现代密码学

Modern Cryptography

张方国

中山大学计算机学院

Office: Room 305, IM School Building

E-mail: isszhfg@mail.sysu.edu.cn

HomePage: https://cse.sysu.edu.cn/content/2460

第七讲 分组密码（一）

- 引言
- 代换置换网络
- 堆积引理
- 线性密码分析

引言

大部分分组密码都是乘积密码。乘积密码通常伴随着一系列置换与代换操作，常见的乘积密码是迭代密码。

我们通过一个密码体制的实例来说明迭代密码：这个密码明确定义了一个轮函数和一个密钥编排方案，一个明文的加密将经过Nr轮类似的过程。

设K是一个确定长度的随机二元密钥，用K来生成Nr个轮密钥（也叫子密钥）$K^1,\cdots,K^{Nr}$，轮密钥的列表$K^1,\cdots,K^{Nr}$就是密钥编排方案。密钥编排方案由K经一个固定的、公开的算法生成。

迭代密码

加密：轮函数g以轮密钥(K^r)和当前状态w^{r-1}作为它的两个输入。下一个状态定义为$w^r = g(w^{r-1}, K^r)$。初态w^0被定义为明文x，密文y定义为经过所有Nr轮后的状态。加密过程：

$$w^0 \leftarrow x$$

$$w^1 \leftarrow g(w^0, K^1)$$

$$w^2 \leftarrow g(w^1, K^2)$$

$$\vdots$$

$$w^{Nr-1} \leftarrow g(w^{Nr-2}, K^{Nr-1})$$

$$w^{Nr} \leftarrow g(w^{Nr-1}, K^{Nr})$$

$$y \leftarrow w^{Nr}$$

迭代密码

解密：为了能够解密，轮函数g在其第二个自变量固定的条件下必须是单射函数，这等价于存在函数g^{-1}，对所有的w和y，有$g^{-1}(g(w,y),y)=w$。解密过程：

