-CPA}}) = \mathbf{Adv}(\mathbf{A}^{\text{FTG-CPA}})$$

Task 1c) LoR-CPA $\Longrightarrow$ FtG-CPA



Task 1c) LoR-CPA $\Longrightarrow$ FtG-CPA



- **Querying:** $\mathbf{A}^{\text{LoR-CPA}}$ submits M_i twice to its oracle
- **Challenge/Guessing:** Exactly as in FtG-CPA game

$$\mathbf{Adv}(\mathbf{A}^{\text{LoR-CPA}}) = \mathbf{Adv}(\mathbf{A}^{\text{FtG-CPA}})$$

Parity Security

- For all n-bit strings $X = (x_1, \ldots, x_n)$:

$$\textsc{Parity}(X) = x_1 \oplus x_2 \oplus \ldots \oplus x_n$$

Parity-Chosen-Plaintext-Security (Par-CPA) Experiment

The oracle chooses $K \xleftarrow{\$} \{0,1\}^k$

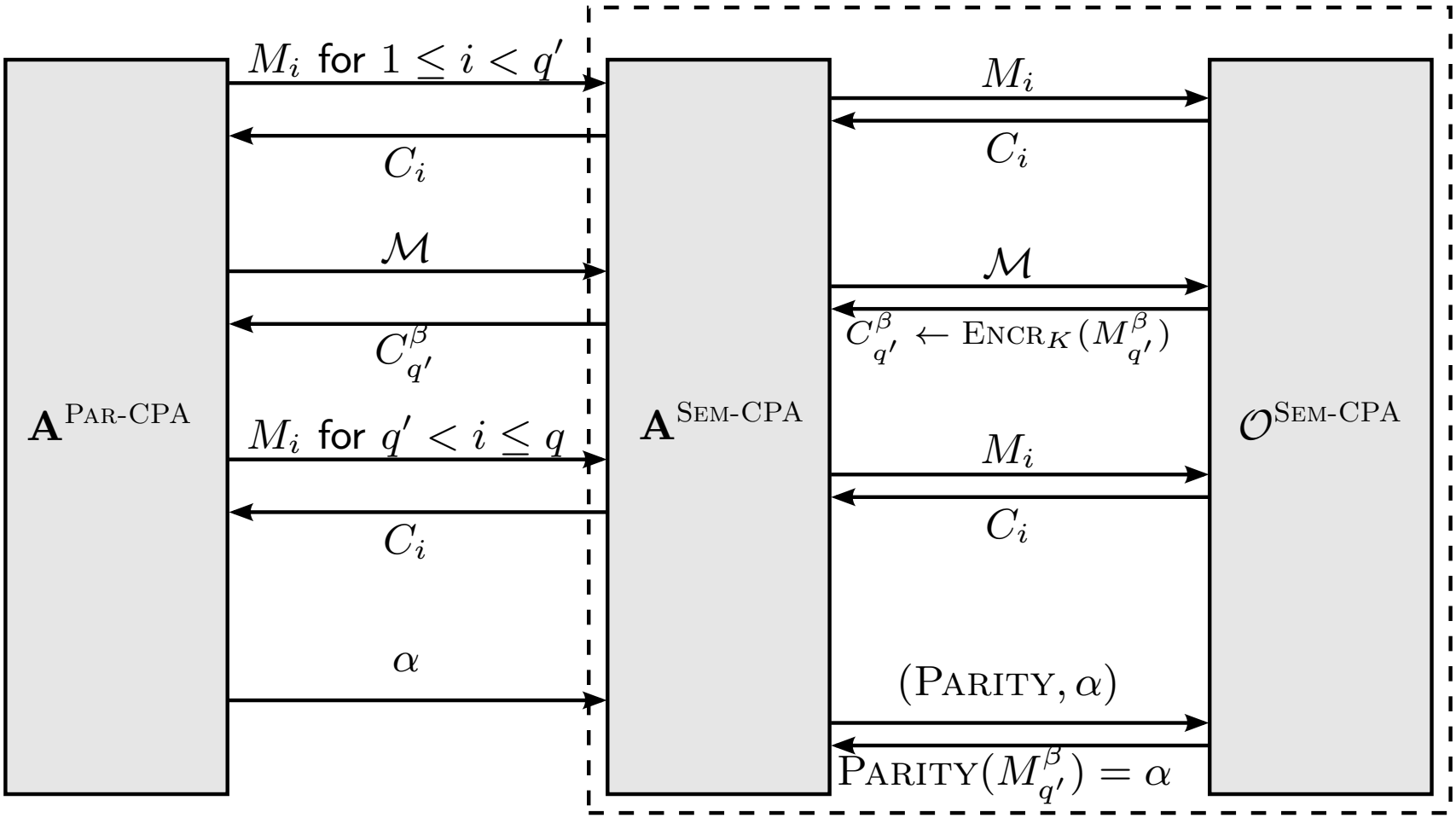
1. For $1 \leq i \leq q' < q$:
   - Eve chooses $M_i \in \{0,1\}^n$ and asks the oracle for $C_i \leftarrow \textsc{Encr}_K(M_i)$.
2. Eve chooses a distribution $\mathcal{M}$ of n-bit plaintexts and sends $\mathcal{M}$ to the oracle.
3. The oracle chooses uniformly at random a message $M \xleftarrow{\$}_{\mathcal{M}} \{0,1\}^n$ according to $\mathcal{M}$ and responds with $C \leftarrow \textsc{Encr}_K(M)$.
4. For $q' + 1 \leq i \leq q$:
   - Eve chooses $M_i \in \{0,1\}^n$ and asks the oracle for $C_i \leftarrow \textsc{Encr}_K(M_i)$
5. Eve outputs a bit $\beta \in \{0,1\}$. She wins iff $\textsc{Parity}(M) = \beta$.

