doi:10.11959/j.issn.1000-436x.2017214

对轻量级分组密码 I-PRESENT-80 和 I-PRESENT-128 的 biclique 攻击

崔杰，左海风，仲红
（安徽大学计算机科学与技术学院，安徽 合肥 230039）

摘 要：I-PRESENT 是一种适用于 RFID、无线传感节点等资源受限环境的代换——置换型分组密码。利用中间筛选技术来构造 I-PRESENT 的 biclique 结构，首次对全轮 I-PRESENT-80 和 I-PRESENT-128 算法进行了 biclique 攻击。结果表明，biclique 对 I-PRESENT-80 和 I-PRESENT-128 攻击的数据复杂度分别为 2^{26} 和 2^{36} 个选择密文；攻击的时间复杂度分别为 $2^{79.48}$ 和 $2^{127.33}$ 次加密。攻击在时间复杂度和数据复杂度上均优于穷举。利用提出的 I-PRESENT 的密钥相关性技术，攻击的时间复杂度可以进一步降低到 $2^{78.61}$ 和 $2^{126.48}$ 。
关键词：轻量级分组密码；PRESENT；预计算匹配；biclique 攻击
中图分类号：TN918.1 **文献标识码**：A

Biclique cryptanalysis on lightweight block ciphers I-PRESENT-80 and I- PRESENT-128

CUI Jie, ZUO Hai-feng, ZHONG Hong
(College of Computer Science and Technology, Anhui University, Hefei 230039, China)

Abstract: I-PRESENT was a lightweight SPN block cipher for resource-constraint environments such as RFID tags and sensor networks. The biclique structures of I-PRESENT with sieve-in-the-middle technique was an constracted. The biclique cryptanalysis schemes on full-round I-PRESENT-80 and I-PRESENT-128 were proposed for the first time. The results show that the data complexity of the biclique cryptanalysis on I-PRESENT-80 and I-PRESENT-128 is 2^{26} and 2^{36} chosen ciphertexts respectively，and the time complexity on them is $2^{79.48}$ and $2^{127.33}$ encryptions respectively. The time and data complexity are better than that of the exhaustive attack. In addition, the time complexity on them can be reduced to $2^{78.61}$ and $2^{126.48}$ encryptions by using related-key technology of I-PRESENT.
Key words: lightweight block cipher, PRESENT, matching-with-precomputations, biclique cryptanalysis

1 引言

随着射频识别标签(RFID tags)、物联网(Internet of Things)和无线传感节点(wireless sensor node)等低资源设备的发展，使轻量级密码技术逐渐成为一种热门研究领域，在该领域中寻找满足不同低资源设备安全目的的解决方案就显得尤为重要。迄今为止，有许多的轻量级分组密码都满足低资源设备的要求，比如 PRESENT[1]、the KATAN and KTANTAN families[2]、LBLOCK[3]、LED[4]、PRINCE[5]和 the Simon and the Speck families[6]。

I-PRESENT[7]算法是一种代换——置换网络型对合的轻量级分组密码，是对 PRESENT 算法的改进。I-PRESENT 的主要优点是使用了对合函数，使加密电路和解密电路完全相同，以及密码的混淆扩散速度更迅速。这对于需要实现 2 种电路的密码环境来说是一个比较大的优势。

biclique 攻击方法首次是由 Khovratovich 等[8]在

收稿日期：2017-05-18；**修回日期**：2017-08-10
基金项目：国家自然科学基金资助项目(No.61502008, No.61572001)；安徽省自然科学基金资助项目（No.1508085QF132）
Foundation Items: The National Natural Science Foundation of China (No.61502008, No.61572001), The Natural Science Foundation of Anhui Province (No.1508085QF132)

2012 年提出的。相对来说，biclique 攻击是比较新的技术。biclique 结构就是一个完全二部图，即起始状态的每一条边都和结束状态的一条边相连。另外，biclique 结构中的每条路径都是通过唯一的密钥相连。如果路径中没有共享活动的非线性加密单元，敌手通过 biclique 结构就可以高效地测试一系列的候选密钥。最终会取得降低密码攻击开销的效果，或者是可以增加中间相遇攻击以及其他攻击的轮数。biclique 攻击方法对分组密码的密钥恢复有着比较强的适用能力。2011 年，Bogdanov 等[9]对 AES 进行了 biclique 攻击,由于这是第一次针对单密钥模型分组密码的全轮攻击，他们的工作受到了广泛的关注。自该攻击方法结合密码实现之后，基于 biclique 结构的密钥恢复攻击被广泛应用到一系列的分组密码中，包括 3D 密码[10]、SQUARE[11]、HIGHT[12]、Piccolo[13]、ARIA[14]、LBlock[15]、TWINE[16]、IDEA[17]、KLEIN[18]，mCrypton[19]，所有的这些工作，都是第一次对全轮轻量级分组密码的攻击。

2 I-PRESENT 算法

I-PRESENT 算法结构如图 1 所示。

2.1 加密

加密函数可以看作输入 64 位的明文以及 32 个 64 位的轮密钥的集合。轮密钥由主密钥通过密钥扩展算法生成。加密过程简单地说就是一个明文依次经过 15 次轮函数迭代、对合操作以及 15 轮的逆轮函数迭代之后输出密文。

2.2 解密

解密过程与加密过程完全相同，除了解密轮密钥是加密轮密钥的逆序值，即解密轮密钥 $k[0]$等于加密轮密钥 $k[31]$…解密轮密钥 $k[31]$等于加密轮密钥 $k[0]$。加解密算法流程如下。

```
I-PRESENT_Encrypt(state,subkey){
    for(i=0;i<15;i++){
        Mixkey(state,subkey[i]);
        STrans(state);
        PTrans(state);
    }
    Invo(state);
    for(i=15;i<30;i++){
        PTransInv(state);
        STransInv(state);
        Mixkey(state,subkey[i]);
    }
}
```

2.3 轮函数

轮密钥加层(Mixkey)，得到的具体结果是轮密钥的值异或上当前状态的值。

S 盒层(STrans，STransInv)，输入状态分成 16 个 4 位的半字，然后对每个半字使用一个 4×4 的 S 盒操作。S 盒是非线性的函数，每个 4 位的输入对应得到一个 4 位的输出。I-PRESENT 用到的 S 盒的映射如表 1 所示，表 1 中均为十六进制