$$w^{Nr} \leftarrow y$$

$$w^{Nr-1} \leftarrow g^{-1}(w^{Nr}, K^{Nr})$$

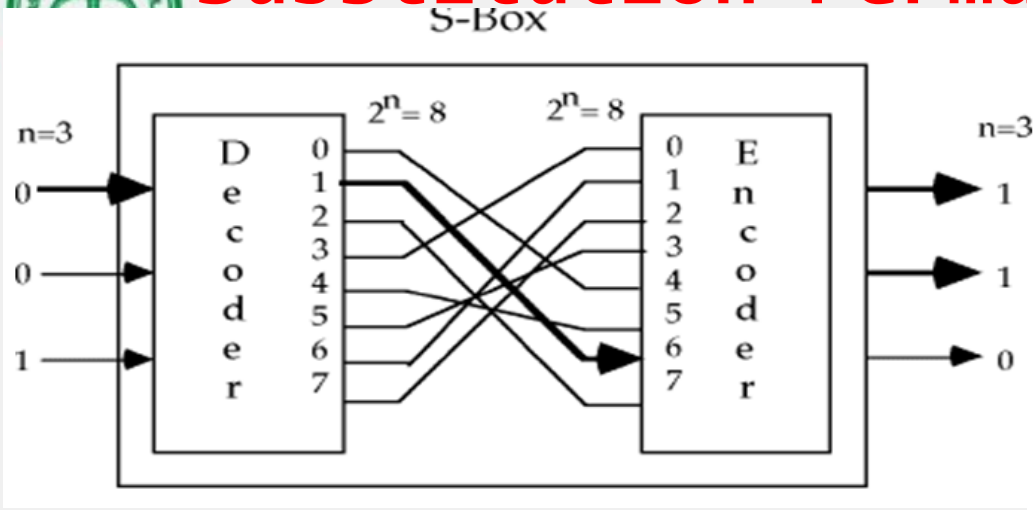
$$\vdots$$

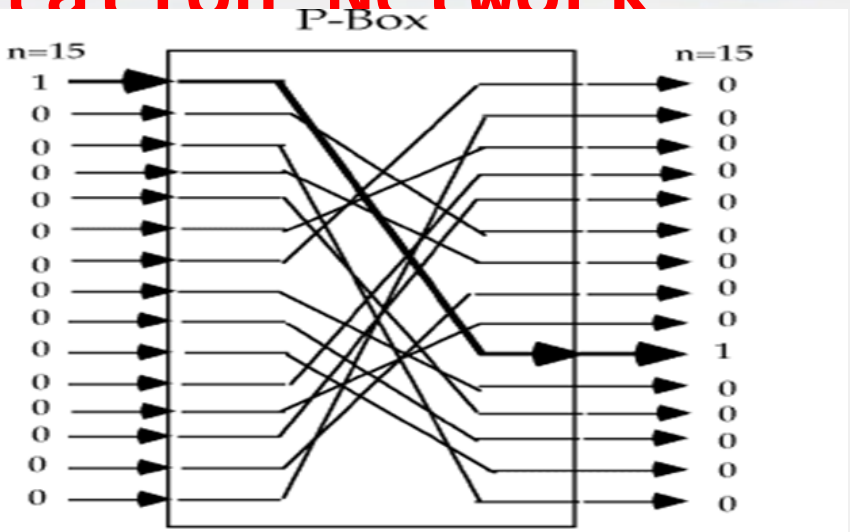
$$w^{1} \leftarrow g^{-1}(w^{2}, K^{2})$$

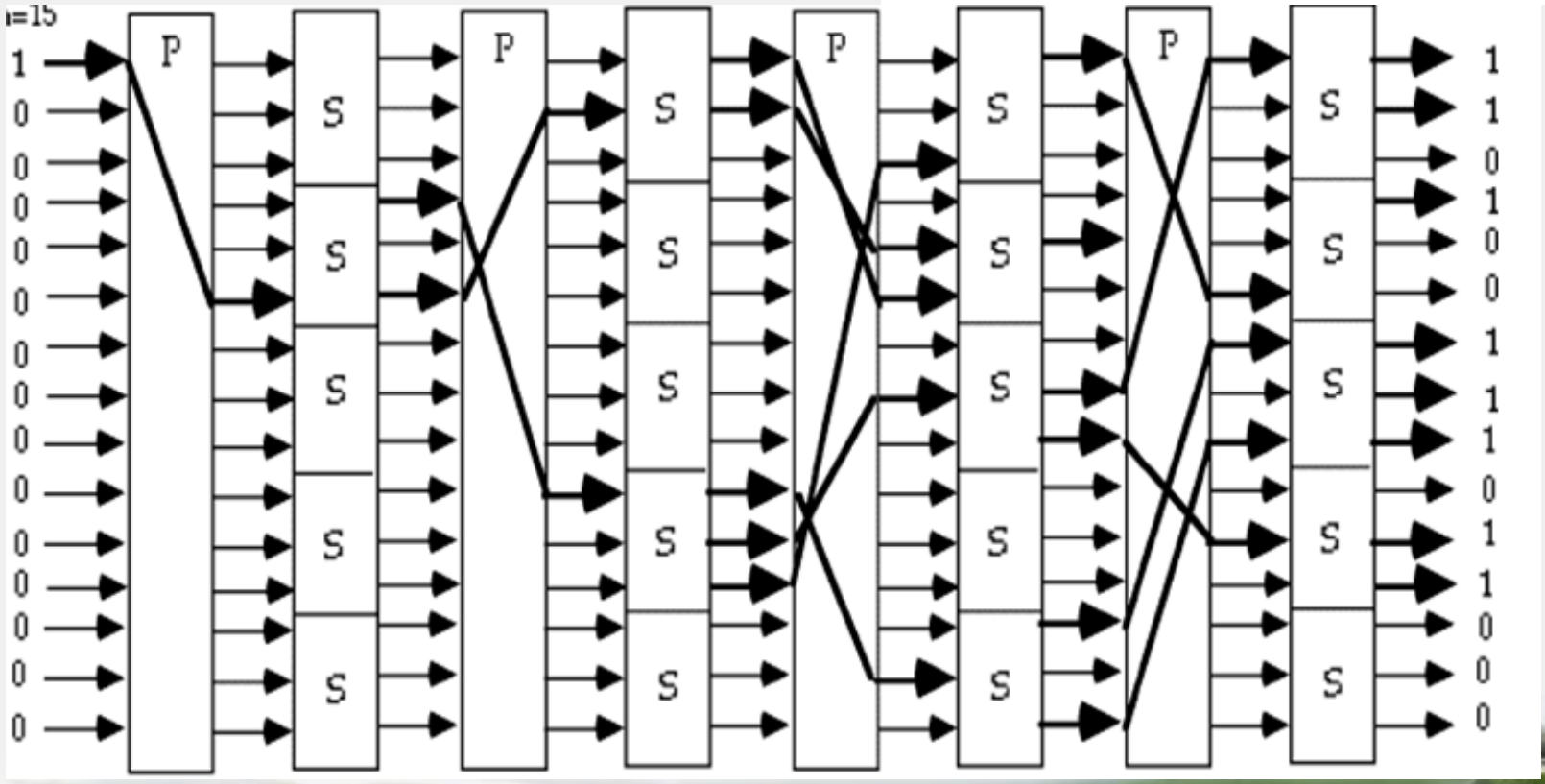
$$w^{0} \leftarrow g^{-1}(w^{1}, K^{1})$$

$$x \leftarrow w^{0}$$

Substitution-Permutation Network







代换-置换网络(SPN)