Task 2: Parity Security

a) Prove (or disprove): $\textsc{Sem}\text{-CPA} \implies \textsc{Par}\text{-CPA}$

b) Prove (or disprove): $\textsc{Par}\text{-CPA} \implies \textsc{Sem}\text{-CPA}$

Task 2a) SEM-CPA $\Longrightarrow$ PAR-CPA



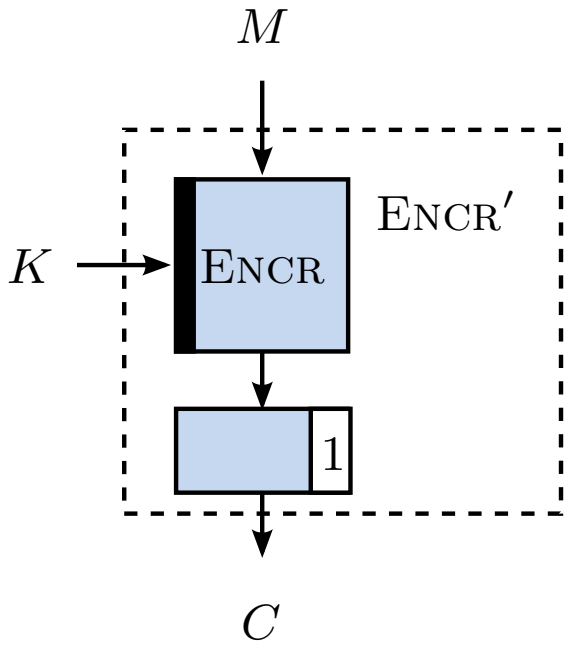
Task 2a) SEM-CPA $\Longrightarrow$ PAR-CPA

- **Initialization:** As in usual SEM-CPA game
- **Querying:** $\mathbf{A}^{\text{SEM-CPA}}$ simply forwards queries from and to $\mathbf{A}^{\text{FTG-CPA}}$
- **Guessing:**
  - $\mathbf{A}^{\text{PAR-CPA}}$ outputs β' as guess for PARITY (M)
  - $\mathbf{A}^{\text{SEM-CPA}}$ chooses $f(M) := \text{PARITY}(M)$ and $\alpha = \beta'$.