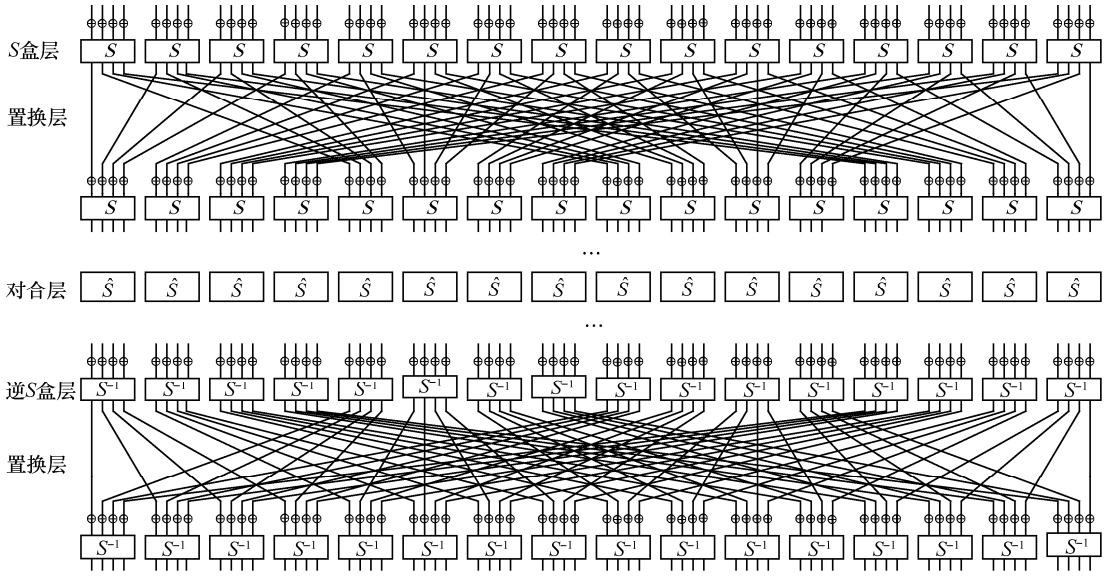


图 1 I-PRESENT 算法结构

值。s 指 S 盒操作用在加密阶段，其逆运算 s^{-1} 用在解密阶段。例如，s 的 4 位半字输入 x=1，则 s 的输出 $s(1)=6$。如果 x=6 是 s^{-1} 的输入，则输出 $s^{-1}(6)=1$。

表 1　I-PRESENT 的 *S* 盒映射关系

x	$s(x)$	$s^{-1}(x)$	$\hat{s}(x)$
0	D	8	E
1	6	2	A
2	1	F	2
3	F	9	C
4	4	4	4
5	8	7	8
6	B	1	F
7	5	E	D
8	0	5	5
9	3	C	9
A	A	A	1
B	C	6	B
C	9	B	3
D	E	0	7
E	7	D	0
F	2	3	6

置换层(PTrans, PTransInv)，PTrans 函数表示的是 64 位输入状态的置换操作。设表达式 $X=x_{63}x_{62}\cdots x_0, Y=y_{63}y_{62}\cdots y_0$ 分别表示 PTrans 函数的输入和输出状态，则 $y_0=x_0$，$y_{16}=x_1$，…，$y_{47}=x_{62}$，$y_{63}=x_{63}$，PTrans 的置换规则如表 2 所示。

2.4　对合(Invo)

对合操作中 64 位的输入状态被划分成了 16 个 4 位的半字，然后对每个半字进行一个 4×4 的 S 盒操作，用 $\hat{s}$ 表示。$\hat{s}$ 映射关系参考表 1。

2.5　密钥扩展

I-PRESENT 支持 2 种长度的密钥，分别为 80 位和 128 位。

2.5.1　80 位密钥

设当前主密钥为 $k_{79}k_{78}\cdots k_0$。第 i 轮密钥 $K^i=\kappa_{63}\kappa_{32}\cdots\kappa_0=k_{79}k_{78}\cdots k_{16}$。轮密钥的生成以及主密钥状态更新过程如下。

1) 当前主密钥循环左移 53 位，表示为 $[k_{79}k_{78}\cdots k_1k_0]=[k_{26}k_{25}\cdots k_{28}k_{27}]$。

表 2　I-PRESENT 置换层位置变换关系

x	y	x	y	x	y	x	y
0	0	16	4	32	8	48	12
1	16	17	20	33	24	49	28
2	32	18	36	34	40	50	44
3	48	19	52	35	56	51	60
4	1	20	5	36	9	52	13
5	17	21	21	37	25	53	29
6	33	22	37	38	41	54	45
7	49	23	53	39	57	55	61
8	2	24	6	40	10	56	14
9	18	25	22	41	26	57	30
10	34	26	38	42	42	58	46
11	50	27	54	43	58	59	62
12	3	28	7	44	11	60	15
13	19	29	23	45	27	61	31
14	35	30	39	46	43	62	47
15	51	31	55	47	59	63	63

2) 当前主密钥中最左侧的 4 位做 $s(x)$ 变换，表示为 $[k_{79}k_{78}k_{77}k_{76}]=s[k_{79}k_{78}k_{77}k_{76}]$，函数 $s(x)$ 的映射关系如表 1 所示。

3) 当前主密钥中的 $k_{19}k_{18}k_{17}k_{16}k_{15}$ 位与轮计数 i (i 扩充为 5 位的二进制值)进行异或，表示为 $[k_{19}k_{18}k_{17}k_{16}k_{15}]=[k_{19}k_{18}k_{17}k_{16}k_{15}]\oplus i$。

4) 从当前主密钥中提取第 i 轮密钥 $K^i=\kappa_{63}\kappa_{32}\cdots\kappa_0=k_{79}k_{78}\cdots k_{16}$，$i$ 值加 1。

轮计数 $i=31$ 后停止状态更新。

2.5.2　128 位密钥

设当前主密钥为 $k_{127}k_{126}\cdots k_0$。第 i 轮轮密钥 $K^i=\kappa_{63}\kappa_{32}\cdots\kappa_0=k_{127}k_{126}\cdots k_{64}$。轮密钥的生成以及主密钥状态更新过程如下。

1) 当前主密钥循环左移 53 位，表示为 $[k_{127}k_{126}\cdots k_1k_0]=[k_{74}k_{73}\cdots k_{76}k_{75}]$。

2)主密钥中最左侧的 8 位做 $s(x)$ 变换，表示为 $[k_{127}k_{126}k_{125}k_{124}]=s[k_{127}k_{126}k_{125}k_{124}]$，$[k_{123}k_{122}k_{121}k_{120}]=s[k_{123}k_{122}k_{121}k_{120}]$，函数 $s(x)$ 的映射关系如表 1 所示。

3)当前主密钥中的 $k_{67}k_{66}k_{65}k_{64}k_{63}$ 位与轮计数 i (i 扩充为 5 位的二进制值)进行异或，表示为 $[k_{67}k_{66}k_{65}k_{64}k_{63}]=[k_{67}k_{66}k_{65}k_{64}k_{63}]\oplus i$。

4) 当前主密钥中提取第 i 轮轮密钥 $K^i=\kappa_{63}\kappa_{32}\cdots\kappa_0=k_{127}k_{126}\cdots k_{64}$，$i$ 值加 1。

轮计数 $i=31$ 后停止状态更新。

3 biclique 攻击

本节给出了 biclique 攻击的简短综述，这个简述是基于 Bogdanov[9]中的描述。

3.1 定义

biclique 是一个完全二部图(如图 2 所示)，即起始状态集 S 中的一个元素通过一条路径与结束状态集中的每个元素相连。用 S_j 表示 S 中的元素，C_i 表示 C 中的元素。从 S_j 到 C_i 一条路径表示子密码 β 在一些密钥 $K[i,j]$ 下的加密操作。

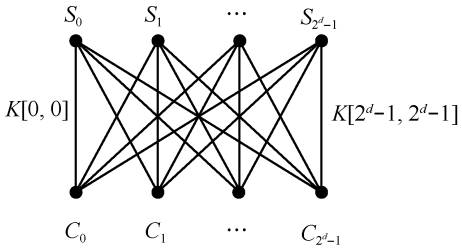


图 2 d 维的 biclique 示意

如果 $\forall i,j\in\{0,\cdots,2^d-1\}$ 有 $S_j \xleftarrow[\beta]{K[i,j]} C_i$，则称三元组 $[\{S_j\},\{C_i\},\{K[i,j]\}]$ 为一个 d 维的 biclique。

