Cryptography

Lecture 3

Arkady Yerukhimovich

September 4, 2024

Announcements

1. Homework will now be due on Fridays at 5pm
2. Homework 1 is out now on Piazza
3. Office hours: Starting next week
   - Monday 2:15-3:15
   - Friday 2:30-3:30

Outline

Lecture 2 Review

- Probability review
- Perfectly-secure private-key encryption
- One-time pad

Outline

The One-Time Pad

One-Time Pad Encryption Scheme

- Let $\mathcal{M} = \mathcal{K} = \mathcal{C} = \{0,1\}^{\ell}$
- Gen: $k \leftarrow \mathcal{K}$
- Enc: $c = k \oplus m$ ($\oplus$ denotes bitwise exclusive-OR)
- Dec: $m = k \oplus c$

Correctness: For all $k \in \mathcal{K}$ and all $m \in \mathcal{M}$,

$$\mathsf{Dec}_k(\mathsf{Enc}_k(m)) =$$

The One-Time Pad

One-Time Pad Encryption Scheme

- Let $\mathcal{M} = \mathcal{K} = \mathcal{C} = \{0,1\}^\ell$
- Gen: $k \leftarrow \mathcal{K}$
- Enc: $c = k \oplus m$ ($\oplus$ denotes bitwise exclusive-OR)
- Dec: $m = k \oplus c$

Correctness: For all $k \in \mathcal{K}$ and all $m \in \mathcal{M}$,

$$\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = k \oplus (k \oplus m) = (k \oplus k) \oplus m = 0^\ell \oplus m = m$$

Security: The OTP is perfectly secret

$$\Pr[M = m \mid C = c] = \Pr[M = m]$$

The One-Time Pad

One-Time Pad Encryption Scheme

Gen: $k \leftarrow \mathcal{K}$ Enc: $c = k \oplus m$

Theorem: The OTP is perfectly secret ($\Pr[M = m \mid C = c] = \Pr[M = m]$)

$$\Pr[M = m \mid C = c]$$

The One-Time Pad

One-Time Pad Encryption Scheme

Gen: $k \leftarrow \mathcal{K}$ Enc: $c = k \oplus m$

Theorem: The OTP is perfectly secret ($\Pr[M = m \mid C = c] = \Pr[M = m]$)

$$\Pr[M = m \mid C = c] \quad = \quad \frac{\Pr[C = c \mid M = m] \cdot \Pr[M = m]}{\Pr[C = c]}$$

The One-Time Pad

One-Time Pad Encryption Scheme

Gen: $k \leftarrow \mathcal{K}$ Enc: $c = k \oplus m$

Theorem: The OTP is perfectly secret ($\Pr[M = m \mid C = c] = \Pr[M = m]$)

$$\Pr[M = m \mid C = c] = \frac{\Pr[C = c \mid M = m] \cdot \Pr[M = m]}{\Pr[C = c]}$$

$$\begin{aligned}\Pr[C = c \mid M = m] &= \Pr[\mathrm{Enc}_K(m) = c] = \Pr[m \oplus K = c] \\ &= \Pr[K = m \oplus c] = 2^{-\ell}\end{aligned}$$

The One-Time Pad

One-Time Pad Encryption Scheme

Gen: $k \leftarrow \mathcal{K}$ Enc: $c = k \oplus m$

Theorem: The OTP is perfectly secret ($\Pr[M = m \mid C = c] = \Pr[M = m]$)

$2^{-\ell}$

$$\Pr[M = m \mid C = c] = \frac{\Pr[C = c \mid M = m] \cdot \Pr[M = m]}{\Pr[C = c]}$$

$2^{-\ell}$

$$\begin{aligned}
\Pr[C = c \mid M = m] &= \Pr[\mathsf{Enc}_K(m) = c] = \Pr[m \oplus K = c] \\
&= \Pr[K = m \oplus c] = 2^{-\ell} \\
\Pr[C = c] &= \sum_{m' \in \mathcal{M}} \Pr[C = c \mid M = m'] \cdot \Pr[M = m'] \\
&= 2^{-\ell} \cdot \sum_{m' \in \mathcal{M}} \Pr[M = m'] = 2^{-\ell}
\end{aligned}$$

Limitations of the One-Time Pad

The one-time pad has some critical limitations that make it not ideal for real-world use.

