Block Ciphers

Brandon Thompson

September 30, 2019

1 Stream Cipher vs Block Cipher

Stream ciphers encrypt one bit or byte at a time using the XOR operation. Block cipher is one where a section of plain text is treated as whole and used to produce a cipher text of equal length (typically 64 or 128 bits).

Feistal Cipher: we can approximate the ideal block cipher by utilizing the concept of a product cipher, the execution of two or more simple ciphers in sequence such that the final result is is cryptographically stronger than any of the component ciphers. Goal is to develop a block cipher with a key length of k bits and block length of n bits, allowing a total of 2^k possible transformations.

- **Substitution:** Each plain text element or group of elements is uniquely replaced by a corresponding cipher text element or group of elements.
- **Permutation:** A sequence of plain text elements is replaced by a permutation of that sequence. No elements are added, deleted, or replaced, in the sequence, rather the order of the elements is changed.

Claude Shannon introduced the terms *diffusion* and *confusion* to capture the two basic building blocks for cryptography systems, to hinder statistical analysis of the cipher text. In **diffusion** the statistical structure of the plain text is dissipated into long-range statistics of the cipher text. This is achieved by having each plain text digit effect the value of many cipher text digits. An example is to encrypt a message $M = m_1, m_2, m_3, \ldots$ of characters with an averaging operation:

$$y_n = \left(\sum_{i=1}^{k} m_{n+i} \right) \mod 26$$

adding k successive letters to get a cipher text letter y_n. Letter frequencies in the cipher text will be more nearly equal than in the plain text. In a binary block cipher, diffusion can be achieved by repeatedly performing some permutation on the data followed by applying a function to that permutation.

Every block cipher involves a transformation of a block of plain text into a block of cipher text given functions $f_1, \ldots, f_d : \{0,1\}^n \to \{0,1\}^n$
goal: build inevitable function $F : \{0,1\}^{2n} \to \{0,1\}^{2n}$
decryption

2 The Function

- Pseudo Random Function **(PRF)** defined over (K,X,Y):

$$\text{F: K} \times \text{X} \rightarrow \text{Y}$$

Such that exists "efficient" algorithm to evaluate F(k,x)

- Pseudo Random Permutation **(PRP)** defined over (K,X):

$$\text{E: K} \times \text{X} \rightarrow \text{Y}$$

Such that:

1. Exists "efficient" <u>deterministic</u> algorithm to evaluate E(k,x)
2. The Function E(k, $\cdot$) is one-to-one
3. Exists "efficient" inversion algorithm D(k,y)

Examples:

- 3DES: K $\times$ X $\rightarrow$ X where X = $\{0,1\}^{64}$, K = $\{0,1\}^{168}$
- AES: K $\times$ X $\rightarrow$ X where K = X = $\{0,1\}^{128}$

Functionally, any PRP is also a PRF, A PRP is a PRF where X=Y and is efficiently invertible.

3 DES: The Data Encryption Standard

Given plain text block of n-bits and key of k-bits, block ciphers encrypt/decrypt based on these block and key sizes.

1. 3DES: $n = 64$ bits, $k = 168$ bits
2. AES: $n = 128$ bits, $k = 128, 192, 256$ bits

Key is expanded to $k = k_1, \ldots, k_n$ message is encrypted using round functions $R(k, m)$, 3DES uses 48 iterations, AES-128 uses 10 iterations.

3.1 Feistel Network

Given Functions $f_1, \ldots, f_d : \{0,1\}^n$
Goal: build invertible function $F : \{0,1\}^{2n}$

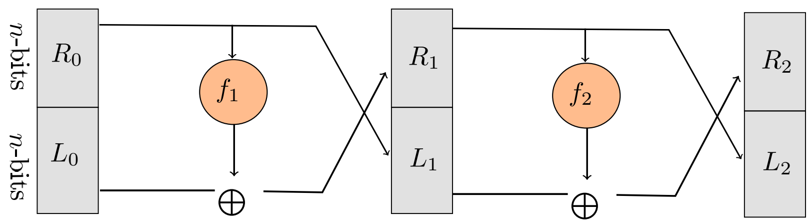

Figure 1: Feistal Network

$$\text{Symbol Representation: } \begin{cases} R_i = f_i\left(R_{i-1}\right) \bigoplus L_{i-1} \\ L_i = R_{i-1} \end{cases}$$

For all $f_1, \ldots, f_d : \{0,1\}^n \rightarrow \{0,1\}^n$ Feistal network F: $\{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$ is invertible. Apply $f_1, \ldots, f_d$ in reverse order to decrypt. This process is used on many block ciphers, but not AES.

