C R Y P T O G R A P H Y

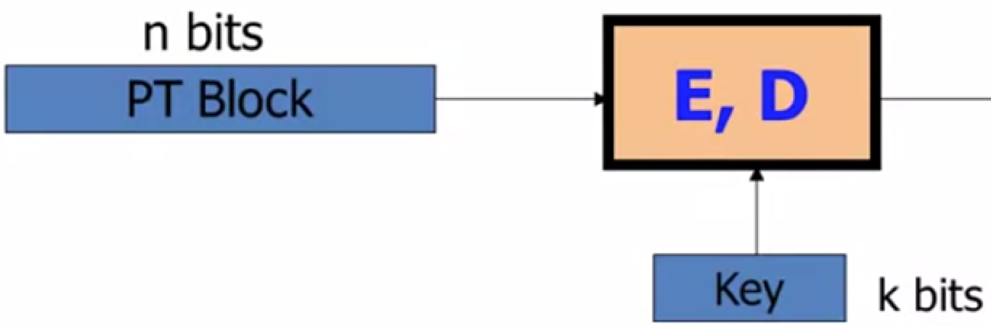
WEEK 2
BLOCK CIPHERS

/01

What are Block Ciphers?

Now that we understand stream ciphers, we're gonna move on and talk about a more powerful primitive called a block cipher.

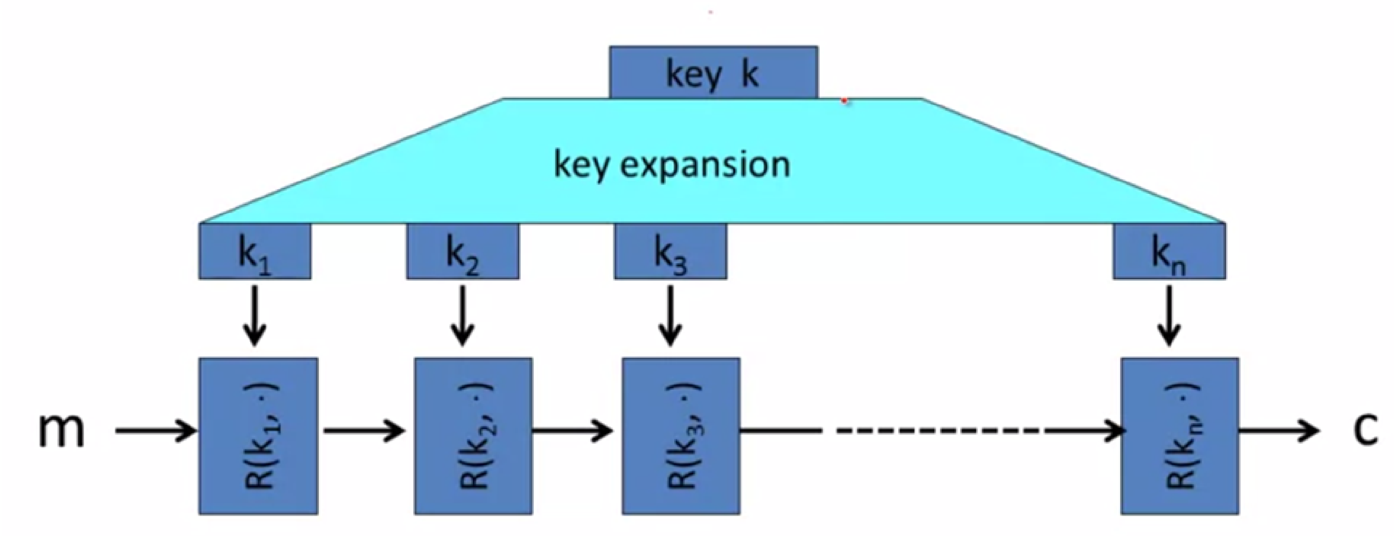
Crypto work horse



典型块加密:

- 3DES　　n = 64　bits，k = 168 bits
- AES　　n = 128 bits，k = 128, 192, 256 bits

Block Ciphers Built by Iteration



将k扩展为一系列密钥k1 ~ kn，统称轮密钥

R(k, m) 为轮函数。使用轮函数对m迭代加密

3DES(n = 48)　　　　AES-128(n = 10)

Performance

AMD Opteron, 2.2 GHz (Linux)

	Cipher	Block/key size	Speed (MB/sec)
stream	RC4		126
	Salsa20/12		643
	Sosemanuk		727
block	3DES	64/168	13
	AES	128/128	109

