·计算机工程与应用·

PRESENT密码的差分故障攻击

陈伟建，赵思宇*，邹瑞杰，张晓宁

(电子科技大学通信与信息工程学院 成都 611731)

【摘要】针对PRESENT密码算法的差分故障攻击，分析PRESENT算法差分故障传播特点的方式，优化导入故障位置，利用组合穷举搜索，建立不同的攻击模型来快速获取原始密钥。结果表明，影响PRESENT算法的差分故障攻击结果有两个因素：攻击轮数和故障密文数目。在倒数第二轮攻击平均需要30个故障密文就可以成功恢复出该轮64 bit轮密钥，在低轮数针对该密码算法进行差分故障攻击，仅仅需要9个故障密文就能恢复全部密钥。同时这种攻击方式在单故障密文的密钥搜索复杂度和攻击复杂度分别为2^{26}和2^{31}。

关 键 词 差分故障攻击；故障密文数目；轻量级分组密码；PRESENT算法

中图分类号 TP309 **文献标志码** A doi:10.3969/j.issn.1001-0548.2019.06.010

The Differential Fault Attack of PRESENT Cipher

CHEN Wei-jian, ZHAO Si-yu*, ZOU Rui-jie, and ZHANG Xiao-ning

(School of Communication and Information Engineering, University of Electronic Science and Technology of China Chengdu 611731)

Abstract **Abstract**：Aiming at the differential fault attack of PRESENT cipher algorithm, the differential fault propagation characteristics of PRESENT algorithm are analyzed and the import fault locations are optimized in this paper. On this base, different attack models for quickly obtaining the original cipher are established by using combination exhaustive search. The encryption process and attack process of PRESENT algorithm are implemented by using C++. The results reveal that there are two factors that affect the differential fault attack of PRESENT, the number of attack and the number of fault ciphertext. In the last second round attack, an average of 30 pieces of fault ciphertext are required to restore 64bit round key, while an average of only 9 pieces of fault ciphertexts are necessary to restore all keys in preceding rounds. Meanwhile, the attack complexity of this attack method of single fault cipher is 2^{26}, while the key search complexity is 2^{31}.

Key words differential fault analysis; fault ciphertext number; lightweight block cipher; PRESENT algorithm

差分故障攻击是由文献[1]结合数学研究方法和物理方法提出的一种新的密码分析方式。该方法被应用到多经典分组密码上，如AES[2]、SMS4[3]、FOX[4]等。同时，随着物联网的发展，轻量级分组密码在智能卡、RFID标签和无线传感器网络等资源受限的环境下有着很好的表现，许多密码学学者也对这些密码进行了差分故障攻击分析，如HIGHT[5]、PRINCE[6]、Piccolo[7]等等。

PRESENT密码算法是由文献[8]提出的一种超轻量级密码算法。其分组长度为64 bit，密钥大小分别为80 bit和128 bit，密码结构设计为SPN。随后，为了确保PRESENT密码算法的安全强度，许多学者采用不同的攻击方式对该算法的安全性进行了一系列研究。文献[8]针对16轮的PRESENT算法进行了积分攻击结果分析，攻击复杂度为2^{20}。文献[9]对18轮的PRESENT算法进行了多重差分链攻击，最后计算攻击复杂度为2^{81}。文献[10]采用差分代数对PRESENT算法进行攻击，攻击复杂度为$2^{64.58}$。文献[11]利用多模型差分故障分析方法，分析结果表明在两种不同长度随机错误的条件下，最少需要17个故障密文才能恢复该轮64 bit原始密钥。文献[12]对21轮的PRESENT算法进行Biclique攻击分析，攻击复杂度为$2^{78.9}$。文献[13]采用差分能量对PRESENT算法进行攻击研究，根据分析3 000个波形就能还原原始

收稿日期：2017– 04– 24；修回日期：2019– 06 – 19

作者简介：陈伟建(1956–)，男，教授 ，主要从事无线与移动通信、密码理论与技术方面的研究.

