Cryptology

Sabyasachi Karati

Assistant Professor
Cryptology and Security Research Unit (C.S.R.U)
R. C. Bose Centre for Cryptology and Security
Indian Statistical Institute (ISI)
Kolkata, India

Lecture 04

Computational Ciphers and Semantic Security

A Computational Approach

Till Now

- Studied Classical Cryptography.

A Computational Approach

Till Now

- Studied Classical Cryptography.
  - Studied some historical ciphers and how they can be broken.

Till Now

- Studied Classical Cryptography.
    - Studied some historical ciphers and how they can be broken.
    - Studied Perfect Security, also known as Information-Theoretic Security.

A Computational Approach

Till Now

- Studied Classical Cryptography.
  - Studied some historical ciphers and how they can be broken.
  - Studied Perfect Security, also known as Information-Theoretic Security.
    - Mathematically proven secure in ciphertext-only attack, even when the adversary has unlimited computational power.
    - Impractical.

A Computational Approach

Till Now

- Studied Classical Cryptography.
  - Studied some historical ciphers and how they can be broken.
  - Studied Perfect Security, also known as Information-Theoretic Security.
    - Mathematically proven secure in ciphertext-only attack, even when the adversary has unlimited computational power.
    - Impractical.

New Direction

- Modern Cryptography moved from perfect security to computational security.

A Computational Approach

Till Now

- Studied Classical Cryptography.
  - Studied some historical ciphers and how they can be broken.
  - Studied Perfect Security, also known as Information-Theoretic Security.
    - Mathematically proven secure in ciphertext-only attack, even when the adversary has unlimited computational power.
    - Impractical.

New Direction

- Modern Cryptography moved from perfect security to computational security.
- Computational security is weaker notion of security than Perfect security.

- Kerckhoffs actually spelled out six principles.
- Following is relevant to the new approach.

Kerckhoffs' Principle

A [cipher] must be practically, if not mathematically, indecipherable.

- Kerckhoffs actually spelled out six principles.
- Following is relevant to the new approach.

Kerckhoffs' Principle

A [cipher] must be practically, if not mathematically, indecipherable.

- It suffices to use a scheme that cannot be broken in reasonable time with any reasonable probability of success.

Required Relaxations

Two Relaxations

- Security is only preserved against efficient adversaries,

Required Relaxations

Two Relaxations

- Security is only preserved against efficient adversaries, and
- Adversaries can potentially succeed with some very small probability (small enough so that we are not concerned that it will ever really happen).

Required Relaxations

Two Relaxations

- Security is only preserved against efficient adversaries, and
- Adversaries can potentially succeed with some very small probability (small enough so that we are not concerned that it will ever really happen).

To achieve Meaningful Theory

- Concrete Approach,

Required Relaxations

Two Relaxations

- Security is only preserved against efficient adversaries, and
- Adversaries can potentially succeed with some very small probability (small enough so that we are not concerned that it will ever really happen).

To achieve Meaningful Theory

- Concrete Approach, and
- Asymptotic Approach.

Concrete Approach

- Quantifies the security by bounding the maximum success probability of any adversary running for at most some specified amount of time.

Concrete Approach

- Quantifies the security by bounding the maximum success probability of any adversary running for at most some specified amount of time.
- Let t, ϵ be positive constants with $\epsilon \leqslant 1$. We say:

A cryptographic scheme is (t, ϵ)-secure if every adversary running for time at most t succeeds in breaking the scheme with probability at most ϵ.

Concrete Approach

- Quantifies the security by bounding the maximum success probability of any adversary running for at most some specified amount of time.
- Let t, ϵ be positive constants with $\epsilon \leqslant 1$. We say:

A cryptographic scheme is (t, ϵ)-secure if every adversary running for time at most t succeeds in breaking the scheme with probability at most ϵ.

Example

Against a cryptographic scheme with key length is n, an adversary running for time t (in CPU cycles) succeeds in breaking the scheme with probability at most $t/2^n$.

Concrete Approach

- Quantifies the security by bounding the maximum success probability of any adversary running for at most some specified amount of time.
- Let t, ϵ be positive constants with $\epsilon \leqslant 1$. We say:

A cryptographic scheme is (t, ϵ)-secure if every adversary running for time at most t succeeds in breaking the scheme with probability at most ϵ.

Example

Against a cryptographic scheme with key length is n, an adversary running for time t (in CPU cycles) succeeds in breaking the scheme with probability at most $t/2^n$.

- Let $n = 80$ and adversary has a computer with 4 GHz processor

Concrete Approach

- Quantifies the security by bounding the maximum success probability of any adversary running for at most some specified amount of time.
- Let t, ϵ be positive constants with $\epsilon \leqslant 1$. We say:

A cryptographic scheme is (t, ϵ)-secure if every adversary running for time at most t succeeds in breaking the scheme with probability at most ϵ.

Example

Against a cryptographic scheme with key length is n, an adversary running for time t (in CPU cycles) succeeds in breaking the scheme with probability at most $t/2^n$.

- Let $n = 80$ and adversary has a computer with 4 GHz processor
  - Let $t = 2^{60}$. Then $t = \frac{2^{60}}{4\times 10^9}$ sec $\approx$ 9 years with success probability 2^{-20}.

Concrete Approach

- Quantifies the security by bounding the maximum success probability of any adversary running for at most some specified amount of time.
- Let t, ϵ be positive constants with $\epsilon \leqslant 1$. We say:

A cryptographic scheme is (t, ϵ)-secure if every adversary running for time at most t succeeds in breaking the scheme with probability at most ϵ.

Example

Against a cryptographic scheme with key length is n, an adversary running for time t (in CPU cycles) succeeds in breaking the scheme with probability at most $t/2^n$.

- Let $n = 80$ and adversary has a computer with 4 GHz processor
  - Let $t = 2^{60}$. Then $t = \frac{2^{60}}{4\times 10^9}\,\mathsf{sec} \approx 9$ years with success probability 2^{-20}.
  - Let $t = 2^{80}$. Then $t = 9 \times 2^{20}$ years with success probability 1.

Concrete Approach

- Typical value of key length $n = 128$.

Concrete Approach

- Typical value of key length $n = 128$.
- How big is this number 2^{128}?

Concrete Approach

- Typical value of key length $n = 128$.
- How big is this number 2^{128}?

According to physicists' estimates the number of seconds since the big bang is on the order of 2^{58}.

Concrete Approach

Warning

- Precise concrete guarantees are difficult to provide.
- Concrete security claims must be interpreted careful.

Concrete Approach

Warning

- Precise concrete guarantees are difficult to provide.
- Concrete security claims must be interpreted careful.
- Consider the following claim:

No adversary running for 5 years can break a given scheme with probability better than ϵ.

Concrete Approach

Warning

- Precise concrete guarantees are difficult to provide.
- Concrete security claims must be interpreted careful.
- Consider the following claim:

No adversary running for 5 years can break a given scheme with probability better than ϵ.

- Questions
  - What type of computing power does this assume?

Concrete Approach

Warning

- Precise concrete guarantees are difficult to provide.
- Concrete security claims must be interpreted careful.
- Consider the following claim:

No adversary running for 5 years can break a given scheme with probability better than ϵ.

- Questions
  - What type of computing power does this assume?
  - Does this take into account future advances in computing power (e.g., Moore's Law)?

Concrete Approach

Warning

- Precise concrete guarantees are difficult to provide.
- Concrete security claims must be interpreted careful.
- Consider the following claim:

No adversary running for 5 years can break a given scheme with probability better than ϵ.

- Questions
    - What type of computing power does this assume?
    - Does this take into account future advances in computing power (e.g., Moore's Law)?
    - Does the estimate assume the use of off-the-shelf algorithms, or dedicated software implementations optimized for the attack?

Concrete Approach

Warning

- Precise concrete guarantees are difficult to provide.
- Concrete security claims must be interpreted careful.
- Consider the following claim:

No adversary running for 5 years can break a given scheme with probability better than ϵ.

- Questions
  - What type of computing power does this assume?
  - Does this take into account future advances in computing power (e.g., Moore's Law)?
  - Does the estimate assume the use of off-the-shelf algorithms, or dedicated software implementations optimized for the attack?
  - Says little about the success probability of an adversary running for 2 years (other than the fact that it can be at most ϵ) and says nothing about the success probability of an adversary running for 10 years.

- This approach, rooted in complexity theory.

Asymptotic Approach

- This approach, rooted in complexity theory.
- Introduces a new parameter: Security Parameter $n \in \mathbb{N}$.

Asymptotic Approach

- This approach, rooted in complexity theory.
- Introduces a new parameter: Security Parameter $n \in \mathbb{N}$.
    - Known to honest parties and the adversary.

Asymptotic Approach

- This approach, rooted in complexity theory.
- Introduces a new parameter: Security Parameter $n \in \mathbb{N}$.
  - Known to honest parties and the adversary.
  - When honest parties initialize a scheme, they choose some value n as the security parameter and then runs $\mathcal{G}$.

Asymptotic Approach

- This approach, rooted in complexity theory.
- Introduces a new parameter: Security Parameter $n \in \mathbb{N}$.
  - Known to honest parties and the adversary.
  - When honest parties initialize a scheme, they choose some value n as the security parameter and then runs $\mathcal{G}$.
  - Running time and success probability of the adversary are functions Security Parameter n.

Asymptotic Approach

- This approach, rooted in complexity theory.
- Introduces a new parameter: Security Parameter $n \in \mathbb{N}$.
  - Known to honest parties and the adversary.
  - When honest parties initialize a scheme, they choose some value n as the security parameter and then runs $\mathcal{G}$.
  - Running time and success probability of the adversary are functions Security Parameter n.
    1. Efficient adversaries: Randomized (i.e., probabilistic) algorithms running in time polynomial in n (PPT algorithms).

Asymptotic Approach

- This approach, rooted in complexity theory.
- Introduces a new parameter: Security Parameter $n \in \mathbb{N}$.
  - Known to honest parties and the adversary.
  - When honest parties initialize a scheme, they choose some value n as the security parameter and then runs $\mathcal{G}$.
  - Running time and success probability of the adversary are functions Security Parameter n.
    1. Efficient adversaries: Randomized (i.e., probabilistic) algorithms running in time polynomial in n (PPT algorithms).
    2. For real-world efficiency: It is required that honest parties run in polynomial time. For example, algorithms $\mathcal{G}$, $\mathcal{E}$ and $\mathcal{D}$ all run in polynomial time.

Asymptotic Approach

- This approach, rooted in complexity theory.
- Introduces a new parameter: Security Parameter $n \in \mathbb{N}$.
  - Known to honest parties and the adversary.
  - When honest parties initialize a scheme, they choose some value n as the security parameter and then runs $\mathcal{G}$.
  - Running time and success probability of the adversary are functions Security Parameter n.
    1. Efficient adversaries: Randomized (i.e., probabilistic) algorithms running in time polynomial in n (PPT algorithms).
    2. For real-world efficiency: It is required that honest parties run in polynomial time. For example, algorithms $\mathcal{G}$, $\mathcal{E}$ and $\mathcal{D}$ all run in polynomial time.
    3. Small probabilities of success: Probabilities that are called negligible that may as well be regarded as being equal to zero for all practical purposes.

A cryptographic scheme is secure if every PPT adversary succeeds in breaking the scheme with only negligible probability.

A cryptographic scheme is secure if every PPT adversary succeeds in breaking the scheme with only negligible probability.

- **Example:**
  - Suppose that an adversary running for n^3 minutes can succeed in breaking the scheme with probability $2^{40} \cdot 2^{-n}$, where $\epsilon(n) = 2^{-n}$.

A cryptographic scheme is secure if every PPT adversary succeeds in breaking the scheme with only negligible probability.

- **Example:**
  - Suppose that an adversary running for n^3 minutes can succeed in breaking the scheme with probability $2^{40} \cdot 2^{-n}$, where $\epsilon(n) = 2^{-n}$.
  - $n = 40$**:** Means that an adversary running for 40^3 minutes $\approx$ 6 weeks, can break the scheme with probability 1.

A cryptographic scheme is secure if every PPT adversary succeeds in breaking the scheme with only negligible probability.

- **Example:**
  - Suppose that an adversary running for n^3 minutes can succeed in breaking the scheme with probability $2^{40} \cdot 2^{-n}$, where $\epsilon(n) = 2^{-n}$.
  - $n = 40$**:** Means that an adversary running for 40^3 minutes $\approx$ 6 weeks, can break the scheme with probability 1.
  - $n = 50$**:** Even then an adversary running for 50^3 minutes $\approx$ 3 months can break the scheme with probability 2^{-10}.

A cryptographic scheme is secure if every PPT adversary succeeds in breaking the scheme with only negligible probability.

- **Example:**
  - Suppose that an adversary running for n^3 minutes can succeed in breaking the scheme with probability $2^{40} \cdot 2^{-n}$, where $\epsilon(n) = 2^{-n}$.
  - $n = 40$**:** Means that an adversary running for 40^3 minutes $\approx$ 6 weeks, can break the scheme with probability 1.
  - $n = 50$**:** Even then an adversary running for 50^3 minutes $\approx$ 3 months can break the scheme with probability 2^{-10}.
  - $n = 500$**:** But an adversary running for 200 years breaks the scheme only with probability 2^{-460}.

Asymptotic Approach

- Larger security parameter indicates a greater level of security.

Asymptotic Approach

- Larger security parameter indicates a greater level of security.
- Usually, The security parameter determines the length of the key.

Asymptotic Approach

- Larger security parameter indicates a greater level of security.
- Usually, The security parameter determines the length of the key.
- The longer the key, the higher the security.

Asymptotic Approach

- Larger security parameter indicates a greater level of security.
- Usually, The security parameter determines the length of the key.
- The longer the key, the higher the security.
- Enables honest parties to defend against increases in computing power as well as algorithmic advances.

- **Example:** In a cryptographic scheme with security parameter n, honest parties need $10^6 \cdot n^2$ cycles, and an adversary requires $10^8 \cdot n^4$ cycles to succeed with probability $2^{20} \cdot 2^{-n}$.

- **Example:** In a cryptographic scheme with security parameter n, honest parties need $10^6 \cdot n^2$ cycles, and an adversary requires $10^8 \cdot n^4$ cycles to succeed with probability $2^{20} \cdot 2^{-n}$.
  - $n = 50$ and all parties 1 GHz machine:
    - Honest parties need $10^6 \times 50^2 \mathsf{cc} = 2.5\mathsf{sec}$.
    - Adversary needs approximately 1 week and success probability is 2^{-30}.

Asymptotic Approach

- **Example:** In a cryptographic scheme with security parameter n, honest parties need $10^6 \cdot n^2$ cycles, and an adversary requires $10^8 \cdot n^4$ cycles to succeed with probability $2^{20} \cdot 2^{-n}$.
  - $n = 50$ and all parties 1 GHz machine:
    - Honest parties need $10^6 \times 50^2 \text{cc} = 2.5 \text{sec}$.
    - Adversary needs approximately 1 week and success probability is 2^{-30}.
  - All parties upgrades to 16 GHz machines.
    - Honest parties upgrade the security parameter, and let now $n = 100$.
    - Honest parties need $10^6 \times 100^2 \text{cc} = 0.625 \text{sec}$.
    - Adversary needs approximately 10^7 sec $\approx$ 16 weeks and success probability is 2^{-80}.

Asymptotic Approach

- **Example:** In a cryptographic scheme with security parameter n, honest parties need $10^6 \cdot n^2$ cycles, and an adversary requires $10^8 \cdot n^4$ cycles to succeed with probability $2^{20} \cdot 2^{-n}$.
  - $n = 50$ and all parties 1 GHz machine:
    - Honest parties need $10^6 \times 50^2 \text{cc} = 2.5\text{sec}$.
    - Adversary needs approximately 1 week and success probability is 2^{-30}.
  - All parties upgrades to 16 GHz machines.
    - Honest parties upgrade the security parameter, and let now $n = 100$.
    - Honest parties need $10^6 \times 100^2 \text{cc} = 0.625\text{sec}$.
    - Adversary needs approximately 10^7 sec $\approx$ 16 weeks and success probability is 2^{-80}.
- The effect of faster computer made the adversary's job harder.

Mathematical details

Negligible Function

A function $f : \mathbb{N} \to \mathbb{R}$ is called negligible if for all $c \in \mathbb{R}_{>0}$ there exists a $n_0 \in \mathbb{N}$ such that for all integers $n \geqslant n_0$, we have $|\, f(n) \,| < \frac{1}{n^c}$.

Mathematical details

Negligible Function

A function $f : \mathbb{N} \to \mathbb{R}$ is called negligible if for all $c \in \mathbb{R}_{>0}$ there exists a $n_0 \in \mathbb{N}$ such that for all integers $n \geqslant n_0$, we have $| f(n) | < \frac{1}{n^c}$.

Theorem

A function $f : \mathbb{N} \to \mathbb{R}$ is negligible if and only if for all $c > 0$, we have

$$\lim_{n \to \infty} f(n) n^c = 0.$$

Mathematical details

Negligible Function

A function $f : \mathbb{N} \to \mathbb{R}$ is called negligible if for all $c \in \mathbb{R}_{>0}$ there exists a $n_0 \in \mathbb{N}$ such that for all integers $n \geqslant n_0$, we have $| f(n) | < \frac{1}{n^c}$.

Theorem

A function $f : \mathbb{N} \to \mathbb{R}$ is negligible if and only if for all $c > 0$, we have

$$\lim_{n \to \infty} f(n) n^c = 0.$$

Negligible Function (Alternate Definition)

A function $f : \mathbb{N} \to \mathbb{R}$ is called negligible if for all polynomial $p(n) > 0$ there exists a $n_0 \in \mathbb{N}$ such that or all integers $n \geqslant n_0$, we have $| f(n) | < \frac{1}{p(n)}$.

Mathematical details

Examples:

- Some negligible functions: 2^{-n}, $2^{-\sqrt{n}}$, $n^{-\log n}$.

Mathematical details

Examples:

- Some negligible functions: 2^{-n}, $2^{-\sqrt{n}}$, $n^{-\log n}$.
  - $2^{-n} \leqslant 10^{-6}$ when $n \geqslant 20$, that is $2^{20} = 1048576$.
  - $2^{-\sqrt{n}} \leqslant 10^{-6}$ when $n \geqslant 400$, that is $2^{\sqrt{400}} = 1048576$.
  - $n^{-\log n} \leqslant 10^{-6}$ when $n \geqslant 32$, that is $32^5 = 33554432$.

Mathematical details

Examples:

- Some negligible functions: 2^{-n}, $2^{-\sqrt{n}}$, $n^{-\log n}$.
  - $2^{-n} \leqslant 10^{-6}$ when $n \geqslant 20$, that is $2^{20} = 1048576$.
  - $2^{-\sqrt{n}} \leqslant 10^{-6}$ when $n \geqslant 400$, that is $2^{\sqrt{400}} = 1048576$.
  - $n^{-\log n} \leqslant 10^{-6}$ when $n \geqslant 32$, that is $32^5 = 33554432$.
- Does $n^{-\log n}$ approach zero more quickly than $2^{-\sqrt{n}}$?

Mathematical details

Examples:

- Some negligible functions: 2^{-n}, $2^{-\sqrt{n}}$, $n^{-\log n}$.
  - $2^{-n} \leqslant 10^{-6}$ when $n \geqslant 20$, that is $2^{20} = 1048576$.
  - $2^{-\sqrt{n}} \leqslant 10^{-6}$ when $n \geqslant 400$, that is $2^{\sqrt{400}} = 1048576$.
  - $n^{-\log n} \leqslant 10^{-6}$ when $n \geqslant 32$, that is $32^5 = 33554432$.
- Does $n^{-\log n}$ approach zero more quickly than $2^{-\sqrt{n}}$?
  - It seems so.
  - In reality, $2^{-\sqrt{n}} < n^{-\log n}$ for all $n \geqslant 65536$.

Mathematical details

Super-poly Function

A function $f : \mathbb{N} \to \mathbb{R}$ is called super-poly if $\dfrac{1}{f}$ is negligible function.

Mathematical details

Super-poly Function

A function $f : \mathbb{N} \longrightarrow \mathbb{R}$ is called super-poly if $\dfrac{1}{f}$ is negligible function.

Poly-Bounded Function

A function $f : \mathbb{N} \longrightarrow \mathbb{R}$ is poly-bounded if there exists $c, d \in \mathbb{R}_{\geqslant 0}$ such that for all $n \in \mathbb{N}$, we have

$$| f(n) | < n^c + d.$$

Super-poly Function

A function $f : \mathbb{N} \longrightarrow \mathbb{R}$ is called super-poly if $\frac{1}{f}$ is negligible function.

Poly-Bounded Function

A function $f : \mathbb{N} \longrightarrow \mathbb{R}$ is poly-bounded if there exists $c, d \in \mathbb{R}_{\geqslant 0}$ such that for all $n \in \mathbb{N}$, we have

$$| f(n) | < n^c + d.$$

- If f is a poly-bounded function, then $\frac{1}{f}$ is definitely not a negligible function.

Mathematical details

Super-poly Function

A function $f : \mathbb{N} \rightarrow \mathbb{R}$ is called super-poly if $\frac{1}{f}$ is negligible function.

