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 in 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)

- Gen(L) outputs a secret key K (must be prob)
  - security parameter

- Enc takes as input a secret key K & a msg m ∈ M, and outputs a ciphertext C

- Dec takes as input a key K and a ciphertext C and outputs a msg m.

Correctness: ∀n ∈ N ∀m ∈ M

\[ Pr[ Dec(K, Enc(K,m)) = m ] = 1 \]
K = Gen(L)

One-Time Pad:

- Gen(L) outputs a random secret key K ∈ {0,1}^L
- Enc(K,m) = K ⊕ M (xor coordinate-wise)
- Dec(K,C) = K ⊕ C

msg space M = {0,1}^L

Example:

K = (011010)

m = (101001)

C = Enc(K,m) = (110111) 📰

Dec(K,C) = (101001) ✓

Correctness: ∀K ∈ {0,1}^L ∀m ∈ {0,1}^L

Dec(K, Enc(K,m)) = K ⊕ (K ⊕ m) = (K ⊕ m) ⊕ m = m

Security: Perfect security!
\[ \forall m \in \{0,1\}^n \quad \operatorname{Enc}(k, m) \in U_{10^{15}} \quad \Box \]

\[ \Rightarrow (m, \operatorname{Enc}(k, m)) \equiv (m, U_{10^{15}}) \]

Even if \( m \) is known, the ciphertext \( \operatorname{Enc}(k, m) \) 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: \( \operatorname{Enc}(k, m) \) does not hide \[ \| m \| \]

```
length of m.
```

This is somewhat inherent.

Namely:

\[ (m_1, m_2, \operatorname{Enc}(k, m_1), \operatorname{Enc}(k, m_2)) \neq (m_1, m_2, U_1, U_2) \]

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

\(--\) Goal: Many-time secure encryption scheme. \( \forall k \in \mathcal{K} \forall m \in \mathcal{M} \)

\[ (m_1, m_2, \operatorname{Enc}(k, m_1), \ldots, \operatorname{Enc}(k, m_t)) \equiv (m_1, m_2, U_1, \ldots, U_t) \]

Impossible!

1. Intuitively, \( \operatorname{Enc}(k, m_1), \ldots, \operatorname{Enc}(k, m_t) \) give too much information about \( k \) (unless \( \mathcal{K} \) grows with \( t \)).

2. Also, impossible if \( \operatorname{Enc} \) is a deterministic function since then

\[ (m, m, \operatorname{Enc}(k, m), \operatorname{Enc}(k, m)) \neq (m, m, U_1, U_2) \]

since \( \operatorname{Enc}(k, m) = \operatorname{Enc}(k, m) \).

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)

**Definition:** An encryption scheme $(\text{Gen}, \text{Enc}, \text{Dec})$ is CPA secure if
\[
\forall n \in \mathbb{N} \quad \forall t = \text{poly}(n) \quad \forall m_1, \ldots, m_t \in \mathcal{M}
\]
\[
\left( \text{Enc}(K, m_1), \ldots, \text{Enc}(K, m_t) \right) \equiv \left( \text{Enc}(K, \mathcal{U}_1), \ldots, \text{Enc}(K, \mathcal{U}_t) \right)
\]
where $K \overset{\$}{\leftarrow} \text{Gen}(\lambda^n)$, $\mathcal{U}_1, \ldots, \mathcal{U}_t \in \mathcal{M}$, computationally indistinguishable.

**Simplified Version!**

In the typical definition each $m_i$ can be adaptively and adversarially chosen after seeing $\text{Enc}(K, m_1), \ldots, \text{Enc}(K, m_t)$.

**Definition:** Two distribution ensembles $\left\{ A_{\cdot} \right\}_{a \in \mathcal{A}}$ and $\left\{ B_{\cdot} \right\}_{b \in \mathcal{B}}$ are

computationally indistinguishable if $\forall \text{PPT} \text{ distinguisher } \mathcal{D}$ there exists a negligible function $\mu: \mathbb{N} \rightarrow [0, 1]$
\[
\forall n \in \mathbb{N}
\]
\[
\left| \Pr_{x \leftarrow A_{\cdot}} [\mathcal{D}(x) = 1] - \Pr_{y \leftarrow B_{\cdot}} [\mathcal{D}(y) = 1] \right| \leq \mu(n)
\]

**Definition:** A function $\mu: \mathbb{N} \rightarrow [0, 1]$ is negligible if $\forall$ constant $c \in \mathbb{N}$ there exists a constant $n_c \in \mathbb{N}$
\[
\forall n > n_c \quad \mu(n) \leq \frac{1}{n^c}
\]

**Constructing CPA secure encryption scheme:**

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

**Namely:** Use a short random string to generate many random (looking) strings.

**Pseudorandom Function**

A pseudorandom function $F$ satisfies $\forall n \in \mathbb{N} \quad \forall t = \text{poly}(n)$
\[
\forall \text{distinct } x_1, \ldots, x_t \in \{0, 1\}^n
\]
\[
\left( F(K, x_1), \ldots, F(K, x_t) \right) \equiv \left( U_1, \ldots, U_t \right)
\]
for $K \overset{\$}{\leftarrow} \{0, 1\}^n$.

**Simplified Version!**

In the typical def, each $x_i$ can be adversarial and adaptively chosen, seeing $F(K, x_1), \ldots, F(K, x_t)$.

**Next class:** We will see a candidate construction of a PRF: AES.

In theory, we know how to construct a PRF from one-way functions (non-CPA encryption from PRF)
Gen($\mathbb{F}$): output random $k \in \mathbb{F}^{n \times n}$

Enc($k$, $m$): Choose a random $r \in \mathbb{F}^{n \times n}$ and output $(r, \mathbf{f}_k(r) \oplus M)$

Dec($k$, $c$): compute $\mathbf{w} = \mathbf{F}(k, r^T)$, output $\mathbf{w} \odot c$

$(r, y)$