如果要是做到 biclique 一般化，就要注意集合 S 和 C 不需要有相同的元素个数。假设 $\forall i\in\{0,\cdots,2^{d_1}-1\},\forall j\in\{0,\cdots,2^{d_2-1}\}$，有 $S_j \xleftarrow[\beta]{K[i,j]} C_i$ 则称三元组 $[\{S_j\},\{C_i\},\{K[i,j]\}]$ 为一个 $d_1\times d_2$ 维的 biclique。

接下来，给出一个简单的 d 维 biclique。假设敌手获得一个分组密码 E，并且要加入基于 biclique 的攻击。首先，敌手要把密钥空间划分为 2^{k-2d} 个子空间，每个子空间包含 2^{2d} 个密钥，其中，k 表示的密钥长度，d 表示所用 biclique 的维数。之后，敌手寻找适当的方法定义一个组合，即函数 $E=\beta\circ E_2\circ E_1$，其中，$E_1$ 的作用是把明文 P 映射为中间状态 v；E_2 的作用是把中间状态 v 映射为另一个中间状态 S；β 表示在子密码中把 S 映射为密文 C。上面描述的具体过程可以表示为 $P\xrightarrow{E_1} v\xrightarrow{E_2} S\xrightarrow{\beta} C$。

敌手可以在密码的任意部分构造 biclique，并且可以利用中间相遇攻击或相似暴力攻击的算法来计算密码剩余部分的复杂度。值得注意的是，敌手必须通过加密或解密预言机获得大量的明密文对。图 3 解释的是 biclique 攻击的结构。

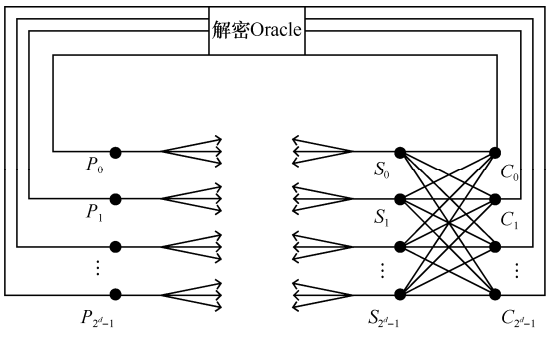


图 3 biclique 攻击的结构

Bogdanov 等[9]在文中给出了 2 种不同的 biclique 攻击范例。一种叫作独立的 biclique (independent biclique)，这种结构可以在低资源消耗的情况下被构造，但是却只能覆盖少量的轮数。另一种是长 biclique (long biclique)，这种结构虽然可以覆盖更多的轮数，但是现实中很难构造实现。所以在结合现实的情况下，本文选择使用第一种方法来构造 biclique 的结构。

3.2 独立 biclique 结构的构造

1) 基础计算：设 f 在密钥 $K[0,0]$ 作用下将 S_0 映射为 C_0： $S_0 \xrightarrow[f]{K[0,0]} C_0$。

2) 利用 2^d 个 Δ_i 差分与 2^d 个 ∇_j 差分构造 2^{2d} 个 (Δ_i,∇_j) 差分。

Δ_i 差分：该差分路径使输入 0 差分在密钥差分 Δ_i^K 的作用下得到输出差分 Δ_i。

$$0\xrightarrow[f]{\Delta_i^K}\Delta_i$$

其中，$\Delta_0^K=0,\Delta_0=0$。

∇_j 差分：该差分路径使输入 ∇_j 差分在密钥差分 Δ_i^K 的作用下得到输出差分 0。

$$\nabla_j\xrightarrow[f]{\Delta_j^K}0$$

其中，$\nabla_0^K=0,\nabla_0=0$。

当 Δ_i 差分与 ∇_j 差分没有共用的活动非线性部件时，用直接异或的方式组合差分 Δ_i 与差分 ∇_j 构成 (Δ_i,∇_j) 差分。

$$\forall i,j\in\{0,1,\cdots,2^d-1\}:\nabla_j\xrightarrow[f]{\Delta_i^K\oplus\nabla_j^K}\Delta_i$$

3) 结合 S_0，C_0，$K[0,0]$ 与 (Δ_i,∇_j) 差分构造 biclique: $S_0\oplus\nabla_j\xrightarrow[f]{K[0,0]\oplus\Delta_i^K\oplus\nabla_j^K}C_0\oplus\Delta_i$ 令 $S_j=S_0\oplus\nabla_j$，$C_i=C_0\oplus\Delta_i$，

$$K[i,j]=K[0,0]\oplus\Delta_i^K\oplus\nabla_j^K$$

若所有的 $i+j>0$，都有 $\Delta_i^K \neq \nabla_j^K$，则所有的 $K[i,j]$ 互不相同，这样，就得到了一个 d 维的 biclique 结构。

3.3 预计算的匹配

Bogdanov 等[9]提出了预计算匹配的技术，这种技术是计算密码中未被 biclique 结构覆盖的轮函数复杂度的高效技术。利用此技术，敌手首先要选择一个中间状态 v，它把除了被 biclique 覆盖部分之外的密码分成子密码 E_1 和 E_2。然后敌手预计算并存储 2^d 个 $\overrightarrow{v_{i,0}}$ 的值，同样的操作可以得到 2^d 个 $\overleftarrow{v_{0,j}}$，表示为 $P_i \to \vec{v}, \overleftarrow{v} \leftarrow S$。其余的操作的表达式为 $P_i \xrightarrow{K[i,j]} \overrightarrow{v_{i,j}}$，$\overleftarrow{v_{i,j}} \xleftarrow{K[i,j]} S_j$。敌手只需重计算轮密钥生成部分以及与存储值不同的轮置换的部分。通过该方法，计算匹配的开销相对于穷举法来说大幅度的降低。另外，如果要进一步降低开销，需要仅匹配中间状态 v 的部分位来实现。

3.4 复杂度计算

对每个 biclique 来说，敌手都要测试 2^{2d} 个密钥。因此，就需要构造 2^{k-2d} 个 bicliques 去覆盖整个密钥空间。在复杂度的计算上，文献[9]给出了计算式为

$$C_{\text{full}} = 2^{k-2d}(C_{\text{biclique}} + C_{\text{decrypt}} + C_{\text{precomp}} + C_{\text{recomp}} + C_{\text{falsepos}})$$

其中，C_{biclique} 表示的是在子密码 β 中计算 2×2^d 个差分轨迹的开销；C_{decrypt} 表示的是解密预言机中解密 2^d 个密文的复杂度；C_{precomp} 表示的是决定 $\overleftarrow{v_{0,j}}$ 和 $\overrightarrow{v_{i,0}}$ 的计算式 $E_2 \circ E_1$ 的开销；C_{recomp} 表示的是重计算 2^{2d} 个 $\overleftarrow{v_{i,j}}$ 和 $\overrightarrow{v_{i,j}}$ 的开销；C_{falsepos} 表示的是消除错误密钥的复杂度。

