NBC 分组密码算法设计文档

徐洪 段明 谭林 戚文峰 王中孝

信息工程大学

2019 年 10 月

摘　要

本报告包括NBC算法的具体描述、设计原理、安全性分析、性能分析等。

NBC算法整体采用改进的第二类广义Feistel结构，支持128/128，128/256，256/256三个不同版本的分组长度和密钥长度。算法以16比特字为基本运算单元，非线性S盒基于16 级非线性反馈移位寄存器构造，扩散层采用简单的16比特字间的置换，选择了扩散效果最好的一组置换。分组长度为128比特时，整体结构为8分支广义Feistel结构，分组长度为256比特时，整体结构为16分支广义Feistel结构。

NBC算法各部件都采用了轻量化的设计，轮函数结构简单，扩散层采用了16比特字间的置换，非线性S盒基于16级非线性反馈移位寄存器构造，只用到3个与，1个与非和8个异或运算，具有非常低的硬件成本，也便于进行侧信道防护。软件实现时S盒变换也可以通过查表实现。

算法针对差分分析、线性分析、不可能差分分析、积分分析、零相关线性分析等主要分析方法都有较充足的安全冗余。

目 录

1. 算法描述 3
1.1 符号说明 3
1.2 轮函数 4
1.3 S 盒 5
1.4 密钥扩展算法 6
2. 设计原理 7
2.1 轮函数的设计 7
2.2 S 盒设计 8
2.3 密钥扩展算法的设计 9
3. 安全性分析 10
3.1 差分分析 10
3.2 线性分析 13
3.3 不可能差分分析 16
3.4 零相关线性分析 20
3.5 积分分析 24
3.6 总体评估 27
4. 性能分析 27
5. 优缺点分析 28

1. 算法描述

NBC 算法采用改进的第二类广义 Feistel 结构，轮函数如图 1.1.1 和图 1.1.2 所示，其中 S 为 16 比特的 S 盒，π 为 16 比特块的置换。

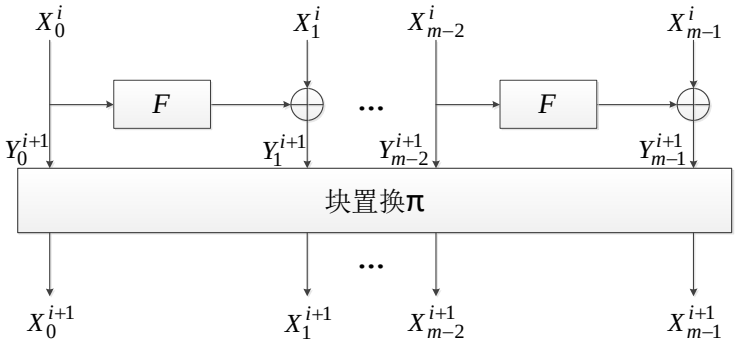

图 1.1.1　NBC 算法的轮函数

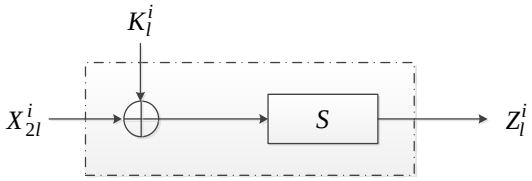

图 1.1.2　F 函数 $(0 \le l \le m/2-1)$

算法以 16 比特为基本处理单元，以下将分组长度为 16m 比特，密钥长度为 16n 比特的 NBC 算法简记为 NBC 16m /16n 算法。算法支持 128/128，128/256，256/256 等三种分组和密钥规模，迭代轮数分别为 32，34，38，基本参数见下表。当不强调密钥长度时，也简单记 NBC 128 为分组长度为 128 比特的 NBC 算法，NBC 256 为分组长度为 256 比特的 NBC 算法。

表 1.1.1　NBC 算法基本参数

算法版本	分组长度	密钥长度	轮数	m	n
NBC 128/128	128	128	32	8	8
NBC 128/256	128	256	34	8	16
NBC 256/256	256	256	38	16	16

1.1 符号说明

X^i ：第 i 轮的输入

X_j^i ：X^i 的第 j 个块，$0 \le j \le m-1$

K^i ：第 i 轮的轮子密钥

K_j^i ：K^i 的第 j 个块，$0 \le j \le m/2-1$

⊕：异或

⊗：比特与

+：模 2^n 加法

<<<s：循环左移 s 比特

>>>s：循环右移 s 比特

1.2 轮函数

对任意 $0 \le i \le 31$，设第 i 轮的输入为 $X^i = (X_0^i || X_1^i || \cdots || X_{m-1}^i)$，输出为 $X^{i+1} = (X_0^{i+1} || X_1^{i+1} || \cdots || X_{m-1}^{i+1})$，第 i 轮的轮子密钥为 $K^i = (K_0^i || K_1^i || \cdots || K_{m/2-1}^i)$，则有

$$Y_{2l}^{i+1} = X_{2l}^i,\ Y_{2l+1}^{i+1} = S(X_{2l}^i \oplus K_l^i) \oplus X_{2l+1}^i,\quad 0 \le l \le \frac{m}{2} - 1,$$

$$X^{i+1} = \pi(Y^{i+1}),\quad 0 \le i \le 30,$$

$$X^{32} = Y^{32}\ 。$$

为保证加解密的一致性，最后一轮不进行块置换。输入明文为 $P = X^0$，输出密文为 $C = X^{32}$。

对于 128 和 256 比特分组，相应的块置换π分别为

$$\pi_8 = (30147256),$$

$$\pi_{16} = (7\ 2\ 13\ 4\ 11\ 8\ 3\ 6\ \ 15\ 0\ 9\ 10\ 1\ 14\ \ 5\ 12),$$

其中(30147256)中的 3 表示输入的第 0 块移到输出的第 3 块的位置，其余依此类推。图 1.2.1 和 1.2.2 分别为 128 和 256 比特分组的 NBC 算法的轮函数示意图。

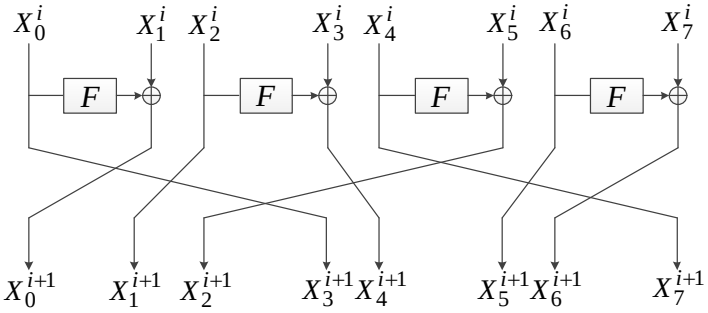

图 1.2.1　NBC 128 算法的轮函数

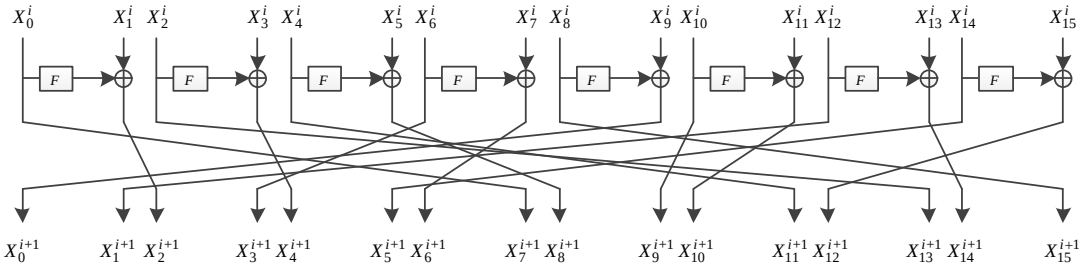

图 1.2.2　NBC 256 算法的轮函数

1.3 S 盒

NBC 算法的 S 盒基于 16 级的非线性反馈移位寄存器（NFSR）来构造，参见图 1.3.1。设 S 盒的 16 比特输入为$s_0 s_1 \ldots s_{15}$，将该 16 比特从左至右依次填充至 16 个寄存器内，寄存器迭代 20 拍后的全体内部状态即为 S 盒的输出。

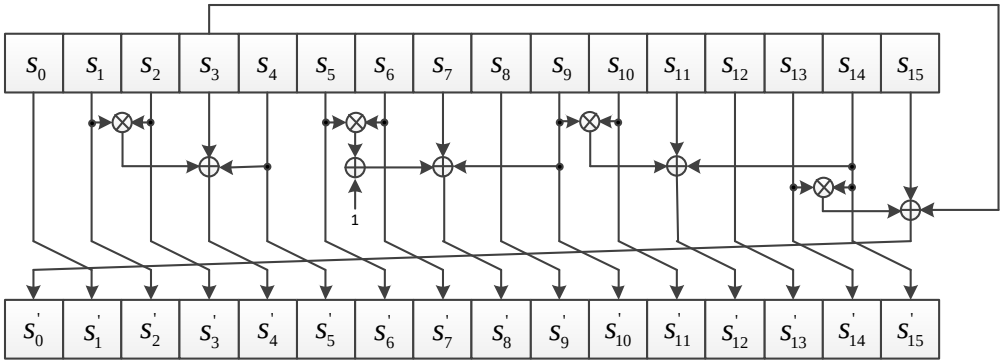

图 1.3.1　用于构造 S 盒的非线性反馈移位寄存器

寄存器每迭代一拍，除常规移位外以非线性方式同时更新 4 个比特，设某时刻寄存器的输入为 $s_0, s_1, \ldots, s_{15}$，输出为 $s_0', s_1', \ldots, s_{15}'$，则其状态更新函数如下：

$$s_4' = s_3 \oplus (s_1 \otimes s_2) \oplus s_4,$$

$$s_8' = s_7 \oplus (s_5 \otimes s_6) \oplus 1 \oplus s_9,$$

$$s_{12}' = s_{11} \oplus (s_9 \otimes s_{10}) \oplus s_{14},$$

$$s_0' = s_{15} \oplus (s_{13} \otimes s_{14}) \oplus s_3,$$

$$s_i' = s_{i-1},\ i \in \{1,2,3,\ 5,6,7,\ 9,10,11,\ 13,14,15\}。$$

寄存器迭代 20 拍可以保证 S 盒的各分量的代数次数都达到最大值 15，并且具有

好的差分和线性性质。

1.4 密钥扩展算法

密钥扩展算法的设计与 S 盒类似，采用基于 n 比特字的 16 级非线性反馈移位寄存器，其中 128 比特密钥对应的 $n=8$，256 比特密钥对应的 $n=16$。密钥扩展算法中引入模 2^n 加法代替比特与，增加了循环移位以保证不同字之间的充分混淆，增加了轮常数以避免出现弱密钥（如全零子密钥）。密钥扩展算法中用到的非线性反馈移位寄存器如图 1.4.1 所示，图中的状态字 w_j 和常数 c_j 中各比特都是按照高位在前，低位在后的顺序存放。