Limitations of the One-Time Pad

The one-time pad has some critical limitations that make it not ideal for real-world use.

- Can only use a key k to encrypt at most one message

$c_1 = k \oplus m_1$, $c_2 = k \oplus m_2$

$$c_1 \oplus c_2 = m_1 \oplus m_2$$

Limitations of the One-Time Pad

The one-time pad has some critical limitations that make it not ideal for real-world use.

- Can only use a key k to encrypt at most one message
- Need the length of the key to be as long as the message (i.e., $|k| \geq |m|$)

Limitations of the One-Time Pad

The one-time pad has some critical limitations that make it not ideal for real-world use.

- Can only use a key k to encrypt at most one message
- Need the length of the key to be as long as the message (i.e., $|k| \geq |m|$)

Why?

Outline

Limitations of Perfect Secrecy

Theorem: If $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a perfectly secret encryption scheme with message space $\mathcal{M}$, then $|\mathcal{K}| \geq |\mathcal{M}|$.

Limitations of Perfect Secrecy

Theorem: If $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a perfectly secret encryption scheme with message space $\mathcal{M}$, then $|\mathcal{K}| \geq |\mathcal{M}|$.

Proof: (by contradiction)

- Assume that Π is perfectly secure and that $|\mathcal{K}| < |\mathcal{M}|$.

Limitations of Perfect Secrecy

Theorem: If $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a perfectly secret encryption scheme with message space $\mathcal{M}$, then $|\mathcal{K}| \geq |\mathcal{M}|$.

Proof: (by contradiction)

- Assume that Π is perfectly secure and that $|\mathcal{K}| < |\mathcal{M}|$.
- Let M be chosen uniformly over $\mathcal{M}$ and let $c \in \mathcal{C}$ be any ciphertext s.t. $\Pr[C = c] > 0$.

Limitations of Perfect Secrecy

Theorem: If $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a perfectly secret encryption scheme with message space $\mathcal{M}$, then $|\mathcal{K}| \geq |\mathcal{M}|$.

Proof: (by contradiction)

- Assume that Π is perfectly secure and that $|\mathcal{K}| < |\mathcal{M}|$.
- Let M be chosen uniformly over $\mathcal{M}$ and let $c \in \mathcal{C}$ be any ciphertext s.t. $\Pr[C = c] > 0$.
- Define $\mathcal{M}(c) = \{m \mid m = \mathsf{Dec}_k(c) \text{ for some } k \in \mathcal{K}\}$

Limitations of Perfect Secrecy

Theorem: If $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a perfectly secret encryption scheme with message space $\mathcal{M}$, then $|\mathcal{K}| \geq |\mathcal{M}|$.

Proof: (by contradiction)

- Assume that Π is perfectly secure and that $|\mathcal{K}| < |\mathcal{M}|$.
- Let M be chosen uniformly over $\mathcal{M}$ and let $c \in \mathcal{C}$ be any ciphertext s.t. $\Pr[C = c] > 0$.
- Define $\mathcal{M}(c) = \{m \mid m = \mathsf{Dec}_k(c) \text{ for some } k \in \mathcal{K}\}$
  - Observe that $|\mathcal{M}(c)| \leq |\mathcal{K}|$

Limitations of Perfect Secrecy

Theorem: If $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a perfectly secret encryption scheme with message space $\mathcal{M}$, then $|\mathcal{K}| \geq |\mathcal{M}|$.

Proof: (by contradiction)

- Assume that Π is perfectly secure and that $|\mathcal{K}| < |\mathcal{M}|$.
- Let M be chosen uniformly over $\mathcal{M}$ and let $c \in \mathcal{C}$ be any ciphertext s.t. $\Pr[C = c] > 0$.
- Define $\mathcal{M}(c) = \{m \mid m = \mathsf{Dec}_k(c) \text{ for some } k \in \mathcal{K}\}$
  - Observe that $|\mathcal{M}(c)| \leq |\mathcal{K}|$
- By assumption, $|\mathcal{K}| < |\mathcal{M}|$, so there exists $m' \in \mathcal{M}$ such that $m' \notin \mathcal{M}(c)$

Limitations of Perfect Secrecy

Theorem: If $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ is a perfectly secret encryption scheme with message space $\mathcal{M}$, then $|\mathcal{K}| \geq |\mathcal{M}|$.

