轻量级密码 TweGIFT 的中间相错统计故障分析研究

李 玮[1),2)] 刘 源[1)] 谷大武[2)] 黄佳音[1)] 陆海宁[3)]

[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)](上海交通大学网络空间安全学院 上海 200240)

摘 要 TweGIFT 算法是 Chakraborti 等学者于 2021 年提出的轻量级可调分组密码算法，旨在保护智慧城市、智能制造、智联农业等领域中物联网设备的数据安全。本文基于唯密文基本假设，采取随机半字节故障，提出了一种新型唯密文故障分析方法，即中间相错统计故障分析。该方法基于中间相错策略和统计故障分析，通过随机注入半字节故障，获取故障密文并进行统计学分析，设计余弦距离-汉明重量(COS-HW)和余弦距离-极大似然估计(COS-MLE)两种新型区分器，最少以 184 个故障破译 TweGIFT 算法全部版本的 128 比特主密钥。相较于传统统计故障分析和经典的平方欧氏距离、汉明重量、极大似然估计以及 Wasserstein 距离-极大似然估计区分器，本文提出的中间相错统计故障分析，使故障注入轮数更深一轮，两种新型区分器 COS-HW、COS-MLE，使所需故障数减少 77.77%，成功率达到 99%以上。该结果聚焦于新型唯密文分析，有助于推动轻量级密码算法的进一步研究。

关键词 轻量级密码；TweGIFT；中间相错策略；统计故障分析；唯密文故障分析

中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2025.01696

Research on the Miss-in-the-Middle Statistical Fault Analysis of the TweGIFT Lightweight Cryptosystem

LI Wei[1),2)] LIU Yuan[1)] GU Da-Wu[2)] HUANG Jia-Yin[1)] LU Hai-Ning[3)]

[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)](School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240)

Abstract The TweGIFT lightweight tweakable cryptosystem, proposed by Chakraborti et al. in 2021, aims to protect the data security of Internet of Things devices in smart cities, intelligent manufacturing, and smart agriculture. TweGIFT is the core block cipher for LOTUS/LOCUS and ESTATE, which are candidate algorithms in the International Lightweight Cryptography Competition. Hence, the research on the security of the TweGIFT cipher is significant. The TweGIFT lightweight tweakable cryptosystem adopts a substitution-permutation network and consists of two versions: TweGIFT-64 and TweGIFT-128, whose blocks sizes are 64 and 128 bits, respectively. Its key size is 128 bits, with 28 or 40 rounds of encryption and decryption. It has perfect diffusion and compatibility and can resist many types of attacks, such as differential analysis, Boomerang analysis and collision fault analysis. This paper advances the miss-in-the-middle statistical fault analysis (MSFA) of the TweGIFT cryptosystem. Based on the ciphertext-

收稿日期：2024-11-01；在线发布日期：2025-04-03。本课题得到国家重点研发计划(2020YFA0712300)、国家自然科学基金项目(62472286)、上海市自然科学基金(24ZR1401300)、上海市扬帆计划(23YF1401000)和中央高校基本科研业务基金(223202D-25)资助。**李 玮**，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。**刘 源**，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。**黄佳音**，硕士研究生，主要研究方向为分组密码的故障分析。**陆海宁**(通信作者)，硕士，工程师，主要研究方向为密码工程与网络安全协议。E-mail：hnlu@sjtu.edu.cn。

only attacking assumption and the random nibble-oriented fault model, the MSFA relies on the miss-in-the-middle strategy and statistical fault analysis by injecting random nibble-oriented faults, collecting faulty-affected ciphertexts, and analyzing statistical properties to recover the secret key. This paper designs two novel distinguishers of Cosine Distance-Hamming Weight (COS-HW) and Cosine Distance-Maximum Likelihood Estimation (COS-MLE). The experiments consider the deepest injection rounds, the number of faults, the success rate, and latency to study the performance of the new fault analysis and distinguishers. The deepest injection rounds is the deepest position where the attacker can inject the first fault. In fault analysis, the fault injection of shallow rounds can only interfere with local operations, the influence range is small, and the key information obtained by the attacker is limited. Deeper injection rounds will spread to more critical operations along the path, significantly expanding its propagation range. This diffusion effect enables attackers to collect more intermediate state information related to the key, thus improving attack efficiency. The number of faults is the minimum required for retrieving the secret key with the maximum probability. The smaller the faults, the better the attacking performance and the distinguisher, the success rate refers to the probability of recovering the secret key. When the rate reaches no less than 99%, the attacker can recover the secret key in most cases. Latency is the time required to retrieve the secret key using different distinguishers. The complexity serves as metrics to assess the computational time and total data processed required for recovery the key. In practice, latency and complexity are important indicators for measuring the performance of fault analysis and distinguishers. It only requires 132 and 184 faults to recover the 128-bit secret key of TweGIFT-64 and TweGIFT-128. Compared to traditional statistical fault analysis and classical distinguishers such as SEI, HW, MLE, and WD-MLE, the MSFA method and two distinguishers of COS-HW and COS-MLE, together enable deeper fault injection locations, reduce the number of faults by 77.77% and achieve a success of over 99%. As a result, the miss-in-the-middle statistical fault analysis threatens the security of the TweGIFT lightweight cryptosystem. This study focuses on novel ciphertext-only analysis and contributes to advancing further research on lightweight cryptosystems.