Poly-Bounded Function

A function $f : \mathbb{N} \rightarrow \mathbb{R}$ is poly-bounded if there exists $c, d \in \mathbb{R}_{\geqslant 0}$ such that for all $n \in \mathbb{N}$, we have

$$| f(n) | < n^c + d.$$

- If f is a poly-bounded function, then $\frac{1}{f}$ is definitely not a negligible function.

Example

$$\begin{aligned} f \quad &: \quad \mathbb{N} \rightarrow \mathbb{R} \\ f \quad &= \quad \begin{cases} \frac{1}{n}, & n \text{ is even} \\ 2^{-n}, & n \text{ is odd} \end{cases} \end{aligned}$$

- f is neither negligible nor $1/f$ is poly-bounded or super-poly.

Mathematical details

Proposition

Let ϵ_1 and ϵ_2 be negligible functions, and p_1 and p_2 be poly-bounded functions.

Proposition

Let ϵ_1 and ϵ_2 be negligible functions, and p_1 and p_2 be poly-bounded functions. Then,

1. The function $\epsilon_3(n) = \epsilon_1(n) + \epsilon_2(n)$ is negligible.

Mathematical details

Proposition

Let ϵ_1 and ϵ_2 be negligible functions, and p_1 and p_2 be poly-bounded functions. Then,

1. The function $\epsilon_3(n) = \epsilon_1(n) + \epsilon_2(n)$ is negligible.
2. For any positive polynomial $p(n)$, the function $\epsilon_4(n) = p(n) \cdot \epsilon_1(n)$ is negligible.

Proposition

Let ϵ_1 and ϵ_2 be negligible functions, and p_1 and p_2 be poly-bounded functions. Then,

1. The function $\epsilon_3(n) = \epsilon_1(n) + \epsilon_2(n)$ is negligible.
2. For any positive polynomial $p(n)$, the function $\epsilon_4(n) = p(n) \cdot \epsilon_1(n)$ is negligible.
3. The function $p_3(n) = p_1(n) + p_2(n)$ and $p_4(n) = p_1(n) \cdot p_2(n)$ are poly-bounded.

Proposition

Let ϵ_1 and ϵ_2 be negligible functions, and p_1 and p_2 be poly-bounded functions. Then,

1. The function $\epsilon_3(n) = \epsilon_1(n) + \epsilon_2(n)$ is negligible.
2. For any positive polynomial $p(n)$, the function $\epsilon_4(n) = p(n) \cdot \epsilon_1(n)$ is negligible.
3. The function $p_3(n) = p_1(n) + p_2(n)$ and $p_4(n) = p_1(n) \cdot p_2(n)$ are poly-bounded.

Proof

Exercise.

Efficient Algorithm

Let A be an algorithm (possibly probabilistic) that takes as input a security parameter $n \in \mathbb{N}$, as well as other parameters encoded as a bit string $x \in \{0,1\}^{\leqslant p(n)}$ for some fixed polynomial p. We call A an efficient algorithm if there exist a poly-bounded function t and a negligible function ϵ such that for all $n \in \mathbb{N}$, and all $x \in \{0,1\}^{\leqslant p(n)}$, the probability that the running time of A on input (n, x) exceeds $t(n)$ is at most $\epsilon(n)$.

Asymptotic Security: Definition

Definition (Asymptotic Security)

A scheme is *secure* if for every PPT adversary $\mathcal{A}$ carrying out an attack of some formally specified type, the probability that $\mathcal{A}$ *succeeds* in the attack is $\epsilon(n)$.

Asymptotic Security: Definition

Definition (Asymptotic Security)

A scheme is *secure* if for every PPT adversary $\mathcal{A}$ carrying out an attack of some formally specified type, the probability that $\mathcal{A}$ *succeeds* in the attack is $\epsilon(n)$.

Note: Nothing is guaranteed for values $n \leq n_0$.

Computational Cipher

A Computational Cipher with security parameter $n \in \mathbb{N}$ is a tuple of PPT algorithms $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ such that:

- $\mathcal{M}$: A finite message space, like all finite length strings from $\{0,1\}^*$.
- $\mathcal{C}$: A finite ciphertext space, like all finite length strings from $\{0,1\}^*$.
- $\mathcal{K}$: A finite key space, like $\{0,1\}^n$.

Computational Cipher

Computational Cipher

A Computational Cipher with security parameter $n \in \mathbb{N}$ is a tuple of PPT algorithms $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ such that:

- $\mathcal{M}$: A finite message space, like all finite length strings from $\{0,1\}^*$.
- $\mathcal{C}$: A finite ciphertext space, like all finite length strings from $\{0,1\}^*$.
- $\mathcal{K}$: A finite key space, like $\{0,1\}^n$.
- $\mathcal{G}$: $\mathcal{G}$ is a PPT algorithm.

$$k \xleftarrow{R} \mathcal{G}(1^n).$$

We assume w.l.o.g. that any key $k \leftarrow \mathcal{G}(1^n)$ satisfies $| k | \geqslant n$.

Computational Cipher

Computational Cipher

A Computational Cipher with security parameter $n \in \mathbb{N}$ is a tuple of PPT algorithms $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ such that:

- $\mathcal{M}$: A finite message space, like all finite length strings from $\{0,1\}^*$.
- $\mathcal{C}$: A finite ciphertext space, like all finite length strings from $\{0,1\}^*$.
- $\mathcal{K}$: A finite key space, like $\{0,1\}^n$.
- $\mathcal{G}$: $\mathcal{G}$ is a PPT algorithm.

  $$k \xleftarrow{R} \mathcal{G}(1^n).$$

  We assume w.l.o.g. that any key $k \leftarrow \mathcal{G}(1^n)$ satisfies $|\, k \,| \geqslant n$.
- $\mathcal{E}$: $\mathcal{E}$ is a PPT algorithm.

  $$c \xleftarrow{R} \mathcal{E}(k,m), \text{ where } m \in \mathcal{M}.$$

Computational Cipher

A Computational Cipher with security parameter $n \in \mathbb{N}$ is a tuple of PPT algorithms $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ such that:

- $\mathcal{M}$: A finite message space, like all finite length strings from $\{0,1\}^*$.
- $\mathcal{C}$: A finite ciphertext space, like all finite length strings from $\{0,1\}^*$.
- $\mathcal{K}$: A finite key space, like $\{0,1\}^n$.
- $\mathcal{G}$: $\mathcal{G}$ is a PPT algorithm.

  $$k \xleftarrow{R} \mathcal{G}(1^n).$$

  We assume w.l.o.g. that any key $k \leftarrow \mathcal{G}(1^n)$ satisfies $|\, k \,| \geqslant n$.
- $\mathcal{E}$: $\mathcal{E}$ is a PPT algorithm.

  $$c \xleftarrow{R} \mathcal{E}(k, m), \text{ where } m \in \mathcal{M}.$$

- $\mathcal{D}$: We assume that $\mathcal{D}$ is deterministic.

  $$m := \mathcal{D}(k, c), \text{ and returns } \perp \text{ on error.}$$

Private-key Encryption Scheme: Definition

Correctness

For all $k \in \mathcal{K}$ and $m \in \mathcal{M}$, we have

$$c \xleftarrow{R} \mathcal{E}(m, k); \;\; m' := \mathcal{D}(k, c),$$

and $m = m'$ with probability 1.

The Basic Assumptions

- Basic notion of security: Security against a *ciphertext-only attack* where the adversary observes only *a single ciphertext*.

The Basic Assumptions

- Basic notion of security: Security against a *ciphertext-only attack* where the adversary observes only *a single ciphertext*.
- Equivalently, security when a given key is used to encrypt just a *single message*.

The Basic Assumptions

- Basic notion of security: Security against a *ciphertext-only attack* where the adversary observes only *a single ciphertext*.
- Equivalently, security when a given key is used to encrypt just a *single message*.
- **Assumptions about the adversary's capabilities:**
  - It only eavesdrops.
  - It runs in polynomial time.

The Basic Assumptions

- Basic notion of security: Security against a *ciphertext-only attack* where the adversary observes only *a single ciphertext*.
- Equivalently, security when a given key is used to encrypt just a *single message*.
- **Assumptions about the adversary's capabilities:**
  - It only eavesdrops.
  - It runs in polynomial time.
- No assumptions are made about the adversary's *strategy* in trying to decipher the ciphertext it observes.

Indistinguishability

- Defined as an attack game played between two parties,
  - a challenger, and
  - an adversary.

Indistinguishability

- Defined as an attack game played between two parties,
  - a challenger, and
  - an adversary.
- In general, the challenger follows a very simple, fixed protocol.

Indistinguishability

- Defined as an attack game played between two parties,
  - a challenger, and
  - an adversary.
- In general, the challenger follows a very simple, fixed protocol.
- However, an adversary may follow an arbitrary (but still efficient) protocol.

- Defined as an attack game played between two parties,
  - a challenger, and
  - an adversary.
- In general, the challenger follows a very simple, fixed protocol.
- However, an adversary may follow an arbitrary (but still efficient) protocol.
- The challenger and the adversary send messages back and forth to each other, as specified by their protocols.

Indistinguishability

- Defined as an attack game played between two parties,
  - a challenger, and
  - an adversary.
- In general, the challenger follows a very simple, fixed protocol.
- However, an adversary may follow an arbitrary (but still efficient) protocol.
- The challenger and the adversary send messages back and forth to each other, as specified by their protocols.
- At the end of the game, the adversary may output some value.

Indistinguishability

- Defined as an attack game played between two parties,
  - a challenger, and
  - an adversary.
- In general, the challenger follows a very simple, fixed protocol.
- However, an adversary may follow an arbitrary (but still efficient) protocol.
- The challenger and the adversary send messages back and forth to each other, as specified by their protocols.
- At the end of the game, the adversary may output some value.
- The attack game also defines a probability space, and this in turn defines the adversary's advantage, which is determined by the probability of one or more events.

Indistinguishability

- Defined as an attack game played between two parties,
  - a challenger, and
  - an adversary.
- In general, the challenger follows a very simple, fixed protocol.
- However, an adversary may follow an arbitrary (but still efficient) protocol.
- The challenger and the adversary send messages back and forth to each other, as specified by their protocols.
- At the end of the game, the adversary may output some value.
- The attack game also defines a probability space, and this in turn defines the adversary's advantage, which is determined by the probability of one or more events.
- In Indistinguishability game, there are two alternative experiments.
  - In both experiments, the adversary follows the same protocol.
  - The challenger behaves differently in each experiment.

Indistinguishability Game

For a given cipher $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$, and for a given adversary $\mathcal{A}$, we define two experiments: Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

- The adversary computes $m_0, m_1 \in \mathcal{M}$, of the same length, and sends them to the challenger.

Indistinguishability Game

For a given cipher $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$, and for a given adversary $\mathcal{A}$, we define two experiments: Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

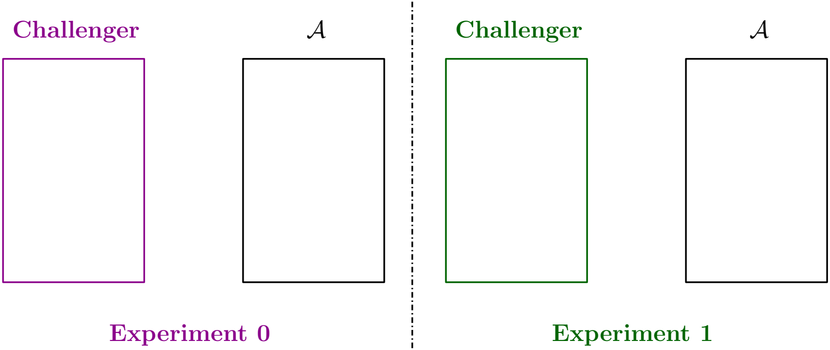
- The adversary computes $m_0, m_1 \in \mathcal{M}$, of the same length, and sends them to the challenger.
- The challenger computes $k \xleftarrow{R} \mathcal{K}$; $c \xleftarrow{R} \mathcal{E}(k, m_b)$, and sends c to the adversary.

Indistinguishability Game

For a given cipher $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$, and for a given adversary $\mathcal{A}$, we define two experiments: Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

- The adversary computes $m_0, m_1 \in \mathcal{M}$, of the same length, and sends them to the challenger.
- The challenger computes $k \xleftarrow{R} \mathcal{K}$; $c \xleftarrow{R} \mathcal{E}(k, m_b)$, and sends c to the adversary.
- The adversary outputs a bit $\hat{b} \in \{0, 1\}$.

Experiment b, where $b \in \{0, 1\}$

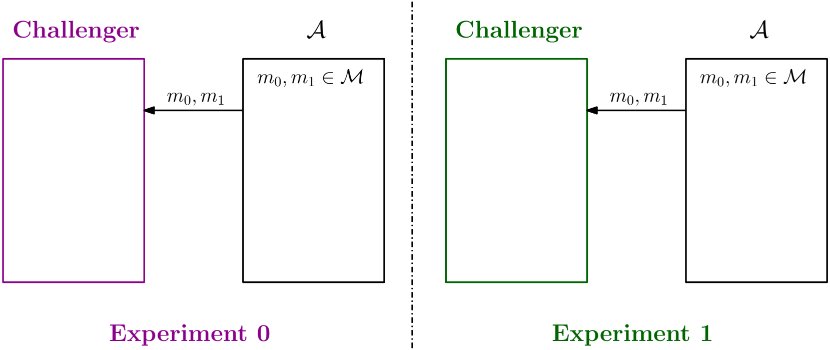


Experiment b, where $b \in \{0, 1\}$

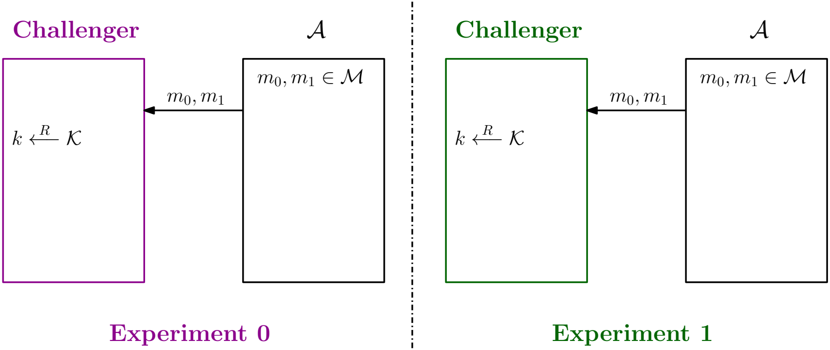
Challenger	$\mathcal{A}$	**Challenger**	$\mathcal{A}$
	$m_0, m_1 \in \mathcal{M}$		$m_0, m_1 \in \mathcal{M}$

Experiment 0 | **Experiment 1**

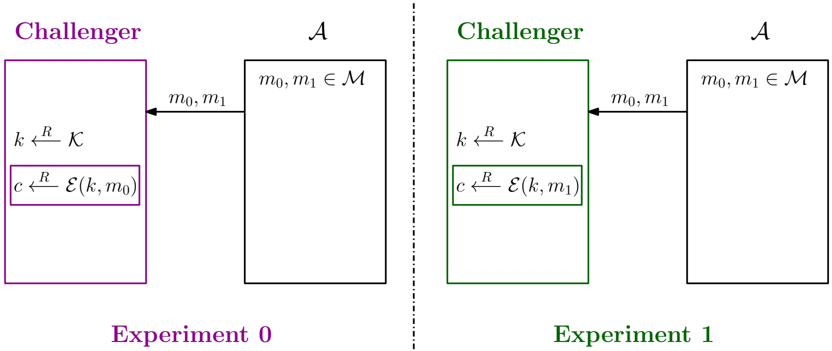
Experiment b, where $b \in \{0, 1\}$



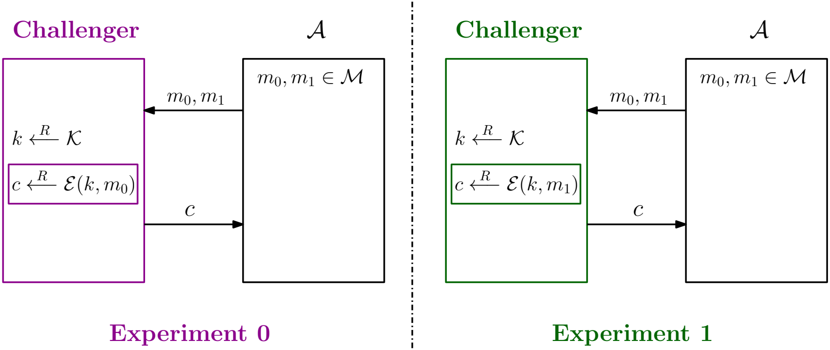
Experiment b, where $b \in \{0, 1\}$



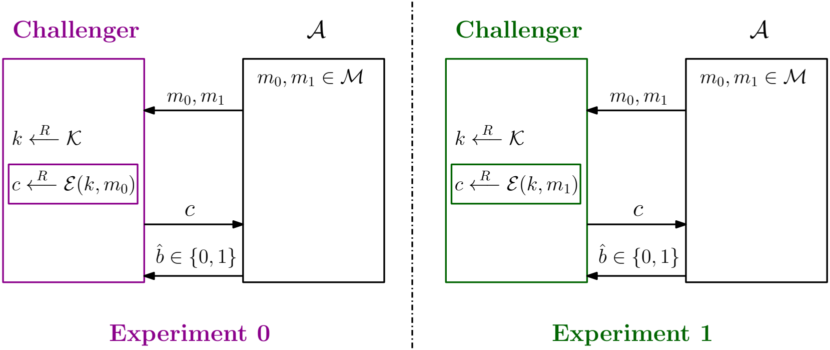
Experiment b, where $b \in \{0, 1\}$



Experiment b, where $b \in \{0, 1\}$



Experiment b, where $b \in \{0, 1\}$



Indistinguishability Advantage

For $b = 0, 1$, let W_b be the event that $\mathcal{A}$ outputs 1 in Experiment b. We define the advantage of $\mathcal{A}$ in the Indistinguishability attack with respect to $\mathfrak{E}$ as

$$\mathsf{INDadv}[\mathcal{A}, \mathfrak{E}] = | \Pr[W_0] - \Pr[W_1] | .$$

Indistinguishability

Indistinguishability Advantage

For $b = 0, 1$, let W_b be the event that $\mathcal{A}$ outputs 1 in Experiment b. We define the advantage of $\mathcal{A}$ in the Indistinguishability attack with respect to $\mathfrak{E}$ as

$$\mathsf{INDadv}[\mathcal{A}, \mathfrak{E}] = \mid \Pr[W_0] - \Pr[W_1] \mid .$$

Source of Randomness

Events W_0 and W_1 are defined with respect to the probability space determined by:

- the random choice of k,

Indistinguishability

Indistinguishability Advantage

For $b = 0, 1$, let W_b be the event that $\mathcal{A}$ outputs 1 in Experiment b. We define the advantage of $\mathcal{A}$ in the Indistinguishability attack with respect to $\mathfrak{E}$ as

$$\mathsf{INDadv}[\mathcal{A}, \mathfrak{E}] = \mid \Pr[W_0] - \Pr[W_1] \mid .$$

Source of Randomness

Events W_0 and W_1 are defined with respect to the probability space determined by:

- the random choice of k,
- random choices made (if any) by the encryption algorithm,

Indistinguishability

Indistinguishability Advantage

For $b = 0, 1$, let W_b be the event that $\mathcal{A}$ outputs 1 in Experiment b. We define the advantage of $\mathcal{A}$ in the Indistinguishability attack with respect to $\mathfrak{E}$ as

$$\mathsf{INDadv}[\mathcal{A}, \mathfrak{E}] = |\Pr[W_0] - \Pr[W_1]|.$$

Source of Randomness

Events W_0 and W_1 are defined with respect to the probability space determined by:

- the random choice of k,
- random choices made (if any) by the encryption algorithm, and
- the random choices made (if any) by the adversary.

Indistinguishability

Indistinguishability Advantage

For $b = 0, 1$, let W_b be the event that $\mathcal{A}$ outputs 1 in Experiment b. We define the advantage of $\mathcal{A}$ in the Indistinguishability attack with respect to $\mathfrak{E}$ as

$$\mathsf{INDadv}[\mathcal{A}, \mathfrak{E}] = |\Pr[W_0] - \Pr[W_1]|.$$

Source of Randomness

Events W_0 and W_1 are defined with respect to the probability space determined by:

- the random choice of k,
- random choices made (if any) by the encryption algorithm, and
- the random choices made (if any) by the adversary.

$$0 \leqslant \mathsf{INDadv}[\mathcal{A}, \mathfrak{E}] \leqslant 1.$$

Indistinguishable

We say $\mathfrak{E}$ is (computationally) indistinguishable in the presence of an eavesdropper if for all PPT adversaries $\mathcal{A}$ there exists a negligible function ϵ such that

$$\mathsf{INDadv}[\mathcal{A}, \mathfrak{E}] \leqslant \epsilon(n).$$

Indistinguishable

We say $\mathfrak{E}$ is (computationally) indistinguishable in the presence of an eavesdropper if for all PPT adversaries $\mathcal{A}$ there exists a negligible function ϵ such that

$$\mathsf{INDadv}[\mathcal{A}, \mathfrak{E}] \leqslant \epsilon(n).$$

- We achieve perfect security if $\mathsf{INDadv}[\mathcal{A}, \mathfrak{E}] = 0$.

