PRESENT 轻量级密码的中间相遇统计故障分析

李 玮[1),2),3),4)] 朱晓铭[1)] 谷大武[2)] 李嘉耀[1)] 蔡天培[1)]

[1)](东华大学计算机科学与技术学院 上海 201620)

[2)](上海交通大学计算机科学与工程系 上海 200240)

[3)](上海市可扩展计算与系统重点实验室 上海 200240)

[4)](上海市信息安全综合管理技术研究重点实验室 上海 200240)

摘 要 PRESENT 算法是于 2007 年在国际密码硬件与嵌入式系统会议提出的一种轻量级分组密码,2012 年成为国际轻量级算法标准 ISO/IEC-29192-2,适用于物联网中射频识别标签、网络传感器、智能卡等设备的数据保护.本文结合 PRESENT 密码的设计结构和实现特点,基于统计分析和中间相遇分析策略,提出了一种中间相遇统计故障分析方法,设计了皮尔逊相关系数-汉明重量、库尔贝克莱布勒散度-汉明重量区分器和杰卡德相似系数-汉明重量-极大似然估计等区分器,可以分别破译 PRESENT 密码全部版本的 80 比特和 128 比特原始密钥.该方法攻击轮数更深,故障数和耗时更少,有效地扩展了攻击范围,提升了攻击能力.结果表明,中间相遇统计故障分析对 PRESENT 密码构成了严重威胁.该研究为轻量级密码的实现安全研究提供了有价值的参考.

关键词 轻量级密码;PRESENT;故障分析;中间相遇分析;密码分析

中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2023.00353

Meet-in-the-Middle Statistical Fault Analysis of the PRESENT lightweight Cryptosystem

LI Wei[1),2),3),4)] ZHU Xiao-Ming[1)] GU Da-Wu[2)] LI Jia-Yao[1)] CAI Tian-Pei[1)]

[1)](*School of Computer Science and Technology, Donghua University, Shanghai* 201620)

[2)](*Department of Computer and Science and Engineering, Shanghai Jiao Tong University, Shanghai* 200240)

[3)](*Shanghai Key Laboratory of Scalable Computing and System, Shanghai* 200240)

[4)](*Shanghai Key Laboratory of Integrate Administration Technologies for Information Security, Shanghai* 200240)

Abstract The lightweight block cipher PRESENT was presented at the International Conference on Cryptographic Hardware and Embedded Systems (CHES) in 2007, and it has become a standard of ISO/IEC-29192-2 for lightweight cryptosystems since 2012. Within the environment of the Internet of Things, this cipher is applicable to Radio Frequency Identification, network sensors, and smart cards. Its secret key can be either 80 or 128 bits long, while the block is 128 bits long. Since the publication of the PRESENT, numerous forms of cryptanalysis have been developed in order to evaluate its level of security. Some examples of these attacks include the differential attack, the linear attack, the integral attack, the algebraic attack, the fault attack, the side-cube attack, and the Biclique attack. All of these cryptanalyses are on the basis of the assumption that an attack using a chosen plaintext or an attack using a known plaintext would occur. In order to carry out a chosen-plaintext attack, an attacker must have the ciphertexts along with the plaintexts

收稿日期:2022-01-13;在线发布日期:2022-09-30.本课题得到国家自然科学基金项目(61772129,61932014)、国家密码发展基金目(MMJJ201801001)、上海市自然科学基金(21YF1401200)、上海市可扩展计算与系统重点实验室开放课题、上海市信息安全综合管理技术研究重点实验室开放课题和中央高校基本科研业务费专项资金资助.**李 玮**(通信作者),博士,教授,博士生导师,中国计算机学会(CCF)会员,主要研究领域为对称密码算法的设计与分析.E-mail:liwei.cs.cn@gmail.com.**朱晓铭**,硕士研究生,主要研究方向为分组密码的故障分析.**谷大武**,博士,教授,博士生导师,主要研究领域为密码学与计算机安全.**李嘉耀**,博士研究生,主要研究方向为分组密码的故障分析.**蔡天培**,硕士研究生,主要研究方向为分组密码的安全性分析.

that go along with them, whereas a known-plaintext attack requires a significant quantity of plaintexts and ciphertexts. The ciphertext-only attack is distinct from the chosen-plaintext attack and the known-plaintext attack, which require the ciphertext only in order to be successful. In this sense, the ciphertext-only attack is relevant in a range of scenarios. There is no research that has been released yet regarding PRESENT's resilience to the ciphertext-only attack. This study presents a novel meet-in-the-middle statistical fault analysis using the approach of statistical analysis and meet-in-the-middle attack of PRESENT. This analysis is able to decrypt the 80-bit and 128-bit secret keys of PRESENT with a series of new distinguishers that include the Pearson correlation coefficient-Hamming weight, the Kullback Leibler divergence-Hamming weight, and the Jaccard similarity coefficient-Hamming weight-maximum likelihood estimation, respectively. Furthermore, it examines the meet-in-the-middle statistical fault analysis by using a software simulation of PRESENT. It also evaluates the efficiency of various distinguishers in terms of accuracy, success probability, number of faults, latency, and complexity. When comparing the effectiveness of different distinguishers in recovering the subkey or secret key, the root-mean-square error is the metric that is most commonly utilized. The success probability measures how likely it is that the cryptosystem may be cracked utilizing a particular set of distinguishers that are simply dependent on the ciphertexts. The latency refers to the amount of time required to retrieve the secret key for PRESENT. There is a complexity to the time required, a complexity to the data involved, and a complexity to the memory engaged, all of which add to the overall complexity. The results of the simulation studies show that PRESENT is susceptible to an attack using a meet-in-the-middle statistical fault analysis. It only takes 496, 496, and 480 faults, respectively, for the novel distinguishers of Pearson correlation coefficient-Hamming weight, Kullback Leibler divergence-Hamming weight, and Jaccard similarity coefficient-Hamming weight-maximum likelihood estimate to break PRESENT. The attack is superior to the methods that came before it in terms of accuracy, probability of success, latency, and complexity. The findings provide a useful resource to evaluate and develop cryptographic algorithms in a secure way.

Keywords lightweight cryptosystems; PRESENT; fault analysis; meet-in-the-middle attack; cryptanalysis

1 引 言

近年来,随着射频识别标签、网络传感器、智能卡等设备广泛应用于智能环保、智慧医疗、智能家居和公共安全等领域,物联网正逐渐深入到人们的衣食住行中.然而,针对物联网中日益突出的数据安全问题,传统密码算法由于受到设备计算、存储能力等硬件实现的制约,无法保证其中数据的保密性、完整性和认证性等.轻量级密码算法在资源受限设备中表现优异,其设计与分析已成为密码学研究的主流方向之一[1-4].

PRESENT 算法由 Bogdanov 等学者在 2007 年于 CHES 会议上提出的轻量级分组密码,2012 年成为国际轻量级密码标准 ISO/IEC-29192-2:2019.为了适用于物联网中资源受限的设备,该算法设计目标为具有功耗低、吞吐低和效率高等特点[5-6].自 PRESENT 密码公布以来,差分分析、线性分析、积分分析、代数分析、故障分析、旁路立方体分析、Biclique 分析等诸多密码分析技术对其进行了安全性研究[7-16].

不同于传统的密码分析方法,故障分析在评测密码体制的实现安全性中发挥了至关重要的作用.攻击者通过在密码设备中注入故障,结合错误信息,可以快速完成密钥破译,具有威力强、破译时间短等特点.1997 年,Boneh 等学者通过导入随机故障,首次成功破译 RSA 算法[17].后来,故障分析的种类不断扩展,产生了差分故障分析、不可能差分故障分析、中间相遇故障分析、线性故障分析、无效故障分析、代数故障分析、统计故障分析、统计无效故障分

析和持久故障分析等多种故障分析方法[18-27]. 故障分析已成为评测密码实现安全性的重要指标之一.

在故障分析中，根据攻击者监听加密通讯信息能力的不同，基本假设级别由强到弱可以分为选择密文攻击、选择明文攻击、已知明文攻击和唯密文攻击等. 如果在唯密文攻击下，攻击者能成功破译密码，那么在其它假设下，该密码也无安全性可言. 在所有故障分析方法中，仅有统计故障分析的基本假设为唯密文攻击，此时攻击者能力最弱，仅依靠随机密文，即可以破译密码，适用于物联网环境中受限设备的数据安全分析.

Fuhr 等学者于 2013 年首次提出了针对 AES 密码算法的唯密文故障分析[28]. 攻击者仅需收集受故障影响的随机错误密文，使用统计分析可以恢复出正确密钥. 2016 年，Dobraunig 等学者实现了面向硬件的认证加密算法的统计故障分析[29]. 近年来，LED 和 Piccolo 等密码算法被发现均不能抵御统计故障分析的攻击[30-31]. 然而，统计故障分析方法存在一定的局限性，即攻击轮数范围受到限制，故障注入仅能选择较浅的运算轮数，否则计算复杂度过高，攻击难以实现. 本文基于统计故障分析和中间相遇分析策略，提出了针对 PRESENT 密码的中间相遇统计故障分析方法，并设计实现了皮尔逊相关系数-汉明重量、库尔贝克莱布勒散度-汉明重量和杰卡德相似系数-汉明重量-极大似然估计等新型区分器，实现了在更深轮数中注入故障，并减少了故障数和耗时，达到更好的攻击效果. 该结果为轻量级密码抵御故障分析的安全性提供了重要参考.

2　相关工作

PRESENT 算法自公布以来，其安全性受到国内外学者的深入分析与研究. 2008 年，Wang 利用 14 轮差分路径实现了 16 轮的差分分析[7]. 同年，Z'aba 等学者采用结合位模式的积分分析，实现了 7 轮的积分分析[8]. 2009 年，Nakahara 等学者使用 ElimLin 算法和 PolyBori 框架下的 F4 算法，实现了 5 轮的代数分析[9]. 同年，Yang 等学者通过第 3 轮的输出位泄露信息，使用旁路立方体攻击恢复了主密钥[10]. 2010 年，学者 Cho 采用 8 个单位向量作为线性独立近似值，实现了 25 轮的多维线性分析[11]. 2013 年，Gong 等学者采用两条独立相关密钥差分链，构造出维度为 8 的 Biclique 实现，提出了 21 轮的密钥恢复攻击[12]. 近年来，PRESENT 密码的故障分析研究取得了一些进展. 2010 年，Wang 等学者分别在第 30 轮和第 31 轮子密钥更新过程中注入 3 个半字节和 8 个半字节故障，可以恢复最后一轮子密钥的 53 比特[13]. 2012 年，Zhao 等学者通过 8 次故障注入，通过分析密文索引和差分特征从而将 PRESENT 的密钥搜索空间降低至 $2^{14.7}$，实现了第 29 轮的差分故障攻击[14]. 2014 年，De Santis 等学者提出了统计故障分析，通过在第 30 轮注入故障并分析错误密文，分别恢复了 80 比特和 128 比特主密钥[15]. 2016 年，Huang 等学者通过构建逆向方程，在第 28 轮注入时钟毛刺故障，最少仅需 1 次故障即可实现了代数故障攻击[16]. 针对 PRESENT 的密码分析方法中，仅有统计故障分析和中间相遇统计故障分析的基本假设为唯密文攻击，上述密码分析方法的对比如表 1 所示.