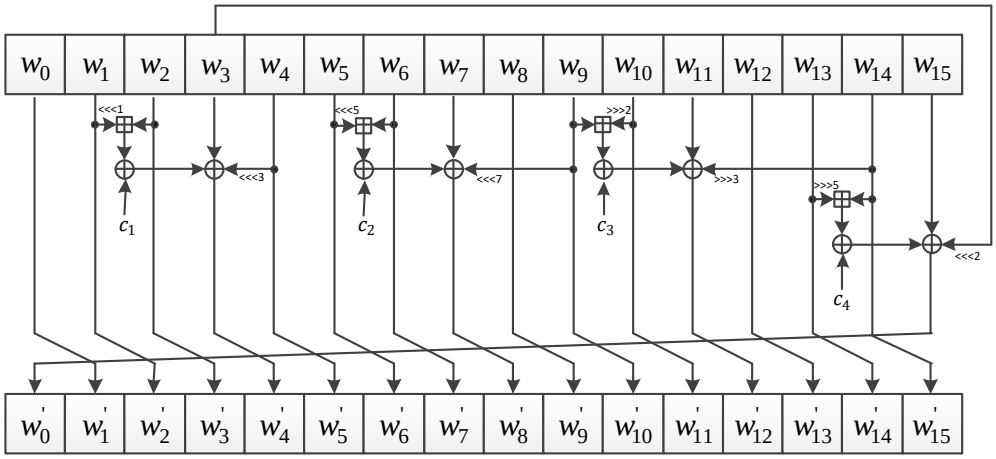

图 1.4.1 密钥扩展算法中用到的非线性反馈移位寄存器

初始种子密钥为 $16n$ 比特，分成 16 个 n 比特字 $K=(k_0\|k_1\|\cdots\|k_{15})$。用它们填充密钥寄存器的初态，即有 $w_j=k_j$，$0\le j\le 15$。提取密钥寄存器的前 $8m$ 比特作为初始轮的轮子密钥。之后寄存器每迭代 4 拍，输出一轮轮子密钥，每次选取密钥寄存器的前 $8m$ 比特作为轮子密钥。要产生 32 轮加密的所有轮子密钥，密钥寄存器总共需要迭代 124 拍。轮子密钥的具体产生方式如下：

(1) 对 $0\le j\le 15$，令 $w_j=k_j$；

(2) 输出密钥寄存器的前 $8m$ 比特作为第 0 轮的轮子密钥 K^0。

(3) 对 $1\le i\le 31$，顺序执行下列步骤，产生后 31 轮的轮子密钥 K^i：

(3.1)寄存器迭代 4 拍，按下列方式更新寄存器状态：

$$w_4' = ((w_1 <<< 1) + w_2) \oplus w_3 \oplus (w_4 <<< 3) \oplus c_1,$$

$$w_8' = ((w_5 <<< 5) + w_6) \oplus w_7 \oplus (w_9 <<< 7) \oplus c_2,$$

$$w_{12}' = (w_9 + (w_{10} >>> 2)) \oplus w_{11} \oplus (w_{14} >>> 3) \oplus c_3,$$

$$w_0' = ((w_{13} >>> 5) + w_{14}) \oplus w_{15} \oplus (w_3 <<< 2) \oplus c_4,$$

$$w_i' = w_{i-1},\ i \in \{1,2,3\} \cup \{5,6,7\} \cup \{9,10,11\} \cup \{13,14,15\};$$

(3.2) 选取密钥寄存器的前 $8m$ 比特作为第 i 轮的轮子密钥 K^i。

三种情况下输出的轮子密钥分别为：

128/128：$K^i = (w_0 \| w_1 \| w_1 \| w_3 \| w_4 \| w_5 \| w_6 \| w_7) = (K_0^i || K_1^i || K_2^i || K_3^i)$，

128/256：$K^i = (w_0 \| w_1 \| w_1 \| w_3) = (K_0^i || K_1^i || K_2^i || K_3^i)$，

256/256：$K^i = (w_0 \| w_1 \| w_1 \| w_3 \| w_4 \| w_5 \| w_6 \| w_7) = (K_0^i || K_1^i || \cdots || K_7^i)$。

密钥扩展算法中用到的轮常数 c_j $(0 \le j \le 3)$为随机数，256 比特密钥时用到的 4 个 16 比特随机数分别为：

$$c_1 = 0x5A82 = (0101\ 1010\ 1000\ 0010),$$
$$c_2 = 0x6ED9 = (0110\ 1110\ 1101\ 1001),$$
$$c_3 = 0xA953 = (1010\ 1001\ 0101\ 0011),$$
$$c_4 = 0xCA62 = (1100\ 1010\ 0110\ 0010),$$

它们分别为$\sqrt{2}/4$，$\sqrt{3}/4$，$\sqrt{7}/4$，$\sqrt{10}/4$ 的小数部分的前 16 比特。128 比特密钥时用到的 4 个 8 比特随机数分别为它们的前 8 比特，即有：

$c_1 = 0x5A = (0101\ 1010)$，$c_2 = 0x6E = (0110\ 1110)$，
$c_3 = 0xA9 = (1010\ 1001)$，$c_4 = 0xCA = (1100\ 1010)$。

2. 设计原理

2.1 轮函数的设计

轮函数采用 Suzaki 等在 FSE 2010 会议上给出的改进型的第二类广义 Feistel（GFS）结构，扩散层选用了新的块置换，该结构比扩散层采用循环移位的第二

类广义 Feistel 结构扩散效果更好。比如，m 分支的第二类广义 Feistel 结构需要 m 轮才能完全扩散，而采用这种改进版的结构，最优情况下 8 分支仅需要 7 轮，16 分支仅需要 8 轮就可以完全扩散。兼顾最小活动 S 盒个数达到最优，对于 8 分支和 16 分支的广义 Feistel 结构，我们分别采用了 Suzaki 等给出的第 1 个和第 10 个最优扩散层实例。它们不同轮数对应的最小活动 S 盒个数如下表。

表 2.1.1 不同轮数最小活动 S 盒个数（NBC 128 算法，8 分支 GFS）

轮数	8	9	10	11	12	13	14	15	16	20	32
活动 S 盒个数	11	12	14	16	18	19	20	21	23	30	50

表 2.1.2 不同轮数最小活动 S 盒个数（NBC 256 算法，16 分支 GFS）

轮数	8	9	10	11	12	13	14	15	16	20	24	28
活动 S 盒数	11	14	18	22	24	27	30	32	35	44	53	62

当 S 盒的差分和线性性质较好时，经过适当轮迭代后可以保证算法具有足够的抵抗差分和线性分析的能力。

2.2 S 盒设计

在 S 盒的构造上，我们采用了 Galois 结构的 16 级非线性移位寄存器（NFSR）来构造轻量化的 16 比特 S 盒，除常规移位外每次以非线性方式更新 4 个比特。硬件实现时仅需要 3 个与门，1 个与非门和 8 个异或，约需要 $3\times1.25+1\times1+8\times2=20.75$ 个标准门电路。

寄存器迭代 20 拍后寄存器各个比特关于初始状态的代数次数都可以达到 15 次。构造出的 S 盒的差分均匀度为 $\mathrm{Diff}(S)=22$，最大差分概率为 $22/2^{16}\approx2^{-11.541}$，线性度为 $\mathrm{Lin}(S)=1572$，最大线性概率为 $1572/2^{16}\approx2^{-5.382}$。这里的差分均匀度和线性度分别为：

$$\mathrm{Diff}(S)=\max_{0\neq\alpha\in F_2^n}\max_{\beta\in F_2^n}\#\{X\in F_2^n: S(X\oplus\alpha)\oplus S(X)=\beta\},$$

$$\mathrm{Lin(S)}=\max_{\alpha\in F_2^n}\max_{0\neq\beta\in F_2^n}|\#\{X\in F_2^n:\beta\cdot S(X)=\alpha\cdot X\}-\#\{X\in F_2^n:\beta\cdot S(X)\neq\alpha\cdot X\}|。$$

作为比较，我们也考虑了其它类型的 16 比特 S 盒的构造。由于基于域$F_{2^{16}}$上的逆函数或者基于 8 比特 S 盒迭代构造 16 比特 S 盒的实现成本较高，我们从 4 比特轻量 S 盒出发，基于 4 分支广义 Feistel 结构和 SPN 结构迭代构造了一些轻量 16 比特 S 盒实例。表 2.2.1 列出了不同构造方式下 S 盒各分量的代数次数都达到最大值 15 时需要的迭代轮数，以及相应的差分均匀度、线性度和需要的门电路。

表 2.2.1 不同方式迭代构造的 16 比特 S 盒的性能比较

	迭代轮数	差分均匀度	线性度	门电路估计
NFSR	20	22	1572	20.75
广义 Feistel	10	22	1952	65.04
SPN	4	20	1728	247.56

从表中数据可以看出，采用三种方式迭代构造的 S 盒的差分和线性密码性质大致相当。采用 SPN 结构需要的迭代轮数最少，但每轮需要 4 个 4 比特 S 盒和一个具有较大分支数的扩散矩阵，其硬件实现成本最高，而采用我们基于 NFSR 构造的 S 盒需要的门电路数最少，综合性能最好。

2.3 密钥扩展算法的设计

密钥扩展算法采用了与 S 盒构造类似的基于 n 比特字的 16 级非线性反馈移位寄存器结构，128 比特和 256 比特密钥情况下，字长分别为 8 比特和 16 比特。寄存器非线性部分的比特与运算改成模 2^n 加法运算，同时引入循环移位使得寄存器各个字之间可以更快扩散。寄存器中引入轮常数以抵抗滑动攻击，并避免出现弱密钥。测试结果显示，即使输入为全 0 密钥比特，至多经过 6 轮迭代后寄存器的状态比特已经平衡；而当输入密钥有 1 比特差分时，至多经过 15 轮迭代后，密钥寄存器的状态比特有一半有差分。