通信作者：赵思宇, E-mail: 568142925@qq.com

密钥。

本文主要对密钥为80 bit的PRESENT算法进行差分故障攻击研究，根据该算法在随机引入单字节故障后，故障传播只会在固定位置出现的特性，利用优化后的差分故障攻击方法对其整个加密过程进行攻击，最后得到实验结果，经过5轮以上加密，平均仅需要引入9个单字节故障密文就能恢复该轮的64 bit轮密钥。

1 PRESENT加密算法

PRESENT算法整个加密过程需要经过31轮加密后，再经过与轮密钥异或的白化操作，输出最后的密文。每轮具体的加密过程如下：

1) 轮密钥异或运算层：每轮加密输出64 bit中间状态与64位的轮密钥进行异或运算。

2) S盒代换层：将异或后64 bit输出在经过16个相同的4-bit输入和4-bit输出S盒。

3) P置换层：通过查询置换表$P(i)$，对S盒的64 bit输出进行重新排列。

在经过31轮加密后，与第32轮轮密钥再进行一次白化异或运算，输出最后的64 bit密文。

PRESENT密码的每轮轮密钥生成算法主要经过循环移位、S盒代换、与当前加密轮数数目异或运算后得到64bit的密钥。本文主要研究密钥为80 bit ($K = k_{79}k_{78}\cdots k_1k_0$)的PRESENT算法，具体密钥生成算法如下：

$$k_{79}k_{78}\cdots k_1k_0 = k_{18}k_{17}\cdots k_{20}k_{19}$$

$$k_{79}k_{78}k_{77}k_{76} = S[k_{79}k_{78}k_{77}k_{76}]$$

$$k_{19}k_{18}k_{17}k_{16}k_{15} = [k_{19}k_{18}k_{17}k_{16}k_{15}] \oplus \text{roundconut}$$

式中，roundconut表示当前加密轮数，用与PRESENT加密过程相同的64 bit S盒代替运算所用的S盒。

2 PRESENT算法差分故障攻击

2.1 PRESENT算法故障传播特性

为了能更好地提高针对PRESENT算法的差分故障攻击效率，首先针对PRESENT算法的故障传播特性进行分析。假设在PRESENT算法加密过程的第r轮随机位引入一个单字节错误，输入为(X_0，X_1，X_2，γ_3，X_4，X_5，X_6，X_7)，经过S盒代换层和P置换层后，它的第r+1轮输入为(γ_0，X_1，γ_2，X_3，γ_4，X_5，γ_6，X_7)，经过分析可以发现PRESENT故障传播有如下特性，如图1所示。

进一步在不同的位置插入故障，经过分析可以发现，在加密过程中任意r轮第0字节、第1字节、第2字节、第3字节引入故障，结果故障只会传播在最后密文C_0,C_2,C_4,C_6位置处，其他位置对应的密文不受影响；采用相同的故障引入方式，在任意r轮的第4字节、第5字节、第6字节、第7字节引入单字节故障，故障只会在最后密文的C_1,C_3,C_5,C_7位置发处，其他位置对应正确的单字节密文。由此可以在该密码算法的奇数或者偶数位置随机引入故障，最后错误密文会出现在相应的位置上，通过差分公式得出错误密文与原始密文，相应这些密码字节位置上的密文不一直为零。通过这种故障传播特性可以进一步对PRESENT算法进行差分故障攻击。