表 1　针对 PRESENT-80/128 算法的安全性分析对比

分析类型	基本假设	攻击轮数	首次故障注入轮	文献
线性分析	已知明文攻击	25/25	—	[11]
代数分析	选择明文攻击	5/5	—	[9]
积分分析	选择明文攻击	7/7	—	[8]
差分分析	选择明文攻击	16/—	—	[7]
Biclique 分析	选择明文攻击	21/21	—	[12]
旁路立方体分析	选择明文攻击	31/—	—	[10]
差分故障分析	选择明文攻击	31/31	第 29/29 轮	[13-14]
代数故障分析	选择明文攻击	31/31	第 28/28 轮	[16]
统计故障分析	唯密文攻击	31/31	第 30/30 轮	[15]
中间相遇统计故障分析	唯密文攻击	31/31	第 29/29 轮	本文

Fuhr 等学者于 2013 年提出了针对 AES 密码的统计故障分析，攻击者通过导入随机故障收集一组错误输出样本，再通过部分解密计算获得注入故障处的单字节中间状态，最后利用不同区分器分析中间状态的分布律，从而恢复出密钥. 在该方法中，攻击者将故障注入在倒数第 2 轮，利用平方欧氏距离、汉明重量和极大似然估计区分器，分别以 320、288 和 224 个故障即可成功破译 128 比特主密钥[28]. 2014 年，De Santis 等学者提出了 PRESENT 轻量级密码的统计故障分析[15]. 2016 年，Dobraunig 等学者在 8 位微控制器、智能卡芯片以及通用微控制器三种不同的硬件平台上，利用时钟故障、激光故障和时钟篡改等方法注入故障，分别需要 30、16、1200 个故障即可成功破译 AES 密码最后一轮子密钥中的 4 个字节，实现了基于硬件的认证加密算法的统计故障分析[29]. 近年来，Li 等学者针对 LED 和 Piccolo 等轻量级密码算法进行了统计故障分析，构建了拟合优度、最大后验估计和拟合优度-平方欧氏距离等

区分器,以更佳的攻击效率恢复出密钥[30-31].因此,在上述唯密文故障攻击方法中,攻击者通过在加密过程中注入故障,导致中间状态产生不均匀的分布并最终影响输出的密文,收集错误密文样本集,并穷举密钥空间解密得到错误中间状态样本集,计算得到每个密钥候选值及其对应的错误中间状态的区分器值,利用区分器筛选出最符合错误中间状态理论分布的样本,其对应的密钥候选值即为正确子密钥.

中间相遇分析是经典的密码分析方法之一,于1977年Diffie和Hellman提出并针对双重DES算法进行了安全性评估[32].攻击者在已知明文和密文的情况下,将加密过程拆分成前向加密和后向解密两部分.穷举所有的前向过程的密钥加密明文并存储结果,再穷举所有的后向过程解密密文,将解密的结果与存储的正向加密的结果相匹配,若一致认为对应的两组密钥为正确密钥.中间相遇分析通过预计算并存储部分过程的穷举结果,可以大幅降低分析方法的复杂度.

本文提出了中间相遇统计故障分析方法,并评估了平方欧氏距离、汉明重量、极大似然估计、拟合优度、最大后验概率、拟合优度-平方欧氏距离等现有区分器以及皮尔逊相关系数-汉明重量、库尔似系数-汉明重量-极大似然估计等新型区分器的攻击效果.目前,针对AES、LED等代换置换网络(SPN)密码的传统唯密文故障分析仅能支持倒数第二轮的故障注入,本文利用中间相遇策略设计新型统计故障分析方法,可以深入到密码倒数第三轮(即PRESENT算法的第29轮)中进行,在此基础上构建皮尔逊相关系数-汉明重量、库尔贝克莱布勒散度-汉明重量和杰卡德相似系数-汉明重量-极大似然估计区分器等新型区分器,不仅继续保持故障数、耗时等方面的优势,而且可以提升故障攻击的准确度.表2分别给出了PRESENT密码的80比特和128比特主密钥的破译结果.该方法为检测SPN型密码抵御统计故障攻击的安全实现提供有价值的参考.

表2 统计故障分析和中间相遇统计故障分析破译PRESENT-80/128算法主密钥的结果

区分器	英文简称	统计故障分析[15]			中间相遇统计故障分析		
		首次故障注入轮	故障数	耗时/min	首次故障注入轮	故障数	耗时/min
汉明重量	HW	第30/30轮	480/540	6.41/7.22	第29/29轮	448/512	0.69/0.79
极大似然估计	MLE	第30/30轮	488/549	8.00/9.00	第29/29轮	462/528	1.62/1.85
平方欧氏距离	SEI	第30/30轮	688/774	12.52/14.09	第29/29轮	623/712	0.84/0.96
拟合优度	GF	第30/30轮	872/981	15.83/17.81	第29/29轮	749/856	1.09/1.25
最大后验估计	MAP	第30/30轮	696/783	7.96/8.96	第29/29轮	490/560	1.62/1.85
拟合优度-平方欧式距离	GF-SEI	第30/30轮	792/891	10.11/11.37	第29/29轮	735/840	1.52/1.73
皮尔逊相关系数-汉明重量	PCC-HW	—	—	—	第29/29轮	434/496	0.68/0.77
库尔贝克莱布勒散度-汉明重量	KLD-HW	—	—	—	第29/29轮	434/496	0.65/0.75
杰卡德相似系数-汉明重量-极大似然估计	JSC-HW-MLE	—	—	—	第29/29轮	420/480	0.70/0.80

3 PRESENT算法

3.1 符号说明

设Z_2^e为e比特的二进制向量集;

记$X\in(Z_2^4)^{16}$为明文,$Y=y^{63}y^{62}\cdots y^1y^0\in(Z_2^4)^{16}$为密文;

记$MK\in(Z_2^4)^v$为$4v$比特主密钥,其中$v\in\{20,32\}$;记$RK_i=(rk_i^{63}rk_i^{62}\cdots rk_i^1rk_i^0)\in(Z_2^4)^{16}$为第$i$轮子密钥,$K_i=(k_i^{4v-1}\cdots k_i^1k_i^0)$为第$i$轮密钥寄存器的值,其中$K_1=MK$,$i\in[1,32]$;

记ARK为子密钥加,SL、SL^{-1}为S盒及S盒的逆,PL、PL^{-1}为P置换及P置换的逆;

记$A_i=(a_i^{63}a_i^{62}\cdots a_i^1a_i^0)\in(Z_2^4)^{16}$、$B_i=(b_i^{63}b_i^{62}\cdots b_i^1b_i^0)\in(Z_2^4)^{16}$和$C_i=(c_i^{63}c_i^{62}\cdots c_i^1c_i^0)\in(Z_2^4)^{16}$为第$i$轮子密钥加、S盒和P置换后的状态,其中$i\in[1,32]$;

记$\oplus$、$\&$、$\|$分别表示按位异或、与、级联,$\gg$、$\ll$和$\ggg$分别表示比特串的右移、左移和循环右移;$\sum$、$\prod$分别表示连加、连乘;

记$\sim$为元素的故障值,$(\cdot)^j$为比特串的第j个比特.

3.2 PRESENT密码

PRESENT密码是一种具有代换置换网络结构的典型轻量级分组密码,分为PRESENT-80和PRESENT-128两个版本,其中两个版本的轮函数,分组长度和迭代轮数均相同,仅主密钥长度和密钥编排方案存在不同,具体参数如表3所示.

表 3　PRESENT 密码各版本参数

版本	分组长度/bit	密钥长度/bit	迭代轮数
PRESENT-80	64	80	31
PRESENT-128	64	128	31

PRESENT 算法由加密、解密和密钥编排方案三部分组成，其中解密是加密的逆运算，子密钥的使用顺序与加密相反. 密码结构如图 1 所示. 轮函数包括子密钥加、S 盒和 P 置换，明文经过 31 轮迭代，白化后生成密文. 解密部分的子密钥的使用顺序与加密相反. 算法结构和两个版本的密钥编排方案如算法 1～算法 3 所示.

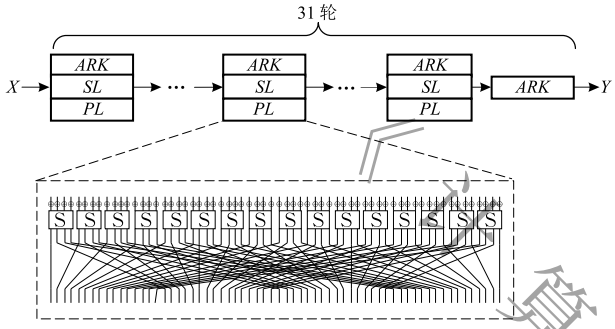

图 1　PRESENT 算法结构

算法 1.　PRESENT 的加密算法.

输入：明文 X，子密钥 $RK_1, RK_2, \cdots, RK_{31}, RK_{32}$

输出：密文 Y

1. $T=X$;
2. FOR $i=1$ TO 31 DO
3. 　$T=PL(SL(T \oplus RK_i))$;
4. END FOR
5. $Y=T \oplus RK_{32}$;
6. RETURN Y.

算法 2.　PRESENT-80 版本的密钥编排方案.

输入：主密钥 MK

输出：子密钥 $RK_1, RK_2, \cdots, RK_{31}, RK_{32}$

1. $K_1=MK$;
2. $RK_1=K_1 \gg 16$;
3. FOR $i=2$ TO 32 DO
4. 　$K_i=K_{i-1} \lll 61$;
5. 　$(k_i^{79}k_i^{78}k_i^{77}k_i^{76})=SL(k_i^{79}k_i^{78}k_i^{77}k_i^{76})$;
6. 　FOR $j=15$ TO 19 DO
7. 　　$k_i^j=k_i^j \oplus (i \& (0\text{x}1 \ll (j-15)))$;
8. END FOR
9. $RK_i=K_i \gg 16$;
10. END FOR
11. RETURN $RK_1, RK_2, \cdots, RK_{31}, RK_{32}$.

算法 3.　PRESENT-128 版本的密钥编排方案.

输入：主密钥 MK

输出：子密钥 $RK_1, RK_2, \cdots, RK_{31}, RK_{32}$

1. $K_1=MK$;
2. $RK_1=K_1 \gg 64$;
3. FOR $i=2$ TO 32 DO
4. 　$K_i=K_{i-1} \lll 61$;
5. 　$(k_i^{127}k_i^{126}k_i^{125}k_i^{124})=SL(k_i^{127}k_i^{126}k_i^{125}k_i^{124})$;
6. 　$(k_i^{123}k_i^{122}k_i^{121}k_i^{120})=SL(k_i^{123}k_i^{122}k_i^{121}k_i^{120})$;
7. 　FOR $j=62$ TO 66 DO
8. 　　$k_i^j=k_i^j \oplus (i \& (0\text{x}1 \ll (j-62)))$;
9. 　END FOR
10. 　$RK_i=K_i \gg 64$;
11. END FOR
12. RETURN $RK_1, RK_2, \cdots, RK_{31}, RK_{32}$.

4　中间相遇统计故障分析

4.1　基本假设和故障模型

本文采用的基本假设是唯密文攻击. 攻击者使用相同主密钥加密明文，通过“与”操作，采用半字节随机故障模型实现故障注入，影响中间状态值产生非均匀分布，获取随机错误密文. 基于“与”运算，比特“0”和“1”出现的概率分别为 0.75 和 0.25，以 8 为例，其分布概率为

$$(0.25) \cdot (0.75)^3 \approx 0.1055,$$

依次可以推导出其它半字节值的分布概率. 若在均匀状态分布中，“0”和“1”出现的概率为 0.5 和 0.5，每个半字节值的分布概率均为

$$(0.5)^4 = 0.0625,$$

图 2 给出了半字节在非均匀分布和均匀分布状态下的累加分布规律.

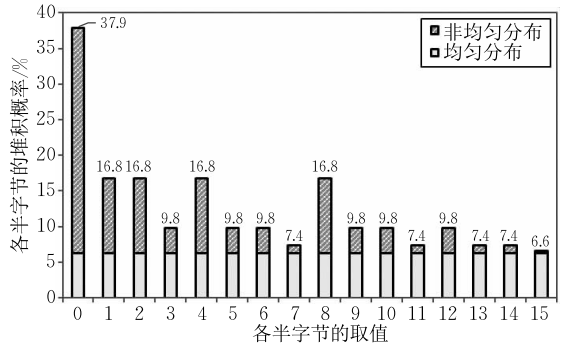

图 2　半字节的累加分布律对比

4.2　中间相遇分析策略

1977 年，学者 Diffie 和 Hellman 首次提出中间

相遇攻击,并应用于分析分组密码的安全性[32].它的特点在于能够综合利用加密前向与解密后向的信息获取信息,利用空间换取时间,从而降低时间复杂度.不可能差分分析和 Biclique 分析等典型密码分析方法的思想也来源于中间相遇攻击[33-34].