密码体制3.1 （代换-置换网络):

设l,m和Nr都是正整数，$\pi_s:\{0,1\}^l\to\{0,1\}^l$和$\pi_p:\{1,...,lm\}\to\{1,...,lm\}$都是置换。设$\mathrm{P}=\mathrm{C}=\{0,1\}^{lm}$，$K\subseteq(\{0,1\}^{lm})^{Nr+1}$是由初始密钥$K$用密码编排算法生成的所有可能的密钥编排方案之集。对一个密钥编排方案$K^1,\cdots,K^{Nr}$，我们使用下面的算法3.1来加密明文x。

代换-置换网络(SPN)

算法3.1 $SPN(x, \pi_s, \pi_p, (K^1, ..., K^{Nr+1}))$

$w^0 \leftarrow x$

for $r \leftarrow 1$ to $Nr-1$

do

$$\begin{cases} u^r \leftarrow w^{r-1} \oplus K^r \\ for\ i \leftarrow 1\ to\ m \\ do\ v^r_{<i>} \leftarrow \pi_s(u^r_{<i>}) \\ w^r \leftarrow (v^r_{\pi_p(1)}, \ldots, v^r_{\pi_p(lm)}) \end{cases}$$

$u^{Nr} \leftarrow w^{Nr-1} \oplus K^{Nr}$

for $i \leftarrow 1$ to m

do $v^{Nr}_{<i>} \leftarrow \pi_S(u^{Nr}_{<i>})$

$y \leftarrow v^{Nr} \oplus K^{Nr+1}$

output(y)

代换-置换网络(SPN)

在算法3.1中，u^r是第r轮对S盒的输入，v^r是第r轮对S盒的输出。w^r由v^r应用置换π_p得到，然后u^{r+1}由轮密钥K^{r+1}异或w^r得到（这叫做轮密钥混合），最后一轮没有用置换π_p。因此，如果对密钥编排方案做适当修改并用S盒的逆来取代S 盒，那么该加密算法也能用来解密。

该SPN的第一个和最后一个操作都是异或轮密钥，这叫做白化。白化可使一个不知道密钥的攻击者，无法开始进行一个加密或解密操作。

代换-置换网络(SPN)

Example 3.1 Suppose that $\ell = m = \mathrm{Nr} = 4$. Let π_S be defined as follows, where the input (i.e., z) and the output (i.e., $\pi_S(z)$) are written in hexadecimal notation, $(0 \leftrightarrow (0,0,0,0), 1 \leftrightarrow (0,0,0,1), \ldots, 9 \leftrightarrow (1,0,0,1), A \leftrightarrow (1,0,1,0), \ldots, F \leftrightarrow (1,1,1,1))$:

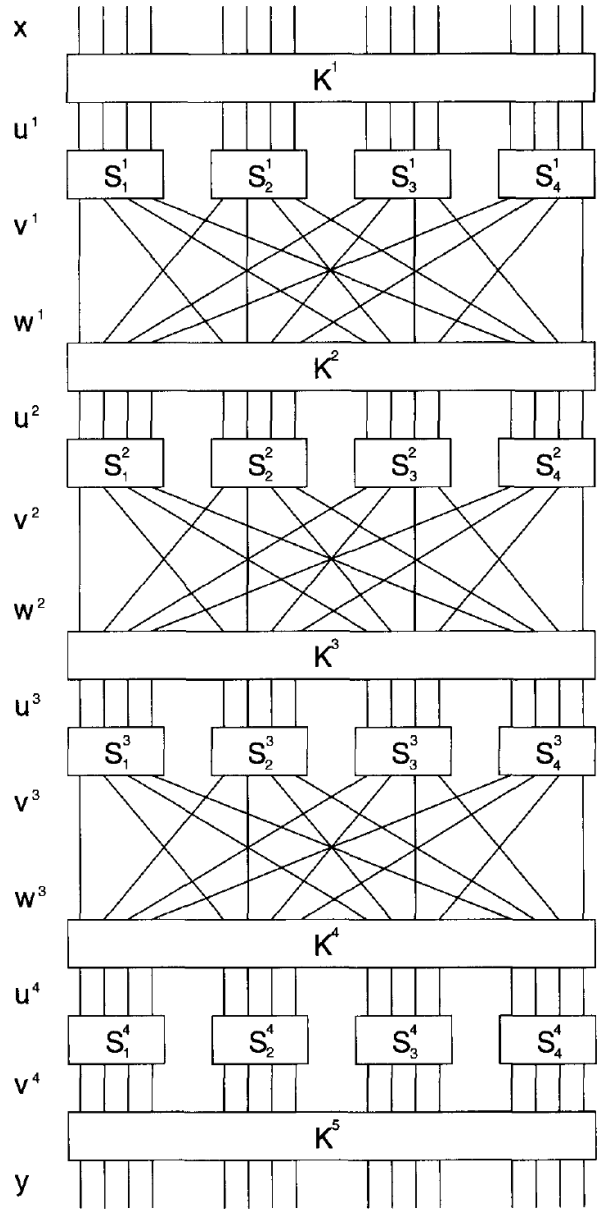
z	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\pi_S(z)$	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