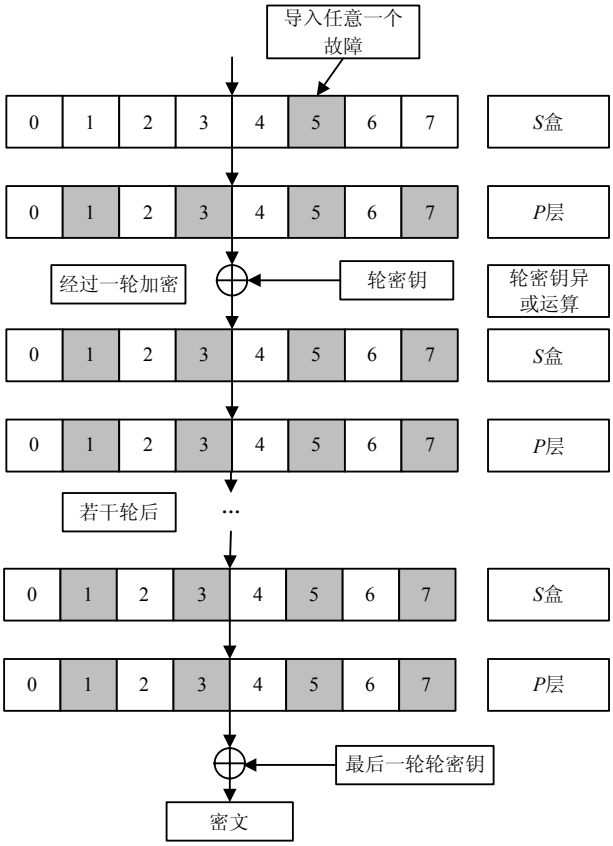

图1 PRESENT单字节差分故障传播图

2.2 差分故障攻击模型和基本假设

针对PRESENT算法采用单字节攻击模型，在随机位置引入随机故障。为了能更好地对PRESENT算法进行分析，在攻击之前提出两点基本假设：

1) 攻击者可以在加密过程中的任意轮对指定的中间状态位置加入故障，但具体的故障值和引入故障位置未知，能够正确记录原始密文和错误密文。

2) 采用相同的明文和轮密钥加密过程中，攻击者能够多次对该密码算法加密相同的轮数和在相同位置引入故障，并记录最后错误密文结果。

2.3　攻击原理

任意选择一组明文和密钥，通过加密算法得到正确密文。然后在加密过程的第30轮处导入故障，结合差分故障攻击特点，获取加密过程中密文的部分特性。进一步通过加密算法的逆过程，推导出该轮64 bit轮密钥。在同样的密文和密钥的条件下，多次重复上述过程，直到恢复的轮密钥为唯一值。在第29轮使用同样步骤获取第二个轮密钥后，通过密钥生成算法，就能推导出PRESENT算法的整个原始密钥。

2.4　攻击步骤

具体的攻击算法分为以下9个步骤：

1) 随机生成PRESENT算法的一个明文P和一个轮密钥K_i，经过31轮加密后得到正确密文C；

2) 用第1)步明文P和原始密钥K_i再经过第30轮加密后，在PRESENT使用的S盒中随机诱导一个单字节的故障，经过最后一轮加密后得到错误密文C_{31}^*，再与第31轮正确密文进行差分运算得到差分错误密文ΔC_{31}^*：

$$\Delta C_{31}^* = C_{31} \oplus C_{31}^*$$

3) 第31轮经过P置换层后ΔC_{31}^*的值用ΔY_{31}表示，经过S盒代换后的值用ΔX_{31}表示，分别为：

$$\Delta Y_{31} = \Delta C_{31}^*$$

$$\Delta X_{31} = \mathrm{PL}^{-1}(\Delta Y_{31})$$

式中，PL^{-1}表示经过P置换层的逆运算。

因此第30轮最后的输出差分密文为：

$$\Delta C_{30}^* = \mathrm{SL}^{-1}(\Delta X_{31})$$

式中，SL^{-1}表示经过S盒的逆运算。

经过差分运算后，非零位只会出现在4个位置上，每个位置可能值为256，所以ΔC_{30}^*的所有值有1 020个，穷举出这1 020个值，并且把它们存入集合M中。由前面的分析得知，在ΔC_{30}^*中所有非零字节只会出现4个位置$(\Delta X_0, 0, \Delta X_2, 0, \Delta X_4, 0, \Delta X_6, 0)$或$(0, \Delta X_1, 0, \Delta X_3, 0, \Delta X_5, 0, \Delta X_7)$，所以其他零位对差分故障攻击没有意义。假设对密文的0、2、4、6位置进行攻击，选择差分输出结果的前两个非零字节密文，根据下面公式计算得出：

$$S^{-1}(x_{0,30}) \oplus S^{-1}(x_{0,30} \oplus \Delta x_{0,30}) = \Delta c_{0,30}$$

$$S^{-1}(x_{2,30}) \oplus S^{-1}(x_{2,30} \oplus \Delta x_{2,30}) = \Delta c_{2,30}$$

穷举$(\Delta x_{0,30}, \Delta x_{2,30})$所有的$2^{16}$可能值，依次计算出$(\Delta c_{0,30}, \Delta c_{2,30})$的所有值。