Indistinguishability

m_0 and m_1 are of the same length

- An encryption scheme should be able to encrypt arbitrary-length messages.

Indistinguishability

m_0 and m_1 are of the same length

- An encryption scheme should be able to encrypt arbitrary-length messages.
- This restriction captures the notion that an encryption of a message is allowed to leak the length of the message (but nothing else).

Message Recovery Attack

For a given cipher $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$, and for a given adversary $\mathcal{A}$, the attack game proceeds as follows:

- The challenger computes $m \xleftarrow{R} \mathcal{M}$, $k \xleftarrow{R} \mathcal{K}$ and $c \xleftarrow{R} \mathcal{E}(k, m)$.
- The challenger then sends c to adversary.

Message Recovery Attack

For a given cipher $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$, and for a given adversary $\mathcal{A}$, the attack game proceeds as follows:

- The challenger computes $m \xleftarrow{R} \mathcal{M}$, $k \xleftarrow{R} \mathcal{K}$ and $c \xleftarrow{R} \mathcal{E}(k, m)$.
- The challenger then sends c to adversary.
- The adversary outputs a message $\hat{m} \in \mathcal{M}$.

Message Recovery Attack

For a given cipher $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$, and for a given adversary $\mathcal{A}$, the attack game proceeds as follows:

- The challenger computes $m \xleftarrow{R} \mathcal{M}$, $k \xleftarrow{R} \mathcal{K}$ and $c \xleftarrow{R} \mathcal{E}(k, m)$.
- The challenger then sends c to adversary.
- The adversary outputs a message $\hat{m} \in \mathcal{M}$.

Let W be the event that $\hat{m} = m$ and $\mathcal{A}$ wins.

Message Recovery Attack

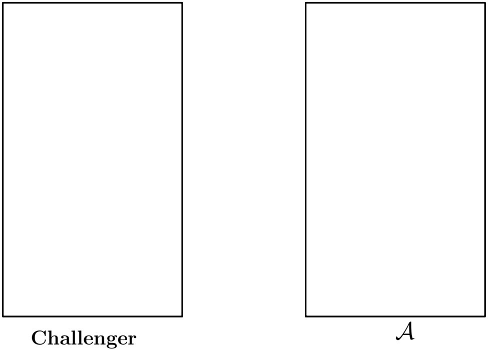
For a given cipher $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$, and for a given adversary $\mathcal{A}$, the attack game proceeds as follows:

- The challenger computes $m \xleftarrow{R} \mathcal{M}$, $k \xleftarrow{R} \mathcal{K}$ and $c \xleftarrow{R} \mathcal{E}(k, m)$.
- The challenger then sends c to adversary.
- The adversary outputs a message $\hat{m} \in \mathcal{M}$.

Let W be the event that $\hat{m} = m$ and $\mathcal{A}$ wins. We define the advantage of $\mathcal{A}$ in message recovery attack with respect to $\mathfrak{E}$ as

$$\mathsf{MRadv}[\mathcal{A}, \mathfrak{E}] = \left| \Pr[W] - \frac{1}{|\mathcal{M}|} \right|.$$

Message Recovery Attack



Message Recovery Attack

$m \xleftarrow{R} \mathcal{M}$

Challenger

$\mathcal{A}$

Message Recovery Attack

$m \xleftarrow{R} \mathcal{M}$

$k \xleftarrow{R} \mathcal{K}$

Challenger

$\mathcal{A}$

Message Recovery Attack

$m \xleftarrow{R} \mathcal{M}$

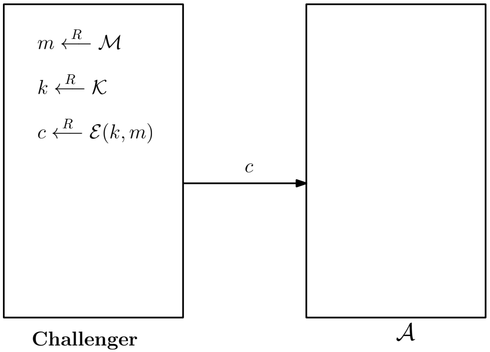
$k \xleftarrow{R} \mathcal{K}$

$c \xleftarrow{R} \mathcal{E}(k, m)$

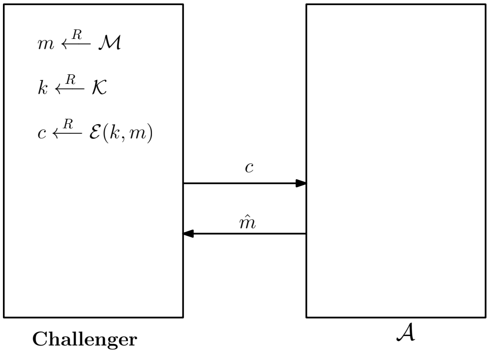
Challenger

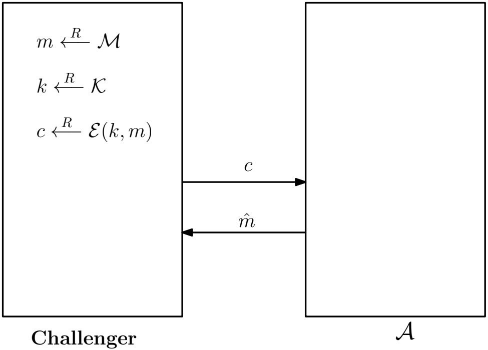
$\mathcal{A}$

Message Recovery Attack



Message Recovery Attack





$\mathcal{A}$ **wins if** $\hat{m} = m$

Message Recovery Attack

Security against Message Recovery

A cipher $\mathfrak{E}$ is secure against message recovery if for all PPT adversaries $\mathcal{A}$ there exists a negligible function ϵ such that

$$\mathsf{MRadv}[\mathcal{A}, \mathfrak{E}] \leqslant \epsilon(n).$$

Theorem

Let $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$. If $\mathfrak{E}$ is secure against indistinguishability attack, then $\mathfrak{E}$ is secure against message recovery attack.

Message Recovery Attack

Theorem

Let $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$. If $\mathfrak{E}$ is secure against indistinguishability attack, then $\mathfrak{E}$ is secure against message recovery attack.

Proof

- Let $\mathfrak{E}$ is secure against indistinguishability attack.

Message Recovery Attack

Theorem

Let $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$. If $\mathfrak{E}$ is secure against indistinguishability attack, then $\mathfrak{E}$ is secure against message recovery attack.

Proof

- Let $\mathfrak{E}$ is secure against indistinguishability attack.
- Goal: For all PPT adversaries, the message recovery advantage of $\mathcal{A}$ is negligible.

Theorem

Let $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$. If $\mathfrak{E}$ is secure against indistinguishability attack, then $\mathfrak{E}$ is secure against message recovery attack.

Proof

- Let $\mathfrak{E}$ is secure against indistinguishability attack.
- Goal: For all PPT adversaries, the message recovery advantage of $\mathcal{A}$ is negligible.
- Let there be an efficient adversary $\mathcal{A}$ that can win the message recovery game with probability p.
- In the message recovery game, $\Pr[W] = p$.

Theorem

Let $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$. If $\mathfrak{E}$ is secure against indistinguishability attack, then $\mathfrak{E}$ is secure against message recovery attack.

Proof

- Let $\mathfrak{E}$ is secure against indistinguishability attack.
- Goal: For all PPT adversaries, the message recovery advantage of $\mathcal{A}$ is negligible.
- Let there be an efficient adversary $\mathcal{A}$ that can win the message recovery game with probability p.
- In the message recovery game, $\Pr[W] = p$.
- Then we have,

$$\mathsf{MRadv}[\mathcal{A}, \mathfrak{E}] = \left| p - \frac{1}{|\mathcal{M}|} \right|.$$

Proof

- Construct an efficient adversary $\mathcal{B}$ of the indistinguishability attack.
    - $\mathcal{B}$ uses $\mathcal{A}$ as a sub-routine.

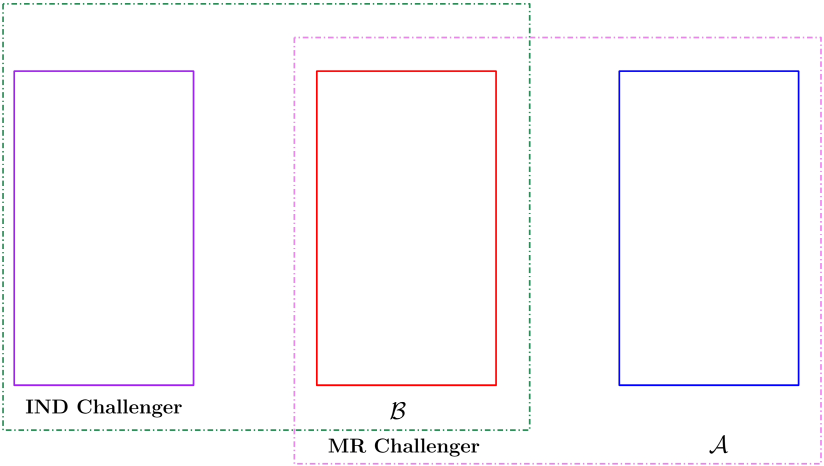
Proof

- Construct an efficient adversary $\mathcal{B}$ of the indistinguishability attack.
  - $\mathcal{B}$ uses $\mathcal{A}$ as a sub-routine.
  - $\mathcal{A}$ is a black-box to $\mathcal{B}$.

Proof

- Construct an efficient adversary $\mathcal{B}$ of the indistinguishability attack.
  - $\mathcal{B}$ uses $\mathcal{A}$ as a sub-routine.
  - $\mathcal{A}$ is a black-box to $\mathcal{B}$.
  - $\mathcal{B}$ behaves as the challenger to $\mathcal{A}$.

Message Recovery Attack



Proof

- Indistinguishability Experiment b:

  1. $\mathcal{B}$ computes $m_0, m_1 \xleftarrow{R} \mathcal{M}$, and sends them to indistinguishability challenger.

Proof

- Indistinguishability Experiment b:

  1. $\mathcal{B}$ computes $m_0, m_1 \xleftarrow{R} \mathcal{M}$, and sends them to indistinguishability challenger.
  2. Indistinguishability challenger computes $k \xleftarrow{R} \mathcal{K}$ and $c \xleftarrow{R} \mathcal{E}(k, m_b)$.

Message Recovery Attack

Proof

- Indistinguishability Experiment b:

1. $\mathcal{B}$ computes $m_0, m_1 \xleftarrow{R} \mathcal{M}$, and sends them to indistinguishability challenger.
2. Indistinguishability challenger computes $k \xleftarrow{R} \mathcal{K}$ and $c \xleftarrow{R} \mathcal{E}(k, m_b)$.
3. Indistinguishability challenger sends c to $\mathcal{B}$.
4. $\mathcal{B}$ then forwards c to $\mathcal{A}$.

Proof

- Indistinguishability Experiment b:
    1. $\mathcal{B}$ computes $m_0, m_1 \xleftarrow{R} \mathcal{M}$, and sends them to indistinguishability challenger.
    2. Indistinguishability challenger computes $k \xleftarrow{R} \mathcal{K}$ and $c \xleftarrow{R} \mathcal{E}(k, m_b)$.
    3. Indistinguishability challenger sends c to $\mathcal{B}$.
    4. $\mathcal{B}$ then forwards c to $\mathcal{A}$.
    5. $\mathcal{A}$ returns $\hat{m}$ to $\mathcal{B}$.

Proof

- Indistinguishability Experiment b:

  1. $\mathcal{B}$ computes $m_0, m_1 \xleftarrow{R} \mathcal{M}$, and sends them to indistinguishability challenger.
  2. Indistinguishability challenger computes $k \xleftarrow{R} \mathcal{K}$ and $c \xleftarrow{R} \mathcal{E}(k, m_b)$.
  3. Indistinguishability challenger sends c to $\mathcal{B}$.
  4. $\mathcal{B}$ then forwards c to $\mathcal{A}$.
  5. $\mathcal{A}$ returns $\hat{m}$ to $\mathcal{B}$.
  6. $\mathcal{B}$ outputs $\hat{b} = 1$ if $\hat{m} = m_1$, else $\hat{b} = 0$ in the indistinguishability game.