Further, let π_P be defined as follows:

z	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\pi_P(z)$	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16

SPN实例

这个SPN中的16个s盒都采用πs，置换是πp



Now let's work out a sample encryption using this SPN. We represent all data in binary notation. Suppose the key is

$$K = 0011\ 1010\ 1001\ 0100\ 1101\ 0110\ 0011\ 1111.$$

Then the round keys are as follows:

$$K^1 = 0011\ 1010\ 1001\ 0100$$
$$K^2 = 1010\ 1001\ 0100\ 1101$$
$$K^3 = 1001\ 0100\ 1101\ 0110$$
$$K^4 = 0100\ 1101\ 0110\ 0011$$
$$K^5 = 1101\ 0110\ 0011\ 1111.$$

Suppose that the plaintext is

$$x = 0010\ 0110\ 1011\ 0111.$$

Then the encryption of x proceeds as follows:

$$w^0 = 0010\ 0110\ 1011\ 0111$$
$$K^1 = 0011\ 1010\ 1001\ 0100$$
$$u^1 = 0001\ 1100\ 0010\ 0011$$
$$v^1 = 0100\ 0101\ 1101\ 0001$$
$$w^1 = 0010\ 1110\ 0000\ 0111$$
$$K^2 = 1010\ 1001\ 0100\ 1101$$
$$u^2 = 1000\ 0111\ 0100\ 1010$$
$$v^2 = 0011\ 1000\ 0010\ 0110$$
$$w^2 = 0100\ 0001\ 1011\ 1000$$
$$K^3 = 1001\ 0100\ 1101\ 0110$$
$$u^3 = 1101\ 0101\ 0110\ 1110$$
$$v^3 = 1001\ 1111\ 1011\ 0000$$
$$w^3 = 1110\ 0100\ 0110\ 1110$$
$$K^4 = 0100\ 1101\ 0110\ 0011$$
$$u^4 = 1010\ 1001\ 0000\ 1101$$
$$v^4 = 0110\ 1010\ 1110\ 1001$$
$$K^5 = 1101\ 0110\ 0011\ 1111, \quad \text{and}$$
$$y = 1011\ 1100\ 1101\ 0110$$

is the ciphertext.

代换-置换网络(SPN)

SPN具有以下特色：（1）无论从硬件还是软件角度来看，这种设计均简单、有效。在软件方面，一个S盒通常以查表形式来实现。注意S盒所需的存储要求是$l2^l$比特。硬件实现需要使用相对小的S盒。在例3.1中，S盒的存储要求是$2^6 = 4 * 2^4$比特。

(2)SPN应该有更长的密钥长度和分组长度。比如AES（即Rijndael）就是SPN的一个例子。

(3)SPN有许多变体，比如使用不同的S盒，比如DES；在每一轮中包含一个可逆的线性变换，比如AES。

线性密码分析

在1993年的欧洲密码年会上，日本学者Mitsuru Matsui(松井充)提出了对DES算法的一种新的攻击方法，即线性密码分析，是一种已知明文攻击

Mitsuru Matsui to deliver 2018 IACR Distinguished Lecture

25 Years of Linear Cryptanalysis - Early History and Path Search Algorithm

Asiacrypt 2018, in Brisbane, Queensland, Australia.