Keywords　lightweight cryptosystem; TweGIFT; miss-in-the-middle strategy; statistical fault analysis; ciphertext-only fault analysis

1　引　言

随着智慧城市、智能制造和智联农业等领域的蓬勃发展，低功耗和高实时性的物联网设备得到广泛应用[1]。这些设备对数据安全性和隐私保护有着极高的要求，然而，在实际运行过程中物联网设备产生的光、电、声、磁等侧信道信息可能会导致机密泄露，构成信息安全隐患。此外，这些领域中的智能终端设备通常面临资源受限的问题，如计算能力和存储空间的限制，要求密码算法必须具备高性能、低功耗和低存储的特点。传统密码算法难以满足这些要求，因此，轻量级密码算法应运而生，并已经成为智慧城市、智能制造和智联农业等领域保障信息安全的重要手段，广泛用于保护信息的保密性、完整性和可认证性，并已成为密码学领域的研究热点[2-6]。

2021年，Chakraborti等学者[7]设计了 TweGIFT 轻量级可调分组密码算法，同时提出以 TweGIFT 算法为核心分组密码的认证加密算法 tHyENA[8]。国际轻量级密码竞赛第二轮候选算法 LOTUS/LOCUS 和 ESTATE 也采用了 TweGIFT 算法作为核心分组密码[9-10]，进一步验证了 TweGIFT 算法的实用性和重要性。TweGIFT 轻量级可调分组密码有两个版本，即 TweGIFT-64 和 TweGIFT-128，均采用代换置换网络(SPN)结构，密钥长度均为 128 比特，分组长度分别为 64 比特和 128 比特，迭代轮数分别为 28 轮和 40 轮，调柄长度为 4 比特，具

有良好的可扩展性和安全性,能够抵御差分分析、Boomerang 分析和碰撞故障分析等[11-13]。

灰盒模型的出现标志着密码算法的安全性评估进入新阶段,传统黑盒模型下的安全性分析不再能全面评估密码算法是否安全[14]。灰盒模型允许攻击者在获取输入和输出外,还能访问密码执行过程中泄露的侧信道信息。侧信道分析是指基于灰盒模型,通过测量和分析加密过程中的侧信道信息,对密码算法进行安全性分析的过程[15]。常见的侧信道分析包括缓存分析、功耗分析和故障分析等。以 GIFT 密码及其衍生算法的侧信道分析为例。2021 年,Reinbrecht 等学者[16]针对 GIFT 算法进行了缓存分析,通过在加密过程中寻找已使用的缓存地址,进一步获取密钥信息,从而恢复完整密钥。2023 年,Mozipo 等学者[17]针对 GIFT-COFB、Ascon、Sparkle、Grain、PHOTON-Beetle、Romulus 和 TinyJambu 算法进行功耗分析,他们使用汉明距离来估计不同算法的泄漏模型中不同中间状态之间的功耗差异。2024 年,Joshi 等学者[18]提出一种半永久型故障分析,并将其应用于 GIFT 和 Elephant 算法。通过该分析方法,攻击者通过降低 GIFT 算法 S 盒的非线性,使其易受到线性密码分析的攻击。