由于密钥寄存器每迭代 4 拍才输出一轮轮子密钥，每次只输出密钥寄存器的前一半或者四分之一比特，因此同一轮子密钥的不同块之间、前后轮轮子密钥的

不同块之间不存在简单的依赖关系，这也增加了密钥恢复过程的复杂度。由于密钥寄存器按非线性方式迭代更新，仅知道部分子密钥块仍然难以有效恢复全部主密钥。

3. 安全性分析

本节从差分分析、线性分析、不可能差分分析、零相关线性分析和积分分析等方面给出 NBC 算法抵抗这些主要分析方法的能力评估。

3.1 差分分析

不考虑 S 盒的具体差分分布，利用 MILP 方法，我们可以得到 NBC 算法不同轮数最小差分活动 S 盒个数如下：

表 3.1.1 NBC 算法不同轮数最小差分活动 S 盒个数

轮数	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
NBC 128	0	1	2	3	4	6	8	11	12	14	16	18	19	20	21	23
NBC 256	0	1	2	3	4	6	8	11	14	18	22	24	27	30	32	35

结合 NBC 算法中 S 盒的具体差分分布，容易给出 NBC 算法的差分分析如下：

3.1.1 NBC 128 算法的差分分析

NBC 算法中使用的 S 盒的最大差分概率约为 $2^{-11.541}$，对于 NBC 128 算法，由表 3.1.1 知，该算法 9 轮迭代后至少有 12 个差分活动 S 盒，故相应的 9 轮最大差分概率为：

$$p_9 \le (2^{-11.541})^{12} \approx 2^{-138.492} < 2^{-128}。$$

因此，NBC 128 算法经过 9 轮迭代后可以抵抗基本差分分析，而算法总的迭代轮数至少为 32 轮，有足够的安全冗余。

进而，根据 NBC 128 算法的结构特点，我们还可以构造如图 3.1.1 和表 3.1.2 所示的 NBC 128 算法的 7 轮差分路径。

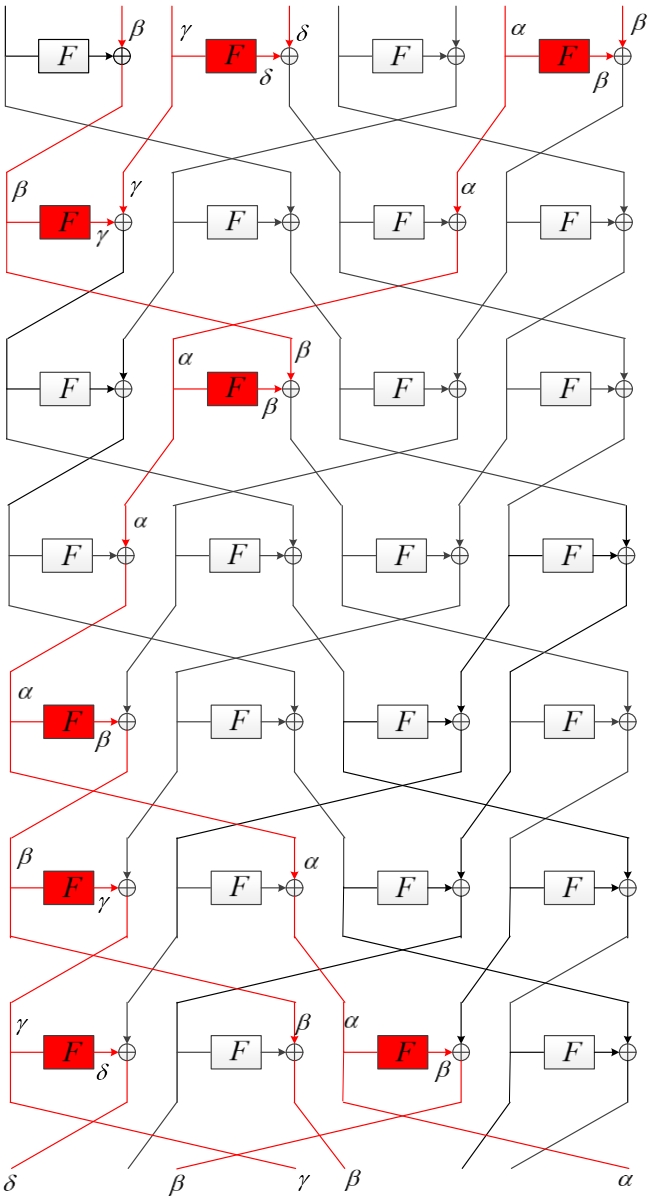

图 3.1.1 NBC 128 算法的 7 轮差分路径

表 3.1.2 NBC 128 算法的 7 轮差分及概率

轮数	输出差分值	单轮差分概率	总差分概率
0	$(0, \beta, \gamma, \delta, 0, 0, \alpha, \beta)$		$p(\alpha, \beta)^4\, p(\beta, \gamma)^2\, p(\gamma, \delta)^2$
1	$(\beta, \gamma, 0, 0, 0, \alpha, 0, 0)$	$p(\gamma, \delta)\, p(\alpha, \beta)$	
2	$(0, 0, \alpha, \beta, 0, 0, 0, 0)$	$p(\beta, \gamma)$	
3	$(0, \alpha, 0, 0, 0, 0, 0, 0)$	$p(\alpha, \beta)$	
4	$(\alpha, 0, 0, 0, 0, 0, 0, 0)$	1	
5	$(\beta, 0, 0, \alpha, 0, 0, 0, 0)$	$p(\alpha, \beta)$	
6	$(\gamma, 0, 0, \beta, \alpha, 0, 0, 0)$	$p(\beta, \gamma)$	
7	$(\delta, 0, \beta, \gamma, \beta, 0, 0, \alpha)$	$p(\gamma, \delta)\, p(\alpha, \beta)$	

如果 NBC 128 算法中 S 盒的输入输出差分满足 $\alpha \to \beta \to \gamma \to \delta$，相应的差分概率分别为 $p(\alpha,\beta), p(\beta,\gamma), p(\gamma,\delta), p(\delta,\xi), p(\xi,\eta)$。记初始轮输入差分为 $(0, \beta, \gamma, \delta, 0, 0, \alpha, \beta)$，则一轮加密后的输出差分为 $(\beta, \gamma, 0, 0, 0, \alpha, 0, 0)$的概率为 $p(\gamma, \delta)\, p(\alpha, \beta)$。经过第二轮加密后，输出差分为$(0, 0, \alpha, \beta, 0, 0, 0, 0)$的概率为 $p(\beta, \gamma)$。类似可以得到如表 3.1.2 所示的完整的 7 轮差分值及相应概率。从表 3.1.2 可以看出，7 轮差分路径

$$(0, \beta, \gamma, \delta, 0, 0, \alpha, \beta) \to_7 (\delta, 0, \beta, \gamma, \beta, 0, 0, \alpha)$$

成立的概率为

$$p_7 = p(\alpha, \beta)^4\, p(\beta, \gamma)^2\, p(\gamma, \delta)^2 \text{ 。}$$

验证发现，当 $(\alpha, \beta, \gamma, \delta) = (0500, c00d, 004c, 284f)$时(十六进制表示)，上述 7 轮差分特征成立的概率为

$$p_7 = \frac{18^4 \cdot 10^2 \cdot 12^2}{2^{16 \cdot 8}} \approx 2^{-97.507}\text{。}$$

对于 NBC 128/128 算法，将此区分器向后扩展 2 轮，猜测 7 个子密钥块共 112 比特密钥，可以实现 9 轮 NBC 128/128 算法的密钥恢复攻击。对于 NBC 128/256 算法，将此区分器向后扩展 4 轮，猜测 15 个子密钥块共 240 比特密钥，可以实现 11 轮 NBC 128/256 算法的密钥恢复攻击。

3.1.2 NBC 256 算法的差分分析

对于 NBC 256 算法，由表 3.1.1 知，该算法经过 12 轮迭代后至少有 24 个差分活动 S 盒，相应的 12 轮最大差分概率为：

$$p_{12} \le (2^{-11.541})^{24} \approx 2^{-276.984} < 2^{-256},$$

因此，NBC 256 算法经过 12 轮迭代后可以抵抗基本差分分析和线性分析，而算法总迭代轮数为 38 轮，有足够的安全冗余。

进而，根据 NBC 256 算法的结构特点，我们也可以类似给出 NBC 256 算法的差分路径。如果 NBC 256 算法中 S 盒的输入输出差分满足 $\alpha \to \beta \to \gamma \to \delta \to$

$\xi \to \eta$，相应的差分概率分别为 $p(\alpha,\beta), p(\beta,\gamma), p(\gamma,\delta), p(\delta,\xi), p(\xi,\eta)$，则可以构成如表 3.1.3 所示的 10 轮差分路径，各轮差分值及相应的概率如下。

表 3.1.3 NBC 256 算法的 10 轮差分及概率

轮数	输出差分值	单轮差分概率
0	(γ, δ, γ, δ, γ, δ, 0, 0, 0, δ, 0, β, ξ, η, α, β)	
1	(δ, ξ, 0, 0, 0, α, 0, γ, 0, 0, β, γ, β, γ, 0, 0)	$p(\alpha,\beta)\, p(\gamma,\delta)^3\, p(\xi,\eta)$
2	(0, β, 0, 0, 0, 0, γ, δ, α, β, 0, 0, 0, 0, 0, 0)	$p(\beta,\gamma)^2\, p(\delta,\xi)$
3	(0, 0, β, γ, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, α)	$p(\alpha,\beta)\, p(\gamma,\delta)$
4	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, α, β)	$p(\beta,\gamma)$
5	(0, α, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)	$p(\alpha,\beta)$
6	(0, 0, α, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)	1
7	(0, 0, 0, 0, β, 0, 0, 0, 0, 0, 0, 0, 0, α, 0, 0)	$p(\alpha,\beta)$
8	(0, 0, 0, 0, 0, 0, 0, 0, γ, 0, 0, β, 0, 0, α, 0)	$p(\beta,\gamma)$
9	(δ, 0, 0, 0, 0, α, 0, 0, 0, 0, β, 0, β, 0, 0, γ)	$p(\alpha,\beta)\, p(\gamma,\delta)$
10	(0, β, ξ, 0, 0, 0, 0, δ, α, β, γ, 0, γ, 0, γ, 0)	$p(\beta,\gamma)^2\, p(\delta,\xi)$
(γ, δ, γ, δ, γ, δ, 0, 0, 0, δ, 0, β, ξ, η, α, β) $\to_{10}$ (0, β, ξ, 0, 0, 0, 0, δ, α, β, γ, 0, γ, 0, γ, 0)		
10 轮差分概率 $p(\alpha,\beta)^5\, p(\beta,\gamma)^6\, p(\gamma,\delta)^5\, p(\delta,\xi)^2\, p(\xi,\eta)$		