Abstractly: PRPs and PRFs

- PRF：伪随机函数(Pseudo Random Function)
- PRF 定义在（K：密钥空间，X：输入空间，Y：输出空间
- $F: K \times X \to Y$
- 存在高效率的算法实现F(k, x)

- PRP：伪随机置换(Pseudo Random Permutation)
- PRP 定义在（K：密钥空间，X：集合）
- E：$K \times X \to X$
- 存在高效率确定性算法实现E(k, x)
- E(l, ·) 是一一对应的
- 存在高效率逆向算法D(k, y)

Running Examples

AES: $K \times X \to X$ 其中 $K=X=\{0,1\}^{128}$

3DES: $K \times X \to X$ 其中 $X = \{0,1\}^{64}$, $K=\{0,1\}^{168}$

PRP 是一种 PRF 的特殊情况

Secure PRFs

Let F: $K \times X \to Y$ be a PRF

$$\begin{cases} \text{Funs[X,Y]: the set of \underline{all} functions from X to Y} \\ S_F = \{\ F(k,\cdot)\ \text{ s.t. } \ k \in K\ \} \subseteq \text{Funs[X,Y]} \end{cases}$$

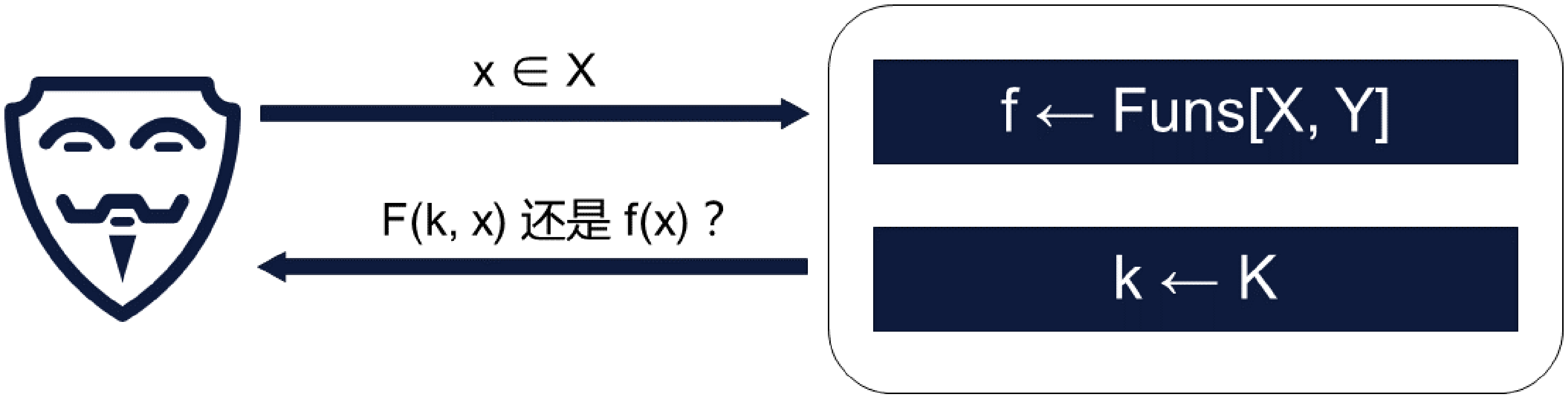
Funs[X, Y] 大小：　|X|^|Y|

SF大小：　|K|

PRF 从 Funs[X, Y]中筛选出了一个很小的，由K指定的F函数集合

Secure PRFs

一个安全的PRF满足：
无法区别 Funs[X, Y]中的一个随机函数 与 SF中的一个随机函数



PRF to PRG

Let $F: K \times \{0,1\}^n \to \{0,1\}^n$ be a secure PRF.