Proof: (by contradiction)

- Assume that Π is perfectly secure and that $|\mathcal{K}| < |\mathcal{M}|$.
- Let M be chosen uniformly over $\mathcal{M}$ and let $c \in \mathcal{C}$ be any ciphertext s.t. $\Pr[C = c] > 0$.
- Define $\mathcal{M}(c) = \{m \mid m = \text{Dec}_k(c) \text{ for some } k \in \mathcal{K}\}$
  - Observe that $|\mathcal{M}(c)| \leq |\mathcal{K}|$
- By assumption, $|\mathcal{K}| < |\mathcal{M}|$, so there exists $m' \in \mathcal{M}$ such that $m' \notin \mathcal{M}(c)$
- But, then $\Pr[M = m' \mid C = c] = 0 \neq \Pr[M = m']$ (i.e., Π is not perfectly secret), contradiction!

Limitations of Perfect Secrecy

Theorem: If $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a perfectly secret encryption scheme with message space $\mathcal{M}$, then $|\mathcal{K}| \geq |\mathcal{M}|$.

Proof: (by contradiction)

- Assume that Π is perfectly secure and that $|\mathcal{K}| < |\mathcal{M}|$.
- Let M be chosen uniformly over $\mathcal{M}$ and let $c \in \mathcal{C}$ be any ciphertext s.t. $\Pr[C = c] > 0$.
- Define $\mathcal{M}(c) = \{m \mid m = \mathsf{Dec}_k(c) \text{ for some } k \in \mathcal{K}\}$
  - Observe that $|\mathcal{M}(c)| \leq |\mathcal{K}|$
- By assumption, $|\mathcal{K}| < |\mathcal{M}|$, so there exists $m' \in \mathcal{M}$ such that $m' \notin \mathcal{M}(c)$
- But, then $\Pr[M = m' \mid C = c] = 0 \neq \Pr[M = m']$ (i.e., Π is not perfectly secret), contradiction!

Take Away

Perfectly secure encryption must have keys as long as the message.

Outline

Proof Techniques for this Class

We will use the following proof techniques in this class:

Proof Techniques for this Class

We will use the following proof techniques in this class:

- Direct Proof

Proof Techniques for this Class

We will use the following proof techniques in this class:

- Direct Proof
- Proof by Contradiction

Proof Techniques for this Class

We will use the following proof techniques in this class:

- Direct Proof
- Proof by Contradiction
- Proof by Reduction

Proof Techniques for this Class

We will use the following proof techniques in this class:

- Direct Proof
- Proof by Contradiction
- Proof by Reduction
- Proof by Induction

Exercise

Consider a OTP encryption:

- What happens if $k = 0^\ell$?

$c = m \oplus 0^\ell = m$

Exercise

Consider a OTP encryption:

- What happens if $k = 0^\ell$?
- Is this a problem?

Exercise

Consider a OTP encryption:

- What happens if $k = 0^\ell$?
- Is this a problem?
- Suppose we modify Gen to never output $k = 0^\ell$, is this still perfectly secure?

Exercise

Consider a OTP encryption:

- What happens if $k = 0^\ell$?
- Is this a problem?
- Suppose we modify Gen to never output $k = 0^\ell$, is this still perfectly secure?
- Why or why not?

Outline

Asymptotics in Cryptography

- A function $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ is *polynomial* if there exist $c \in \mathbb{Z}^+$ such that $f(n) < n^c$ for all n

Asymptotics in Cryptography

- A function $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ is *polynomial* if there exist $c \in \mathbb{Z}^+$ such that $f(n) < n^c$ for all n
  - Interpretation: f grows no faster than a polynomial in n.

