Quantitative Security of Block Ciphers: Designs and Cryptanalysis Tools

Thomas Baignères, EPFL

Crypto Group
Information Security Group
@UCL
February 5, 2009

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

UCL Université catholique de Louvain

Prologue

Cryptography: the Basics

Originally, cryptography aims at ensuring confidentiality through an insecure channel.

Cryptography: the Basics

Originally, cryptography aims at ensuring confidentiality through an insecure channel.

Bob

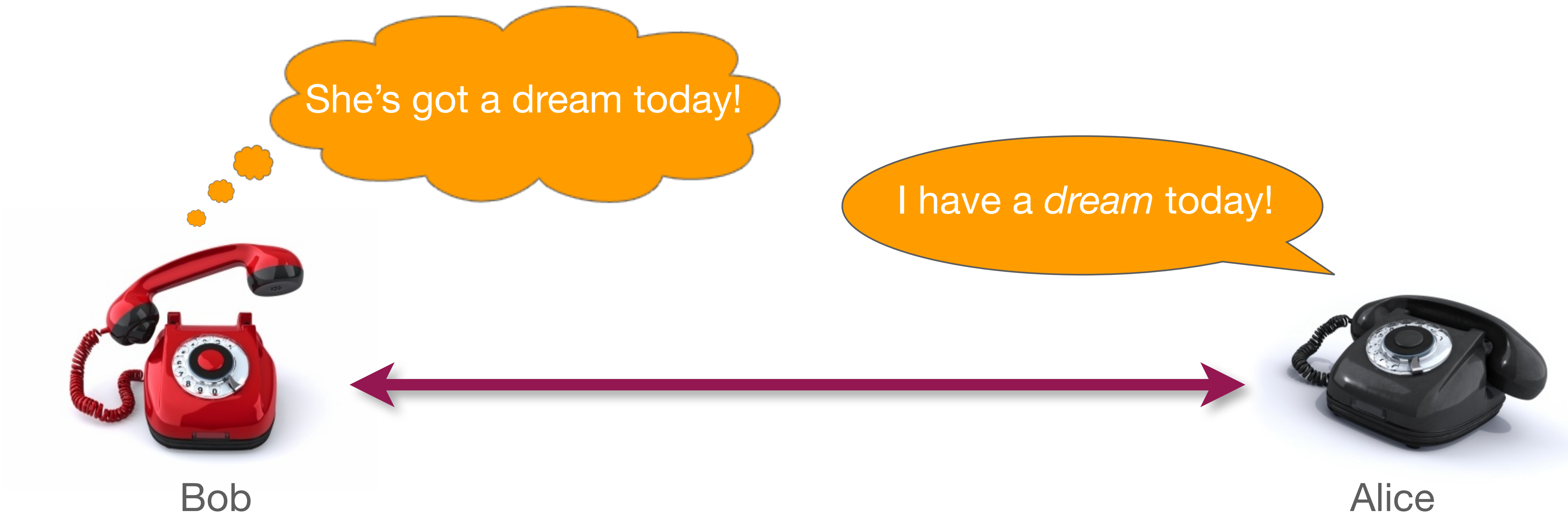
Alice

Cryptography: the Basics

Originally, cryptography aims at ensuring confidentiality through an insecure channel.

Cryptography: the Basics

Originally, cryptography aims at ensuring confidentiality through an insecure channel.



Cryptography: the Basics

Originally, cryptography aims at ensuring confidentiality through an insecure channel.

Cryptography: the Basics

Originally, cryptography aims at ensuring confidentiality through an insecure channel.

Cryptography: the Basics

Originally, cryptography aims at ensuring confidentiality through an insecure channel.

Bob

Alice

Cryptography: the Basics

Originally, cryptography aims at ensuring confidentiality through an insecure channel.

Cryptography: the Basics

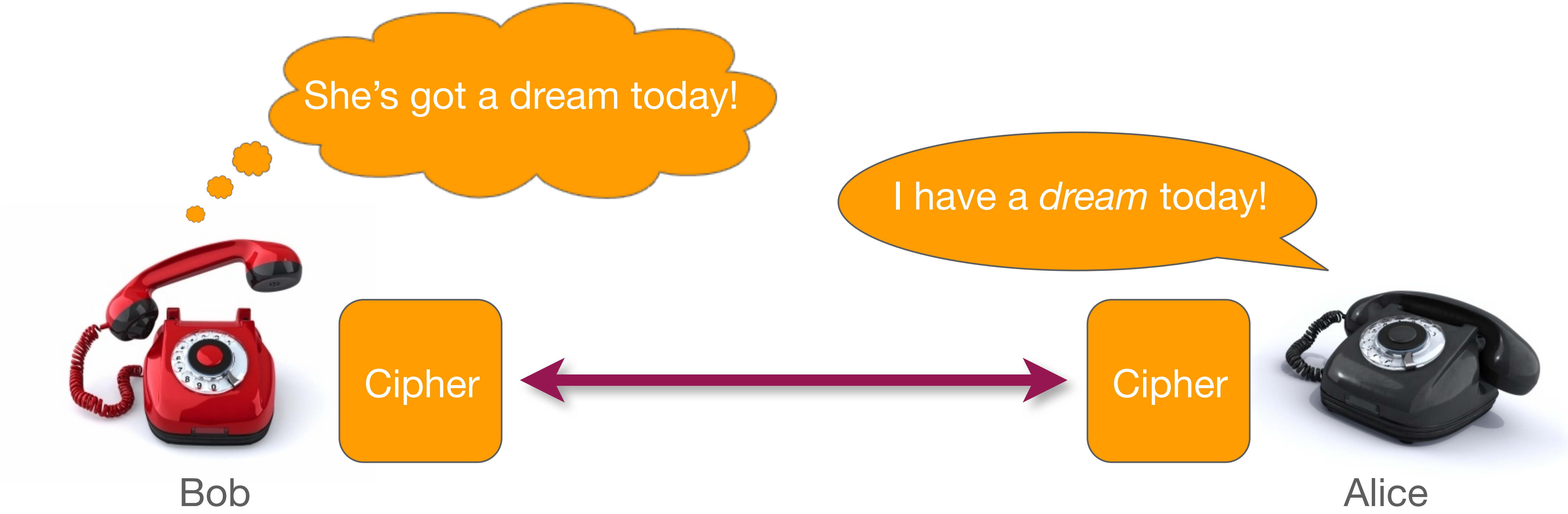
Originally, cryptography aims at ensuring confidentiality through an insecure channel.

Cryptography: the Basics

Originally, cryptography aims at ensuring confidentiality through an insecure channel.

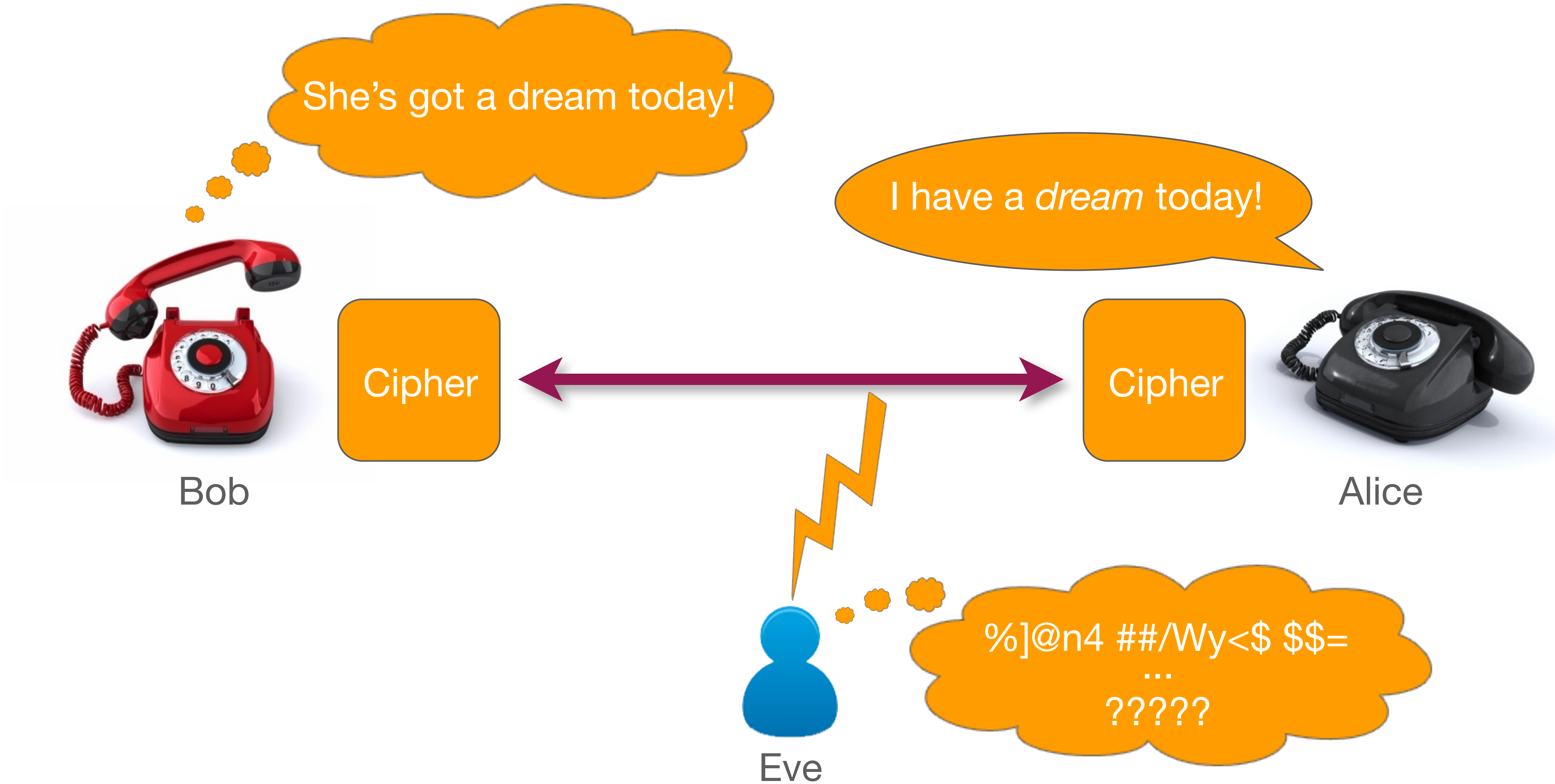
Cryptography: the Basics

Originally, cryptography aims at ensuring confidentiality through an insecure channel.



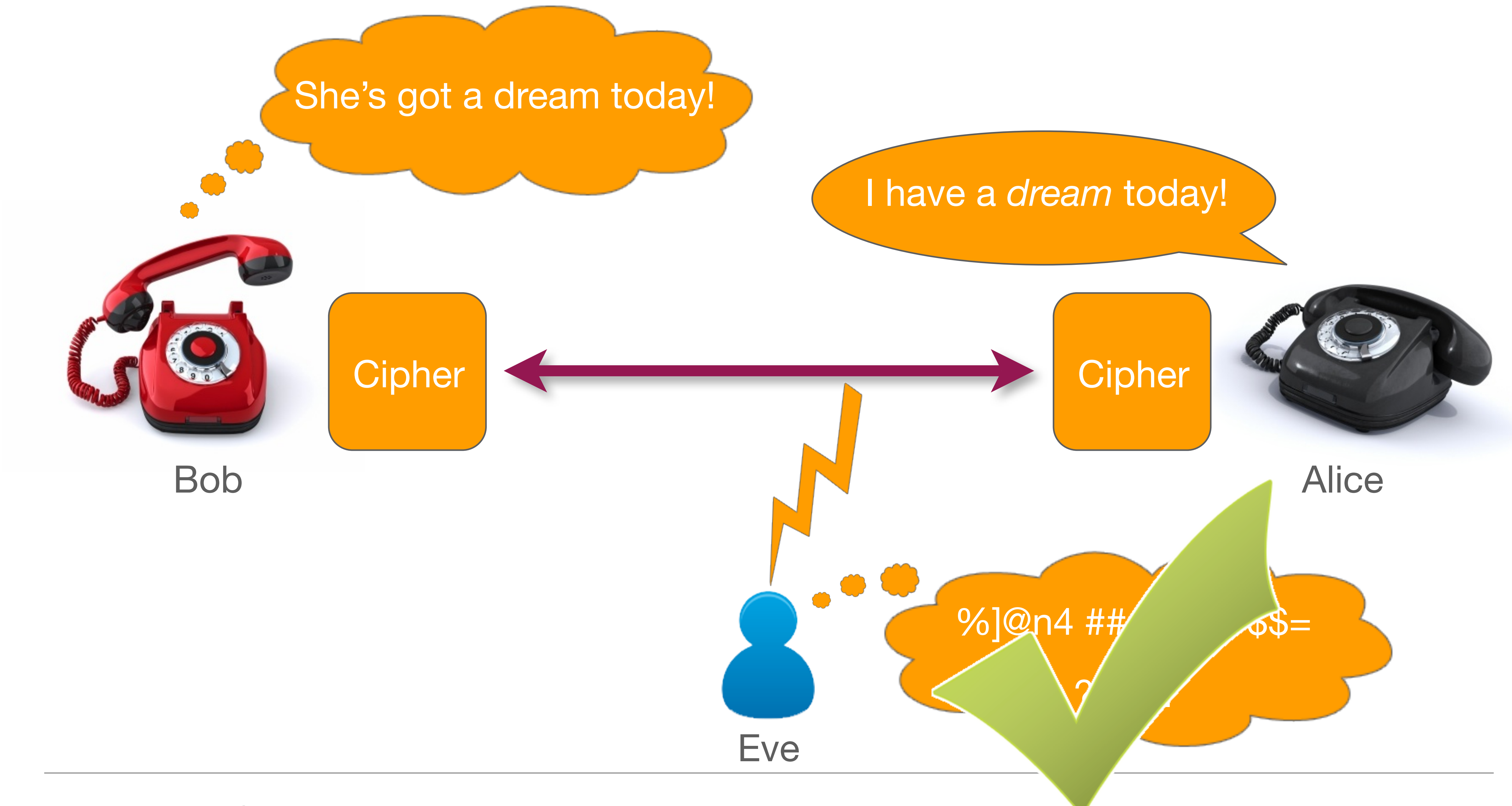
Cryptography: the Basics

Originally, cryptography aims at ensuring confidentiality through an insecure channel.



Cryptography: the Basics

Originally, cryptography aims at ensuring confidentiality through an insecure channel.



What should we expect from the cipher?

What should we expect from the cipher?

Intuitively, turning *%]@n4 ##/Wy<$ $$=* into *I have a dream today!* should be hard, except for Alice and Bob.

What should we expect from the cipher?

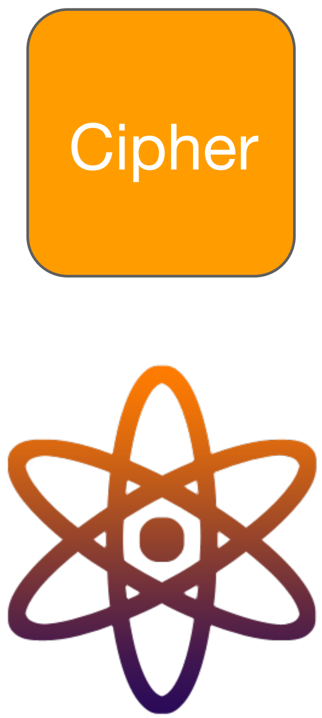
Intuitively, turning *%]@n4 ##/Wy<$ $$=* into *I have a dream today!* should be hard, except for Alice and Bob.

Fact: cryptographers are paranoid ⇒ they sometimes require more !

It should be hard for Eve to guess wether she’s looking at an encrypted message (ciphertext) or to pure rubish (random string).

The security requirements in terms of a game...

The security requirements in terms of a game...



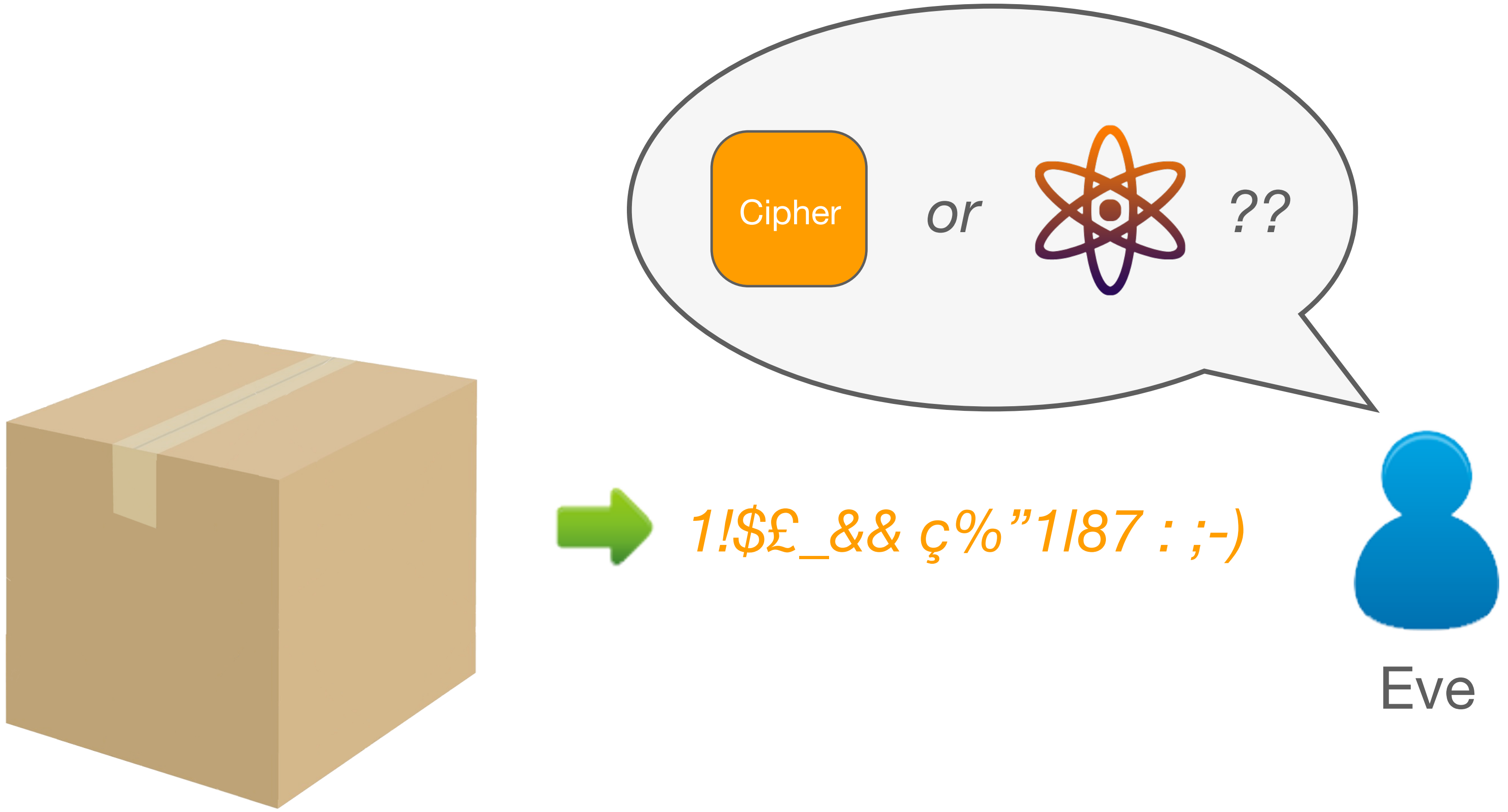
The security requirements in terms of a game...

The security requirements in terms of a game...

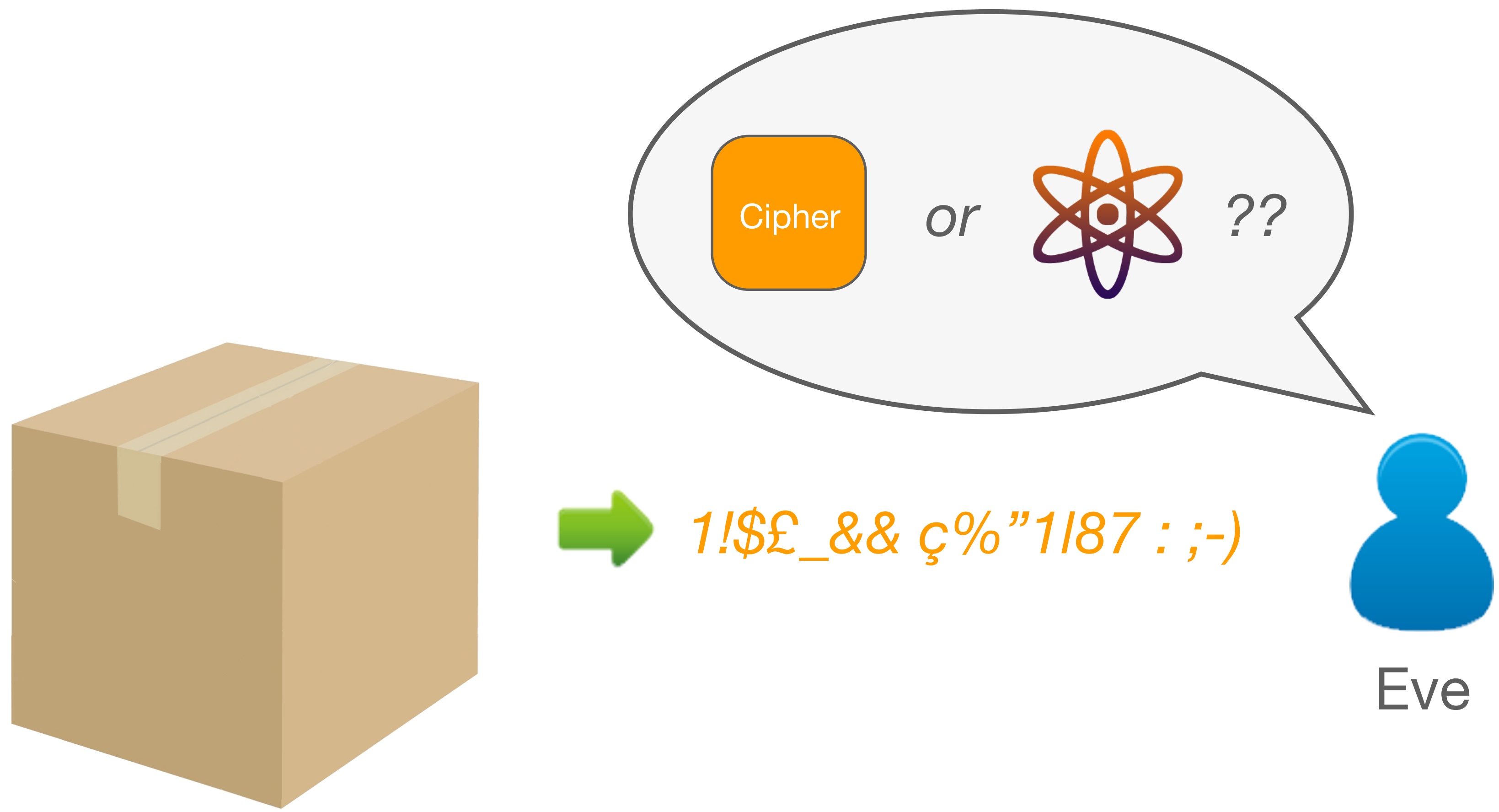
The security requirements in terms of a game...

1!$£_&& ç%”1l87 : ;-)

The security requirements in terms of a game...

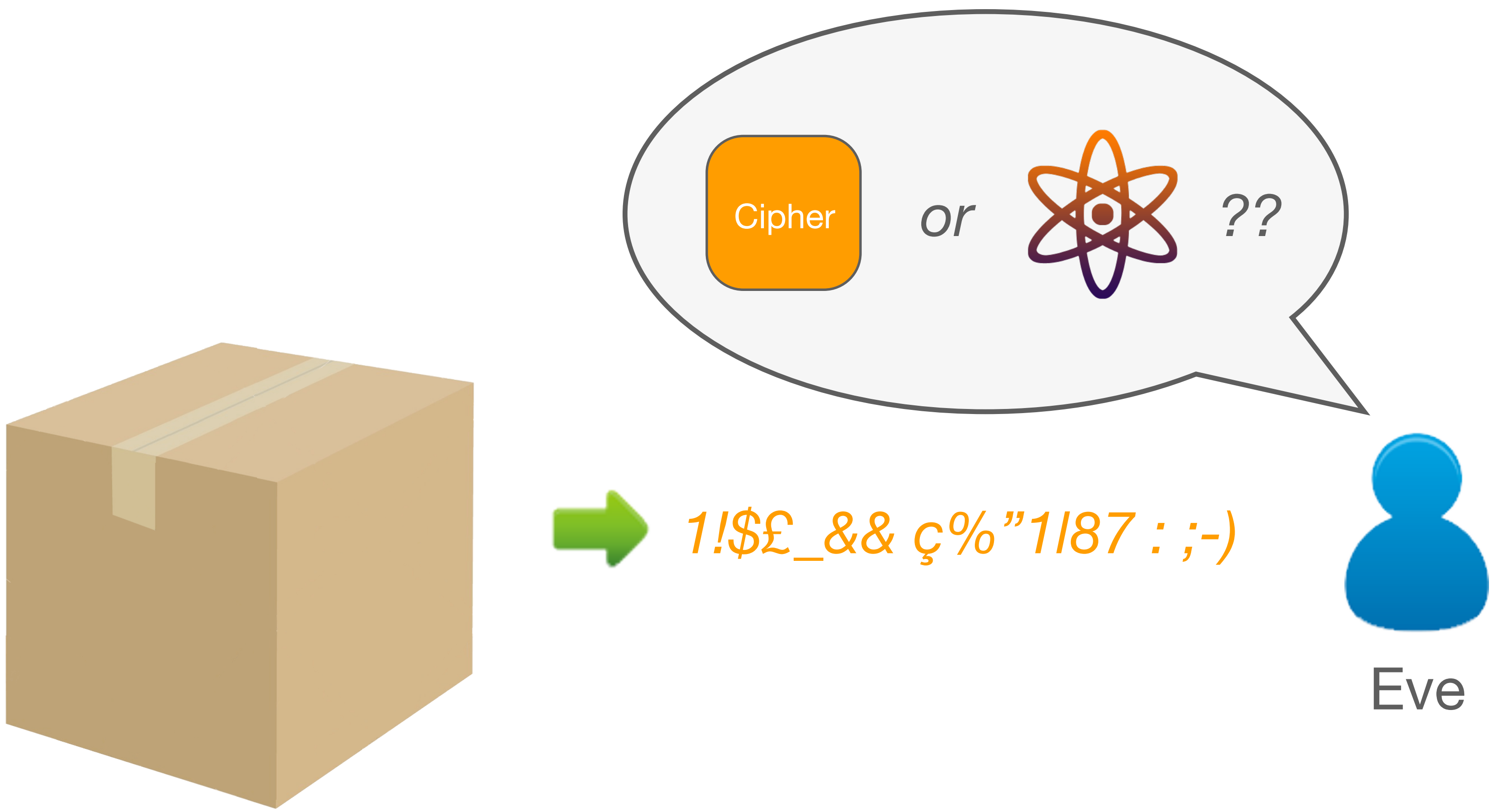


The security requirements in terms of a game...



- Eve wins if she guesses correctly.

The security requirements in terms of a game...



- Eve wins if she guesses correctly.
- Objective for the cryptographer: make sure that Eve cannot do better than guessing correctly 50% of the time.