在灰盒模型中,攻击者还具备引入故障的能力,从而进一步发展出故障分析。故障分析主要用于评估密码算法的安全边界。现在已经成为检测算法实现安全性的重要密码分析手段[19-22]。具体地,在密码算法加密运算中,攻击者通过电压故障、激光注入及电磁干扰等手段注入故障,进而引发加密过程中的错误输出,再通过收集和分析这些错误输出,结合对中间状态的分析,攻击者可以恢复密钥。与此同时,不断有学者提出新型的故障分析方法。2018 年,Zhang 等学者[23]提出持续故障分析,并应用于 AES 算法,通过分析注入的故障和持续性故障生成的混合密文进行密钥恢复。2024 年,Beigizad 等学者[24]提出链接故障分析,并用于 AES 和 PRESENT 算法,该方法主要通过使用指令跳转等方式,链接两个中间状态以获取密钥信息,从而恢复主密钥。2025 年,Gao 等学者[25]根据 GIFT 算法轮函数的扩散特性,提出了两种基于字节的差分故障分析模型,故障注入位置为倒数第二、三轮,分别仅需 79 和 16 个故障即可恢复密钥。

在密码分析中,不同攻击方法实施难度不同,对攻击者的能力要求也不同。因此,根据不同基本假设,通常将密码分析方法分为选择密文攻击、选择明文攻击、已知明文攻击和唯密文攻击等。其中,唯密文攻击假设攻击者只能通过仅有的密文信息进行密钥恢复,对于攻击者能力及实际环境要求最弱。统计故障分析(Statistical Fault Analysis,SFA)就是基于唯密文基本假设,由 Fuhr 等学者[26]于 2013 年首次提出,并将该方法应用于 AES 算法,针对最后四轮成功恢复密钥。此后,统计故障分析作为一种重要密码分析方法,被用于轻量级认证加密算法、流密码以及后量子密码[27-32]。在统计故障分析中,攻击者基于唯密文基本假设,通过获取故障密文进行部分中间状态推导,并结合统计学原理来恢复部分密钥,再使用密钥编排算法恢复全部密钥。这种方法可以迅速且有效判断密码是否安全。因此,唯密文故障分析广泛用于评估轻量级密码算法的安全性,保障用户的数据安全。

目前,针对 TweGIFT 轻量级可调分组密码,还未有公开成果讨论其抵御新型唯密文故障分析的能力。本文针对 TweGIFT 可调密码的结构与特性,提出了中间相错统计故障分析(Miss-in-the-Middle Statistical Fault Analysis,MSFA)方法。本方法采用唯密文攻击基本假设,结合中间相错策略与统计学原理的优势,相较于传统统计故障分析方法,对 TweGIFT 密码算法进行更深一轮的故障注入和安全性分析,分析效果更有效。同时,本文基于余弦距离原理设计了新型组合区分器,余弦距离-汉明重量(COS-HW)和余弦距离-极大似然估计(COS-MLE),有效提高了破译效率并降低了实现代价。本文实验结果聚焦于新型唯密文分析,有助于推动轻量级密码算法的进一步研究。

本文第 2 节概述 TweGIFT 算法已发表安全分析及统计故障分析的相关研究成果;第 3 节介绍 TweGIFT 算法的基本知识;第 4 节详述中间相错统计故障分析,介绍经典和新型区分器;第 5 节分析实验结果,并与统计故障分析结果进行对比;第 6 节总结全文。

2 相关工作

自 TweGIFT 算法提出以来,其安全性引起国内外学者的广泛关注。自 2019 年以来,Chakraborti 等人[9-10]对其进行差分分析,结果显示 TweGIFT-64 和 TweGIFT-128 的微分特征概率分别可以在第 8 轮和第 10 轮达到 2^{-64} 和 2^{-76}。2021 年,Zhang 等学者[11]在基于深度神经网络构建神经网络差分区分

器，并利用固定差分值训练 4 至 8 轮区分器，成功恢复 128 比特主密钥[11]。同年，Liu 等学者[13]使用碰撞故障分析方法对 TweGIFT 算法进行安全性分析，64 个故障密文即可恢复 128 比特主密钥[13]。2023 年，Chakraborti 等学者[9-10]对其进行 Boomerang 分析，找到两条概率为 1 的 5 轮路径，结果证明与 GIFT 算法相比，针对 TweGIFT 算法的 10 轮 Boomerang 分析更有效[12]。表 1 展示了该算法所有版本现有的安全性分析比较。

表 1　针对 TweGIFT-64 和 TweGIFT-128 的安全性分析

分析类型	基本假设	攻击轮数	最深注入轮数	文献
差分分析	已知明文攻击	8/10	—	[9-11]
Boomerang 分析	已知明文攻击	10/10	—	[12]
碰撞故障分析	已知明文攻击	—/40(全轮)	—/第 2 轮	[13]
统计故障分析	唯密文攻击	28/40(全轮)	第 27/39 轮	本文
中间相错统计故障分析	唯密文攻击	28/40(全轮)	第 26/38 轮	本文