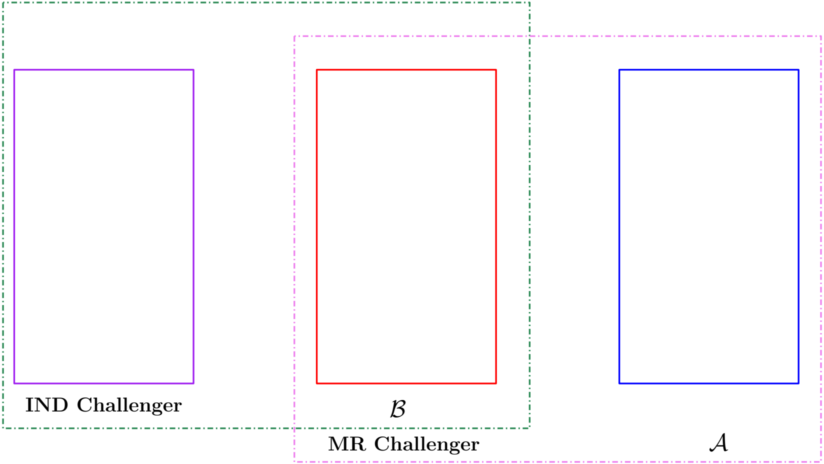


Figure: Experiment b

Message Recovery Attack

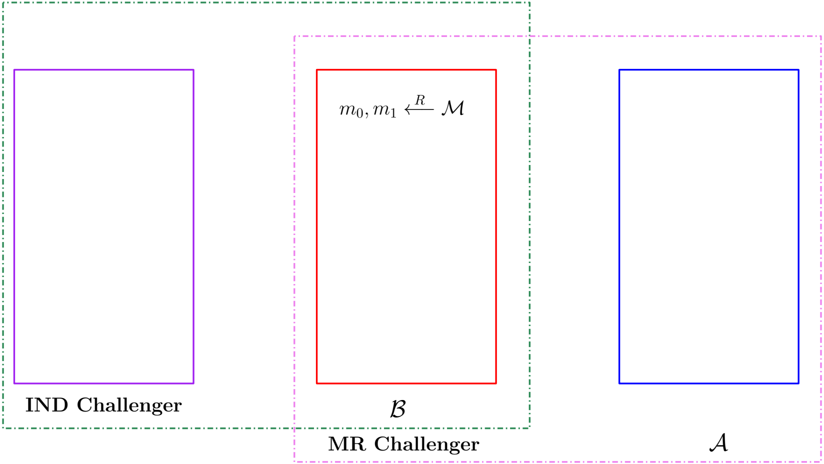


Figure: Experiment b

Message Recovery Attack

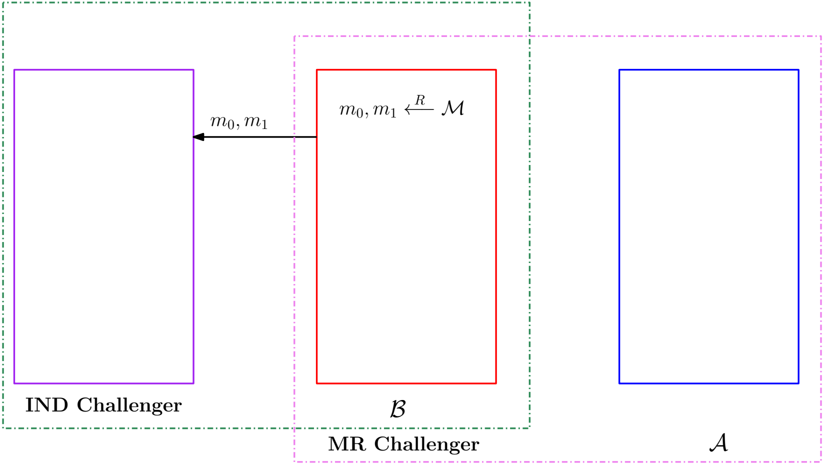


Figure: Experiment b

Message Recovery Attack

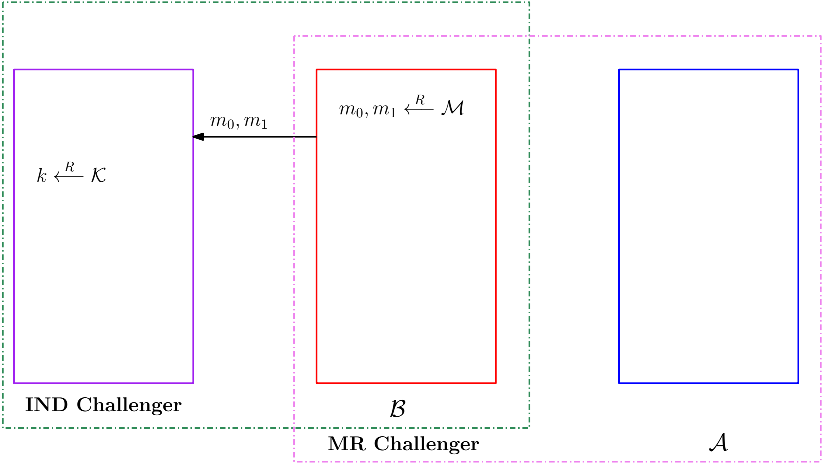


Figure: Experiment b

Message Recovery Attack

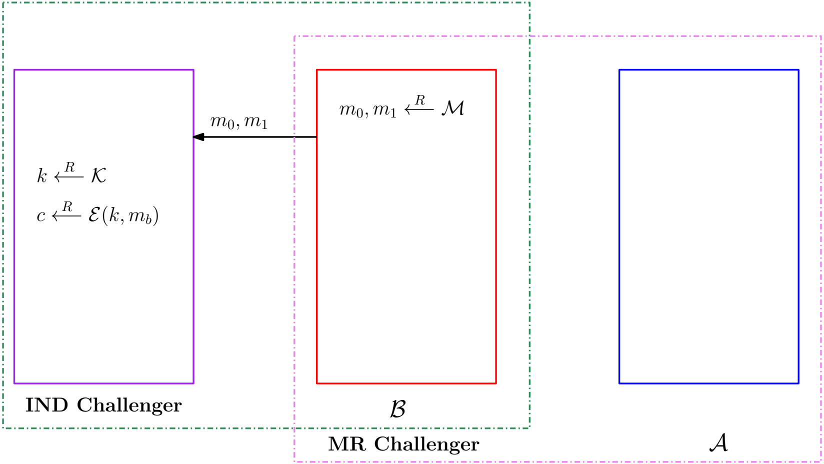


Figure: Experiment b

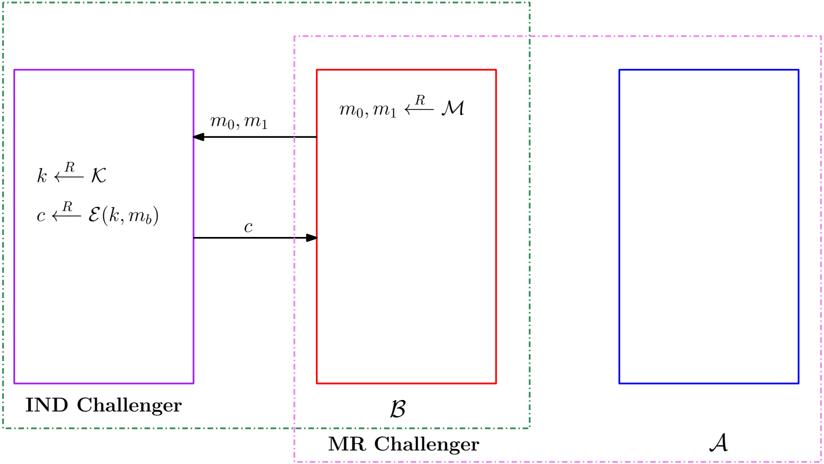


Figure: Experiment b

Message Recovery Attack

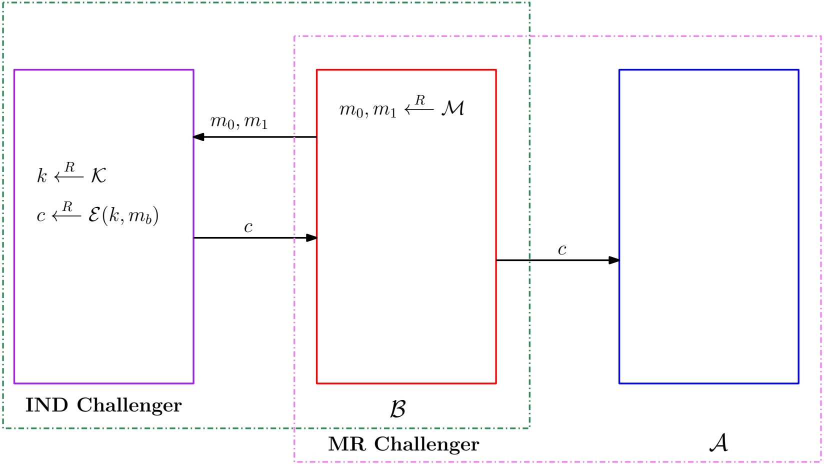


Figure: Experiment b

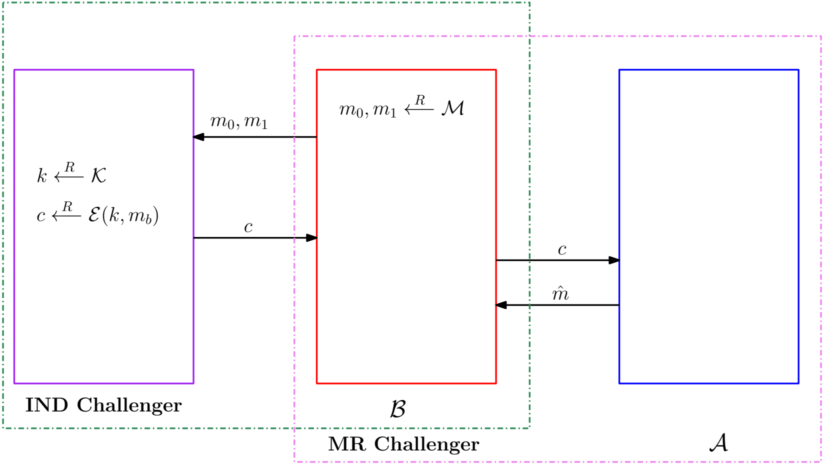


Figure: Experiment b

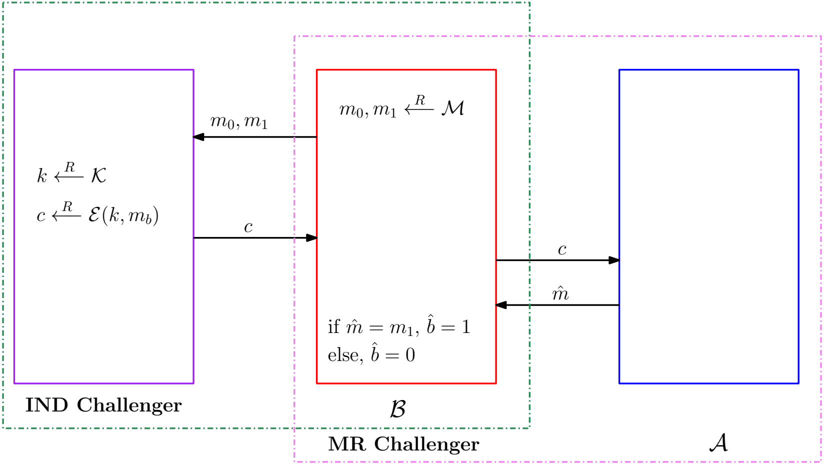


Figure: Experiment b

Message Recovery Attack

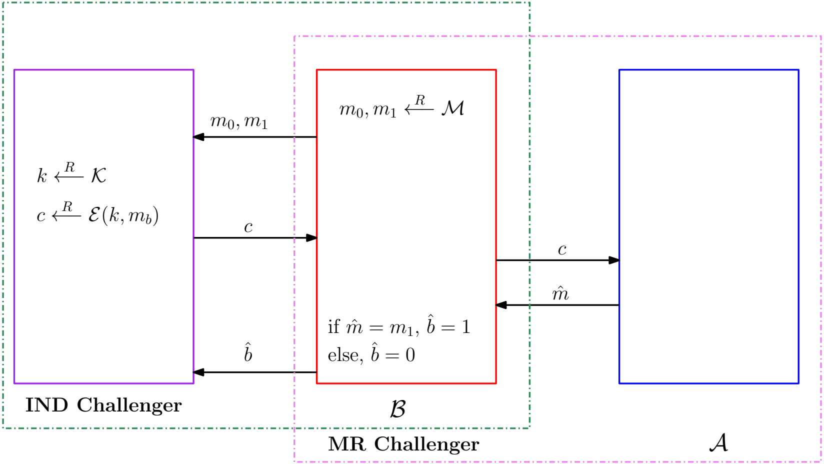


Figure: Experiment b

Message Recovery Attack

Proof

- For $b = 0, 1$, let $p_b = \Pr[W_b]$ for adversary $\mathcal{B}$ in indistinguishability game.

Proof

- For $b = 0, 1$, let $p_b = \Pr[W_b]$ for adversary $\mathcal{B}$ in indistinguishability game.
- The Advantage of $\mathcal{B}$ in indistinguishability game is

$$\mathsf{INDadv}[\mathcal{B}, \mathfrak{E}] = |\Pr[W_0] - \Pr[W_1]| = |p_0 - p_1|.$$

Proof

- For $b = 0, 1$, let $p_b = \Pr[W_b]$ for adversary $\mathcal{B}$ in indistinguishability game.
- The Advantage of $\mathcal{B}$ in indistinguishability game is

$$\mathsf{INDadv}[\mathcal{B}, \mathfrak{E}] = \mid \Pr[W_0] - \Pr[W_1] \mid = \mid p_0 - p_1 \mid .$$

- As $\mathfrak{E}$ is secure in the indistinguishability game, then there exists negligible function ϵ such that

$$\mathsf{INDadv}[\mathcal{B}, \mathfrak{E}] \leqslant \epsilon(n).$$

Message Recovery Attack

Proof

- For $b = 0, 1$, let $p_b = \Pr[W_b]$ for adversary $\mathcal{B}$ in indistinguishability game.
- The Advantage of $\mathcal{B}$ in indistinguishability game is

$$\text{INDadv}[\mathcal{B}, \mathfrak{E}] = \mid \Pr[W_0] - \Pr[W_1] \mid = \mid p_0 - p_1 \mid .$$

- As $\mathfrak{E}$ is secure in the indistinguishability game, then there exists negligible function ϵ such that

$$\text{INDadv}[\mathcal{B}, \mathfrak{E}] \leqslant \epsilon(n).$$

- We have to show:

$$\text{MRadv}[\mathcal{A}, \mathfrak{E}] \leqslant \text{INDadv}[\mathcal{B}, \mathfrak{E}]$$

Proof

$$p_1 = \Pr[W_1]$$

Message Recovery Attack

Proof

$$
\begin{aligned}
p_1 &= \Pr[W_1] \\
&= \Pr[\mathcal{B} \text{ outputs } \hat{b} = 1 \text{ in Experiment } 1]
\end{aligned}
$$

Message Recovery Attack

Proof

$$
\begin{aligned}
p_1 &= \Pr[W_1] \\
&= \Pr[\mathcal{B} \text{ outputs } \hat{b} = 1 \text{ in Experiment } 1] \\
&= \Pr[\mathcal{A} \text{ outputs } \hat{m} = m_1 \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_1)]
\end{aligned}
$$

Message Recovery Attack

Proof

$$
\begin{aligned}
p_1 &= \Pr[W_1] \\
&= \Pr[\mathcal{B} \text{ outputs } \hat{b} = 1 \text{ in Experiment } 1] \\
&= \Pr[\mathcal{A} \text{ outputs } \hat{m} = m_1 \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_1)] \\
&= \Pr[\mathcal{A} \text{ wins in Message Recovery attack}]
\end{aligned}
$$

Proof

$$
\begin{aligned}
p_1 &= \Pr[W_1] \\
&= \Pr[\mathcal{B} \text{ outputs } \hat{b} = 1 \text{ in Experiment 1}] \\
&= \Pr[\mathcal{A} \text{ outputs } \hat{m} = m_1 \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_1)] \\
&= \Pr[\mathcal{A} \text{ wins in Message Recovery attack}] \\
&= p.
\end{aligned}
$$

Message Recovery Attack

Proof

- Let $\mathbf{m}_0$ and $\mathbf{m}_1$ be two random variables by takes value from $\mathcal{M}$ uniformly at random.
    - $\mathbf{m}_0$ and $\mathbf{m}_1$ are independent.

Proof

- Let $\mathbf{m}_0$ and $\mathbf{m}_1$ be two random variables by takes value from $\mathcal{M}$ uniformly at random.
  - $\mathbf{m}_0$ and $\mathbf{m}_1$ are independent.
- Let $\mathbf{k}$ be a random variable that chooses the key from $\mathcal{K}$ uniformly at random.
  - $\mathbf{k}$ is independent of $\mathbf{m}_0$ and $\mathbf{m}_1$.

Proof

- Let $\mathbf{m}_0$ and $\mathbf{m}_1$ be two random variables by takes value from $\mathcal{M}$ uniformly at random.
  - $\mathbf{m}_0$ and $\mathbf{m}_1$ are independent.
- Let $\mathbf{k}$ be a random variable that chooses the key from $\mathcal{K}$ uniformly at random.
  - $\mathbf{k}$ is independent of $\mathbf{m}_0$ and $\mathbf{m}_1$.
- Define random variable $\mathbf{c} = \mathcal{E}(\mathbf{k}, \mathbf{m}_0)$.
- $\mathbf{k}, \mathbf{m}_0$ are independent of $\mathbf{m}_1 \Rightarrow \mathbf{c}$ is independent of $\mathbf{m}_1$.

Proof

- Let $\mathbf{m}_0$ and $\mathbf{m}_1$ be two random variables by takes value from $\mathcal{M}$ uniformly at random.
  - $\mathbf{m}_0$ and $\mathbf{m}_1$ are independent.
- Let $\mathbf{k}$ be a random variable that chooses the key from $\mathcal{K}$ uniformly at random.
  - $\mathbf{k}$ is independent of $\mathbf{m}_0$ and $\mathbf{m}_1$.
- Define random variable $\mathbf{c} = \mathcal{E}(\mathbf{k}, \mathbf{m}_0)$.
- $\mathbf{k}, \mathbf{m}_0$ are independent of $\mathbf{m}_1 \Rightarrow \mathbf{c}$ is independent of $\mathbf{m}_1$.
- Let $\hat{\mathbf{m}}$ be a random variable that defines the output of $\mathcal{A}$.

Message Recovery Attack

Proof

- Let $\mathbf{m}_0$ and $\mathbf{m}_1$ be two random variables by takes value from $\mathcal{M}$ uniformly at random.
  - $\mathbf{m}_0$ and $\mathbf{m}_1$ are independent.
- Let $\mathbf{k}$ be a random variable that chooses the key from $\mathcal{K}$ uniformly at random.
  - $\mathbf{k}$ is independent of $\mathbf{m}_0$ and $\mathbf{m}_1$.
- Define random variable $\mathbf{c} = \mathcal{E}(\mathbf{k}, \mathbf{m}_0)$.
- $\mathbf{k}, \mathbf{m}_0$ are independent of $\mathbf{m}_1 \Rightarrow \mathbf{c}$ is independent of $\mathbf{m}_1$.
- Let $\hat{\mathbf{m}}$ be a random variable that defines the output of $\mathcal{A}$.
  - Output of $\mathcal{A}$ depends on $\mathbf{c}$ (and may be on some internal randomness of $\mathcal{A}$).

Proof

- Let $\mathbf{m}_0$ and $\mathbf{m}_1$ be two random variables by takes value from $\mathcal{M}$ uniformly at random.
    - $\mathbf{m}_0$ and $\mathbf{m}_1$ are independent.
- Let $\mathbf{k}$ be a random variable that chooses the key from $\mathcal{K}$ uniformly at random.
    - $\mathbf{k}$ is independent of $\mathbf{m}_0$ and $\mathbf{m}_1$.
- Define random variable $\mathbf{c} = \mathcal{E}(\mathbf{k}, \mathbf{m}_0)$.
- $\mathbf{k}, \mathbf{m}_0$ are independent of $\mathbf{m}_1 \Rightarrow \mathbf{c}$ is independent of $\mathbf{m}_1$.
- Let $\hat{\mathbf{m}}$ be a random variable that defines the output of $\mathcal{A}$.
    - Output of $\mathcal{A}$ depends on $\mathbf{c}$ (and may be on some internal randomness of $\mathcal{A}$).
    - Therefore, $\hat{\mathbf{m}}$ is independent of $\mathbf{m}_1$.

$$p_0 = \Pr[W_0]$$

Message Recovery Attack

$$
\begin{aligned}
p_0 &= \Pr[W_0] \\
&= \Pr[\mathcal{B} \text{ outputs } \hat{b} = 1 \text{ in Experiment } 0]
\end{aligned}
$$

$$\begin{aligned}
p_0 &= \Pr[W_0] \\
&= \Pr[\mathcal{B} \text{ outputs } \hat{b} = 1 \text{ in Experiment } 0] \\
&= \Pr[\mathcal{A} \text{ outputs } m_1 \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_0)]
\end{aligned}$$

$$
\begin{aligned}
p_0 &= \Pr[W_0] \\
&= \Pr[\mathcal{B} \text{ outputs } \hat{b} = 1 \text{ in Experiment } 0] \\
&= \Pr[\mathcal{A} \text{ outputs } m_1 \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_0)] \\
&= \sum_{m \in \mathcal{M}} \Pr[\mathbf{m}_1 = m \wedge \hat{\mathbf{m}} = m]
\end{aligned}
$$

Message Recovery Attack

$$
\begin{aligned}
p_0 &= \Pr[W_0] \\
&= \Pr[\mathcal{B} \text{ outputs } \hat{b} = 1 \text{ in Experiment } 0] \\
&= \Pr[\mathcal{A} \text{ outputs } m_1 \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_0)] \\
&= \sum_{m \in \mathcal{M}} \Pr[\mathbf{m}_1 = m \wedge \hat{\mathbf{m}} = m] \\
&= \sum_{m \in \mathcal{M}} \Pr[\mathbf{m}_1 = m] \Pr[\hat{\mathbf{m}} = m], \text{ by independence of } \mathbf{m}_1 \text{ and } \hat{\mathbf{m}}
\end{aligned}
$$

Message Recovery Attack

$$
\begin{aligned}
p_0 &= \Pr[W_0] \\
&= \Pr[\mathcal{B} \text{ outputs } \hat{b} = 1 \text{ in Experiment } 0] \\
&= \Pr[\mathcal{A} \text{ outputs } m_1 \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_0)] \\
&= \sum_{m \in \mathcal{M}} \Pr[\mathbf{m}_1 = m \wedge \hat{\mathbf{m}} = m] \\
&= \sum_{m \in \mathcal{M}} \Pr[\mathbf{m}_1 = m] \Pr[\hat{\mathbf{m}} = m], \text{ by independence of } \mathbf{m}_1 \text{ and } \hat{\mathbf{m}} \\
&= \sum_{m \in \mathcal{M}} \frac{1}{|\mathcal{M}|} \Pr[\hat{\mathbf{m}} = m], \ \mathbf{m}_1 \text{ is uniform over } \mathcal{M}
\end{aligned}
$$

$$
\begin{aligned}
p_0 &= \Pr[W_0] \\
&= \Pr[\mathcal{B} \text{ outputs } \hat{b} = 1 \text{ in Experiment 0}] \\
&= \Pr[\mathcal{A} \text{ outputs } m_1 \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_0)] \\
&= \sum_{m \in \mathcal{M}} \Pr[\mathbf{m}_1 = m \wedge \hat{\mathbf{m}} = m] \\
&= \sum_{m \in \mathcal{M}} \Pr[\mathbf{m}_1 = m] \Pr[\hat{\mathbf{m}} = m], \text{ by independence of } \mathbf{m}_1 \text{ and } \hat{\mathbf{m}} \\
&= \sum_{m \in \mathcal{M}} \frac{1}{|\mathcal{M}|} \Pr[\hat{\mathbf{m}} = m], \ \mathbf{m}_1 \text{ is uniform over } \mathcal{M} \\
&= \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \Pr[\hat{\mathbf{m}} = m]
\end{aligned}
$$

$$
\begin{aligned}
p_0 &= \Pr[W_0] \\
&= \Pr[\mathcal{B} \text{ outputs } \hat{b} = 1 \text{ in Experiment } 0] \\
&= \Pr[\mathcal{A} \text{ outputs } m_1 \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_0)] \\
&= \sum_{m \in \mathcal{M}} \Pr[\mathbf{m}_1 = m \wedge \hat{\mathbf{m}} = m] \\
&= \sum_{m \in \mathcal{M}} \Pr[\mathbf{m}_1 = m] \Pr[\hat{\mathbf{m}} = m], \text{ by independence of } \mathbf{m}_1 \text{ and } \hat{\mathbf{m}} \\
&= \sum_{m \in \mathcal{M}} \frac{1}{|\mathcal{M}|} \Pr[\hat{\mathbf{m}} = m], \ \mathbf{m}_1 \text{ is uniform over } \mathcal{M} \\
&= \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \Pr[\hat{\mathbf{m}} = m] \\
&= \frac{1}{|\mathcal{M}|}, \text{ As } \sum_{m \in \mathcal{M}} \Pr[\hat{\mathbf{m}} = m] = 1.
\end{aligned}
$$

Proof

$$\mathsf{INDadv}[\mathcal{B},\mathfrak{E}] \quad = \quad |\Pr[W_0] - \Pr[W_1]|$$

Message Recovery Attack

Proof

$$\begin{aligned} \mathsf{INDadv}[\mathcal{B},\mathfrak{E}] &= |\Pr[W_0]-\Pr[W_1]| \\ &= |p_0-p_1| \end{aligned}$$

Message Recovery Attack

Proof

$$
\begin{aligned}
\mathsf{INDadv}[\mathcal{B},\mathfrak{E}] &= |\Pr[W_0]-\Pr[W_1]| \\
&= |p_0-p_1| \\
&= \left| \frac{1}{|\mathcal{M}|}-p \right|
\end{aligned}
$$

Proof

$$
\begin{aligned}
\mathsf{INDadv}[\mathcal{B},\mathfrak{E}] &= \mid \Pr[W_0] - \Pr[W_1] \mid \\
&= \mid p_0 - p_1 \mid \\
&= \left| \frac{1}{\mid \mathcal{M} \mid} - p \right| \\
&= \left| p - \frac{1}{\mid \mathcal{M} \mid} \right|
\end{aligned}
$$

Proof

$$
\begin{aligned}
\mathsf{INDadv}[\mathcal{B},\mathfrak{E}] &= \mid \Pr[W_0] - \Pr[W_1] \mid \\
&= \mid p_0 - p_1 \mid \\
&= \left| \frac{1}{\mid \mathcal{M} \mid} - p \right| \\
&= \left| p - \frac{1}{\mid \mathcal{M} \mid} \right| \\
&= \mathsf{MRadv}[\mathcal{A},\mathfrak{E}]
\end{aligned}
$$

Proof

$$\begin{aligned}
\mathsf{INDadv}[\mathcal{B},\mathfrak{E}] &= |\Pr[W_0]-\Pr[W_1]| \\
&= |p_0 - p_1| \\
&= \left| \frac{1}{|\mathcal{M}|} - p \right| \\
&= \left| p - \frac{1}{|\mathcal{M}|} \right| \\
&= \mathsf{MRadv}[\mathcal{A},\mathfrak{E}]
\end{aligned}$$

Therefore, we have

$$\mathsf{MRadv}[\mathcal{A},\mathfrak{E}] = \mathsf{INDadv}[\mathcal{B},\mathfrak{E}].$$

Parity Prediction

- If an encryption scheme is secure
  - It should be hard to recover the whole message,

Parity Prediction

- If an encryption scheme is secure
  - It should be hard to recover the whole message,
  - Not only that, it should be hard to compute any partial information about the message, like parity or a particular bit of the message.

Parity Prediction

- If an encryption scheme is secure
  - It should be hard to recover the whole message,
  - Not only that, it should be hard to compute any partial information about the message, like parity or a particular bit of the message.

Parity

- Let m be a message.
- Parity = XOR of all bits of m

Parity Prediction

- If an encryption scheme is secure
  - It should be hard to recover the whole message,
  - Not only that, it should be hard to compute any partial information about the message, like parity or a particular bit of the message.

Parity

- Let m be a message.
- Parity = XOR of all bits of m

$$\mathsf{Parity}(m) = \begin{cases} 1, & \text{if number of 1's in } m \text{ is odd} \\ 0, & \text{if number of 1's in } m \text{ is even} \end{cases}$$

Parity Prediction

For a given cipher $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$, and for a given adversary $\mathcal{A}$, the attack game proceeds as follows:

- The challenger computes $m \xleftarrow{R} \mathcal{M}$, $k \xleftarrow{R} \mathcal{K}$ and $c \xleftarrow{R} \mathcal{E}(k, m)$.
- The challenger then sends c to adversary.

Parity Prediction

For a given cipher $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$, and for a given adversary $\mathcal{A}$, the attack game proceeds as follows:

- The challenger computes $m \xleftarrow{R} \mathcal{M}$, $k \xleftarrow{R} \mathcal{K}$ and $c \xleftarrow{R} \mathcal{E}(k, m)$.
- The challenger then sends c to adversary.
- The adversary outputs a message $\hat{b} \in \{0, 1\}$.

Parity Prediction

For a given cipher $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$, and for a given adversary $\mathcal{A}$, the attack game proceeds as follows:

- The challenger computes $m \xleftarrow{R} \mathcal{M}$, $k \xleftarrow{R} \mathcal{K}$ and $c \xleftarrow{R} \mathcal{E}(k, m)$.
- The challenger then sends c to adversary.
- The adversary outputs a message $\hat{b} \in \{0, 1\}$.

Let W be the event that $\hat{b} = \mathsf{Parity}(m)$ and $\mathcal{A}$ wins.