$$\mathbf{Adv}(\mathbf{A}^{\text{SEM-CPA}}) = \mathbf{Adv}(\mathbf{A}^{\text{PAR-CPA}})$$

Task 2b) PAR-CPA $\not\Rightarrow$ SEM-CPA

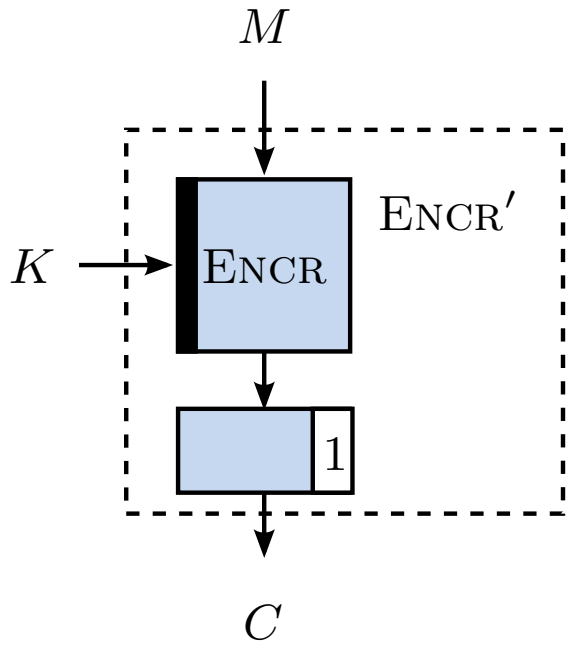
- Assume: SEM-CPA-secure $\textsc{Encr} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$
- $\textsc{lsb} : \{0,1\}^n \to \{0,1\}$ returns the least significant bit
- Define: $\textsc{Encr}' : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$:



$$\textsc{Encr}'_K(M) := \textsc{Encr}_K(M)[n..2] \,\|\, \textsc{lsb}(M).$$

Task 2b) PAR-CPA $\not\Rightarrow$ SEM-CPA

- Assume: SEM-CPA-secure $\textsc{Encr} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$
- $\textsc{lsb} : \{0,1\}^n \to \{0,1\}$ returns the least significant bit
- Define: $\textsc{Encr}' : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$:



$$\textsc{Encr}'_K(M) := \textsc{Encr}_K(M)[n..2] \,\|\, \textsc{lsb}(M).$$

- Clearly: $\textsc{Encr}'$ is **not** SEM-CPA-secure, but can be PAR-CPA-secure
- It follows: PAR-CPA $\not\Rightarrow$ SEM-CPA