Part I: On the (In)Security of Block Ciphers: Tools for the Security Analysis

Outline

Distinguishers between two sources

Projection-based distinguishers
between two sources

Practical Implications for block ciphers

Outline

- ➡ Distinguishers between two sources
- Projection-based distinguishers between two sources
- Practical Implications for block ciphers

- The game: distinguishing between two sources of randomness
- The optimal solution
- Complexity analysis: How many samples do we need to distinguish with a given efficiency?

Outline

Distinguishers between two sources

Projection-based distinguishers between two sources

Practical Implications for block ciphers

- What if the optimal solution cannot be implemented?
- Distinguishing in practice using compression
- Example: Generalized linear distinguisher

Outline

Distinguishers between two sources

Projection-based distinguishers between two sources

➡ Practical Implications for block ciphers

- Cryptanalysis of SAFER K/SK
- DEAN

Outline

Distinguishers between two sources

Projection-based distinguishers between two sources

Practical Implications for block ciphers

- Cryptanalysis of SAFER K/SK
- DEAN

[BJVa04] [BSVsac07] [BVicits08]

Part I: On the (In)Security of Block Ciphers:

Tools for the Security Analysis

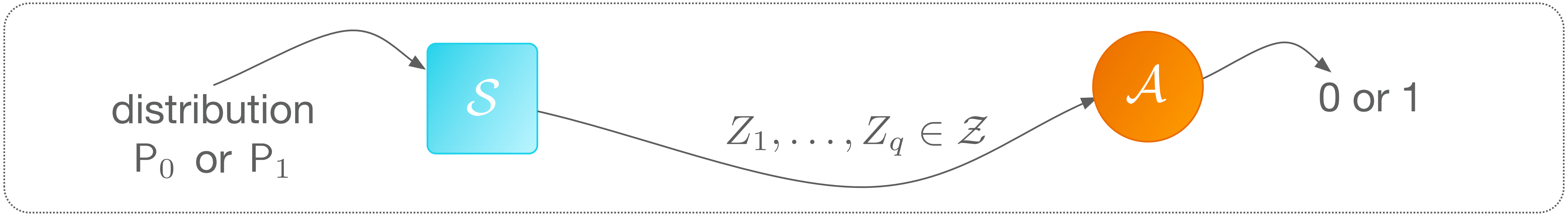
- Distinguisher between two Sources

The Game

- P_0 and P_1 are two arbitrary distributions over a finite set $\mathcal{Z}$.

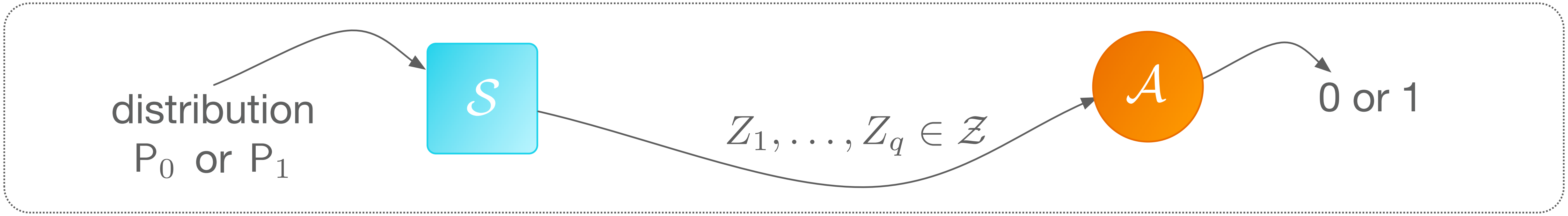
The Game

- P_0 and P_1 are two arbitrary distributions over a finite set $\mathcal{Z}$.



The Game

- P_0 and P_1 are two arbitrary distributions over a finite set $\mathcal{Z}$.



- The ability of $\mathcal{A}$ to distinguish P_0 from P_1 is its advantage:

$$\mathrm{Adv}_{\mathcal{A}}(\mathsf{P}_0, \mathsf{P}_1) = |\Pr_{\mathsf{P}_0}[\mathcal{A}(Z_1, \ldots, Z_q) = 1] - \Pr_{\mathsf{P}_1}[\mathcal{A}(Z_1, \ldots, Z_q) = 1]|$$

Example: Biased Die (pl. dice, from Old French “dé”)

Example: Biased Die (pl. dice, from Old French “dé”)

Example: Biased Die (pl. dice, from Old French “dé”)

$P_0 = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$

Example: Biased Die (pl. dice, from Old French "dé")

Probability of throwing a '1' with the first dice

$$P_0 = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$$

Probability of throwing a '4' with the first dice

Example: Biased Die (pl. dice, from Old French “dé”)

$\mathsf{P}_0 = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$

$\mathsf{P}_1 = (\frac{1}{6}, \frac{1}{6}, \frac{2}{6}, 0, \frac{1}{6}, \frac{1}{6})$

Example: Biased Die (pl. dice, from Old French “dé”)

$P_0 = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$

No way to throw a ‘4’ with this dice...

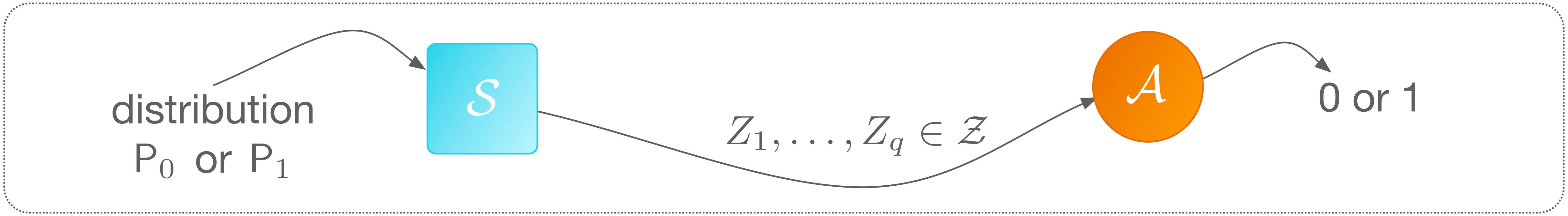
$P_1 = (\frac{1}{6}, \frac{1}{6}, \frac{2}{6}, 0, \frac{1}{6}, \frac{1}{6})$

That’s a VERY biased dice!

Example: Biased Die (pl. dice, from Old French “dé”)

$P_0 = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$

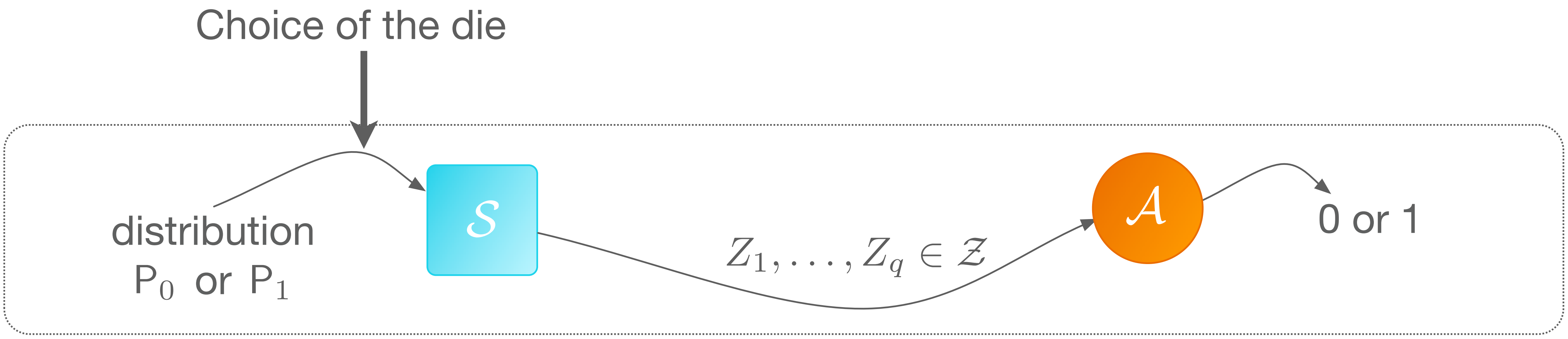
$P_1 = (\frac{1}{6}, \frac{1}{6}, \frac{2}{6}, 0, \frac{1}{6}, \frac{1}{6})$



Example: Biased Die (pl. dice, from Old French “dé”)

$\mathrm{P}_0 = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$

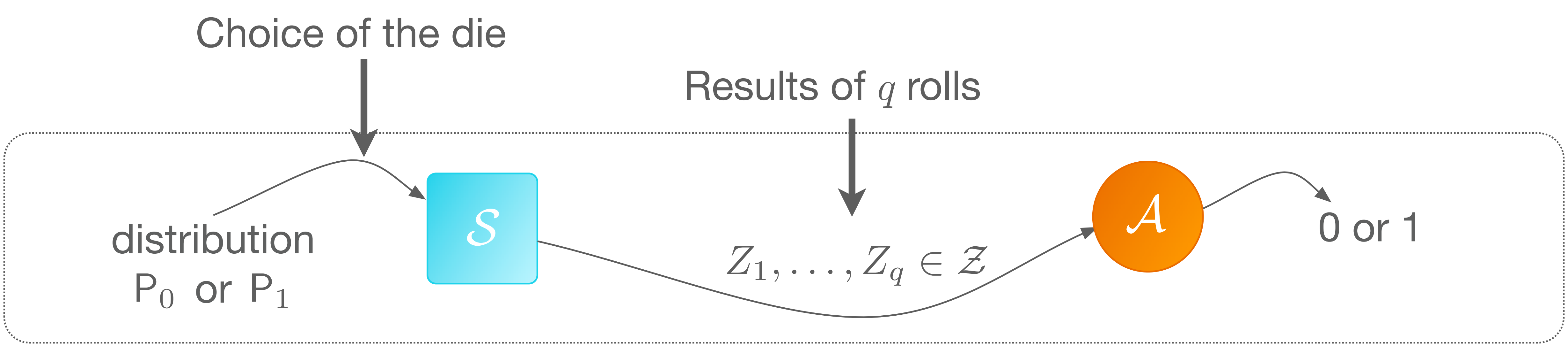
$\mathrm{P}_1 = (\frac{1}{6}, \frac{1}{6}, \frac{2}{6}, 0, \frac{1}{6}, \frac{1}{6})$



Example: Biased Die (pl. dice, from Old French “dé”)

$P_0 = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$

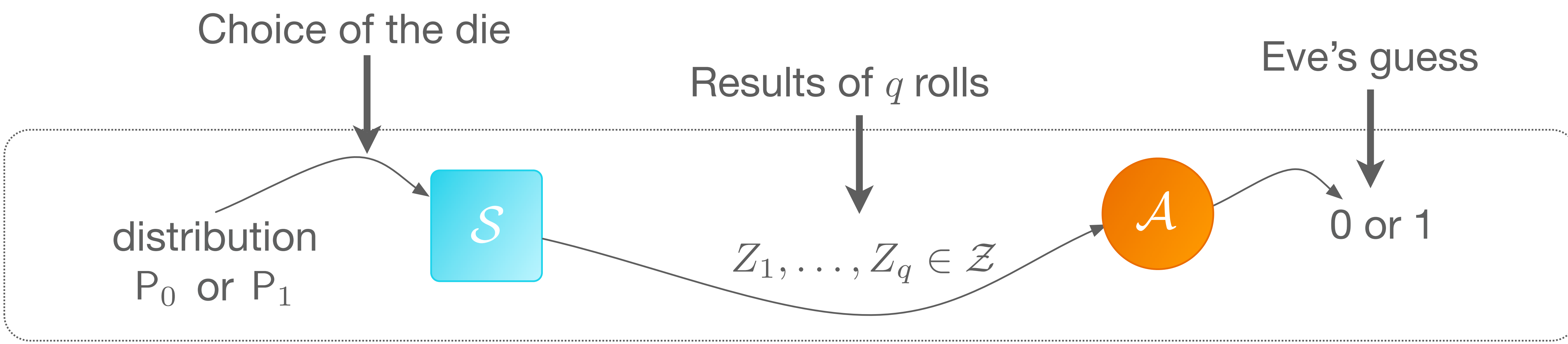
$P_1 = (\frac{1}{6}, \frac{1}{6}, \frac{2}{6}, 0, \frac{1}{6}, \frac{1}{6})$



Example: Biased Die (pl. dice, from Old French “dé”)

$\mathsf{P}_0 = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$

$\mathsf{P}_1 = (\frac{1}{6}, \frac{1}{6}, \frac{2}{6}, 0, \frac{1}{6}, \frac{1}{6})$



An Optimal Distinguisher

- $\mathcal{A}$ is computationally unbounded (deterministic)
- q samples are independent (order is irrelevant)
- What matters: the number of occurrences of each symbol of $\mathcal{Z}$ in the string $Z_1, \ldots, Z_q$
- Equivalently: the type $\mathsf{P}_{Z_1,\ldots,Z_q}$ of the sequence:

$$\mathsf{P}_{Z_1,\ldots,Z_q}[a] = \frac{\#\{i \, : \, Z_i = a\}}{q}$$

An Optimal Distinguisher

- $\mathcal{A}$ is computationally unbounded (deterministic)
- q samples are independent (order is irrelevant)
- What matters: the number of occurrences of each symbol of $\mathcal{Z}$ in the string $Z_1, \ldots, Z_q$
- Equivalently: the type $\mathsf{P}_{Z_1,\ldots,Z_q}$ of the sequence:

$$\mathsf{P}_{Z_1,\ldots,Z_q}[a] = \frac{\#\{i \,:\, Z_i = a\}}{q}$$

- Example: $\mathcal{Z} = \{1, 2, 3, 4, 5, 6\}$, $q = 31$ and

$$Z_1, Z_2, \ldots, Z_{31} = 1\,6\,3\,5\,6\,2\,2\,3\,1\,6\,3\,2\,6\,3\,6\,5\,5\,1\,2\,3\,6\,5\,1\,3\,2\,2\,5\,6\,5\,3\,1$$

An Optimal Distinguisher

- $\mathcal{A}$ is computationally unbounded (deterministic)
- q samples are independent (order is irrelevant)
- What matters: the number of occurrences of each symbol of $\mathcal{Z}$ in the string $Z_1, \ldots, Z_q$
- Equivalently: the type $\mathsf{P}_{Z_1,\ldots,Z_q}$ of the sequence:

$$\mathsf{P}_{Z_1,\ldots,Z_q}[a] = \frac{\#\{i \,:\, Z_i = a\}}{q}$$

- Example: $\mathcal{Z} = \{1, 2, 3, 4, 5, 6\}$, $q = 31$ and

$$Z_1, Z_2, \ldots, Z_{31} = 1\,6\,3\,5\,6\,2\,2\,3\,1\,6\,3\,2\,6\,3\,6\,5\,5\,1\,2\,3\,6\,5\,1\,3\,2\,2\,5\,6\,5\,3\,1$$

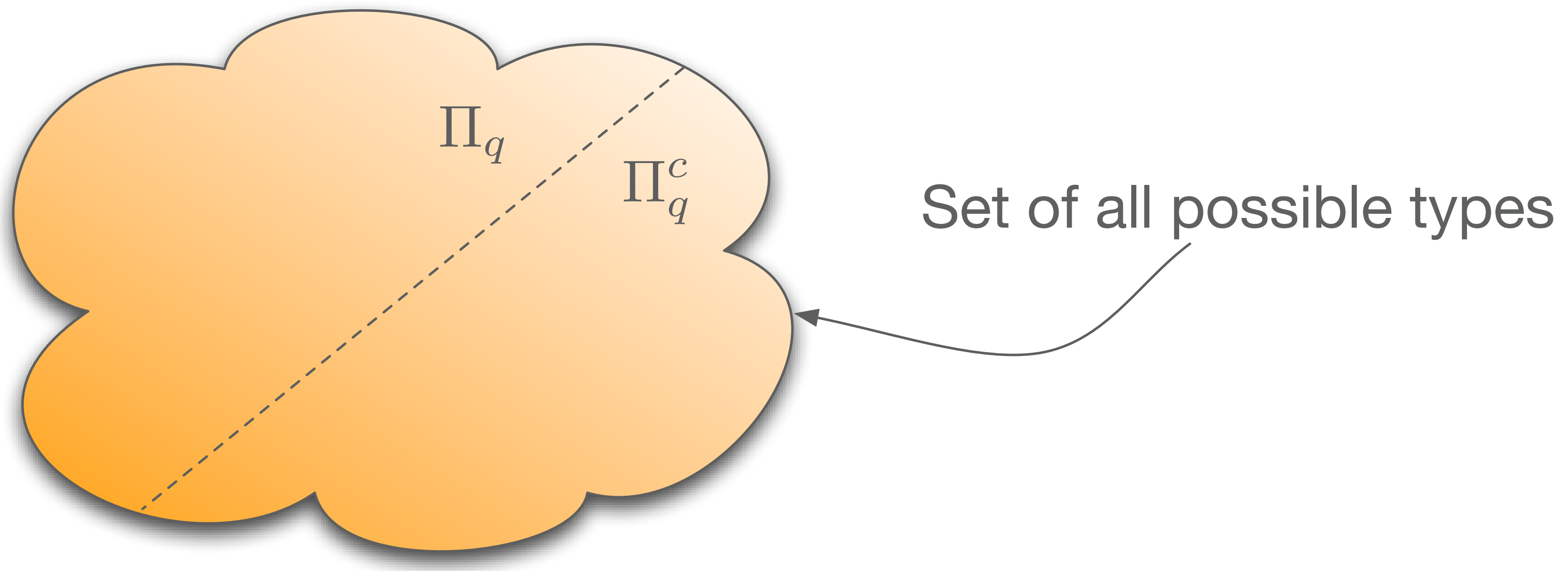
$$\mathsf{P}_{Z_1,\ldots,Z_{31}}[1] = \frac{5}{31} \qquad \mathsf{P}_{Z_1,\ldots,Z_{31}}[2] = \frac{6}{31} \qquad \mathsf{P}_{Z_1,\ldots,Z_{31}}[3] = \frac{7}{31}$$

$$\mathsf{P}_{Z_1,\ldots,Z_{31}}[4] = 0 \qquad \mathsf{P}_{Z_1,\ldots,Z_{31}}[5] = \frac{6}{31} \qquad \mathsf{P}_{Z_1,\ldots,Z_{31}}[6] = \frac{7}{31}$$

An Optimal Distinguisher

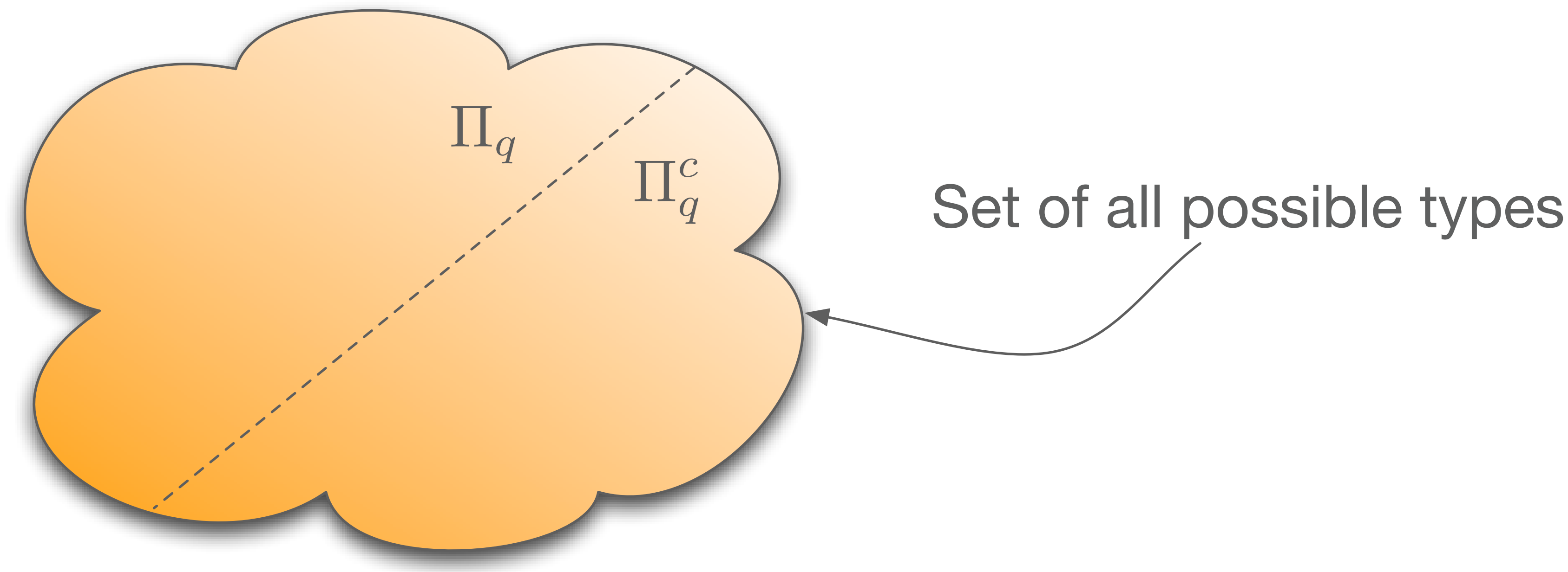
$\mathcal{A}$ uniquely determined by Π_q: $\mathsf{P}_{Z_1,\ldots,Z_q} \in \Pi_q \Leftrightarrow \mathcal{A}(Z_1,\ldots,Z_q) = 1$

An Optimal Distinguisher



$\mathcal{A}$ uniquely determined by Π_q: $\mathsf{P}_{Z_1,\ldots,Z_q} \in \Pi_q \Leftrightarrow \mathcal{A}(Z_1,\ldots,Z_q) = 1$

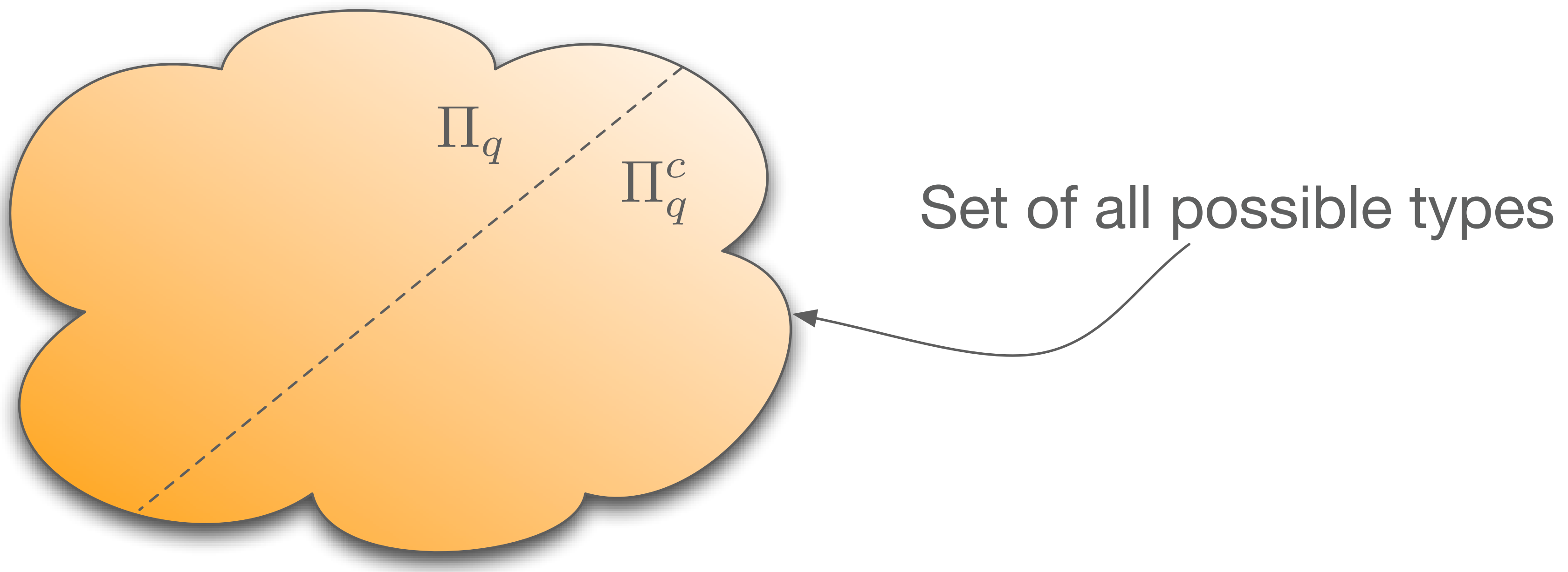
An Optimal Distinguisher



$\mathcal{A}$ uniquely determined by Π_q: $\mathsf{P}_{Z_1,\ldots,Z_q} \in \Pi_q \Leftrightarrow \mathcal{A}(Z_1,\ldots,Z_q) = 1$

Number of such Π_q is finite ➡ Number of possible adversaries is finite.

An Optimal Distinguisher



$\mathcal{A}$ uniquely determined by Π_q: $\quad \mathrm{P}_{Z_1,\ldots,Z_q} \in \Pi_q \Leftrightarrow \mathcal{A}(Z_1,\ldots,Z_q) = 1$

Number of such Π_q is finite ➡ Number of possible adversaries is finite.

An optimal distinguisher exists!

? Can it be determined?

An Optimal Distinguisher

Using maximum-likelihood techniques, the q-limited distinguisher $\mathcal{A}^\star$ which outputs 1 when by

$$\mathrm{D}(\mathsf{P}_{Z_1,\ldots,Z_q} \| \mathsf{P}_1) \leq \mathrm{D}(\mathsf{P}_{Z_1,\ldots,Z_q} \| \mathsf{P}_0)$$

can be shown to be optimal.

An Optimal Distinguisher

Using maximum-likelihood techniques, the q-limited distinguisher $\mathcal{A}^\star$ which outputs 1 when by

$$\mathrm{D}(\mathsf{P}_{Z_1,\ldots,Z_q}\|\mathsf{P}_1) \leq \mathrm{D}(\mathsf{P}_{Z_1,\ldots,Z_q}\|\mathsf{P}_0)$$

can be shown to be optimal.

$$\mathrm{D}(p\|q) = \sum_{a\in\mathcal{Z}} p[a] \log \frac{p[a]}{q[a]}$$

always non-negative, 0 iff $p=q$, infinite iff $\mathsf{Supp}(p) \not\subseteq \mathsf{Supp}(q)$

Data Complexity Analysis

Using the theory of types & Sanov's theorem asymptotic data complexity of $\mathcal{A}^\star$.

Data Complexity Analysis

Using the theory of types & Sanov's theorem ➡ asymptotic data complexity of $\mathcal{A}^{\star}$.

Data Complexity Analysis

Using the theory of types & Sanov's theorem → asymptotic data complexity of $\mathcal{A}^\star$.

Theorem

Let P_0 and P_1 be two distributions s.t. $\mathrm{Supp}(\mathsf{P}_0) \cup \mathrm{Supp}(\mathsf{P}_1) = \mathcal{Z}$. The advantage of $\mathcal{A}^\star$ verifies

$$1 - \mathrm{BestAdv}_q(\mathsf{P}_0, \mathsf{P}_1) \doteq 2^{-q\mathrm{C}(\mathsf{P}_0,\mathsf{P}_1)}$$

where

$$\mathrm{C}(\mathsf{P}_0, \mathsf{P}_1) = - \inf_{0<\lambda<1} \log \sum_{a \in \mathsf{Supp}(\mathsf{P}_0) \cap \mathsf{Supp}(\mathsf{P}_1)} \mathsf{P}_0[a]^{1-\lambda} \mathsf{P}_1[a]^{\lambda}$$

is the Chernoff information between P_0 and P_1.