整个密码攻击的计算复杂度取决于重计算的复杂度。攻击中对内存的要求主要体现在存储 2^d 个中间状态上。需要注意的是，攻击中的内存要求具有较低的数据复杂度，并且可以存储所有提前需要的明密文对。因此，C_{decrypt} 的复杂度相对于整个攻击的时间复杂度来说是可以忽略的。

3.5 I-PRESENT 密钥扩展相关性技术

如果在密钥扩展阶段不计移位的操作，那么提取每一轮的轮密钥的过程中仅改变主密钥中的某些位，由此可以看出轮密钥之间存在相关性。合理利用轮密钥之间的关系可以减少攻击过程中需要猜测的密钥量，从而减少攻击的复杂度。

利用 I-PRESENT 的密钥扩展算法生成的轮密钥其实就是主密钥经过循环移位、S 盒变换和异或操作后，截取左边的 64 位生成的，所以 I-PRESENT 算法中不同轮的轮密钥的某些位是主密钥中相同的位生成。进行攻击时，如果需要猜测这些轮密钥位，则可以将猜测轮密钥等价为猜测主密钥的相关位，从而减少所猜测轮密钥的位数，达到降低计算复杂度的目的。而在猜测轮密钥时，只需确定需要猜测的轮密钥位与主密钥的哪些位有关即可。

命题 1　对于 I-PRESENT-80，若需要猜测轮密钥 K_i^j (表示 i 轮密钥的第 j 个半字)，则只需要猜测 4 位主密钥 $K_{[a,(a+3)]}$ ($[i,j]$ 指第 i 到第 j 位，共 $j-i+1$ 位)所在的位即可，其中，$a = 4j + 19 + 27i \bmod 80$。

命题 2　对于 I-PRESENT-128，若需要猜测轮密钥 K_i^j，则只需要猜测 4 位主密钥 $K_{[a,(a+3)]}$ 所在的位即可，其中，$a = 4j + 67 + 27i \bmod 128$。

符号说明如下。

p:密码的轮数；

n:主密钥长度；

m:轮密钥中受影响的位；

s:总的猜测位。

计算密钥猜测量的核心算法描述如下：

```
int p, n, m, s=0;
    for i=1; i(n/m +1)<=p; i++
    for j=1; j<=n/m &&i(j+1)<=p; j++
        s=s+mj;
        s=s+n;
end for
end for
```

3.6 中间筛选攻击(sieve-in-the-middle)

中间相遇攻击(MITM)由 Differ 等提出，现在已经成为一种重要的分析方法。近些年，中间相遇攻击已经应用到许多密码基元中，包括分组密码、流密码和散列函数等。中间筛选攻击是对中间相遇攻击方法的一种改进，这种技术能够使攻击覆盖更多的密码轮数，并且能够降低 biclique 攻击所需要的数据复杂度。

为了降低重计算的开销，本文攻击中使用了中间筛选的方法。在文献[20]中详细介绍了这种方法。应用此方法，只需要选取输入状态中最左边的 16 位作为子密码函数 $E_s = SL \circ PL \circ AK \circ SL$ 的输入，其中 SL、PL、AK 分别表示 S 盒层、置换层和轮密钥加，相应地得到中间状态位(63,62,61,60,47,46,45,

44,37,36,35,34,15,14,13,12)的输出，如图4所示。输入和输出预存储的值以及相应的密钥可以作为匹配的变量来过滤错误密钥。

4 对全轮I-PRESENT-80算法的biclique攻击

在本节中，I-PRESENT-80的biclique攻击将会被详细的介绍。攻击分为3步：密钥空间的划分、biclique的构造、剩余轮的匹配。

4.1 密钥空间的划分

把80位的密钥空间划分为2^{68}个集合，每个集合包含2^{12}个密钥。其中，每个密钥由提取过轮密钥RK^{29}之后的当前主密钥状态所得。因为轮密钥生成的过程为双射，每一个寄存器状态都唯一地对应于一个密钥，这种密钥划分能够覆盖整个密钥空间。集合中初始密钥$K[0,0]$是固定了12位为0的80位秘密密钥，剩下的68位将迭代取所有可能的值。集合$\{K[i,j]\}$中密钥的定义是和基础密钥$K[0,0]$以及相关的差分Δ_i^K,∇_j^K有关，其中，$i,j\in\{0,\cdots,2^6-1\}$。

密钥所在位$(k_{60},k_{59},k_{58},k_{57},k_{56},k_{55})$、$(k_{24},k_{23},k_{22},k_{21},k_{20},k_{19})$上的差分分别为$i$和$j$，其他位置的差分为0。其中，$K[0,0]$在这些位置上固定取值为0，剩余的位置上任意取值。

4.2 包括3轮的6维biclique结构

biclique仅覆盖密码的最后3轮，按上述密钥划分可知图5中RK^{29}的Δ_i和∇_j没有共享的活动位。分析可知，构造一个低数据复杂度攻击方案是可行的。最后，本文在28~30轮上构造了一个6维的biclique结构。图5形象化地表示出Δ_i和∇_j的差分轨迹，“---”表示的是被差分Δ_i^K影响的位，“---”表示的是被差分∇_j^K影响的位，可看出2种轨迹没有共享活动的非线性组成部件（这里是指S盒）。另外，最终轮上可以看出，Δ_i轨迹仅影响密文C_i的26位，因此，敌手在固定一个密文C_0后，只需收集2^{26}个选择密文即可。