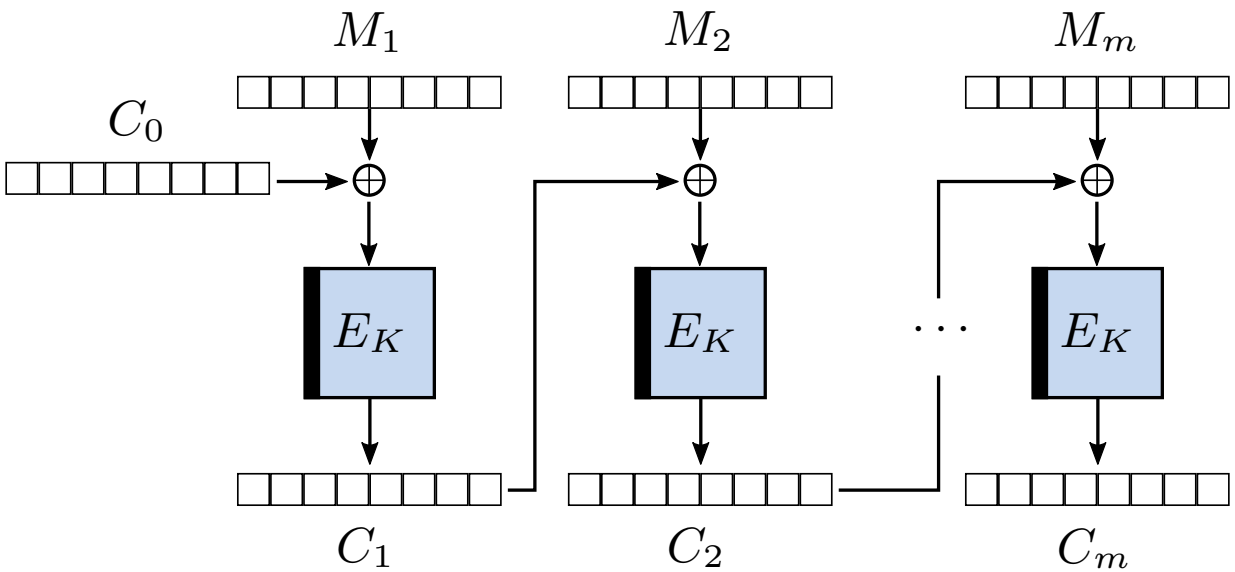
Task 2b) PAR-CPA $\not\Longrightarrow$ SEM-CPA

- Define $\mathbf{A}^{\text{SEM-CPA}}$:
  - Chooses $\mathcal{M}$ as the uniform distribution over all n-bit plaintexts
  - Derive $\alpha \leftarrow \text{LSB}(C_{q'})$
  - Provide $f(M) := \text{LSB}(M)$ and α as final steps to the oracle.
- $\mathbf{A}^{\text{SEM-CPA}}$ always wins the SEM-CPA-game against ENCR$'$
- **But:** Assuming ENCR is SEM-CPA-secure and $n > 1$:
  $\Longrightarrow$ No information about parity in ciphertexts
  (For $n = 1$, the leaked LSB would be the parity)

Task 3

Padding-oracle Attack on CBC

- System: AES-CBC-encryption (1 block = 16 bytes)
- Known: Ciphertext $(C_0, \ldots, C_m)$
- Goal: Recover the original plaintext $(M_1, \ldots, M_m)$