统计故障分析是重要唯密文分析方法之一，最早由 Fuhr 等学者[26]提出，该方法在唯密文基本假设下，利用注入随机故障产生的错误密文，推导部分中间状态，结合统计学思想恢复部分主密钥，最后通过密钥编排算法恢复完整密钥。2017 年，Nozaki 等学者[27]使用统计故障分析，先后在 Midori 算法第 15 轮和第 14 轮利用时钟毛刺注入故障，成功破译 Midori 算法[27]。2019 年，Ramezanpour 等学者[28]提出统计无效故障分析方法，针对 Ascon 算法使用双重故障注入模型，通过修改密钥分组的划分长度，可以调节双重故障的注入次数和密钥的搜索空间。2024 年，Li 等学者[33]基于统计故障分析，结合不可能和代数思想，提出不可能统计故障分析和基于代数的统计故障分析等多种新型唯密文故障分析方法，并针对 SUNDAE-GIFT 和 Pyjamask 等轻量级认证加密算法和分组密码进行安全性分析。

本文聚焦于 TweGIFT 可调算法的结构特性，提出中间相错统计故障分析方法，结合中间相错策略与统计故障分析各自的优势，拓展了统计故障分析方法的种类，对 TweGIFT 算法的实现安全性进行评估。同时，本文还使用传统统计故障分析方法进行对比分析，选取最深注入轮数、故障数以及成功率作为评估指标。其中，最深注入轮数代表了攻击者可以注入首个故障的最深位置。在故障分析中，浅层轮数的故障注入往往只能干扰局部运算，影响范围较小，攻击者获取的密钥信息也有限。较深轮数的故障会沿着路径扩散至更多关键运算，显著扩大其传播范围。这种扩散效果通常使攻击者能够收集到更多与密钥相关的中间状态信息，从而提升攻击效率。因此，为最大化攻击效果，攻击者应尽量在较深的轮数注入故障。表 2 详细列出了统计故障分析和中间相错统计故障分析对 TweGIFT 各版本的安全性分析结果。相较于传统的统计故障分析，以及现有的平方欧氏距离(SEI)、汉明重量(HW)、极大似然估计值(MLE)和 Wasserstein 距离-极大似然估计(WD-MLE)区分器，本文提出的中间相错统计故障分析，结合余弦距离-汉明重量(COS-HW)、余弦距离-极大似然估计(COS-MLE)两种新型组合区分器，故障注入轮数实现更深一轮突破，同时所需故障数减少 77.77%，性能表现有较大提升。

表 2　统计故障分析和中间相错统计故障分析分别破译 TweGIFT 所有版本主密钥的结果

区分器	TweGIFT-64			TweGIFT-128		
	最深注入轮数	故障数	成功率/%	最深注入轮数	故障数	成功率/%
平方欧氏距离(SEI)[26]	27/26	656/280	≥99	39/38	1600/544	≥99
汉明重量(HW)[26]	27/26	352/184	≥99	39/38	1120/392	≥99
极大似然估计(MLE)[26]	27/26	368/200	≥99	39/38	1056/376	≥99
Wasserstein 距离-极大似然估计(WD-MLE)[33]	27/26	320/176	≥99	39/38	928/336	≥99
余弦距离-极大似然估计(COS-MLE)	27/26	288/148	≥99	39/38	864/192	≥99
余弦距离-汉明重量(COS-HW)	27/26	272/132	≥99	39/38	768/184	≥99

3　TweGIFT 密码

3.1　符号说明

设 Z_2^e 表示 e 比特的二进制向量集；

记 $X\in(Z_2^4)^j$ 和 $Y\in(Z_2^4)^j$，分别表示不同长度的明文和密文，其中 $j\in\{16,32\}$；

记 $K\in(Z_2^{16})^8$ 表示长度 128 比特的主密钥，其中 $K=sk_7\|sk_6\|sk_5\|sk_4\|sk_3\|sk_2\|sk_1\|sk_0$；

记 r 表示迭代轮数，$r\in\{28,40\}$；

记 $RK_i\in(Z_2^2)^j$ 表示第 $i+1$ 轮的轮密钥，其中 $RK_i=rk_i^0rk_i^1\cdots rk_i^{2\cdot j-1}$，$i\in[0,r-1]$，$j\in\{16,32\}$；

记 $T\in Z_2^4$ 和 $RT_i\in Z_2^4$ 分别表示初始调柄和第 $i+1$ 轮的轮调柄，其中 $i\in[0,r-1]$；

记 C 和RC_i分别表示 7 比特初始常量和第 $i+1$ 轮的轮常量,其中 $i\in[0,r-1]$;