Mitsuru Matsui studied mathematics and received the B.S. and M.S degrees from Kyoto University in 1985 and 1987, respectively. He joined Mitsubishi Electric Corporation in 1987 and has since been engaged in R&D of cryptography and information security. He earned the PhD in Information Science and Technology from the University of Tokyo in 2006. He is an executive fellow of Mitsubishi Electric

线性密码分析

原则上可用于任何的迭代密码。

假设能够在一个明文比特子集与最后一轮即将进行代换的输入状态比特子集之间找到一个概率线性关系，即存在一个比特子集使得其中元素的异或表现出非随机的分布（比如，该异或值以偏离1/2的概率取值0）。

现在假设一个攻击者拥有大量的用同一未知密钥K加密的明-密文对（即已知明文攻击）。对每一个明-密文对，将用所有可能的候选密钥来对最后一轮解密。对每一个候选密钥，计算包含在线性关系中的相关状态比特的异或值，然后确定上述的线性关系是否成立，如果成立，就在对应于特定候选密钥的计数器上加1，在这个过程的最后，我们希望计数频率离明-密文对数的一半最远的候选密钥含有那些密钥比特的正确值。

堆积原理

设$\mathbf{X}_1, \mathbf{X}_2, \ldots$是取值于集合$\{0,1\}$上的独立随机变量。设$p_1, p_2, \ldots$都是实数，且$0 \le p_i \le 1$，$i = 1,2,\ldots$，同时令

$$Pr[\mathbf{X}_i = 0] = p_i \quad i = 1,2,\ldots$$

则

$$Pr[\mathbf{X}_i = 1] = 1 - p_i \quad i = 1,2,\ldots$$

则$\mathbf{X}_i \oplus \mathbf{X}_j$具有如下概率分布：

$$Pr[\mathbf{X}_i \oplus \mathbf{X}_j = 0] = p_i p_j + (1-p_i)(1-p_j)$$

$$Pr[\mathbf{X}_i \oplus \mathbf{X}_j = 1] = p_i(1-p_j) + (1-p_i)p_j$$

对取值于$\{0,1\}$上的随机变量，用分布偏差来表示概率分布常常是很方便的。X_i的偏差被定义为：

$$\varepsilon_i = p_i - \frac{1}{2}$$

注意下列事实：对$i = 1,2,...$

$$-\frac{1}{2} \leq \varepsilon_i \leq \frac{1}{2}$$

$$Pr[\mathbf{X}_i = 0] = \frac{1}{2} + \varepsilon_i$$

$$Pr[\mathbf{X}_i = 1] = \frac{1}{2} - \varepsilon_i$$

引理3.1（堆积引理）设$\mathbf{X}_{i_1},\ldots,\mathbf{X}_{i_t}$是独立随机变量，$\varepsilon_{i_1,i_2,\ldots,i_t}(i_1<i_2<\ldots<i_k)$表示随机变量$\mathbf{X}_{i_1}\oplus\mathbf{X}_{i_2}\oplus\ldots\oplus\mathbf{X}_{i_t}$的偏差，则

$$\varepsilon_{i_1,i_2,\ldots,i_t}=2^{k-1}\prod_{j=1}^{k}\varepsilon_{i_j}$$

推论3.2设$\mathbf{X}_{i_1},\ldots,\mathbf{X}_{i_t}$是独立随机变量，$\varepsilon_{i_1,i_2,\ldots,i_t}(i_1<i_2<\ldots<i_k)$表示随机变量$\mathbf{X}_{i_1}\oplus\mathbf{X}_{i_2}\oplus\ldots\oplus\mathbf{X}_{i_t}$的偏差，若对某$j$，有$\varepsilon_{i_j}=0$，则$\varepsilon_{i_1,i_2,\ldots,i_t}=0$。

S盒的线性逼近

假设一个S盒$\pi_s : \{0,1\}^m \to \{0,1\}^n$。我们没有假定$\pi_s$是一个置换，甚至也未假定$m = n$。

m重输入$X = (x_1, ..., x_m)$均匀随机地从集合$\{0,1\}^m$中选取，即每一个坐标x_i定义一个随机变量$\mathbf{X}_i$，$\mathbf{X}_i$取值于$\{0,1\}$，并且偏差$\varepsilon_i = 0$。更进一步，这m个随机变量相互独立。

则：如果$(y_1, ..., y_n) \neq \pi_s(x_1, ..., x_m)$，则

$$Pr[\mathbf{X}_1 = x_1, ..., \mathbf{X}_m = x_m, \mathbf{Y}_1 = y_1, ..., \mathbf{Y}_n = y_n] = 0$$

如果$(y_1, ..., y_n) = \pi_s(x_1, ..., x_m)$，则

$$Pr[\mathbf{X}_1 = x_1, ..., \mathbf{X}_m = x_m, \mathbf{Y}_1 = y_1, ..., \mathbf{Y}_n = y_n] = 2^{-m}$$

$$Pr[\mathbf{Y}_1 = y_1, ..., \mathbf{Y}_n = y_n | \mathbf{X}_1 = x_1, ..., \mathbf{X}_m = x_m] = 1$$