验证发现，当 (α, β, γ, δ, ξ,η) = (011f, 1a01, 800b, 0dd4, 18fa, 1f38)时 (十六进制表示)，上述 10 轮差分特征成立的概率为

$$p_{10} = \frac{10^5 \cdot 22^6 \cdot 10^5 \cdot 10^2 \cdot 12}{2^{16 \cdot 19}} \approx 2^{-233.795}。$$

对于 NBC 256/256 算法，将此区分器向后扩展 2 轮，猜测 12 个子密钥块共 192 比特密钥，可以实现 12 轮 NBC 256/256 算法的密钥恢复攻击。

3.2 线性分析

不考虑 S 盒的具体线性分布规律，利用 MILP 方法，我们可以得到 NBC 算法不同轮数最小线性活动 S 盒个数如下：

表 3.2.1 NBC 算法不同轮数最小线性活动 S 盒个数

轮数	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
NBC 128	0	1	2	3	4	6	8	11	12	14	16	18	19	20	21	23
NBC 256	0	1	2	3	4	6	8	11	14	18	22	24	27	30	32	35

结合 NBC 算法中 S 盒的具体线性分布，容易给出 NBC 算法的线性分析如下：

3.2.1 NBC 128 算法的线性分析

NBC 算法中 S 盒的最大线性概率约为 $2^{-5.382}$，对于 NBC 128 算法，由表 3.2.1 知，该算法经过 9 轮迭代后至少有 12 个线性活动 S 盒，故相应的 9 轮最大线性概率为：

$$q_9 \le (2^{-5.382})^{16} \approx 2^{-64.584} < 2^{-64},$$

因此 NBC 128 算法经过 9 轮迭代后可以抵抗基本线性分析，而算法总的迭代轮数至少为 32 轮，有足够的安全冗余。

类似于差分路径的构造，根据 NBC 128 算法的结构特点，我们也可以构造如表 3.2.2 所示的 NBC 128 算法的 7 轮线性路径。假定 NBC 128 算法中 S 盒的输入输出掩码满足 $\alpha \to \beta \to \gamma \to \delta$, 相应的线性概率分别为 $q(\alpha,\beta), q(\beta,\gamma), q(\gamma,\delta)$。表 3.2.2 列出了 7 轮线性路径 $(\alpha, \beta, \gamma, 0, \gamma, \delta, 0, 0) \to_7 (0, \alpha, 0, \gamma, \beta, \gamma, \delta, 0)$ 各中间状态的掩码值和相应的线性概率。

表 3.2.2 NBC 128 算法的 7 轮线性路径及概率

轮数	输出掩码值	单轮线性概率	总线性概率
0	$(\alpha, \beta, \gamma, 0, \gamma, \delta, 0, 0)$		$q(\alpha, \beta)^2\, q(\beta, \gamma)^2\, q(\gamma, \delta)^4$
1	$(\beta, \gamma, \delta, 0, 0, 0, 0, 0)$	$q(\alpha, \beta)\, q(\gamma, \delta)$	
2	$(\gamma, \delta, 0, 0, 0, 0, 0, 0)$	$q(\beta, \gamma)$	
3	$(\delta, 0, 0, 0, 0, 0, 0, 0)$	$q(\gamma, \delta)$	
4	$(0, 0, 0, \delta, 0, 0, 0, 0)$	1	
5	$(0, \gamma, 0, 0, \delta, 0, 0, 0)$	$q(\gamma, \delta)$	
6	$(\gamma, 0, 0, \beta, 0, 0, 0, \delta)$	$q(\beta, \gamma)$	
7	$(0, \alpha, 0, \gamma, \beta, \gamma, \delta, 0)$	$q(\alpha, \beta)\, q(\gamma, \delta)$	
7 轮线性路径 $(\alpha, \beta, \gamma, 0, \gamma, \delta, 0, 0) \to (0, \alpha, 0, \gamma, \beta, \gamma, \delta, 0)$			

验证发现，当 $(\alpha, \beta, \gamma, \delta) = (0x0001, 0xbeb2, 0x4fa3, 0x3867)$ 时，上述 7 轮线性特征成立的概率为

$$q_7 = \frac{944^2 \cdot 920^2 \cdot 948^4}{2^{16 \cdot 8}} \approx 2^{-48.989}。$$

对于 NBC 128/128 算法，将此区分器向后扩展 2 轮，猜测 7 个子密钥块共 112 比特密钥，可以实现 9 轮 NBC 128/128 算法的密钥恢复攻击。对于 NBC 128/256 算法，将此区分器向后扩展 4 轮，猜测 15 个子密钥块共 240 比特密钥，可以实现 11 轮 NBC 128/256 算法的密钥恢复攻击。

3.2.2 NBC 256 算法的线性分析

对于 NBC 256 算法，由表 3.2.1 知，该算法经过 12 轮迭代后至少有 24 个线性活动 S 盒，相应的 12 轮最大线性概率为：

$$q_{12} \le (2^{-5.382})^{24} \approx 2^{-129.168} < 2^{-128},$$

因此，NBC 256 算法经过 12 轮迭代后可以抵抗基本线性分析，而算法总迭代轮数为 38 轮，有足够的安全冗余。

根据 NBC 256 算法的结构特点，我们也可以类似给出 NBC 256 算法的线性路径。如果 NBC 256 算法中 S 盒的输入输出掩码满足 $\alpha \to \beta \to \gamma \to \delta \to \xi \to \eta$，线性概率分别为 $q(\alpha,\beta), q(\beta,\gamma), q(\gamma,\delta), q(\delta,\xi), q(\xi,\eta)$，则可以如下构造如表 3.2.3 所示的 NBC 256 算法的 10 轮线性路径。

表 3.2.3 NBC 256 算法的 10 轮线性路径

轮数	输出掩码值	单轮线性概率
0	(0, 0, 0, η, ξ, 0, γ, δ, α, β, γ, δ, γ, 0, γ, δ)	
1	(β, γ, 0, 0, η, 0, δ, 0, 0, 0, δ, ξ, δ, ξ, 0, 0)	$q(\alpha,\beta)\, q(\gamma,\delta)^3\, q(\xi,\eta)$
2	(0, 0, γ, δ, 0, 0, 0, 0, 0, 0, ξ, η, 0, 0, ξ, 0)	$q(\beta,\gamma)\, q(\delta,\xi)^2$
3	(0, 0, 0, 0, δ, ξ, 0, 0, 0, 0, η, 0, 0, 0, 0, 0)	$q(\gamma,\delta)\, q(\xi,\eta)$
4	(0, 0, 0, 0, 0, 0, 0, 0, ξ, η, 0, 0, 0, 0, 0, 0)	$q(\delta,\xi)$
5	(η, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)	$q(\xi,\eta)$
6	(0, 0, 0, 0, 0, 0, 0, η, 0, 0, 0, 0, 0, 0, 0, 0)	1
7	(0, 0, 0, ξ, 0, 0, η, 0, 0, 0, 0, 0, 0, 0, 0, 0)	$q(\xi,\eta)$
8	(0, 0, 0, η, ξ, 0, 0, 0, 0, 0, 0, 0, 0, 0, δ, 0, 0)	$q(\delta,\xi)$
9	(0, γ, 0, 0, η, 0, 0, 0, 0, 0, 0, ξ, 0, ξ, δ, 0)	$q(\gamma,\delta)\, q(\xi,\eta)$

10	$(0, \delta, \gamma, 0, 0, \delta, 0, \beta, \quad 0, \delta, \xi, \eta, 0, 0, \xi, 0)$	$q(\beta,\gamma)\, q(\delta,\xi)^2$
$(0, 0, 0, \eta, \xi, 0, \gamma, \delta, \alpha, \beta, \gamma, \delta, \gamma, 0, \gamma, \delta) \rightarrow (0, \delta, \gamma, 0, 0, \delta, 0, \beta, 0, \delta, \xi, \eta, 0, 0, \xi, 0)$		
10 轮线性概率 $q(\alpha,\beta)\, q(\beta,\gamma)^2\, q(\gamma,\delta)^5\, q(\delta,\xi)^6\, q(\xi,\eta)^5$		

验证发现，当 $(\alpha, \beta, \gamma, \delta, \xi, \eta)$ =(0x0001,0xbeb2,0x4fa3,0x3867,0x4bbf,0x7e77) 时（十六进制表示），上述 10 轮线性特征成立的概率为

$$q_{10} = \frac{944 \cdot 920^2 \cdot 948^5 \cdot 924^6 \cdot 996^5}{2^{16 \cdot 19}} \approx 2^{-116.072}。$$

对于 NBC 256/256 算法，将此区分器向后扩展 2 轮，猜测 12 个子密钥块共 192 比特密钥，可以实现 12 轮 NBC 256/256 算法的密钥恢复攻击。

3.3 不可能差分分析

本节采用 Boura 等在 2014 年亚密会上提出的不可能差分分析的一般理论模型给出对 NBC 算法的不可能差分分析。记$|\Delta_{in}|$和$|\Delta_{out}|$分别为输入差分Δ_{in}和输出差分Δ_{out}中非零块对应的总比特数，c_{in}和 c_{out}分别表示部分加密和部分解密过程中需要满足过滤条件的比特数（匹配比特数），$|k_{in}|$和$|k_{out}|$分别表示部分加密和部分解密过程中需要猜测的实际密钥比特数。再记 N 为密钥筛选过程具有指定输入差分和输出差分的明文对数，它必须满足：$N \geq 2^{c_{in}+c_{out}}$。利用 N 组明文对筛选后错误密钥留下的概率为 $P = (1 - 2^{c_{in}+c_{out}})^N \approx e^{-N/2^{c_{in}+c_{out}}}$，不可能差分分析过程中需要的选择明文数为

$$C_N = \max\left\{\min_{\Delta \in \{\Delta_{in}, \Delta_{out}\}} \left\{\sqrt{N 2^{n+1-|\Delta|}}\right\}, N 2^{n+1-|\Delta_{in}|-|\Delta_{out}|}\right\},$$