Parity Prediction

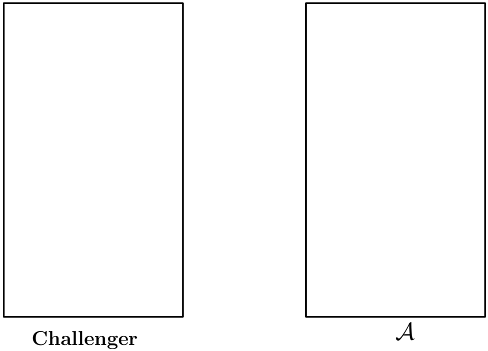
For a given cipher $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$, and for a given adversary $\mathcal{A}$, the attack game proceeds as follows:

- The challenger computes $m \xleftarrow{R} \mathcal{M}$, $k \xleftarrow{R} \mathcal{K}$ and $c \xleftarrow{R} \mathcal{E}(k, m)$.
- The challenger then sends c to adversary.
- The adversary outputs a message $\hat{b} \in \{0, 1\}$.

Let W be the event that $\hat{b} = \mathsf{Parity}(m)$ and $\mathcal{A}$ wins. We define the advantage of $\mathcal{A}$ in parity prediction attack with respect to $\mathfrak{E}$ as

$$\mathsf{PARITYadv}[\mathcal{A}, \mathfrak{E}] = \left| \Pr[W] - \frac{1}{2} \right|.$$

Parity Prediction



Parity Prediction

$m \xleftarrow{R} \mathcal{M}$

Challenger

$\mathcal{A}$

Parity Prediction

$m \xleftarrow{R} \mathcal{M}$

$k \xleftarrow{R} \mathcal{K}$

Challenger

$\mathcal{A}$

Parity Prediction

$m \xleftarrow{R} \mathcal{M}$

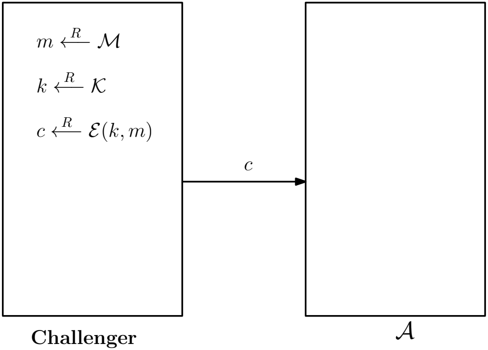
$k \xleftarrow{R} \mathcal{K}$

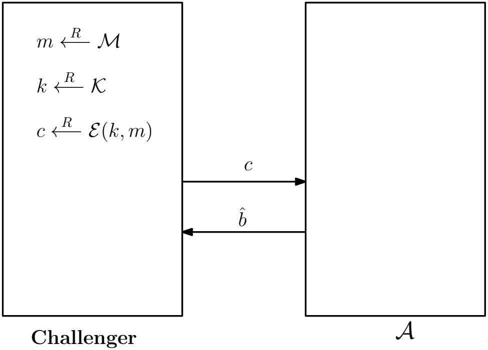
$c \xleftarrow{R} \mathcal{E}(k, m)$

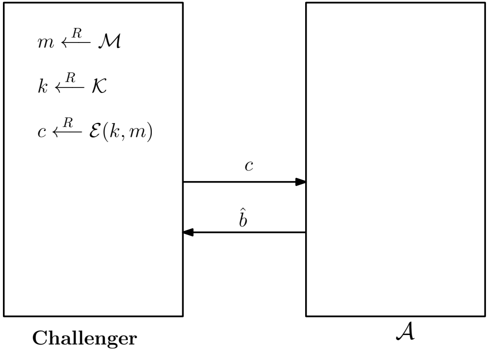
Challenger

$\mathcal{A}$

Parity Prediction




$m \xleftarrow{R} \mathcal{M}$
$k \xleftarrow{R} \mathcal{K}$
$c \xleftarrow{R} \mathcal{E}(k, m)$
c
$\hat{b}$
Challenger
$\mathcal{A}$



$\mathcal{A}$ **wins if** $\hat{b} = \mathsf{Parity}(m)$

Security against Parity Prediction

A cipher $\mathfrak{E}$ is secure against parity prediction attack if for all PPT adversaries $\mathcal{A}$, there exists a negligible function ϵ such that

$$\mathsf{PARITYadv}[\mathcal{A}, \mathfrak{E}] \leqslant \epsilon(n).$$

Parity Prediction

Theorem

Let $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$, $\mathcal{M} = \{0,1\}^L$ with security parameter $n \in \mathbb{N}$. If $\mathfrak{E}$ is secure against indistinguishability attack, then $\mathfrak{E}$ is secure against parity prediction attack.

Parity Prediction

Theorem

Let $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$, $\mathcal{M} = \{0,1\}^L$ with security parameter $n \in \mathbb{N}$. If $\mathfrak{E}$ is secure against indistinguishability attack, then $\mathfrak{E}$ is secure against parity prediction attack.

Proof

- Let $\mathfrak{E}$ is secure against indistinguishability attack.

Parity Prediction

Theorem

Let $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$, $\mathcal{M} = \{0,1\}^L$ with security parameter $n \in \mathbb{N}$. If $\mathfrak{E}$ is secure against indistinguishability attack, then $\mathfrak{E}$ is secure against parity prediction attack.

Proof

- Let $\mathfrak{E}$ is secure against indistinguishability attack.
- Goal: For all PPT adversaries, the parity prediction advantage of $\mathcal{A}$ is negligible.

Parity Prediction

Theorem

Let $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$, $\mathcal{M} = \{0,1\}^L$ with security parameter $n \in \mathbb{N}$. If $\mathfrak{E}$ is secure against indistinguishability attack, then $\mathfrak{E}$ is secure against parity prediction attack.

Proof

- Let $\mathfrak{E}$ is secure against indistinguishability attack.
- Goal: For all PPT adversaries, the parity prediction advantage of $\mathcal{A}$ is negligible.
- Let there be an efficient adversary $\mathcal{A}$ that can win the parity prediction game with probability p.
- In the parity prediction game, $\Pr[W] = p$.

Parity Prediction

Theorem

Let $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$, $\mathcal{M} = \{0,1\}^L$ with security parameter $n \in \mathbb{N}$. If $\mathfrak{E}$ is secure against indistinguishability attack, then $\mathfrak{E}$ is secure against parity prediction attack.

Proof

- Let $\mathfrak{E}$ is secure against indistinguishability attack.
- Goal: For all PPT adversaries, the parity prediction advantage of $\mathcal{A}$ is negligible.
- Let there be an efficient adversary $\mathcal{A}$ that can win the parity prediction game with probability p.
- In the parity prediction game, $\Pr[W] = p$.
- Then we have,

$$\mathsf{PARITYadv}[\mathcal{A}, \mathfrak{E}] = \left| \, p - \frac{1}{2} \, \right|.$$

Proof

- Construct an efficient adversary $\mathcal{B}$ of the indistinguishability attack.
  - $\mathcal{B}$ uses $\mathcal{A}$ as a sub-routine.

Parity Prediction

Proof

- Construct an efficient adversary $\mathcal{B}$ of the indistinguishability attack.
  - $\mathcal{B}$ uses $\mathcal{A}$ as a sub-routine.
  - $\mathcal{A}$ is a black-box to $\mathcal{B}$.

Proof

- Construct an efficient adversary $\mathcal{B}$ of the indistinguishability attack.
  - $\mathcal{B}$ uses $\mathcal{A}$ as a sub-routine.
  - $\mathcal{A}$ is a black-box to $\mathcal{B}$.
  - $\mathcal{B}$ behaves as the challenger to $\mathcal{A}$.

Parity Prediction

Proof

- Indistinguishability Experiment b:

  1. $\mathcal{B}$ computes $m_0 \xleftarrow{R} \mathcal{M}$

Proof

- Indistinguishability Experiment b:

  1. $\mathcal{B}$ computes $m_0 \xleftarrow{R} \mathcal{M}$ and sets $m_1 \longleftarrow m_0 \oplus (0^{L-1} \| 1)$, and sends m_0, m_1 to indistinguishability challenger.

Proof

- Indistinguishability Experiment b:

  1. $\mathcal{B}$ computes $m_0 \xleftarrow{R} \mathcal{M}$ and sets $m_1 \longleftarrow m_0 \oplus (0^{L-1} \| 1)$, and sends m_0, m_1 to indistinguishability challenger.
  2. Indistinguishability challenger computes $k \xleftarrow{R} \mathcal{K}$ and $c \xleftarrow{R} \mathcal{E}(k, m_b)$.

Proof

- Indistinguishability Experiment b:

1. $\mathcal{B}$ computes $m_0 \xleftarrow{R} \mathcal{M}$ and sets $m_1 \longleftarrow m_0 \oplus (0^{L-1} \| 1)$, and sends m_0, m_1 to indistinguishability challenger.
2. Indistinguishability challenger computes $k \xleftarrow{R} \mathcal{K}$ and $c \xleftarrow{R} \mathcal{E}(k, m_b)$.
3. Indistinguishability challenger sends c to $\mathcal{B}$.
4. $\mathcal{B}$ then forwards c to $\mathcal{A}$.

Proof

- Indistinguishability Experiment b:

1. $\mathcal{B}$ computes $m_0 \xleftarrow{R} \mathcal{M}$ and sets $m_1 \longleftarrow m_0 \oplus (0^{L-1} \| 1)$, and sends m_0, m_1 to indistinguishability challenger.
2. Indistinguishability challenger computes $k \xleftarrow{R} \mathcal{K}$ and $c \xleftarrow{R} \mathcal{E}(k, m_b)$.
3. Indistinguishability challenger sends c to $\mathcal{B}$.
4. $\mathcal{B}$ then forwards c to $\mathcal{A}$.
5. $\mathcal{A}$ returns $\hat{b}$ to $\mathcal{B}$.

Proof

- Indistinguishability Experiment b:

1. $\mathcal{B}$ computes $m_0 \xleftarrow{R} \mathcal{M}$ and sets $m_1 \longleftarrow m_0 \oplus (0^{L-1} \| 1)$, and sends m_0, m_1 to indistinguishability challenger.
2. Indistinguishability challenger computes $k \xleftarrow{R} \mathcal{K}$ and $c \xleftarrow{R} \mathcal{E}(k, m_b)$.
3. Indistinguishability challenger sends c to $\mathcal{B}$.
4. $\mathcal{B}$ then forwards c to $\mathcal{A}$.
5. $\mathcal{A}$ returns $\hat{b}$ to $\mathcal{B}$.
6. $\mathcal{B}$ outputs $\bar{b} = 1$ if $\hat{b} = \mathsf{Parity}(m_0)$, else $\bar{b} = 0$ in the indistinguishability game.

