CSCI 1510

- Limitations of Perfect Security
- Definition of Computational Security : Concrete vs. Asymptotic
- Definition of Semantic Security
- Pseudorandom Generator (PRG)

Last Lecture

Perfectly secure symmetric-key encryption

- Definitions 1, 2, 3

  $\forall m_0, m_1 \in \mathcal{M}, \forall c \in \mathcal{C}$:

  $\Pr[\mathrm{Enc}_K(m_0) = c] = \Pr[\mathrm{Enc}_K(m_1) = c]$

- Construction: OTP

- Limitations: $|\mathcal{M}| \leq |\mathcal{K}|$.

How to relax the security definition?

Limitations of Perfect Security

Thm If $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ is a perfectly secure encryption scheme with message space $\mathcal{M}$ & key space $\mathcal{K}$, then $|\mathcal{M}| \leq |\mathcal{K}|$.

Proof: Assume $|\mathcal{K}| < |\mathcal{M}|$.

Pick an arbitrary $c \in \mathcal{C}$ where $\Pr[C = c] > 0$.

$\mathcal{M}(c) := \{ m \mid m = \text{Dec}_k(c) \text{ for some } k \in \mathcal{K} \}$.

$|\mathcal{M}(c)| \leq |\mathcal{K}| < |\mathcal{M}|$.

$\exists m' \in \mathcal{M}$ s.t. $m' \notin \mathcal{M}(c)$.

$\Pr[M = m' \mid C = c] = 0 \neq \Pr[M = m']$.

Computational Security

Perfect Security:

① Absolutely no information is leaked

② $\mathcal{A}$ has unlimited computational power

Relaxation (Practical Purpose):

① "Tiny" information can be leaked

② $\mathcal{A}$ has limited computational power

How to formalize?

Computational Security

- Concrete Approach:

  A scheme is (t, ε)-secure if $\forall A$ running in time $\leq t$ succeeds in breaking the scheme with probability $\leq \varepsilon$.

Example: $(2^{128}, 2^{-60})$-secure encryption scheme.

What's the problem?

Computational Security

- Asymptotic Approach:

  Introduce a security parameter n (public) — λ, measuring how "hard" it is for $\mathcal{A}$ to break the scheme.

  All honest parties run in time $\mathrm{poly}(n)$.

  Security can be tuned by changing n.

A scheme is (t, ε)-secure if $\forall \mathcal{A}$ running in time $\mathrm{poly}(n)$ succeeds in breaking the scheme with probability $\mathrm{negl}(n)$.

(t: $\mathrm{poly}(n)$; ε: "negligible" in n)

Polynomial & Negligible

"Efficient": Probabilistic polynomial time (PPT)

Def A function $f: \mathbb{N} \to \mathbb{R}^+$ is polynomial if

$$\exists c \in \mathbb{N} \text{ s.t. } f(n) \in O(n^c)$$

Def A function $f: \mathbb{N} \to \mathbb{R}^+$ is negligible if

$$\forall \text{polynomial } p, \ \exists N \in \mathbb{N} \text{ s.t. } \forall n > N, \ f(n) < \frac{1}{p(n)}.$$

$$\Longleftrightarrow \forall c \in \mathbb{N}, \ f(n) \in o(n^{-c})$$

Examples: 2^{-n}, $2^{-\sqrt{n}}$, $n^{-\log n}$

Exercise: Is this a negligible function?

$$f(n) := \begin{cases} 2^{-n} & \text{if } n \text{ is even} \\ 1/n^2 & \text{if } n \text{ is odd} \end{cases}$$

Negligible Function

Def A function $f: \mathbb{N} \to \mathbb{R}^+$ is negligible if
$\forall$ polynomial p, $\exists N \in \mathbb{N}$ s.t. $\forall n > N$, $f(n) < \frac{1}{p(n)}$.

Claim 1 If f, g are negligible functions, then $f+g$ is also negligible.

Claim 2 If f is negligible, p is polynomial, then $f \cdot p$ is also negligible.

