Lecture 3

Reading: Katz-Lindell chapters 2 & 3

Today: Encryption schemes

- One Time Pad (OTP)
- Security: one-time vs. many-time security
- Security against chosen plaintext attacks (CPA-security)
- Impossibility ☹
- Overcoming the impossibility by relying on hardness assumptions ☺
- Using hardness to generate randomness:
  - Define Pseudo Random Functions (PRFs) ← in practice AES is used as a PRF, AES will be covered next lecture
- Use PRF to construct a CPA secure encryption scheme

One-Time Pad [Gilbert Vernam 1917]

Syntax of encryption scheme: 3 probabilistic polynomial time (PPT) algorithms (Gen, Enc, Dec)

- $\text{Gen}(1^n)$ outputs a secret key K (must be prob.)
  security parameter
- Enc takes as input a secret key K & a msg $m \in \mathcal{M}$ and outputs a ciphertext c
- Dec takes as input a key K and a chiphertext c and outputs a msg m.

Correctness: $\forall n \in \mathbb{N} \quad \forall m \in \mathcal{M}$

$$\Pr_{K \leftarrow \text{Gen}(1^n)}[\text{Dec}(K, \text{Enc}(K,m)) = m] = 1$$

One-Time Pad:

- $\text{Gen}(1^n)$ outputs a random secret key $K \leftarrow \{0,1\}^n$
- $\text{Enc}(K,m) = K \oplus m$ (XOR coordinate-wise)
- $\text{Dec}(K,C) = K \oplus C$

msg space $\mathcal{M} = \{0,1\}^n$

Example: $K = (0\,1\,1\,0\,1\,0)$

$m = (1\,0\,1\,0\,0\,1)$ $\oplus$

$C = \text{Enc}(K,m) = (1\,1\,0\,0\,1\,1)$

$\text{Dec}(K,C) = (1\,0\,1\,0\,0\,1)$ ✓

Correctness: $\forall K \in \{0,1\}^n \quad \forall m \in \{0,1\}^n$

$$\text{Dec}(K, \text{Enc}(K,m)) = K \oplus (K \oplus m) = (K \oplus K) \oplus m = 0 \oplus m = m$$ ✓

Security: Perfect security!

$\underline{\forall m} \in \{0,1\}^n \qquad Enc(k,m) \equiv U_{\{0,1\}^n}$ ☺

$$\Rightarrow \Big(m,\ Enc(k,m)\Big) \equiv \Big(m,\ U_{\{0,1\}^n}\Big)$$

Even if m is known the ciphertext $Enc(k,m)$ is random!

This may seem to be a strong requirement, after all m is not known...

However, the adv may have some information about m (such as the header).

Also, we do not want to assume that m is drawn from some distribution.

This seems to be a very strong security! ← Note: $Enc(k,m)$ does not hide $|m|$ ← length of m.

This is somewhat inherent.

However, it is only one-time secure.

Namely:

$$\Big(m_1, m_2, \overset{k\oplus m_1}{Enc(k,m_1)}, \overset{k\oplus m_2}{Enc(k,m_2)}\Big) \not\equiv$$

$$(m_1, m_2, U_1, U_2)$$

In particular, given $Enc(k,m_1)$, $Enc(k,m_2)$ one can learn $m_1 \oplus m_2$.

Goal: Many-time secure encryption scheme. $\forall t\ \forall m_1,\dots m_t \in \mathcal{M}$

$$\Big(m_1,\dots m_t,\ Enc(k,m_1),\dots, Enc(k,m_t)\Big) \equiv$$

$$(m_1,\dots m_t,\ U_1\ \dots\ U_t)$$

Impossible!

① Intuitively, $Enc(k,m_1),\dots, Enc(k,m_t)$ give too much information about K (unless $|K|$ grows with t)

② Also, impossible if Enc is a deterministic function since then

$$\Big(m, m, Enc(k,m), Enc(k,m)\Big) \not\equiv$$

$$(m, m, U_1, U_2)$$

since $Enc(k,m) = Enc(k,m)$.

To overcome the latter impossibility result we use randomized encryption

To overcome the first impossibility result we rely on hardness assumptions

We cannot get many-time security against an all powerful adversary,

but we can get many-time security against a bounded (i.e., poly-time) adversary!

modern cryptography!

We next define many-time security against a poly-time adv. known as security against Chosen Plaintext Attacks (CPA)

<u>Definition</u>: An encryption scheme (Gen, Enc, Dec) is CPA secure if

$\forall n \in \mathbb{N} \quad \forall t = poly(n) \quad \forall m_1, \ldots, m_t \in \mathcal{M}$

$$\left(Enc(K, m_1), \ldots, Enc(K, m_t)\right) \cong \left(Enc(K, U_1), \ldots, Enc(K, U_t)\right)$$

computationally indistinguishable.

where $K \leftarrow Gen(1^n)$, $U_1 \ldots U_t \leftarrow \mathcal{M}$

Simplified Version!

In the typical definition each m_i can be adaptively and adversarially chosen after seeing $Enc(K, m_1), \ldots, Enc(K, m_{i-1})$

<u>Definition</u>: Two distribution ensembles $\{A_n\}_{n \in \mathbb{N}}$ & $\{B_n\}_{n \in \mathbb{N}}$ are computationally indistinguishable if $\forall$ PPT distinguisher $\mathcal{D}$ $\exists$ negligible function $\mu : \mathbb{N} \to [0,1]$

s.t. $\forall n \in \mathbb{N}$

$$\left| \Pr_{a \leftarrow A_n}[\mathcal{D}(a) = 1] - \Pr_{b \leftarrow B_n}[\mathcal{D}(b) = 1] \right| \leq \mu(k)$$

<u>Definition</u>: A function $\mu : \mathbb{N} \to [0,1]$ is negligible if $\forall$ constant $c \in \mathbb{N}$ $\exists$ constant $n_c \in \mathbb{N}$

s.t. $\forall n \geq n_c \quad \mu(n) \leq \frac{1}{n^c}$

Constructing CPA secure encryption scheme :

<u>Idea</u>: Use a random pad $K \leftarrow \{0,1\}^n$ to produce many random looking pads $K_1, \ldots, K_t$ and use these to pad $m_1, \ldots, m_t$.

<u>Namely</u>: use a short random string to generate many random (looking) strings.

Pseudorandom Function

A pseudo random function F satisfies $\forall n \in \mathbb{N}$ $\forall t = poly(n)$

$\forall$ distinct $x_1, \ldots, x_t \in \{0,1\}$

$$\left(F(K, x_1), \ldots, F(K, x_t)\right) \cong \left(U_1, \ldots, U_t\right)$$

for $K \leftarrow \{0,1\}^n$.

simplified version!

In the typical def. each x_i can be adversar[ially] and adaptively chosen, seeing $F(K, x_1), \ldots, F(K, x_{i-1})$

<u>Next class</u>: We will see a candidate construction of a PRF : AES.

In theory, we know how to construct a PRF from one-way functions (more

CPA encryption from PRF

$\mathrm{Gen}(1^n)$ output random $K \leftarrow \{0,1\}^n$

$\mathrm{Enc}(k, m)$: choose a random $r \leftarrow \{0,1\}^n$ and output $\left(r,\ \overbrace{f_k(r)}^{\text{pad}} \oplus m\right)$

$\mathrm{Dec}(k, \underset{\substack{\shortparallel \\ (r,y)}}{c})$: compute $\omega = F(k, r)$, output $\omega \oplus c$