Then the following $G: K \to \{0,1\}^{nt}$ is a secure PRG:

$$G(k) = F(k,0) \parallel F(k,1) \parallel \cdots \parallel F(k,t)$$

Key property: parallelizable

Security from PRF property: $F(k, \cdot)$ indist. from random function $f(\cdot)$

/02

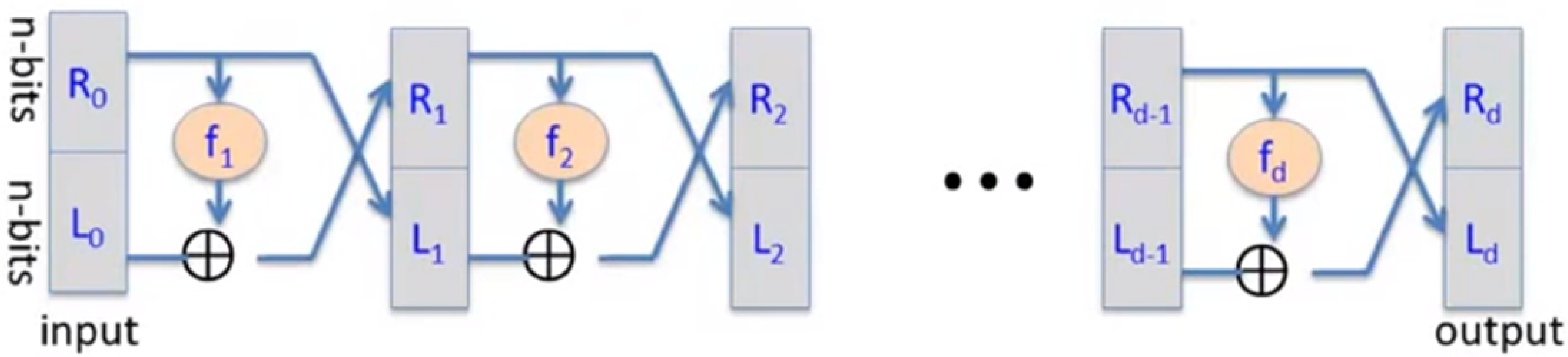
The data encryption standard (DES)

So now that we understand what block ciphers are, let's look at a classic example called the Data Encryption Standard.