Data Complexity Analysis

Using the theory of types & Sanov's theorem ➡ asymptotic data complexity of $\mathcal{A}^\star$.

Theorem

Let P_0 and P_1 be two distributions s.t. $\mathrm{Supp}(\mathsf{P}_0) \cup \mathrm{Supp}(\mathsf{P}_1) = \mathcal{Z}$. The advantage of $\mathcal{A}^\star$ verifies

$$1 - \mathrm{BestAdv}_q(\mathsf{P}_0, \mathsf{P}_1) \doteq 2^{-q\mathrm{C}(\mathsf{P}_0,\mathsf{P}_1)}$$

where

$$\mathrm{C}(\mathsf{P}_0, \mathsf{P}_1) = - \inf_{0<\lambda<1} \log \sum_{a \in \mathsf{Supp}(\mathsf{P}_0) \cap \mathsf{Supp}(\mathsf{P}_1)} \mathsf{P}_0[a]^{1-\lambda} \mathsf{P}_1[a]^{\lambda}$$

is the Chernoff information between P_0 and P_1.

Notation: $f(q) \doteq g(q)$ means that $f(q) = g(q)e^{o(q)}$, i.e., $\lim_{q\to\infty} \frac{1}{q} \log \frac{f(q)}{g(q)} = 0$.

Data Complexity Analysis

Using the theory of types & Sanov's theorem ➡ asymptotic data complexity of $\mathcal{A}^\star$.

Theorem

Let P_0 and P_1 be two distributions s.t. $\mathrm{Supp}(\mathsf{P}_0) \cup \mathrm{Supp}(\mathsf{P}_1) = \mathcal{Z}$. The advantage of $\mathcal{A}^\star$ verifies

$$1 - \mathrm{BestAdv}_q(\mathsf{P}_0, \mathsf{P}_1) \doteq 2^{-q\mathrm{C}(\mathsf{P}_0,\mathsf{P}_1)}$$

where

$$\mathrm{C}(\mathsf{P}_0, \mathsf{P}_1) \approx \frac{\|\mathsf{P}_1 - \mathsf{P}_0\|_2^2}{8 \ln 2}$$

is the Chernoff information between P_0 and P_1.

Notation: $f(q) \doteq g(q)$ means that $f(q) = g(q)e^{o(q)}$, i.e., $\lim_{q\to\infty} \frac{1}{q} \log \frac{f(q)}{g(q)} = 0$.

Data Complexity Analysis

Using the theory of types & Sanov's theorem ➡ asymptotic data complexity of $\mathcal{A}^\star$.

Theorem

Let P_0 and P_1 be two distributions s.t. $\mathrm{Supp}(\mathsf{P}_0) \cup \mathrm{Supp}(\mathsf{P}_1) = \mathcal{Z}$. The advantage of $\mathcal{A}^\star$ verifies

$$1 - \mathrm{BestAdv}_q(\mathsf{P}_0, \mathsf{P}_1) \approx 2^{-q\mathrm{C}(\mathsf{P}_0,\mathsf{P}_1)}$$

where

$$\mathrm{C}(\mathsf{P}_0, \mathsf{P}_1) \approx \frac{\|\mathsf{P}_1 - \mathsf{P}_0\|_2^2}{8 \ln 2}$$

is the Chernoff information between P_0 and P_1.

Data Complexity Analysis

Using the theory of types & Sanov's theorem ➡ asymptotic data complexity of $\mathcal{A}^\star$.

Theorem

Let P_0 and P_1 be two distributions s.t. $\mathrm{Supp}(\mathrm{P}_0) \cup \mathrm{Supp}(\mathrm{P}_1)$ ... age of $\mathcal{A}^\star$ verifies

$$1 - \mathrm{BestAdv} ...$$

where

is the ... information between P_0 and P_1.

Heuristic: $q \approx 1/\mathrm{C}(\mathrm{P}_0, \mathrm{P}_1)$ allows $\mathcal{A}^\star$ to reach a non-negligible advantage

Example: Biased Coin

$\mathsf{P}_0 = (\frac{1}{2}, \frac{1}{2})$ $\mathsf{P}_1 = \left(\frac{1}{2}(1-\epsilon), \frac{1}{2}(1+\epsilon)\right)$

Example: Biased Coin

$\mathsf{P}_0 = \left(\frac{1}{2}, \frac{1}{2}\right)$ $\mathsf{P}_1 = \left(\frac{1}{2}(1-\epsilon), \frac{1}{2}(1+\epsilon)\right)$

Example: Biased Coin

$\mathsf{P}_0 = (\frac{1}{2}, \frac{1}{2})$ $\quad\quad$ $\mathsf{P}_1 = \left(\frac{1}{2}(1-\epsilon), \frac{1}{2}(1+\epsilon)\right)$

heads $\quad$ tails

Example: Biased Coin

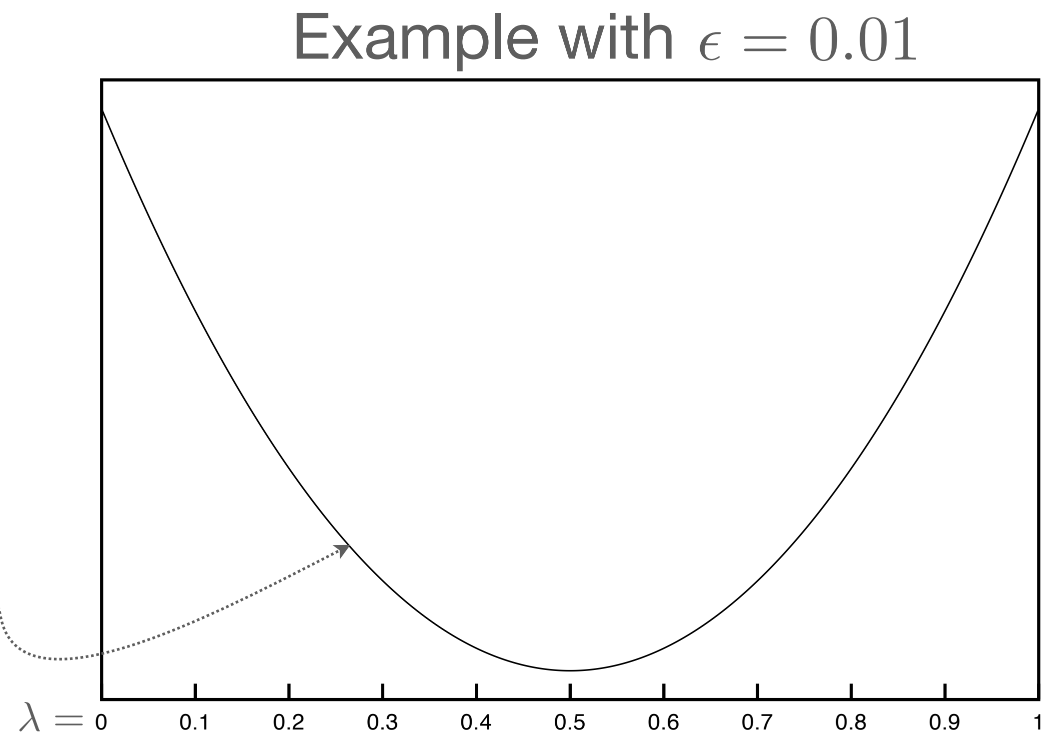
$\mathsf{P}_0 = (\frac{1}{2}, \frac{1}{2})$ $\qquad$ $\mathsf{P}_1 = \left(\frac{1}{2}(1-\epsilon), \frac{1}{2}(1+\epsilon)\right)$

$$\mathrm{C}(\mathsf{P}_0, \mathsf{P}_1) = - \inf_{0<\lambda<1} \log \tfrac{1}{2} \left((1-\epsilon)^\lambda + (1+\epsilon)^\lambda\right)$$

Example: Biased Coin

$\mathsf{P}_0 = \left(\frac{1}{2}, \frac{1}{2}\right)$ $\mathsf{P}_1 = \left(\frac{1}{2}(1-\epsilon), \frac{1}{2}(1+\epsilon)\right)$

$$C(\mathsf{P}_0, \mathsf{P}_1) = -\inf_{0<\lambda<1} \log \frac{1}{2}\left((1-\epsilon)^\lambda + (1+\epsilon)^\lambda\right)$$



Example: Biased Coin

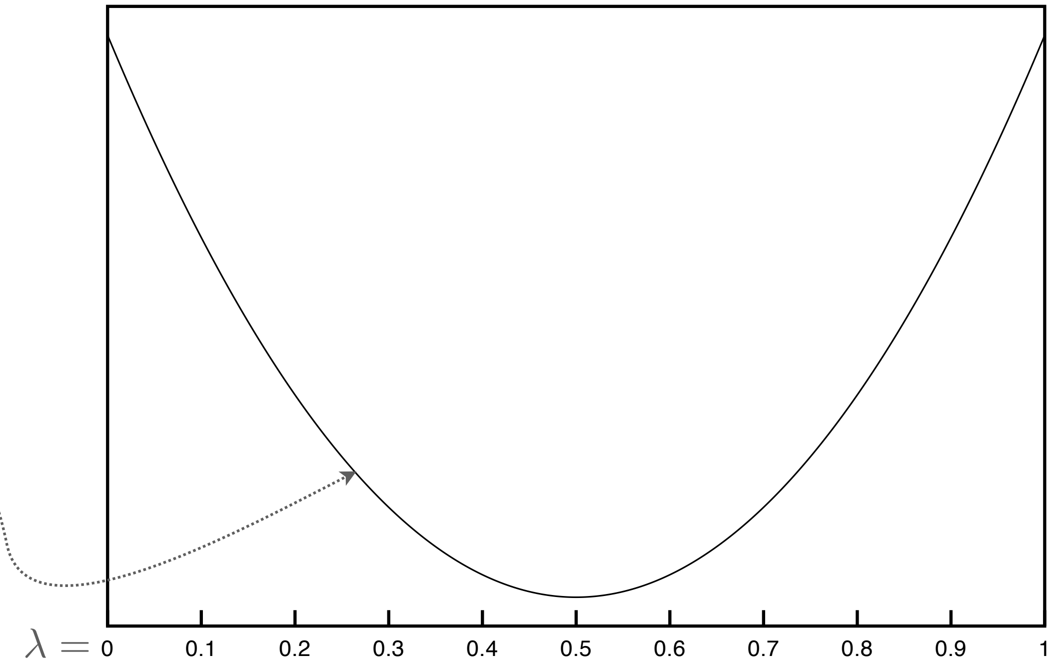
$P_0 = (\frac{1}{2}, \frac{1}{2})$ $P_1 = \left(\frac{1}{2}(1-\epsilon), \frac{1}{2}(1+\epsilon)\right)$

$$C(P_0, P_1) = -\inf_{0<\lambda<1} \log \frac{1}{2}\left((1-\epsilon)^\lambda + (1+\epsilon)^\lambda\right)$$

Minimum reached for $\lambda \approx \frac{1}{2}$

$$C(P_0, P_1) \approx -\log\left(1 - \frac{\epsilon^2}{8}\right) \approx \frac{\epsilon^2}{8 \ln 2}$$

Example with $\epsilon = 0.01$



Example: Biased Coin

$$\mathsf{P}_0 = \left(\tfrac{1}{2}, \tfrac{1}{2}\right) \qquad \mathsf{P}_1 = \left(\tfrac{1}{2}(1-\epsilon), \tfrac{1}{2}(1+\epsilon)\right)$$

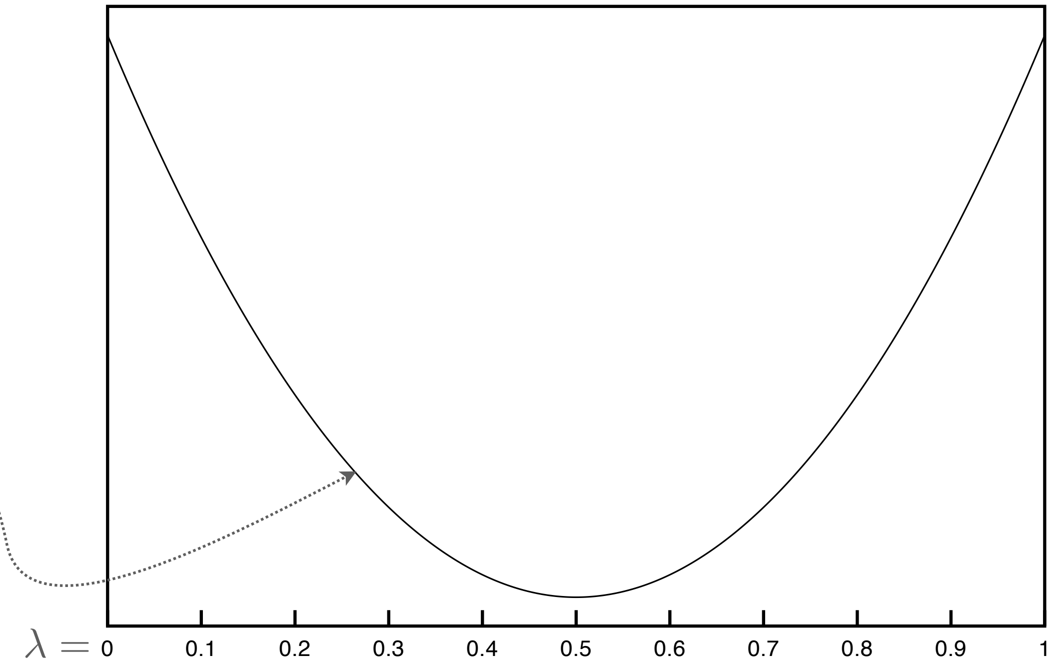
$$\mathrm{C}(\mathsf{P}_0, \mathsf{P}_1) = -\inf_{0<\lambda<1} \log \tfrac{1}{2}\left((1-\epsilon)^\lambda + (1+\epsilon)^\lambda\right)$$

Minimum reached for $\lambda \approx \frac{1}{2}$

$$\mathrm{C}(\mathsf{P}_0, \mathsf{P}_1) \approx -\log\left(1 - \frac{\epsilon^2}{8}\right) \approx \frac{\epsilon^2}{8 \ln 2}$$

$q \approx \dfrac{8 \ln 2}{\epsilon^2}$ allow to reach a non-negligible advantage.

Example with $\epsilon = 0.01$



Example: Biased Dice

$\mathsf{P}_0 = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$ $\mathsf{P}_1 = (\frac{1}{6}, \frac{1}{6}, \frac{2}{6}, 0, \frac{1}{6}, \frac{1}{6})$

Example: Biased Dice

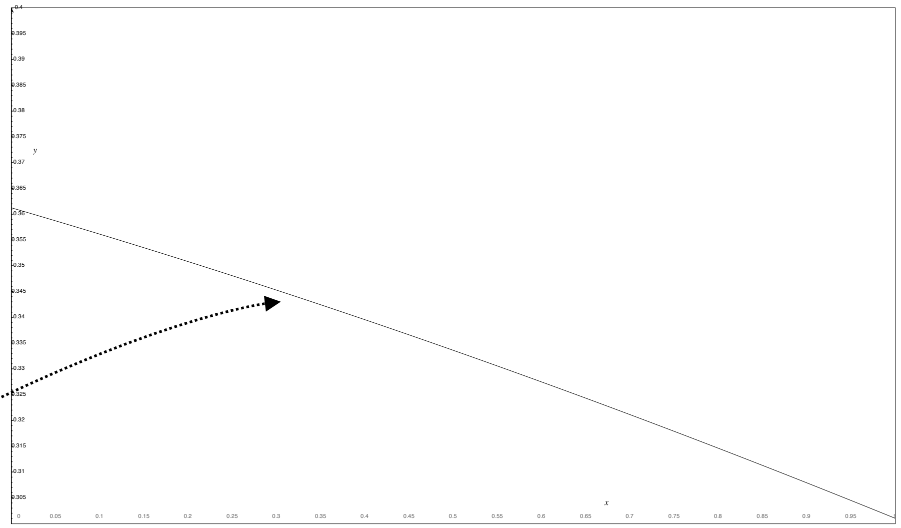
$\mathsf{P}_0 = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$ $\mathsf{P}_1 = (\frac{1}{6}, \frac{1}{6}, \frac{2}{6}, 0, \frac{1}{6}, \frac{1}{6})$

$$\mathrm{C}(\mathsf{P}_0, \mathsf{P}_1) = \max_{0<\lambda<1} \log\left(\frac{6}{2^\lambda + 4}\right)$$

Example: Biased Dice

$\mathsf{P}_0 = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$ $\mathsf{P}_1 = (\frac{1}{6}, \frac{1}{6}, \frac{2}{6}, 0, \frac{1}{6}, \frac{1}{6})$

$$\mathrm{C}(\mathsf{P}_0, \mathsf{P}_1) = \max_{0<\lambda<1} \log\left(\frac{6}{2^{\lambda}+4}\right)$$

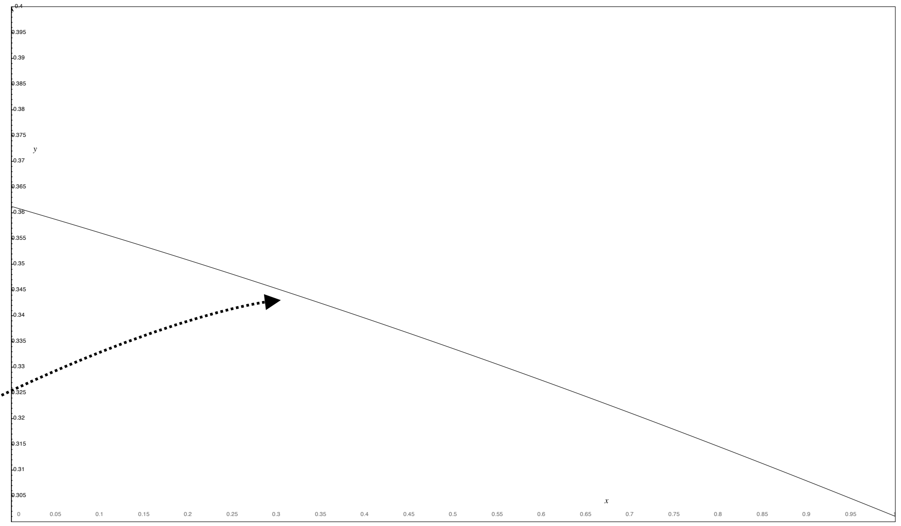


Example: Biased Dice

$\mathsf{P}_0 = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$ $\mathsf{P}_1 = (\frac{1}{6}, \frac{1}{6}, \frac{2}{6}, 0, \frac{1}{6}, \frac{1}{6})$

$$\mathrm{C}(\mathsf{P}_0, \mathsf{P}_1) = \max_{0<\lambda<1} \log\left(\frac{6}{2^\lambda + 4}\right)$$

$$\approx 0.263$$

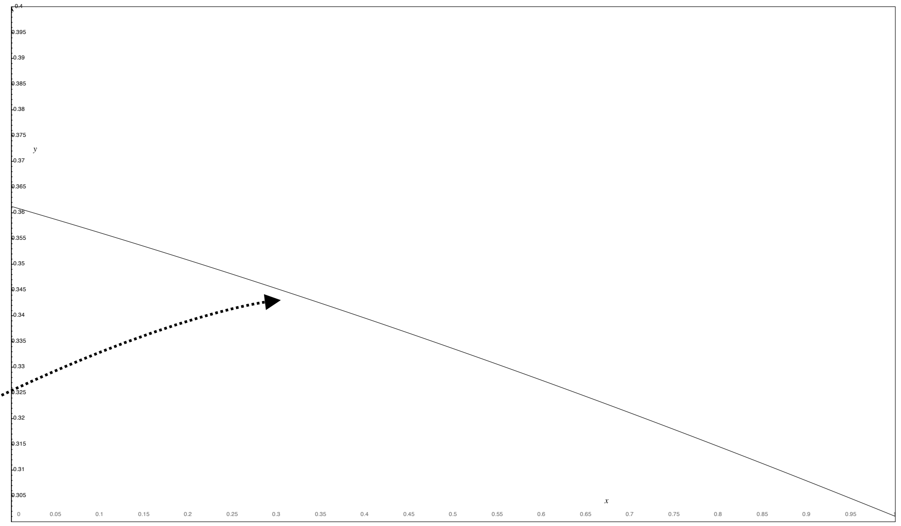


Example: Biased Dice

$P_0 = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$ $P_1 = (\frac{1}{6}, \frac{1}{6}, \frac{2}{6}, 0, \frac{1}{6}, \frac{1}{6})$

$$C(P_0, P_1) = \max_{0<\lambda<1} \log\left(\frac{6}{2^\lambda + 4}\right)$$

$$\approx 0.263$$



➕approx. $\frac{1}{0.263} \approx 3.8$ queries (rolls) are sufficient to distinguish one dice from the other.

➕This is the proof that all this theory has a practical application...

Possible Extensions

- Case where the distributions are "close" to each other
- Case where one of the hypotheses is composite
- Case where one of the two distributions is unknown
- etc.

Part I: On the (In)Security of Block Ciphers:
Tools for the Security Analysis

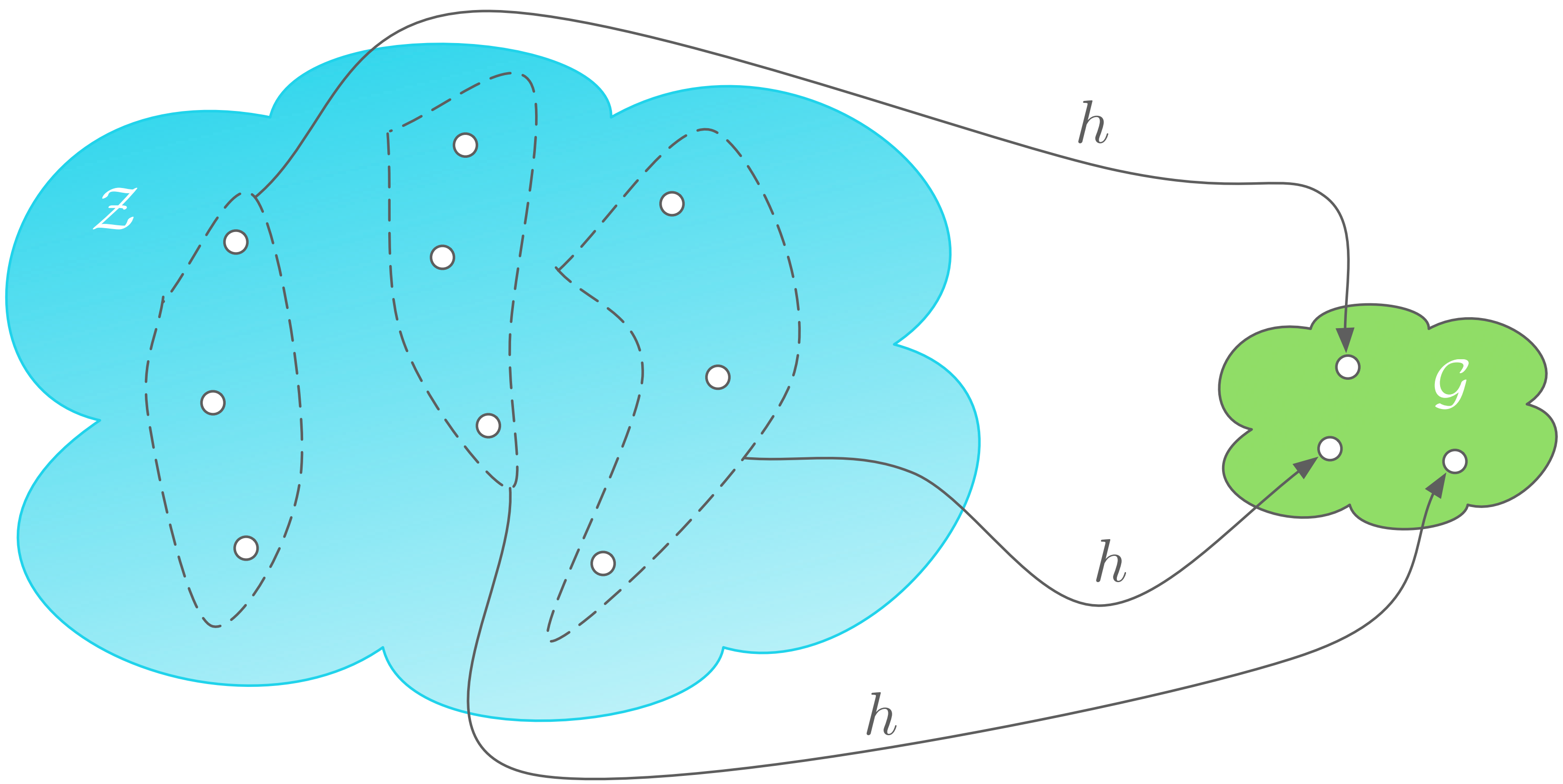
- Projection Based Distinguishers

On the Need for Projection-Based Distinguishers

- If $|\mathcal{Z}|$ is too large, the best distinguisher cannot be implemented.

On the Need for Projection-Based Distinguishers

- If $|\mathcal{Z}|$ is too large, the best distinguisher cannot be implemented.
- Possible solution: reduce the sample size using a projection:



✔ Distinguish in $\mathcal{G}$ instead of $\mathcal{Z}$.

🚫 This reduces the power of the distinguisher.

Example: Linear Distinguishers

- $\mathcal{Z} = \{0,1\}^n \qquad \mathcal{G} = \{0,1\} \qquad \mathsf{P}_0 = \mathsf{U} \qquad \mathsf{P}_1 = \mathsf{P} \qquad h(Z) = a \cdot Z = a_1 Z_1 \oplus \cdots \oplus a_n Z_n$
- This is a linear distinguisher based on the mask a.