计算复杂度为

$$T_{comp} = \left(C_N + \left(N + 2^{|k_{in} \cup k_{out}|} \cdot \frac{N}{2^{c_{in}+c_{out}}}\right) C_E{}' + 2^{|K|} P\right) C_E$$

其中 n 为明文比特数，$C_E{}'$为部分加密和部分解密过程的计算量占整体计算量的比例，C_E为整体计算量大小，存储复杂度为$\min\{N, 2^{|k_{in} \cup k_{out}|}\}$。

3.3.1 NBC 128 算法的分析

利用中间相错方法，容易验证当 $a \neq b$ 时，NBC 128 算法存在形如 $(0a000000) \nrightarrow_{11} (0000b000)$ 的 11 轮不可能差分特征。表 3.3.1 给出了该 11 轮不可能差分特征的矛盾产生过程，其中 0 表示没有差分，*表示有非零差分，？表示不确定的差分，a, b 为两个不同的非零差分。输入差分$\Delta X = (0a000000)$加密 6 轮，输出差分$\Delta Y = (0000b000)$解密 5 轮后在倒数第 3 个块的差分出现矛盾。

表 3.3.1　NBC 128 算法的 11 轮不可能差分特征

	轮数	状态差分	轮数	状态差分	
加密过程	0	0*a*00 0000	6′	*?** 0***b*****	解密过程
	1	*a*000 0000	7	****b** 0*00	
	2	*00*a* 0000	8	0*b*** 0000	
	3	*00* *a*000	9	*b**00 0000	
	4	*0** *00*a*	10	000*b* 0000	
	5	**** ?0*a**	11	0000 *b*000	
	6	?*?* ?***a***??			

图 3.3.1 给出了从 11 轮不可能差分特征 $(0a000000) \nrightarrow_{11} (0000b000)$ 出发，前后各添加 4 轮的 19 轮不可能差分分析的密钥恢复过程，图中标注了密钥恢复过程中需要用到的中间状态差分和相关的子密钥。

表 3.3.2 列出了密钥恢复过程中需要用到的中间状态差分、猜测的子密钥以及差分匹配情况，其中初始明文差分为(**** *?0*)，最终密文差分为(***? **0*)，ΔF_j^i表示第 i 轮第 j 个 F 函数的输出差分，最后一轮没有进行块置换。

表 3.3.2　NBC 128/256 算法 19 轮不可能差分分析的密钥恢复过程

轮数	状态差分	猜测子密钥	猜测密钥比特	匹配关系	匹配数
0	**** *?0*				
1	0*** 00**	K_0^0, K_1^0	32	$\Delta F_0^0 = \Delta X_1^0$ $\Delta F_1^0 = \Delta X_3^0$	32
2	**00 0*00	K_1^1, K_3^1	32	$\Delta F_1^1 = \Delta X_3^1$ $\Delta F_3^1 = \Delta X_7^1$	32
3	00** 0000	K_0^2	16	$\Delta F_0^2 = \Delta X_1^2$	16
4	**0*a*00 0000**	K_1^3	16	$\Delta F_1^3 = \Delta X_3^3$	16
15	**0000 *b*000**	K_2^{15}	16	$\Delta F_2^{15} = \Delta X_2^{16}$	16

16	00*0 000*	K_1^{16}	16	$\Delta F_1^{16} = \Delta X_4^{17}$	16
17	0*00 *0*0	K_2^{17}, K_3^{17}	32	$\Delta F_2^{17} = \Delta X_2^{18}$ $\Delta F_3^{17} = \Delta X_6^{18}$	32
18	*0*0 0***	$K_0^{18}, K_1^{18}, K_3^{18}$	48	$\Delta F_0^{18} = \Delta X_1^{19}$ $\Delta F_1^{18} = \Delta X_3^{19}$	32
19	**** 0**?				

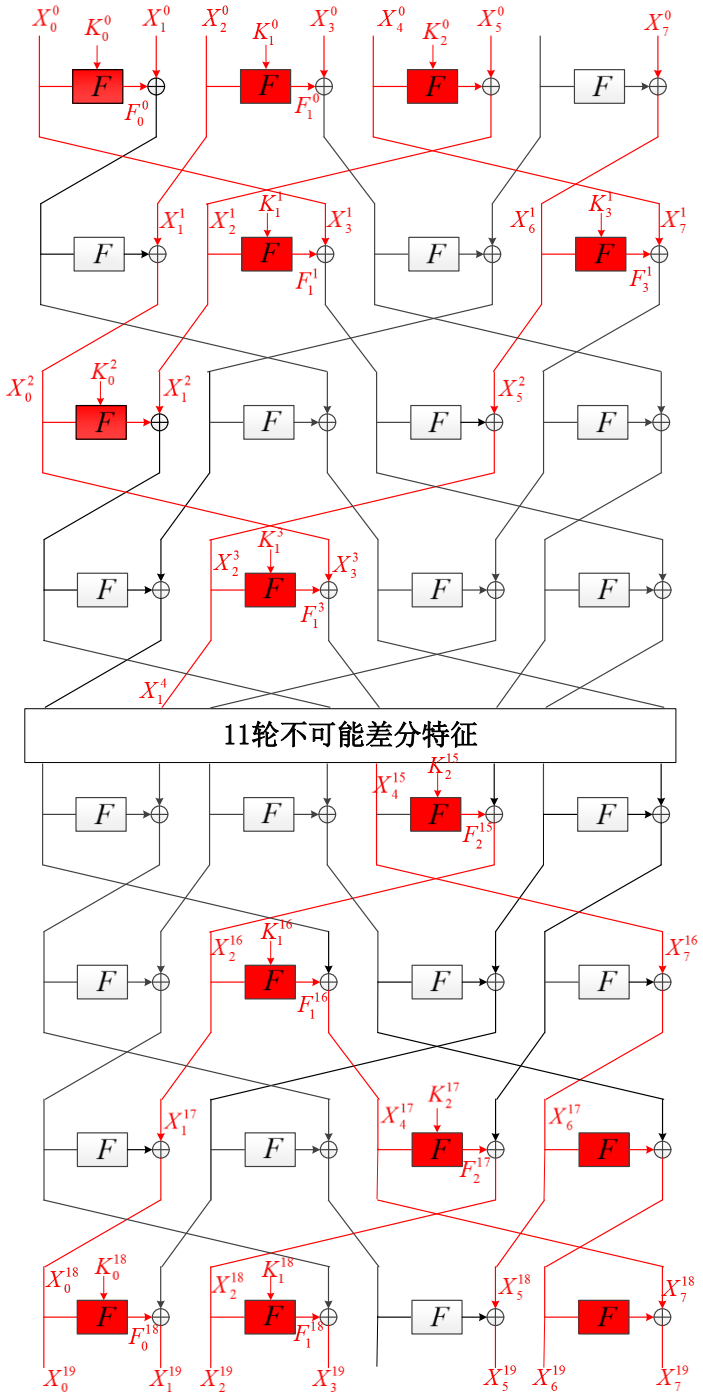

图 3.3.1 NBC 128/256 算法的 19 轮不可能差分分析

根据 Boura 等不可能差分分析的一般理论模型，从选定的不可能差分特

征 $(0a000000) \nrightarrow_{11} (0000b000)$ 出发，分别向前后扩展 3 轮和 4 轮时相关的状态差分数、匹配比特数和密钥比特数如下：

扩展轮数	$\lvert\Delta_{in}\rvert$	$\lvert\Delta_{out}\rvert$	c_{in}	c_{out}	$\lvert k_{in}\rvert$	$\lvert k_{out}\rvert$
3	80	80	64	64	64	64
4	112	112	96	96	112	112

当算法的密钥长度为 256 比特时，按照上面的分析，前后分别扩展 4 轮可以给出 NBC 128/256 算法的 19 轮不可能差分分析，此时有

$$|\Delta_{in}| = |\Delta_{out}| = 112，\ c_{in} = c_{out} = 96，\ |k_{in} \cup k_{out}| = 224，$$

部分加解密过程中总共需要计算 14 次 S 盒的差分，相当于 $2 \cdot 14/(4 \cdot 19) = 2^{-1.44}$ 次 19 轮加密，故$C_E' = 2^{-1.44}$。密钥恢复过程中至少需要 2^{192} 个正确对，当 $N = 2^{197}$ 时攻击效果最好，此时需要的选择明文数 $C_N = 2^{(197+128+1-112)/2} = 2^{107}$ (选择差分形如(**** 0**?)的明文)，利用这些正确对进行密钥筛选后错误密钥留下的概率为

$$P = (1 - 2^{c_{in}+c_{out}})^N \approx e^{-N/2^{c_{in}+c_{out}}} \approx 2^{-1.44 \cdot 32} \approx 2^{-46.08}，$$

攻击需要的总的计算复杂度为

$$T_{comp} = (2^{107} + 2^{197} \cdot (1+2^{32}) \cdot 2^{-1.44} + 2^{256} \cdot 2^{-46.08}) \approx 2^{227.56} \text{ 次 19 轮加密。}$$

当算法的密钥长度为 128 比特时，根据 Boura 等的方法，前后各扩展 3 轮，可以给出 NBC 128/128 算法的 17 轮不可能差分分析，此时

$$|\Delta_{in}| = |\Delta_{out}| = 80，\ c_{in} = c_{out} = 64，\ |k_{in} \cup k_{out}| = 160，$$

部分加解密过程中总共需要计算 8 次 S 盒的差分，相当于 $2 \cdot 8/(4 \cdot 17) = 2^{-2.09}$ 次 17 轮加密，故$C_E' = 2^{-2.09}$。密钥恢复过程中至少需要 2^{128} 个正确对，当 $N = 2^{128}$ 时，攻击效果最好，此时需要的选择明文数量为 $C_N = 2^{128+128+1-112 \cdot 2} = 2^{97}$ (即需要选择 2^{17} 个差分为 (0*00 ***0) 的选择明文结构)，利用这些正确对进行密钥筛选后错误密钥留下的概率为 $P \approx e^{-1} \approx 2^{-1.44}$，攻击需要的总的计算复杂度为

$$T_{comp} = (2^{97} + 2^{128} \cdot (1+1) \cdot 2^{-2.09} + 2^{128} \cdot 2^{-1.44}) \approx 2^{127.75} \text{ 次 17 轮加密。}$$