在传统统计故障分析的方法中,穷举子密钥候选值并推导中间状态样本值的时间较长.如果故障扩散比特数多,那么穷尽搜索子密钥候选值的时间复杂度和攻击耗时将大幅增长.鉴于此,若将中间相遇分析策略应用于统计故障攻击中,可以有效地解决上述问题.通过确定中间相遇分析过程,划分前向过程和后向过程.图 3 给出了故障注入在首个半字节中的故障扩散路径,前向过程包括受故障影响的 C_{29} 加密至 A_{31},后向过程为错误密文解密至 A_{31},最后对两个过程结果进行匹配.在 4.3 节中,攻击者可以在攻击开始前,提前预计算并存储步骤 2 的映射结果,并在步骤 3 计算时直接读取,可以有效降低穷举子密钥候选值的复杂度和耗时.

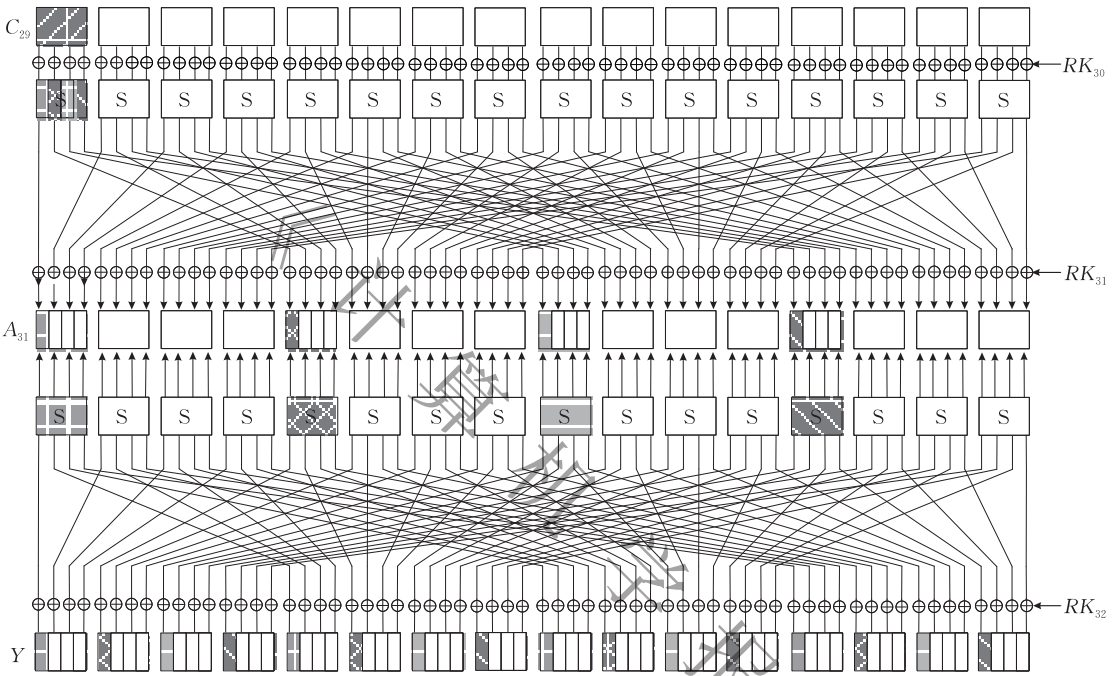

图 3 注入故障后的中间相遇过程

4.3 攻击步骤

本节提出的中间相遇统计故障分析包括以下 4 个步骤:

步骤 1. 故障注入.攻击者在加密运算过程中,将随机半字节故障导入在第 29 轮(倒数第 3 轮)中,生成错误密文.攻击者利用错误密文 Y 的已知 16 比特和子密钥 RK_{32} 的 16 比特,可以推导得到 A_{31} 中所有 4 比特.图 3 给出了故障注入在首个半字节中的故障扩散路径.以推导 A_{31} 中的 4 比特为 $(\tilde{a}_{31}^{0}\tilde{a}_{31}^{16}\tilde{a}_{31}^{32}\tilde{a}_{31}^{48})$ 例,即:

$$(\tilde{a}_{31}^{0}\tilde{a}_{31}^{16}\tilde{a}_{31}^{32}\tilde{a}_{31}^{48})=$$
$$(SL^{-1}((rk_{32}^{0}rk_{32}^{4}rk_{32}^{8}rk_{32}^{12})\oplus(\tilde{y}^{0}\tilde{y}^{4}\tilde{y}^{8}\tilde{y}^{12})))^{0}\ \|$$
$$(SL^{-1}((rk_{32}^{16}rk_{32}^{20}rk_{32}^{24}rk_{32}^{28})\oplus(\tilde{y}^{16}\tilde{y}^{20}\tilde{y}^{24}\tilde{y}^{28})))^{0}\ \|$$
$$(SL^{-1}((rk_{32}^{32}rk_{32}^{36}rk_{32}^{40}rk_{32}^{44})\oplus(\tilde{y}^{32}\tilde{y}^{36}\tilde{y}^{40}\tilde{y}^{44})))^{0}\ \|$$
$$(SL^{-1}((rk_{32}^{48}rk_{32}^{52}rk_{32}^{56}rk_{32}^{60})\oplus(\tilde{y}^{48}\tilde{y}^{52}\tilde{y}^{56}\tilde{y}^{60})))^{0}.$$

步骤 2. 中间相遇分析.如图 3 所示,攻击者依次推导 $(\tilde{c}_{29}^{0}\tilde{c}_{29}^{1}\tilde{c}_{29}^{2}\tilde{c}_{29}^{3})$ 至 $(\tilde{a}_{31}^{0}\tilde{a}_{31}^{16}\tilde{a}_{31}^{32}\tilde{a}_{31}^{48})$ 的过程,分别穷举 C_{29} 的 4 比特、RK_{30} 的 4 比特和 RK_{31} 的 4 比特,可以计算其对应的 A_{31} 的 4 比特 $(\tilde{a}_{31}^{0}\tilde{a}_{31}^{16}\tilde{a}_{31}^{32}\tilde{a}_{31}^{48})$,并存储 2^{12} 组映射结果如下:

$$(\tilde{a}_{31}^{0}\tilde{a}_{31}^{16}\tilde{a}_{31}^{32}\tilde{a}_{31}^{48})=(rk_{31}^{0}rk_{31}^{16}rk_{31}^{32}rk_{31}^{48})\oplus$$
$$(SL(\tilde{c}_{29}^{0}\tilde{c}_{29}^{1}\tilde{c}_{29}^{2}\tilde{c}_{29}^{3})\oplus(rk_{30}^{0}rk_{30}^{1}rk_{30}^{2}rk_{30}^{3}),$$

攻击者选取 $(\tilde{c}_{29}^{0}\tilde{c}_{29}^{1}\tilde{c}_{29}^{2}\tilde{c}_{29}^{3})$ 作为中间状态样本值,将步骤 1 计算得出的 A_{31} 的 4 比特 $(\tilde{a}_{31}^{0}\tilde{a}_{31}^{16}\tilde{a}_{31}^{32}\tilde{a}_{31}^{48})$ 与步骤 2 中计算存储的 $(\tilde{a}_{31}^{0}\tilde{a}_{31}^{16}\tilde{a}_{31}^{32}\tilde{a}_{31}^{48})$ 进行匹配,若匹配一致,则选取 C_{29} 的 4 比特 $(\tilde{c}_{29}^{0}\tilde{c}_{29}^{1}\tilde{c}_{29}^{2}\tilde{c}_{29}^{3})$ 作为中间状态样本值,同时选取步骤 1 中的 RK_{32} 的 16 比特、步骤 2 中的 RK_{31} 的 4 比特和 RK_{30} 的 4 比特作为候选密钥比特串.

步骤 3. 区分器选取.攻击者按照步骤 1、2 中获取一系列候选密钥值及其对应的半字节样本组,使用 4.4 节中的区分器对半字节中间状态样本组进

行统计分析，并计算区分器值，可以获得正确子密钥的 24 比特，包括 RK_{32} 的 16 比特、RK_{31} 的 4 比特和 RK_{30} 的 4 比特. 重复步骤 1 到 3，直到获取子密钥全部比特.

步骤 4. 主密钥恢复. 在 PRESENT-80 版本中，攻击者仅需 RK_{32} 的全部比特和 RK_{31} 中的第 3 至 18 比特，即可得到第 32 轮子密钥寄存器值 K_{32}，即：

$$\begin{aligned}K_{32}&=RK_{32}\|(rk_{31}^{18}rk_{31}^{17}\cdots rk_{31}^{4}rk_{31}^{3})\\&=(k_{32}^{79}k_{32}^{78}\cdots k_{32}^{2}k_{32}^{1}k_{32}^{0}),\end{aligned}$$

再依据算法 2 中密钥编排方案：

$$\begin{aligned}&(k_i^{19}k_i^{18}k_i^{17}k_i^{16}k_i^{15})=(k_i^{19}k_i^{18}k_i^{17}k_i^{16}k_i^{15})\oplus i,\\&(k_i^{79}k_i^{78}k_i^{77}k_i^{76})=SL^{-1}(k_i^{79}k_i^{78}k_i^{77}k_i^{76}),\\&(k_{i-1}^{79}k_{i-1}^{78}\cdots k_{i-1}^{1}k_{i-1}^{0})=(k_i^{60}k_i^{59}\cdots k_i^{62}k_i^{61}),\end{aligned}$$

其中，$i\in[2,32]$. 主密钥 MK 表示为

$$MK=K_1=(k_1^{79}k_1^{78}\cdots k_1^{2}k_1^{1}k_1^{0}).$$

在 PRESENT-128 版本中，攻击者需要获取 RK_{32} 的全部比特、RK_{30} 中的第 3 至 5 比特和 RK_{31} 中的第 3 至 63 比特，可以推导第 32 轮子密钥寄存器 K_{32}，即：

$$\begin{aligned}K_{32}&=RK_{32}\|(rk_{30}^{5}rk_{30}^{4}rk_{30}^{3})\|(rk_{31}^{63}rk_{31}^{62}\cdots rk_{31}^{1}rk_{31}^{0})\\&=(k_{32}^{127}k_{32}^{126}\cdots k_{32}^{2}k_{32}^{1}k_{32}^{0}),\end{aligned}$$

再依据算法 3 中密钥编排方案：

$$\begin{aligned}&(k_i^{66}k_i^{65}k_i^{64}k_i^{63}k_i^{62})=(k_i^{66}k_i^{65}k_i^{64}k_i^{63}k_i^{62})\oplus i,\\&(k_i^{123}k_i^{122}k_i^{121}k_i^{120})=SL^{-1}(k_i^{123}k_i^{122}k_i^{121}k_i^{120}),\\&(k_i^{127}k_i^{126}k_i^{125}k_i^{124})=SL^{-1}(k_i^{127}k_i^{126}k_i^{125}k_i^{124}),\\&(k_{i-1}^{127}k_{i-1}^{126}\cdots k_{i-1}^{1}k_{i-1}^{0})=(k_i^{60}k_i^{59}\cdots k_i^{62}k_i^{61}),\end{aligned}$$

其中，$i\in[2,32]$. 主密钥 MK 表示为

$$MK=K_1=k_1^{127}k_1^{126}\cdots k_1^{2}k_1^{1}k_1^{0}.$$

4.4　区分器

本节采用以下 9 个区分器对 PRESENT 算法进行分析. 错误的密钥候选值对应的中间状态更接近于均匀分布，而正确的密钥候选值对应的中间状态更符合半字节“与”故障后的理论分布. 2013 年，Fuhr 等学者提出针对 AES 算法的统计故障分析，并应用 SEI、HW 和 MLE 等区分器[27]. 后来，Li 等学者提出将 GF、GF-SEI 和 MLE-SEI 等区分器应用于 LED 等密码的分析中[29]. 本节提出了 PRESENT 密码的 PCC-HW、KLD-HW 和 JSC-HW-MLE 等新型区分器，并应用于中间相遇统计故障分析中.