记SC、PB、ARC、ARK和ART分别表示 S 盒替换、比特置换、轮常数加、轮密钥加和轮调柄加,SC^{-1}、PB^{-1}、ARC^{-1}、ARK^{-1}和ART^{-1}分别表示其逆运算;

记GRC、GRK和GRT分别表示轮常量生成算法、轮密钥生成算法和轮调柄生成算法;

记 A_i表示在第 1 个半字节注入故障,经第 $i+1$ 轮 ARK 运算后中间状态的 16 比特,$A_i=(a_i^0a_i^8a_i^{16}a_i^{24}a_i^{32}a_i^{41}a_i^{49}a_i^{57}a_i^{66}a_i^{74}a_i^{82}a_i^{90}a_i^{99}a_i^{107}a_i^{115}a_i^{123})$,$a_i^t$表示相同位置的第 t 比特中间状态,其中 $i\in[0,r-1]$,$t\in[0,4\cdot j]$,$j\in\{16,32\}$;

记$\oplus$、$\|$、$\ggg$、Σ、Π、$\%$和$\langle\cdot\rangle$分别表示按比特异或、连接、循环右移、连加、连乘、取余和距离,$\sim$为受故障影响后的值,0b 表示二进制。

3.2 TweGIFT 算法描述

TweGIFT 算法是一种 SPN 结构的轻量级可调分组密码算法,2021 年由 Chakraborti 等学者[7]提出,算法共有两个版本:TweGIFT-64 和 TweGIFT-128,分组长度分别为 64 比特和 128 比特。表 3 展示了 TweGIFT 算法各版本信息。

表 3 各版本 TweGIFT 算法简介

版本	分组长度/比特	密钥长度/比特	迭代轮数/轮
TweGIFT-64	64	128	28
TweGIFT-128	128	128	40

以 128 比特版本为例,TweGIFT 算法的轮变换主要包含以下五个部分,具体为:

(1)S 盒替换(SC)。对中间状态每 4 比特进行非线性变换,S 盒如表 4 所示。

表 4 TweGIFT 的 S 盒

x	$S(x)$
0	1
1	a
2	4
3	c
4	6
5	f
6	3
7	9
8	2
9	d
a	b
b	7
c	5
d	0
e	8
f	e

(2)比特置换(PB)。对中间状态每比特进行线性排列。

(3)轮常数加(ARC)。中间状态的第 3、7、11、15、19、23 和第 127 个比特与轮常数进行异或运算。

(4)轮密钥加(ARK)。第 $i+1$ 轮的轮密钥取 64 比特与中间状态的 d_i^{4m+1} 和 d_i^{4m+2} 比特进行异或运算,其中 $m\in[0,31]$,$i\in[0,39]$。

(5)轮调柄加(ART)。将 4 比特调柄扩展到 32 比特,当轮数 $i\in\{0,5,10,15,20,25,30,35\}$时,轮调柄与中间状态进行异或运算。

TweGIFT-128 加密过程、密钥扩展算法如算法 1、算法 2 所示。算法结构如图 1 所示。

算法 1. TweGIFT-128 的加密算法

输入:明文 X,密钥 K,调柄 T

输出:密文 Y

1. $C=0000000$;
2. $STATE=X$;
3. FOR $i=0$ TO 39 DO
4. $RC_i=GRC(C)$;
5. $RK_i=GRK(K)$;
6. $RT_i=GRT(T)$;
7. $STATE=SC(STATE)$;
8. $STATE=PB(STATE)$;
9. $STATE=ARC(STATE,RC_i)$;
10. $STATE=ARK(STATE,RK_i)$;
11. IF $(i+1)\%5=0$ AND $i<39$ THEN
12. $STATE=ART(STATE,RT_i)$;
13. END IF
14. END FOR
15. $Y=STATE$;
16. RETURN Y.

算法 2. TweGIFT-128 的密钥扩展算法

输入:主密钥 K

输出:轮密钥$RK_0,RK_1,\cdots,RK_{38},RK_{39}$

1. $K=sk_7\|sk_6\|sk_5\|sk_4\|sk_3\|sk_2\|sk_1\|sk_0$;
2. FOR $i=0$ TO 39 DO
3. $L_i=sk_5\|sk_4$;
4. $R_i=sk_1\|sk_0$;
5. $RK_i=L_i\|R_i$;
6. $sk_7\|\cdots\|sk_0=sk_1\ggg2\|sk_0\ggg12\|\cdots\|sk_2$;
7. END FOR
8. RETURN $RK_0,RK_1,\cdots,RK_{38},RK_{39}$.