Example: Linear Distinguishers

- $\mathcal{Z} = \{0,1\}^n \qquad \mathcal{G} = \{0,1\} \qquad \mathsf{P}_0 = \mathsf{U} \qquad \mathsf{P}_1 = \mathsf{P} \qquad h(Z) = a \cdot Z = a_1 Z_1 \oplus \cdots \oplus a_n Z_n$
- This is a linear distinguisher based on the mask a.
- By implementing the optimal strategy (after the linear compression), the advantage of this linear distinguisher verifies:

$$1 - \mathrm{Adv}(\mathsf{U}, \mathsf{P}) \doteq 2^{-q\mathrm{C}(\overline{\mathsf{U}}, \overline{\mathsf{P}})}$$

Example: Linear Distinguishers

- $\mathcal{Z} = \{0,1\}^n \qquad \mathcal{G} = \{0,1\} \qquad \mathsf{P}_0 = \mathsf{U} \qquad \mathsf{P}_1 = \mathsf{P} \qquad h(Z) = a \cdot Z = a_1 Z_1 \oplus \cdots \oplus a_n Z_n$
- This is a linear distinguisher based on the mask a.
- By implementing the optimal strategy (after the linear compression), the advantage of this linear distinguisher verifies:

$$1 - \mathrm{Adv}(\mathsf{U},\mathsf{P}) \doteq 2^{-q\mathrm{C}(\overline{\mathsf{U}},\overline{\mathsf{P}})}$$

$$a \cdot Z \sim \overline{\mathsf{P}} \Leftrightarrow Z \sim \mathsf{P}$$

$$a \cdot Z \sim \overline{\mathsf{U}} \Leftrightarrow Z \sim \mathsf{U}$$

Example: Linear Distinguishers

- $\mathcal{Z} = \{0,1\}^n \qquad \mathcal{G} = \{0,1\} \qquad \mathsf{P}_0 = \mathsf{U} \qquad \mathsf{P}_1 = \mathsf{P} \qquad h(Z) = a \cdot Z = a_1 Z_1 \oplus \cdots \oplus a_n Z_n$
- This is a linear distinguisher based on the mask a.
- By implementing the optimal strategy (after the linear compression), the advantage of this linear distinguisher verifies:

$$1 - \mathrm{Adv}(\mathsf{U}, \mathsf{P}) \doteq 2^{-q\mathrm{C}(\overline{\mathsf{U}},\overline{\mathsf{P}})}$$

$$a \cdot Z \sim \overline{\mathsf{P}} \Leftrightarrow Z \sim \mathsf{P}$$

$$a \cdot Z \sim \overline{\mathsf{U}} \Leftrightarrow Z \sim \mathsf{U}$$

- Definition: linear probability of P: $\mathrm{LP}_a(\mathsf{P}) = \left(\mathrm{E}_{\mathsf{P}}\left((-1)^{a \cdot Z}\right)\right)^2$

Example: Linear Distinguishers

- $\mathcal{Z} = \{0,1\}^n \qquad \mathcal{G} = \{0,1\} \qquad \mathsf{P}_0 = \mathsf{U} \qquad \mathsf{P}_1 = \mathsf{P} \qquad h(Z) = a \cdot Z = a_1 Z_1 \oplus \cdots \oplus a_n Z_n$
- This is a linear distinguisher based on the mask a.
- By implementing the optimal strategy (after the linear compression), the advantage of this linear distinguisher verifies:

$$1 - \mathrm{Adv}(\mathsf{U}, \mathsf{P}) \doteq 2^{-q\mathrm{C}(\overline{\mathsf{U}}, \overline{\mathsf{P}})}$$

$$a \cdot Z \sim \overline{\mathsf{P}} \Leftrightarrow Z \sim \mathsf{P}$$

$$a \cdot Z \sim \overline{\mathsf{U}} \Leftrightarrow Z \sim \mathsf{U}$$

- Definition: linear probability of P: $\mathrm{LP}_a(\mathsf{P}) = \left(\mathrm{E}_{\mathsf{P}}\left((-1)^{a \cdot Z}\right)\right)^2$
- Roughly: $\mathrm{C}(\overline{\mathsf{U}}, \overline{\mathsf{P}}) \approx \frac{\mathrm{LP}_a(\mathsf{P})}{8 \ln 2}$ ➡ $q \approx \frac{8 \ln 2}{\mathrm{LP}_a(\mathsf{P})}$ are enough (well known...)

Extending the Notion of Linear Probability

- The previous example only works for sets of the form $\mathcal{Z} = \{0,1\}^n$.
- We at least need to generalize the notion of linear probability to arbitrary sets.

Extending the Notion of Linear Probability

- The previous example only works for sets of the form $\mathcal{Z} = \{0,1\}^n$.
- We at least need to generalize the notion of linear probability to arbitrary sets.

Definition

The linear probability of P over the group $\mathcal{Z}$ with respect to the character χ is

$$\mathrm{LP}_\chi(\mathsf{P}) = \left|\mathrm{E}_{\mathsf{P}}\left(\chi(Z)\right)\right|^2$$

Extending the Notion of Linear Probability

- The previous example only works for sets of the form $\mathcal{Z} = \{0,1\}^n$.
- We at least need to generalize the notion of linear probability to arbitrary sets.

Definition

The linear probability of P over the group $\mathcal{Z}$ with respect to the character χ is

$$\mathrm{LP}_\chi(\mathsf{P}) = \left|\mathrm{E}_{\mathsf{P}}\left(\chi(Z)\right)\right|^2$$

- A character of $\mathcal{Z}$ is a homomorphism $\chi : \mathcal{Z} \longrightarrow \mathbf{C}^\times$
- Example: when $\mathcal{Z} = \{0,1\}^n$ we have $\chi(a) = (-1)^{u \cdot a}$ for some u

Extending the Notion of Linear Probability

- The previous example only works for sets of the form $\mathcal{Z} = \{0,1\}^n$.
- We at least need to generalize the notion of linear probability to arbitrary sets.

Definition

The linear probability of P over the group $\mathcal{Z}$ with respect to the character χ is

$$\mathrm{LP}_\chi(\mathsf{P}) = \left|\mathrm{E}_{\mathsf{P}}\left(\chi(Z)\right)\right|^2$$

- A character of $\mathcal{Z}$ is a homomorphism $\chi : \mathcal{Z} \longrightarrow \mathbf{C}^\times$
- Example: when $\mathcal{Z} = \{0,1\}^n$ we have $\chi(a) = (-1)^{u\cdot a}$ for some u
- Consequence: when $\mathcal{Z} = \{0,1\}^n$ this new definition corresponds to the old one!

Lin. Dist. for Sources overs Arbitrary Sets

We have wonderful lemma...

Lin. Dist. for Sources overs Arbitrary Sets

We have wonderful lemma...

Lemma 7.5 *Let* P_0 *be the uniform distribution on a finite subgroup* H *of* $\mathbf{C}^\times$ *of order* d. *Let* $\mathcal{D} = \{\mathsf{P}_u : u \in \mathsf{H}\}$ *be a set of* d *distributions on* H *defined by* (7.10). *The* q*-limited distinguisher between the null hypothesis* $\mathsf{H}_0 : \mathsf{P} = \mathsf{P}_0$ *and the alternate hypothesis* $\mathsf{H}_1 : \mathsf{P} \in \mathcal{D}$ *defined by the distribution acceptance region* $\Pi_q^\star = \Pi^\star \cap \mathcal{P}_q$, *where*

$$\Pi^\star = \left\{ \mathsf{P} \in \mathcal{P} \ : \ \|\mathsf{P}\|_\infty \geq \frac{\log(1-\epsilon)}{\log(1-\epsilon) - \log(1+(d-1)\epsilon)} \right\}, \tag{7.11}$$

is asymptotically optimal and its advantage $\mathrm{BestAdv}_q$ *is such that*

$$1 - \mathrm{BestAdv}_q(\mathsf{H}_0, \mathsf{H}_1) \doteq 2^{q \inf_{0<\lambda<1} \log \frac{1}{d}\left((1+(d-1)\epsilon)^\lambda + (d-1)(1-\epsilon)^\lambda\right)}.$$

Lin. Dist. for Sources overs Arbitrary Sets

We have wonderful lemma...

Lemma 7.5 *Let* P_0 *be the uniform distribution on a finite subgroup* H *of* $\mathbf{C}^\times$ *of order* d. *Let* $\mathcal{D} = \{\mathsf{P}_u : u \in \mathsf{H}\}$ *be a set of* d *distributions on* H *defined by* (7.10). *The* q*-limited distinguisher between the null hypothesis* $\mathsf{H}_0 : \mathsf{P} = \mathsf{P}_0$ *and the alternate hypothesis* $\mathsf{H}_1 : \mathsf{P} \in \mathcal{D}$ *defined by the distribution acceptance region* $\Pi_q^\star = \Pi^\star \cap \mathcal{P}_q$, *where*

$$\Pi^\star = \left\{ \mathsf{P} \in \mathcal{P} : \|\mathsf{P}\|_\infty \geq \frac{\log(1-\epsilon)}{\log(1-\epsilon) - \log(1+(d-1)\epsilon)} \right\} \qquad (7.11)$$

is asymptotically optimal and its advantage $\mathrm{BestAdv}_q$ *is such that*

$$1 - \mathrm{BestAdv}_q(\mathsf{H}_0, \mathsf{H}_1) \doteq 2^{q \inf_{0<\lambda<1} \log \frac{1}{d}\left((1+(d-1)\epsilon)^\lambda + (d-1)(1-\epsilon)^\lambda\right)}.$$

Which shows how to use the generalized LP to build a linear distinguisher over arbitrary sets...

and allows to conclude that a linear distinguisher needs $q \approx \frac{8 \ln 2}{(d-1)\mathrm{LP}^\chi(\mathsf{P})}$ to reach a good advantage.

Part I: On the (In)Security of Block Ciphers:

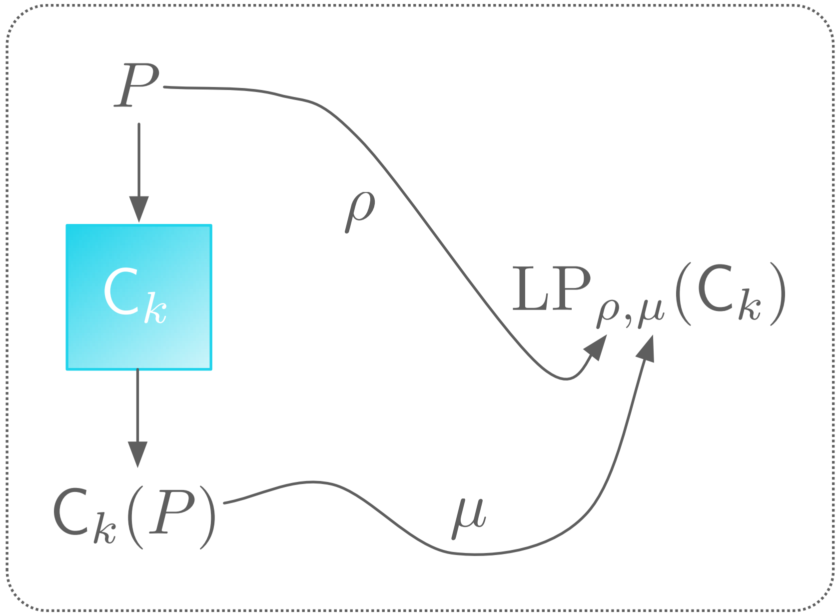
Tools for the Security Analysis

Practical Implications for Block Ciphers

Distinguishing Random Permutations

- A simple trick allows to turn distinguishers of random sources into distinguishers of random permutations (block ciphers).
- All the results on random sources apply to random permutations.
- In the case of the generalization of linear cryptanalysis:

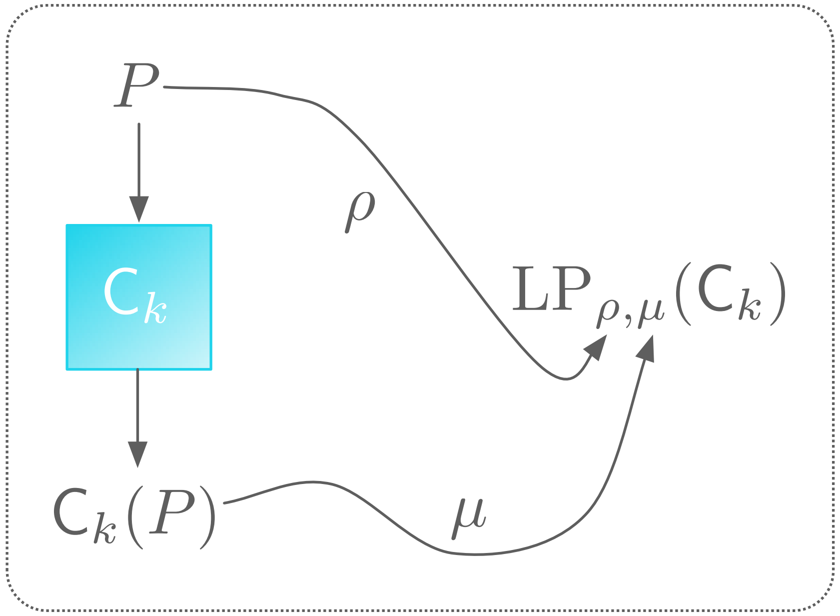
$$\mathrm{LP}_{\rho,\mu}(\mathsf{C}_k) = \left|\mathrm{E}_{P\in_{\mathrm{U}}\mathcal{T}}\left(\overline{\rho}(P)\mu\left(\mathsf{C}_k(P)\right)\right)\right|^2$$



Distinguishing Random Permutations

- A simple trick allows to turn distinguishers of random sources into distinguishers of random permutations (block ciphers).
- All the results on random sources apply to random permutations.
- In the case of the generalization of linear cryptanalysis:

$$\mathrm{LP}_{\rho,\mu}(\mathsf{C}_k) = \left|\mathrm{E}_{P\in_{\mathrm{U}}\mathcal{T}}\left(\overline{\rho}(P)\mu\left(\mathsf{C}_k(P)\right)\right)\right|^2$$

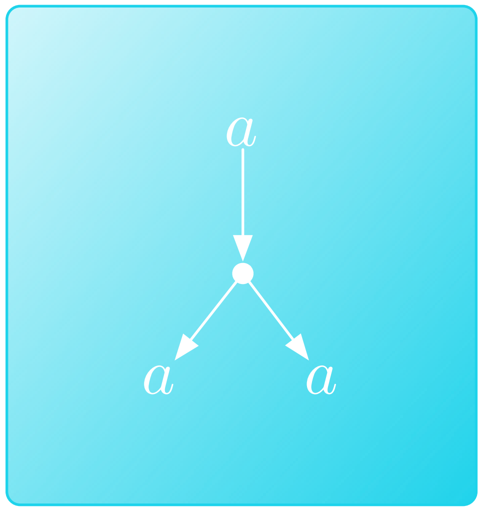


- $\mathrm{ELP}_{\rho,\mu}(\mathsf{C}) = \mathrm{E}_K\left(\mathrm{LP}_{\rho,\mu}(\mathsf{C}_K)\right)$
- $q \approx 8\ln 2/\mathrm{ELP}_{\rho,\mu}(\mathsf{C})$ → find ρ and μ which maximize the ELP

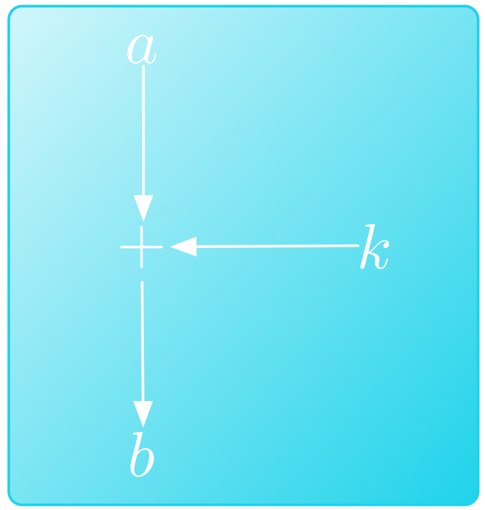
How to find the best input/output characters?

- Apply a bottom-up approach

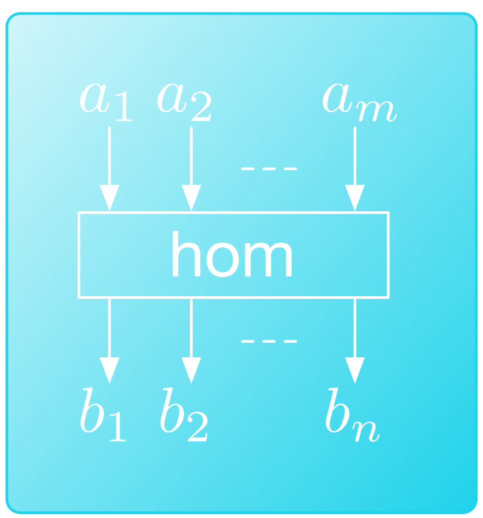
How to find the best input/output characters?



$\mathrm{LP}_{\chi_1\chi_2,\chi_1\|\chi_2} = 1$

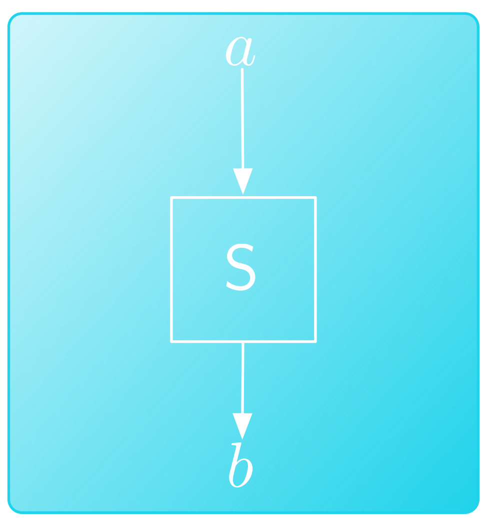


$\mathrm{LP}_{\chi,\chi} = 1$



With $\chi = \chi_1\| \cdots \|\chi_n$

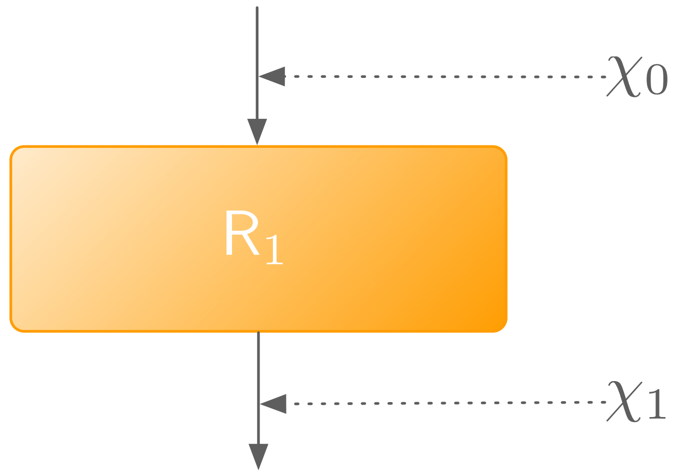
$\mathrm{LP}_{\chi\circ\mathsf{hom},\chi} = 1$



$\mathrm{LP}_{\chi,\rho}$ “by hand”

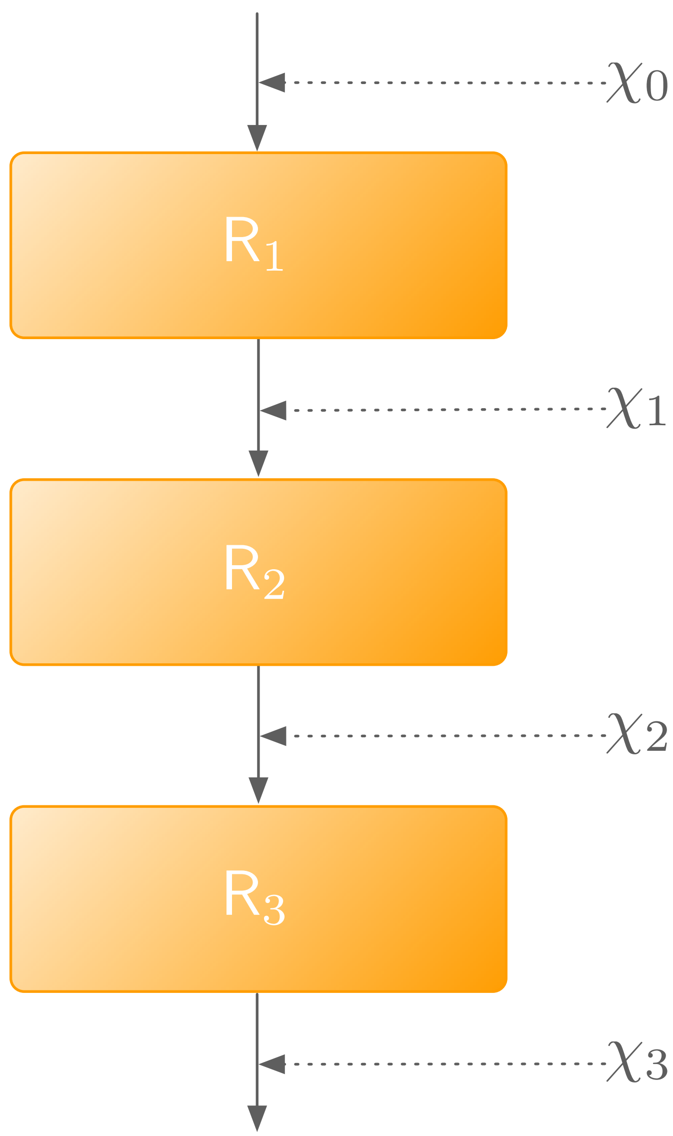
- Apply a bottom-up approach
- We provide a toolbox that allows, for any given output character, to find the input characters that maximizes the ELP over various building blocks.

How to find the best input/output characters?



- Apply a bottom-up approach
- We provide a toolbox that allows, for any given output character, to find the input characters that maximizes the ELP over various building blocks.
- Easy to deduce a ELP over one round

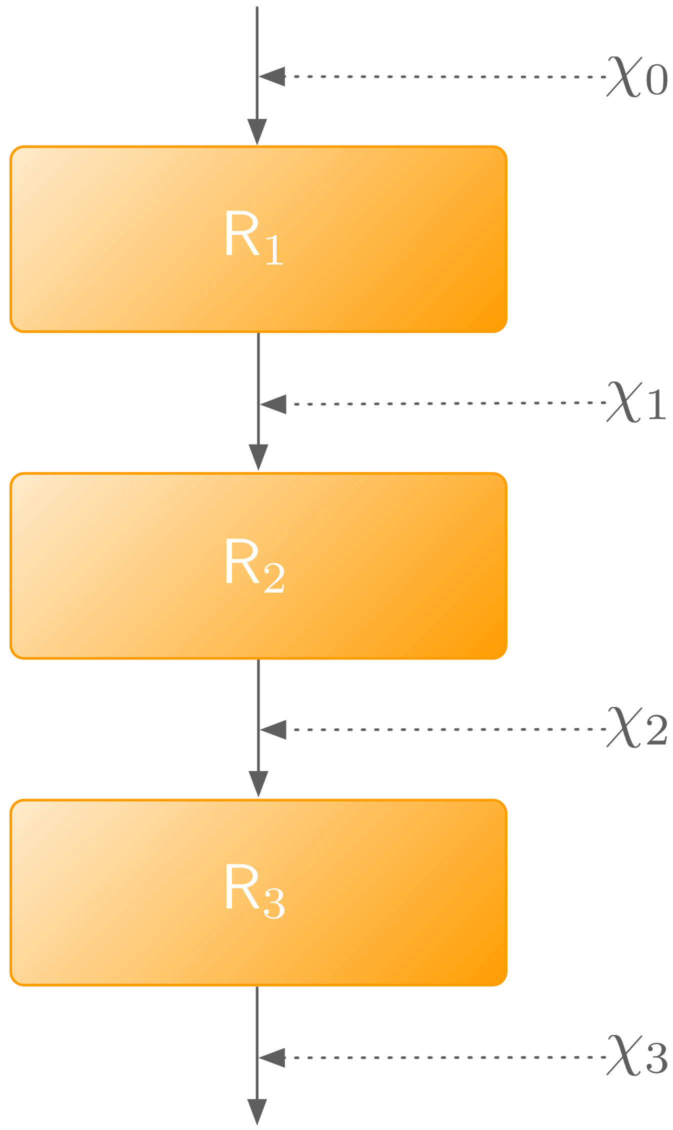
How to find the best input/output characters?



- Apply a bottom-up approach
- We provide a toolbox that allows, for any given output character, to find the input characters that maximizes the ELP over various building blocks.
- Easy to deduce a ELP over one round
- For a Markov cipher $\mathsf{C} = \mathsf{R}_3 \circ \mathsf{R}_2 \circ \mathsf{R}_1$, we show that Nyberg's linear hull effect applies:

$$\mathrm{ELP}_{\chi_0,\chi_3}(\mathsf{C}) = \sum_{\chi_1,\chi_2} \prod_{i=1}^{3} \mathrm{ELP}_{\chi_{i-1},\chi_i}(\mathsf{R}_i)$$

How to find the best input/output characters?



- Apply a bottom-up approach
- We provide a toolbox that allows, for any given output character, to find the input characters that maximizes the ELP over various building blocks.
- Easy to deduce a ELP over one round
- For a Markov cipher $\mathsf{C} = \mathsf{R}_3 \circ \mathsf{R}_2 \circ \mathsf{R}_1$, we show that Nyberg's linear hull effect applies:

$$\mathrm{ELP}_{\chi_0,\chi_3}(\mathsf{C}) = \sum_{\chi_1,\chi_2} \prod_{i=1}^{3} \mathrm{ELP}_{\chi_{i-1},\chi_i}(\mathsf{R}_i)$$

- Use the last property to pile ELP's up:

$$\mathrm{ELP}_{\chi_0,\chi_3}(\mathsf{C}) \geq \prod_{i=1}^{3} \mathrm{ELP}_{\chi_{i-1},\chi_i}(\mathsf{R}_i)$$

Applications on SAFER K/SK

- We attack SAFER with a $\boxplus$-linear cryptanalysis.
- Use the toolbox to find characteristics within SAFER K/SK.
- To compute the complexities we consider several characteristics among the hull (i.e., all characteristics share the same input/output characters).
- To turn distinguishing attacks into key recovery attacks, we also take advantage of the linearity of the key schedule.

Applications on SAFER K/SK

- We attack SAFER with a $\boxplus$-linear cryptanalysis.
- Use the toolbox to find characteristics within SAFER K/SK.
- To compute the complexities we consider several characteristics among the hull (i.e., all characteristics share the same input/output characters).
- To turn distinguishing attacks into key recovery attacks, we also take advantage of the linearity of the key schedule.

Nbr Rounds	Complexity
2	$2^{23}/2^{31}$
3	2^{38}
4	2^{49}
5	2^{56}

Other Applications

- Two new Digital Encryption Algorithm for Numbers (based on the AES): DEAN18 and DEAN27 which respectively encrypts blocks made of 18 and 27 decimal digits.
- Resistance against our generalization of linear cryptanalysis.
- New attacks on TOY100 (toy cipher that encrypts blocks of 32 decimal digits).
- Break 9 (10 ?) rounds out of 12.