综上可见，存在 NBC 128/128 算法的 3+11+3 = 17 轮不可能差分分析，计算

复杂度约为 $2^{127.75}$ 次 17 轮加密；存在 NBC 128/256 算法的 4+11+4=19 轮不可能差分分析，计算复杂度约为 $2^{227.56}$ 次 19 轮加密。

3.2.2 NBC 256 算法的分析

利用中间相错方法，容易验证 NBC 256 算法存在形如 $(0\alpha_1 0\alpha_3 0\alpha_5 0\alpha_7 0\alpha_9 0\alpha_{11} 0\alpha_{13} 0\alpha_{15}) \nrightarrow_{10} (\beta_0 0\beta_2 0\beta_4 0\beta_6 0\beta_8 0\beta_{10} 0\beta_{12} 0\beta_{14} 0)$ 的 14 轮不可能差分特征，其中 α_i, β_j 仅一个非零。特别的，从 14 轮不可能差分特征 (0*00000000000000) $\nrightarrow_{10}$ (*000000000000000)出发，类似于前面的分析，分别向前后扩展 4 轮可以给出 NBC- 256 算法的 22 轮不可能差分分析，相应的状态差分数、匹配比特数和密钥比特数如下：

扩展轮数	$\lvert\Delta_{in}\rvert$	$\lvert\Delta_{out}\rvert$	c_{in}	c_{out}	$\lvert k_{in}\rvert$	$\lvert k_{out}\rvert$
4	128	128	112	112	112	112

部分加解密过程中总共需要计算 14 次 S 盒的差分，相当于 $2\cdot 14/(4\cdot 22) = 2^{-2.65}$ 次 22 轮加密，故 $C_E' = 2^{-2.65}$。密钥恢复过程中至少需要 2^{224} 个正确对，当 $N = 2^{229}$ 时攻击效果最好，此时需要的选择明文数 $C_N = 2^{229+256+1-128\cdot 2} = 2^{230}$ (即需要选择 2^{112} 个差分为(**00 0*0* 00** **00)的选择明文结构)，利用这些正确对进行密钥筛选后，错误密钥留下的概率为

$$P = (1 - 2^{c_{in}+c_{out}})^N \approx e^{-N/2^{c_{in}+c_{out}}} \approx 2^{-1.44\cdot 32} \approx 2^{-46.08},$$

攻击需要的总的计算复杂度为

$$T_{comp} = (2^{230} + 2^{229}\cdot(1+1)\cdot 2^{-2.65} + 2^{256}\cdot 2^{-46.08}) \approx 2^{230.21}$$ 次 22 轮加密。

综上可见，存在 NBC 256/256 算法的 4+14+4 = 22 轮不可能差分分析，计算复杂度约为 $2^{230.21}$ 次 22 轮加密。

3.4 零相关线性分析

3.4.1 NBC 128 算法的分析

利用中间相错方法，容易验证当 $a \neq b$ 时，NBC 128 算法存在形如 $(a0000000) \rightarrow_{11} (0000000b)$ 的 32 维 11 轮零相关线性特征。表 3.4.1 给出了该 11

轮零相关线性特征的矛盾产生过程，其中 0 表示 0 掩码，*表示非零掩码，？表示不确定的掩码，a, b 表示两个不同的非零掩码。输入掩码Γ_X=(a0000000)加密 6 轮，输出掩码Γ_Y=(0000000b) 解密 5 轮后在倒数第 3 个块的掩码值出现矛盾。

表 3.4.1　NBC 128 算法的 11 轮零相关线性特征

	轮数	掩码值	轮数	掩码值	
加密过程	0	a000 0000	6′	?*$\boldsymbol{b}$0 ****	解密过程
	1	000a 0000	7	*b** 00*0	
	2	0*00 a000	8	b000 **00	
	3	*00* 000a	9	00*b 0000	
	4	0*0* **a0	10	0000 b000	
	5	**** *a0?	11	0000 000b	
	6	*?$\boldsymbol{a}$? *???			

图 3.4.1 给出了从 11 轮零相关线性特征(a0000000) $\rightarrow_{11}$ (0000000b) 出发，前面添加 2 轮，后面添加 3 轮的 16 轮零相关线性分析的密钥恢复过程，图中列出了计算X_0^2和X_1^{12}时需要用到的中间状态和子密钥。

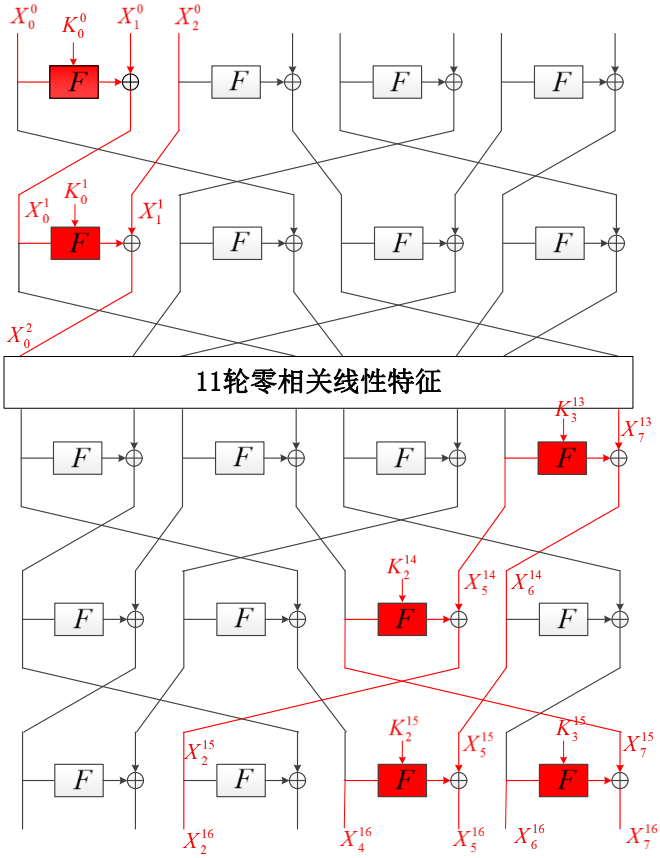

图 3.4.1　NBC 128/256 算法的 16 轮零相关线性分析

表 3.4.2 16 轮零相关线性分析密钥恢复过程的复杂度

步骤	猜测子密钥	计算量	状态空间	状态数
0			$x_0=(\boldsymbol{X_0^0X_1^0}\|\|X_2^{16}X_4^{16}X_5^{16}X_6^{16}X_7^{16})$	$\min\{N,2^{128}\}$
1	K_0^0	$\min\{N\cdot 2^{16}, 2^{128}\cdot 2^{16}\}$	$x_1=(\boldsymbol{X_0^1X_1^1}\|\|X_2^{16}X_4^{16}X_5^{16}X_6^{16}X_7^{16})$	2^{112}
2	K_0^1	$2^{112}\cdot 2^{32}=2^{144}$	$x_2=(X_0^2\|\|X_2^{16}\boldsymbol{X_4^{16}X_5^{16}}X_6^{16}X_7^{16})$	2^{96}
3	K_2^{15}	$2^{96}\cdot 2^{48}=2^{144}$	$x_3=(X_0^2\|\|X_2^{16}X_5^{15}\boldsymbol{X_6^{16}X_7^{16}})$	2^{80}
4	K_3^{15}	$2^{80}\cdot 2^{64}=2^{144}$	$x_4=(X_0^2\|\|\boldsymbol{X_2^{15}}X_5^{15}\boldsymbol{X_7^{15}})$	2^{64}
5	K_2^{14}	$2^{64}\cdot 2^{80}=2^{144}$	$x_5=(X_0^2\|\|\boldsymbol{X_5^{14}X_6^{14}})$	2^{48}
6	K_3^{13}	$2^{48}\cdot 2^{96}=2^{144}$	$x_6=(X_0^2\|\|X_7^{13})$	2^{32}

表 3.4.2 给出了部分加解密过程需要猜测的子密钥及相应的复杂度。从表中可以看出，部分加解密过程共需要猜测 6 个子密钥块共 96 比特，计算复杂度约为 $6\cdot 2^{144}/(4\cdot 16)\approx 2^{140.59}$ 次 16 轮加密，存储复杂度为 $\min\{N, 2^{128}\}$ 个 128 比特字。

按照 Bogdanov 等在 2012 年亚密会上给出的多维零相关线性分析的方法，从 32 维 10 轮零相关线性特征$(a0000000)\rightarrow_{11}(0000000b)$出发，猜测部分子密钥 k，部分加解密得到中间状态 $x_6=(X_0^2\|\|X_7^{13})$，选择 32 个独立的线性掩码基向量 (u_i, v_i)，计算 $z_i=u_iX_0^2+v_iX_7^{13}$。令 $z=(z_1,\ldots,z_{32})$，并记 $V[z]$ 为数组 z 出现的个数。再令

$$T_k=\frac{N\cdot 2^{32}}{1-2^{-32}}\cdot\sum_{z=0}^{2^{32}-1}\left(\frac{V(z)}{N}-\frac{1}{2^{32}}\right)^2$$

当 k 为正确密钥时，统计量 T_k 服从均值为 $\mu_0=(2^{32}-1)\cdot\frac{2^{128}-N}{2^{128}-1}$，方差为 $\sigma_0^2=2\cdot(2^{32}-1)\cdot\left(\frac{2^{128}-N}{2^{128}-1}\right)^2$ 的 χ^2 分布；当 k 为错误密钥时，统计量 T_k 服从均值为 $\mu_1=2^{32}-1$，方差为 $\sigma_1^2=2\cdot(2^{32}-1)$ 的 χ^2 分布。当 N 足够大时，它们可以用正态分布来近似。

设弃真概率为 α，取伪概率为 β，并记 $z_{1-\alpha}=\Phi^{-1}(1-\alpha)$，$z_{1-\beta}=\Phi^{-1}(1-\beta)$，其中 Φ 为标准正态分布的密度函数。给定门限值为 $\tau=\mu_0+\sigma_0z_{1-\alpha}=\mu_1+\sigma_1z_{1-\beta}$，当