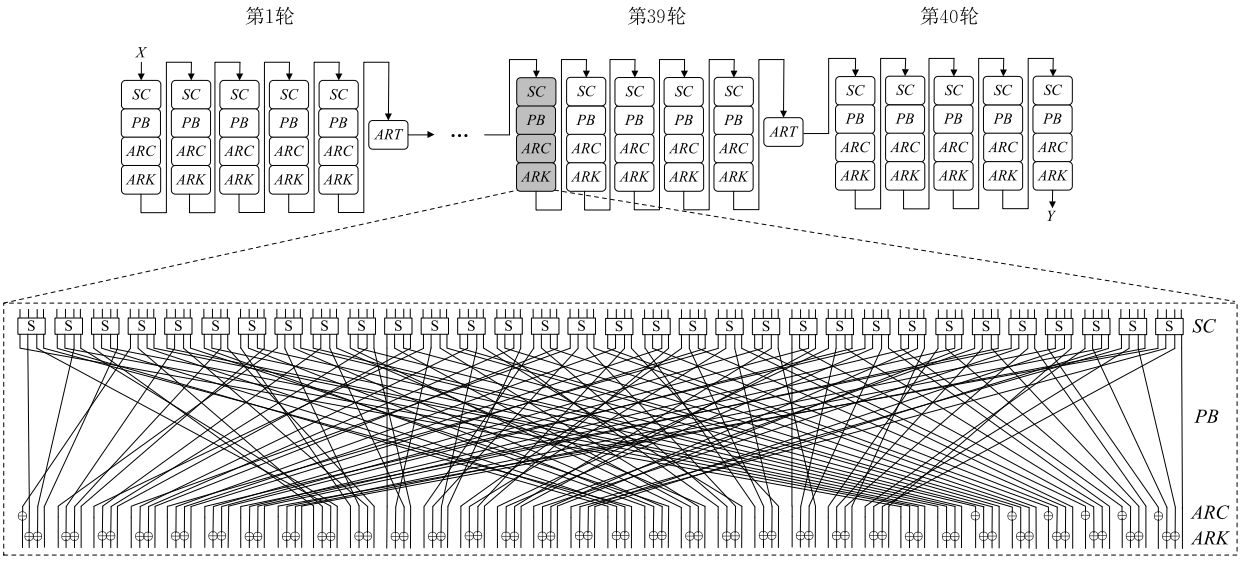

图 1　TweGIFT-128 算法的结构图

4　TweGIFT 的中间相错统计故障分析

4.1　基本假设和故障模型

本文采取唯密文基本假设。即攻击者无法获取明文和密钥，只能基于密文进行密码分析。唯密文攻击假设是密码学中最弱的攻击场景，此时攻击者已知信息最少，分析能力最弱。若密码算法在这种场景下能被破解，则在更强的已知或选择明文等攻击场景下必然不安全。在实际场景中，通常采取脉冲注入器、时钟干扰器、激光二极管及电磁干扰器等工具在密码加密过程中进行故障注入。具体过程为：攻击者采用随机半字节故障模型，使用同一密钥加密多个不同明文，通过按位“与”操作将故障和中间状态进行运算，完成完整加密过程，获取导入故障后的密文。

在无故障注入情况下，各比特位为“0”和“1”的概率均为 50%，因此正确半字节中间状态应服从均匀分布，每个可能值出现的概率为 6.25%。在注入半字节随机故障情况下，中间状态与随机故障按位“与”操作后，各比特位为“0”的概率增加至 75%，为“1”的概率减少至 25%。因此，中间状态服从非均匀分布，半字节中间状态的分布律服从：

$$P_t=\left(\frac{3}{4}\right)^{m-hw(t)}\cdot\left(\frac{1}{4}\right)^{hw(t)}=\frac{3^{m-hw(t)}}{4^m},$$

