The One-Time Pad (Vernam’s Cipher)

- In 1917, Vernam patented a cipher now called the one-time pad that obtains perfect secrecy.
- There was no proof of this fact at the time.
- 25 years later, Shannon introduced the notion of perfect secrecy and demonstrated that the one-time pad achieves this level of security.

The One-Time Pad Scheme

1. Fix an integer $\ell > 0$. Then the message space M, key space K, and ciphertext space C are all equal to $\{0,1\}^{\ell}$.
2. The key-generation algorithm Gen works by choosing a string from $K = \{0,1\}^{\ell}$ according to the uniform distribution.
3. Encryption Enc works as follows: given a key $k \in \{0,1\}^{\ell}$, and a message $m \in \{0,1\}^{\ell}$,output $c \coloneqq k \oplus m$.
4. Decryption Dec works as follows: given a key $k \in \{0,1\}^{\ell}$, and a ciphertext $c \in \{0,1\}^{\ell}$, output $m \coloneqq k \oplus c$.

Security of OTP

Theorem: The one-time pad encryption scheme is perfectly secure.

Proof

Proof: Fix some distribution over M and fix an arbitrary $m \in M$ and $c \in C$. For one-time pad:

$$\Pr[C = c | M = m] = \Pr[M \oplus K = c \mid M = m]$$

$$= \Pr[m \oplus K = c] = \Pr[K = m \oplus c] = \frac{1}{2^{\ell}}$$

Since this holds for all distributions and all m, we have that for every probability distribution over M, every $m_0, m_1 \in M$ and every $c \in C$

$$\Pr[C = c \mid M = m_0] = \frac{1}{2^{\ell}} = \Pr[C = c | M = m_1]$$

Drawbacks of OTP

- Key length is the same as the message length.
  - For every bit communicated over a public channel, a bit must be shared privately.
  - We will see this is not just a problem with the OTP scheme, but an inherent problem in perfectly secret encryption schemes.
- Key can only be used once.
  - You will see in the homework that this is also an inherent problem.

Some Examples

- Is the following scheme perfectly secret?
- Message space $\boldsymbol{M} = \{0,1, \ldots, n-1\}$. Key space $\boldsymbol{K} = \{0,1, \ldots, n-1\}$.
- $\mathrm{Gen}()$ chooses a key k at random from $\boldsymbol{K}$.
- $\mathrm{Enc}_k(m)$ returns $m + k$.
- $Dec_k(c)$ returns $c - k$.

Some Examples

- Is the following scheme perfectly secret?
- Message space $\boldsymbol{M} = \{0,1,\ldots,n-1\}$. Key space $\boldsymbol{K} = \{0,1,\ldots,n-1\}$.
- $\mathrm{Gen}()$ chooses a key k at random from $\boldsymbol{K}$.
- $\mathrm{Enc}_k(m)$ returns $m + k \, mod \, n$.
- $Dec_k(c)$ returns $c - k \, mod \, n$.

Limitations of Perfect Secrecy

Theorem: Let (Gen, Enc, Dec) be a perfectly-secret encryption scheme over a message space $\boldsymbol{M}$, and let $\boldsymbol{K}$ be the key space as determined by Gen. Then $|\boldsymbol{K}| \geq |\boldsymbol{M}|$.

Proof

Proof (by contradiction): We show that if $|\boldsymbol{K}| < |\boldsymbol{M}|$ then the scheme cannot be perfectly secret.

- Assume $|\boldsymbol{K}| < |\boldsymbol{M}|$. Consider the uniform distribution over $\boldsymbol{M}$ and let $c \in \boldsymbol{C}$.
- Let $\boldsymbol{M}(c)$ be the set of all possible messages which are possible decryptions of c.

$$\boldsymbol{M}(c) \coloneqq \{\widehat{\widehat{m}} \mid \widehat{m} = Dec_k(c) for\ some\ \hat{k} \in \boldsymbol{K}\}$$

Proof

$$\boldsymbol{M}(c) \coloneqq \{\, \widehat{m} \mid \widehat{m} = Dec_k(c) for\ some\ \hat{k} \in \boldsymbol{K}\}$$

- $|\boldsymbol{M}(c)| \leq |\boldsymbol{K}|$. Why?
- Since we assumed $|\boldsymbol{K}| < |\boldsymbol{M}|$, this means that there is some $m' \in \boldsymbol{M}$ such that $m' \notin \boldsymbol{M}(c)$.
- But then

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

And so the scheme is not perfectly secret.