4.4.1　现有区分器

(1) 平方欧式距离区分器

欧氏距离是由数学家 Euclid 提出，其平方值称为平方欧式距离. Fuhr 等学者提出平方欧式距离区分器，用于计算样本值的实际分布与均匀分布的平方欧氏距离，用于判断样本值的偏差程度. SEI 最大值表示该错误中间状态与平均分布的距离最远，对应的密钥候选值为正确子密钥的概率最大，表达式为

$$\mathrm{SEI}=\sum_{n=0}^{N}\left(Q_n-\frac{1}{16}\right)^2,$$

其中，Q_n 为中间状态的实际概率，n 为错误中间状态值且 $n\in[0,N]$，通常 $N=15$.

(2) 汉明重量区分器

1954 年，数学家 Reed 提出汉明重量的概念，用于表示二进制字符串中“1”的个数[35]. 注入故障会增加“0”的数量，减少“1”的数量，从而降低半字节中间状态的理论汉明重量，因此 HW 最小值代表该错误中间状态最可能为受故障影响的中间状态，其候选密钥为正确子密钥，表达式为

$$\mathrm{HW}=\frac{1}{f}\sum_{n=0}^{N}(hw_n\cdot V_n),$$

其中，hw_n 为二进制字符串的汉明重量，V_n 表示中间状态实际个数，f 为故障数，n 为错误中间状态值且 $n\in[0,N]$.

(3) 极大似然估计区分器

1938 年，数学家 Wilks 提出极大似然估计，用于估计概率模型的参数[36]. 极大似然估计区分器通过收集的若干实验样本数据，从结果反推使样本出现概率最大的参数. Fuhr 等学者通过建立似然函数，计算每一组中间状态理论分布出现的概率乘积，MLE 最大值表示该中间状态满足理论分布的概率，即候选密钥值为正确子密钥的概率最大，即：

$$\mathrm{MLE}=\prod_{n=0}^{N}(P_n)^{V_n},$$

其中，P_n 为中间状态的理论概率，V_n 为中间状态实际个数，n 为错误中间状态值且 $n\in[0,N]$.

(4) 拟合优度区分器

拟合优度是由数学家 Pearson 于 1990 年提出，用于检验实验结果是否遵循理论分布[37]. 拟合优度区分器是依据某种分布模型，计算一组数据的实际分布与理论分布的拟合优度. GF 最小值表示中间状态的实际分布与理论分布的差异最小，拟合程度最高，该中间状态受故障影响的概率最大，对应的密钥候选值最可能为正确子密钥，即：

$$\mathrm{GF}=\sum_{n=0}^{N}\frac{(U_n-V_n)^2}{U_n},$$

其中，U_n 为中间状态的理论个数，V_n 为中间状态实

际个数，n 为错误中间状态值且 $n\in[0,N]$.

(5)最大后验估计区分器

最大后验估计是贝叶斯模型的常用方法，在极大似然估计的基础上考虑被估计量的先验概率分布. MAP 区分器的最大值表示在先验假设下，该错误中间状态满足理论分布的可能性最大，对应着正确的子密钥，Li 等学者采用的先验假设为故障模型的概率分布函数，表达式为

$$\mathrm{MAP}=\frac{P(\psi|ck_m)\cdot g(ck_m)}{\sum_{m=1}^{M}P(\psi|ck_m)\cdot g(ck_m)},$$

其中，$P(\cdot)$为半字节的概率分布函数，$g(\cdot)$为先验分布函数，ck_m为密钥第 m 个猜测值，ψ 为注入故障处的半字节中间状态，其中 $m\in[0,M]$，$M=2^{24}$.

(6)拟合优度-平方欧式距离区分器

双重区分器通过结合两个区分器的优点，可以提高统计分析效率，拟合优度-平方欧式距离区分器是首个被提出的双重区分器. 先使用拟合优度区分器筛选和理论分布高度拟合的密钥候选值，即：

$$\mathrm{GF}=\sum_{n=0}^{N}\frac{(U_n-V_n)^2}{U_n},$$

通过使用平方欧式距离区分器对上述筛选出的密钥候选值进行二次筛选，同时满足 GF 最小值和 SEI 最大值的错误中间状态，其实际分布与理论分布差异最小且与平均分布距离最远，对应的密钥候选值为正确子密钥的概率最大，表达式为

$$\mathrm{SEI}=\sum_{n=0}^{N}\left(Q_n-\frac{1}{16}\right)^2,$$

其中，U_n为中间状态的理论个数，V_n为中间状态实际个数，Q_n为中间状态的实际概率，n 为错误中间状态值，且 $n\in[0,N]$.

4.4.2 新型区分器

(1)皮尔逊相关系数-汉明重量区分器

19 世纪 80 年代，数学家 Pearson 提出皮尔逊相关系数，适用于测量两个变量之间的相关程度[37]. 本节利用皮尔逊相关系数区分器统计分析错误中间状态，若区分器的绝对值越大，则实际分布和理论分布相关程度越高，并对应正确子密钥. 攻击者采用皮尔逊相关系数区分器缩小密钥候选值搜索范围，即：

$$\mathrm{PCC}=\left|\frac{\mathrm{cov}(P,Q)}{\sigma_P\sigma_Q}\right|,$$

接着，利用汉明重量区分器对皮尔逊系数区分器进行再次筛选，汉明重量最小值对应正确子密钥，即：

$$\mathrm{HW}=\frac{1}{f}\sum_{n=0}^{N}(hw_n\cdot V_n),$$

对于 PCC-HW 区分器，一组错误中间状态同时满足 PCC 最大值和 HW 最小值，则其实际分布与理论分布高度相关，且“0”的占比最大，该中间状态对应正确主密钥. 其中，$\mathrm{cov}(\cdot)$为中间状态理论分布和实际分布的协方差，σ 为中间状态分布样本的标准差，P 为中间状态的理论分布，Q 为中间状态实际分布，hw_n为二进制字符串的汉明重量，V_n为中间状态实际个数，f 为故障数，n 为错误中间状态值且 $n\in[0,N]$.

(2)库尔贝克莱布勒散度-汉明重量区分器

1951 年，密码分析学家和数学家 Kullback 和 Leibler 提出库尔贝克莱布勒散度，用于测量概率分布之间的相似度，广泛应用在机器学习领域[38]. 本节提出的库尔贝克莱布勒散度区分器，用于统计错误中间状态的理论分布与实际分布之间的相似度. 若区分器值越小，则实际分布和理论分布的相似度越高，对应正确子密钥. 具体过程为，使用库尔贝克莱布勒散度区分器筛选密钥候选值，即：

$$\mathrm{KLD}=\sqrt{\sum_{n=0}^{N}\left(Q_n\cdot\ln\frac{P_n}{Q_n}\right)},$$

然后，使用汉明重量区分器对库尔贝克莱布勒散度区分器筛选出的密钥候选值进行二次筛选，此时汉明重量值最小的密钥候选值，表达式为

$$\mathrm{HW}=\frac{1}{f}\sum_{n=0}^{N}(hw_n\cdot V_n),$$

对于 KLD-HW 区分器，同时满足 KLD 最小值和 HW 最小值的错误中间状态，其实际分布与理论分布之间损失的信息量最少，“0”的数量也明显多于“1”，对应了正确子密钥. 其中，P_n为中间状态的理论概率，Q_n为中间状态实际概率，hw_n为二进制字符串的汉明重量，V_n为中间状态实际个数，f 为故障数，n 为错误中间状态值且 $n\in[0,N]$.

(3)杰卡德相似系数-汉明重量-极大似然估计区分器

1912 年，数学家 Jaccard 提出杰卡德相似系数，用于计算有限样本集之间的相似度和差异度[39]. 该区分器可以用于衡量中间状态的理论分布和实际分布的相似度和差异度，其值越大表示相似程度越高，差异度越小，对应正确子密钥. 攻击者先使用杰卡德相似系数区分器缩小候选密钥的搜索空间，表达式为

$$\mathrm{JSC}=\frac{\sum_{n=0}^{N}\min(P_n,Q_n)}{\sum_{n=0}^{N}\max(P_n,Q_n)},$$

然后，使用汉明重量区分器对筛选出的密钥候选值

进行第二次筛选，以缩小搜索空间，表达式为

$$\mathrm{HW}=\frac{1}{f}\sum_{n=0}^{N}(hw_n\cdot V_n),$$

最后，使用极大似然估计区分器进行计算，若极大似然估计值最大，则对应正确子密钥，表达式为

$$\mathrm{MLE}=\prod_{n=0}^{N}(P_n)^{V_n},$$

对于 JSC-HW-MLE 区分器，当某组中间状态同时满足 JSC 最大值、HW 最小值和 MLE 最大值，其实际分布与理论分布的重合部分最大，“0”的占比最低，同时出现理论分布的概率乘积最大，对应了正确子密钥. 其中，P_n 为中间状态的理论概率，Q_n 为中间状态实际概率，hw_n 为二进制字符串的汉明重量，V_n 为中间状态实际个数，f 为故障数，n 为错误中间状态值且 $n\in[0,N]$. 表 4 列出了所有区分器的对比.

表 4　不同区分器的最值选择和筛选过程对比

区分器	取值	筛选过程
SEI	最大值	量化实际分布与均匀分布的距离，筛选出距离最远的统计样本
HW	最小值	量化中间状态的汉明重量，筛选“1”和“0”比例最小的统计样本
MLE	最大值	量化中间状态的理论概率乘积，筛选理论分布概率最大的统计样本
GF	最小值	量化中间状态的实际分布与理论分布的拟合程度，筛选出拟合程度最高的统计样本
MAP	最大值	量化中间状态在先验假设下的理论分布的概率乘积，筛选满足理论分布的可能性最大的统计样本
GF-SEI	GF 最小值，SEI 最大值	先后使用 GF 和 SEI，筛选出与理论分布拟合程度最高，且距离平均分布最远的实际分布
PCC-HW	PCC 最大值，HW 最小值	先后使用 PCC 和 HW，筛选出中间状态与理论分布相关性最高，且“1”占比最小的统计样本
KLD-HW	KLD 最小值，HW 最小值	先后使用 KLD 和 HW，筛选出中间状态信息损失最少，且“0”和“1”比例最高的统计样本
JSC-HW-MLE	JSC 最大值，HW 最小值，MLE 均取最大	先后使用 JSC、HW 和 MLE，筛选出与理论分布重合度最高，“1”占比最小，且出现理论分布概率最大的统计样本

5　实验分析

本实验在 PC 端(CPU 为 Intel Core I7-6700HQ，2.6 GHz，内存为 16 GB)上，利用计算机软件模拟随机故障导入，使用 Java 语言编程进行分析. 鉴于 PRESENT 密码全部版本的加密过程相同，攻击者可以采用相同的故障导入和攻击过程. 本节的中间相遇统计故障分析以恢复子密钥的 24 比特为实验数据单元，基于准确度、成功率、故障数、耗时和复杂度等指标评测不同区分器的攻击效果.

5.1　准确度

平均方根误差(*RMSE*)能够衡量不同区分器恢复 PRESENT 子密钥时的准确度. 若筛选出的密钥候选值的个数越少，则实际个数与理论个数之间的平均绝对误差越小，该区分器的准确度越高. *RMSE* 的表达式为

$$RMSE=\sqrt{\frac{1}{\theta}\sum_{\alpha=1}^{\theta}(h(\alpha)-1)},$$

其中，θ 表示实验次数，$h(\alpha)$ 表示第 α 次实验筛选出的密钥候选值的实际个数，理论个数为 1. 图 4 统计了各个区分器破译部分子密钥时，不同故障数对应的 *RMSE* 值. 其中横轴与纵轴分别表示故障数和 *RMSE* 值，不同标记曲线表示不同的区分器的准确度，*RMSE* 值越低，代表所选区分器的准确度越高. 与现有区分器相比，新型区分器 PCC-HW、KLD-HW 和 JSC-HW-MLE 的准确度更高且稳定，如图 4 和附录 A1 所示.