Parity Prediction

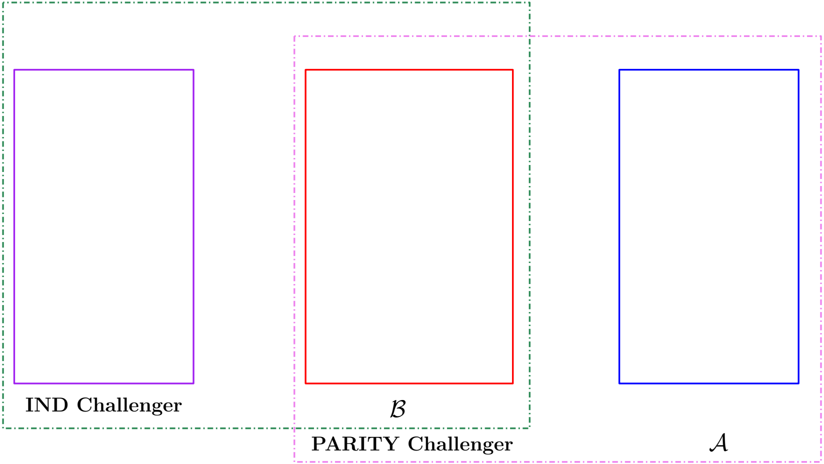


Figure: Experiment b

Parity Prediction

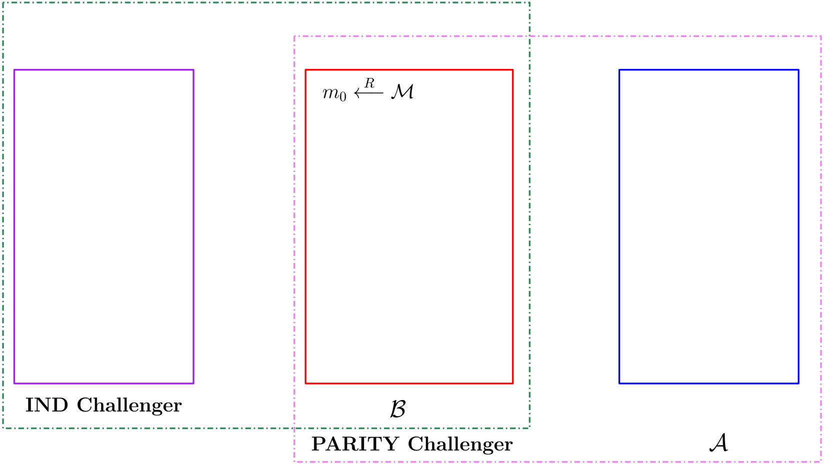


Figure: Experiment b

Parity Prediction

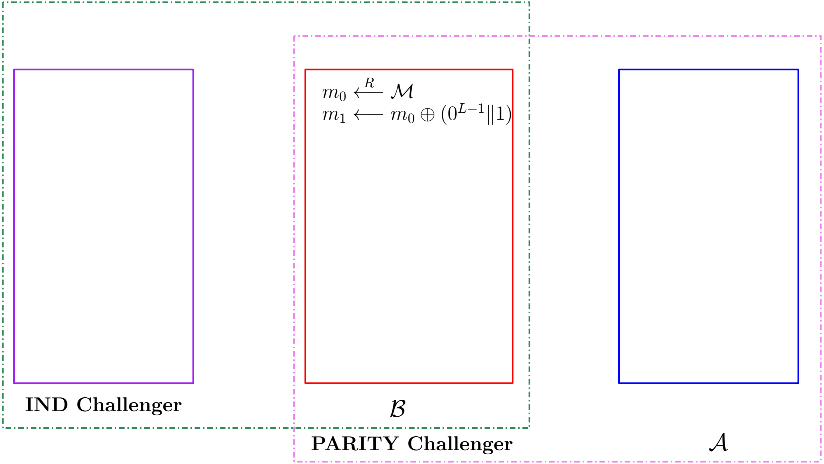


Figure: Experiment b

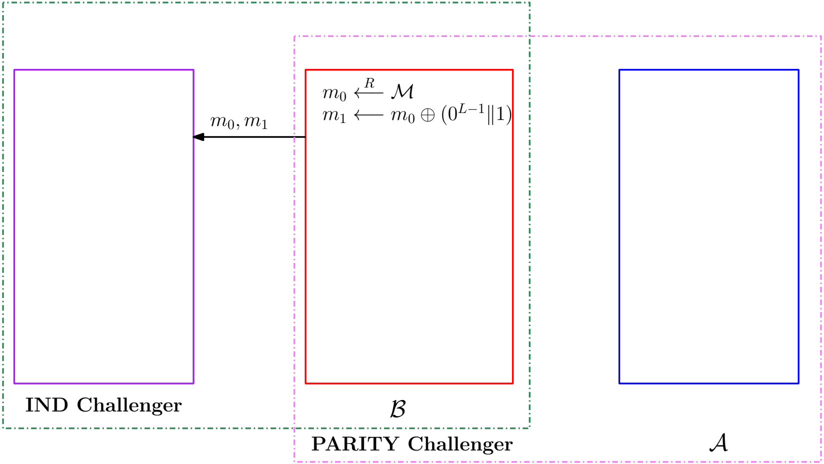


Figure: Experiment b

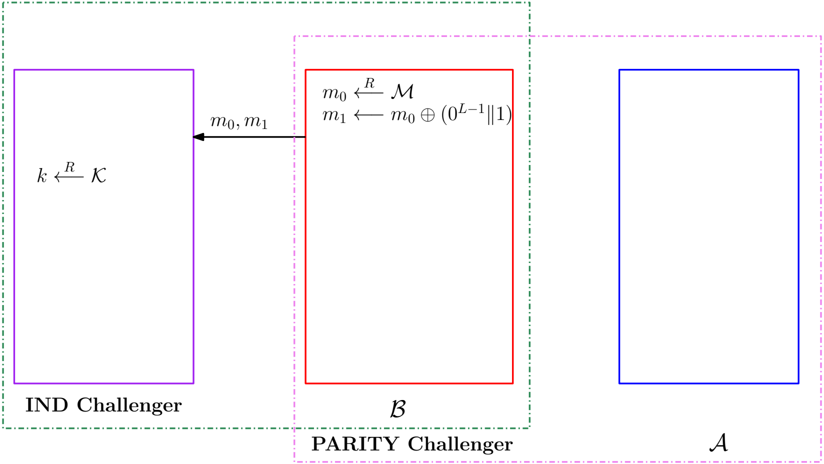


Figure: Experiment b

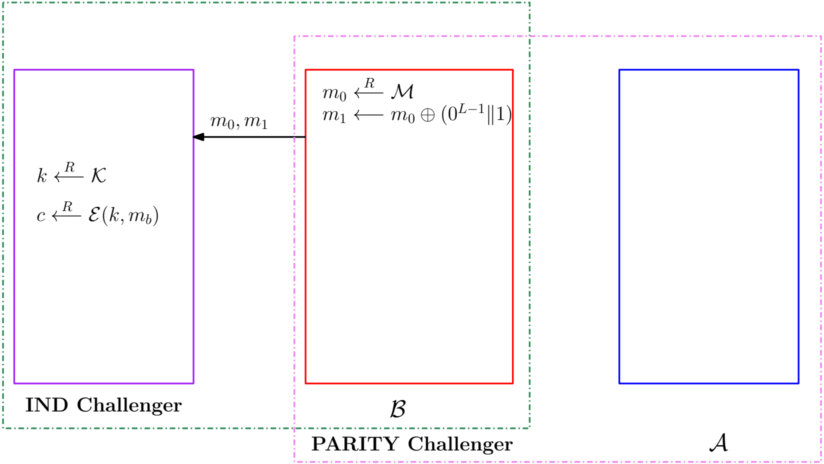


Figure: Experiment b

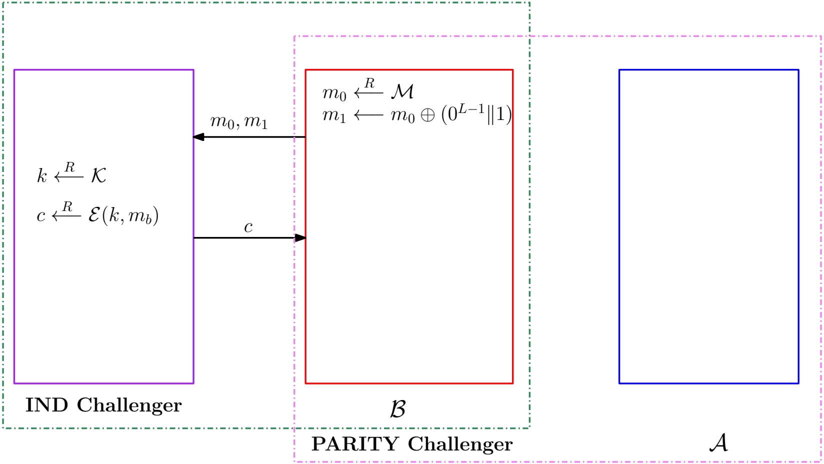


Figure: Experiment b

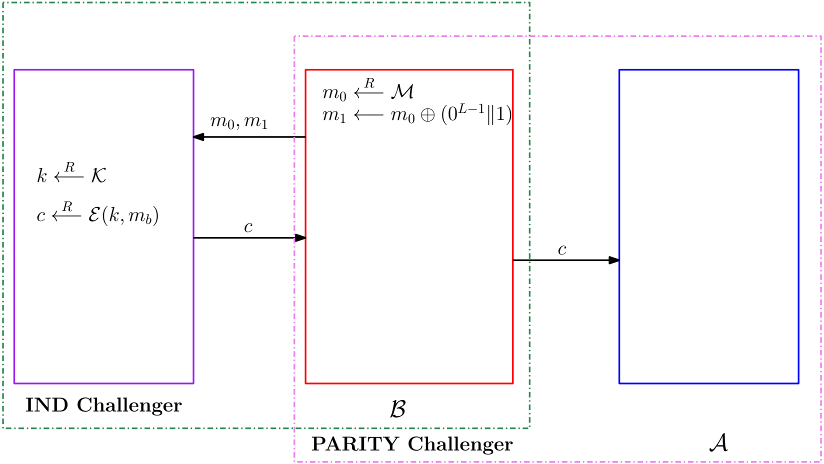


Figure: Experiment b

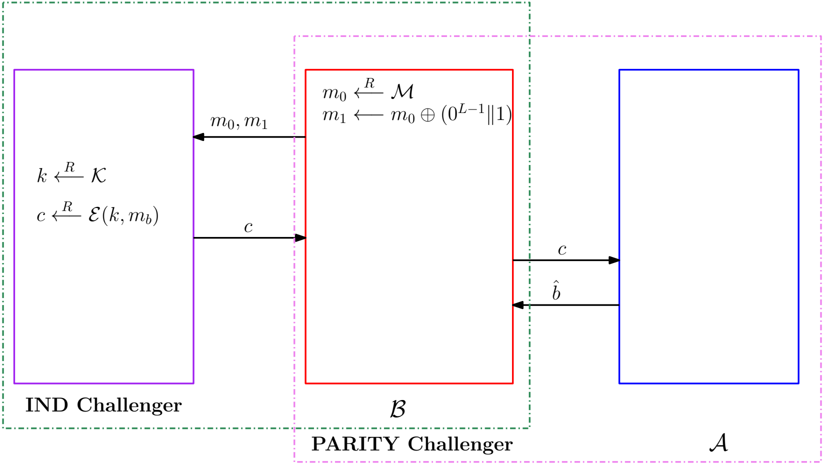


Figure: Experiment b

Parity Prediction

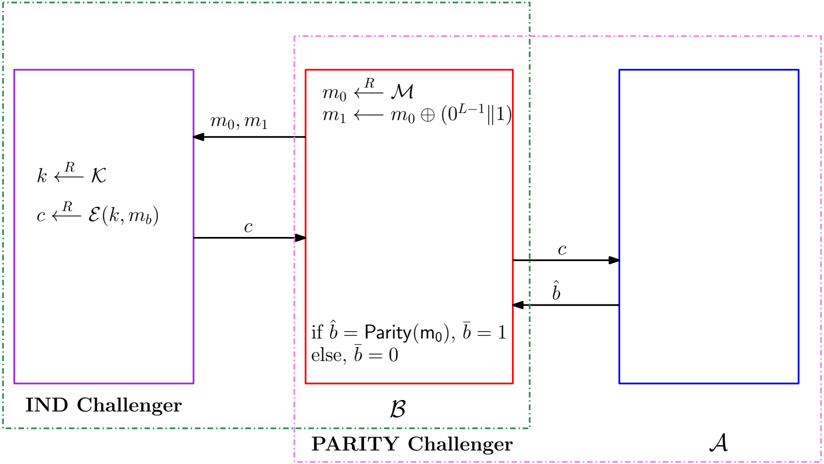


Figure: Experiment b

Parity Prediction

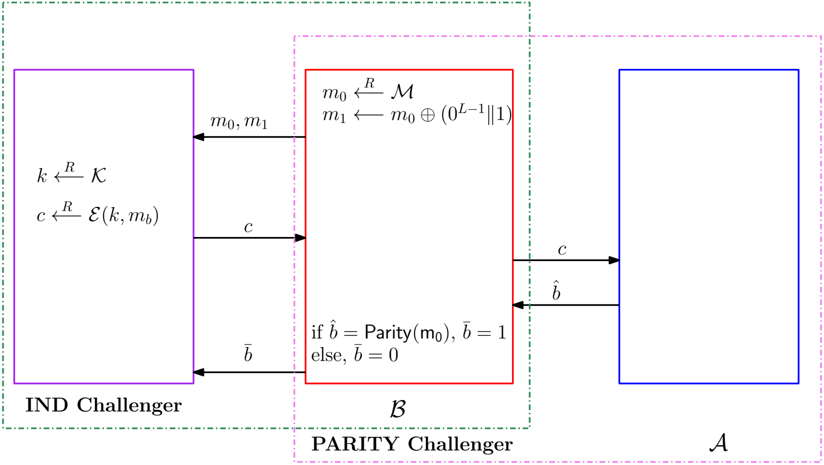


Figure: Experiment b

Proof

- For $b = 0, 1$, let $p_b = \Pr[W_b]$ for adversary $\mathcal{B}$ in indistinguishability game.

Proof

- For $b = 0, 1$, let $p_b = \Pr[W_b]$ for adversary $\mathcal{B}$ in indistinguishability game.
- The Advantage of $\mathcal{B}$ in indistinguishability game is

$$\mathsf{INDadv}[\mathcal{B}, \mathfrak{E}] = |\Pr[W_0] - \Pr[W_1]| = |p_0 - p_1|.$$

Proof

- For $b = 0, 1$, let $p_b = \Pr[W_b]$ for adversary $\mathcal{B}$ in indistinguishability game.
- The Advantage of $\mathcal{B}$ in indistinguishability game is

$$\mathsf{INDadv}[\mathcal{B}, \mathfrak{E}] = \mid \Pr[W_0] - \Pr[W_1] \mid = \mid p_0 - p_1 \mid .$$

- As $\mathfrak{E}$ is secure in the indistinguishability game, then there exists negligible function ϵ such that

$$\mathsf{INDadv}[\mathcal{B}, \mathfrak{E}] \leqslant \epsilon(n).$$

Parity Prediction

Proof

- For $b = 0, 1$, let $p_b = \Pr[W_b]$ for adversary $\mathcal{B}$ in indistinguishability game.
- The Advantage of $\mathcal{B}$ in indistinguishability game is

$$\mathsf{INDadv}[\mathcal{B}, \mathfrak{E}] = \mid \Pr[W_0] - \Pr[W_1] \mid = \mid p_0 - p_1 \mid .$$

- As $\mathfrak{E}$ is secure in the indistinguishability game, then there exists negligible function ϵ such that

$$\mathsf{INDadv}[\mathcal{B}, \mathfrak{E}] \leqslant \epsilon(n).$$

- We have to show:

$$\mathsf{PARITYadv}[\mathcal{A}, \mathfrak{E}] \leqslant \mathsf{INDadv}[\mathcal{B}, \mathfrak{E}]$$

Proof

- Regardless of whether m_0 or m_1 is encrypted, the distribution of m_b is uniform over $\mathcal{M}$ for $\mathcal{A}$.

Proof

- Regardless of whether m_0 or m_1 is encrypted, the distribution of m_b is uniform over $\mathcal{M}$ for $\mathcal{A}$.

$$p_0 = \Pr[W_0]$$

Parity Prediction

Proof

- Regardless of whether m_0 or m_1 is encrypted, the distribution of m_b is uniform over $\mathcal{M}$ for $\mathcal{A}$.

$$\begin{aligned} p_0 &= \Pr[W_0] \\ &= \Pr[\mathcal{B} \text{ outputs } \bar{b} = 1 \text{ in Experiment } 0] \end{aligned}$$

Proof

- Regardless of whether m_0 or m_1 is encrypted, the distribution of m_b is uniform over $\mathcal{M}$ for $\mathcal{A}$.

$$\begin{aligned} p_0 &= \Pr[W_0] \\ &= \Pr[\mathcal{B} \text{ outputs } \bar{b} = 1 \text{ in Experiment } 0] \\ &= \Pr[\mathcal{A} \text{ outputs } \hat{b} = \mathsf{Parity}(m_0) \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_0)] \end{aligned}$$

Proof

- Regardless of whether m_0 or m_1 is encrypted, the distribution of m_b is uniform over $\mathcal{M}$ for $\mathcal{A}$.

$$
\begin{aligned}
p_0 &= \Pr[W_0] \\
&= \Pr[\mathcal{B} \text{ outputs } \bar{b} = 1 \text{ in Experiment } 0] \\
&= \Pr[\mathcal{A} \text{ outputs } \hat{b} = \mathsf{Parity}(m_0) \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_0)] \\
&= p.
\end{aligned}
$$

Proof

- Regardless of whether m_0 or m_1 is encrypted, the distribution of m_b is uniform over $\mathcal{M}$ for $\mathcal{A}$.

$$\begin{aligned} p_0 &= \Pr[W_0] \\ &= \Pr[\mathcal{B} \text{ outputs } \bar{b} = 1 \text{ in Experiment } 0] \\ &= \Pr[\mathcal{A} \text{ outputs } \hat{b} = \mathsf{Parity}(m_0) \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_0)] \\ &= p. \end{aligned}$$

$$p_1 = \Pr[W_1]$$

Proof

- Regardless of whether m_0 or m_1 is encrypted, the distribution of m_b is uniform over $\mathcal{M}$ for $\mathcal{A}$.

$$
\begin{aligned}
p_0 &= \Pr[W_0] \\
&= \Pr[\mathcal{B} \text{ outputs } \bar{b} = 1 \text{ in Experiment } 0] \\
&= \Pr[\mathcal{A} \text{ outputs } \hat{b} = \mathsf{Parity}(m_0) \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_0)] \\
&= p.
\end{aligned}
$$

$$
\begin{aligned}
p_1 &= \Pr[W_1] \\
&= \Pr[\mathcal{B} \text{ outputs } \bar{b} = 1 \text{ in Experiment } 1]
\end{aligned}
$$

Proof

- Regardless of whether m_0 or m_1 is encrypted, the distribution of m_b is uniform over $\mathcal{M}$ for $\mathcal{A}$.

$$
\begin{aligned}
p_0 &= \Pr[W_0] \\
&= \Pr[\mathcal{B} \text{ outputs } \bar{b} = 1 \text{ in Experiment } 0] \\
&= \Pr[\mathcal{A} \text{ outputs } \hat{b} = \mathsf{Parity}(m_0) \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_0)] \\
&= p.
\end{aligned}
$$

$$
\begin{aligned}
p_1 &= \Pr[W_1] \\
&= \Pr[\mathcal{B} \text{ outputs } \bar{b} = 1 \text{ in Experiment } 1] \\
&= \Pr[\mathcal{A} \text{ outputs } \hat{b} = \mathsf{Parity}(m_0) \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_1)]
\end{aligned}
$$

Proof

- Regardless of whether m_0 or m_1 is encrypted, the distribution of m_b is uniform over $\mathcal{M}$ for $\mathcal{A}$.

$$
\begin{aligned}
p_0 &= \Pr[W_0] \\
&= \Pr[\mathcal{B} \text{ outputs } \bar{b} = 1 \text{ in Experiment } 0] \\
&= \Pr[\mathcal{A} \text{ outputs } \hat{b} = \mathsf{Parity}(m_0) \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_0)] \\
&= p.
\end{aligned}
$$

$$
\begin{aligned}
p_1 &= \Pr[W_1] \\
&= \Pr[\mathcal{B} \text{ outputs } \bar{b} = 1 \text{ in Experiment } 1] \\
&= \Pr[\mathcal{A} \text{ outputs } \hat{b} = \mathsf{Parity}(m_0) \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_1)] \\
&= 1 - \Pr[\mathcal{A} \text{ outputs } \hat{b} = \mathsf{Parity}(m_1) \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_1)]
\end{aligned}
$$

Proof

- Regardless of whether m_0 or m_1 is encrypted, the distribution of m_b is uniform over $\mathcal{M}$ for $\mathcal{A}$.

$$
\begin{aligned}
p_0 &= \Pr[W_0] \\
&= \Pr[\mathcal{B} \text{ outputs } \bar{b} = 1 \text{ in Experiment } 0] \\
&= \Pr[\mathcal{A} \text{ outputs } \hat{b} = \mathsf{Parity}(m_0) \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_0)] \\
&= p.
\end{aligned}
$$

$$
\begin{aligned}
p_1 &= \Pr[W_1] \\
&= \Pr[\mathcal{B} \text{ outputs } \bar{b} = 1 \text{ in Experiment } 1] \\
&= \Pr[\mathcal{A} \text{ outputs } \hat{b} = \mathsf{Parity}(m_0) \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_1)] \\
&= 1 - \Pr[\mathcal{A} \text{ outputs } \hat{b} = \mathsf{Parity}(m_1) \text{ given } c \xleftarrow{R} \mathcal{E}(k, m_1)] \\
&= 1 - p.
\end{aligned}
$$

Parity Prediction

Proof

$$\mathsf{INDadv}[\mathcal{B},\mathfrak{E}] = |\Pr[W_0] - \Pr[W_1]|$$

Parity Prediction

Proof

$$
\begin{aligned}
\mathsf{INDadv}[\mathcal{B},\mathfrak{E}] &= |\Pr[W_0] - \Pr[W_1]| \\
&= |p_0 - p_1|
\end{aligned}
$$

Parity Prediction

Proof

$$\begin{aligned}
\mathsf{INDadv}[\mathcal{B},\mathfrak{E}] &= |\Pr[W_0]-\Pr[W_1]| \\
&= |p_0 - p_1| \\
&= |\ p-(1-p)\ |
\end{aligned}$$

Parity Prediction

Proof

$$
\begin{aligned}
\mathsf{INDadv}[\mathcal{B},\mathfrak{E}] &= |\Pr[W_0]-\Pr[W_1]| \\
&= |p_0 - p_1| \\
&= |\ p-(1-p)\ | \\
&= |\ 2p-1\ |
\end{aligned}
$$

Proof

$$
\begin{aligned}
\mathsf{INDadv}[\mathcal{B},\mathfrak{E}] &= |\Pr[W_0]-\Pr[W_1]| \\
&= |p_0-p_1| \\
&= |\ p-(1-p)\ | \\
&= |\ 2p-1\ | \\
&= 2\left|\ p-\frac{1}{2}\ \right|
\end{aligned}
$$

Parity Prediction

Proof

$$
\begin{aligned}
\mathsf{INDadv}[\mathcal{B},\mathfrak{E}] &= |\Pr[W_0]-\Pr[W_1]| \\
&= |\,p_0-p_1\,| \\
&= |\;p-(1-p)\;| \\
&= |\;2p-1\;| \\
&= 2\left|\;p-\frac{1}{2}\;\right| \\
&= 2\cdot\mathsf{PARITYadv}[\mathcal{A},\mathfrak{E}]
\end{aligned}
$$

Parity Prediction

Proof

$$
\begin{aligned}
\mathsf{INDadv}[\mathcal{B},\mathfrak{E}] &= |\Pr[W_0] - \Pr[W_1]| \\
&= |p_0 - p_1| \\
&= |\ p - (1-p)\ | \\
&= |\ 2p - 1\ | \\
&= 2\left|\ p - \frac{1}{2}\ \right| \\
&= 2 \cdot \mathsf{PARITYadv}[\mathcal{A},\mathfrak{E}]
\end{aligned}
$$

Therefore, we have

$$\mathsf{PARITYadv}[\mathcal{A},\mathfrak{E}] = \frac{1}{2} \cdot \mathsf{INDadv}[\mathcal{B},\mathfrak{E}].$$

Bit Guessing: an alternative characterization of Indistinguishability

- This alternative characterization needs a new attack game.

Bit Guessing: an alternative characterization of Indistinguishability

- This alternative characterization needs a new attack game.
- The role played by the adversary is exactly the same as before.

Bit Guessing: an alternative characterization of Indistinguishability

- This alternative characterization needs a new attack game.
- The role played by the adversary is exactly the same as before.
- Instead of having two different experiments, there is just a single experiment.

Bit Guessing: an alternative characterization of Indistinguishability

- This alternative characterization needs a new attack game.
- The role played by the adversary is exactly the same as before.
- Instead of having two different experiments, there is just a single experiment.
- The challenger chooses $b \in \{0, 1\}$ at random and runs Experiment b of previous indistinguishability game.

Bit Guessing: an alternative characterization of Indistinguishability

Bit Guessing Indistinguishability Game

For a given cipher $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$, and for a given adversary $\mathcal{A}$, we define the Experiment as:

Bit Guessing Indistinguishability Game

For a given cipher $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$, and for a given adversary $\mathcal{A}$, we define the Experiment as:

- The adversary computes $m_0, m_1 \in \mathcal{M}$, of the same length, and sends them to the challenger.

Bit Guessing Indistinguishability Game

For a given cipher $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$, and for a given adversary $\mathcal{A}$, we define the Experiment as:

- The adversary computes $m_0, m_1 \in \mathcal{M}$, of the same length, and sends them to the challenger.
- The challenger computes $b \xleftarrow{R} \{0, 1\}$.

Bit Guessing Indistinguishability Game

For a given cipher $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$, and for a given adversary $\mathcal{A}$, we define the Experiment as:

- The adversary computes $m_0, m_1 \in \mathcal{M}$, of the same length, and sends them to the challenger.
- The challenger computes $b \xleftarrow{R} \{0, 1\}$.
- The challenger computes $k \xleftarrow{R} \mathcal{K}$; $c \xleftarrow{R} \mathcal{E}(k, m_b)$, and sends c to the adversary.

Bit Guessing: an alternative characterization of Indistinguishability

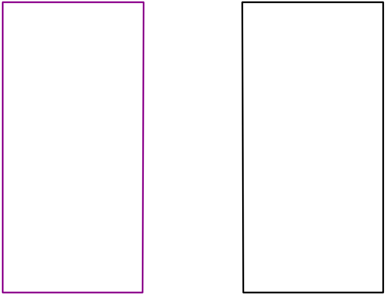
Bit Guessing Indistinguishability Game

For a given cipher $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ with security parameter $n \in \mathbb{N}$, and for a given adversary $\mathcal{A}$, we define the Experiment as:

- The adversary computes $m_0, m_1 \in \mathcal{M}$, of the same length, and sends them to the challenger.
- The challenger computes $b \xleftarrow{R} \{0, 1\}$.
- The challenger computes $k \xleftarrow{R} \mathcal{K}$; $c \xleftarrow{R} \mathcal{E}(k, m_b)$, and sends c to the adversary.
- The adversary outputs a bit $\hat{b} \in \{0, 1\}$.

Bit Guessing Indistinguishability Experiment

Challenger $\mathcal{A}$



Experiment

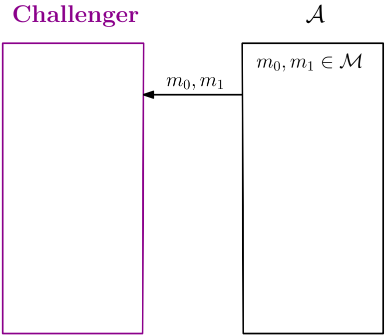
Bit Guessing Indistinguishability Experiment

Challenger $\mathcal{A}$

$m_0, m_1 \in \mathcal{M}$

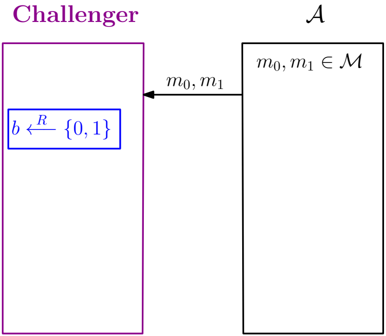
Experiment

Bit Guessing Indistinguishability Experiment



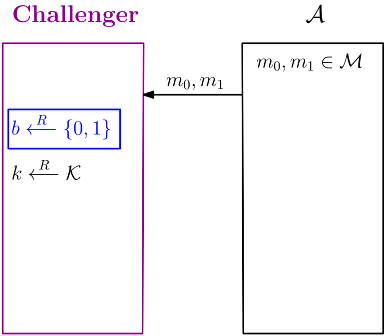
Experiment

Bit Guessing Indistinguishability Experiment



Experiment

Bit Guessing Indistinguishability Experiment



Experiment

Bit Guessing Indistinguishability Experiment

Challenger | $\mathcal{A}$

$m_0, m_1 \in \mathcal{M}$

$\xleftarrow{m_0, m_1}$

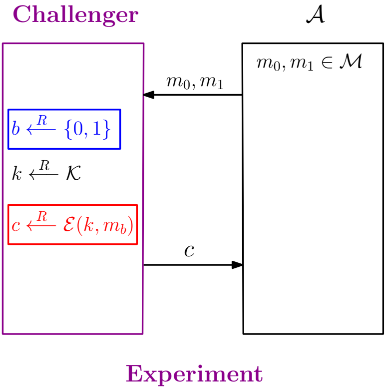
$\boxed{b \xleftarrow{R} \{0,1\}}$

$k \xleftarrow{R} \mathcal{K}$

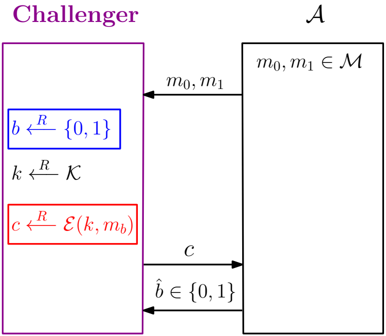
$\boxed{c \xleftarrow{R} \mathcal{E}(k, m_b)}$

Experiment

Bit Guessing Indistinguishability Experiment



Bit Guessing Indistinguishability Experiment



Experiment

Indistinguishability Advantage

Let W be the event that $\mathcal{A}$ outputs $\hat{b} = b$ in the Experiment. We define advantage of $\mathcal{A}$ in the Indistinguishability attack with respect to $\mathfrak{E}$ as

$$\mathsf{INDadv}^*[\mathcal{A}, \mathfrak{E}] = \left| \Pr[W] - \frac{1}{2} \right|.$$

Bit Guessing: an alternative characterization of Indistinguishability

Indistinguishability Advantage

Let W be the event that $\mathcal{A}$ outputs $\hat{b} = b$ in the Experiment. We define advantage of $\mathcal{A}$ in the Indistinguishability attack with respect to $\mathfrak{E}$ as

$$\mathsf{INDadv}^*[\mathcal{A}, \mathfrak{E}] = \left| \Pr[W] - \frac{1}{2} \right|.$$

Source of Randomness

Event W is defined with respect to the probability space determined by:

- the random choice of b,

Bit Guessing: an alternative characterization of Indistinguishability

Indistinguishability Advantage

Let W be the event that $\mathcal{A}$ outputs $\hat{b} = b$ in the Experiment. We define advantage of $\mathcal{A}$ in the Indistinguishability attack with respect to $\mathfrak{E}$ as

$$\mathsf{INDadv}^*[\mathcal{A}, \mathfrak{E}] = \left| \Pr[W] - \frac{1}{2} \right|.$$

Source of Randomness

Event W is defined with respect to the probability space determined by:

- the random choice of b,
- the random choice of k,
- random choices made (if any) by the encryption algorithm, and
- the random choices made (if any) by the adversary.