4) 把第3)步得出的$(\Delta c_{0,30}, \Delta c_{2,30})$和穷举集合$M$中第0、2位字节值进行比较，如匹配结果相同，就将对应的$(\Delta c_{0,30}, \Delta c_{2,30})$存入一个新的集合$T_1$中。

5) 对集合T_1中的每一条记录进行单字节扩展。首先从T_1中取出一个元素α_t，穷举$x_{4,30}$所有的2^8可能值，计算：

$$S^{-1}(\alpha_{0,30}) \oplus S^{-1}(\alpha_{0,30} \oplus \Delta x_{0,30}) = \Delta c_{0,30}$$

$$S^{-1}(\alpha_{2,30}) \oplus S^{-1}(\alpha_{2,30} \oplus \Delta x_{2,30}) = \Delta c_{2,30}$$

$$S^{-1}(x_{4,30}) \oplus S^{-1}(x_{4,30} \oplus \Delta x_{4,30}) = \Delta c_{4,30}$$

然后把$(\Delta c_{0,30}, \Delta c_{2,30}, \Delta c_{4,30})$与集合$M$中的第0、2、4位字节值进行比较。如果匹配结果相同，将$(\Delta c_{0,30}, \Delta c_{2,30}, \Delta c_{4,30})$存入新的集合$T_2$中。

6) 重复步骤5)，扩展出中间状态的最后一个字节$x_{6,30}$，将结果存入集合T_3中，此时T_3中可能有多个4字节的元素。

7) 多次重复步骤1)~6)，在加密过程中随机引入故障密文，直至T_3中只剩下唯一的4字节元素，此元素就代表正确加密中间状态(x_0, x_2, x_4, x_6)。在加密过程中，对密文的后4位进行攻击，按照步骤1)~6)，最后恢复出该算法的后4位中间状态(x_1, x_3, x_5, x_7)。

8) 此后再根据PRESENT的P置换层和轮密钥异或加密过程，推导出最后异或时使用的密钥K_{32}：

$$Y_{31} = \mathrm{PL}(X_{31})$$

$$K_{32} = C_{31} \oplus Y_{31} = C_{31} \oplus \mathrm{PL}(X_{31})$$

9) 同样在第29轮对PRESENT密码算法进行差分故障攻击，根据已经恢复的K_{32}计算出第30轮的密文C_{30}，重复步骤1)~8)，恢复出密钥K_{31}。

3　复杂度分析和实验结果

3.1　攻击复杂度分析

本文根据故障传播特性建立的攻击方式，相比于传统采用的PRESENT密钥直接穷尽搜索有一定的优化，如直接穷尽搜索密钥，需要2^{64}复杂度。本文采用依次扩展的方法，根据每一轮中间状态匹配的数量，来降低下一轮穷尽搜索数，复杂度为：

$$n_1 2^{16} + n_2 2^8 + n_3 2^8 \tag{1}$$

$$n_1 \leqslant n_2 \leqslant 2^8$$

$$n_2 \leqslant n_3 \leqslant 2^8$$

式中，n_1代表穷举前两个字节与可能正确值相匹配数目；n_2、n_3代表秘钥后面两个字节匹配数目。在提供一个故障密文的情况下，通过大量实验获取n_1、n_2、n_3的平均值，可以计算出搜索该轮32 bit密钥的平均计算复杂度为2^{25}，搜索该轮64 bit原始密钥的计算复杂度为2^{26}。

根据PRESENT算法特点，攻击者需要在不同的

位置多次引入不同的错误，再根据匹配表把筛选出的可能正确的中间状态取交集。根据文献[14]可知，恢复出一个轮密钥需要的最少故障数目为：

$$\begin{cases} 0 & d=0 \\ \left\lceil \dfrac{2n}{d} \right\rceil & 1 \leqslant d \leqslant n \end{cases} \tag{2}$$

式中，n表示经过S盒代换层运算后输出比特数目；d表示经过线性层P置换后，扩展最后故障输出的比特数目。本文对PRESENT算法进行单字节攻击，恢复出原始轮密钥需的攻击复杂度为：

$$\begin{cases} 0 & d=0 \\ (n_1 2^{16}+n_2 2^8+\cdots+n_b 2^8)\left\lceil \dfrac{ng}{\mathrm{si}\times d} \right\rceil & 1 \leqslant d \leqslant n \end{cases} \tag{3}$$

