Cryptography Definitions

1 Fundamentals

1.1 Notation

- For two distributions D_0, D_1 we use $D_0 \equiv D_1$ to indicate that the distributions are *identical*.
- For two ensembles X_0, X_1, we use $X_0 \overset{c}{\approx} X_1$ to indicate that the ensembles are *computationally indistinguishable*.

1.2 Negligible Functions

A function $\nu : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is *negligible* if $\forall c \in \mathbb{Z}_{\geq 0}$, $\exists \Lambda \in \mathbb{N}$ such that $\forall \lambda \in \mathbb{N}$ and $\lambda > \Lambda$, it holds that

$$\nu(\lambda) \leq \frac{1}{\lambda^c}$$

1.3 Ensembles

Let $\mathcal{I}$ be a countable index set. An *ensemble* indexed by $\mathcal{I}$ is a sequence of distributions $\{X_i\}_{i \in \mathcal{I}}$.

Nearly always $\mathcal{I}$ will be the set of natural numbers $\mathbb{N}$.

1.4 Computational Indistinguishability

Two probability ensembles $X = \{X_i\}_{i \in \mathbb{N}}$ and $Y = \{Y_i\}_{i \in \mathbb{N}}$ are *computationally indistinguishable* if for every non-uniform PPT adversary $\mathcal{A}$, there exists a negligible function $\nu(\lambda)$ such that for all $\lambda \in \mathbb{N}$:

$$\left|\Pr_{x \leftarrow X_\lambda}\left[\mathcal{A}(1^\lambda, x) = 1\right] - \Pr_{y \leftarrow Y_\lambda}\left[\mathcal{A}(1^\lambda, y) = 1\right]\right| \leq \nu(\lambda)$$

where the probability is over sampling from the distributions X_λ and Y_λ, and the randomness of $\mathcal{A}$.

1.5 Hybrid Lemma

Let λ be the security parameter and $n := n(\lambda)$ be a polynomial. If $X_1 \dots X_n$ are probability ensembles such that for all $i \in \{0, \dots, n-1\}$ $X_i \overset{c}{\approx} X_{i+1}$, then $X_1 \overset{c}{\approx} X_n$.

2 Primitives

2.1 Encryption Scheme

An encryption scheme consists of three possibly probabilistic algorithms:

- $\mathsf{KeyGen}() \to k$ outputs a key $k \in \mathcal{K}$.
- $\mathsf{Enc}(k, m) \to \mathsf{ct}$ takes key k and message $m \in \mathcal{M}$ and outputs ciphertext $\mathsf{ct} \in \mathcal{C}$.
- $\mathsf{Dec}(k, \mathsf{ct}) \to m$ takes key k and ciphertext ct and outputs message m.

An encryption scheme must be *correct*, i.e. it must be the case that for all messages $m \in \mathcal{M}$ and keys $k \in \mathcal{K}$,

$$\Pr[\mathsf{Dec}(k, \mathsf{Enc}(k, m)) = m] = 1$$

2.2 Pseudorandom Generator

A *deterministic* algorithm G is called a *pseudorandom generator* if:

- G can be computed in polynomial time
- On input any $s \in \{0,1\}^\lambda$, G outputs a $\ell(\lambda)$-bit string such that $\ell(\lambda) > \lambda$
- $\left\{G(s) : s \leftarrow\!\$\ \{0,1\}^\lambda\right\} \overset{c}{\approx} \left\{r : r \leftarrow\!\$\ \{0,1\}^{\ell(\lambda)}\right\}$

The *stretch* of G is defined as $\ell(\lambda) - \lambda$.

There exist PRGs with any polynomial amount of stretch.

2.3 Pseudorandom Function

Let X and Y be sets, and let $\mathsf{Funs}[X, Y]$ be the set of all functions $F : X \to Y$.

For a family of functions $\{F_k\}_{k \in \{0,1\}^\lambda}$ where $F_k : X \to Y$ for all k, define the following two games (indexed by a bit b) between an adversary and a challenger:

Game b:

- If $b = 0$, the challenger samples $k \leftarrow\!\$\ \{0,1\}^\lambda$ and sets $f := F_k$.
- If $b = 1$, the challenges samples $f \leftarrow\!\$\ \mathsf{Funs}[X, Y]$.