Theorem

For every cipher $\mathfrak{E}$ and every PPT adversary $\mathcal{A}$, we have

$$\mathsf{INDadv}[\mathcal{A}, \mathfrak{E}] = 2 \cdot \mathsf{INDadv}^*[\mathcal{A}, \mathfrak{E}].$$

Theorem

For every cipher $\mathfrak{E}$ and every PPT adversary $\mathcal{A}$, we have

$$\mathsf{INDadv}[\mathcal{A},\mathfrak{E}] = 2 \cdot \mathsf{INDadv}^*[\mathcal{A},\mathfrak{E}].$$

Proof

- Let p_0 be the probability that the adversary outputs 1 in Experiment 0 of Attack Game of INDadv.

Theorem

For every cipher $\mathfrak{E}$ and every PPT adversary $\mathcal{A}$, we have

$$\mathsf{INDadv}[\mathcal{A},\mathfrak{E}] = 2 \cdot \mathsf{INDadv}^*[\mathcal{A},\mathfrak{E}].$$

Proof

- Let p_0 be the probability that the adversary outputs 1 in Experiment 0 of Attack Game of INDadv.
- Let p_1 be the probability that the adversary outputs 1 in Experiment 1 of Attack Game of INDadv.

Bit Guessing: an alternative characterization of Indistinguishability

Theorem

For every cipher $\mathfrak{E}$ and every PPT adversary $\mathcal{A}$, we have

$$\mathsf{INDadv}[\mathcal{A},\mathfrak{E}] = 2 \cdot \mathsf{INDadv}^*[\mathcal{A},\mathfrak{E}].$$

Proof

- Let p_0 be the probability that the adversary outputs 1 in Experiment 0 of Attack Game of INDadv.
- Let p_1 be the probability that the adversary outputs 1 in Experiment 1 of Attack Game of INDadv.
- In Attack Game of INDadv^*, the Experiment 0 and Experiment 1 are conditioned over the choice of b.

Proof

- From Experiment 0: $\Pr[\hat{b} = 1 \mid b = 0] = p_0$.
- From Experiment 1: $\Pr[\hat{b} = 1 \mid b = 1] = p_1$.

Bit Guessing: an alternative characterization of Indistinguishability

Proof

- From Experiment 0: $\Pr[\hat{b} = 1 \mid b = 0] = p_0$.
- From Experiment 1: $\Pr[\hat{b} = 1 \mid b = 1] = p_1$.

$$\Pr[\hat{b} = b] = \Pr[\hat{b} = 0 \mid b = 0]\Pr[b = 0] + \Pr[\hat{b} = 1 \mid b = 1]\Pr[b = 1]$$

Proof

- From Experiment 0: $\Pr[\hat{b} = 1 \mid b = 0] = p_0$.
- From Experiment 1: $\Pr[\hat{b} = 1 \mid b = 1] = p_1$.

$$
\begin{aligned}
\Pr[\hat{b} = b] &= \Pr[\hat{b} = 0 \mid b = 0]\Pr[b = 0] + \Pr[\hat{b} = 1 \mid b = 1]\Pr[b = 1] \\
&= \Pr[\hat{b} = 0 \mid b = 0] \cdot \frac{1}{2} + \Pr[\hat{b} = 1 \mid b = 1] \cdot \frac{1}{2}, \\
&\quad \text{as } b \text{ is chosen uniformly from } \{0, 1\}
\end{aligned}
$$

Bit Guessing: an alternative characterization of Indistinguishability

Proof

- From Experiment 0: $\Pr[\hat{b} = 1 \mid b = 0] = p_0$.
- From Experiment 1: $\Pr[\hat{b} = 1 \mid b = 1] = p_1$.

$$
\begin{aligned}
\Pr[\hat{b} = b] &= \Pr[\hat{b} = 0 \mid b = 0]\Pr[b = 0] + \Pr[\hat{b} = 1 \mid b = 1]\Pr[b = 1] \\
&= \Pr[\hat{b} = 0 \mid b = 0] \cdot \frac{1}{2} + \Pr[\hat{b} = 1 \mid b = 1] \cdot \frac{1}{2}, \\
&\quad \text{as } b \text{ is chosen uniformly from } \{0, 1\} \\
&= \frac{1}{2}\left[\Pr[\hat{b} = 0 \mid b = 0] + \Pr[\hat{b} = 1 \mid b = 1]\right]
\end{aligned}
$$

Bit Guessing: an alternative characterization of Indistinguishability

Proof

- From Experiment 0: $\Pr[\hat{b} = 1 \mid b = 0] = p_0$.
- From Experiment 1: $\Pr[\hat{b} = 1 \mid b = 1] = p_1$.

$$
\begin{aligned}
\Pr[\hat{b} = b] &= \Pr[\hat{b} = 0 \mid b = 0]\Pr[b = 0] + \Pr[\hat{b} = 1 \mid b = 1]\Pr[b = 1] \\
&= \Pr[\hat{b} = 0 \mid b = 0] \cdot \frac{1}{2} + \Pr[\hat{b} = 1 \mid b = 1] \cdot \frac{1}{2}, \\
&\quad \text{as } b \text{ is chosen uniformly from } \{0, 1\} \\
&= \frac{1}{2}\left[\Pr[\hat{b} = 0 \mid b = 0] + \Pr[\hat{b} = 1 \mid b = 1]\right] \\
&= \frac{1}{2}\left[1 - \Pr[\hat{b} = 1 \mid b = 0] + \Pr[\hat{b} = 1 \mid b = 1]\right]
\end{aligned}
$$

Bit Guessing: an alternative characterization of Indistinguishability

Proof

- From Experiment 0: $\Pr[\hat{b} = 1 \mid b = 0] = p_0$.
- From Experiment 1: $\Pr[\hat{b} = 1 \mid b = 1] = p_1$.

$$
\begin{aligned}
\Pr[\hat{b} = b] &= \Pr[\hat{b} = 0 \mid b = 0]\Pr[b = 0] + \Pr[\hat{b} = 1 \mid b = 1]\Pr[b = 1] \\
&= \Pr[\hat{b} = 0 \mid b = 0] \cdot \frac{1}{2} + \Pr[\hat{b} = 1 \mid b = 1] \cdot \frac{1}{2}, \\
&\quad \text{as } b \text{ is chosen uniformly from } \{0,1\} \\
&= \frac{1}{2}\left[\Pr[\hat{b} = 0 \mid b = 0] + \Pr[\hat{b} = 1 \mid b = 1]\right] \\
&= \frac{1}{2}\left[1 - \Pr[\hat{b} = 1 \mid b = 0] + \Pr[\hat{b} = 1 \mid b = 1]\right] \\
&= \frac{1}{2}(1 - p_0 + p_1)
\end{aligned}
$$

Proof

$$\mathsf{INDadv}^*[\mathcal{A}, \mathfrak{E}] = \left| \Pr[W] - \frac{1}{2} \right|$$

Bit Guessing: an alternative characterization of Indistinguishability

Proof

$$\begin{aligned}\mathsf{INDadv}^*[\mathcal{A},\mathfrak{E}] &= \left|\ \Pr[W] - \frac{1}{2}\ \right| \\ &= \left|\ \Pr[\hat{b} = b] - \frac{1}{2}\ \right|\end{aligned}$$

Proof

$$
\begin{aligned}
\mathsf{INDadv}^*[\mathcal{A},\mathfrak{E}] &= \left| \Pr[W] - \frac{1}{2} \right| \\
&= \left| \Pr[\hat{b} = b] - \frac{1}{2} \right| \\
&= \left| \frac{1}{2}(1 - p_0 + p_1) - \frac{1}{2} \right|
\end{aligned}
$$

Bit Guessing: an alternative characterization of Indistinguishability

Proof

$$
\begin{aligned}
\mathsf{INDadv}^*[\mathcal{A},\mathfrak{E}] &= \left| \Pr[W] - \frac{1}{2} \right| \\
&= \left| \Pr[\hat{b} = b] - \frac{1}{2} \right| \\
&= \left| \frac{1}{2}(1 - p_0 + p_1) - \frac{1}{2} \right| \\
&= \left| \frac{1}{2} + \frac{1}{2}(p_1 - p_1) - \frac{1}{2} \right|
\end{aligned}
$$

Bit Guessing: an alternative characterization of Indistinguishability

Proof

$$
\begin{aligned}
\mathsf{INDadv}^*[\mathcal{A},\mathfrak{E}] &= \left|\ \Pr[W] - \frac{1}{2}\ \right| \\
&= \left|\ \Pr[\hat{b} = b] - \frac{1}{2}\ \right| \\
&= \left|\ \frac{1}{2}(1 - p_0 + p_1) - \frac{1}{2}\ \right| \\
&= \left|\ \frac{1}{2} + \frac{1}{2}(p_1 - p_1) - \frac{1}{2}\ \right| \\
&= \frac{1}{2}\left|\ p_1 - p_0\ \right|
\end{aligned}
$$

Bit Guessing: an alternative characterization of Indistinguishability

Proof

$$
\begin{aligned}
\mathsf{INDadv}^*[\mathcal{A},\mathfrak{E}] &= \left| \Pr[W] - \frac{1}{2} \right| \\
&= \left| \Pr[\hat{b} = b] - \frac{1}{2} \right| \\
&= \left| \frac{1}{2}(1 - p_0 + p_1) - \frac{1}{2} \right| \\
&= \left| \frac{1}{2} + \frac{1}{2}(p_1 - p_1) - \frac{1}{2} \right| \\
&= \frac{1}{2} \mid p_1 - p_0 \mid \\
&= \frac{1}{2}\mathsf{INDadv}[\mathcal{A},\mathfrak{E}].
\end{aligned}
$$

- Let $f(m) = \textsf{Parity}(m)$ and $\mathfrak{E}$ is computationally indistinguishable in presence of an eavesdropper.

- Let $f(m) = \textsf{Parity}(m)$ and $\mathfrak{E}$ is computationally indistinguishable in presence of an eavesdropper. We have seen:
  - If m is chosen uniformly at random, then no adversary can predict parity with probability better than 1/2 given $\mathcal{E}(k,m)$.

Semantic Security

- Let $f(m) = \mathsf{Parity}(m)$ and $\mathfrak{E}$ is computationally indistinguishable in presence of an eavesdropper. We have seen:
  - If m is chosen uniformly at random, then no adversary can predict parity with probability better than 1/2 given $\mathcal{E}(k,m)$.
- Indistinguishability means that no PPT adversary can learn any function of the plaintext given the ciphertext, regardless of the distribution of the message being sent.

- Let $f(m) = \mathsf{Parity}(m)$ and $\mathfrak{E}$ is computationally indistinguishable in presence of an eavesdropper. We have seen:
  - If m is chosen uniformly at random, then no adversary can predict parity with probability better than 1/2 given $\mathcal{E}(k,m)$.
- Indistinguishability means that no PPT adversary can learn any function of the plaintext given the ciphertext, regardless of the distribution of the message being sent.
  - If there exists any adversary who correctly computes $f(m)$ with some probability when given $\mathcal{E}(k,m)$, then there exists another adversary that can correctly compute $f(m)$ with the same probability without being given the ciphertext at all (and only knowing the distribution on m).

Semantic Security

- m is chosen uniformly at random from some set $\mathcal{S} \subset \{0,1\}^n$.

- m is chosen uniformly at random from some set $\mathcal{S} \subset \{0,1\}^n$.
- In an asymptotic setting, we will work with an infinite set $\mathcal{S} \subset \{0,1\}^*$.

- m is chosen uniformly at random from some set $\mathcal{S} \subset \{0,1\}^n$.
- In an asymptotic setting, we will work with an infinite set $\mathcal{S} \subset \{0,1\}^*$.
- Let security parameter be n.

- m is chosen uniformly at random from some set $\mathcal{S} \subset \{0,1\}^n$.
- In an asymptotic setting, we will work with an infinite set $\mathcal{S} \subset \{0,1\}^*$.
- Let security parameter be n.
- Plaintext is chosen uniformly from $\mathcal{S}_n \stackrel{def}{=} \mathcal{S} \cap \{0,1\}^n$.

- m is chosen uniformly at random from some set $\mathcal{S} \subset \{0,1\}^n$.
- In an asymptotic setting, we will work with an infinite set $\mathcal{S} \subset \{0,1\}^*$.
- Let security parameter be n.
- Plaintext is chosen uniformly from $\mathcal{S}_n \stackrel{def}{=} \mathcal{S} \cap \{0,1\}^n$.
- $\mathcal{S}$ is efficiently sampleable.
  - There exists some probabilistic polynomial-time algorithm that, on input 1^n, outputs a uniform element of $\mathcal{S}_n$.

- m is chosen uniformly at random from some set $\mathcal{S} \subset \{0,1\}^n$.
- In an asymptotic setting, we will work with an infinite set $\mathcal{S} \subset \{0,1\}^*$.
- Let security parameter be n.
- Plaintext is chosen uniformly from $\mathcal{S}_n \stackrel{def}{=} \mathcal{S} \cap \{0,1\}^n$.
- $\mathcal{S}$ is efficiently sampleable.
  - There exists some probabilistic polynomial-time algorithm that, on input 1^n, outputs a uniform element of $\mathcal{S}_n$.
- The functions f that can be computed in polynomial time.

Theorem

Let $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a fixed-length computational cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ for security parameter n that is computationally indistinguishable in the presence of an eavesdropper. Then for any PPT algorithm $\mathcal{A}$, there is a PPT algorithm $\mathcal{A}'$ such that for any efficiently-sampleable set $\mathcal{S}$ and any function f, there is a negligible function ϵ such that:

$$\left|\Pr[\mathcal{A}(1^n, \mathcal{E}(k, m)) = f(m)] - \Pr[\mathcal{A}'(1^n) = f(m)]\right| \leq \mathsf{negl}(n),$$

where the where m is chosen uniformly at random from $\mathcal{S}_n \stackrel{def}{=} \mathcal{S} \cap \{0, 1\}^n$ and the probabilities are taken over the choice of m and the key k, and any random coins used by $\mathcal{A}, \mathcal{A}'$, and the encryption process.

Semantic Security

Proof (Sketch)

- $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is computationally indistinguishable in the presence of an eavesdropper.
  - No PPT adversary $\mathcal{A}$ can distinguish between $\mathcal{E}(k, m)$ and $\mathcal{E}(k, 1^n)$, for any $m \in \{0, 1\}^n$.

Proof (Sketch)

- $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is computationally indistinguishable in the presence of an eavesdropper.
  - No PPT adversary $\mathcal{A}$ can distinguish between $\mathcal{E}(k, m)$ and $\mathcal{E}(k, 1^n)$, for any $m \in \{0,1\}^n$.
- Let $\mathcal{A}$ successfully computes $f(m)$ given $\mathcal{E}(k, m)$.

Proof (Sketch)

- $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is computationally indistinguishable in the presence of an eavesdropper.
  - No PPT adversary $\mathcal{A}$ can distinguish between $\mathcal{E}(k, m)$ and $\mathcal{E}(k, 1^n)$, for any $m \in \{0,1\}^n$.
- Let $\mathcal{A}$ successfully computes $f(m)$ given $\mathcal{E}(k, m)$.
- **Claim:** $\mathcal{A}$ computes $f(m)$ given $\mathcal{E}(k, 1^n)$ with almost same probability.

Proof (Sketch)

- $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is computationally indistinguishable in the presence of an eavesdropper.
  - No PPT adversary $\mathcal{A}$ can distinguish between $\mathcal{E}(k, m)$ and $\mathcal{E}(k, 1^n)$, for any $m \in \{0, 1\}^n$.
- Let $\mathcal{A}$ successfully computes $f(m)$ given $\mathcal{E}(k, m)$.
- **Claim:** $\mathcal{A}$ computes $f(m)$ given $\mathcal{E}(k, 1^n)$ with almost same probability.
- Otherwise, $\mathcal{A}$ could be used to distinguish between $\mathcal{E}(k, m)$ and $\mathcal{E}(k, 1^n)$.

Proof (Sketch)

- Construction of $\mathcal{B}$ (distinguisher): a PPT adversary that uses $\mathcal{A}$ as sub-routine in bit-guessing game:

Proof (Sketch)

- Construction of $\mathcal{B}$ (distinguisher): a PPT adversary that uses $\mathcal{A}$ as sub-routine in bit-guessing game:
  - Let $\mathcal{S}$ be efficiently-sampleable.

Proof (Sketch)

- Construction of $\mathcal{B}$ (distinguisher): a PPT adversary that uses $\mathcal{A}$ as sub-routine in bit-guessing game:
  - Let $\mathcal{S}$ be efficiently-sampleable.
  - $\mathcal{B}$ samples $m \xleftarrow{R} \mathcal{S}_n$.

Proof (Sketch)

- Construction of $\mathcal{B}$ (distinguisher): a PPT adversary that uses $\mathcal{A}$ as sub-routine in bit-guessing game:
    - Let $\mathcal{S}$ be efficiently-sampleable.
    - $\mathcal{B}$ samples $m \xleftarrow{R} \mathcal{S}_n$.
    - $\mathcal{B}$ sets $m_0 = m$ and $m_1 = 1^n$.

Proof (Sketch)

- Construction of $\mathcal{B}$ (distinguisher): a PPT adversary that uses $\mathcal{A}$ as sub-routine in bit-guessing game:
  - Let $\mathcal{S}$ be efficiently-sampleable.
  - $\mathcal{B}$ samples $m \xleftarrow{R} \mathcal{S}_n$.
  - $\mathcal{B}$ sets $m_0 = m$ and $m_1 = 1^n$.
  - $\mathcal{B}$ sends m_0 and m_1 to challenger.

Proof (Sketch)

- Construction of $\mathcal{B}$ (distinguisher): a PPT adversary that uses $\mathcal{A}$ as sub-routine in bit-guessing game:
  - Let $\mathcal{S}$ be efficiently-sampleable.
  - $\mathcal{B}$ samples $m \xleftarrow{R} \mathcal{S}_n$.
  - $\mathcal{B}$ sets $m_0 = m$ and $m_1 = 1^n$.
  - $\mathcal{B}$ sends m_0 and m_1 to challenger.
  - Challenger $b \xleftarrow{R} \{0,1\}$, $k \xleftarrow{R} \mathcal{K}$ and sends $c \xleftarrow{R} \mathcal{E}(k, m_b)$ to $\mathcal{B}$.

Proof (Sketch)

- Construction of $\mathcal{B}$ (distinguisher): a PPT adversary that uses $\mathcal{A}$ as sub-routine in bit-guessing game:
  - Let $\mathcal{S}$ be efficiently-sampleable.
  - $\mathcal{B}$ samples $m \xleftarrow{R} \mathcal{S}_n$.
  - $\mathcal{B}$ sets $m_0 = m$ and $m_1 = 1^n$.
  - $\mathcal{B}$ sends m_0 and m_1 to challenger.
  - Challenger $b \xleftarrow{R} \{0,1\}$, $k \xleftarrow{R} \mathcal{K}$ and sends $c \xleftarrow{R} \mathcal{E}(k, m_b)$ to $\mathcal{B}$.
  - $\mathcal{B}$ forwards c to $\mathcal{A}$.

Proof (Sketch)

- Construction of $\mathcal{B}$ (distinguisher): a PPT adversary that uses $\mathcal{A}$ as sub-routine in bit-guessing game:
  - Let $\mathcal{S}$ be efficiently-sampleable.
  - $\mathcal{B}$ samples $m \xleftarrow{R} \mathcal{S}_n$.
  - $\mathcal{B}$ sets $m_0 = m$ and $m_1 = 1^n$.
  - $\mathcal{B}$ sends m_0 and m_1 to challenger.
  - Challenger $b \xleftarrow{R} \{0,1\}$, $k \xleftarrow{R} \mathcal{K}$ and sends $c \xleftarrow{R} \mathcal{E}(k, m_b)$ to $\mathcal{B}$.
  - $\mathcal{B}$ forwards c to $\mathcal{A}$.
  - $\mathcal{B}$ output 1 if and only if $\mathcal{A}$ outputs $f(m)$.

Proof (Sketch)

- Construction of $\mathcal{B}$ (distinguisher): a PPT adversary that uses $\mathcal{A}$ as sub-routine in bit-guessing game:
  - Let $\mathcal{S}$ be efficiently-sampleable.
  - $\mathcal{B}$ samples $m \xleftarrow{R} \mathcal{S}_n$.
  - $\mathcal{B}$ sets $m_0 = m$ and $m_1 = 1^n$.
  - $\mathcal{B}$ sends m_0 and m_1 to challenger.
  - Challenger $b \xleftarrow{R} \{0, 1\}$, $k \xleftarrow{R} \mathcal{K}$ and sends $c \xleftarrow{R} \mathcal{E}(k, m_b)$ to $\mathcal{B}$.
  - $\mathcal{B}$ forwards c to $\mathcal{A}$.
  - $\mathcal{B}$ output 1 if and only if $\mathcal{A}$ outputs $f(m)$.
- This observation, gives us the idea of $\mathcal{A}'$.