统计量 $T_k<\tau$ 时判定猜测的密钥为正确候选密钥。能够正常区分的数据量 N 约为

$$N=\frac{2^{128}\left(z_{1-\alpha}+z_{1-\beta}\right)}{\sqrt{(2^{32}-1)/2}-z_{1-\alpha}}$$

当算法的密钥长度为 256 比特时，按照表 3.4.2 的分析可以给出对 NBC 128/256 算法的 16 = 2+11+3 轮零相关线性分析，不妨设弃真概率 $\alpha=2^{-2.7}$，取伪概率 $\beta=2^{-20}$，则 $z_{1-\alpha}\approx1$，$z_{1-\beta}\approx3.9$，需要的明密文对数 $N=\frac{2^{128}\left(z_{1-\alpha}+z_{1-\beta}\right)}{\sqrt{(2^{32}-1)/2}-z_{1-\alpha}}\approx2^{114.8}$。由于错误密钥保留的概率 $\beta=2^{-20}$，故总的计算复杂度约为 $2^{140.59}+2^{256-20}\approx2^{236}$ 次 15 轮加密，存储复杂度为 $2^{114.8}$ 个 128 比特字。

当算法的密钥长度为 128 比特时，按照表 3.4.2 的分析至多只能给出对 NBC 128/128 算法的 15 = 2+11+2 轮零相关线性分析，部分加解密过程需要猜测 4 个子密钥块共 64 比特，计算复杂度约为 $4\cdot2^{112}/(4\cdot15)\approx2^{108.09}$ 次 15 轮加密，存储复杂度为 2^{96} 个 96 比特字。同上，不妨设弃真概率 $\alpha=2^{-2.7}$，取伪概率 $\beta=2^{-20}$，则 $z_{1-\alpha}\approx1$，$z_{1-\beta}\approx3.9$，需要的明密文对数 $N=\frac{2^{128}\left(z_{1-\alpha}+z_{1-\beta}\right)}{\sqrt{(2^{32}-1)/2}-z_{1-\alpha}}\approx2^{114.8}$。由于错误密钥保留的概率 $\beta=2^{-24}$，故总的计算复杂度约为 $2^{108.09}+2^{128-20}\approx2^{109.05}$ 次 15 轮加密。

3.3.2 NBC 256 算法的分析

利用中间相错方法，容易验证 NBC 256 算法存在形如 $(\alpha_0 0\alpha_2 0\alpha_4 0\alpha_6 0\alpha_8 0\alpha_{10} 0\alpha_{12} 0\alpha_{14} 0)\rightarrow_{14}(0\beta_1 0\beta_3 0\beta_5 0\beta_7 0\beta_9 0\beta_{11} 0\beta_{13} 0\beta_{15})$ 的 14 轮零相关线性特征，其中 α_i,β_j 仅一个非零。下面以 14 轮零相关线性特征 (*000000000000000) $\rightarrow_{14}$ (0*000000000000000)为例，分别向前后扩展 3 轮给出本算法的 20 轮零相关线性分析。

类似于 NBC 128 算法的分析，中间状态 $(X_0^3||X_1^{17})$ 与 160 比特的明密文状态 $(X_2^0X_3^0X_{10}^0X_{11}^0X_{14}^0||X_2^{20}X_3^{20}X_6^{20}X_{12}^{20}X_{13}^{20})$ 有关，密钥恢复过程部分加解密时需要猜测 11 个子密钥块，计算复杂度约为 $11\cdot2^{224}/(8\cdot20)\approx2^{220.14}$ 次 20 轮加密，存储复杂度为 2^{208} 个 208 比特字。同上，不妨设弃真概率 $\alpha=2^{-2.7}$，取伪概率 $\beta=2^{-20}$，则 $z_{1-\alpha}\approx1$，$z_{1-\beta}\approx3.9$，需要的明密文对数 $N=\frac{2^{256}\left(z_{1-\alpha}+z_{1-\beta}\right)}{\sqrt{(2^{32}-1)/2}-z_{1-\alpha}}\approx2^{242.8}$。由于错误密钥

保留的概率 $\beta = 2^{-20}$，故总的计算复杂度约为 $2^{220.14}+ 2^{256-20} \approx 2^{236}$ 次 21 轮加密。

3.5 积分分析

3.5.1 NBC 128 算法的分析

利用基于可分性质的积分特征搜索方法，容易验证 NBC 128 算法存在形如 $(A_{15}A_{16}A_{16}A_{16}A_{16}A_{16}A_{16}A_{16}) \rightarrow_{12} (UBUBUBUB)$ 的 12 轮积分特征，即当第 0 块恰有 15 个活动比特时，奇数块都是平衡块。

当算法的密钥长度为 256 比特时，以X_7^{12}为平衡块为例，图 3.5.1 给出了相应的 NBC 128/256 算法 18 轮积分分析的密钥恢复过程。由于 $X_7^{12} = Z_3^{12} \oplus X_6^{13}$，利用中间相遇思想，判断$\oplus X_7^{12} = 0$可以转化为判断$\oplus Z_3^{12} = \oplus X_6^{13}$。图 3.4.1 中红色和橙色部分分别列出了计算$Z_3^{12}$和$X_6^{13}$时需要用到的中间状态和子密钥，绿色部分为计算$Z_3^{12}$和$X_6^{13}$时都用到的中间状态和子密钥。

NBC 128/256 算法 18 轮积分分析的主要步骤如下：

(1) 选择 2^{127} 个明文，它们仅在第一块有一个常数比特，其余均为活动比特；

(2) 猜测相关的 12 个子密钥块，计算$\oplus Z_3^{12}$；

(3) 猜测相关的 5 个子密钥块，计算$\oplus X_6^{13}$；

(4) 若$\oplus Z_3^{12} = \oplus X_6^{13}$，保留相应的子密钥为候选密钥；

(5) 猜测其它密钥，恢复出所有主密钥。

积分分析的计算量主要集中在第 2 步，计算 $Z_3^{12} = \mathrm{S}(X_6^{12} \oplus K_3^{12})$ 需要猜测的子密钥和相应的复杂度如表 3.5.1 所示。从表中可以看出，部分加解密过程需要猜测 11 个子密钥块共 176 比特，计算复杂度约为 $5 \cdot 2^{208}/(4 \cdot 18) \approx 2^{204.15}$ 次 18 轮加密，存储复杂度为 2^{128} 个 128 比特字。利用匹配关系$\oplus Z_3^{12} = \oplus X_6^{13}$，错误密钥保留的概率为 2^{-16}，故总的计算复杂度约为 $2^{204.15} + 2^{256-16} \approx 2^{240}$ 次 18 轮加密。