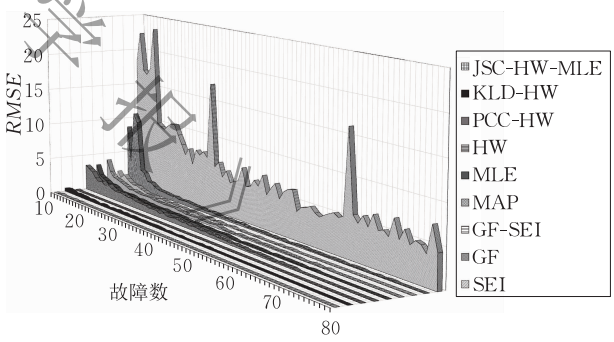

图 4　不同区分器恢复 PRESENT 部分子密钥的准确度

5.2　成功率

成功率是指不同区分器破译密码的概率. 图 5 统计了各个区分器恢复 24 比特子密钥时，不同故障数对应的成功率. 其中，横轴和纵轴分别表示故障数和破译成功率，不同标记曲线代表不同区分器破译成功率的变化趋势. 在故障数达到 100 时，SEI 的最高成功率仅为 10%，GF 和 GF-SEI 的最高成功率均为 97%，其它区分器的成功率均可达到 99%及以上. 新型区分器 PCC-HW、KLD-HW 和 JSC-HW-MLE 率先达到 99%及以上成功率，如图 5 和附录 A2 所示.

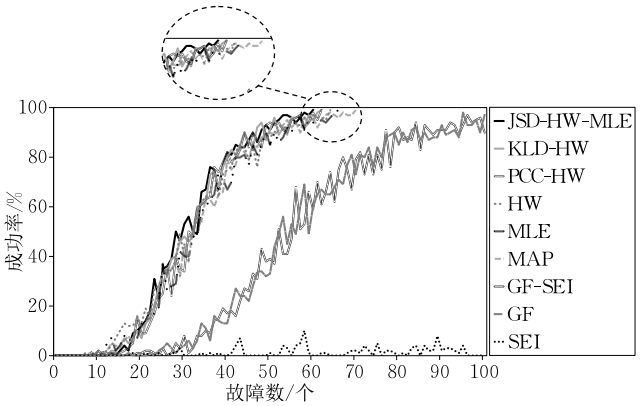

图 5 不同区分器恢复 PRESENT 部分子密钥的成功率

5.3 故障数

故障数是指不同区分器以最大概率破译密码所需最少故障数.在恢复主密钥时,需要重复攻击多次直到获取到 K_{32} 的全部比特,再依据 4.3 节的步骤 4 逆向推到得到主密钥.由于中间相遇统计故障分析在单次攻击时比统计故障分析多恢复 4 比特的密钥信息,因此仅需更少的攻击次数可以破译主密钥.以 PRESENT-80 为例,攻击者使用统计故障分析先重复攻击平均 4 次恢复最后一轮子密钥,再重复攻击平均 4 次恢复倒数第二轮子密钥的 16 比特,一共需要攻击平均 8 次,而中间相遇统计故障分析由于单次攻击多恢复的 4 比特密钥信息,一共攻击平均 7 次即可恢复主密钥.表 2 统计了传统统计故障分析和中间相遇统计故障分析在恢复 PRESENT 密码主密钥时的总故障数,计算如下:

$$F_{\mathrm{sfa}}=\begin{cases}8\cdot f_{\mathrm{sfa}}, & \text{在 PRESENT-80 中}\\ 9\cdot f_{\mathrm{sfa}}, & \text{在 PRESENT-128 中}\end{cases},$$

$$F_{\mathrm{msfa}}=\begin{cases}7\cdot f_{\mathrm{msfa}}, & \text{在 PRESENT-80 中}\\ 8\cdot f_{\mathrm{msfa}}, & \text{在 PRESENT-128 中}\end{cases},$$

其中,f_{sfa} 表示统计故障分析恢复 20 比特子密钥所需故障数,f_{msfa} 表示中间相遇统计故障分析恢复 24 比特子密钥所需故障数.以 PRESENT-128 版本为例,对于 SEI、GF、GF-SEI、MAP、MLE 和 HW 区分器,统计故障分析分别需要 86、109、99、87、61 和 60 个故障,即以最大概率破译 20 比特子密钥;对于 SEI、GF、GF-SEI、MAP、MLE 和 HW、PCC-HW、KLD-HW 和 JSC-HW-MLE 等区分器,中间相遇统计故障分析分别需要 89、107、105、70、66、64、62、62 和 60 个故障,能以最大的概率破译 24 比特子密钥.因此,对于新型区分器 PCC-HW、KLD-HW 和 JSC-HW-MLE,中间相遇统计故障分析最少仅需 496、496 和 480 个故障数可以恢复 128 比特主密钥,在故障数上更具优势,如表 2 所示.

5.4 耗时

耗时是指破译 PRESENT 算法所需要的时间,包括故障导入、密钥猜测和统计分析等过程.破译主密钥的总耗时等于破译部分子密钥的单次耗时乘以攻击次数.中间相遇统计故障分析一次攻击恢复 24 比特的子密钥信息,统计故障分析一次攻击恢复 20 比特的子密钥信息.因此,中间相遇统计故障分析仅需更少的攻击次数可以获取足够子密钥信息恢复主密钥.以 PRESENT-80 为例,统计故障分析先重复攻击平均 4 次恢复最后一轮子密钥,再攻击平均 4 次恢复倒数第二轮子密钥的信息,一共重复攻击平均 8 次,而中间相遇统计故障分析一共仅需平均 7 次即可获取足够的子密钥信息以破译主密钥.表 2 统计了传统统计故障分析和中间相遇统计故障分析在恢复 PRESENT 密码主密钥时的总耗时,计算如下:

$$T_{\mathrm{sfa}}=\begin{cases}8\cdot t_{\mathrm{sfa}}, & \text{在 PRESENT-80 中}\\ 9\cdot t_{\mathrm{sfa}}, & \text{在 PRESENT-128 中}\end{cases},$$

$$T_{\mathrm{msfa}}=\begin{cases}7\cdot t_{\mathrm{msfa}}, & \text{在 PRESENT-80 中}\\ 8\cdot t_{\mathrm{msfa}}, & \text{在 PRESENT-128 中}\end{cases},$$

其中,t_{sfa} 表示统计故障分析恢复 20 比特子密钥的耗时,t_{msfa} 表示中间相遇统计故障分析恢复 24 比特子密钥的耗时.以 PRESENT-128 版本为例,当成功率达到最高时,对于区分器 SEI、GF、GF-SEI、MAP、MLE 和 HW,统计故障分析恢复 20 比特子密钥需要时间分别为 93.9、118.7、75.8、59.7、60.0 和 48.1 s;对于区分器 SEI、GF、GF-SEI、MAP、MLE、HW、PCC-HW、KLD-HW 和 JSC-HW-MLE,中间相遇故障统计故障分析需要时间分别为 7.2、9.1、13.4、13.9、13.9、5.9、5.8、5.6 和 6.0 s,如附录 A3 和图 6 所示,其中,横轴与纵轴分别表示故障数和时间堆积,不同标记线条代表不同的区分器.因此,对于新

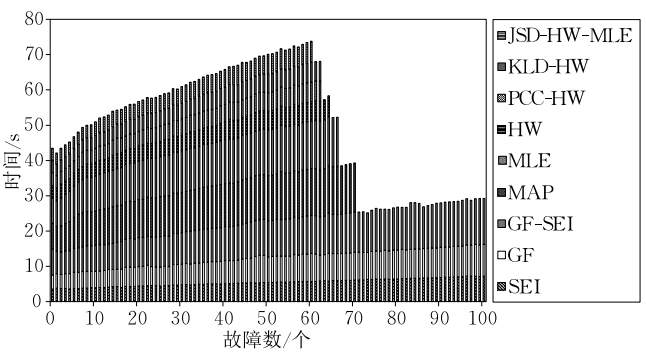

图 6 不同区分器恢复 PRESENT 部分子密钥的耗时

型区分器 KLD-HW、PCC-HW 和 JSC-HW-MLE，中间相遇统计故障分析恢复 128 比特主密钥的总耗时为 0.75、0.77 和 0.80 min，排名为第一、二和四，如表 2 所示.

5.5　复杂度

破译 PRESENT 密码所消耗的时间量和数据量可分别用时间复杂度和数据复杂度衡量. 时间复杂度可以计算为

$$F \cdot \psi \cdot \omega,$$

数据复杂度的计算公式为

$$F \cdot \psi,$$

其中，F 表示恢复主密钥所需的故障数，ψ 表示候选密钥个数，ω 为不同区分器的时间复杂度. 表 5 给出了 PRESENT 密码的 80 比特和 128 比特主密钥的破译结果，其中统计故障分析的数据是依据 De Santis 等学者所提出的攻击方案，复现攻击过程并统计现有区分器的数据[15]. 其中，对于中间相遇统计故障分析，区分器 KLD-HW、PCC-HW 和 JSC-HW-MLE 在时间复杂度和数据复杂度值排名前列，并优于现有双区分器，如表 5 所示.

综合表 2 和表 5，与传统统计故障分析相比，本节提出的针对 PRESENT 密码的中间相遇统计故障分析不仅可以将注入轮数更深一轮，而且在故障数、耗时以及复杂度方面均具有优势.

表 5　不同区分器的时间复杂度和数据复杂度对比

区分器	统计故障分析[15]				中间相遇统计故障分析			
	PRESENT-80		PRESENT-128		PRESENT-80		PRESENT-128	
	时间	数据	时间	数据	时间	数据	时间	数据
HW	$2^{32.99}$	$2^{28.91}$	$2^{33.16}$	$2^{29.08}$	$2^{28.89}$	$2^{24.81}$	$2^{29.09}$	$2^{25.00}$
MLE	$2^{33.02}$	$2^{28.93}$	$2^{33.19}$	$2^{29.10}$	$2^{28.94}$	$2^{24.85}$	$2^{29.13}$	$2^{25.04}$
SEI	$2^{33.43}$	$2^{29.43}$	$2^{33.60}$	$2^{29.60}$	$2^{29.28}$	$2^{25.28}$	$2^{29.48}$	$2^{25.48}$
GF	$2^{33.77}$	$2^{29.77}$	$2^{33.94}$	$2^{29.94}$	$2^{29.55}$	$2^{25.55}$	$2^{29.74}$	$2^{25.71}$
MAP	$2^{33.61}$	$2^{29.44}$	$2^{33.78}$	$2^{29.61}$	$2^{29.11}$	$2^{24.94}$	$2^{29.30}$	$2^{25.13}$
GF-SEI	$2^{33.72}$	$2^{29.63}$	$2^{33.89}$	$2^{29.80}$	$2^{29.61}$	$2^{25.52}$	$2^{29.80}$	$2^{25.71}$
PCC-HW	—	—	—	—	$2^{28.93}$	$2^{24.76}$	$2^{29.12}$	$2^{24.95}$
KLD-HW	—	—	—	—	$2^{28.93}$	$2^{24.76}$	$2^{29.12}$	$2^{24.95}$
JSC-HW-MLE	—	—	—	—	$2^{28.96}$	$2^{24.71}$	$2^{29.15}$	$2^{24.91}$

注：本表采用文献[15]的统计故障分析方法，通过复现统计获得复杂度.

6　结束语

本文提出了针对 PRESENT 轻量级密码算法的中间相遇统计故障分析，结合统计分析和中间相遇策略，优化了现有统计故障分析结果，并设计实现了 PCC-HW、KLD-HW 和 JSC-HW-MLE 等新型区分器，降低了攻击代价，提升了攻击效果. 由此可见，在唯密文攻击下，中间相遇统计故障分析对 PRESENT 轻量级密码的安全性产生严重威胁. 在物联网中使用该密码时，应考虑对其更多轮数加以保护. 该结果为轻量级密码抵御故障分析提供了有价值的参考.