式中，n_b表示穷举搜索中间状态与正确状态匹配数目；g表示还原出密钥K_0所需要的理论最少轮密钥数目；si表示S盒输入的比特数目。

在加密运算过程中倒数第2轮的非线性层S盒或者线性P置换层引入任意一个单字节故障，最后产生的差分密文不一直零的只有32 bit，根据式(2)理论上最少需要4个单字节故障密文才能恢复出一个PRESENT算法的轮密钥。恢复出全部64 bit原始密钥至少需要8个单字节故障密文，并且经过连续两轮的随机攻击。根据式(3)可以计算理论攻击复杂度，其中$n_1 n_2 \cdots n_b$根据多次实验得出平均值。那么在一个故障密文的情况下，恢复该轮PRESENT的32 bit密钥平均攻击复杂度为2^{30}，恢复该轮64 bit密钥的平均攻击复杂度为2^{31}。相比于多重差分链攻击、差分代数攻击、Biclique等攻击方式，本文采用的差分故障攻击在很大程度上降低了攻击时所需的复杂度。

3.2 故障密文实验数据

表1 第30轮和第29轮攻击故障样本数据

实验序号	第30轮需要故障密文数目和攻击时间/ms	第29轮需要故障密文数目和攻击时间/ms
1	23(19 832)	25(23 874)
2	24(20 340)	21(19 925)
3	24(20 445)	26(23 722)
4	23(19 534)	25(22 846)
5	25(21 092)	25(22 641)
6	22(18 542)	21(19 381)
7	21(18 262)	22(20 697)
8	21(17 719)	23(21 131)
9	24(20 090)	24(21 543)
10	22(18 477)	23(20 956)

使用Visual Studio 2015工具在普通PC(CPU为Intel(R)Core(TM)i7 4790，3.60 GHz，内存8 GB)上用C++语言实现了PRESENT密码算法的加解密过程和差分故障攻击过程。实验加密明文取值为123456ab，初始密钥值取值为a0b1c2d3e4f5，建立随机攻击位置，随机故障值攻击模型一共完成100组实验。表1为多次实验中的一组数据，给出了能够恢复出唯一轮密钥需要的故障数目和攻击时间。

在PRESENT算法加密过程中的第30轮和29轮导入随机故障，最后恢复该轮密钥大约需要24个故障密文。继续降低估计轮数，所需要的故障密文也逐渐下降，图2给出了在不同轮数攻击，需要的故障数目与攻击成功率之间的关系。

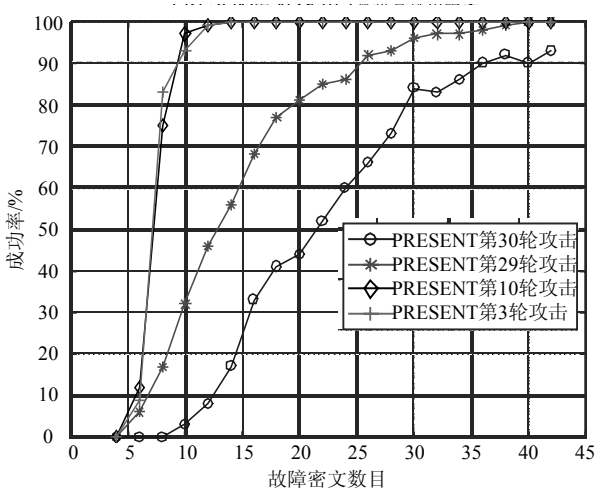

图2 PRESENT不同轮数与故障数目攻击结果

通过图2可以得出，差分故障攻击轮数和导入故障数目对PRESENT最后的攻击结果会产生显著影响。在低轮数导入故障只需要12个故障密文就能100%恢复该轮密钥，最好情况下只需要6个故障密文。经过大量重复试验统计得出在低轮数进行差分故障攻击平均需要9个故障密文，这与式(2)所描述的理论故障数目基本相符。

4 结 束 语

利用优化后差分故障攻击方法，对PRESENT整个加密轮数进行攻击。通过C++语言得到实验结果，最后发现影响PRESENT攻击结果的因素有攻击轮数和故障密文数目。分析表明随着攻击轮数的降低，我们需要的故障密文数目随之下降。在低轮数进行随机位置随机故障值的差分故障攻击，平均只需要9个故障密文就能恢复该轮64 bit原始密钥，该方法的单故障密文密钥搜索复杂度和攻击复杂度也同时降低为2^{26}和2^{31}。

参 考 文 献

[1] BIHAM E, SHAMIR A. Differential fault analysis of Secret key cryptosystems[J]. Lecture Notes in Computer Science, 1997, 1294: 513-525.