Task 3

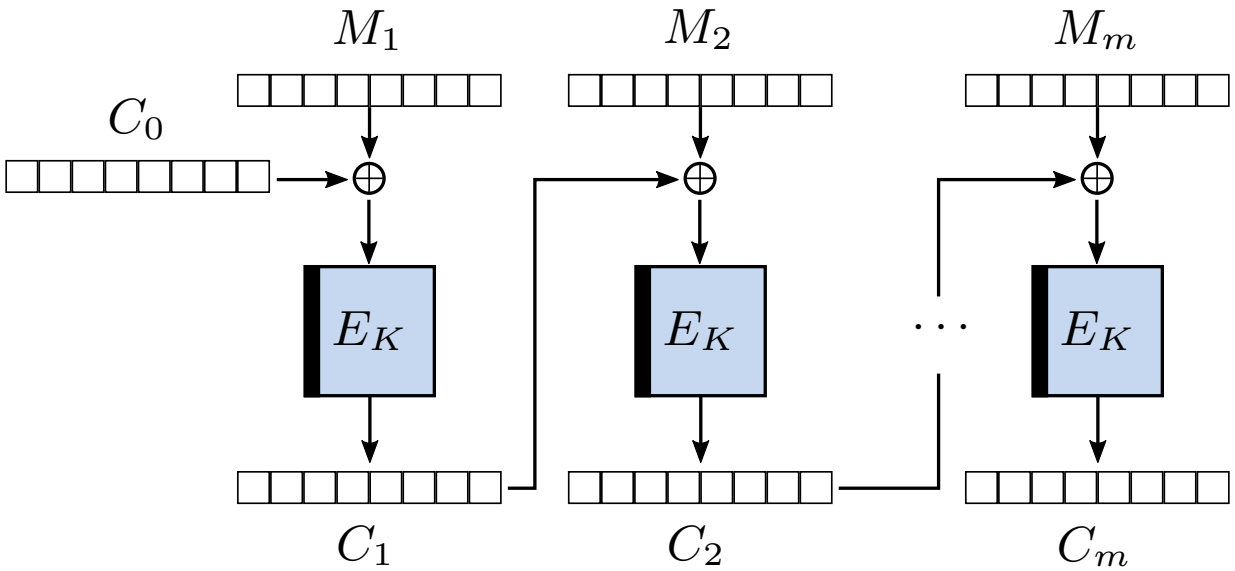
Padding-oracle Attack on CBC

Padding:

$$N = 16 - (|M| \bmod 16)$$
$$M = M \,\|\, (\langle N \rangle)^N$$

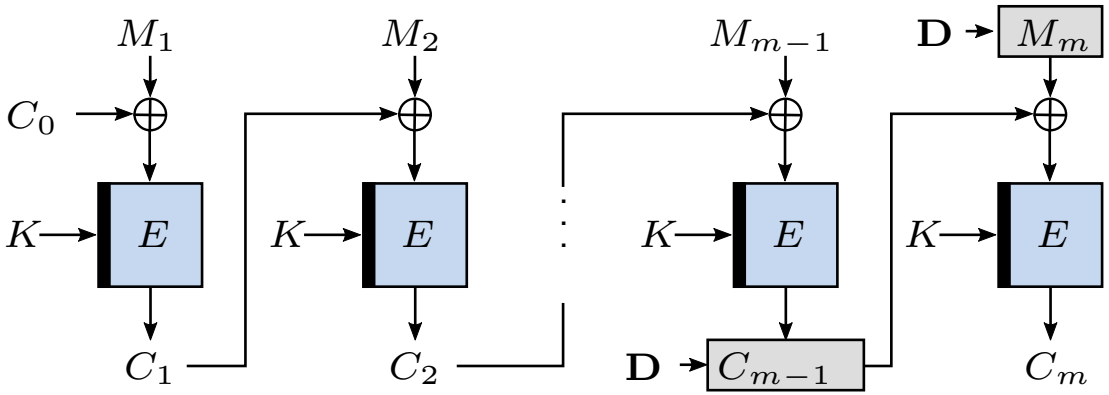
E.g.:

$$\mathsf{pad}((M_1, \ldots, M_{15}) = (M_1, \ldots, M_{15}, 1)$$
$$\mathsf{pad}((M_1, \ldots, M_7)) = (M_1, \ldots, M_7, 9, \ldots, 9)$$
$$\mathsf{pad}((M_1, \ldots, M_{16})) = (M_1, \ldots, M_{16}, 16, \ldots, 16).$$



Task 3

Padding-oracle Attack on CBC



1: **for all** Blocks i from $m-1$ downto 0 **do**
2: $D := (D^{15}, \ldots, D^0) = (0, \ldots, 0)$
3: **for all** Bytes j from 0 to 15 **do**
4: **for** v from 0 to 255 **do**
5: Compute Byte $\mathbf{D}^j := \mathbf{v} \oplus (\mathbf{j}+\mathbf{1})$
6: Ask for the decryption of
7: $C' := (C_0, \ldots, C_{i-1}, \mathbf{C_i} \oplus \mathbf{D}, C_{i+1})$
8: **if** C' is deemed *valid* **then**
9: Store byte $M_{i+1}^j := v$
10: For all $k \in \{0, \ldots, j\}$: $D^k := M_{i+1}^j \oplus (j+1) \oplus (j+2)$
11: Guess next byte (**goto** 3)
12: **return** The recovered plaintext $M = (M_1, \ldots, M_m)$

Recap

- Reductionist Proofs
- Encryption $\neq$ Authenticated Encryption

Questions?