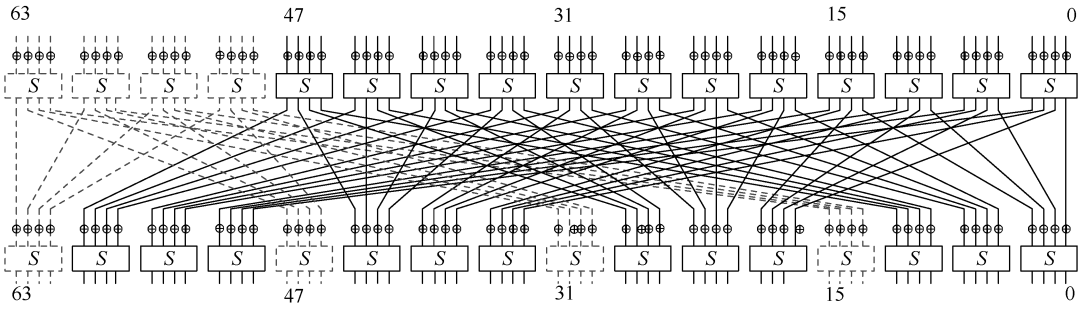


图4 I-PRESENT中的中间筛选操作示意

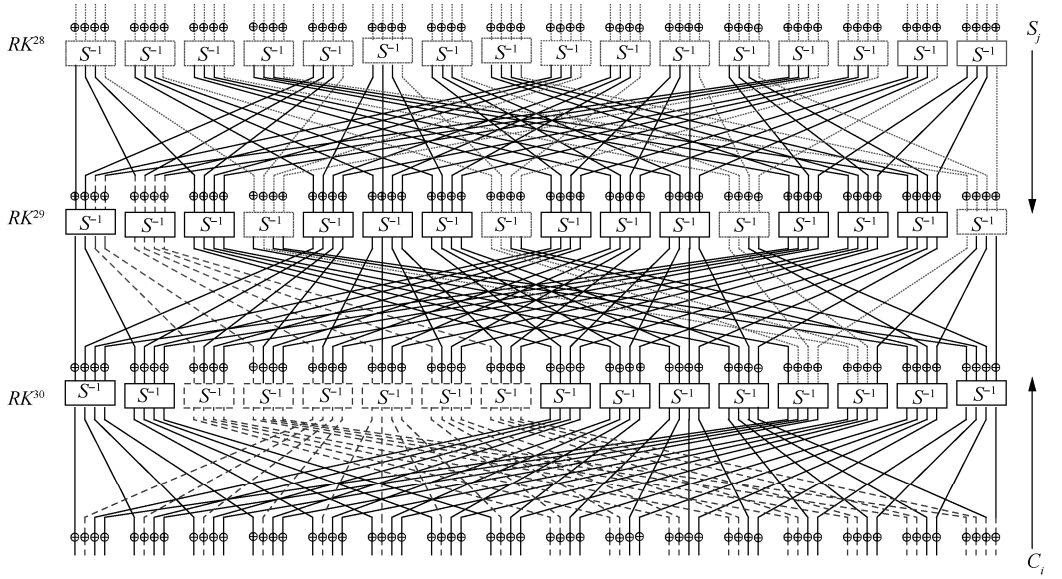


图5 全轮I-PRESENT-80覆盖在28~30轮上的biclique

4.3　27 轮的匹配

本节通过使用预计算匹配算法覆盖密码的 1~27 轮。密码迭代 15 轮之后取状态$\overrightarrow{v_{i,j}}$和 16 轮 S 盒操作之前的状态$\overleftarrow{v_{i,j}}$进行匹配。输入状态中作为中间筛选函数 E_s 的 16 位中的 12 位需要重新构造。考虑到攻击的复杂度，最值得注意的是，那些必须要重计算 S 盒的数目。为了找到一个相对于总开销来说低消耗且单方向的重计算操作，本文把 I-PRESENT 看作一个间接 4 位组操作密码。从而，匹配阶段所需要重计算的数目大约等于轮变换函数与密钥扩展中所需的 S 盒操作数目。这些操作在图 6(a)中以虚线轨迹形象化地表示出来。图 6 中可以看出在第一轮中有 4 个活动的 S 盒，第二轮中有 9 个，3~12 轮共 160 个，13 轮和 14 轮共 24 个，15 轮有 4 个。所以每个前向计算中，共有 201 个激活的 S 盒。图 6(b)中虚线轨迹表示的是后向计算过程，

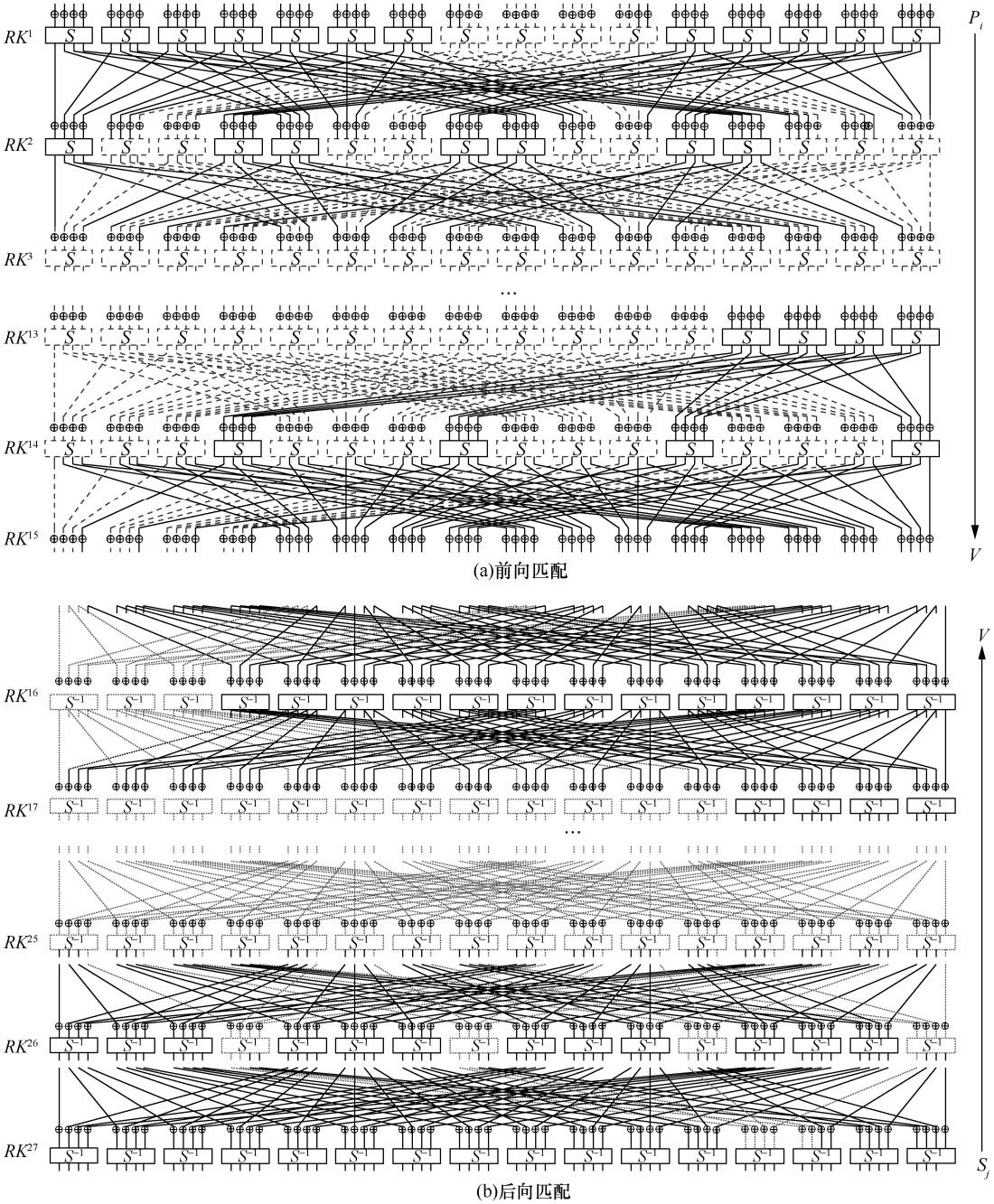


图 6　全轮 I-PRESENT-80 攻击的匹配过程

其中，26轮有4个活动S盒，18~25轮中共128个，17轮中有12个，16轮中有3个。所以每个后向计算中，共有147个激活的S盒。另外，加上在密钥分发阶段的9个活动S盒，重计算的总数为357个S盒。

4.4 复杂度计算

I-PRESENT-80攻击总的计算复杂度可以按照以下计算式计算。

$$C_{\text{full}} = 2^{k-2d}(C_{\text{biclique}} + C_{\text{precomp}} + C_{\text{recomp}} + C_{\text{falsepos}})$$