[2] GIRAUD C. DFA on AES[M]. [S.l.]: Advanced Encryption Standard-AES, 2005.

[3] 张蕾, 吴文玲. SMS4密码算法的差分故障攻击[J]. 计算机学报, 2006, 9(29): 2596-2602.
ZHANG Lei, WU Wen-ling. Differential fault analysis on SMS4[J]. Chinese Journal of Computers, 2006, 9(29): 2596-2602.

[4] BREVEGLIERI L, KOREN I, MAISTRI P. A fault attack against the fox cipher family[C]//Fault Diagnosis and Tolerance in Cryptography-FDTC. [S.l.]: [s.n.], 2006, 4236: 98-105.

[5] 范伟杰, 吴文玲, 张蕾. HIGHT算法的差分故障攻击[J]. 中国科学院研究生院学报, 2012, 29(2): 271-276.
FAN Wei-jie, WU Wen-ling, ZHANG Lei. Differential fault analysis on HIGHT[J]. Journal of University of Chinese Academy of Sciences, 2012, 29(2): 271-276.

[6] SONG L, HU L. Differential fault attack on the PRINCE block cipher[J]. Lecture Notes in Computer Science, 2013, 8162: 43-54.

[7] 赵光耀, 李瑞林, 孙兵, 等. Piccolo算法的差分故障分析[J]. 计算机学报, 2012, 35(9): 1918-1926.
ZHAO Guang-yao, LI Rui-lin, SUN Bing, et al. Differential fault analysis on Piccol[J]. Chinese Journal of Computers, 2012, 35(9): 1918-1926.

[8] BOGDANOV A, KNUDSEN L R, LEANDER G, et al. PRESENT: An ultralightweight block cipher[J]. Lecture Notes in Computer Science, 2007, 4727: 450-466.

[9] COLLARD B, STANDAERT F X. A statistical saturation attack against the block cipher PRESENT[C]//Proceedings of the Topics in Cryptology. [S.l.]: [s.n.], 2009: 95-210.

[10] WANG M Q, SUN Y, SUN N, et al. Algebraic techniques in differential cryptanalysis revisited[C]//Information Security and Privacy. Melbourne, Australia: [s.n.], 2011: 120-141.

[11] 唐明, 沈菲, 邓慧, 等. PRESENT的多模型差分错误分析[J]. 计算机工程与科学, 2011, 33(10): 39-44.
TANG Ming, SHEN Fei, DENG Hui, et al. A multi-model differential fault analysis on PRESENT[J]. Computer Engineering & Science, 2011, 33(10): 39-44.

[12] 龚征, 刘树生, 温雅敏, 等. 缩减轮数PRESENT算法的Biclique分析[J]. 计算机学报, 2013, 36(6): 1139-1148.
GONG Zheng, LIU Shu-sheng, WEN Ya-min, et al. Biclique analysis on the reduced-round PRESENT[J]. Chinese Journal of Computers, 2013, 36(6): 1139-1148.

[13] DUAN X Y, CUI Q, WANG S X, et al. Differential power analysis attack and efficient countermeasures on PRESENT[C]//2016 8th IEEE International Conference on Communication Software and Networks. Beijing, China: IEEE, 2016: 80-12.

[14] 李玮, 谷大武, 赵辰, 等. 物联网环境下LED轻量级分组密码算法的安全性分析[J]. 计算机学报, 2012, 35(3): 434-444.
LI Wei, GU Da-wu, ZHAO Chen, et al. Security analysis of the LED lightweight cipher in the internet of things[J]. Chinese Journal of Computers, 2012, 35(3): 434-444.

编 辑 叶 芳