- The adversary $\mathcal{A}$ submits a series of queries, where each query is a value $x_i \in X$. For each, the challenger computes $y_i := f(x_i)$ and returns y_i to $\mathcal{A}$.
- The adversary $\mathcal{A}$ outputs a bit b'.

Let W_b be the event that $\mathcal{A}$ outputs 1 in **Game** b.

A family of functions $\{F_k\}_{k\in\{0,1\}^\lambda}$ where $F_k : X \to Y$ for all k is *pseudorandom* if:

- $F_k(x)$ can be computed in polynomial time
- For all non-uniform PPT adversaries $\mathcal{A}$, there exists a negligible function $\nu(\lambda)$ such that $\forall\lambda \in N$:

$$|\Pr[W_0] - \Pr[W_1]| \leq \nu(\lambda)$$

3 Properties

3.1 One-Time Uniform Ciphertext Security

An *encryption scheme* satisfies *one-time uniform ciphertext security* if $\forall m \in \mathcal{M}$,

$$D_0 = \left\{ \mathsf{ct} : \begin{matrix} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m) \end{matrix} \right\} \equiv D_1 = \{\mathsf{ct} : \mathsf{ct} \leftarrow\!\!\$\, \mathcal{C}\}$$

3.2 One-Time Perfect Security

An *encryption scheme* satisfies *one-time perfect security* if $\forall m_0, m_1 \in \mathcal{M}$,

$$D_0 = \left\{ \mathsf{ct} : \begin{matrix} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0) \end{matrix} \right\} \equiv D_1 = \left\{ \mathsf{ct} : \begin{matrix} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1) \end{matrix} \right\}$$

3.3 One-Time Computational Security

An *encryption scheme* satisfies *one-time computational security* if $\forall m_0, m_1 \in \mathcal{M}$,

$$D_0 = \left\{ \mathsf{ct} : \begin{matrix} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_0) \end{matrix} \right\} \overset{\text{c}}{\approx} D_1 = \left\{ \mathsf{ct} : \begin{matrix} k \leftarrow \mathsf{KeyGen}() \\ \mathsf{ct} \leftarrow \mathsf{Enc}(k, m_1) \end{matrix} \right\}$$

3.4 Multi-Message Security

An *encryption scheme* satisfies *multi-message security* if $\forall\{(m_0^i, m_1^i)\}_{i=1}^{q(\lambda)}$ where $q(\lambda)$ is a polynomial:

$$D_0 = \left\{ \vec{\mathsf{ct}} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \vec{\mathsf{ct}} \leftarrow \left\{\mathsf{Enc}(k, m_0^i)\right\}_{i=1}^{q(\lambda)} \end{array} \right\} \overset{c}{\approx} D_1 = \left\{ \vec{\mathsf{ct}} : \begin{array}{c} k \leftarrow \mathsf{KeyGen}() \\ \vec{\mathsf{ct}} \leftarrow \left\{\mathsf{Enc}(k, m_1^i)\right\}_{i=1}^{q(\lambda)} \end{array} \right\}$$

4 Constructions

4.1 One-Time Pad

One-time Pad is an *encryption scheme* for $\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^\lambda$ defined as follows:

- $\mathsf{KeyGen}() : k \leftarrow\!\!\$\ \{0,1\}^\lambda$
- $\mathsf{Enc}(k, m) : \mathsf{ct} := k \oplus m$
- $\mathsf{Dec}(k, ct) : m := k \oplus \mathsf{ct}$

One-Time pad satisfies *one-time uniform ciphertext security.*

4.2 Pseudorandom One-Time Pad

Let λ be the security parameter and $\ell(\lambda)$ be a polynomial. Let G be a *pseudorandom generator* with stretch $\ell(\lambda) - \lambda$. Below is an *encryption scheme* for $\mathcal{K} = \{0,1\}^\lambda$ and $\mathcal{M} = \mathcal{C} = \{0,1\}^{\ell(\lambda)}$.

- $\mathsf{KeyGen}(1^\lambda) : k \leftarrow\!\!\$\ \{0,1\}^\lambda$
- $\mathsf{Enc}(k, m) : \mathsf{ct} := G(k) \oplus m$
- $\mathsf{Dec}(k, \mathsf{ct}) : m := G(k) \oplus \mathsf{ct}$

Pseudorandom one-time pad satisfies *one-time computational security.*