# ☙ CS6222 Cryptography ❧

Topic: Introduction | Date: Aug 29, 2024
Lecturer: Wei-Kai Lin (TA: Arup Sarker) | Scriber: Dhriti Gampa

---

## Private Key Encryption Scheme

An encryption scheme for any message, **m**, in the message space **M**, if the syntax of functions, correctness, and privacy of private key hold, is **(Gen, Enc, Dec)**.

$Gen() \to k$: A randomized generate function that outputs a key, **k**, in the keyspace, **K**,.

$Enc_k(m) \to ct$: An encryption function that utilizes **k** to return the ciphertext, **ct**, for the message **m** in the message space.

$Dec_k(ct) \to m$: A decryption function that utilizes **k** to return the deciphered text **m** for the ciphertext, **ct**.

**Correctness:** $\forall\, m \in M$, $Pr[Dec_k(Enc_k(m)) = m] = 1$, where $k \leftarrow Gen()$.

Private Key encryption states that you will always (with probability 1) recover the original message m if you use the same key to encrypt and decrypt the message.

## Shannon Secrecy Encryption Scheme

A private key encryption scheme **(M, K, Gen, Enc, Dec)** is a Shannon secret if

$\forall$ distributions $d \in D$, $\forall$ decrypted messages $m' \in M$, and $\forall$ ciphertexts $c \in C$

$Pr[m = m' \mid Enc_k(m) = c] = Pr[m = m']$ where $Gen()$ returns a key $k \in K$ and the distribution $d$ returns $m$

The cipher text doesn't carry more information than the original message. Therefore the cipher text has the same distribution as the original message.

**Perfect Secrecy Encryption Scheme** A private key encryption scheme **(M, K, Gen, Enc, Dec)** is perfectly secret if any two messages $m_0$ and $m_1 \in$ message space $M$ result in the same ciphertext $c$ distribution.

$\forall\, m_0, m_1 \in M$ and $\forall\, c \in C$

$Pr[Enc_k(m_0) = c] = Pr[Enc_k(m_1) = c]$

$Gen()$ returns a key $k$, and the key $k$ used for the encryption of both messages does not need to be the same

*Shannon Secrecy and Perfect Secrecy have a bi-directional relationship. Therefore Shannon Secrecy implies Perfect Secrecy and vice versa.

**One-Time Pad Encryption Scheme** A perfectly secure private-key encryption scheme **(M, K, Gen, Enc, Dec)**

$M = \{0, 1\}^n$

$K = \{0,1\}^n$

$Gen() : \{0,1\}^n \leftarrow k_1k_2...k_n = k$

$Enc_k(m_1m_2...m_n)$, $c_1c_2...c_n$ where $c_i = m_i \oplus k_i$

$Dec_k(c_1c_2...c_n)$, $m_1m_2...m_n$ where $m_i = c_i \oplus k_i$

One time pad is not crackable even with infinite computational power $\leftarrow$ it is a perfect encryption scheme. Therefore, $|K| \geq |M|$ and it is optimal in key length. The $Pr[Enc_k(m_2) = c] = 0$.

However if $|K| < |M|$ there is a loss of security because the $Pr[Enc_k(m_2) = c] > 0$ and perfect secrecy is violated. Also, security is reduced if the same key is used for multiple messages

**Efficient Computation** Extend a short truly random string to a long "random=looking" string. It is efficient because it runs in polynomial time. Randomized Algorithm (PPT): A randomized (or probabilistic) algorithm A, or a probabilistic polynomial-time Turing Machine (PPT), is a deterministic algorithm with an additional random tape, where each bit on this tape is chosen uniformly and independently. The algorithm's computation is denoted as $y \leftarrow A(c;r)$, where r represents the random bits. The algorithm runs in time T(n), so the running time is less than or equal to T for all inputs (x,r).