S盒的线性逼近

- 例题

X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4
0	0	0	0	1	1	1	0
0	0	0	1	0	1	0	0
0	0	1	0	1	1	0	1
0	0	1	1	0	0	0	1
0	1	0	0	0	0	1	0
0	1	0	1	1	1	1	1
0	1	1	0	1	0	1	1
0	1	1	1	1	0	0	0
1	0	0	0	0	0	1	1
1	0	0	1	1	0	1	0
1	0	1	0	0	1	1	0
1	0	1	1	1	1	0	0
1	1	0	0	0	1	0	1
1	1	0	1	1	0	0	1
1	1	1	0	0	0	0	0
1	1	1	1	0	1	1	1

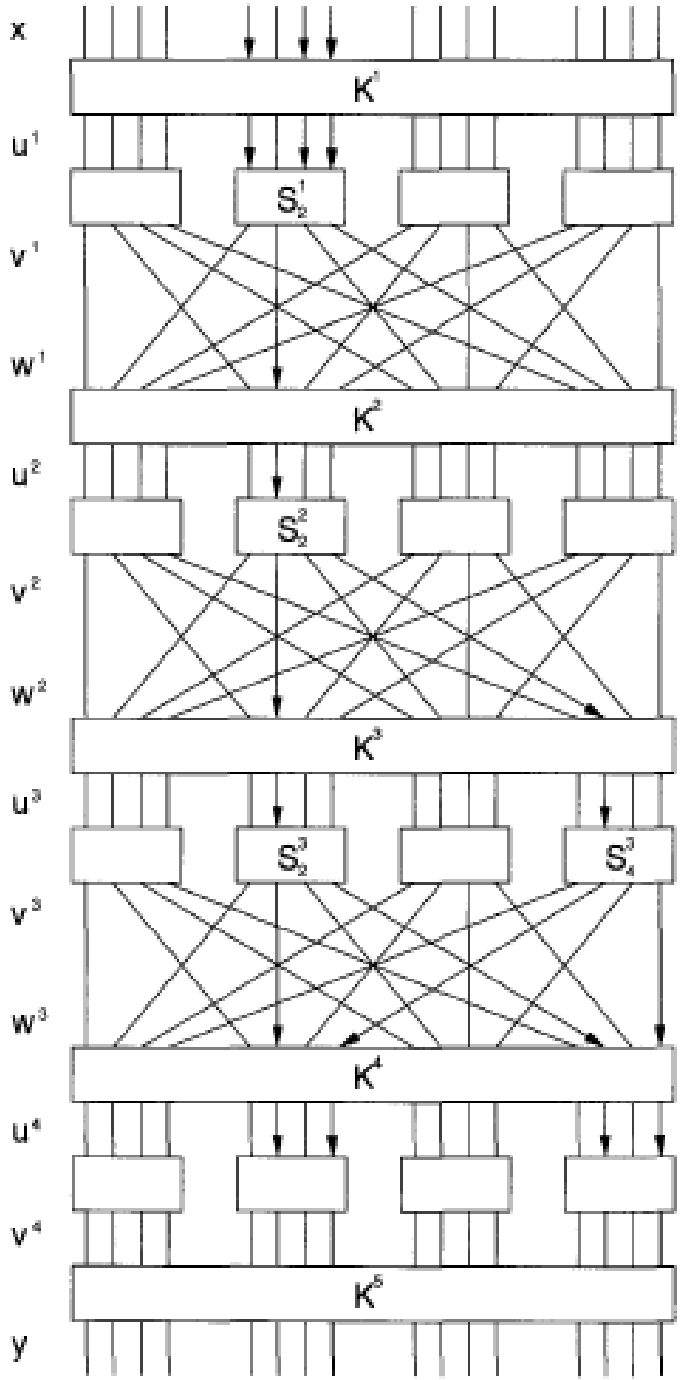
S盒的线性逼近

$$\left(\bigoplus_{i=1}^{4} a_i \mathbf{X_i}\right) \oplus \left(\bigoplus_{i=1}^{4} b_i \mathbf{Y_i}\right)$$