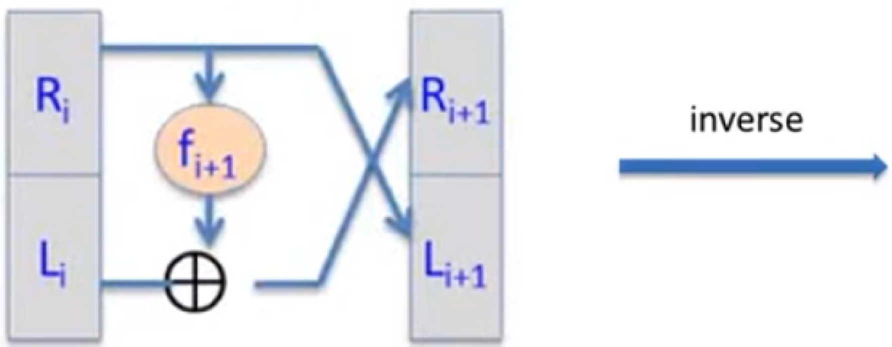
Feistel Network – Core Idea of DES

给出函数 $f_1, \ldots, f_d: \{0, 1\}^n \rightarrow \{0, 1\}^n$

目标：建立可逆函数 F：$\{0, 1\}^{2n} \rightarrow \{0, 1\}^{2n}$



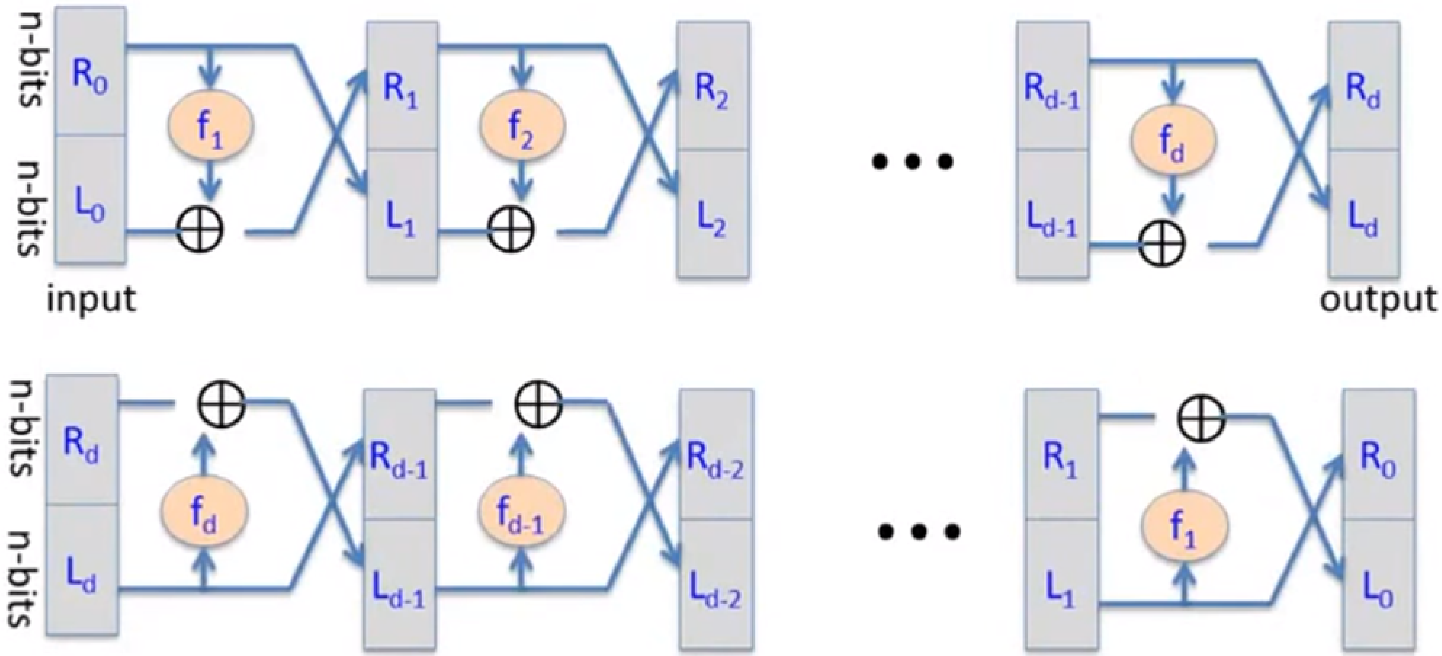
Feistel Network – Construct Inverse



$$R_i = L_{i+1}$$

$$L_i = f_{i+1}(L_{i+1}) \oplus R_{i+1}$$

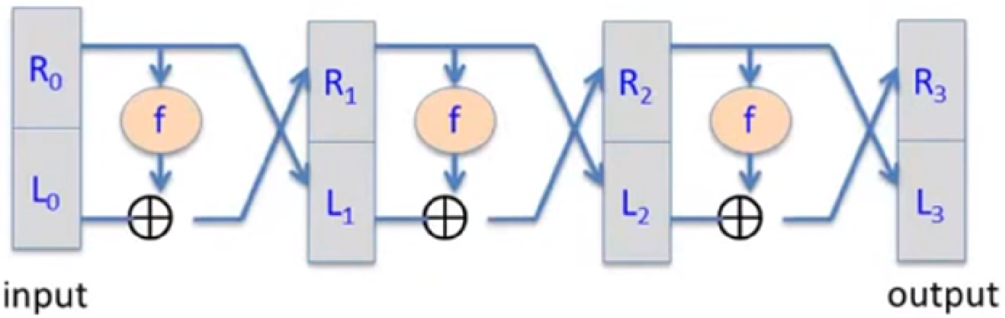
Feistel Network – Decryption Circuit



Security of Feistel Network

f: $K \times \{0,1\}^n \longrightarrow \{0,1\}^n$ a secure PRF

$\Rightarrow$ 3-round Feistel F: $K^3 \times \{0,1\}^{2n} \longrightarrow \{0,1\}^{2n}$ a secure PRP