参　考　文　献

[1] Naito Y, Sasaki Y, Sugawara T. Lightweight authenticated encryption mode suitable for threshold implementation// Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques. Zagreb, Croatia, 2020: 705-735

[2] Iwata T, Khairallah M, Minematsu K, et al. Duel of the titans: The romulus and remus families of lightweight AEAD algorithms. IACR Transactions on Symmetric Cryptology, 2020, 2020(1): 43-120

[3] Naito Y, Sugawara T. Lightweight authenticated encryption mode of operation for tweakable block ciphers. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2020, 2020(1): 66-94

[4] Daemen J, Hoffert S, Peeters M, et al. Xoodyak, a lightweight cryptographic scheme. IACR Transactions on Symmetric Cryptology, 2020, 2020(S1): 60-87

[5] Bogdanov A, Knudsen L R, Leander G, et al. PRESENT: An ultra-lightweight block cipher//Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems. Vienna, Austria, 2007: 450-466

[6] ISO/IEC 29192-2: 2019. Information security-lightweight cryptography-Part 2: Block ciphers. International Organization for Standardization, 2019

[7] Wang Mei-Qin. Differential cryptanalysis of reduced-round PRESENT//Proceedings of the International Conference on Cryptology in Africa. Casablance, Morocco, 2008: 40-49

[8] Z'aba M R, Raddum H, Henricksen M, et al. Bit-pattern based integral attack//Proceedings of the International Workshop on Fast Software Encryption. Lausanne, Switzerland, 2008:

363-381

[9] Nakahara J, Sepehrdad P, Zhang Bing-Sheng, et al. Linear (hull) and algebraic cryptanalysis of the block cipher PRESENT //Proceedings of the International Conference on Cryptology and Network Security. Kanazawa, Japan, 2009: 58-75

[10] Yang Lin, Wang Mei-Qin, Qiao Si-Yuan. Side channel cube attack on PRESENT//Proceedings of the International Conference on Cryptology and Network Security. Kanazawa, Japan, 2009: 379-391

[11] Cho J Y. Linear cryptanalysis of reduced-round PRESENT// Proceedings of the Cryptographers' Track at the RSA Conference. San Francisco, USA, 2010: 302-317

[12] Gong Zheng, Liu Shu-Sheng, Wen Ya-Min, et al. Biclique cryptanalysis of the reduced-round PRESENT. Chinese Journal of Computers, 2013, 36(6): 1139-1148(in Chinese)
(龚征, 刘树生, 温雅敏等. 缩减轮数 PRESENT 算法的 Biclique 分析. 计算机学报, 2013, 36(6): 1139-1148)

[13] Wang Gao-Li, Wang Shao-Hui. Differential fault analysis on PRESENT key schedule//Proceedings of the International Conference on Computational Intelligence and Security. Nanning, China, 2010: 362-366

[14] Zhao Xin-Jie, Guo Shi-Ze, Wang Tao, et al. Fault-propagate pattern based DFA on PRESENT and PRINT cipher. Wuhan University Journal of Natural Sciences, 2012, 17(6): 485-493

[15] De Santis F, Guillen O M, Sakic E, et al. Ciphertext-only fault attacks on PRESENT//Proceedings of the International Workshop on Lightweight Cryptography for Security and Privacy. Istanbul, Turkey, 2014: 85-108

[16] Huang Jing, Zhao Xin-Jie, Zhang Fan, et al. Improvement and evaluation for algebraic fault attacks on PRESENT. Journal on Communications, 2016, 37(8): 144-156(in Chinese)
(黄静, 赵新杰, 张帆等. PRESENT 代数故障攻击的改进与评估. 通信学报, 2016, 37(8): 144-156)

[17] Boneh D, DeMillo R A, Lipton R J. On the importance of checking cryptographic protocols for faults//Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Konstanz, Germany, 1997: 37-51

[18] Dusart P, Letourneux G, Vivolo O. Differential fault analysis on AES//Proceedings of the International Conference on Applied Cryptography and Network Security. Kunming, China, 2003: 293-306

[19] Derbez P, Fouque P A, Leresteux D. Meet-in-the-middle and impossible differential fault analysis on AES//Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems. Nara, Japan, 2011: 274-291

[20] Courtois N T, Jackson K, Ware D. Fault-algebraic attacks on inner rounds of DES//Proceedings of the Strategies Telecom and Multimedia. Sophia Antipolis, France, 2010: 22-24

[21] Biham E, Shamir A. Differential fault analysis of secret key cryptosystems//Proceedings of the Annual International Cryptology Conference. Santa Barbara, USA, 1997: 513-525

[22] Zhang Lei, Wu Wen-Ling. Differential fault analysis on SMS4. Chinese Journal of Computers, 2006, 29(9): 1594-1600(in Chinese)
(张蕾, 吴文玲. SMS4 密码算法的差分故障攻击. 计算机学报, 2006, 29(9): 1594-1600)

[23] Li Wei, Gu Da-Wu, Zhao Chen, et al. Security analysis of the LED lightweight cipher in the Internet of Things. Chinese Journal of Computers, 2012, 35(3): 434-445(in Chinese)
(李玮, 谷大武, 赵辰等. 物联网环境下 LED 轻量级密码算法的安全性分析. 计算机学报, 2012, 35(3): 434-445)

[24] Zhao Xin-Jie, Guo Shi-Ze, Wang Tao, et al. Research of algebraic fault analysis on Piccolo. Chinese Journal of Computers, 2013, 36(4): 882-894(in Chinese)
(赵新杰, 郭世泽, 王韬等. Piccolo 密码代数故障分析研究. 计算机学报, 2013, 36(4): 882-894)

[25] Gruber M, Probst M, Tempelmeier M. Statistical ineffective fault analysis of GIMLI//Proceedings of the IEEE International Workshop on Hardware Oriented Security and Trust. San Jose, USA, 2020: 252-261

[26] Zhang Fan, Zhang Yi-Ran, Jiang Hui-Long, et al. Persistent fault attack in practice. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2020, 2020(2): 172-195

[27] Zheng S, Liu X, Zang S, et al. A persistent fault-based collision analysis against the advanced encryption standard. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2021, 40(6): 1117-1129

[28] Fuhr T, Jaulmes E, Lomné V, et al. Fault attacks on AES with faulty ciphertexts only//Proceedings of the 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography. Washington, USA, 2013: 108-118

[29] Dobraunig C, Eichlseder M, Korak T, et al. Statistical fault attacks on nonce-based authenticated encryption schemes// Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security. Berlin, Germany: Springer, 2016: 369-395

[30] Li Wei, Liao Lin-Feng, Gu Da-Wu, et al. Ciphertext-only fault analysis on the LED lightweight cryptosystem in the Internet of Things. IEEE Transactions on Dependable and Secure Computing, 2019, 16(3): 454-461

[31] Li Wei, Li Jia-Yao, Gu Da-Wu, et al. Statistical fault analysis of the Piccolo lightweight cryptosystem. Chinese Journal of Computers, 2021, 44(10): 2104-2121(in Chinese)
(李玮, 李嘉耀, 谷大武等. 轻量级密码算法 Piccolo 的统计故障分析. 计算机学报, 2021, 44(10): 2104-2121)

[32] Diffie W, Hellman M E. Special feature exhaustive cryptanalysis of the NBS Data Encryption Standard. Computer, 1977, 10(6): 74-84

[33] Biham E, Biryukov A, Shamir A. Cryptanalysis of Skipjack

reduced to 31 rounds using impossible differentials//Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques. Prague, Czech Republic, 1999: 12-23

[34] Bogdanov A, Khovratovich D, Rechberger C. Biclique cryptanalysis of the full AES//Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security. Seoul, Korea, 2011: 344-371

[35] Reed I. A class of multiple-error-correcting codes and the decoding scheme. Transactions of the IRE Professional Group on Information Theory, 1954, 4(4): 38-49

[36] Wilks S S. The large-sample distribution of the likelihood ratio for testing composite hypotheses. The Annals of Mathematical Statistics, 1938, 9(1): 60-62

[37] Pearson K. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 1900, 50(302): 157-175

[38] Kullback S, Leibler R A. On information and sufficiency. The Annals of Mathematical Statistics, 1951, 22(1): 79-86

[39] Jaccard P. The distribution of the flora in the alpine zone. New Phytologist, 1912, 11(2): 37-50

附录 A.　实验数据.

明文：随机生成

PRESENT-80 版本主密钥：0123 4567 8901 2345 6789

PRESENT-128 版本主密钥：0123 4567 89ab cedf fedc ba98 7654 3210

表 A1　各区分器恢复 PRESENT 算法 24 比特子密钥的 *RMSE* 值

故障数	HW	MLE	SEI	GF	MAP	GF-SEI	PCC-HW	KLD-HW	JSD-HW-MLE
0	4096.00	4096.00	4096.00	4096.00	4096.00	4096.00	4096.00	4096.00	4096.00
1	1024.00	1024.00	1097.82	1024.00	1024.00	1024.00	65.16	256.00	63.99
2	244.46	278.09	671.76	764.39	264.57	248.12	12.57	39.96	8.12
3	131.28	117.38	387.65	444.71	114.81	228.98	1.56	9.81	1.04
4	50.03	61.02	178.82	123.86	52.69	60.36	0.79	2.87	0.71
5	26.69	27.56	102.90	73.11	34.50	61.57	0.24	1.47	0.17
6	14.04	13.14	63.32	19.34	16.12	18.38	0.14	1.02	0.26
7	9.09	8.53	67.76	6.95	9.47	7.47	0.00	0.62	0.10
8	6.37	5.95	56.47	5.13	6.48	3.93	0.10	0.65	0.17
9	4.35	4.48	43.04	3.93	4.55	2.99	0.00	0.41	0.00
10	3.51	3.42	18.29	8.63	4.00	2.19	0.10	0.46	0.00
11	3.17	2.07	22.39	3.72	2.45	1.40	0.00	0.54	0.00
12	2.84	1.80	17.12	10.83	1.76	1.96	0.00	0.45	0.14
13	2.15	1.58	16.60	9.72	1.64	1.51	0.00	0.22	0.00
14	1.78	1.27	17.55	2.81	1.44	1.14	0.00	0.33	0.00
15	1.97	1.30	23.33	2.56	1.26	0.97	0.00	0.28	0.00
16	1.56	1.11	9.89	1.51	1.05	1.01	0.00	0.26	0.10
17	1.85	0.72	9.50	1.50	0.79	0.77	0.00	0.14	0.00
18	1.50	0.73	9.28	1.13	0.91	0.64	0.00	0.22	0.10
19	1.30	0.72	9.88	0.85	0.76	0.45	0.00	0.14	0.00
20	1.28	0.80	10.34	0.79	0.69	0.45	0.00	0.00	0.00
21	1.36	0.74	10.35	0.77	0.69	0.48	0.00	0.17	0.00
22	1.21	0.55	9.05	0.69	0.73	0.26	0.00	0.20	0.00
23	1.12	0.67	7.83	0.63	0.74	0.39	0.00	0.00	0.00
24	1.17	0.61	6.57	0.72	0.59	0.33	0.00	0.14	0.00
25	0.97	0.49	7.53	0.51	0.54	0.20	0.00	0.10	0.00
26	0.92	0.54	5.56	0.57	0.54	0.36	0.00	0.22	0.00
27	0.91	0.41	7.67	0.62	0.49	0.26	0.00	0.10	0.00
28	0.89	0.44	7.13	0.54	0.54	0.24	0.00	0.20	0.00
29	0.72	0.44	7.77	0.48	0.54	0.28	0.00	0.10	0.00
30	0.79	0.57	7.12	0.45	0.57	0.17	0.00	0.17	0.00
31	0.77	0.46	17.35	0.49	0.40	0.22	0.00	0.14	0.00
32	0.56	0.39	6.14	0.32	0.42	0.20	0.00	0.10	0.00
33	0.81	0.50	6.94	0.33	0.28	0.24	0.00	0.14	0.00
34	0.57	0.46	5.09	0.44	0.35	0.10	0.00	0.10	0.00
35	0.57	0.41	6.09	0.37	0.32	0.14	0.00	0.10	0.00
36	0.71	0.33	4.62	0.28	0.42	0.20	0.00	0.14	0.00