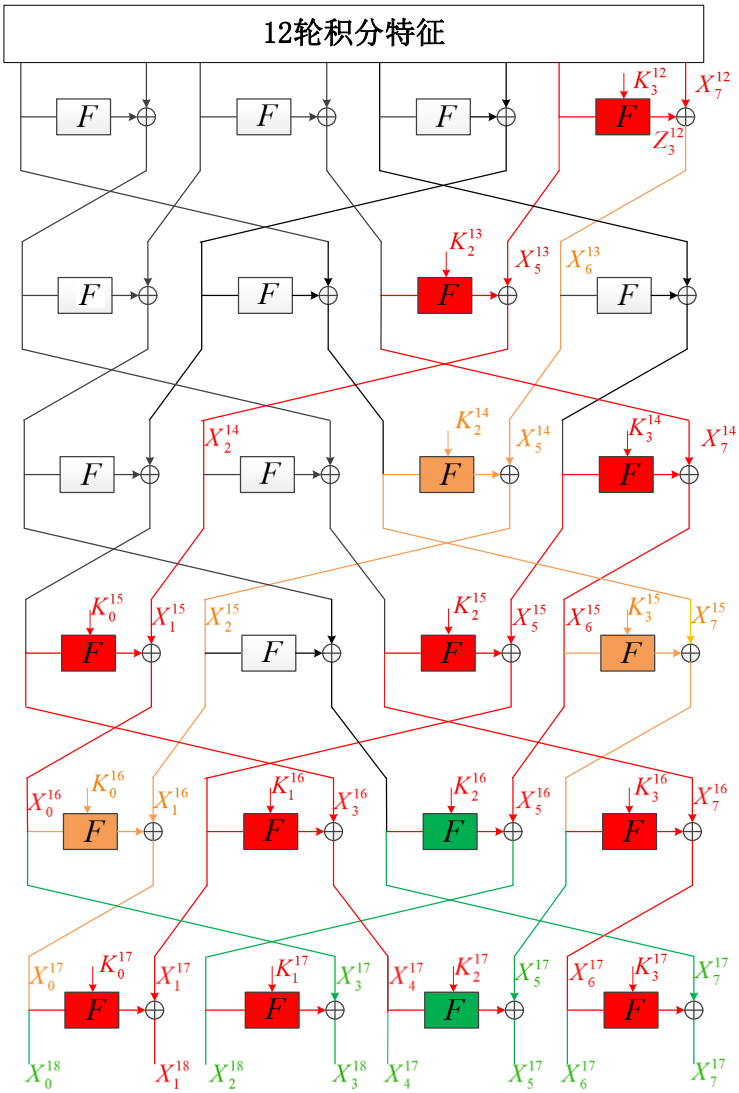

图 3.5.1　NBC 128/256 算法 18 轮积分分析的密钥恢复过程

表 3.5.1　18 轮积分分析计算 $Z_3^{12} = \mathrm{S}(X_5^{13} \oplus K_3^{12})$的复杂度

步骤	猜测子密钥	计算量	状态空间	状态数
0			$x_0 = (\boldsymbol{X_0^{18}} \boldsymbol{X_1^{18}} X_2^{18} X_3^{18} X_4^{18} X_5^{18} X_6^{18} X_7^{18})$	2^{128}
1	K_0^{17}	$2^{128} \cdot 2^{16} = 2^{144}$	$x_1 = (X_1^{17} \boldsymbol{X_2^{18}} \boldsymbol{X_3^{18}} X_4^{18} X_5^{18} X_6^{18} X_7^{18})$	2^{112}
2	K_1^{17}	$2^{112} \cdot 2^{32} = 2^{144}$	$x_2 = (X_1^{17} X_2^{17} X_3^{17} \boldsymbol{X_4^{18}} \boldsymbol{X_5^{18}} X_6^{18} X_7^{18})$	2^{112}
3	K_2^{17}	$2^{112} \cdot 2^{48} = 2^{160}$	$x_3 = (X_1^{17} X_2^{17} X_3^{17} X_4^{17} X_5^{17} \boldsymbol{X_6^{18}} \boldsymbol{X_7^{18}})$	2^{112}
4	K_3^{17}	$2^{112} \cdot 2^{64} = 2^{176}$	$x_4 = (X_1^{17} \boldsymbol{X_2^{17}} X_3^{17} X_4^{17} X_5^{17} X_6^{17} \boldsymbol{X_7^{17}})$	2^{112}
5	K_2^{16}	$2^{112} \cdot 2^{80} = 2^{192}$	$x_5 = (X_0^{16} X_5^{16} X_1^{17} X_4^{17} \boldsymbol{X_5^{17}} \boldsymbol{X_6^{17}})$	2^{96}
6	K_3^{16}	$2^{96} \cdot 2^{96} = 2^{192}$	$x_6 = (X_0^{16} X_5^{16} \boldsymbol{X_1^{17}} \boldsymbol{X_4^{17}} X_7^{16})$	2^{80}
7	K_1^{16}	$2^{80} \cdot 2^{112} = 2^{192}$	$x_7 = (\boldsymbol{X_0^{16}} X_2^{16} \boldsymbol{X_3^{16}} X_5^{16} X_7^{16})$	2^{80}
8	K_0^{15}	$2^{80} \cdot 2^{128} = 2^{208}$	$x_8 = (X_1^{15} \boldsymbol{X_2^{16}} X_5^{16} \boldsymbol{X_7^{16}})$	2^{64}
9	K_2^{15}	$2^{64} \cdot 2^{144} = 2^{208}$	$x_9 = (X_1^{15} \boldsymbol{X_5^{15}} \boldsymbol{X_6^{15}})$	2^{48}

10	K_3^{14}	$2^{48} \cdot 2^{160} = 2^{208}$	$x_{10} = (X_2^{14} X_7^{14})$	2^{32}
11	K_2^{13}	$2^{32} \cdot 2^{176} = 2^{208}$	$x_{11} = X_5^{13}$	2^{16}
12	K_3^{12}	$2^{16} \cdot 2^{192} = 2^{208}$	$x_{12} = Z_3^{12}$	2^{16}

注意到对于选定的 2^{127} 个明文，除X_7^{12}外，X_1^{12}, X_3^{12}, X_5^{12}也都为平衡块，计算它们需要猜测的子密钥数和计算量与 X_7^{12}相当，故可以同时利用这 4 个平衡块排除错误密钥，总的计算量约为 $4 \cdot 2^{204.15} + 2^{256-4 \cdot 16} \approx 2^{206.15}$ 次 18 轮加密。

当算法的密钥长度为 128 比特时，同样以 X_7^{12}为平衡块为例，只能给出 NBC 128/128 算法的 12+4=16 轮积分分析，其中Z_3^{12}的值与 64 比特密文$X_2^{16} X_3^{16} X_5^{16} X_7^{16}$有关，计算$Z_3^{12}$时需要猜测 $K_0^{15} K_2^{15} K_3^{14} K_2^{13} K_3^{12}$等 5 个子密钥，部分解密过程的计算量约为 $5 \cdot 2^{96}/(4 \cdot 16) \approx 2^{92.32}$ 次 16 轮加密，存储复杂度为 2^{64} 个 64 比特字。类似于上面的分析，若同时考虑另外的 2 个平衡块，总的计算复杂度可以降为 $3 \cdot 2^{92.32} + 2^{128-3 \cdot 16} \approx 2^{93.90}$ 次 16 轮加密。

3.4.2 NBC 256 算法的分析

利用基于可分性质的积分特征搜索方法，容易验证 NBC 256 算法存在形如 $(A_{15} A_{16} A_{16} \cdots A_{16}) \rightarrow_{12} (UB \cdots UB)$ 的 16 轮积分特征，即当第 0 块恰有 15 个活动比特时奇数块都是平衡块。

下面以X_{15}^{16}为平衡块为例考虑 NBC 256/256 算法的积分分析。同样利用中间相遇思想，判断 $\oplus X_{15}^{16} = 0$ 可以转化为判断 $\oplus Z_7^{16} = \oplus X_{12}^{17}$ 是否成立，而$Z_7^{16} = \mathrm{S}(X_5^{17} \oplus K_7^{16})$ 与 160 比特密文$X_1^{22} X_2^{22} X_3^{22} X_5^{22} X_6^{22} X_7^{22} X_8^{22} X_9^{22} X_{10}^{22} X_{11}^{22} X_{12}^{22} X_{15}^{22}$有关，计算$Z_7^{16}$时需要猜测 $K_7^{16} K_2^{17} K_5^{18} K_4^{19} K_7^{19} K_2^{20} K_4^{20} K_7^{20} K_0^{21} K_2^{21} K_3^{21} K_5^{21} K_7^{21}$等 13 个子密钥块，部分解密过程的计算量约为 $8 \cdot 2^{224}/(8 \cdot 22) \approx 2^{219.54}$ 次 22 轮加密，存储复杂度为 2^{160} 个 160 比特字。类似于上面的分析，同时考虑另外的 2 个平衡块，总的计算复杂度可以降为 $3 \cdot 2^{219.54} + 2^{256-3.16} \approx 2^{221.13}$ 次 22 轮加密。

3.6 总体评估

综合上面的分析，对于 NBC 128/128 算法，7 轮完全扩散，9 轮迭代后可以抵抗基本差分分析和线性分析，存在 7 轮差分路径和 9 轮差分攻击，存在 7 轮线性路径和 11 轮线性攻击，至多存在 11 轮不可能差分路径和 17 轮不可能差分攻击，存在 11 轮零相关线性路径和 15 轮零相关线性攻击，存在 12 轮积分路径和 16 轮积分攻击，而算法的迭代轮数为 32 轮，有足够的安全冗余。

对于 NBC 128/256 算法，7 轮完全扩散，9 轮迭代后可以抵抗基本差分分析和线性分析，存在 7 轮差分路径和 11 轮差分攻击，存在 7 轮线性路径和 11 轮线性攻击，至多存在 11 轮不可能差分路径和 19 轮不可能差分攻击，存在 11 轮零相关线性路径和 16 轮零相关线性攻击，存在 12 轮积分路径和 18 轮积分攻击，而算法的迭代轮数为 34 轮，有足够的安全冗余。

对于 NBC 256/256 算法，8 轮完全扩散，12 轮迭代后可以抵抗基本差分分析和线性分析，存在 10 轮差分路径和 12 轮差分攻击，存在 10 轮线性路径和 12 轮线性攻击，至多存在 14 轮不可能差分路径和 22 轮不可能差分攻击，存在 14 轮零相关线性路径和 21 轮零相关线性攻击，存在 16 轮积分路径和 22 轮积分攻击，而算法的迭代轮数为 38 轮，也有足够的安全冗余。

4. 性能分析

NBC 算法轮函数结构简单，算法加解密的轮函数和密钥扩展算法用到的基本运算如表 4.1.1 和表 4.1.2 所示。

表 4.1.1 轮函数中涉及的基本运算（以 16 比特为单位）

算法	S 盒变换	密钥加	轮变换加	块置换	迭代轮数
NBC 128/128	4 个 S 盒	4 个 XOR	4 个 XOR	8 个块	32
NBC 128/256	4 个 S 盒	4 个 XOR	4 个 XOR	8 个块	34
NBC 256/256	8 个 S 盒	8 个 XOR	8 个 XOR	16 个块	38

表 4.1.2 密钥扩展算法中涉及的基本运算（以 *n* 比特为单位）

算法	Addition	XOR	Rotation	Size *n*	Round
NBC 128/128	4	12	8	8	4×32
NBC 128/256	4	12	8	16	4×34
NBC 256/256	4	12	8	16	4×38

NBC 算法的 S 盒基于 16 级非线性移位寄存器迭代 20 拍产生，S 盒变换中涉及到的基本运算及硬件成本参见表 4.1.3，S 盒硬件实现成本仅需要约 20.75 标准门电路。

表 4.1.3 S 盒变换中涉及的基本运算及硬件成本

运算类型	AND	NAND	XOR	total
运算数	3	1	8	12
标准门电路数(GE)	4.75	1	16	20.75

NBC 算法的 S 盒、轮函数和密钥扩展算法都采用了非常轻量化的设计，硬件实现时能耗低，非常适合资源受限环境应用。S 盒中使用的非线性运算非常少，也便于进行侧信道防护。

软件实现时，为提高算法加解密效率，S 盒变换可以查表处理，存储 16 比特 S 盒需要 $16 \cdot 2^{16} = 2^{20}$ 比特，即约 128KB 的内存。算法以 16 比特字为基本单元，运算非常简单，在 8/32/64 位软实现平台都可以方便实现。

NBC 算法的详细软硬件实现性能参见自测报告。

5. 优缺点分析

NBC算法的结构非常简单，便于软硬件实现和安全性分析。算法整体采用改进的第二类广义Feistel 结构，轮函数和S盒都采用了轻量化的设计。算法采用了16比特的大S盒，S盒的设计基于非线性移位寄存器迭代构造，可以以非常低的硬件成本构造密码性质优良的大规模非线性S盒，这也是本算法的主要特色和创新之处。

为保证S盒具有好的实现性能和强的安全性，在设计非线性反馈移位寄存器时，我们采用了包含运算量尽量少、扩散速度尽量快的寄存器结构和相关参数。构造S盒只用到了3个与，1个与非和8个异或，硬件实现成本非常低，迭代生成S盒的过程基于bit-slice实现，用到的非线性操作非常少，也便于进行侧信道防护。

NBC 算法的扩散层采用了简单的 16 比特字的置换，为保证各个 16 比特字之间尽快混淆和扩散，我们选用了扩散效果最好的改进型第二类广义 Feistel 结构中的实例。这些结构的安全性已有较详细的分析，我们也进行了验证。结合密钥扩展算法的设计，本算法在抵抗差分分析、线性分析、不可能差分分析、零相关线性分析、积分分析等方面都有足够的安全冗余。

NBC 算法采用多分支广义 Feistel 结构，扩散层为简单置换，与常规 SPN 结构相比，本算法完全扩散速度相对较慢，迭代轮数相对较高，软件实现的加解密速度相对较低，硬件实现的时延相对较高。