Asymptotics in Cryptography

- A function $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ is *polynomial* if there exist $c \in \mathbb{Z}^+$ such that $f(n) < n^c$ for all n
  - Interpretation: f grows no faster than a polynomial in n.
  - Examples: $f(n) = n^2$, $f(n) = 100n^3 + n + 6$

Asymptotics in Cryptography

- A function $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ is *polynomial* if there exist $c \in \mathbb{Z}^+$ such that $f(n) < n^c$ for all n
  - Interpretation: f grows no faster than a polynomial in n.
  - Examples: $f(n) = n^2$, $f(n) = 100n^3 + n + 6$
- A function $f : \mathbb{Z}^+ \to [0, 1]$ is *negligible* if *for every* polynomial p it holds that $f(n) < 1/p(n)$ for large enough n.

Asymptotics in Cryptography

- A function $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ is *polynomial* if there exist $c \in \mathbb{Z}^+$ such that $f(n) < n^c$ for all n
  - Interpretation: f grows no faster than a polynomial in n.
  - Examples: $f(n) = n^2$, $f(n) = 100n^3 + n + 6$
- A function $f : \mathbb{Z}^+ \to [0, 1]$ is *negligible* if *for every* polynomial p it holds that $f(n) < 1/p(n)$ for large enough n.
  - Interpretation: f goes to 0 faster than any inverse polynomial in n

Asymptotics in Cryptography

- A function $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ is *polynomial* if there exist $c \in \mathbb{Z}^+$ such that $f(n) < n^c$ for all n
  - Interpretation: f grows no faster than a polynomial in n.
  - Examples: $f(n) = n^2$, $f(n) = 100n^3 + n + 6$
- A function $f : \mathbb{Z}^+ \to [0, 1]$ is *negligible* if *for every* polynomial p it holds that $f(n) < 1/p(n)$ for large enough n.
  - Interpretation: f goes to 0 faster than any inverse polynomial in n
  - Examples: $f(n) = 2^{-n}, f(n) = 2^{-\log^2 n}$

$$\lim_{n \to \infty} \frac{n^2}{2^n} = 0 \qquad 2^{\log^2 n} = n^{\log n}$$

Asymptotics in Cryptography

- A function $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ is *polynomial* if there exist $c \in \mathbb{Z}^+$ such that $f(n) < n^c$ for all n
  - Interpretation: f grows no faster than a polynomial in n.
  - Examples: $f(n) = n^2$, $f(n) = 100n^3 + n + 6$
- A function $f : \mathbb{Z}^+ \to [0, 1]$ is *negligible* if *for every* polynomial p it holds that $f(n) < 1/p(n)$ for large enough n.
  - Interpretation: f goes to 0 faster than any inverse polynomial in n
  - Examples: $f(n) = 2^{-n}, f(n) = 2^{-\log^2 n}$

Computational Security

A cryptographic scheme is *computationally secure* if any *probabilistic polynomial time (PPT) adversary* only breaks security with at most a *negligible probability*.

Necessity of Relaxations

Consider the case of private-key encryption with $|\mathcal{K}| < |\mathcal{M}|$:

Necessity of Relaxations

Consider the case of private-key encryption with $|\mathcal{K}| < |\mathcal{M}|$:

- Adversary Runtime:

Necessity of Relaxations

Consider the case of private-key encryption with $|\mathcal{K}| < |\mathcal{M}|$:

- Adversary Runtime:
  Suppose $\mathcal{A}$ can run in unbounded time.

Necessity of Relaxations

Consider the case of private-key encryption with $|\mathcal{K}| < |\mathcal{M}|$:

- Adversary Runtime:
  Suppose $\mathcal{A}$ can run in unbounded time. Given a ciphertext c, try to decrypt with all $k \in \mathcal{K}$, some m must be missing.

$\mathcal{A}$ run in time 2^{ℓ}

Necessity of Relaxations

Consider the case of private-key encryption with $|\mathcal{K}| < |\mathcal{M}|$:

- Adversary Runtime:
  Suppose $\mathcal{A}$ can run in unbounded time. Given a ciphertext c, try to decrypt with all $k \in \mathcal{K}$, some m must be missing.
- Adversary Success Probability:
  Consider an $\mathcal{A}$ that learns pairs $(m_1, c_1), \ldots, (m_\ell, c_\ell)$.