Part II: Designs and Security Proofs

Outline

Block Ciphers

Dial $\mathbf{C}$ for Cipher

KFC: the Krazy Feistel Cipher

Outline

- Block Ciphers
- Dial $\mathbb{C}$ for Cipher
- KFC: the Krazy Feistel Cipher

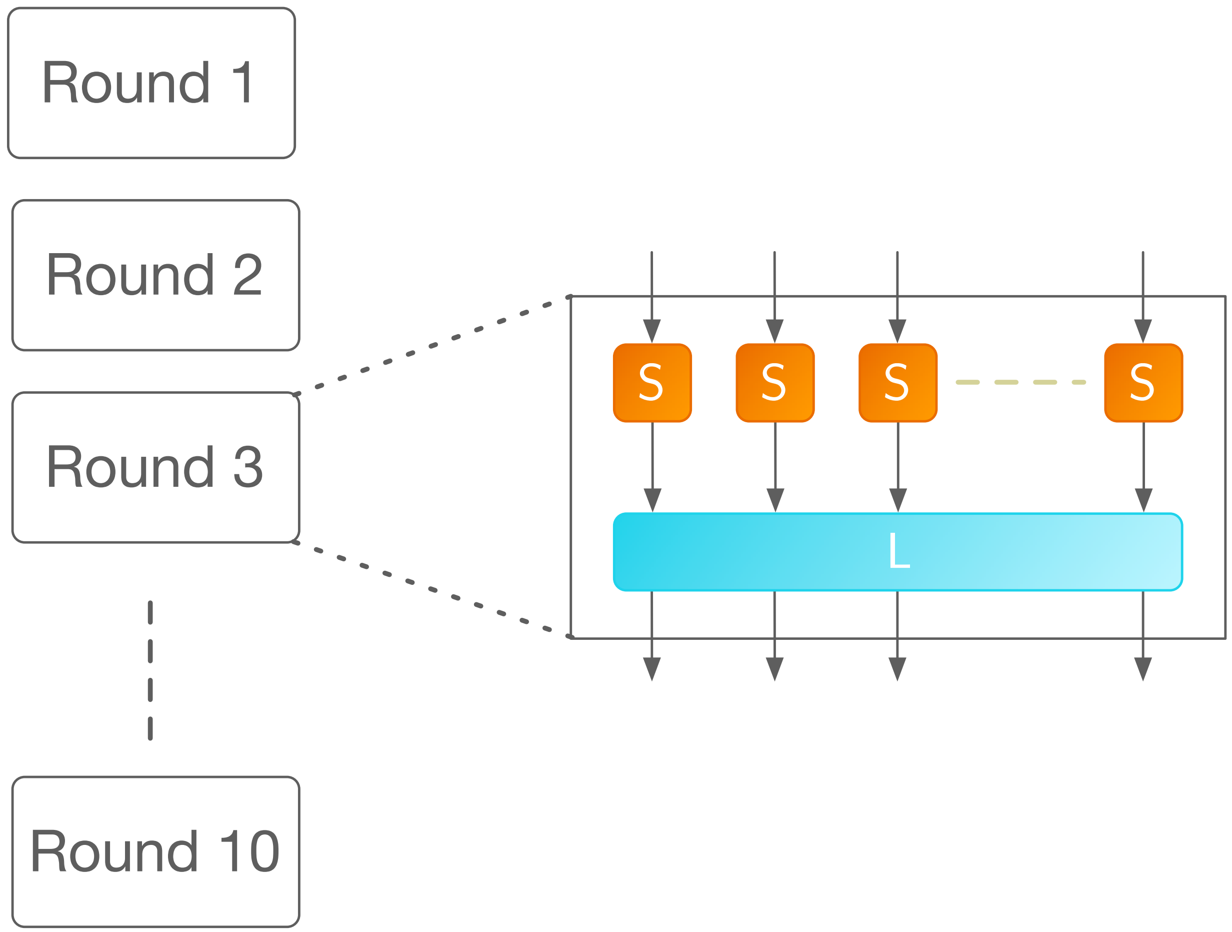
- The Luby-Rackoff Model
- Vaudenay's decorrelation theory

Outline

Block Ciphers

Dial C for Cipher

KFC: the Krazy Feistel Cipher

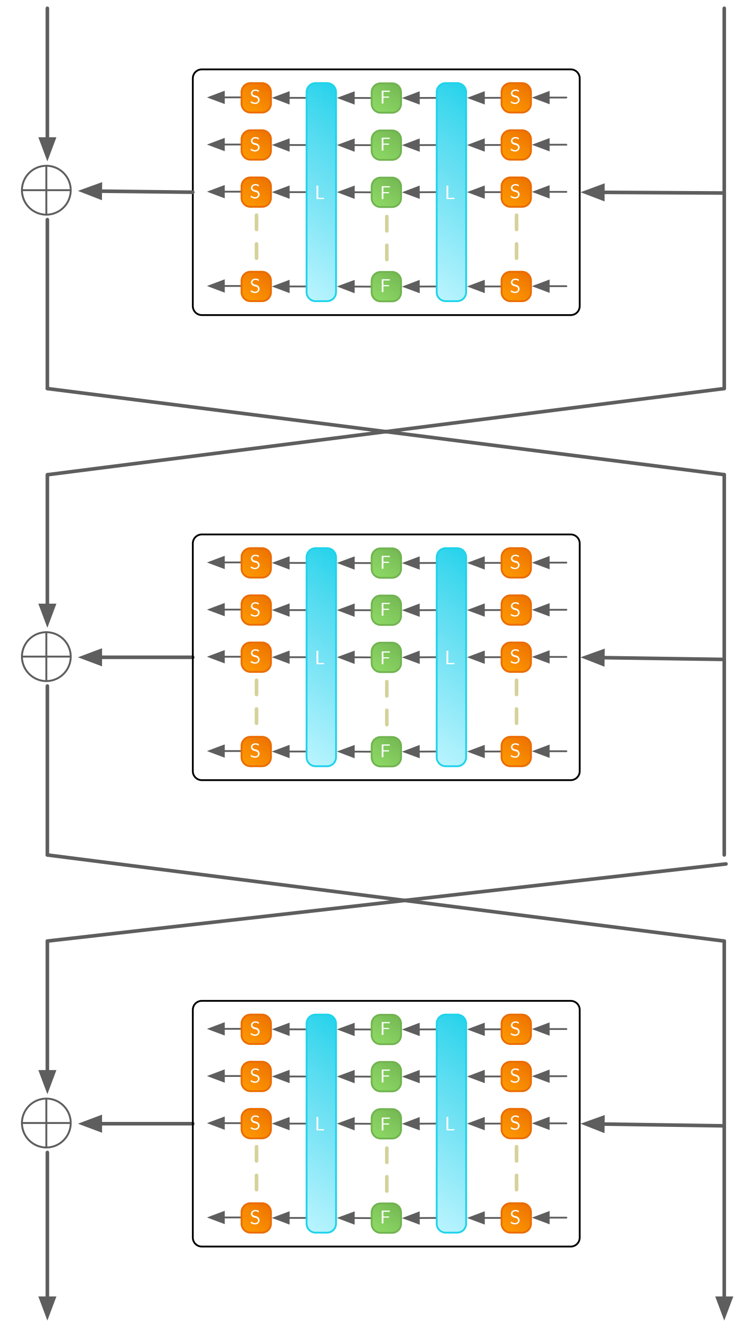


Outline

Block Ciphers

Dial C for Cipher

➡ KFC: the Krazy Feistel Cipher



Outline

Block Ciphers

Dial $\mathbb{C}$ for Cipher

➡ KFC: the Krazy Feistel Cipher

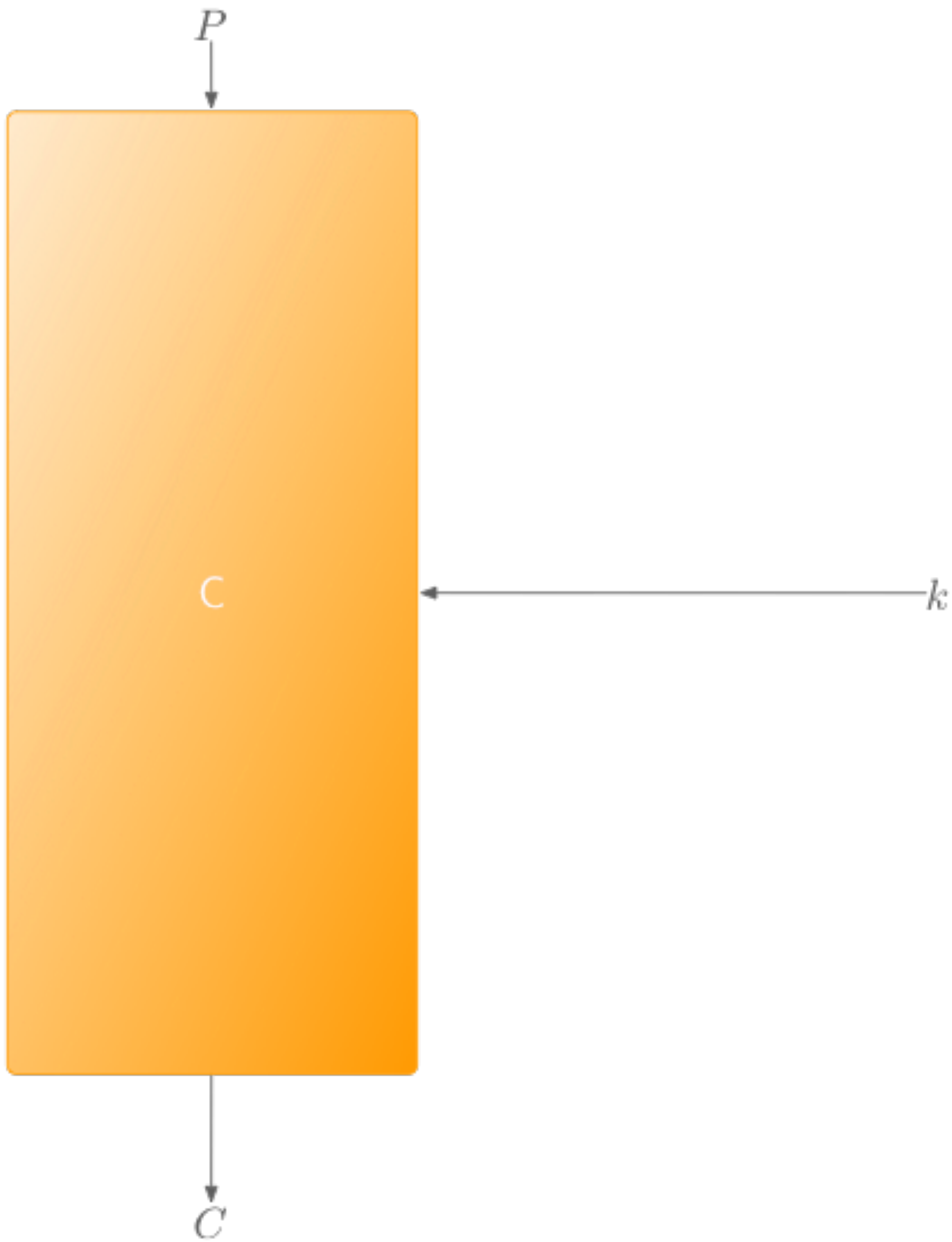
[BVsac05] [BFsac06] [BFa06]

Part II: Designs and Security Proofs

Block Ciphers

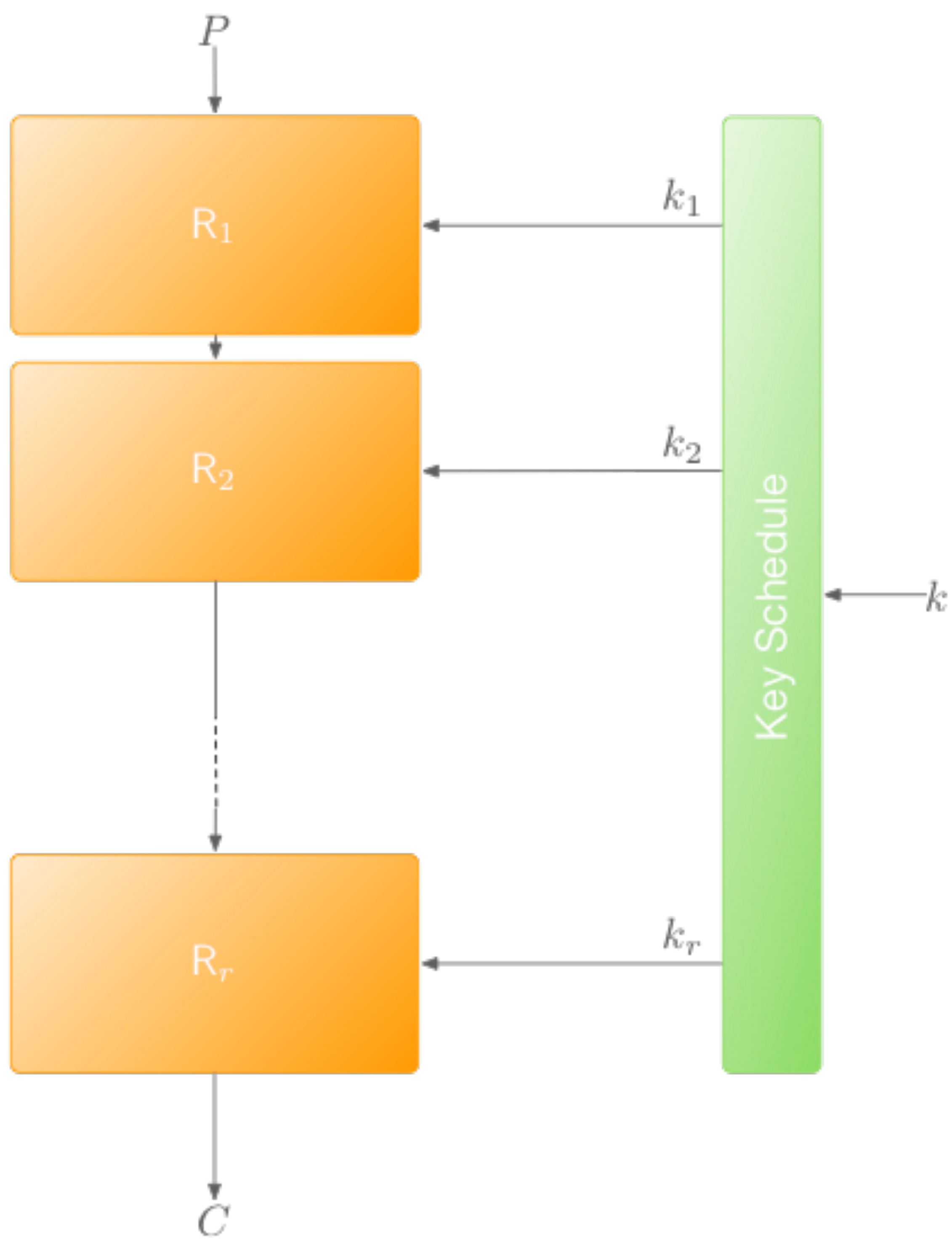
A Typical Iterated Block Cipher

- A block cipher on a finite set is a family of permutations on that set, indexed by a parameter call the key.



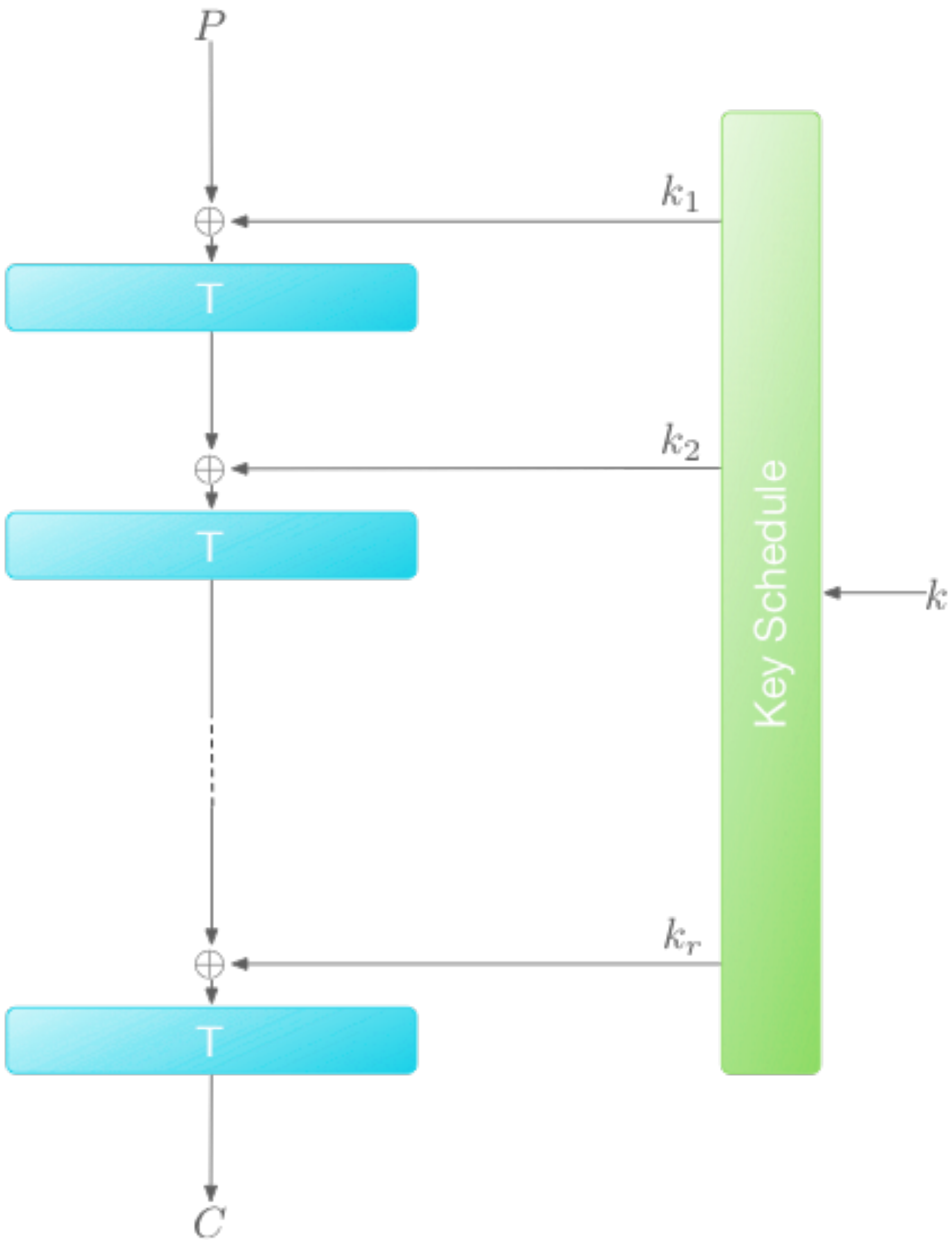
A Typical Iterated Block Cipher

- A block cipher on a finite set is a family of permutations on that set, indexed by a parameter call the key.
- Such a cipher is usually iterated, i.e., made of several rounds.
- Each round is parameterized by a key derived from the main secret key by means of a Key Schedule.



A Typical Iterated Block Cipher

- A block cipher on a finite set is a family of permutations on that set, indexed by a parameter call the key.
- Such a cipher is usually iterated, i.e., made of several rounds.
- Each round is parameterized by a key derived from the main secret key by means of a Key Schedule.
- Usually, the rounds all share the same design, e.g., a round key addition followed by a fixed (nonlinear) transformation.

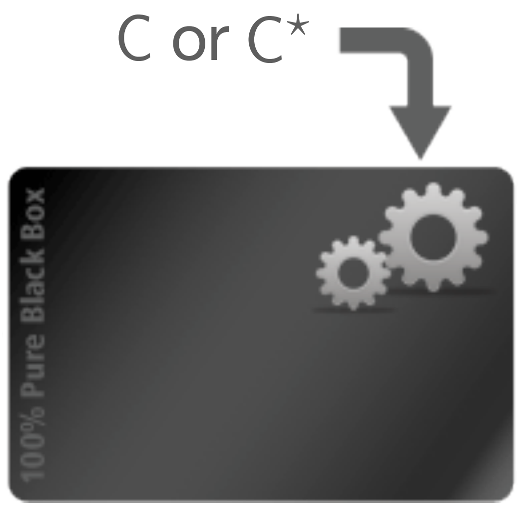


What Should we Expect from a Block Cipher?

It should be fast and secure!

What Should we Expect from a Block Cipher?

It should be fast and secure!

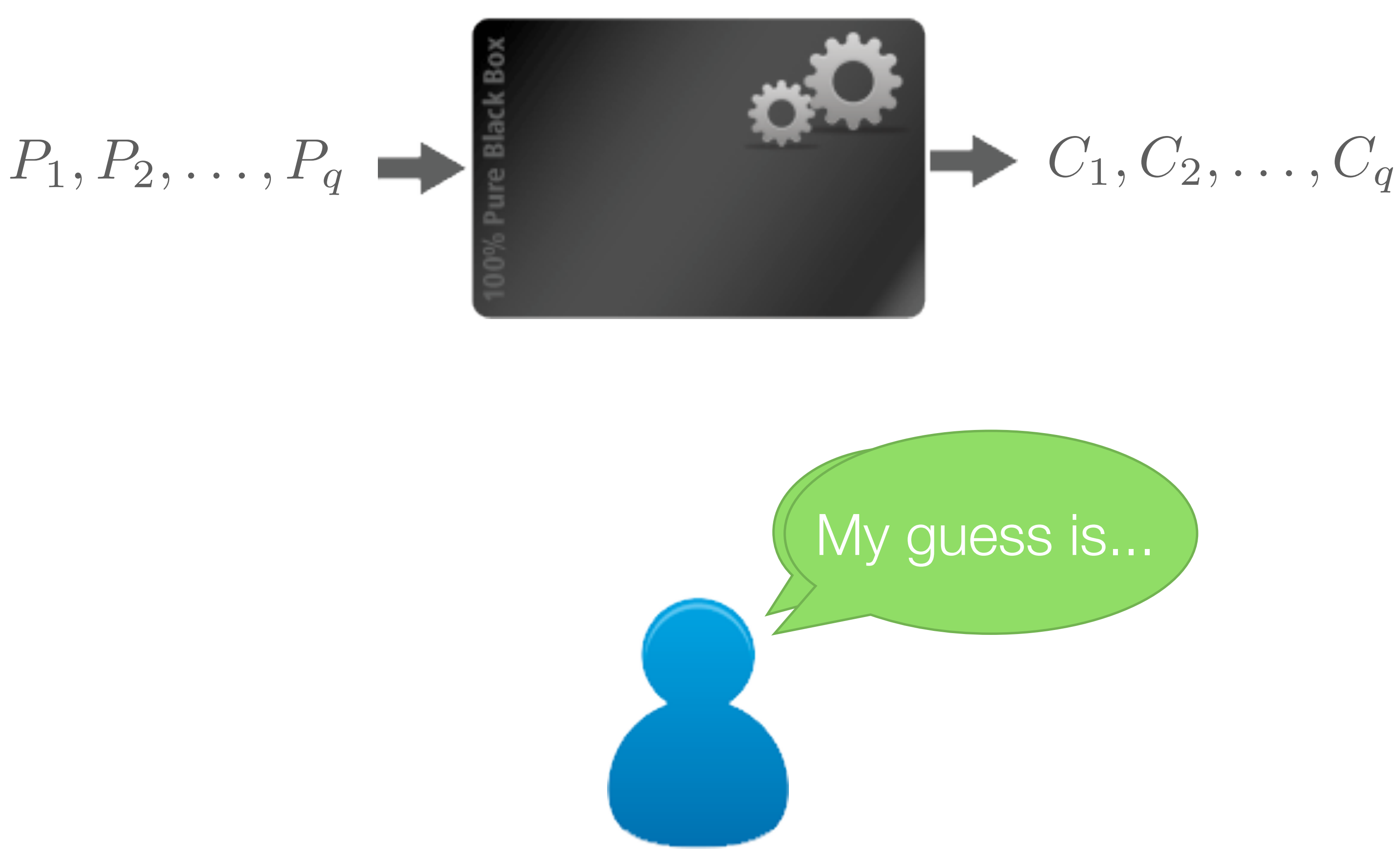


What Should we Expect from a Block Cipher?

It should be fast and secure!

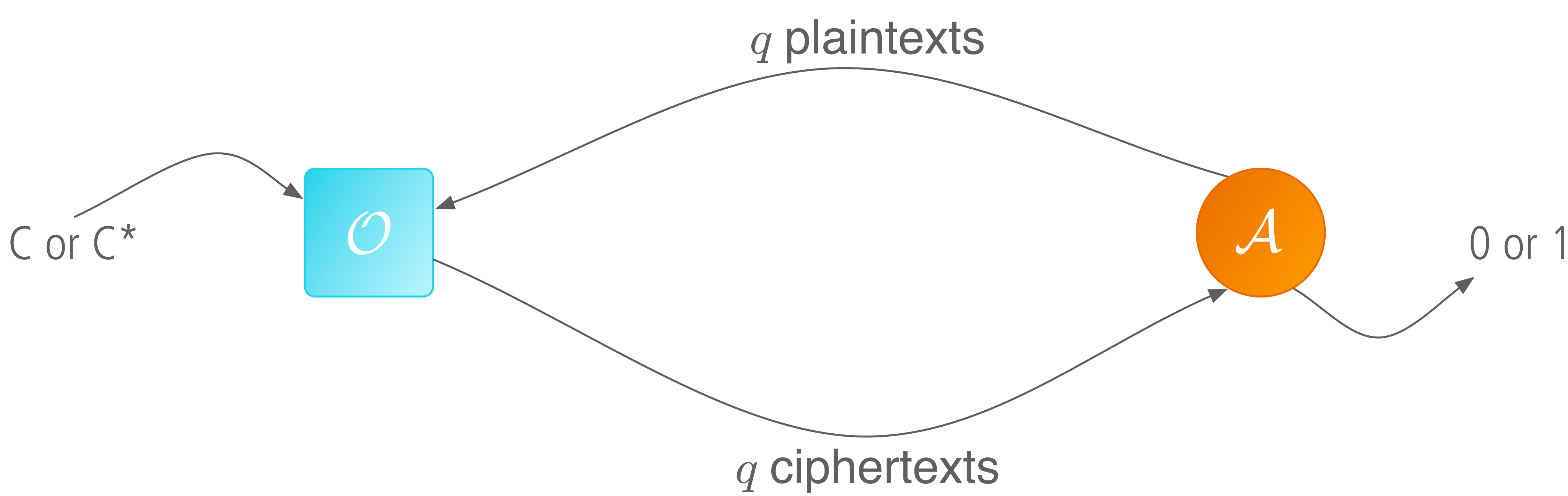
What Should we Expect from a Block Cipher?

It should be fast and secure!