其中，t 表示半字节中间状态的所有可能的取值情况，P_t 表示半字节中间状态取值为 t 的理论概率，$hw(t)$ 表示 t 的汉明重量，m 表示注入故障的比特数。在本文中，$t\in[0,15]$，$hw(t)\in[0,4]$，$m=4$。根据半字节按位“与”故障模型，在注入故障后，任何中间状态与 0b1111“与”操作后，都不会发生改变，因此半字节中间状态的分布律服从：

$$P_t=\frac{3^{m-hw(t)}-1}{4^m-2^m},$$

无故障注入及受故障影响后的半字节中间状态累加分布律如图 2 所示。

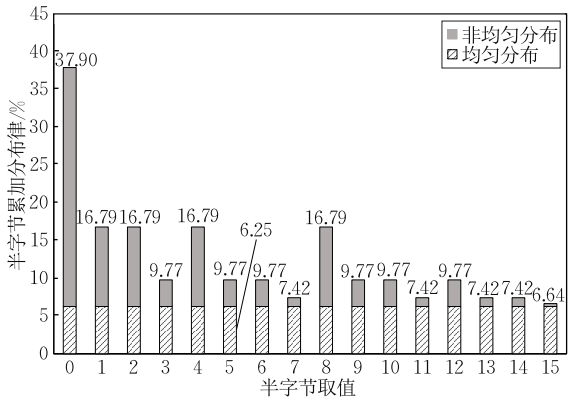

图 2　半字节中间状态的累加分布律

4.2　中间相错分析策略

中间相错分析策略由 Biham 等学者[34-35]于 1999 年首次提出，起初用于 Skipjack、IDEA 和 Khufu 密码算法的安全性分析，成功恢复完整密钥，并有效减少攻击覆盖轮数。2019 年，Sadegh 等学者[36]研究了 SIMECK 分组密码的零相关性和不可能差分分析，基于改进的中间相错策略找到新的零相关线性区分器。2021 年，Xie 等学者[37]将中间相错攻击应用于量子密码，并提出一种用于查找一般分组密码

不可能差分的量子算法。同年，Wang 等学者[38]利用中间相遇策略改进类 SIMON 分组密码的不可能差分特性，对于 SIMON 算法的所有版本，分别给出 11、12、13、16 和 19 轮不可能子空间轨迹，对于 SIMECK 算法所有版本，分别给出的 15 和 17 轮不可能子空间轨迹，极大程度降低了破译类 SIMON 分组密码的时间复杂度。

在经典中间相错分析中，攻击者利用中间状态的不匹配来证明某些输入输出发生的概率为 0，从而构造不可能差分攻击的区分器来恢复密码算法的主密钥，但是构造不可能事件所需的时间开销很大。在传统的统计故障分析中，所需故障数较多，易受到硬件设备的限制。为解决时间开销大和故障数多这两个问题，本文将中间相错策略和统计故障分析的优势相结合，加深了故障注入轮数，降低了时间开销，减少了所需故障数，并设计新型双重区分器，进一步降低了故障数和时间开销。

本文提出的中间相错统计故障分析，通过利用概率为零的中间相错路径，结合故障影响后产生的不均匀分布，来进行密钥筛选。该分析的关键步骤主要包括构造中间相错路径和筛选密钥。利用中间相错路径，进一步缩小候选密钥空间，通过中间状态的不均匀分布性质，最终恢复主密钥。以在 A_{37} 第一个半字节注入随机故障为例，攻击者获取故障密文后，经过倒推分析得到中间状态 $(\tilde{a}_{37}^{0}\tilde{a}_{37}^{1}\tilde{a}_{37}^{2}\tilde{a}_{37}^{3})$，存在中间相错路径：

$$(\tilde{a}_{37}^{0}\tilde{a}_{37}^{1}\tilde{a}_{37}^{2}\tilde{a}_{37}^{3})\oplus 0\mathrm{b}1111\neq 0,$$

利用该路径缩小密钥搜索空间，提高攻击效率。

4.3 攻击过程

以 TweGIFT-128 为实例，本文提出的中间相错统计故障分析，其核心攻击过程分为四个步骤：

步骤 1. 随机故障注入。在 TweGIFT 算法加密过程进行至倒数第三轮时，攻击者采用按位“与”操作选择一个半字节注入随机半字节故障，完成加密操作后，生成故障密文 $\tilde{Y}$。图 3 展示了在倒数第三轮第一个半字节注入故障的扩散路径。