其中，k=80，d=6；C_{biclique}表示构造单个biclique结构所需要的复杂度，本文攻击中，此复杂度为$2^{3.63}(\approx 2^{6+1}\times(\frac{3}{31}))$次加密；$C_{\text{precomp}}$表示的是匹配检测的预计算复杂度，把它应用到攻击中，得到的复杂度结果为$2^{5.85}(\approx 2^{6}\times(\frac{28}{31}))$次加密；$C_{\text{recomp}}$表示的是重计算$2^{2d}$个$\overleftarrow{v_{i,j}}$，$\overrightarrow{v_{i.j}}$的复杂度，由此可得本文中重计算复杂度为$2^{11.44}(\approx 2^{12}\times\frac{357}{31\times(16+1)})$次加密；$C_{\text{falsepos}}$表示的是消除错误密钥的复杂度，本文中前部分提到，构造12位的匹配变量$\overleftarrow{v_{i,j}}$和$\overrightarrow{v_{i,j}}$，因而$\overrightarrow{v_{i,j}}$输入到中间筛选中的4位状态是未知的，也就是说，至多有$2^{16-12}=2^4$个可能的输出值$\overleftarrow{v_{i,j}}$，所以，匹配中得到12位都未知的$\overleftarrow{v_{i,j}}$的可能性为$2^{4-12}=2^{-8}$，对于一个包含2^{12}个密钥的集合，根据上面的说明可以得出此复杂度为$2^4(=2^{12-8})$次加密。

所以，攻击总的复杂度为

$$C_{\text{full}} = 2^{68}(2^{3.63}+2^{5.85}+2^{11.44}+2^4)\approx 2^{79.48}$$

4.5 密钥相关性分析

在I-PRESENT的密钥扩展阶段中，异或轮计数运算输出的每个位仅仅和输入的相应位有关，所以，没有扩散效果，但是S盒的每一个输出位都和输入位有关，也就是说，具有一定的扩散效果。由于主密钥长度80位和循环左移53位的设定，每4轮已经打乱的4位半字经过循环又重新组成一个新半字。然而做过S盒变换的半字必须经过4轮以及4轮的倍数之后才有可能再做S盒操作，所以扩散发生在最初分组的半字中。

每轮的轮密钥改变主密钥中的9位，其中，前4位是S盒操作引起的，后5位是由于轮常数加引起的。因此，在猜测轮密钥时可以利用I-PRESENT密钥扩展相关性技术（见第3.5节）。比如，第一轮猜测80位的主密钥，那么在第二轮的时候仅需要猜测受影响的9位，继而第三轮的时候猜测9×2=18位，依次进行直到猜测9×8=72位之后，再从80位的主密钥开始猜测。I-PRESENT共有30轮变换，所以总共需要猜测1 319位，相对于每轮80位的猜测来说，效率上提高了将近一倍。利用I-PRESENT密钥扩展相关性技术，攻击的复杂度将降为原来的0.55倍。对于I-PRESENT-80来说，攻击的复杂度将降为$2^{79.48}\times 0.55\approx 2^{78.61}$。

5 对I-PRESENT-128的biclique攻击

5.1 密钥空间的划分

把128位的密钥空间划分为2^{118}个集合，每个集合包含2^{10}个密钥。其中每个密钥由提取过轮密钥RK^{29}之后的密钥寄存器状态所得。因为子密钥生成的过程是双射的过程，每一个寄存器状态都唯一地对应于一个密钥，所以这种密钥空间的划分方案能够覆盖整个密钥空间。集合中初始密钥$K[0,0]$是固定了10位为0的128位秘密密钥，剩下的118位将迭代取所有可能的值。集合$\{K[i,j]\}$中的密钥定义是和基础密钥$K[0,0]$以及差分Δ_i^K,∇_j^K有关的，其中，$i,j\in\{0,\cdots,2^5-1\}$。

密钥所在位$(k_{14},k_{13},k_{12},k_{11}k_{10})$、$(k_{73},k_{74},k_{75},k_{76},k_{77})$的差分分别为$i$和$j$，其他位置的差分为0。其中，$K[0,0]$在这些位置上固定取值为0，剩余的其他位置上任意取值。

5.2 包括4轮的5维biclique结构

biclique仅覆盖密码的最后三轮，按上述密钥划分可知图7中RK^{29}的Δ_i和∇_j没有共享的活动位。分析可知，构造一个数据复杂度低的攻击方案是可行的。最后，本文在27~30轮上构造了一个5维的biclique结构。图7形象化地表示出Δ_i和∇_j的差分轨迹，“---”表示的是被差分Δ_i^K影响的位，“---”表示的是被差分∇_j^K影响的位。从图7可以看出，2种轨迹没有共享活动的非线性部件（这里是指S盒）。另外，最终轮上可以看出，Δ_i轨迹仅影响密文C_i的36位，因此，敌手在固定一个密文C_0后，只需收集2^{36}个选择密文即可。

5.3 26轮的匹配

通过使用预计算匹配程序去覆盖密码的1~26轮。

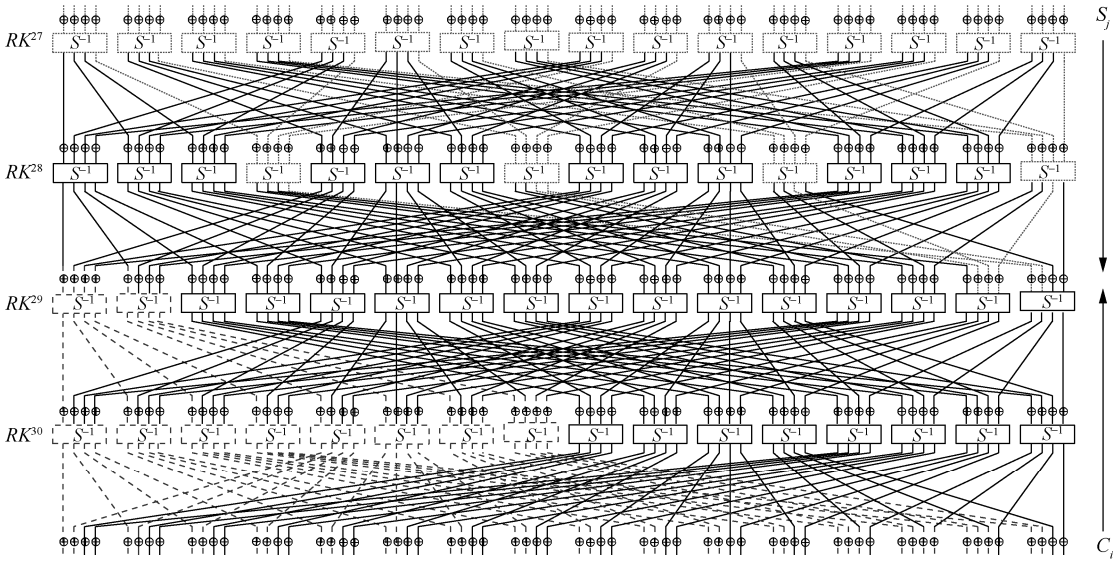


图 7　全轮 I-PRESENT-128 覆盖在 27-30 轮上的 biclique

密码迭代 19 轮之后取状态$\overrightarrow{v_{i,j}}$和 21 轮 S 盒操作之前的状态$\overleftarrow{v_{i,j}}$进行匹配。输入状态中作为中间筛选函数 E_s 的 16 位中的 12 位需要重新构造。考虑到攻击的复杂度，最值得注意的是，那些必须要重计算 S 盒的数目。为了找到一个相对于总开销来说低消耗的单方向的重计算操作，本文把 I-PRESENT 看作一个间接 4 位组操作密码。从而，匹配阶段所需要重计算的数目大约等于轮变换函数与密钥分发中所需的 S 盒操作数目。这些操作在图 8(a)中以虚线轨迹形象化地表示出来，共有 255 个激活的 S 盒。图 8(b)中虚线轨迹表示的是后向计算过程，每个后向计算中，共有 51 个激活的 S 盒。另外，加上在密钥分发阶段的 5 个活动 S 盒，重计算的总数为 311 个 S 盒。