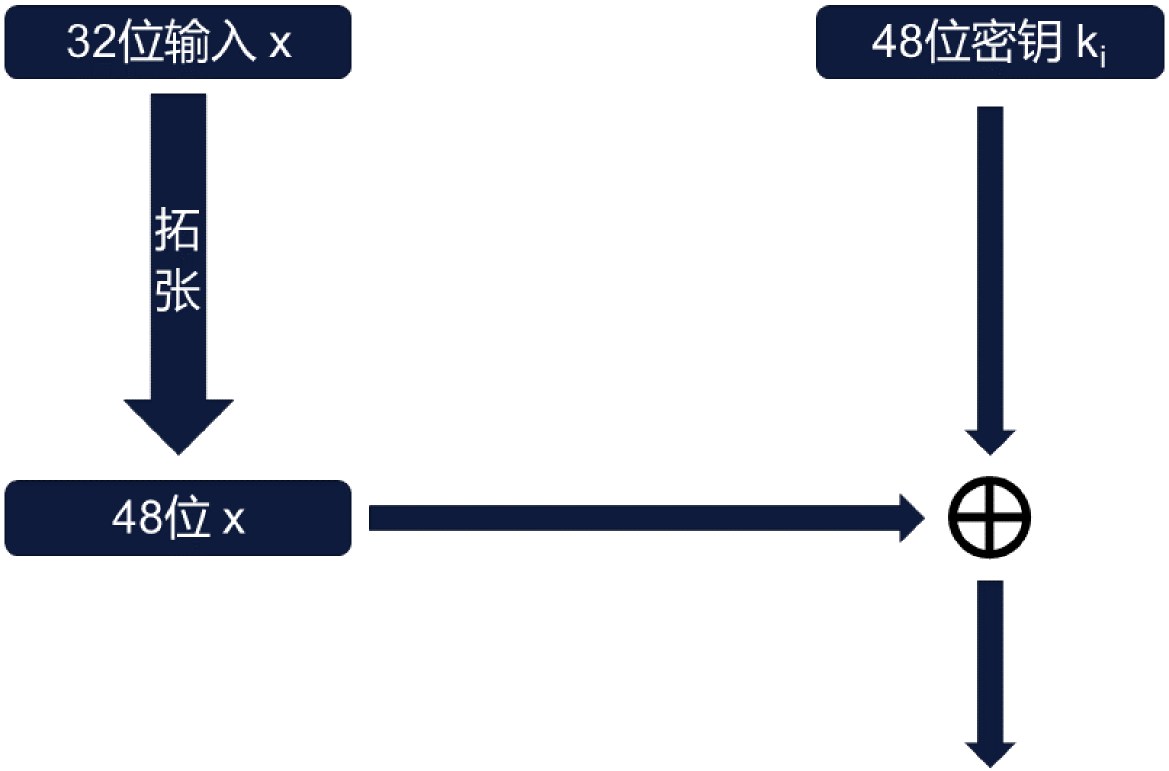


三次运算使用了三个独立密钥

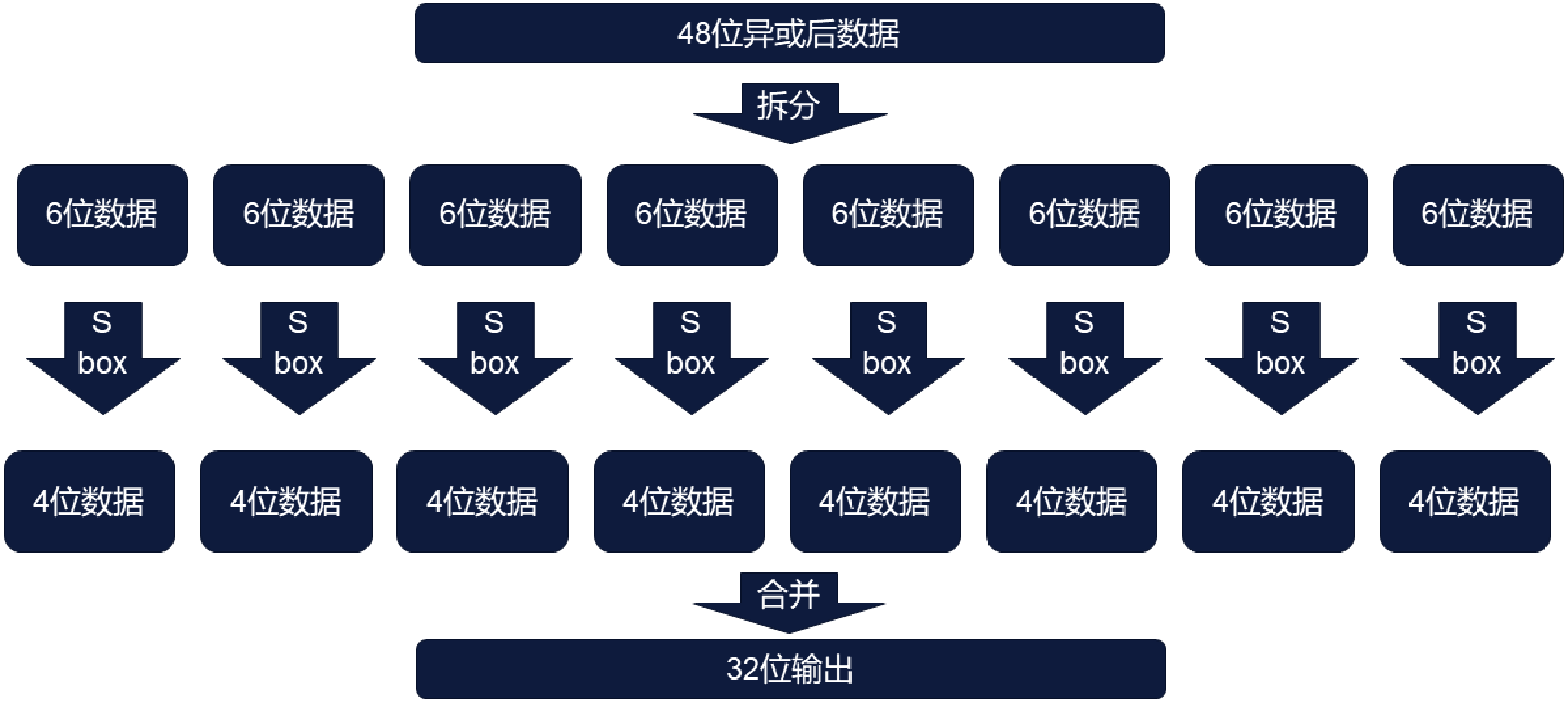
DES

- DES 进行了16轮 Feistel 网络
- 16轮函数 $f_i(x)$ 为一个 $F(k_i, X)$ 用16个独立密钥推导出
- 16个48位密钥由一个56位DES密钥扩张而成
- 颠倒16个密钥的使用顺序即可解密

Function F



Function F


48位异或后数据
拆分
6位数据 6位数据 6位数据 6位数据 6位数据 6位数据 6位数据 6位数据
S box S box S box S box S box S box S box S box
4位数据 4位数据 4位数据 4位数据 4位数据 4位数据 4位数据 4位数据
合并
32位输出

S-box

- 函数：$\{0,1\}^6 \rightarrow \{0,1\}^4$，由查找表实现

S_5		Middle 4 bits of input															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Outer bits	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

$$011011 \rightarrow 1001$$

Example: Bad S-box Choice

Suppose:

$$S_i(x_1, x_2, \ldots, x_6) = (\ x_2 \oplus x_3,\ \ x_1 \oplus x_4 \oplus x_5,\ \ x_1 \oplus x_6,\ \ x_2 \oplus x_3 \oplus x_6\ )$$

or written equivalently: $S_i(\mathbf{x}) = A_i \cdot \mathbf{x}$ (mod 2)

$$\begin{bmatrix} 0&1&1&0&0&0 \\ 1&0&0&1&1&0 \\ 1&0&0&0&0&1 \\ 0&1&1&0&0&1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} x_2 \oplus x_3 \\ x_1 \oplus x_4 \oplus x_5 \\ x_1 \oplus x_6 \\ x_2 \oplus x_3 \oplus x_6 \end{bmatrix}$$