Corollary If g is non-negligible, p is polynomial, then $\frac{g}{p}$ is also non-negligible.

Concrete $\rightarrow$ Asymptotic

A scheme is (t, ε)-secure if $\forall \mathcal{A}$ running in time $\leq t$ succeeds in breaking the scheme with probability $\leq \varepsilon$.

$$\Downarrow$$

A scheme is secure if $\forall$ PPT $\mathcal{A}$ succeeds in breaking the scheme with probability $\leq$ negligible.

Computationally Secure Encryption

- **Syntax:**

  A symmetric-key encryption scheme is defined by PPT algorithms (Gen, Enc, Dec):

  $$k \leftarrow \text{Gen}(1^n)$$

  $$c \leftarrow \text{Enc}_k(m) \qquad m \in \{0,1\}^*$$

  $$m/\perp := \text{Dec}_k(c)$$

- **Correctness:** $\forall n$, $\forall k$ output by $\text{Gen}(1^n)$, $\forall m \in \{0,1\}^*$

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

Computationally Secure Encryption

Def 1 A symmetric-key encryption scheme (Gen, Enc, Dec)

is **semantically secure** if $\forall$ **PPT** $\mathcal{A}$, $\exists$ negligible function $\varepsilon(\cdot)$ s.t.

computationally indistinguishable

$$\Pr[b = b'] \le \frac{1}{2} + \varepsilon(n)$$

$C(1^n)$		$\mathcal{A}(1^n)$
	$\xleftarrow[\lvert m_0\rvert = \lvert m_1\rvert]{m_0,\, m_1 \in \{0,1\}^*}$	
$k \leftarrow \mathrm{Gen}$		
$b \xleftarrow{\$} \{0,1\}$		
$c \leftarrow \mathrm{Enc}_k(m_b)$		
	$\xrightarrow{\quad c \quad}$	
		output b'

Computationally Secure Encryption

Def 2 A symmetric-key encryption scheme (Gen, Enc, Dec) is semantically secure if $\forall$ PPT $\mathcal{A}$, $\exists$ negligible function $\varepsilon(\cdot)$ s.t.

computationally indistinguishable

$$\left| \Pr[b'=1 \mid b=0] - \Pr[b'=1 \mid b=1] \right| \le \varepsilon(n)$$

$C(1^n, b)$		$\mathcal{A}(1^n)$
	$\xleftarrow[\lvert m_0\rvert = \lvert m_1\rvert]{m_0, m_1 \in \{0,1\}^*}$	
$k \leftarrow \text{Gen}$		
$c \leftarrow \text{Enc}_k(m_b)$		
	$\xrightarrow{c}$	
		output b'

Computationally Secure Encryption

Def 1 A symmetric-key encryption scheme (Gen, Enc, Dec) is semantically secure if $\forall$ PPT $\mathcal{A}$:

$$\Pr[b = b'] \le \frac{1}{2} + \text{negl}(n) \qquad \text{in Game 1.}$$

$$\Updownarrow$$

Def 2 $$\left| \Pr[b' = 1 \mid b = 0] - \Pr[b' = 1 \mid b = 1] \right| \le \text{negl}(n) \qquad \text{in Game 2.}$$

Constructing Secure Encryption

Pseudorandom Generator (PRG)

$\Downarrow$

Semantically Secure Encryption

(Pseudo)randomness

What does it mean to be random?

Is this string random?

011011010110001

010101010101010

What does it mean to be pseudorandom?

Pseudorandomness

- Concrete Definition:

  D: a distribution over n-bit strings.