5.4　复杂度计算

I-PRESENT-128 攻击总的计算复杂度可以按照以下计算式计算

$$C_{\text{full}} = 2^{k-2d}(C_{\text{biclique}} + C_{\text{precomp}} + C_{\text{recomp}} + C_{\text{falsepos}})$$

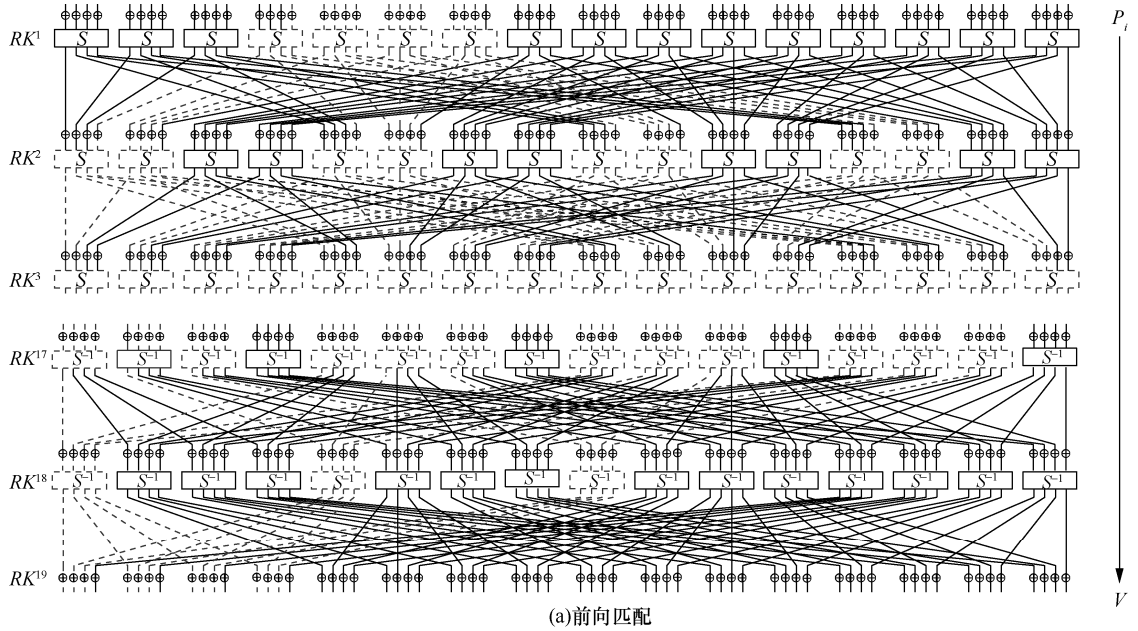
其中，k=128，d=5；C_{biclique} 表示的构造单个 biclique 结构所需要的复杂度，本文攻击中，此复杂度为 $2^{3.05}(\approx 2^{5+1}\times(\frac{4}{31}))$ 次加密；C_{precomp} 表示的是匹配检测的预计算复杂度，把它应用到攻击中，得到的复杂度结果为 $2^{4.79}(\approx 2^{5}\times(\frac{27}{31}))$ 次加密；C_{recomp} 表示的是重计算 2^{2d} 个$\overleftarrow{v_{i,j}}$和$\overrightarrow{v_{i,j}}$的开销，本文为 $2^{9.24}(\approx 2^{10}\times\frac{311}{31\times(16+1)})$ 次加密；C_{falsepos} 表示的是消除错误密钥的复杂度，本文前部分提到，构造 12 位的匹配变量$\overrightarrow{v_{i,j}}$和$\overleftarrow{v_{i,j}}$，因而$\overrightarrow{v_{i,j}}$输入到中间筛选中的 4 位状态是未知的，也就是说，至多有 $2^{16-12}=2^4$ 个可能的输出值$\overleftarrow{v_{i,j}}$，所以，匹配中得到 12 位都未知的$\overleftarrow{v_{i,j}}$的可能性为 $2^{4-12}=2^{-8}$，对于一个包含 2^{10} 个密钥的集合，根据上面的说明可以得出此复杂度为 $2^2(=2^{10-8})$ 次加密。

所以，攻击总的复杂度为

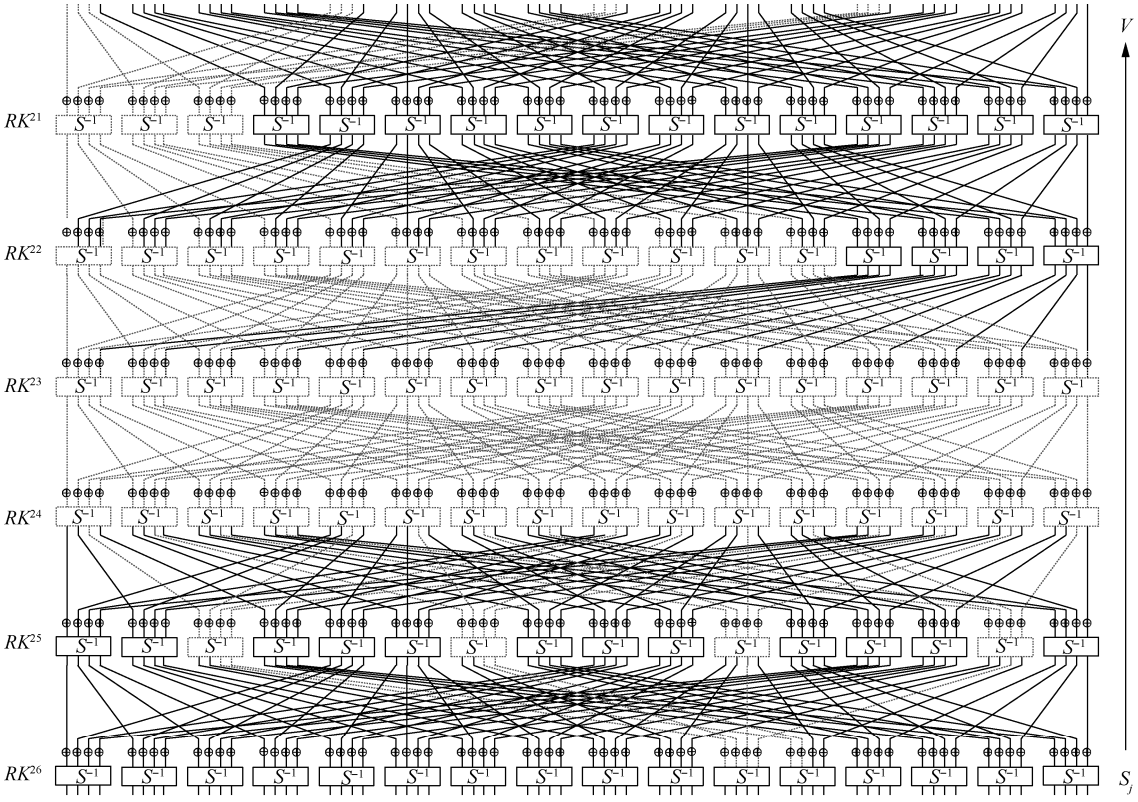
$$C_{\text{full}} = 2^{118}(2^{3.05}+2^{4.79}+2^{9.24}+2^2) \approx 2^{127.33}$$