Key is 56-bits, key expansion generates 16 keys of 48-bits. Uses S-box lookup table $S_i := \{0,1\}^6 \rightarrow \{0,1\}^4$ If there is a linear relationship DES can be represented by matrix vector product.
Thus the entire DES cipher would be linear.

4 Attack on Block Cipher

4.1 Exhaustive Search

Goal: Given a few input pairs $(m_i, c_i = E(k_i, m_i))$ $i = 1, \ldots, 3$ find key k.

Lemma 4.1 *Suppose DES is an* ***ideal cipher***
$\left(2^{56}\right.$ random invertible functions$\left.\right)$
Then $\forall\ m, c$ there is at most ***one*** *key s.t. $c = DES\left(k, m\right)$*

$$Pr\left[\exists\ k' \neq K, c = DES\left(k, m\right) = DES\left(k', m\right)\right] \leq \sum_{k' \in \{0,1\}^{56}} Pr\left[DES\left(k, m\right) = \right.$$

$$\mathrm{DES}(k', m) \leq 2^{56} \cdot \tfrac{1}{2^{64}} = \tfrac{1}{2^{8}}$$

For two DES pairs

$$\left(m_1, c_1 = DES\left(k, m_1\right)\right), \left(m_2, c_2 = DES\left(k, m_2\right)\right)$$

Unicity probability $\approx 1 - \frac{1}{2^{71}}$
For AES-128: given two input/output pairs, unicity probability $\approx 1 - \frac{1}{2^{128}}$
Two input/output pairs are enough for exhaustive key search.

4.2 DES Challenge

msg = "The unknown messages is: XXXX ..."
CT $= c_1, c_2, c_3, c_4$
Goal: find $k \in \{0,1\}^{56}$ s.t. $DES\left(k_i, m_i\right) = c_i$ for $i = 1, 2, 3$

56-bit ciphers should no be used, 128-bit key = 2^{72} days

5 Strengthen DES against exhaustive search

5.1 Method 1: Triple DES

- Let $E : K \times M \to M$ be a block cipher
- define 3E: $K^3 \times M \to M$ as

$$3E\left(\left(k_1, k_2, k_3\right), m\right) = E\left(k_1, D\left(k_2, E\left(k_3, m\right)\right)\right)$$

$$\mathrm{k}_1 = k_2 = k_3 \implies \text{single DES}$$

For 3DES: Key size $= 3 \times 56 = 168$ bits. $3\times$ slower than DES. Simple attack in time $= 2^{118}$

5.2 Why not double DES?

5.2.1 Meet in the middle attack

- Define $2E\left(\left(k_1, k_2\right), m\right) = E\left(k_1, E\left(k_2, m\right)\right)$
- find $\left(k_1, k_2\right)$ s.t. $E\left(k_1, E\left(k_2, m\right)\right) = C$

- equivocally: $E(k_2, m) = D(k_1, c)$

  two tables of keys and cipher texts can be combined to find both keys.

  space $\approx 2^{56}$

Attack:

1. Build table of $k_0, \ldots, k_n$ and $E(k_0, M), \ldots, E(k_n, M)$
2. $\forall\ k \in \{0,1\}^{56}$: test if $D(k, C)$ is in 2nd column.
   if so then $E(k_i, M) = D(k, C) \implies (k_i, k) = (k_2, k_1)$

time $= 2^{56} \log\left(2^{56}\right) + 2^{56} \log\left(2^{56}\right) < 2^{63} \ll 2^{112}$

Same attack on 3DES: time $= 2^{118}$, space $= 2^{56}$.

5.3 Method 2: DESX