（续表 A1）

故障数	HW	MLE	SEI	GF	MAP	GF-SEI	PCC-HW	KLD-HW	JSD-HW-MLE
37	0.62	0.28	5.01	0.22	0.30	0.00	0.00	0.14	0.00
38	0.54	0.32	5.28	0.33	0.28	0.17	0.00	0.10	0.00
39	0.61	0.32	7.19	0.24	0.26	0.10	0.00	0.00	0.00
40	0.44	0.20	4.87	0.20	0.33	0.00	0.00	0.10	0.00
41	0.56	0.30	5.00	0.32	0.40	0.10	0.00	0.00	0.00
42	0.52	0.26	6.04	0.26	0.22	0.17	0.00	0.00	0.00
43	0.39	0.20	5.42	0.14	0.10	0.00	0.00	0.10	0.00
44	0.52	0.22	7.03	0.17	0.10	0.10	0.00	0.00	0.00
45	0.36	0.30	4.63	0.17	0.00	0.00	0.00	0.00	0.00
46	0.44	0.33	5.65	0.22	0.22	0.00	0.00	0.00	0.00
47	0.35	0.20	6.48	0.20	0.17	0.00	0.00	0.00	0.00
48	0.37	0.14	4.65	0.10	0.10	0.10	0.00	0.10	0.00
49	0.36	0.00	5.77	0.17	0.20	0.00	0.00	0.00	0.00
50	0.24	0.14	6.24	0.10	0.20	0.00	0.00	0.10	0.00
51	0.36	0.14	6.38	0.22	0.10	0.00	0.00	0.00	0.00
52	0.26	0.14	3.78	0.00	0.17	0.00	0.00	0.00	0.00
53	0.30	0.14	4.49	0.10	0.17	0.10	0.00	0.00	0.00
54	0.17	0.14	4.64	0.17	0.14	0.00	0.00	0.00	0.00
55	0.17	0.10	5.00	0.17	0.10	0.10	0.00	0.00	0.00
56	0.28	0.00	4.36	0.22	0.14	0.10	0.00	0.00	0.00
57	0.24	0.14	4.03	0.10	0.14	0.00	0.00	0.00	0.00
58	0.22	0.14	4.46	0.14	0.10	0.00	0.00	0.00	0.00
59	0.17	0.10	4.83	0.14	0.10	0.00	0.00	0.00	0.00
60	0.10	0.14	5.16	0.10	0.10	0.00	0.00	0.00	0.00
61	0.22	0.10	4.76	0.10	0.00	0.00	0.00	0.00	—
62	0.14	0.10	5.09	0.14	0.00	0.00	0.00	0.00	—
63	0.17	0.10	6.21	0.14	0.14	0.00	—	—	—
64	0.17	0.00	5.66	0.10	0.10	0.00	—	—	—
65	—	0.10	4.83	0.00	0.00	0.00	—	—	—
66	—	0.10	5.94	0.10	0.14	0.00	—	—	—
67	—	—	4.78	0.10	0.00	0.00	—	—	—
68	—	—	6.49	0.10	0.00	0.00	—	—	—
69	—	—	4.76	0.00	0.00	0.00	—	—	—
70	—	—	5.03	0.00	0.00	0.00	—	—	—
71	—	—	4.44	0.10	—	0.00	—	—	—
72	—	—	6.84	0.00	—	0.00	—	—	—
73	—	—	4.50	0.10	—	0.00	—	—	—
74	—	—	5.92	0.10	—	0.00	—	—	—
75	—	—	4.31	0.00	—	0.00	—	—	—
76	—	—	5.01	0.00	—	0.00	—	—	—
77	—	—	4.76	0.00	—	0.00	—	—	—
78	—	—	3.91	0.00	—	0.00	—	—	—
79	—	—	7.51	0.00	—	0.00	—	—	—
80	—	—	4.42	0.00	—	0.00	—	—	—
81	—	—	4.98	0.10	—	0.00	—	—	—
82	—	—	4.25	0.00	—	0.00	—	—	—
83	—	—	6.46	0.10	—	0.00	—	—	—
84	—	—	5.09	0.00	—	0.00	—	—	—
85	—	—	6.37	0.00	—	0.00	—	—	—
86	—	—	4.83	0.00	—	0.00	—	—	—
87	—	—	5.14	0.00	—	0.00	—	—	—
88	—	—	5.75	0.00	—	0.00	—	—	—
89	—	—	5.04	0.00	—	0.00	—	—	—
90	—	—	5.08	0.00	—	0.00	—	—	—
91	—	—	4.21	0.10	—	0.00	—	—	—
92	—	—	4.97	0.00	—	0.00	—	—	—
93	—	—	4.67	0.00	—	0.00	—	—	—
94	—	—	5.31	0.00	—	0.00	—	—	—
95	—	—	6.01	0.00	—	0.00	—	—	—
96	—	—	5.24	0.00	—	0.00	—	—	—
97	—	—	4.89	0.00	—	0.00	—	—	—
98	—	—	3.99	0.00	—	0.00	—	—	—
99	—	—	5.52	0.00	—	0.00	—	—	—
100	—	—	6.69	0.00	—	0.00	—	—	—

表 A2　各区分器恢复 PRESENT 算法 24 比特子密钥的成功率

故障数	HW	MLE	SEI	GF	MAP	GF-SEI	PCC-HW	KLD-HW	JSD-HW-MLE
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.02	0.00	0.00	0.01	0.00	0.00	0.00	0.00
9	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.02	0.02	0.00	0.00	0.00
11	0.01	0.01	0.00	0.00	0.01	0.00	0.00	0.01	0.00
12	0.02	0.05	0.00	0.00	0.02	0.00	0.01	0.00	0.00
13	0.04	0.07	0.00	0.00	0.05	0.00	0.01	0.02	0.01
14	0.07	0.02	0.00	0.00	0.04	0.00	0.01	0.02	0.00
15	0.10	0.04	0.00	0.00	0.03	0.00	0.01	0.04	0.02
16	0.13	0.08	0.00	0.00	0.06	0.00	0.02	0.07	0.04
17	0.12	0.06	0.00	0.01	0.08	0.00	0.03	0.05	0.02
18	0.12	0.07	0.01	0.00	0.05	0.00	0.05	0.07	0.09
19	0.15	0.07	0.00	0.00	0.16	0.01	0.15	0.10	0.10
20	0.13	0.13	0.00	0.00	0.11	0.01	0.09	0.07	0.12
21	0.19	0.14	0.01	0.00	0.13	0.02	0.09	0.13	0.14
22	0.15	0.16	0.00	0.01	0.18	0.02	0.13	0.15	0.18
23	0.27	0.17	0.00	0.00	0.20	0.01	0.17	0.20	0.31
24	0.28	0.19	0.01	0.02	0.25	0.00	0.23	0.28	0.23
25	0.33	0.37	0.00	0.03	0.30	0.03	0.25	0.33	0.35
26	0.43	0.26	0.00	0.05	0.32	0.00	0.36	0.33	0.36
27	0.34	0.28	0.01	0.03	0.35	0.01	0.24	0.39	0.39
28	0.39	0.29	0.01	0.01	0.38	0.05	0.40	0.40	0.50
29	0.49	0.42	0.03	0.03	0.32	0.03	0.35	0.46	0.46
30	0.40	0.40	0.00	0.04	0.45	0.08	0.34	0.48	0.53
31	0.47	0.40	0.00	0.06	0.38	0.01	0.50	0.46	0.56
32	0.50	0.48	0.00	0.03	0.52	0.03	0.48	0.48	0.50
33	0.55	0.48	0.00	0.06	0.57	0.08	0.59	0.60	0.49
34	0.62	0.58	0.01	0.13	0.54	0.08	0.61	0.61	0.66
35	0.64	0.60	0.00	0.11	0.64	0.10	0.63	0.62	0.67
36	0.66	0.63	0.01	0.06	0.63	0.09	0.64	0.73	0.76
37	0.74	0.65	0.00	0.16	0.60	0.14	0.76	0.74	0.75
38	0.64	0.74	0.01	0.14	0.65	0.15	0.74	0.68	0.67
39	0.73	0.71	0.00	0.13	0.66	0.19	0.78	0.76	0.77
40	0.75	0.67	0.00	0.13	0.79	0.17	0.82	0.80	0.82
41	0.76	0.70	0.01	0.20	0.75	0.18	0.78	0.83	0.81
42	0.82	0.80	0.04	0.27	0.77	0.16	0.80	0.81	0.85
43	0.85	0.80	0.07	0.24	0.84	0.22	0.83	0.85	0.84
44	0.84	0.81	0.00	0.24	0.82	0.27	0.90	0.90	0.84
45	0.79	0.77	0.00	0.22	0.79	0.26	0.84	0.86	0.87
46	0.84	0.85	0.00	0.28	0.86	0.33	0.90	0.84	0.90
47	0.76	0.81	0.00	0.34	0.84	0.31	0.87	0.92	0.86
48	0.87	0.85	0.00	0.34	0.88	0.43	0.84	0.94	0.91
49	0.89	0.90	0.00	0.32	0.87	0.38	0.93	0.88	0.93
50	0.88	0.85	0.01	0.33	0.90	0.39	0.94	0.91	0.91
51	0.90	0.90	0.00	0.44	0.89	0.37	0.93	0.96	0.94
52	0.93	0.86	0.00	0.49	0.91	0.51	0.94	0.89	0.95
53	0.92	0.88	0.04	0.56	0.93	0.41	0.91	0.92	0.97
54	0.92	0.92	0.03	0.45	0.95	0.41	0.90	0.89	0.97
55	0.96	0.90	0.00	0.53	0.92	0.47	0.95	0.96	0.96
56	0.93	0.96	0.04	0.49	0.94	0.68	0.96	0.96	0.94
57	0.93	0.96	0.05	0.60	0.90	0.51	0.97	0.97	0.96
58	0.96	0.91	0.10	0.69	0.96	0.53	0.92	0.92	0.98
59	0.96	0.95	0.01	0.49	0.94	0.66	0.94	0.97	0.97
60	0.97	0.94	0.00	0.60	0.96	0.59	0.98	0.95	0.99
61	0.96	0.94	0.00	0.65	0.92	0.70	0.95	0.98	—
62	0.96	0.95	0.01	0.67	0.95	0.61	0.99	0.99	—
63	0.95	0.94	0.01	0.65	0.97	0.66	—	—	—
64	0.99	0.96	0.00	0.65	0.98	0.74	—	—	—
65	—	0.98	0.00	0.68	0.97	0.62	—	—	—

（续表 A2）

故障数	HW	MLE	SEI	GF	MAP	GF-SEI	PCC-HW	KLD-HW	JSD-HW-MLE
66	—	0.99	0.00	0.71	0.96	0.75	—	—	—
67	—	—	0.00	0.69	0.98	0.80	—	—	—
68	—	—	0.01	0.77	0.97	0.75	—	—	—
69	—	—	0.02	0.81	0.97	0.67	—	—	—
70	—	—	0.02	0.73	0.99	0.79	—	—	—
71	—	—	0.01	0.81	—	0.82	—	—	—
72	—	—	0.04	0.80	—	0.73	—	—	—
73	—	—	0.02	0.84	—	0.81	—	—	—
74	—	—	0.00	0.76	—	0.73	—	—	—
75	—	—	0.05	0.84	—	0.81	—	—	—
76	—	—	0.00	0.80	—	0.75	—	—	—
77	—	—	0.02	0.88	—	0.91	—	—	—
78	—	—	0.02	0.89	—	0.76	—	—	—
79	—	—	0.01	0.90	—	0.90	—	—	—
80	—	—	0.00	0.85	—	0.92	—	—	—
81	—	—	0.01	0.90	—	0.87	—	—	—
82	—	—	0.00	0.86	—	0.88	—	—	—
83	—	—	0.03	0.91	—	0.87	—	—	—
84	—	—	0.04	0.92	—	0.94	—	—	—
85	—	—	0.00	0.87	—	0.83	—	—	—
86	—	—	0.04	0.90	—	0.93	—	—	—
87	—	—	0.02	0.87	—	0.91	—	—	—
88	—	—	0.02	0.88	—	0.89	—	—	—
89	—	—	0.10	0.88	—	0.91	—	—	—
90	—	—	0.02	0.93	—	0.91	—	—	—
91	—	—	0.03	0.89	—	0.91	—	—	—
92	—	—	0.03	0.90	—	0.94	—	—	—
93	—	—	0.02	0.91	—	0.93	—	—	—
94	—	—	0.01	0.88	—	0.89	—	—	—
95	—	—	0.04	0.98	—	0.93	—	—	—
96	—	—	0.00	0.93	—	0.97	—	—	—
97	—	—	0.00	0.93	—	0.94	—	—	—
98	—	—	0.00	0.95	—	0.95	—	—	—
99	—	—	0.00	0.90	—	0.96	—	—	—
100	—	—	0.00	0.97	—	0.90	—	—	—