	b															
a	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
1	8	8	6	6	8	8	6	14	10	10	8	8	10	10	8	8
2	8	8	6	6	8	8	6	6	8	8	10	10	8	8	2	10
3	8	8	8	8	8	8	8	8	10	2	6	6	10	10	6	6
4	8	10	8	6	6	4	6	8	8	6	8	10	10	4	10	8
5	8	6	6	8	6	8	12	10	6	8	4	10	8	6	6	8
6	8	10	6	12	10	8	8	10	8	6	10	12	6	8	8	6
7	8	6	8	10	10	4	10	8	6	8	10	8	12	10	8	10
8	8	8	8	8	8	8	8	8	6	10	10	6	10	6	6	2
9	8	8	6	6	8	8	6	6	4	8	6	10	8	12	10	6
A	8	12	6	10	4	8	10	6	10	10	8	8	10	10	8	8
B	8	12	8	4	12	8	12	8	8	8	8	8	8	8	8	8
C	8	6	12	6	6	8	10	8	10	8	10	12	8	10	8	6
D	8	10	10	8	6	12	8	10	4	6	10	8	10	8	8	10
E	8	10	10	8	6	4	8	10	6	8	8	6	4	10	6	8
F	8	6	4	6	6	8	10	8	8	6	12	6	6	8	10	8

SPN的线性密码分析

- 线性密码分析要求找出一组S盒的线性逼近
- 借助这个线性逼近，导出整个SPN（除最后一轮）的线性逼近
- 利用已有的明密文对，测试候选密钥



- In S_2^1, the random variable $\mathbf{T_1} = \mathbf{U_5^1} \oplus \mathbf{U_7^1} \oplus \mathbf{U_8^1} \oplus \mathbf{V_6^1}$ has bias $1/4$
- In S_2^2, the random variable $\mathbf{T_2} = \mathbf{U_6^2} \oplus \mathbf{V_6^2} \oplus \mathbf{V_8^2}$ has bias $-1/4$
- In S_2^3, the random variable $\mathbf{T_3} = \mathbf{U_6^3} \oplus \mathbf{V_6^3} \oplus \mathbf{V_8^3}$ has bias $-1/4$
- In S_4^3, the random variable $\mathbf{T_4} = \mathbf{U_{14}^3} \oplus \mathbf{V_{14}^3} \oplus \mathbf{V_{16}^3}$ has bias $-1/4$

$$\mathbf{T_1} \oplus \mathbf{T_2} \oplus \mathbf{T_3} \oplus \mathbf{T_4}$$

has bias equal to $2^3(1/4)(-1/4)^3 = -1/32$.

$$\begin{aligned}
\mathbf{T_1} &= \mathbf{U_5^1} \oplus \mathbf{U_7^1} \oplus \mathbf{U_8^1} \oplus \mathbf{V_6^1} = \mathbf{X_5} \oplus \mathbf{K_5^1} \oplus \mathbf{X_7} \oplus \mathbf{K_7^1} \oplus \mathbf{X_8} \oplus \mathbf{K_8^1} \oplus \mathbf{V_6^1} \\
\mathbf{T_2} &= \mathbf{U_6^2} \oplus \mathbf{V_6^2} \oplus \mathbf{V_8^2} \qquad = \mathbf{V_6^1} \oplus \mathbf{K_6^2} \oplus \mathbf{V_6^2} \oplus \mathbf{V_8^2} \\
\mathbf{T_3} &= \mathbf{U_6^3} \oplus \mathbf{V_6^3} \oplus \mathbf{V_8^3} \qquad = \mathbf{V_6^2} \oplus \mathbf{K_6^3} \oplus \mathbf{V_6^3} \oplus \mathbf{V_8^3} \\
\mathbf{T_4} &= \mathbf{U_{14}^3} \oplus \mathbf{V_{14}^3} \oplus \mathbf{V_{16}^3} \qquad = \mathbf{V_8^2} \oplus \mathbf{K_{14}^3} \oplus \mathbf{V_{14}^3} \oplus \mathbf{V_{16}^3}.
\end{aligned}$$

$$\begin{aligned}
&\mathbf{X_5} \oplus \mathbf{X_7} \oplus \mathbf{X_8} \oplus \mathbf{V_6^3} \oplus \mathbf{V_8^3} \oplus \mathbf{V_{14}^3} \oplus \mathbf{V_{16}^3} \\
&\qquad \oplus \mathbf{K_5^1} \oplus \mathbf{K_7^1} \oplus \mathbf{K_8^1} \oplus \mathbf{K_6^2} \oplus \mathbf{K_6^3} \oplus \mathbf{K_{14}^3} \qquad (3.1)
\end{aligned}$$