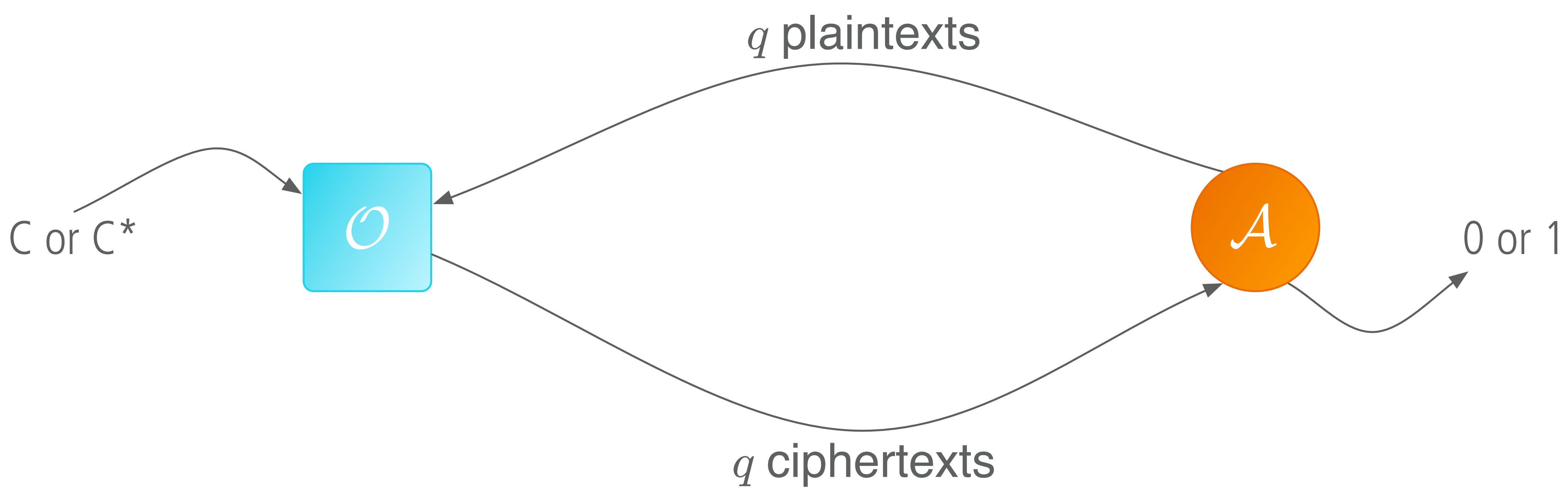
The Luby-Rackoff Model

We consider a q-limited adversary $\mathcal{A}$ in the Luby-Rackoff Model:



The Luby-Rackoff Model

We consider a q-limited adversary $\mathcal{A}$ in the Luby-Rackoff Model:



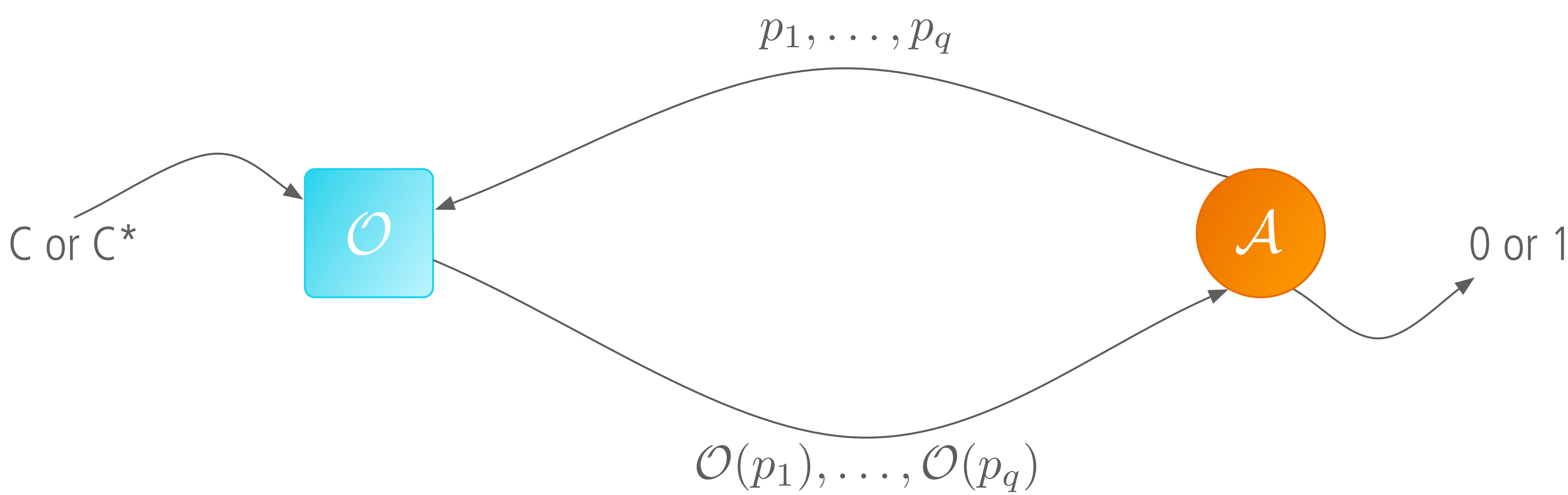
$$\mathrm{Adv}_{\mathcal{A}}(\mathsf{C}, \mathsf{C}^\star) = |\Pr[\mathcal{A}(\mathsf{C}) = 1] - \Pr[\mathcal{A}(\mathsf{C}^\star) = 1]|$$

Advantage of the q-limited adversary $\mathcal{A}$ between C and $\mathsf{C}^\star$

✔ The block cipher C is secure if the advantage of $\mathcal{A}$ is negligible for all $\mathcal{A}$'s.

The Luby-Rackoff Model

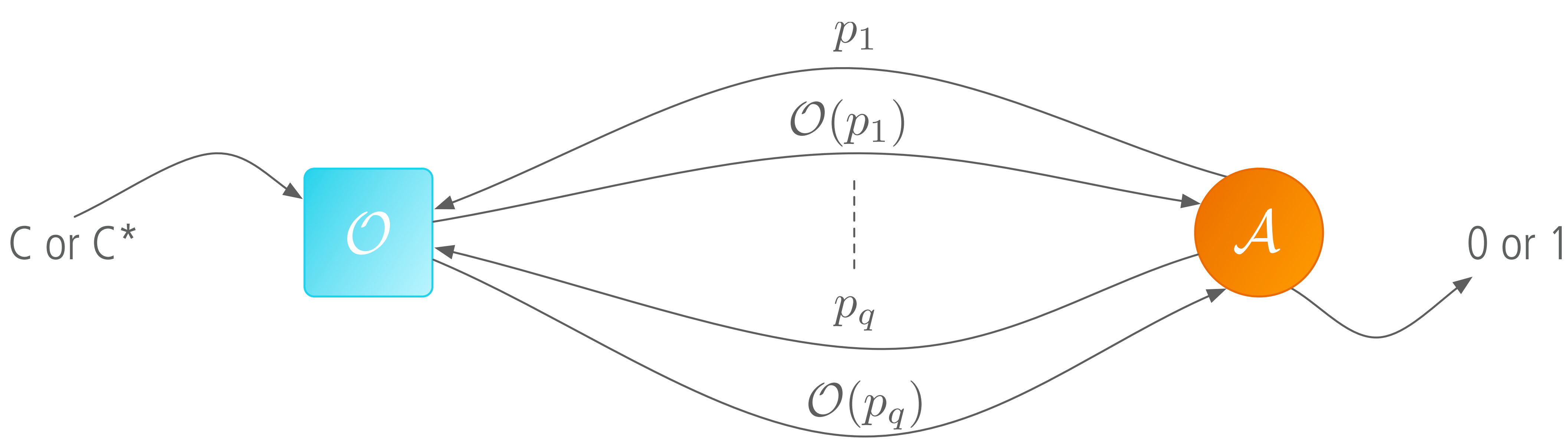
We consider a q-limited adversary $\mathcal{A}$ in the Luby-Rackoff Model:



$\mathcal{A}$ is non-adaptive if the q plaintexts are chosen “at once”.

The Luby-Rackoff Model

We consider a q-limited adversary $\mathcal{A}$ in the Luby-Rackoff Model:



$\mathcal{A}$ is adaptive if plaintext i depends on ciphertexts $1, \ldots, i-1$.

Computing $\mathrm{Adv}_{\mathcal{A}}(\mathsf{C}, \mathsf{C}^\star)$

- Computing the advantage is not a trivial task in general.

- Possible solution: use Vaudenay's Decorrelation Theory.

$$\max_{\mathcal{A}} \mathrm{Adv}_{\mathcal{A}}(\mathsf{C}, \mathsf{C}^\star) = \frac{1}{2} \| [\mathsf{C}]^q - [\mathsf{C}^\star]^q \|$$

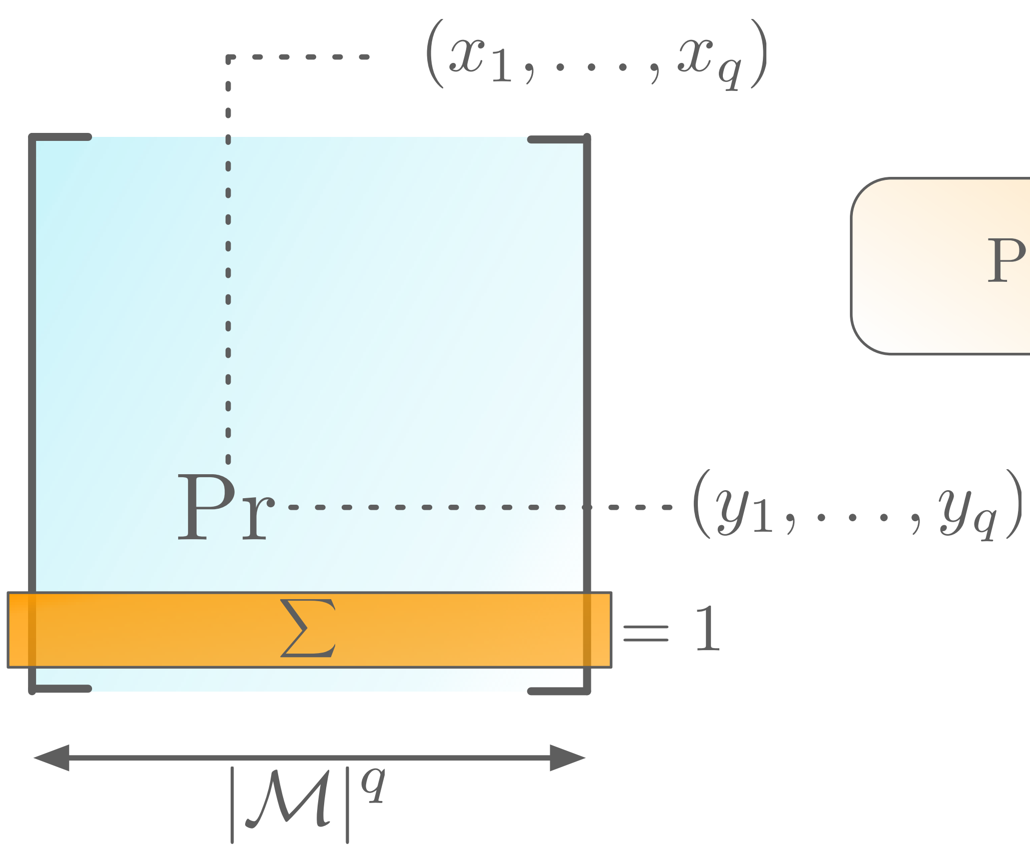
Computing $\mathrm{Adv}_{\mathcal{A}}(\mathsf{C}, \mathsf{C}^\star)$

- Computing the advantage is not a trivial task in general.
- Possible solution: use Vaudenay's Decorrelation Theory.

$$\max_{\mathcal{A}} \mathrm{Adv}_{\mathcal{A}}(\mathsf{C}, \mathsf{C}^\star) = \frac{1}{2} \| [\mathsf{C}]^q - [\mathsf{C}^\star]^q \|$$

$$\Pr = \Pr_{\mathsf{C}}[\mathsf{C}(x_1) = y_1, \ldots, \mathsf{C}(x_q) = y_q]$$

$[\mathsf{C}]^q =$



Example!

On the set $\mathcal{M}$={1,2,3}, the distribution matrices of the perfect cipher C* look like this (at orders 1 and 2):

$$
[\mathsf{C}^\star]^1 = \begin{array}{c} \begin{array}{ccc} (1) & (2) & (3) \end{array} \\ \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix} \end{array} \begin{array}{c} \\ (1) \\ (2) \\ (3) \end{array}
$$

$$
[\mathsf{C}^\star]^2 = \begin{array}{c} \begin{array}{ccccccccc} (1,1) & (1,2) & (1,3) & (2,1) & (2,2) & (2,3) & (3,1) & (3,2) & (3,3) \end{array} \\ \begin{bmatrix} 1/3 & 0 & 0 & 0 & 1/3 & 0 & 0 & 0 & 1/3 \\ 0 & 1/6 & 1/6 & 1/6 & 0 & 1/6 & 1/6 & 1/6 & 0 \\ 0 & 1/6 & 1/6 & 1/6 & 0 & 1/6 & 1/6 & 1/6 & 0 \\ 0 & 1/6 & 1/6 & 1/6 & 0 & 1/6 & 1/6 & 1/6 & 0 \\ 1/3 & 0 & 0 & 0 & 1/3 & 0 & 0 & 0 & 1/3 \\ 0 & 1/6 & 1/6 & 1/6 & 0 & 1/6 & 1/6 & 1/6 & 0 \\ 0 & 1/6 & 1/6 & 1/6 & 0 & 1/6 & 1/6 & 1/6 & 0 \\ 0 & 1/6 & 1/6 & 1/6 & 0 & 1/6 & 1/6 & 1/6 & 0 \\ 1/3 & 0 & 0 & 0 & 1/3 & 0 & 0 & 0 & 1/3 \end{bmatrix} \end{array} \begin{array}{c} \\ (1,1) \\ (1,2) \\ (1,3) \\ (2,1) \\ (2,2) \\ (2,3) \\ (3,1) \\ (3,2) \\ (3,3) \end{array}
$$

Adaptive vs. non-Adaptive Adversaries

- The norm used to compute the distance between two distribution matrices depends on the kind of adversary we consider.

- If $\mathcal{A}$ is adaptive:

$$\max_{\mathcal{A}_{\mathrm{a}}} \mathrm{Adv}_{\mathcal{A}_{\mathrm{a}}}(\mathsf{C}, \mathsf{C}^\star) = \frac{1}{2} \| [\mathsf{C}]^q - [\mathsf{C}^\star]^q \|_{\mathsf{a}}$$

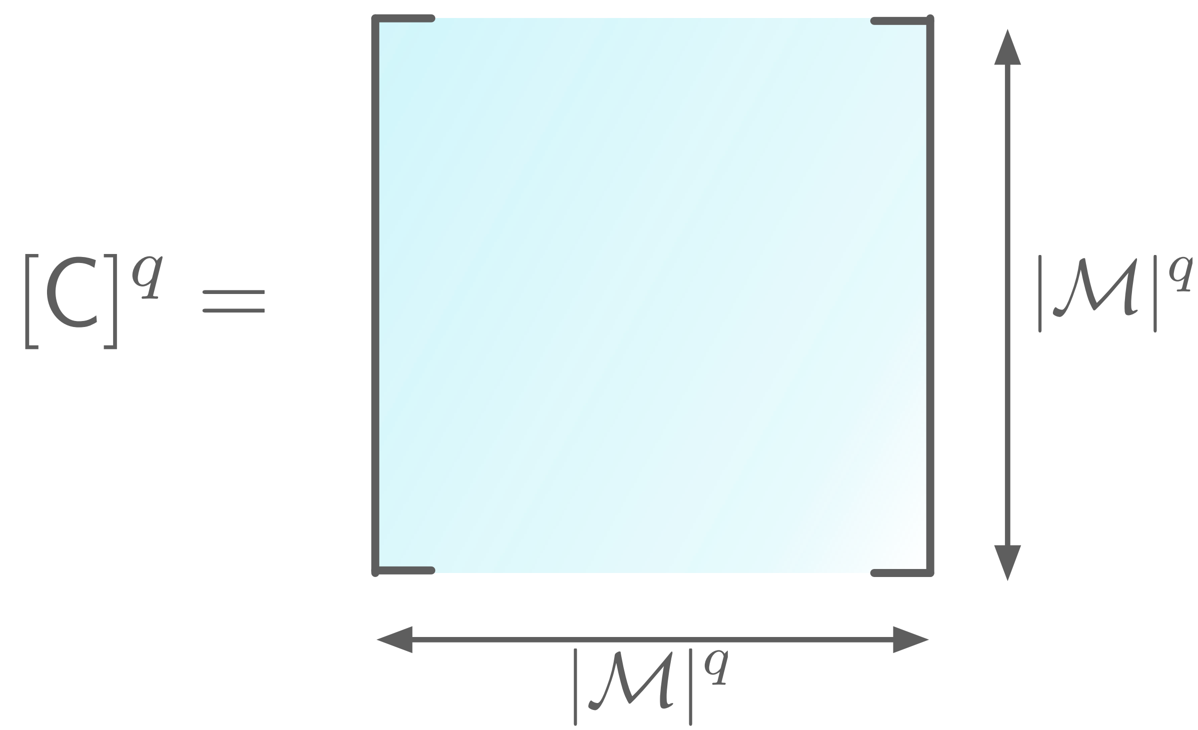
$$\|M\|_{\mathrm{a}} = \max_{x_1} \sum_{y_1} \cdots \max_{x_q} \sum_{y_q} |M_{x,y}|$$

- If $\mathcal{A}$ is non-adaptive:

$$\max_{\mathcal{A}_{\mathrm{na}}} \mathrm{Adv}_{\mathcal{A}_{\mathrm{na}}}(\mathsf{C}, \mathsf{C}^\star) = \frac{1}{2} \| [\mathsf{C}]^q - [\mathsf{C}^\star]^q \|_{\infty}$$

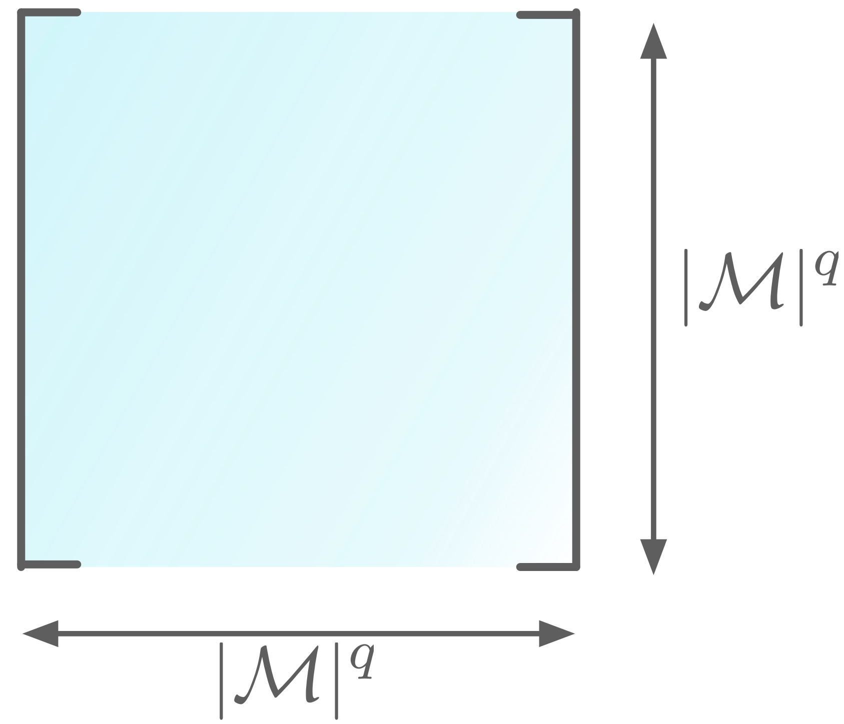
$$\|M\|_{\infty} = \max_{x_1,\ldots,x_q} \sum_{y_1,\ldots,y_q} |M_{x,y}|$$

Are we done then? Not Quite :-<



Are we done then? Not Quite :-<

$[\mathsf{C}]^q =$

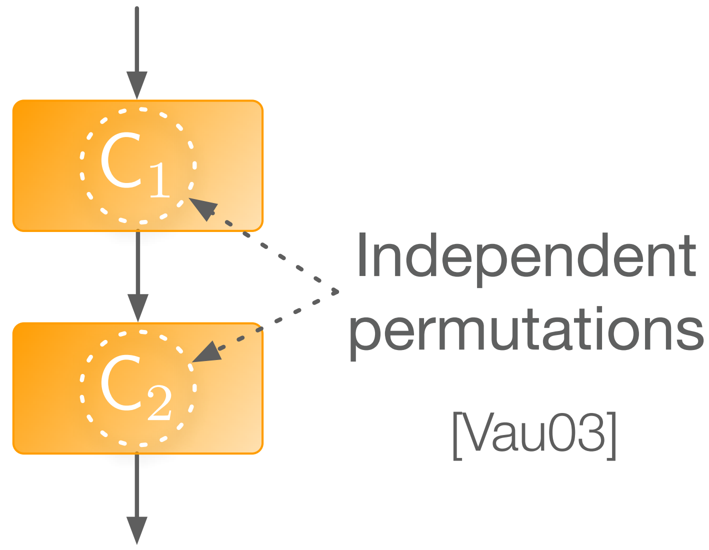


$|\mathcal{M}^q| = 2^{128 \cdot q}$ for a 128-bits block cipher

Tricks for Computing $\mathrm{Adv}_{\mathcal{A}}(\mathsf{C}, \mathsf{C}^\star)$

To deal with the size of the distribution matrices:

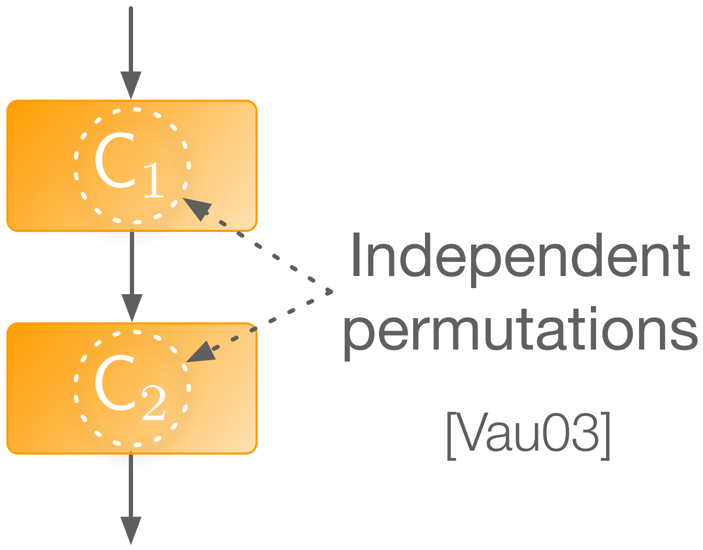
- $[\mathsf{C}_2 \circ \mathsf{C}_1]^q = [\mathsf{C}_1]^q \times [\mathsf{C}_2]^q$



Tricks for Computing $\mathrm{Adv}_{\mathcal{A}}(\mathsf{C}, \mathsf{C}^\star)$

To deal with the size of the distribution matrices:

- $[\mathsf{C}_2 \circ \mathsf{C}_1]^q = [\mathsf{C}_1]^q \times [\mathsf{C}_2]^q$

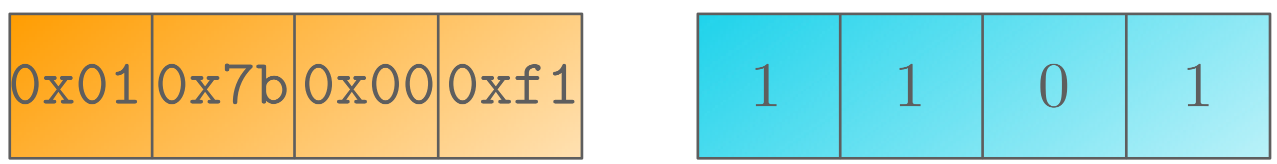


- Take advantage of the symmetries of the block cipher in order to compute the distribution matrix of each round

Notations...

If $a = (a_1, \ldots, a_\ell)$ is an array of m-bit strings, the support of a is the array of $\{0,1\}^\ell$ with 0's at the position where the entry of a is zero and 1's elsewhere

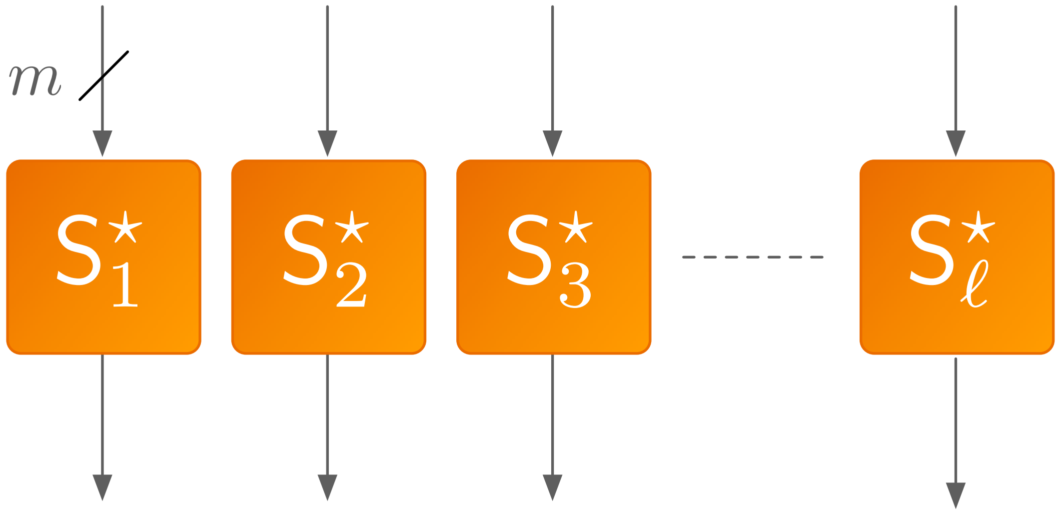
Example:



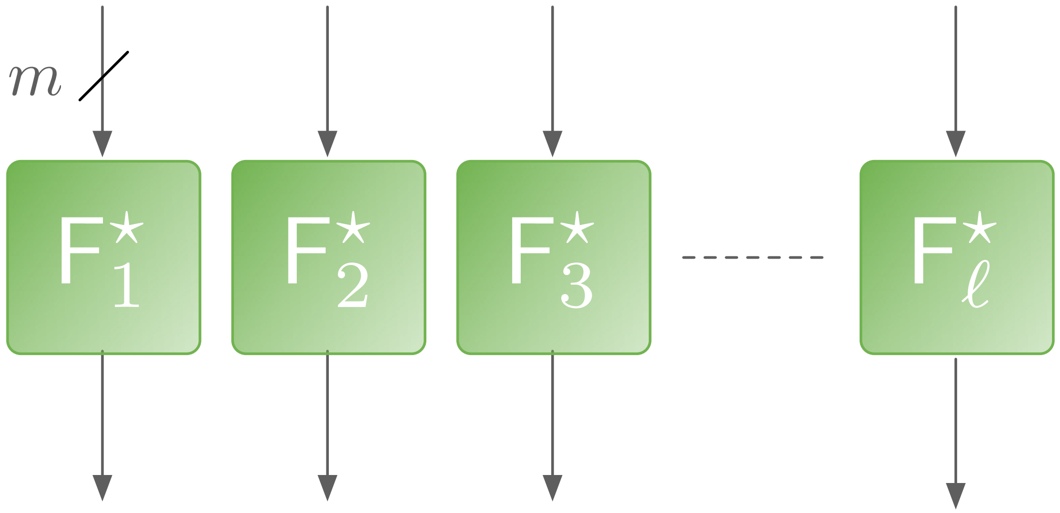
$a = (a_1, a_2, a_3, a_4)$ $\mathrm{supp}(a)$

The weight $w(a)$ of a is the hamming weight of the support (3 in the example).