Necessity of Relaxations

Consider the case of private-key encryption with $|\mathcal{K}| < |\mathcal{M}|$:

- Adversary Runtime:
  Suppose $\mathcal{A}$ can run in unbounded time. Given a ciphertext c, try to decrypt with all $k \in \mathcal{K}$, some m must be missing.
- Adversary Success Probability:
  Consider an $\mathcal{A}$ that learns pairs $(m_1, c_1), \ldots, (m_\ell, c_\ell)$. $\mathcal{A}$ can just guess $k \leftarrow \mathcal{K}$ and check where $\mathsf{Dec}_k(c_i) = m_i$ for all i (i.e., k is the correct key). $\mathcal{A}$ can guess correct k with probability $1/|\mathcal{K}| > 0$.

Necessity of Relaxations

Consider the case of private-key encryption with $|\mathcal{K}| < |\mathcal{M}|$:

- Adversary Runtime:
  Suppose $\mathcal{A}$ can run in unbounded time. Given a ciphertext c, try to decrypt with all $k \in \mathcal{K}$, some m must be missing.
- Adversary Success Probability:
  Consider an $\mathcal{A}$ that learns pairs $(m_1, c_1), \ldots, (m_\ell, c_\ell)$. $\mathcal{A}$ can just guess $k \leftarrow \mathcal{K}$ and check where $\mathsf{Dec}_k(c_i) = m_i$ for all i (i.e., k is the correct key). $\mathcal{A}$ can guess correct k with probability $1/|\mathcal{K}| > 0$.

Computational Security

A cryptographic scheme is *computationally secure* if any *probabilistic polynomial time (PPT) adversary* only breaks security with at most a *negligible probability*.

Why This Choice of Relaxation?

Computational Security

A cryptographic scheme is *computationally secure* if any *probabilistic polynomial time (PPT) adversary* only breaks security with at most a *negligible probability*.

Why This Choice of Relaxation?

- Efficient = probabilistic polynomial time (PPT)
  - Standard from algorithms / complexity theory

Computational Security

A cryptographic scheme is *computationally secure* if any *probabilistic polynomial time (PPT) adversary* only breaks security with at most a *negligible probability*.

Why This Choice of Relaxation?

- Efficient = probabilistic polynomial time (PPT)
  - Standard from algorithms / complexity theory
- Convenient closure properties:

Computational Security

A cryptographic scheme is *computationally secure* if any *probabilistic polynomial time (PPT) adversary* only breaks security with at most a *negligible probability*.

Why This Choice of Relaxation?

- Efficient = probabilistic polynomial time (PPT)
  - Standard from algorithms / complexity theory
- Convenient closure properties:
  - $\text{poly}(n) \cdot \text{poly}(n) = \text{poly}(n)$
    A PPT algorithm making calls to PPT subroutines is PPT

Computational Security

A cryptographic scheme is *computationally secure* if any *probabilistic polynomial time (PPT) adversary* only breaks security with at most a *negligible probability*.

Why This Choice of Relaxation?

- Efficient = probabilistic polynomial time (PPT)
  - Standard from algorithms / complexity theory
- Convenient closure properties:
  - $\text{poly}(n) \cdot \text{poly}(n) = \text{poly}(n)$
    A PPT algorithm making calls to PPT subroutines is PPT
  - $\text{poly}(n) \cdot \text{negl}(n) = \text{negl}(n)$
    Poly many calls to subroutines with negligible success probability, have negligible success probability

Computational Security

A cryptographic scheme is *computationally secure* if any *probabilistic polynomial time (PPT) adversary* only breaks security with at most a *negligible probability*.

Redefining Encryption Functionality

Private-key (symmetric-key) encryption scheme:

- Gen: Outputs randomly chosen key k
- $\mathsf{Enc}(k, m) : c \leftarrow \mathsf{Enc}_k(m)$
- $\mathsf{Dec}(k, c) : m = \mathsf{Dec}_k(c)$

Correctness

For all k output by Gen and all messages m, $\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$

Redefining Encryption Functionality

Private-key (symmetric-key) encryption scheme:

- $\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}$ run in polynomial time (in their input size)
- Gen: Outputs randomly chosen key k
- $\mathsf{Enc}(k, m) : c \leftarrow \mathsf{Enc}_k(m)$
- $\mathsf{Dec}(k, c) : m = \mathsf{Dec}_k(c)$

Correctness

For all k output by Gen and all messages m, $\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$