表 A3 各区分器恢复 PRESENT 算法 24 比特子密钥的耗时 （单位：s）

故障数	HW	MLE	SEI	GF	MAP	GF-SEI	PCC-HW	KLD-HW	JSD-HW-MLE
0	3.50	7.20	3.60	3.95	7.40	7.30	3.66	3.37	3.53
1	3.56	7.34	3.76	4.10	7.00	6.41	3.51	3.21	3.21
2	4.01	7.76	3.69	4.00	7.39	6.40	3.56	3.28	3.32
3	3.66	8.27	3.71	4.17	7.84	6.43	3.64	3.30	3.31
4	3.54	8.63	3.63	4.32	8.18	6.55	3.66	3.39	3.40
5	3.58	8.91	3.69	4.59	8.71	6.82	3.55	3.43	3.41
6	3.52	9.16	3.71	4.63	9.51	6.99	3.62	3.48	3.48
7	3.65	9.44	3.82	4.65	9.71	6.96	4.05	3.53	3.52
8	3.77	9.63	3.75	4.84	9.70	7.26	3.93	3.51	3.54
9	3.72	9.79	3.91	4.73	9.61	7.20	4.01	3.60	3.61
10	3.78	9.98	3.89	4.68	9.71	7.30	4.03	3.66	3.83
11	3.89	10.11	4.00	4.74	9.92	7.41	4.08	3.71	4.25
12	3.93	10.27	3.96	4.80	10.11	7.38	4.09	3.71	4.27
13	3.77	10.31	4.10	4.97	10.22	7.23	4.14	3.78	4.33
14	4.13	10.48	4.23	4.87	10.33	7.44	4.27	3.86	4.38
15	4.33	10.52	4.18	5.00	10.36	7.60	4.37	3.93	3.99
16	4.15	10.61	4.03	5.22	10.42	7.80	4.22	3.93	4.09
17	4.04	10.73	4.25	5.36	10.51	7.95	4.10	4.12	4.13
18	4.14	10.79	4.29	5.50	10.53	8.11	4.18	4.18	4.23
19	4.13	10.88	4.29	5.41	10.73	7.97	4.19	4.11	4.26
20	4.18	10.94	4.40	5.51	10.77	8.23	4.24	4.16	4.27
21	4.30	11.12	4.42	5.63	10.85	8.24	4.27	4.17	4.28
22	4.23	11.14	4.47	5.77	10.97	8.29	4.39	4.23	4.31
23	4.20	11.18	4.45	5.45	10.99	8.47	4.29	4.24	4.34
24	4.34	11.32	4.47	5.33	11.09	8.52	4.31	4.26	4.27

（续表 A3）

故障数	HW	MLE	SEI	GF	MAP	GF-SEI	PCC-HW	KLD-HW	JSD-HW-MLE
25	4.34	11.43	4.44	5.37	11.17	8.67	4.34	4.32	4.35
26	4.39	11.51	4.55	5.45	11.27	8.64	4.41	4.30	4.36
27	4.36	11.62	4.54	5.44	11.41	8.70	4.43	4.34	4.37
28	4.60	11.61	4.69	5.80	11.39	8.85	4.46	4.40	4.51
29	4.42	11.61	4.69	5.81	11.41	8.86	4.51	4.42	4.55
30	4.42	11.79	4.73	5.89	11.40	8.96	4.62	4.47	4.65
31	4.50	11.85	4.77	5.96	11.43	9.02	4.68	4.56	4.69
32	4.69	11.92	4.72	6.04	11.51	9.32	4.70	4.62	4.76
33	4.59	11.94	4.83	6.10	11.62	9.22	4.78	4.65	4.75
34	4.62	11.96	4.87	6.10	11.71	9.33	4.84	4.68	4.86
35	4.67	12.09	4.90	6.21	11.77	9.56	4.81	4.70	4.89
36	4.86	12.11	4.97	6.24	11.89	9.38	4.87	4.82	4.98
37	4.75	12.20	5.04	6.27	11.91	9.49	4.85	4.83	5.05
38	4.74	12.34	5.00	6.26	12.04	9.62	4.89	4.81	4.98
39	4.99	12.38	5.02	6.44	12.20	9.57	4.94	4.89	4.91
40	4.89	12.47	5.06	6.51	12.20	9.69	5.03	4.88	4.99
41	5.50	12.54	5.11	6.55	12.27	9.68	5.08	4.92	4.98
42	5.34	12.52	5.15	6.53	12.24	9.77	5.15	4.99	5.16
43	4.99	12.54	5.18	6.61	12.32	9.82	5.10	4.99	5.43
44	5.04	12.61	5.26	6.75	12.47	9.88	5.17	5.07	5.49
45	5.09	12.58	5.29	6.91	12.57	9.99	5.13	5.06	5.11
46	5.13	12.67	5.29	7.03	12.63	10.16	5.17	5.06	5.08
47	5.17	12.65	5.28	7.47	12.75	10.16	5.21	5.10	5.11
48	5.36	12.76	5.35	7.80	12.68	10.06	5.21	5.08	5.20
49	5.21	12.87	5.36	7.68	12.74	10.20	5.22	5.12	5.22
50	5.31	12.96	5.40	7.25	12.78	10.15	5.29	5.14	5.73
51	5.34	13.04	5.41	7.37	12.79	10.26	5.24	5.18	5.62
52	5.39	13.08	5.46	7.37	12.81	10.41	5.31	5.21	5.57
53	5.38	13.12	5.50	7.42	13.56	10.36	5.39	5.23	5.63
54	5.38	13.16	5.51	7.42	12.96	10.46	5.41	5.28	5.61
55	5.41	13.19	5.52	7.44	12.99	10.54	5.49	5.33	5.69
56	5.63	13.33	5.57	7.70	13.02	10.70	5.48	5.33	5.74
57	5.51	13.30	5.67	7.47	13.08	10.46	5.50	5.33	5.75
58	5.52	13.40	5.66	7.67	13.16	10.69	5.50	5.42	5.78
59	5.54	13.54	5.71	7.79	13.21	10.79	5.54	5.44	5.96
60	5.54	13.51	5.74	7.93	13.28	10.87	5.60	5.51	6.01
61	5.59	13.61	5.68	7.76	13.34	10.87	5.66	5.52	—
62	5.60	13.74	5.81	7.40	13.37	10.84	5.76	5.58	—
63	5.73	13.79	5.88	7.37	13.48	10.91	—	—	—
64	5.93	13.83	5.80	7.89	13.55	11.40	—	—	—
65	—	13.90	5.90	7.67	13.57	11.10	—	—	—
66	—	13.92	5.92	7.67	13.68	11.04	—	—	—
67	—	—	5.92	7.65	13.69	11.19	—	—	—
68	—	—	6.00	7.68	13.78	11.34	—	—	—
69	—	—	6.04	7.78	13.87	11.40	—	—	—
70	—	—	6.09	7.90	13.91	11.42	—	—	—
71	—	—	6.11	7.89	—	11.43	—	—	—
72	—	—	6.09	7.88	—	11.45	—	—	—
73	—	—	6.26	7.65	—	11.29	—	—	—
74	—	—	6.18	7.97	—	11.75	—	—	—
75	—	—	6.37	8.04	—	12.06	—	—	—
76	—	—	6.31	8.04	—	11.88	—	—	—
77	—	—	6.38	8.05	—	11.69	—	—	—
78	—	—	6.34	8.08	—	11.73	—	—	—
79	—	—	6.43	8.16	—	11.99	—	—	—
80	—	—	6.43	8.28	—	12.07	—	—	—
81	—	—	6.50	8.27	—	11.92	—	—	—
82	—	—	6.50	8.17	—	12.03	—	—	—
83	—	—	6.57	8.23	—	13.27	—	—	—
84	—	—	6.62	8.29	—	13.15	—	—	—
85	—	—	6.64	8.32	—	12.95	—	—	—
86	—	—	6.71	8.35	—	11.91	—	—	—
87	—	—	6.69	8.42	—	12.17	—	—	—
88	—	—	6.73	8.53	—	12.37	—	—	—
89	—	—	6.80	8.44	—	12.70	—	—	—
90	—	—	6.79	8.58	—	12.55	—	—	—
91	—	—	6.85	8.62	—	12.73	—	—	—

（续表 A3）

故障数	HW	MLE	SEI	GF	MAP	GF-SEI	PCC-HW	KLD-HW	JSD-HW-MLE
92	—	—	6.91	8.61	—	12.76	—	—	—
93	—	—	6.95	8.71	—	12.73	—	—	—
94	—	—	7.01	8.69	—	12.76	—	—	—
95	—	—	7.05	8.73	—	12.92	—	—	—
96	—	—	7.20	8.93	—	13.04	—	—	—
97	—	—	7.19	8.84	—	12.72	—	—	—
98	—	—	7.16	8.96	—	13.00	—	—	—
99	—	—	7.15	9.05	—	13.12	—	—	—
100	—	—	7.21	9.03	—	13.05	—	—	—

LI Wei, Ph. D., professor, Ph. D. supervisor. Her main research interests include the design and analysis of symmetric ciphers.

ZHU Xiao-Ming, M. S. candidate. His main research interests include fault analysis of block ciphers.

GU Da-Wu, Ph. D., professor, Ph. D. supervisor. His main research interests include cryptology and computer security.

LI Jia-Yao, Ph. D. candidate. His main research interests include fault analysis of block ciphers.

CAI Tian-Pei, M. S. candidate. His main research interests include security analysis of block ciphers.

Background

Our work is supported by the National Natural Science Foundation of China under Grant Nos. 61772129 and 61932014, the National Cryptography Development Fund under Grant No. MMJJ20180101, the Shanghai Natural Science Foundation of China under Grant No. 21YF1401200, the Opening Project of Shanghai Key Laboratory of Scalable Computing and Systems, the Opening Project of Shanghai Key Laboratory of Integrate Administration Technologies for Information Security, and the Fundamental Research Funds for the Central Universities.

The lightweight cryptosystem PRESENT, proposed at CHES, can be implemented with low storage and power consumption in the Internet of Things. It provides a wide range of options among area, throughput, and power consumption. The designers analyzed the security of PRESENT against differential analysis, linear analysis, differential analysis, boomerang analysis, meet-in-the-middle analysis, etc.

In addition to the above classical cryptanalysis, fault analysis has been a competitive threat of a cryptosystem since 1997. It can break the secret key of a cryptosystem by fault injections. Boneh et al. applied the fault analysis successfully to break the RSA cryptosystem with the exploitation of Chinese Remainder Theorem. Later, more kinds of fault analysis are presented, such as differential fault analysis, algebraic fault analysis, meet-in-the-middle fault analysis, statistical fault analysis, etc. The attacking assumptions of most classical cryptanalysis and some fault analysis focus on the known-plaintext attack (KPA) or the chosen-plaintext attack (CPA).

However, the above assumptions are not suitable to apply to the Internet of Things, and the attackers cannot have so strong ability to get the corresponding ciphertexts from the known or designated plaintexts. In the real scenario, the attackers may only have the weakest capability of ciphertext-only attack (COA). In 2013, Fuhr et al. proposed the statistical fault analysis (SFA) of AES in software implementation. Later, Dobraunig et al. successfully applied the SFA on some authenticated encryption schemas in 2016. Li et al. expanded the SFA with new distinguishers on the LED and other cryptosystems.

In the literature, the previous research on the SFA has its limitation, which motivates us to investigate novel strategies and distinguishers of SFA on PRESENT. Our study proposes the meet-in-the-middle statistical fault analysis of PRESENT with three novel distinguishers in the software implementation, which combines the advantages of meet-in-the-middle analysis and statistical fault analysis. The experimental results show that the novel distinguishers of PCC-HW, KLD-HW and JSC-HW-MLE can recover the secret key of PRESENT with the reliability of at least 99% in the MSFA, respectively. The proposed MSFA can make the fault injection deeper, and reduce the faults and latency. It provides a significant reference for analyzing the security of lightweight ciphers in the Internet of Things.