Decorrelation Modules: Layer of Boxes



- Independent random permutations
- Distribution matrix: $[S]^2$

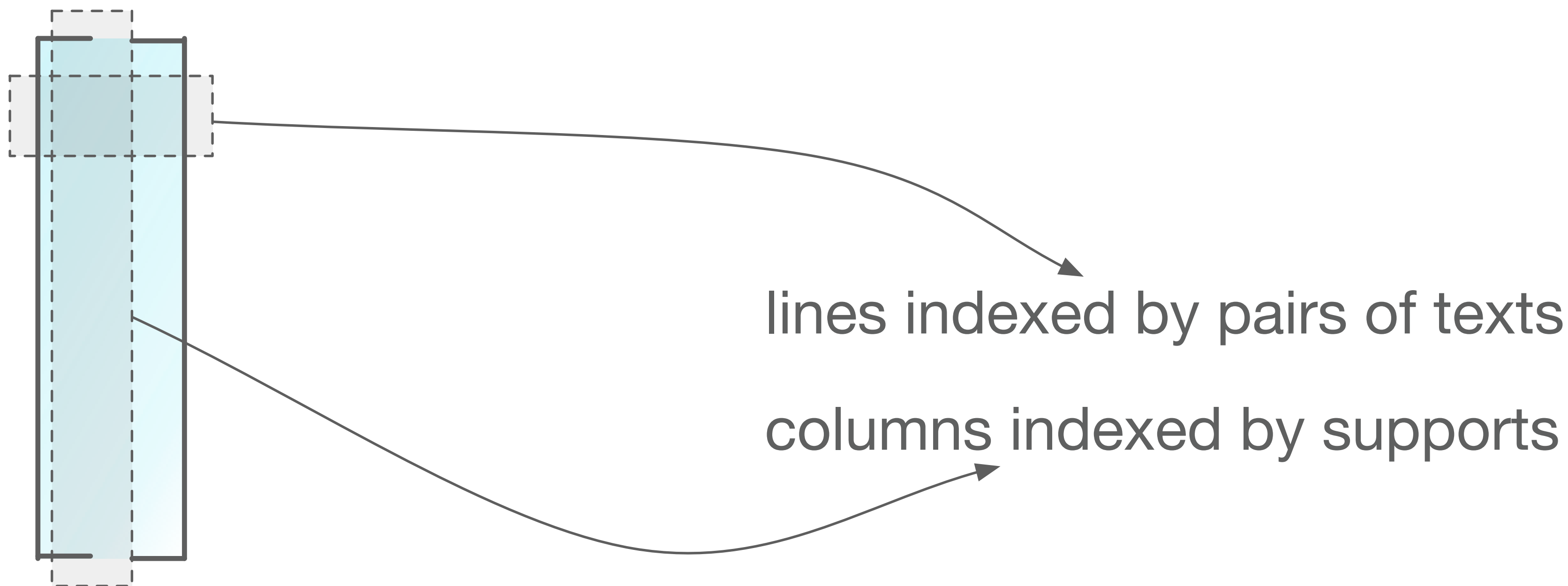


- Independent random functions
- Distribution matrix: $[F]^2$

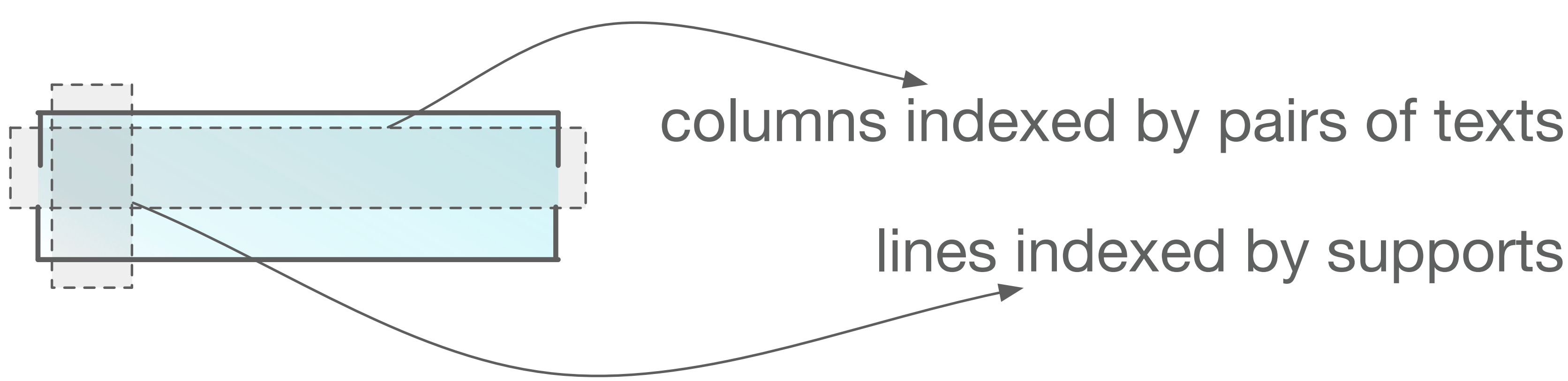
Properties of the two Decorrelation Modules

Introducing the two following transition matrices:

PS =



SP =



Properties of the two Decorrelation Modules

Introducing the two following transition matrices:

$$\mathsf{PS}_{(a,a'),\gamma} = \mathbf{1}_{\gamma=\mathrm{supp}(a\oplus a')}$$

$$\mathsf{SP}_{\gamma,(a,a')} = \mathbf{1}_{\gamma=\mathrm{supp}(a\oplus a')} M^{-\ell}(M-1)^{-w(\gamma)}$$

Properties of the two Decorrelation Modules

Introducing the two following transition matrices:

$$\mathsf{PS}_{(a,a'),\gamma} = \mathbf{1}_{\gamma=\mathrm{supp}(a\oplus a')}$$

$$\mathsf{SP}_{\gamma,(a,a')} = \mathbf{1}_{\gamma=\mathrm{supp}(a\oplus a')} M^{-\ell}(M-1)^{-w(\gamma)}$$

- $\mathsf{SP} \times \mathsf{PS} = \mathsf{Id}$ and $\mathsf{PS} \times \mathsf{SP} = [\mathsf{S}]^2$ (similar result for $[\mathsf{F}]^2$)
- If M is a $2^{2m\ell} \times 2^{2m\ell}$ matrix such that there exists a $2^{\ell} \times 2^{\ell}$ matrix $\overline{\mathsf{M}}$ verifying

$$\mathsf{M} = \mathsf{PS} \times \overline{\mathsf{M}} \times \mathsf{SP}$$

then: $$\|\mathsf{M}\|_{\mathrm{a}} = |||\mathsf{M}|||_{\infty} = |||\overline{\mathsf{M}}|||_{\infty}$$

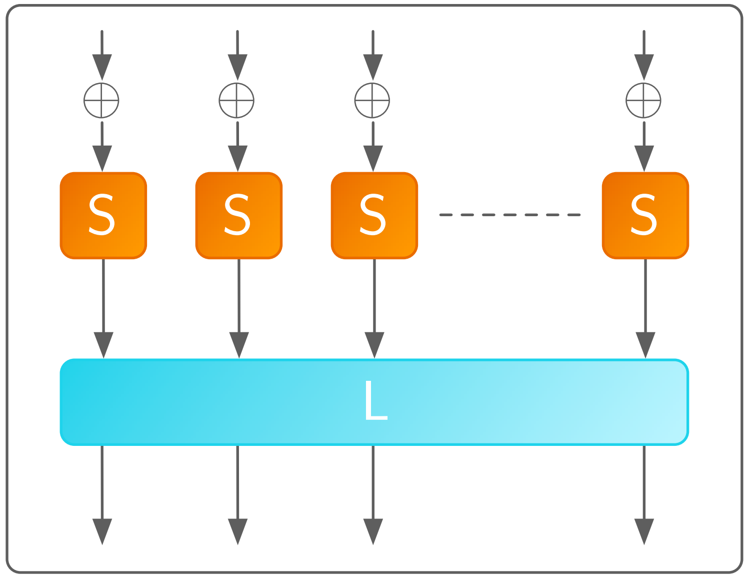
Part II: Designs and Security Proofs

- Dial C for Cipher

Description of $\mathbf{C}$

$\mathbf{C}$ corresponds to the AES where "`addRoundKeys` → `SubBytes`" is replaced by mutually independent random permutations.

AES

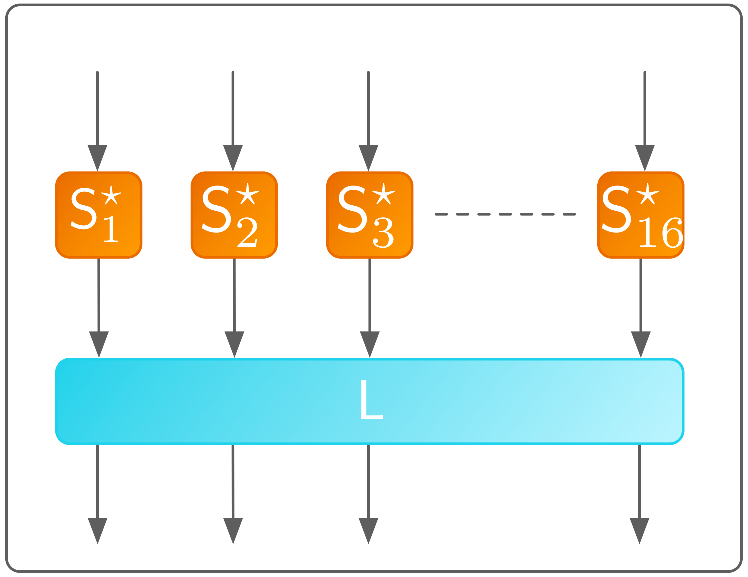


- $\mathbf{C}$ is made of 9 identical rounds, followed by a layer of substitution boxes.
- $\mathbf{C}$ uses $16 \cdot 10 = 160$ mutually independent random 8-bits substitution boxes

Description of $\mathbf{C}$

$\mathbf{C}$ corresponds to the AES where "`addRoundKeys` → `SubBytes`" is replaced by mutually independent random permutations.

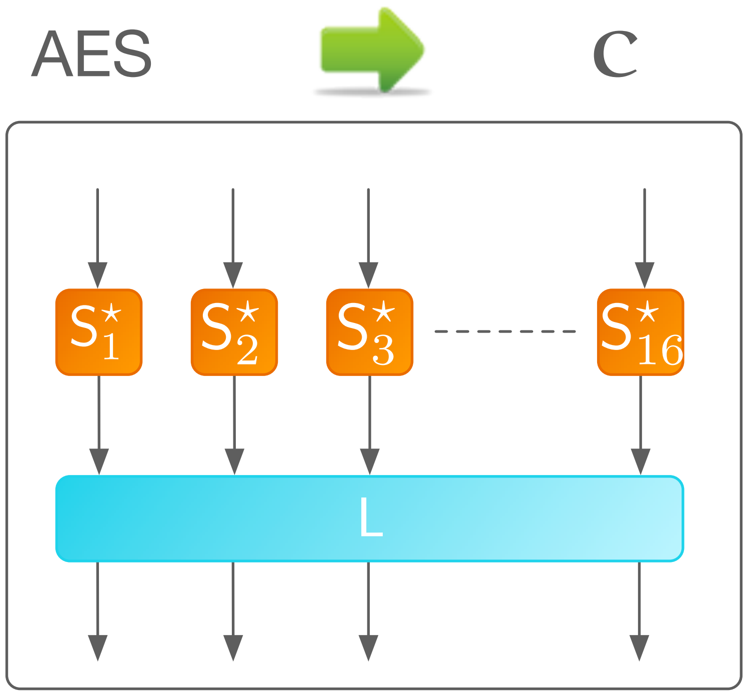
AES → $\mathbf{C}$



- $\mathbf{C}$ is made of 9 identical rounds, followed by a layer of substitution boxes.
- $\mathbf{C}$ uses $16 \cdot 10 = 160$ mutually independent random 8-bits substitution boxes

Description of $\mathbf{C}$

$\mathbf{C}$ corresponds to the AES where "`addRoundKeys` → `SubBytes`" is replaced by mutually independent random permutations.

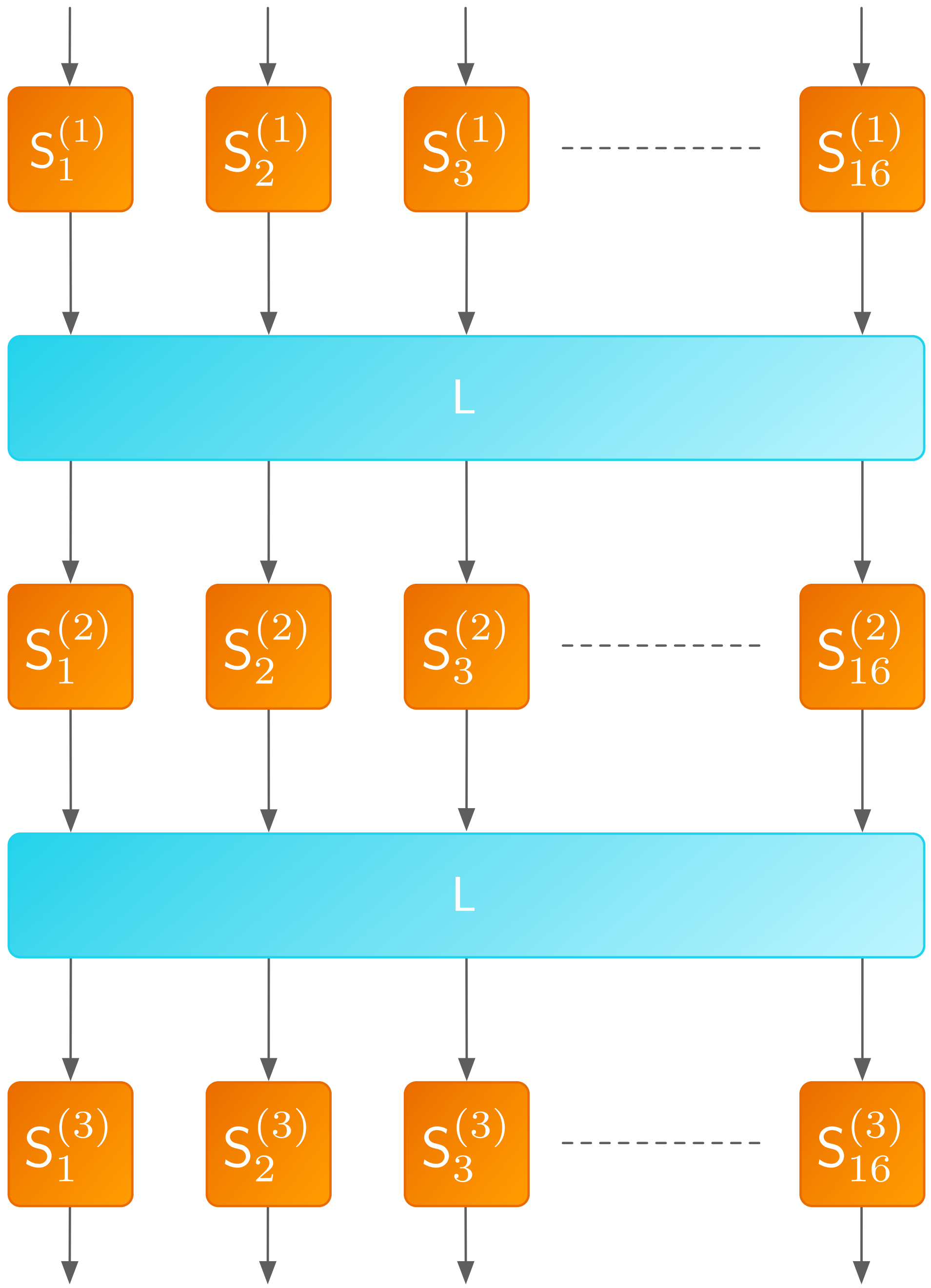


- $\mathbf{C}$ is made of 9 identical rounds, followed by a layer of substitution boxes.
- $\mathbf{C}$ uses $16 \cdot 10 = 160$ mutually independent random 8-bits substitution boxes

Objective: Compute the advantage of the best 2-limited adversary

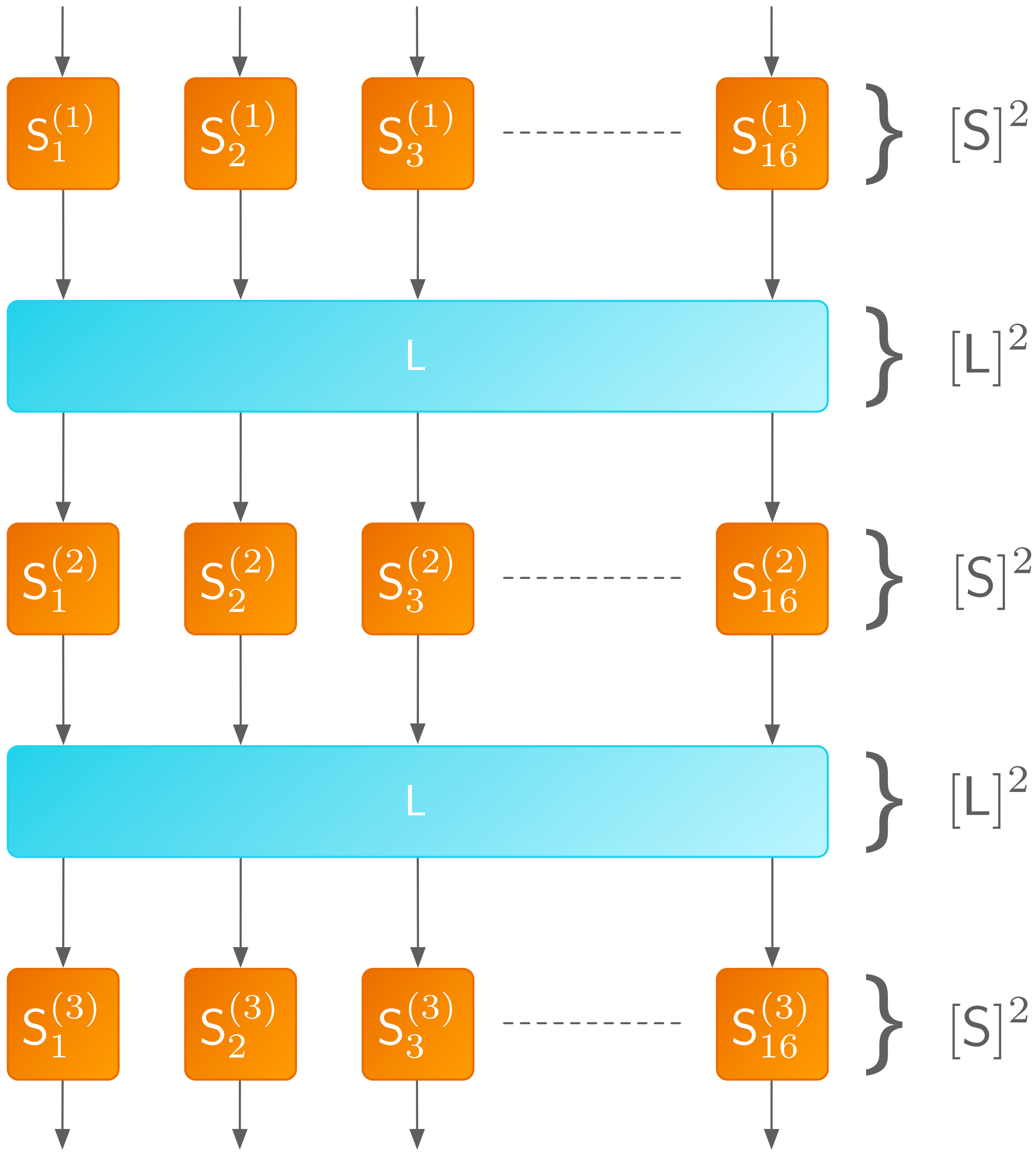
Computing $[\mathbf{C}]^2$

We consider a version of $\mathbf{C}$ reduced to 3 rounds:



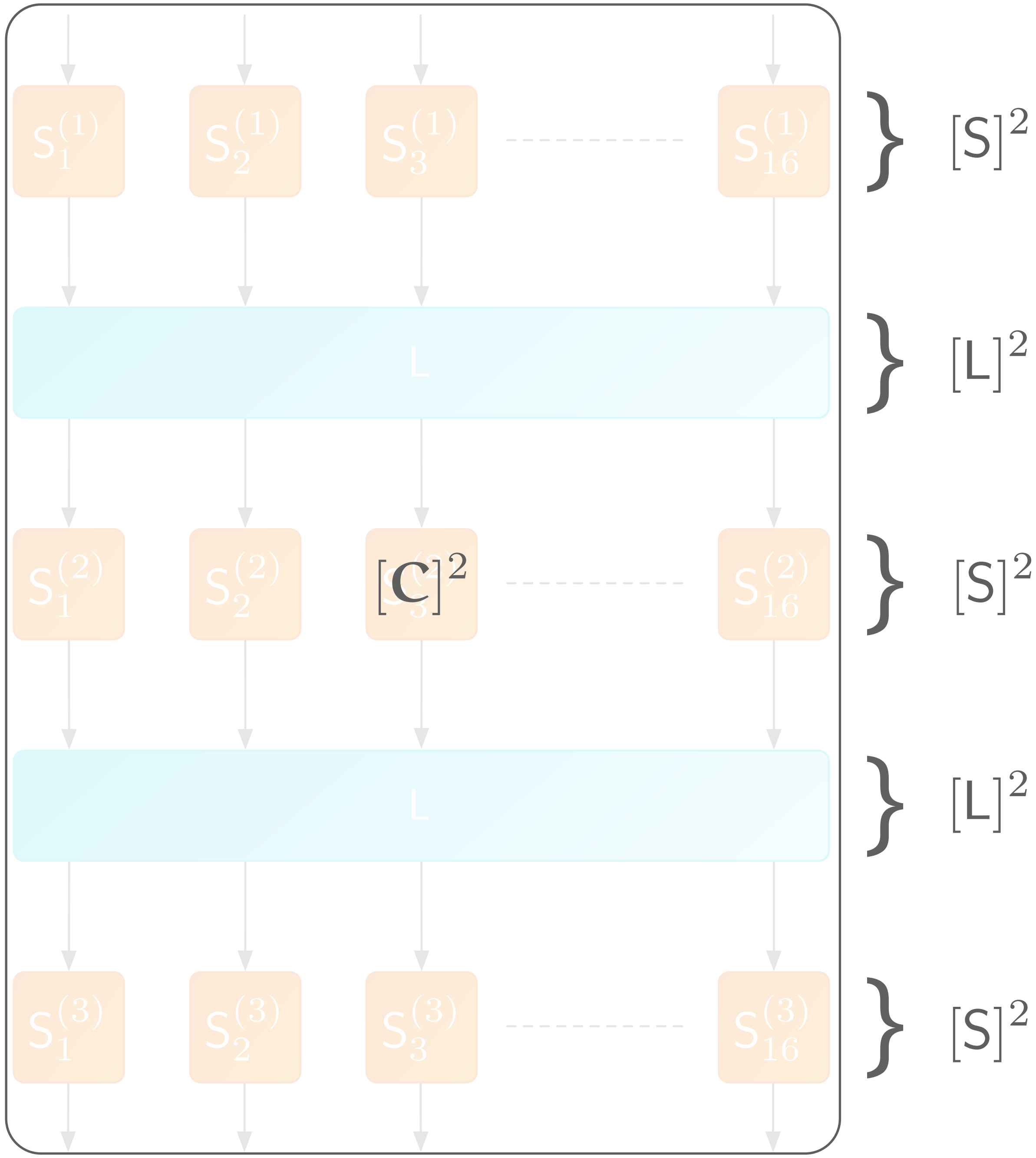
Computing $[\mathbf{C}]^2$

We consider a version of $\mathbf{C}$ reduced to 3 rounds:



Computing $[\mathbf{C}]^2$

We consider a version of $\mathbf{C}$ reduced to 3 rounds:



Computing $[\mathbf{C}]^2$

We consider a version of $\mathbf{C}$ reduced to 3 rounds:

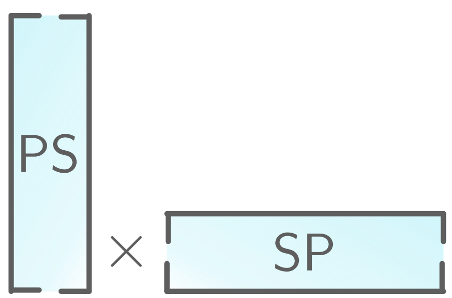
$$[\mathbf{C}]^2 = [\mathsf{S}]^2 \times [\mathsf{L}]^2 \times [\mathsf{S}]^2 \times [\mathsf{L}]^2 \times [\mathsf{S}]^2$$

Computing $[\mathbf{C}]^2$

We consider a version of $\mathbf{C}$ reduced to 3 rounds:

$$[\mathbf{C}]^2 = \quad \times [\mathsf{L}]^2 \times [\mathsf{S}]^2 \times [\mathsf{L}]^2 \times [\mathsf{S}]^2$$

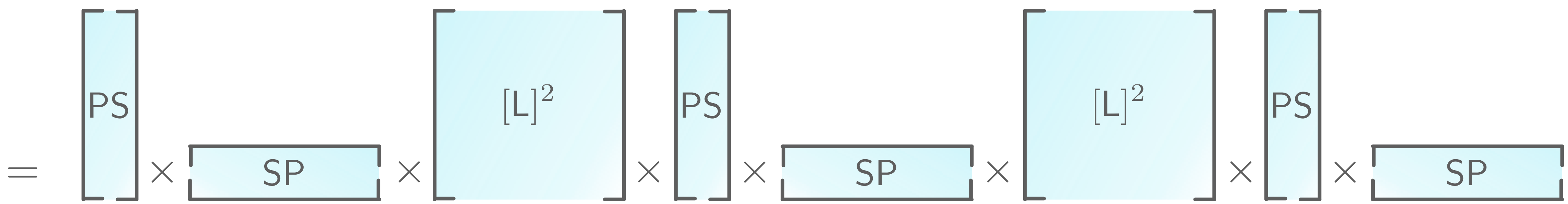
$[\mathsf{S}]^2 =$



Computing $[\mathbf{C}]^2$

We consider a version of $\mathbf{C}$ reduced to 3 rounds:

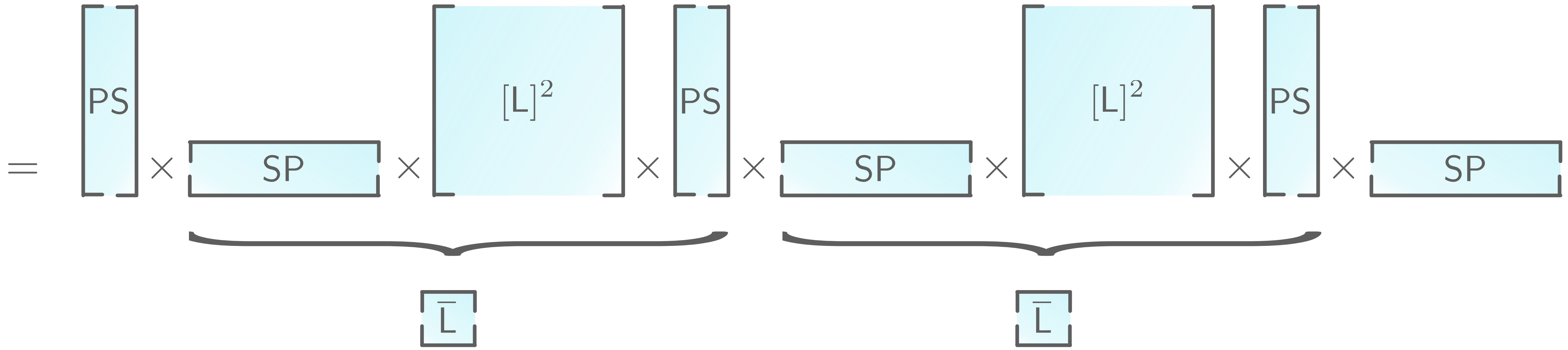
$$[\mathbf{C}]^2 = [\mathsf{S}]^2 \times [\mathsf{L}]^2 \times [\mathsf{S}]^2 \times [\mathsf{L}]^2 \times [\mathsf{S}]^2$$



Computing $[\mathbf{C}]^2$

We consider a version of $\mathbf{C}$ reduced to 3 rounds:

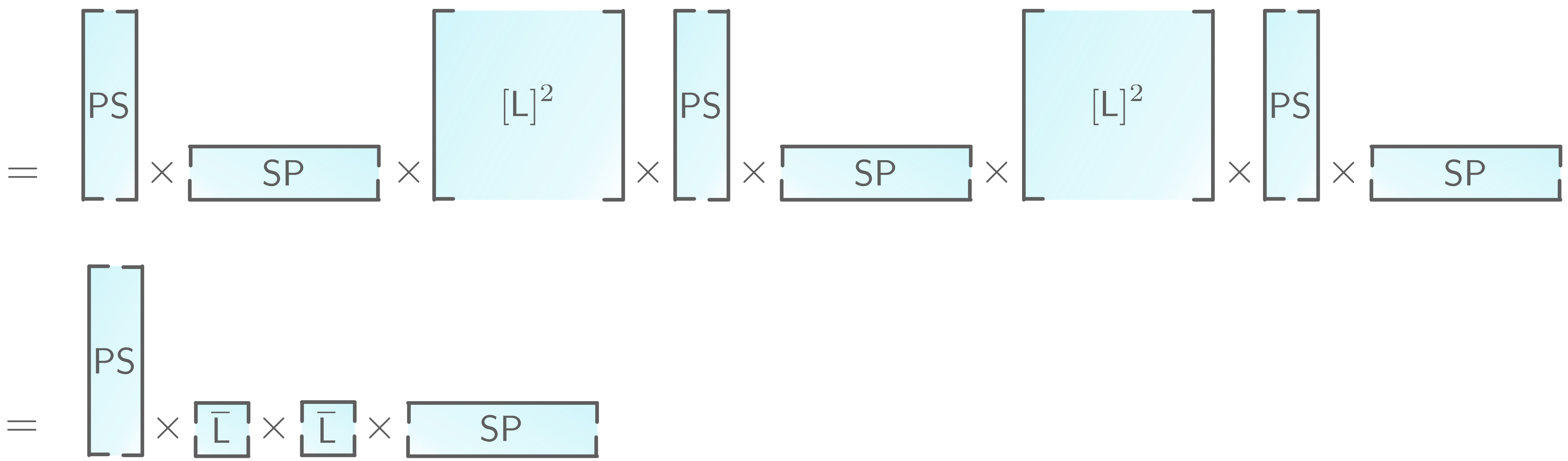
$[\mathbf{C}]^2 = [\mathsf{S}]^2 \times [\mathsf{L}]^2 \times [\mathsf{S}]^2 \times [\mathsf{L}]^2 \times [\mathsf{S}]^2$



Computing $[\mathbf{C}]^2$

We consider a version of $\mathbf{C}$ reduced to 3 rounds:

$$[\mathbf{C}]^2 = [\mathsf{S}]^2 \times [\mathsf{L}]^2 \times [\mathsf{S}]^2 \times [\mathsf{L}]^2 \times [\mathsf{S}]^2$$



Computing $\mathrm{Adv}_{\mathcal{A}}(\mathbf{C}, \mathsf{C}^\star)$

For a r-round version of $\mathbf{C}$ we have:

$$[\mathbf{C}]^2 = \mathsf{PS} \times (\overline{\mathsf{L}})^{r-1} \times \mathsf{SP}$$

where $\overline{\mathsf{L}}$ is a $2^{16} \times 2^{16}$ matrix.