Shannon's Theorem

Let (Gen, Enc, Dec) be an encryption scheme with message space $\boldsymbol{M}$, for which $|\boldsymbol{M}| = |\boldsymbol{K}| = |\boldsymbol{C}|$. The scheme is perfectly secret if and only if:

1. Every key $k \in \boldsymbol{K}$ is chosen with equal probability $1/|\boldsymbol{K}|$ by algorithm Gen.
2. For every $m \in \boldsymbol{M}$ and every $c \in \boldsymbol{C}$, there exists a unique key $k \in \boldsymbol{K}$ such that $Enc_k(m)$ outputs c.

**Theorem only applies when $|\boldsymbol{M}| = |\boldsymbol{K}| = |\boldsymbol{C}|$.

Some Examples

- Is the following scheme perfectly secret?
- Message space $\boldsymbol{M} = \{0,1,\ldots,n-1\}$. Key space $\boldsymbol{K} = \{0,1,\ldots,n-1\}$.
- $\mathrm{Gen}()$ chooses a key k at random from $\boldsymbol{K}$.
- $\mathrm{Enc}_k(m)$ returns $m + k$.
- $Dec_k(c)$ returns $c - k$.

Some Examples

- Is the following scheme perfectly secret?
- Message space $\boldsymbol{M} = \{0,1,\ldots,n-1\}$. Key space $\boldsymbol{K} = \{0,1,\ldots,n-1\}$.
- $\mathrm{Gen}()$ chooses a key k at random from $\boldsymbol{K}$.
- $\mathrm{Enc}_k(m)$ returns $m + k \; mod \; n$.
- $Dec_k(c)$ returns $c - k \; mod \; n$.

The Computational Approach to Security

"An encryption scheme is secure if no adversary learns meaningful information about the plaintext after seeing the ciphertext"

How do you formalize learns meaningful information?

The Computational Approach to Security

- Meaningful Information about plaintext m:
  - $f(m)$ for an efficiently computable function f
- Learn Meaningful Information from the ciphertext:
  - An efficient algorithm that can output $f(m)$ after seeing c but could not output $f(m)$ before seeing c.
- Learn Meaningful Information:
  - The change in probability that an efficient algorithm can output $f(m)$ after seeing c and can output $f(m)$ before seeing c is significant.

Note:

- The intuitive definition from the previous slide is known as “semantic security.”
- We will first see a different, simpler definition known as indistinguishability.
- Later we will see that the two definitions are provably equivalent.

The Computational Approach

Two main relaxations:

1. Security is only guaranteed against efficient adversaries that run for some feasible amount of time.
2. Adversaries can potentially succeed with some very small probability.

Security Parameter

- Integer valued security parameter denoted by n that parameterizes both the cryptographic schemes as well as all involved parties.
- When honest parties initialize a scheme, they choose some value n for the security parameter.
- Can think of security parameter as corresponding to the length of the key.
- Security parameter is assumed to be known to any adversary attacking the scheme.
- View run time of the adversary and its success probability as functions of the security parameter.

Polynomial Time

- Efficient adversaries = Polynomial time adversaries
  – There is some polynomial p such that the adversary runs for time at most $p(n)$ when the security parameter is n.
  – Honest parties also run in polynomial time.
  – The adversary may be much more powerful than the honest parties.

Negligible

- Small probability of success = negligible probability
  - A function f is negligible if for every polynomial p and all sufficiently large values of n it holds that $f(n) < \frac{1}{p(n)}$.
  - Intuition, $f(n) < n^{-c}$ for every constant c, as n goes to infinity.

Negligible