Redefining Encryption Functionality

$1^n = \overbrace{1111111}^{n \text{ times}}$

Private-key (symmetric-key) encryption scheme:

- Gen, Enc, Dec run in polynomial time (in their input size)
- Gen: $k \leftarrow \text{Gen}(1^n)$ – n is called the *security parameter*
- $\text{Enc}(k, m) : c \leftarrow \text{Enc}_k(m)$
- $\text{Dec}(k, c) : m = \text{Dec}_k(c)$

Correctness

For all k output by Gen and all messages m, $\text{Dec}_k(\text{Enc}_k(m)) = m$

$||n|| = \log n$

Redefining Encryption Functionality

Private-key (symmetric-key) encryption scheme:

- Gen, Enc, Dec run in polynomial time (in their input size)
- Gen: $k \leftarrow \mathsf{Gen}(1^n)$ – n is called the *security parameter*
- $\mathsf{Enc}(k, m) : c \leftarrow \mathsf{Enc}_k(m)$ for $m \in \{0,1\}^*$
- $\mathsf{Dec}(k, c) : m = \mathsf{Dec}_k(c)$

Correctness

For all k output by Gen and all messages m, $\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$

Redefining Encryption Functionality

Private-key (symmetric-key) encryption scheme:

- $\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}$ run in polynomial time (in their input size)
- Gen: $k \leftarrow \mathsf{Gen}(1^n)$ – n is called the *security parameter*
- $\mathsf{Enc}(k, m) : c \leftarrow \mathsf{Enc}_k(m)$ for $m \in \{0, 1\}^*$
- $\mathsf{Dec}(k, c) : m = \mathsf{Dec}_k(c)$

Correctness

For all k output by Gen and all messages m, $\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$

Redefining Encryption Functionality

Private-key (symmetric-key) encryption scheme:

- $\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}$ run in polynomial time (in their input size)
- Gen: $k \leftarrow \mathsf{Gen}(1^n)$ – n is called the *security parameter*
- $\mathsf{Enc}(k, m) : c \leftarrow \mathsf{Enc}_k(m)$ for $m \in \{0,1\}^*$
- $\mathsf{Dec}(k, c) : m = \mathsf{Dec}_k(c)$

Correctness

For all n, for all k output by $\mathsf{Gen}(1^n)$ and all messages $m \in \{0,1\}^*$, $\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$

Redefining Encryption Security

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme. Consider the following game between an adversary $\mathcal{A}$ and a challenger:

$\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi}$

- $\mathcal{A}$ outputs two messages $m_0, m_1 \in \mathcal{M}$
- The challenger chooses $k \leftarrow \mathsf{Gen}$, $b \leftarrow \{0, 1\}$, computes $c \leftarrow \mathsf{Enc}_k(m_b)$ and gives c to $\mathcal{A}$
- $\mathcal{A}$ outputs a guess bit b'
- We say that $\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi} = 1$ (i.e., $\mathcal{A}$ wins) if $b' = b$.

Definition: An encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ is *perfectly indistinguishable* if for all $\mathcal{A}$ it holds that

$$\Pr[\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi} = 1] = 1/2$$

Redefining Encryption Security

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme. Consider the following game between an adversary $\mathcal{A}$ and a challenger:

$\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi}(n)$

- $\mathcal{A}$ outputs two messages $m_0, m_1 \in \mathcal{M}$
- The challenger chooses $k \leftarrow \mathsf{Gen}$, $b \leftarrow \{0,1\}$, computes $c \leftarrow \mathsf{Enc}_k(m_b)$ and gives c to $\mathcal{A}$
- $\mathcal{A}$ outputs a guess bit b'
- We say that $\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi} = 1$ (i.e., $\mathcal{A}$ wins) if $b' = b$.

Definition: An encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ is *perfectly indistinguishable* if for all $\mathcal{A}$ it holds that

$$\Pr[\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi} = 1] = 1/2$$

Redefining Encryption Security

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme. Consider the following game between an adversary $\mathcal{A}$ and a challenger:

$\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi}(n)$

- $\mathcal{A}$ outputs two messages m_0, m_1 such that $|m_0| = |m_1|$.
- The challenger chooses $k \leftarrow \mathsf{Gen}$, $b \leftarrow \{0,1\}$, computes $c \leftarrow \mathsf{Enc}_k(m_b)$ and gives c to $\mathcal{A}$
- $\mathcal{A}$ outputs a guess bit b'
- We say that $\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi} = 1$ (i.e., $\mathcal{A}$ wins) if $b' = b$.