Computing $\mathrm{Adv}_{\mathcal{A}}(\mathbf{C}, \mathsf{C}^\star)$

For a r-round version of $\mathbf{C}$ we have:

$$[\mathbf{C}]^2 = \mathsf{PS} \times (\overline{\mathsf{L}})^{r-1} \times \mathsf{SP}$$

where $\overline{\mathsf{L}}$ is a $2^{16} \times 2^{16}$ matrix.

$$\max_{\mathcal{A}} \mathrm{Adv}_{\mathcal{A}}(\mathbf{C}, \mathsf{C}^\star) = \frac{1}{2} |||(\overline{\mathsf{L}})^{r-1} - \overline{\mathsf{C}^\star}|||_\infty$$

Computing $\mathrm{Adv}_{\mathcal{A}}(\mathbf{C}, \mathsf{C}^\star)$

For a r-round version of $\mathbf{C}$ we have:

$$[\mathbf{C}]^2 = \mathsf{PS} \times (\overline{\mathsf{L}})^{r-1} \times \mathsf{SP}$$

where $\overline{\mathsf{L}}$ is a $2^{16} \times 2^{16}$ matrix.

$$\max_{\mathcal{A}} \mathrm{Adv}_{\mathcal{A}}(\mathbf{C}, \mathsf{C}^\star) = \frac{1}{2} |||(\overline{\mathsf{L}})^{r-1} - \overline{\mathsf{C}^\star}|||_\infty$$

Can we reduce the computational complexity even further?

Yes! But the diffusion has to be chosen with care...

Computing $\mathrm{Adv}_{\mathcal{A}}(\mathbf{C}, \mathsf{C}^\star)$

For a r-round version of $\mathbf{C}$ we have:

$$[\mathbf{C}]^2 = \mathsf{PS} \times (\overline{\mathsf{L}})^{r-1} \times \mathsf{SP}$$

where $\overline{\mathsf{L}}$ is a $2^{16} \times 2^{16}$ matrix.

$$\max_{\mathcal{A}} \mathrm{Adv}_{\mathcal{A}}(\mathbf{C}, \mathsf{C}^\star) = \frac{1}{2} |||(\overline{\mathsf{L}})^{r-1} - \overline{\mathsf{C}^\star}|||_\infty$$

Can we reduce the computational complexity even further?

Yes! But the diffusion has to be chosen with care...

$$\max_{\mathcal{A}} \mathrm{Adv}_{\mathcal{A}}(\mathbf{C}, \mathsf{C}^\star) = \frac{1}{2} ||| \left(\overline{\overline{\mathsf{L}}} \times \mathsf{W}\right)^{r-2} \times \overline{\overline{\mathsf{L}}} - \overline{\overline{\mathsf{C}^\star}}|||_\infty$$

Computing the advantage of the best distinguisher (either adaptive or not) only requires operations on 625×625 matrices (instead of $2^{256} \times 2^{256}$ initially).

Values of $\mathrm{Adv}_{\mathcal{A}}(\mathbf{C}, \mathsf{C}^\star)$

r	1	2	3	4	5	6
$\mathrm{Adv}(\mathsf{C}, \mathsf{C}^\star)$	1	1	$2^{-4.0}$	$2^{-23.4}$	$2^{-45.8}$	$2^{-71.0}$

r	7	8	9	10	11	12
$\mathrm{Adv}(\mathsf{C}, \mathsf{C}^\star)$	$2^{-126.3}$	$2^{-141.3}$	$2^{-163.1}$	$2^{-185.5}$	$2^{-210.8}$	$2^{-238.9}$

Values of $\mathrm{Adv}_{\mathcal{A}}(\mathbf{C}, \mathsf{C}^\star)$

r	1	2	3	4	5	6
$\mathrm{Adv}(\mathsf{C}, \mathsf{C}^\star)$	1	1	$2^{-4.0}$	$2^{-23.4}$	$2^{-45.8}$	$2^{-71.0}$
r	7	8	9	10	11	12
$\mathrm{Adv}(\mathsf{C}, \mathsf{C}^\star)$	$2^{-126.3}$	$2^{-141.3}$	$2^{-163.1}$	$2^{-185.5}$	$2^{-210.8}$	$2^{-238.9}$

7 rounds of $\mathbf{C}$ are enough to obtain provable security against 2-limited adversaries

Other Security Results

Using decorrelation techniques, the security results concerning 2-limited adversaries immediately imply security bounds against:

- linear and differential cryptanalysis (the linear hull and the differentials effect being taken into account)
- iterated attacks of order 1

After some more computations, we manage to compute the exact security against LC and DC, prove that no impossible differential exists, and show that $\mathbf{C}$ tends towards the perfect cipher as r increases (as far as LC and DC are concerned).

Part II: Designs and Security Proofs

- KFC: the Krazy Feistel Cipher

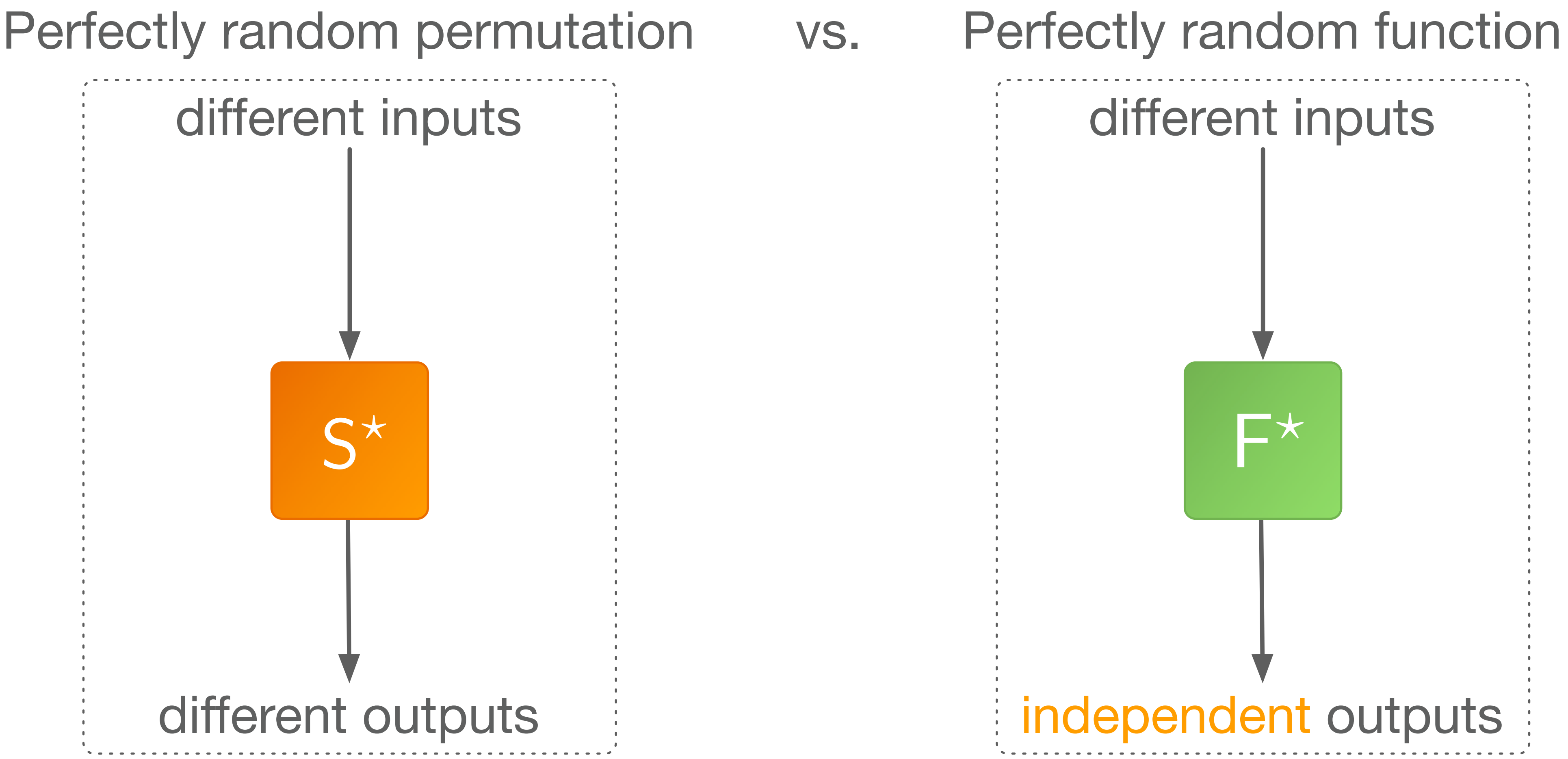
What about Higher Orders?

We did not manage to prove the security of $\mathbf{C}$ against higher q-limited adversaries for $q > 2$.

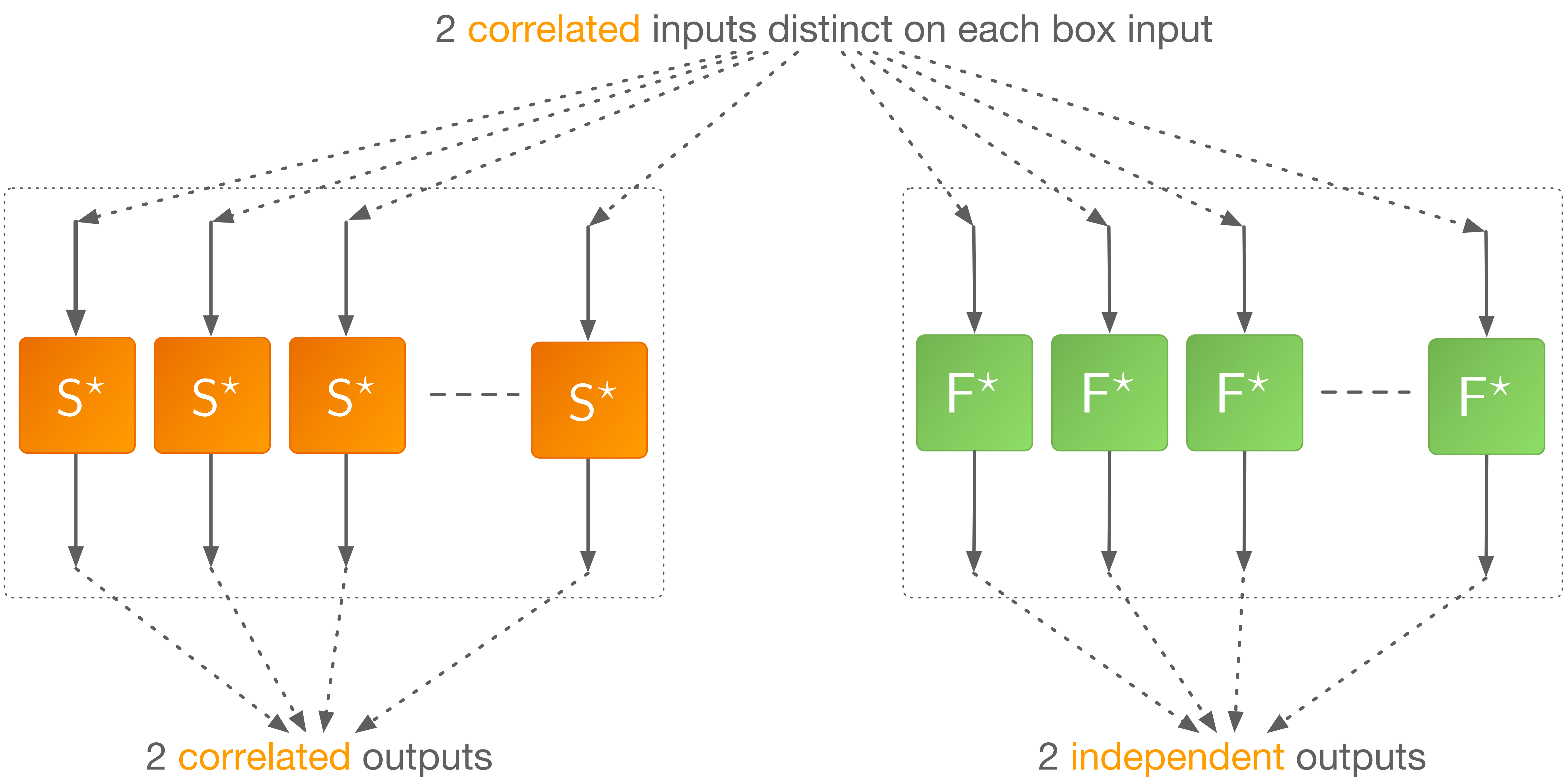
What about Higher Orders?

We did not manage to prove the security of $\mathbf{C}$ against higher q-limited adversaries for $q > 2$.

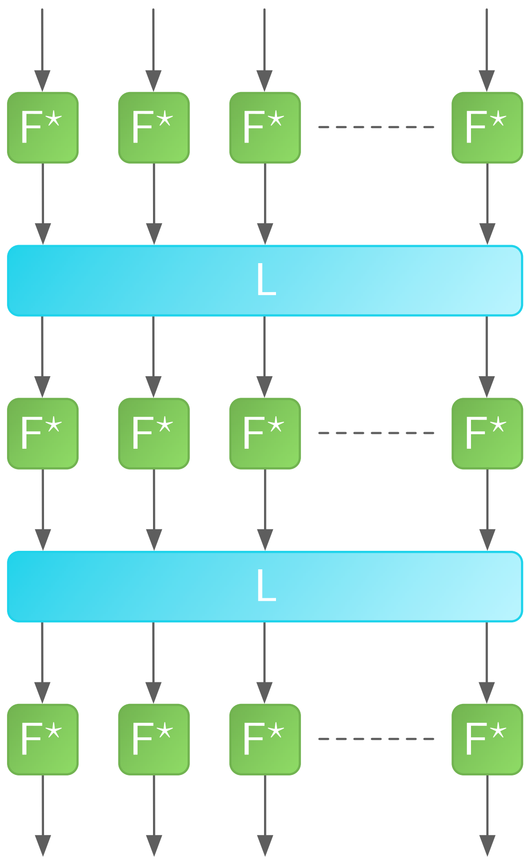
Idea: try to bound the advantage of the best q-limited adversary by that of the best (q-1)-limited adversary.



Rand. Permutations vs. Rand. Functions

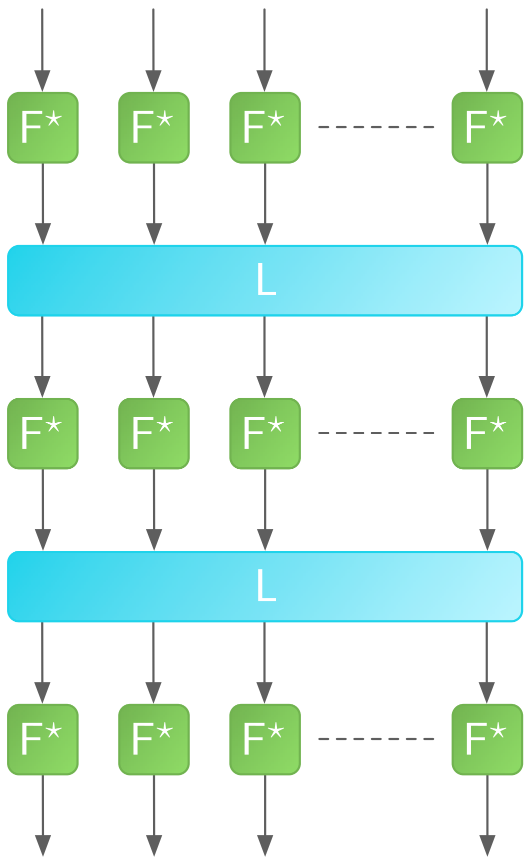


Towards a New Construction



Towards a New Construction

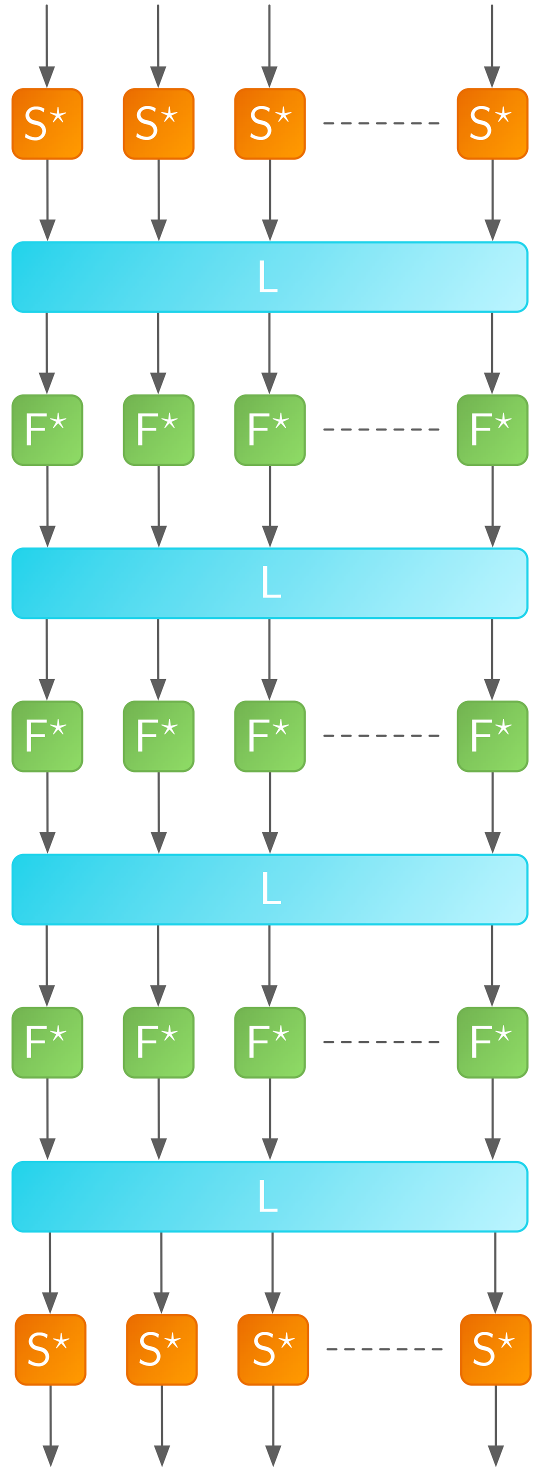
- Non negligible risk of collision after a F-box



Towards a New Construction

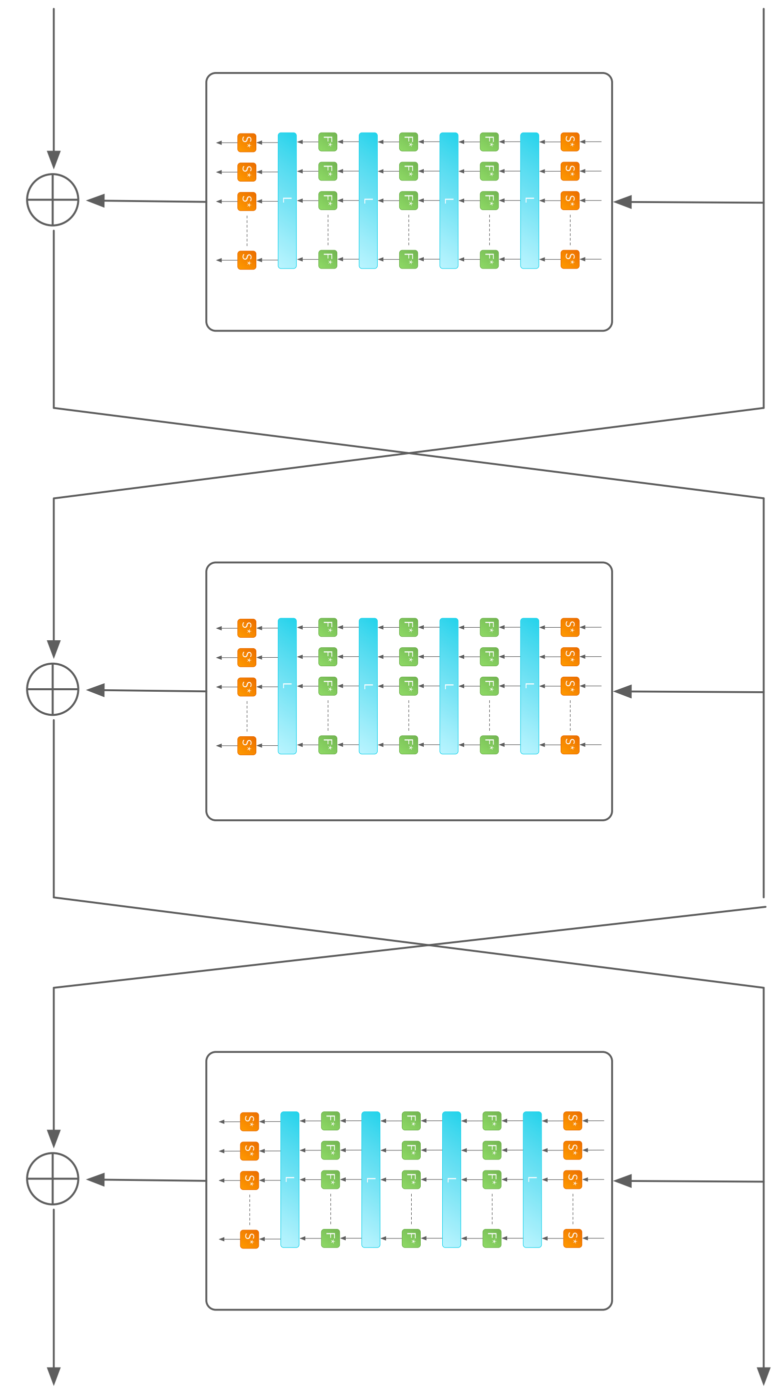
- Non negligible risk of collision after a F-box

- Use the “sandwich technique” to obtain (almost) pairwise independent inputs before the layer of random functions.



Towards a New Construction

- Non negligible risk of collision after a F-box
- Use the “sandwich technique” to obtain (almost) pairwise independent inputs before the layer of random functions.
- The construction is not invertible. We plug it in a Feistel scheme.



Results obtained on KFC

- With this approach, we manage to prove the security against adversaries up to the order 70 (for an unreasonable set of parameters).
- The bounds are not tight at all it is certainly possible to improve our results.

Results obtained on KFC

- With this approach, we manage to prove the security against adversaries up to the order 70 (for an unreasonable set of parameters).
- The bounds are not tight at all it is certainly possible to improve our results.

Part II: Designs and Security Proofs

- Critics

Requirements & Uncovered Attacks

- $\mathbf{C}$ might never fit, say, RFID tags (in the best case, we need 160kB of memory to store the tables).
- We proposed so-called “provably secure” block ciphers...
- ...which are not provably secure against all known attacks.
- e.g., $\mathbf{C}$ is not provably secure against cache attacks or saturation attacks.

On the Independence of the Round Keys

- Our proofs assume that the rounds are mutually independent.
- This is not true in practice: thousands of bits of randomness are derived from a 128 bit key.
- Using a cryptographically secure PRNG, we can show that if an attack applies on the block cipher with the key schedule, but not on the block cipher with mutually independent rounds, then the PRNG's sequence can be distinguished from pure random.

Two Sides to Every Story

- Pessimistic view (not my favorite):
  - Since we need more bits of randomness to generate the boxes than the number of bits we are allowed to encrypt, why not use the bits generated with BBS or QUAD as a one-time-pad... and throw away all the constructions? ☹
- Optimistic View:
  - The assumption about the independence of the round keys has nothing to do with the block cipher itself, but with the key schedule.
  - If a “provably secure” block cipher is broken by an attack against which it should resist ➡ make the key schedule stronger!
  - Making sure that the distribution matrix of the block cipher considered is close to that of C^* appears to be very natural. Independently of the key schedule, it’s a strong security argument.

Conclusion

"[...] the methodology of provable security has become unavoidable in designing and evaluating new schemes"

[JSe03]

“[...] the methodology of provable security has become unavoidable in designing and evaluating new schemes”

[JSe03]

public key schemes

“[...] the methodology of provable security has become unavoidable in designing and evaluating new schemes”
[JSe03]

public key schemes

We hope to have made a significant step towards its extension to block ciphers!

Thank you for your attention!

☺

Publications

[BVicits08] *The Complexity of Distinguishing Distributions*
Joint work with Serge Vaudenay
Published in the proceedings of ICITS 08 (Calgary, Canada)

[BSVsac07] *Linear Cryptanalysis of Non Binary Ciphers (with an application to* SAFER*)*
Joint work with Jacques Stern & Serge Vaudenay
Published in the proceedings of SAC 07 (Ottawa, Canada)

[BFa06] KFC - *The Krazy Feistel Cipher*
Joint work with Matthieu Finiasz
Published in the proceedings of Asiacrypt 06 (Shangai, China)

[BFsac06] *Dial* C *for Cipher*
Joint work with Matthieu Finiasz
Published in the proceedings of SAC 06 (Montreal, Canada)

[BVsac05] *Proving the Security of the* AES *Substitution-Permutation Network*
Joint work with Serge Vaudenay
Published in the proceedings of SAC 05 (Kingston, Canada)

[BJVa04] *How Far Can We Go Beyond Linear Cryptanalysis?*
Joint work with Pascal Junod & Serge Vaudenay
Published in the proceedings of Asiacrypt 04 (Jeju Island, Korea)