5.5　密钥相关性分析

在 I-PRESENT 的密钥分发阶段中，异或轮计数运算输出的每个位仅仅和输入的相应位有关，所以没有扩散效果，但是 S 盒的每一个输出位都和输入位有关，也就是说，其具有一定的扩散效果。由于主密钥长度 128 位和循环左移 53 位，所以每轮的轮密钥改变主密钥中的 13 位，其中，前 8 位是 S 盒操作引起的，后 5 位是由于轮常数加引起的。因此，在猜测轮密钥时可以利用 I-PRESENT 密钥扩展相关性技术。比如，第一轮猜测 128 位的主密钥，那么在第二轮的时候仅需要猜测受影响的 13 位，继而第三轮的时候猜测 13×2=26 位，依次进行直到猜测 13×9=117 位之后，再从 128 位的主密钥开始



(a)前向匹配



(b)后向匹配

图 8　全轮 I-PRESENT-128 攻击的匹配过程

猜测。I-PRESENT 共有 30 轮变换，所以总共需要猜测 2 138 位，相对于每轮 128 位的猜测来说，效率上提高了将近一倍。利用 I-PRESENT 密钥扩展相关性技术，攻击的复杂度将降为原来的 0.556 倍。对于 I-PRESENT-128 来说，攻击的复杂度将降为 $2^{127.33}\times 0.556\approx 2^{126.48}$ 。

6　结束语

本文中首次对全轮轻量级分组密码 I-PRESENT-80 和 I-PRESENT-128 进行了 biclique 攻击，攻击的数据复杂度分别为 2^{26} 和 2^{36} 个选择密文；攻击的时间复杂度分别为 $2^{79.48}$ 和 $2^{127.33}$ 次加密。攻击在时间复杂度

和数据复杂度上优于穷举。此外，通过利用 I-PRESENT 轮密钥之间的相关性，可以进一步降低攻击的时间复杂度。

参考文献：

[1] BOGDANOV A, KNUDSEN L R, LEANDER G, et al. PRESENT: an ultra-lightweight block cipher[C]//Internation Workshop on Cryptographic Hardware and Embedded Systems. 2007: 450-466.

[2] DE C, DUNKELMAN O, KNEŽEVIĆ M. KATAN and KTANTAN-a family of small and efficient hardware-oriented block ciphers[M]// Cryptographic Hardware and Embedded Systems-CHES 2009. Springer, Berlin, Heidelberg, 2009: 272-288.

[3] WU W, ZHANG L. LBlock: a lightweight block cipher[C]//Applied Cryptography and Network Security, 9th InternationalConference. 2011: 327-344.

[4] GUO J, PEYRIN T, POSCHMANN A, et al. The LED block cipher[M]//Cryptographic Hardware and Embedded Systems. 2011: 326-341.

[5] BORGHOFF J, CANTEAUT A, GÜNEYSU T, et al. PRINCE-A low-latency block cipher for pervasive computing applications[C]// International Conference on the Theory and Application of Cryptology and Information Security. 2012:208-225.

[6] BEAULIEU R, SHORS D, SMITH J, et al. The simon and speck families of lightweight block ciphers[J]. Cryptology ePrint Archive. 2013.

[7] ZABA M R, JAMIL N, RUSLI M E, et al. I-PRESENT: an involutive lightweight block cipher[J]. Journal of Information Security, 2014, 5(3): 114-122.

[8] KHOVRATOVICH D, RECHBERGER C, SAVELIEVA A. Bicliques for preimages: attacks on skein-512 and the SHA-2 family[C]// International Conference on FAST Software Encryption. 2012: 244-263.

[9] BOGDANOV A, KHOVRATOVICH D, RECHBERGER C. Biclique cryptanalysis of the full AES[C]//The Theory and Application of Cryptology and Information Security. 2011: 344-371.

[10] 陈少真, 刘佳. 对全轮 3D 分组密码算法的 Biclique 攻击[J]. 计算机学报. 2014, 37(5):1063-1070.
CHEN S Z, LIU J. Biclique cryptanalysis on full 3D block cipher[J]. Chinese Journal of Computers, 2014, 37(5):1063-1070.

[11] MALA H. Biclique-based cryptanalysis of the block cipher SQUARE[J]. Iet Information Security, 2014, 8(3): 207-212.

[12] HONG D, KOO B, KWON D. Biclique attack on the Full HIGHT[C]// International Conference on Information Security and Cryptology. 2011:365-374.

[13] WANG Y, WU W, YU X. Biclique cryptanalysis of reduced-round piccolo block cipher[C]//International Conference on Information Security Practice and Experience. 2012: 337-352.

[14] CHEN S Z, XU T M, CHEN S Z, et al. Biclique attack of the full ARIA-256[C]//IACR Cryptology ePrint Archive. 2012.

[15] WANG Y, WU W, YU X, et al. Security on LBlock against biclique cryptanalysis[M]//Information Security Applications. 2012:1-14.

[16] COBAN M, KARAKOC F, BOZTAS Ö. Biclique cryptanalysis of TWINE[C]//CANS. 2012: 43-55.

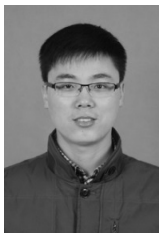
[17] KHOVRATOVICH D, LEURENT G, RECHBERGER C. Narrow-bicliques: cryptanalysis of full IDEA[C]//International Conference on Theory and Applications of Cryptographic Techniques. 2012:392-410.

[18] AHMADIAN Z, SALMASIZADEH M, AREF M R. Biclique cryptanalysis of the full-round KLEIN block cipher[J]. Information Security Iet, 2015, 9(5):294-301.

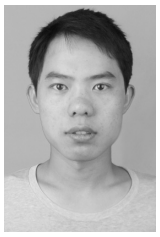
[19] SHAKIBA M, DAKHILALIAN M, MALA H. Non-isomorphic biclique cryptanalysis and its application to full-round mCrypton[J]. IACR Cryptology ePrint Archive, 2013: 141.

[20] CANTEAUT A, NAYA-PLASENCIA M, VAYSSIERE B. Sieve-in-the-middle: improved MITM attacks[M]//Cryptology- CRYPTO 2013. Springer, Berlin, Heidelberg, 2013: 222-240.

作者简介：



崔杰（1980-），男，河南淮阳人，博士，安徽大学副教授、硕士生导师，主要研究方向为网络与信息安全。



左海风（1992-），男，安徽宿州人，安徽大学硕士生，主要研究方向为分组密码的设计与分析。

仲红（1965-），女，安徽固镇人，博士，安徽大学教授、博士生导师，主要研究方向为网络与信息安全。