$$\begin{aligned}
\mathbf{V_6^3} &= \mathbf{U_6^4} \oplus \mathbf{K_6^4} \\
\mathbf{V_8^3} &= \mathbf{U_{14}^4} \oplus \mathbf{K_{14}^4} \\
\mathbf{V_{14}^3} &= \mathbf{U_8^4} \oplus \mathbf{K_8^4} \\
\mathbf{V_{16}^3} &= \mathbf{U_{16}^4} \oplus \mathbf{K_{16}^4}
\end{aligned}$$

$$\begin{aligned}
&\mathbf{X_5} \oplus \mathbf{X_7} \oplus \mathbf{X_8} \oplus \mathbf{U_6^4} \oplus \mathbf{U_8^4} \oplus \mathbf{U_{14}^4} \oplus \mathbf{U_{16}^4} \\
&\quad \oplus \mathbf{K_5^1} \oplus \mathbf{K_7^1} \oplus \mathbf{K_8^1} \oplus \mathbf{K_6^2} \oplus \mathbf{K_6^3} \oplus \mathbf{K_{14}^3} \oplus \mathbf{K_6^4} \oplus \mathbf{K_8^4} \oplus \mathbf{K_{14}^4} \oplus \mathbf{K_{16}^4} \qquad (3.2)
\end{aligned}$$

$$\mathbf{X_5} \oplus \mathbf{X_7} \oplus \mathbf{X_8} \oplus \mathbf{U_6^4} \oplus \mathbf{U_8^4} \oplus \mathbf{U_{14}^4} \oplus \mathbf{U_{16}^4} \qquad (3.3)$$

线性密码分析要求找出一组S盒的线性逼近，这组线性逼近能够用来导出一个整个SPN（除最后一轮）的线性逼近。

通过图3.3的分析，我们得到

$$\mathbf{T}_1 \oplus \mathbf{T}_2 \oplus \mathbf{T}_3 \oplus \mathbf{T}_4 = \mathbf{X}_5 \oplus \mathbf{X}_7 \oplus \mathbf{X}_8 \oplus \mathbf{U}_6^4 \oplus \mathbf{U}_8^4 \oplus \mathbf{U}_{14}^4 \oplus \mathbf{U}_{16}^4$$

$$\oplus \mathbf{K}_5^1 \oplus \mathbf{K}_7^1 \oplus \mathbf{K}_8^1 \oplus \mathbf{K}_6^2 \oplus \mathbf{K}_6^3 \oplus \mathbf{K}_{14}^3 \oplus \mathbf{K}_6^4 \oplus \mathbf{K}_8^4 \oplus \mathbf{K}_{14}^4 \oplus \mathbf{K}_{16}^4 \qquad (3.2)$$

因为密钥比特固定，所以事实上

$$\mathbf{K}_5^1 \oplus \mathbf{K}_7^1 \oplus \mathbf{K}_8^1 \oplus \mathbf{K}_6^2 \oplus \mathbf{K}_6^3 \oplus \mathbf{K}_{14}^3 \oplus \mathbf{K}_6^4 \oplus \mathbf{K}_8^4 \oplus \mathbf{K}_{14}^4 \oplus \mathbf{K}_{16}^4$$

具有固定的值。

因此我们可以使用随机变量

$$\mathbf{X}_5 \oplus \mathbf{X}_7 \oplus \mathbf{X}_8 \oplus \mathbf{U}_6^4 \oplus \mathbf{U}_8^4 \oplus \mathbf{U}_{14}^4 \oplus \mathbf{U}_{16}^4$$

具有偏离0的偏差这一事实允许我们进行线性密码攻击。

假设我们拥有同一未知密钥K加密的T对明-密文。用T来表示T对明-密文的集合。线性攻击将使我们获得$K_{<2>}^5$和$K_{<4>}^5$的8比特密钥，即

$$K_5^5, K_6^5, K_7^5, K_8^5, K_{13}^5, K_{14}^5, K_{15}^5, K_{16}^5$$

这些正是与S盒S_2^4和S_4^4的输出相异或的8比特密钥。256种可能，每一种可能都叫做一个候选子密钥。

对每一个$(x,y)\in$T及每一个子密钥，计算y的一个部分解密并得到$u^4_{<2>}$和$u^4_{<4>}$，然后计算

$$x_5 \oplus x_7 \oplus x_8 \oplus u^4_6 \oplus u^4_8 \oplus u^4_{14} \oplus u^4_{16} \quad (3.4)$$

之值。保持对应于这256个候选子密钥的256个计数器，每当式(3.4)取值为0时，就将对应于该子密钥的计数器加1。

在计数过程中，我们希望大多数的计数器接近于T/2，而真正的候选子密钥对应的计数器具有接近于$T/2\pm T/32$之值，这有助于我们确定正确的8个子密钥比特。