Definition: An encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ is *perfectly indistinguishable* if for all $\mathcal{A}$ it holds that

$$\Pr[\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi} = 1] = 1/2$$

Redefining Encryption Security

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme. Consider the following game between an adversary $\mathcal{A}$ and a challenger:

$\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi}(n)$

- $\mathcal{A}$ outputs two messages m_0, m_1 such that $|m_0| = |m_1|$.
- The challenger chooses $k \leftarrow \mathsf{Gen}(1^n)$, $b \leftarrow \{0,1\}$, computes $c \leftarrow \mathsf{Enc}_k(m_b)$ and gives c to $\mathcal{A}$
- $\mathcal{A}$ outputs a guess bit b'
- We say that $\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi} = 1$ (i.e., $\mathcal{A}$ wins) if $b' = b$.

Definition: An encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ is *perfectly indistinguishable* if for all $\mathcal{A}$ it holds that

$$\Pr[\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi} = 1] = 1/2$$

Redefining Encryption Security

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme. Consider the following game between an adversary $\mathcal{A}$ and a challenger:

$\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi}(n)$

- $\mathcal{A}$ outputs two messages m_0, m_1 such that $|m_0| = |m_1|$.
- The challenger chooses $k \leftarrow \mathsf{Gen}(1^n)$, $b \leftarrow \{0, 1\}$, computes $c \leftarrow \mathsf{Enc}_k(m_b)$ and gives c to $\mathcal{A}$
- $\mathcal{A}$ outputs a guess bit b'
- We say that $\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi} = 1$ (i.e., $\mathcal{A}$ wins) if $b' = b$.

Definition: An encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ is *perfectly indistinguishable* if for all $\mathcal{A}$ it holds that

$$\Pr[\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi} = 1] = 1/2$$

Redefining Encryption Security

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme. Consider the following game between an adversary $\mathcal{A}$ and a challenger:

$\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi}(n)$

- $\mathcal{A}$ outputs two messages m_0, m_1 such that $|m_0| = |m_1|$.
- The challenger chooses $k \leftarrow \mathsf{Gen}(1^n)$, $b \leftarrow \{0,1\}$, computes $c \leftarrow \mathsf{Enc}_k(m_b)$ and gives c to $\mathcal{A}$
- $\mathcal{A}$ outputs a guess bit b'
- We say that $\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi}(n) = 1$ (i.e., $\mathcal{A}$ wins) if $b' = b$.

Definition: An encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ is *perfectly indistinguishable* if for all $\mathcal{A}$ it holds that

$$\Pr[\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi} = 1] = 1/2$$

Redefining Encryption Security

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme. Consider the following game between an adversary $\mathcal{A}$ and a challenger:

$\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi}(n)$

- $\mathcal{A}$ outputs two messages m_0, m_1 such that $|m_0| = |m_1|$.
- The challenger chooses $k \leftarrow \mathsf{Gen}(1^n)$, $b \leftarrow \{0,1\}$, computes $c \leftarrow \mathsf{Enc}_k(m_b)$ and gives c to $\mathcal{A}$
- $\mathcal{A}$ outputs a guess bit b'
- We say that $\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi}(n) = 1$ (i.e., $\mathcal{A}$ wins) if $b' = b$.

Definition: An encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ has *indistinguishable encryptions in the presence of an eavesdropper* if for all PPT $\mathcal{A}$ it holds that

$$\Pr[\mathsf{PrivK}^{eav}_{\mathcal{A},\Pi}(n) = 1] \leq 1/2 + \mathsf{negl}(n)$$

How to Construct

- Recall that we encrypted by computing $\mathsf{Enc}_k(m) = m \oplus k$
- But, if $|k| < |m|$, this is not secure

How to Construct

- Recall that we encrypted by computing $\mathsf{Enc}_k(m) = m \oplus k$
- But, if $|k| < |m|$, this is not secure

Key Idea

What if we had a way to stretch key k into something longer that still looked random?