  D is (t, ε)-pseudorandom if $\forall \mathcal{A}$ running in time $\le t$,

$$\left| \Pr_{x \leftarrow D} [\mathcal{A}(x) = 1] - \Pr_{x \leftarrow U_n} [\mathcal{A}(x) = 1] \right| \le \varepsilon.$$

- Asymptotic Definition:

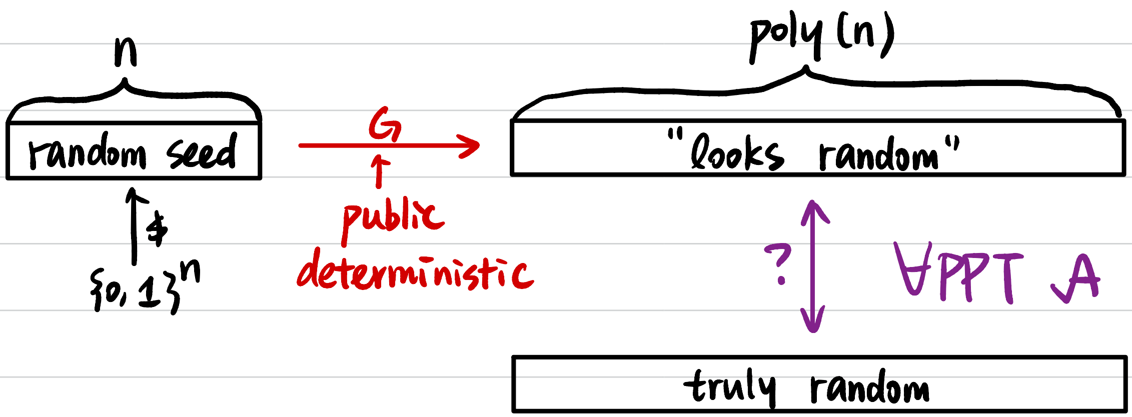
  $D = \{D_1, D_2, \cdots\}$ an ensemble of distributions,

  D_n: a distribution over n-bit string.

  D is pseudorandom if $\forall$ PPT $\mathcal{A}$, $\exists$ negligible function $\varepsilon(\cdot)$ s.t.

$$\left| \Pr_{x \leftarrow D_n} [\mathcal{A}(x) = 1] - \Pr_{x \leftarrow U_n} [\mathcal{A}(x) = 1] \right| \le \varepsilon(n).$$

Pseudorandom Generator (PRG)



$$G: \{0,1\}^n \to \{0,1\}^{\ell(n)} \qquad \ell(n) > n$$

Pseudorandom Generator (PRG)

$G: \{0,1\}^n \to \{0,1\}^{\ell(n)} \qquad \ell(n) > n$

Def 1 G is a pseudorandom generator (PRG) if $\forall$ PPT $\mathcal{A}$, $\exists$ negligible function $\mathrm{negl}(\cdot)$ s.t.

$$\left| \Pr_{s \leftarrow U_n}[\mathcal{A}(G(s)) = 1] - \Pr_{x \leftarrow U_{\ell(n)}}[\mathcal{A}(x) = 1] \right| \le \mathrm{negl}(n)$$

Pseudorandom Generator (PRG)

$G: \{0,1\}^n \to \{0,1\}^{\ell(n)}$ $\quad \ell(n) > n$

Def 2 G is a pseudorandom generator (PRG) if $\forall$ PPT $\mathcal{A}$, $\exists$ negligible function $\mathrm{negl}(\cdot)$ s.t.

$$\Pr[b = b'] \le \frac{1}{2} + \mathrm{negl}(n)$$

$C(1^n)$		$\mathcal{A}(1^n)$
$b \xleftarrow{\$} \{0,1\}$		
If $b=0$, then $s \leftarrow U_n$, $x := G(s)$		
If $b=1$, then $x \leftarrow U_{\ell(n)}$		
	$\xrightarrow{\quad x \quad}$	
		output b'

What if $\mathcal{A}$ is computationally unbounded?

Exercises

$$G(s) = s \,\|\, \bigoplus_{i=1}^{n} s_i$$

Concatenation

Is G a secure PRG?

Exercises

Let $G: \{0,1\}^n \to \{0,1\}^{2n}$ be a PRG.

Construct $H: \{0,1\}^n \to \{0,1\}^{2n}$ as $H(s) := G(s) \oplus (s \| 0^n)$.

Is H necessarily a PRG?