Proof (Sketch)

- Construction of $\mathcal{A}'$:
  - $\mathcal{A}'$ computes $k \xleftarrow{R} \mathcal{G}(1^n)$.

Proof (Sketch)

- Construction of $\mathcal{A}'$:
  - $\mathcal{A}'$ computes $k \xleftarrow{R} \mathcal{G}(1^n)$.
  - Computes $c \xleftarrow{R} \mathcal{E}(k, 1^n)$.

Proof (Sketch)

- Construction of $\mathcal{A}'$:
  - $\mathcal{A}'$ computes $k \xleftarrow{R} \mathcal{G}(1^n)$.
  - Computes $c \xleftarrow{R} \mathcal{E}(k, 1^n)$.
  - Sends c to $\mathcal{A}$.

Semantic Security

Proof (Sketch)

- Construction of $\mathcal{A}'$:
  - $\mathcal{A}'$ computes $k \xleftarrow{R} \mathcal{G}(1^n)$.
  - Computes $c \xleftarrow{R} \mathcal{E}(k, 1^n)$.
  - Sends c to $\mathcal{A}$.
  - $\mathcal{A}'$ outputs whatever $\mathcal{A}$ returns.

Proof (Sketch)

- Construction of $\mathcal{A}'$:
  - $\mathcal{A}'$ computes $k \xleftarrow{R} \mathcal{G}(1^n)$.
  - Computes $c \xleftarrow{R} \mathcal{E}(k, 1^n)$.
  - Sends c to $\mathcal{A}$.
  - $\mathcal{A}'$ outputs whatever $\mathcal{A}$ returns.
- By the above, $\mathcal{A}$ outputs $f(m)$ when run as a sub-routine by $\mathcal{A}'$ with almost the same probability as when it receives $\mathcal{E}(k, m)$.

Proof (Sketch)

- Construction of $\mathcal{A}'$:
  - $\mathcal{A}'$ computes $k \xleftarrow{R} \mathcal{G}(1^n)$.
  - Computes $c \xleftarrow{R} \mathcal{E}(k, 1^n)$.
  - Sends c to $\mathcal{A}$.
  - $\mathcal{A}'$ outputs whatever $\mathcal{A}$ returns.
- By the above, $\mathcal{A}$ outputs $f(m)$ when run as a sub-routine by $\mathcal{A}'$ with almost the same probability as when it receives $\mathcal{E}(k,m)$.
- Thus, $\mathcal{A}'$ fulfills the property required by the claim.

- Semantic security is more general than previous theorem.

Semantic Security

- Semantic security is more general than previous theorem.
- Now we consider:
  - Arbitrary distributions over plaintext
    - Consider an arbitrary distribution $X = \{X_1, X_2, \ldots\}$.

Semantic Security

- Semantic security is more general than previous theorem.
- Now we consider:
  - Arbitrary distributions over plaintext
    - Consider an arbitrary distribution $X = \{X_1, X_2, \ldots\}$.
    - For security parameter n, use distribution X_n to sample plaintext.

- Semantic security is more general than previous theorem.
- Now we consider:
  - Arbitrary distributions over plaintext
    - Consider an arbitrary distribution $X = \{X_1, X_2, \ldots\}$.
    - For security parameter n, use distribution X_n to sample plaintext.
    - Sample space of any X_i contains strings of same length.

Semantic Security

- Semantic security is more general than previous theorem.
- Now we consider:
  - Arbitrary distributions over plaintext
    - Consider an arbitrary distribution $X = \{X_1, X_2, \ldots\}$.
    - For security parameter n, use distribution X_n to sample plaintext.
    - Sample space of any X_i contains strings of same length.
    - For all i, X_i is sampleable.

- Semantic security is more general than previous theorem.
- Now we consider:
  - Arbitrary distributions over plaintext
    - Consider an arbitrary distribution $X = \{X_1, X_2, \ldots\}$.
    - For security parameter n, use distribution X_n to sample plaintext.
    - Sample space of any X_i contains strings of same length.
    - For all i, X_i is sampleable.
  - Leaked information $h(m)$ given to adversary.

Semantic Security

- Semantic security is more general than previous theorem.
- Now we consider:
  - Arbitrary distributions over plaintext
    - Consider an arbitrary distribution $X = \{X_1, X_2, \ldots\}$.
    - For security parameter n, use distribution X_n to sample plaintext.
    - Sample space of any X_i contains strings of same length.
    - For all i, X_i is sampleable.
  - Leaked information $h(m)$ given to adversary.
  - Adversary computes $f(m)$.

Definition (Semantic Security)

A computational cipher $\mathfrak{E} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ is semantically secure in the presence of an eavesdropper if for every PPT adversary $\mathcal{A}$ there exists a PPT adversary $\mathcal{A}'$ such that for all efficiently-sampleable distributions $X = (X_1, X_2, \ldots)$ and all polynomial-time computable functions f and h, there exists a negligible function ϵ such that

$$\left|\Pr[\mathcal{A}(1^n, \mathcal{E}(k,m), h(m)) = f(m)] - \Pr[\mathcal{A}'(1^n, h(m)) = f(m)]\right| \leqslant \epsilon(n),$$

where m is chosen according to distribution X_n, and the probabilities are taken over the choice of m and the key k, and any random coins used by $\mathcal{A}, \mathcal{A}'$, and the encryption process.

Theorem

A computational cipher has indistinguishable encryptions in the presence of an eavesdropper if and only if it is semantically secure in the presence of an eavesdropper.

Computational ciphers: the formalities

System Parameterization

- In the mathematical model (though not always in real-world systems), we associate with $\mathfrak{E}$ families of key, message, and ciphertext spaces, indexed by
  - a security parameter, which is a positive integer, and is denoted by n, and
  - a system parameter, which is a bit string, and is denoted by ν.

System Parameterization

- In the mathematical model (though not always in real-world systems), we associate with $\mathfrak{E}$ families of key, message, and ciphertext spaces, indexed by
  - a security parameter, which is a positive integer, and is denoted by n, and
  - a system parameter, which is a bit string, and is denoted by ν.
- families of finite sets

$$\{\mathcal{K}_{n,\nu}\}_{n,\nu}, \ \{\mathcal{M}_{n,\nu}\}_{n,\nu} \ \text{ and } \ \{\mathcal{C}_{n,\nu}\}_{n,\nu}.$$

Computational ciphers: the formalities

System Parameterization

- In the mathematical model (though not always in real-world systems), we associate with $\mathfrak{E}$ families of key, message, and ciphertext spaces, indexed by
  - a security parameter, which is a positive integer, and is denoted by n, and
  - a system parameter, which is a bit string, and is denoted by ν.
- families of finite sets

$$\{\mathcal{K}_{n,\nu}\}_{n,\nu},\ \{\mathcal{M}_{n,\nu}\}_{n,\nu}\ \text{ and }\ \{\mathcal{C}_{n,\nu}\}_{n,\nu}.$$

1. First choose $n \in \mathbb{N}$ as security parameter.

Computational ciphers: the formalities

System Parameterization

- In the mathematical model (though not always in real-world systems), we associate with $\mathfrak{E}$ families of key, message, and ciphertext spaces, indexed by
  - a security parameter, which is a positive integer, and is denoted by n, and
  - a system parameter, which is a bit string, and is denoted by ν.
- families of finite sets

$$\{\mathcal{K}_{n,\nu}\}_{n,\nu}, \ \{\mathcal{M}_{n,\nu}\}_{n,\nu} \ \text{ and } \ \{\mathcal{C}_{n,\nu}\}_{n,\nu}.$$

1. First choose $n \in \mathbb{N}$ as security parameter.
2. A system parameter ν is generated using an algorithm specific to the cipher.

Computational ciphers: the formalities

System Parameterization

- In the mathematical model (though not always in real-world systems), we associate with $\mathfrak{E}$ families of key, message, and ciphertext spaces, indexed by
  - a security parameter, which is a positive integer, and is denoted by n, and
  - a system parameter, which is a bit string, and is denoted by ν.
- families of finite sets

$$\{\mathcal{K}_{n,\nu}\}_{n,\nu}, \ \{\mathcal{M}_{n,\nu}\}_{n,\nu} \ \text{ and } \ \{\mathcal{C}_{n,\nu}\}_{n,\nu}.$$

1. First choose $n \in \mathbb{N}$ as security parameter.
2. A system parameter ν is generated using an algorithm specific to the cipher.
3. The system parameter ν together with n, gives a detailed description of a fixed instance of the cipher, with

$$(\mathcal{K}, \mathcal{M}, \mathcal{C}) = (\mathcal{K}_{n,\nu}, \mathcal{M}_{n,\nu}, \mathcal{C}_{n,\nu}).$$

Definition (System Parameterization)

A system parameterization is an efficient probabilistic algorithm P that given a security parameter $n \in \mathbb{N}$ as input, outputs a bit string ν, called a system parameter, whose length is always bounded by a polynomial in n.

Definition (System Parameterization)

A system parameterization is an efficient probabilistic algorithm P that given a security parameter $n \in \mathbb{N}$ as input, outputs a bit string ν, called a system parameter, whose length is always bounded by a polynomial in n. $\mathsf{Supp}(P(n))$ denotes the support of the distribution $P(n)$.

Computational ciphers: the formalities

Definition (System Parameterization)

A system parameterization is an efficient probabilistic algorithm P that given a security parameter $n \in \mathbb{N}$ as input, outputs a bit string ν, called a system parameter, whose length is always bounded by a polynomial in n. $\mathsf{Supp}(P(n))$ denotes the support of the distribution $P(n)$.

- A collection $\mathcal{S} = \{\mathcal{S}_{n,\nu}\}_{n,\nu}$ of finite sets of bits strings, where n runs over $\mathbb{N}$ and ν runs over $\mathsf{Supp}(P(n))$, is called a family of spaces with system parameterization P, provided the lengths of all the strings in each of the sets $\mathcal{S}_{n,\nu}$ are bounded by some polynomial p in n.

Computational ciphers: the formalities

Definition (System Parameterization)

A system parameterization is an efficient probabilistic algorithm P that given a security parameter $n \in \mathbb{N}$ as input, outputs a bit string ν, called a system parameter, whose length is always bounded by a polynomial in n. $\mathsf{Supp}(P(n))$ denotes the support of the distribution $P(n)$.

- A collection $\mathcal{S} = \{\mathcal{S}_{n,\nu}\}_{n,\nu}$ of finite sets of bits strings, where n runs over $\mathbb{N}$ and ν runs over $\mathsf{Supp}(P(n))$, is called a family of spaces with system parameterization P, provided the lengths of all the strings in each of the sets $\mathcal{S}_{n,\nu}$ are bounded by some polynomial p in n.
- We say that $\mathcal{S}$ is efficiently recognizable if there is an efficient deterministic algorithm that on input $n \in \mathbb{N}$, $\nu \in \mathsf{Supp}(P(n))$, and $s \in \{0,1\}^{\leqslant p(n)}$, determines if $s \in \mathcal{S}_{n,\nu}$.

Computational ciphers: the formalities

Definition (System Parameterization)

A system parameterization is an efficient probabilistic algorithm P that given a security parameter $n \in \mathbb{N}$ as input, outputs a bit string ν, called a system parameter, whose length is always bounded by a polynomial in n. $\mathsf{Supp}(P(n))$ denotes the support of the distribution $P(n)$.

- A collection $\mathcal{S} = \{\mathcal{S}_{n,\nu}\}_{n,\nu}$ of finite sets of bits strings, where n runs over $\mathbb{N}$ and ν runs over $\mathsf{Supp}(P(n))$, is called a family of spaces with system parameterization P, provided the lengths of all the strings in each of the sets $\mathcal{S}_{n,\nu}$ are bounded by some polynomial p in n.
- We say that $\mathcal{S}$ is efficiently recognizable if there is an efficient deterministic algorithm that on input $n \in \mathbb{N}$, $\nu \in \mathsf{Supp}(P(n))$, and $s \in \{0,1\}^{\leqslant p(n)}$, determines if $s \in \mathcal{S}_{n,\nu}$.
- We say that $\mathcal{S}$ is efficiently sampleable if there is an efficient probabilistic algorithm that on input $n \in \mathbb{N}$ and $\nu \in \mathsf{Supp}(P(n))$, outputs an element uniformly distributed over $\mathcal{S}_{n,\nu}$.

Computational ciphers: the formalities

Definition (System Parameterization)

A system parameterization is an efficient probabilistic algorithm P that given a security parameter $n \in \mathbb{N}$ as input, outputs a bit string ν, called a system parameter, whose length is always bounded by a polynomial in n. $\mathsf{Supp}(P(n))$ denotes the support of the distribution $P(n)$.

- A collection $\mathcal{S} = \{\mathcal{S}_{n,\nu}\}_{n,\nu}$ of finite sets of bits strings, where n runs over $\mathbb{N}$ and ν runs over $\mathsf{Supp}(P(n))$, is called a family of spaces with system parameterization P, provided the lengths of all the strings in each of the sets $\mathcal{S}_{n,\nu}$ are bounded by some polynomial p in n.
- We say that $\mathcal{S}$ is efficiently recognizable if there is an efficient deterministic algorithm that on input $n \in \mathbb{N}$, $\nu \in \mathsf{Supp}(P(n))$, and $s \in \{0,1\}^{\leqslant p(n)}$, determines if $s \in \mathcal{S}_{n,\nu}$.
- We say that $\mathcal{S}$ is efficiently sampleable if there is an efficient probabilistic algorithm that on input $n \in \mathbb{N}$ and $\nu \in \mathsf{Supp}(P(n))$, outputs an element uniformly distributed over $\mathcal{S}_{n,\nu}$.
- We say that $\mathcal{S}$ has an effective length function if there is an efficient deterministic algorithm that on input $n \in \mathbb{N}$, $\nu \in \mathsf{Supp}(P(n))$, and $s \in \mathcal{S}_{n,\nu}$, outputs a non-negative integer, called the length of s.

Definition (Computational ciphers)

A computational cipher $\mathfrak{E}$ consists of a tuple of algorithms $(\mathcal{G}, \mathcal{E}, \mathcal{D})$, along with three families of spaces with system parameterization P:

$$\mathbf{K} = \{\mathcal{K}_{n,\nu}\}_{n,\nu}, \ \mathbf{M} = \{\mathcal{M}_{n,\nu}\}_{n,\nu} \ \text{ and } \ \mathbf{C} = \{\mathcal{C}_{n,\nu}\}_{n,\nu}.$$

1. $\mathbf{M}, \mathbf{C}$ and $\mathbf{K}$ are efficiently recognizable.

Computational ciphers: the formalities

Definition (Computational ciphers)

A computational cipher $\mathfrak{E}$ consists of a tuple of algorithms $(\mathcal{G}, \mathcal{E}, \mathcal{D})$, along with three families of spaces with system parameterization P:

$$\mathbf{K} = \{\mathcal{K}_{n,\nu}\}_{n,\nu}, \ \ \mathbf{M} = \{\mathcal{M}_{n,\nu}\}_{n,\nu} \ \ \text{and} \ \ \mathbf{C} = \{\mathcal{C}_{n,\nu}\}_{n,\nu}.$$

1. $\mathbf{M}, \mathbf{C}$ and $\mathbf{K}$ are efficiently recognizable.
2. $\mathbf{K}$ is efficiently sampleable and $\mathcal{G}$ is such a sampling algorithm.

Computational ciphers: the formalities

Definition (Computational ciphers)

A computational cipher $\mathfrak{E}$ consists of a tuple of algorithms $(\mathcal{G}, \mathcal{E}, \mathcal{D})$, along with three families of spaces with system parameterization P:

$$\mathbf{K} = \{\mathcal{K}_{n,\nu}\}_{n,\nu}, \ \ \mathbf{M} = \{\mathcal{M}_{n,\nu}\}_{n,\nu} \quad \text{and} \quad \mathbf{C} = \{\mathcal{C}_{n,\nu}\}_{n,\nu}.$$

1. $\mathbf{M}, \mathbf{C}$ and $\mathbf{K}$ are efficiently recognizable.
2. $\mathbf{K}$ is efficiently sampleable and $\mathcal{G}$ is such a sampling algorithm.
3. $\mathbf{M}$ has an effective length function.

Computational ciphers: the formalities

Definition (Computational ciphers)

A computational cipher $\mathfrak{E}$ consists of a tuple of algorithms $(\mathcal{G}, \mathcal{E}, \mathcal{D})$, along with three families of spaces with system parameterization P:

$$\mathbf{K} = \{\mathcal{K}_{n,\nu}\}_{n,\nu}, \ \mathbf{M} = \{\mathcal{M}_{n,\nu}\}_{n,\nu} \ \text{ and } \ \mathbf{C} = \{\mathcal{C}_{n,\nu}\}_{n,\nu}.$$

1. $\mathbf{M}, \mathbf{C}$ and $\mathbf{K}$ are efficiently recognizable.
2. $\mathbf{K}$ is efficiently sampleable and $\mathcal{G}$ is such a sampling algorithm.
3. $\mathbf{M}$ has an effective length function.
4. Algorithm $\mathcal{E}$ is an efficient probabilistic algorithm that on input n, ν, k, m, where $n \in \mathbb{N}$, $\nu \in \mathsf{Supp}(P(n))$, $k \in \mathcal{K}_{n,\nu}$, and $m \in \mathcal{M}_{n,\nu}$, always outputs an element of $\mathcal{C}_{n,\nu}$.

Computational ciphers: the formalities

Definition (Computational ciphers)

A computational cipher $\mathfrak{E}$ consists of a tuple of algorithms $(\mathcal{G}, \mathcal{E}, \mathcal{D})$, along with three families of spaces with system parameterization P:

$$\mathbf{K} = \{\mathcal{K}_{n,\nu}\}_{n,\nu}, \ \ \mathbf{M} = \{\mathcal{M}_{n,\nu}\}_{n,\nu} \quad \text{and} \quad \mathbf{C} = \{\mathcal{C}_{n,\nu}\}_{n,\nu}.$$

1. $\mathbf{M}, \mathbf{C}$ and $\mathbf{K}$ are efficiently recognizable.
2. $\mathbf{K}$ is efficiently sampleable and $\mathcal{G}$ is such a sampling algorithm.
3. $\mathbf{M}$ has an effective length function.
4. Algorithm $\mathcal{E}$ is an efficient probabilistic algorithm that on input n, ν, k, m, where $n \in \mathbb{N}$, $\nu \in \mathsf{Supp}(P(n))$, $k \in \mathcal{K}_{n,\nu}$, and $m \in \mathcal{M}_{n,\nu}$, always outputs an element of $\mathcal{C}_{n,\nu}$.
5. Algorithm $\mathcal{D}$ is an efficient deterministic algorithm that on input n, ν, k, m, where $n \in \mathbb{N}$, $\nu \in \mathsf{Supp}(P(n))$, $k \in \mathcal{K}_{n,\nu}$, and $c \in \mathcal{C}_{n,\nu}$, outputs either an element of $\mathcal{M}_{n,\nu}$, or a special symbol $\mathsf{reject} \in \mathcal{M}_{n,\nu}$.

Computational ciphers: the formalities

Definition (Computational ciphers)

A computational cipher $\mathfrak{E}$ consists of a tuple of algorithms $(\mathcal{G}, \mathcal{E}, \mathcal{D})$, along with three families of spaces with system parameterization P:

$$\mathbf{K} = \{\mathcal{K}_{n,\nu}\}_{n,\nu}, \ \mathbf{M} = \{\mathcal{M}_{n,\nu}\}_{n,\nu} \ \text{ and } \ \mathbf{C} = \{\mathcal{C}_{n,\nu}\}_{n,\nu}.$$

1. $\mathbf{M}, \mathbf{C}$ and $\mathbf{K}$ are efficiently recognizable.
2. $\mathbf{K}$ is efficiently sampleable and $\mathcal{G}$ is such a sampling algorithm.
3. $\mathbf{M}$ has an effective length function.
4. Algorithm $\mathcal{E}$ is an efficient probabilistic algorithm that on input n, ν, k, m, where $n \in \mathbb{N}$, $\nu \in \mathsf{Supp}(P(n))$, $k \in \mathcal{K}_{n,\nu}$, and $m \in \mathcal{M}_{n,\nu}$, always outputs an element of $\mathcal{C}_{n,\nu}$.
5. Algorithm $\mathcal{D}$ is an efficient deterministic algorithm that on input n, ν, k, m, where $n \in \mathbb{N}$, $\nu \in \mathsf{Supp}(P(n))$, $k \in \mathcal{K}_{n,\nu}$, and $c \in \mathcal{C}_{n,\nu}$, outputs either an element of $\mathcal{M}_{n,\nu}$, or a special symbol $\mathsf{reject} \in \mathcal{M}_{n,\nu}$.
6. For all n, ν, k, m, c where $n \in \mathbb{N}$, $\nu \in \mathsf{Supp}(P(n))$, $k \in \mathcal{K}_{n,\nu}$, $m \in \mathcal{M}_{n,\nu}$, and $c \in \mathsf{Supp}(\mathcal{E}(n, \nu; k, m))$, we have $\mathcal{D}(n, \nu; k, c) = m$.

End