We say that S_i is a linear function.

Example: Bad S-box Choice

Then entire DES cipher would be linear: ∃fixed binary matrix B s.t.

$$\mathrm{DES}(k,m) = \begin{array}{c c} & 832 \\ 64 & \boxed{B} \end{array} \cdot \begin{pmatrix} m \\ k_1 \\ k_2 \\ \vdots \\ k_{16} \end{pmatrix} = \begin{pmatrix} c \end{pmatrix} \pmod 2$$

But then: $\mathrm{DES}(k,m_1) \oplus \mathrm{DES}(k,m_2) \oplus \mathrm{DES}(k,m_3) = \mathrm{DES}(k, m_1 \oplus m_2 \oplus m_3)$

$$k = \begin{pmatrix} k_1 \\ \vdots \\ k_{16} \end{pmatrix}$$

$$B \begin{pmatrix} m_1 \\ k \end{pmatrix} \oplus B \begin{pmatrix} m_2 \\ k \end{pmatrix} \oplus B \begin{pmatrix} m_3 \\ k \end{pmatrix} = B \begin{pmatrix} m_1 \oplus m_2 \oplus m_3 \\ k \oplus k \oplus k \end{pmatrix}$$

Choosing the S-boxes and P-box

极少数时候表现为线性的 S-box 也会很容易被破解

S-box P-box 选择规则：

- 没有一组输入输出与线性函数相近
- S-box 是 4 到 1 映射，即有4个不同输入可产生1个相同输出


THANKS
1818039 陈文韬