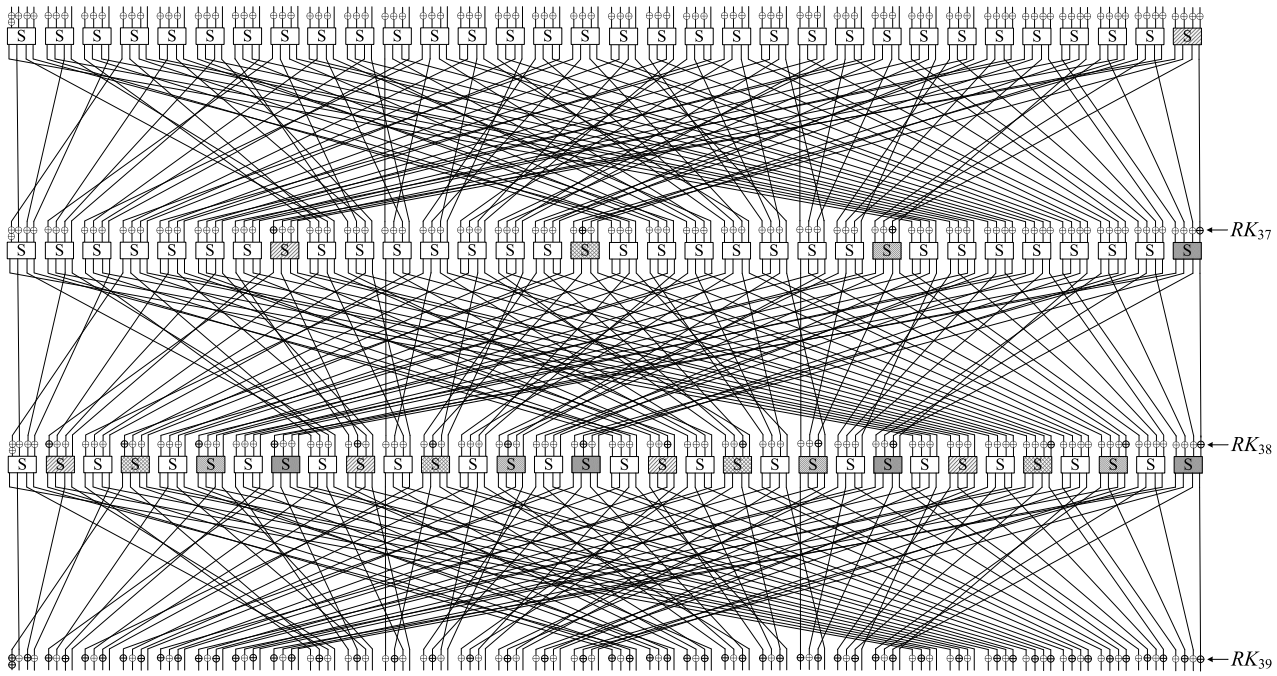

图 3 TweGIFT-128 倒数第三轮注入半字节故障的扩散路径

步骤 2. 中间相错分析。以在倒数第三轮第一个半字节注入随机故障为例。首先，攻击者先穷举倒数第二轮经轮密钥加运算后中间状态的 16 比特 $A_{38}=(a_{38}^{0}a_{38}^{8}a_{38}^{16}a_{38}^{24}a_{38}^{32}a_{38}^{41}a_{38}^{49}a_{38}^{57}a_{38}^{66}a_{38}^{74}a_{38}^{82}a_{38}^{90}a_{38}^{99}a_{38}^{107}a_{38}^{115}a_{38}^{123})$，再穷举 RK_{37} 的 2 比特和 RK_{38} 的 8 比特共 10 比特。然后，攻击者依据算法解密过程，计算出使用 16 比特中间状态 A_{38} 和 10 比特轮密钥，倒推至倒数第三轮经轮密钥加运算后的 4 比特中间状态 $(a_{37}^{0}a_{37}^{1}a_{37}^{2}a_{37}^{3})$ 的对应路径，共 $2^{26}(=2^{16}\cdot 2^{10})$ 组映射结果，并存储该映射矩阵：

$$(a_{37}^{0}a_{37}^{1}a_{37}^{2}a_{37}^{3})=SC^{-1}(PB^{-1}((rk_{37}^{8}rk_{37}^{48})\oplus SC^{-1}(PB^{-1}((rk_{38}^{8}rk_{38}^{10}rk_{38}^{12}rk_{38}^{14}rk_{38}^{48}rk_{38}^{50}rk_{38}^{52}rk_{38}^{54})\oplus (a_{38}^{0}a_{38}^{8}a_{38}^{16}a_{38}^{24}a_{38}^{32}a_{38}^{41}a_{38}^{49}a_{38}^{57}a_{38}^{66}a_{38}^{74}a_{38}^{82}a_{38}^{90}a_{38}^{99}a_{38}^{107}a_{38}^{115}a_{38}^{123})))))。$$

接下来，攻击者使用步骤 1 获得的故障密文 $\tilde{Y}$，枚举 RK_{39} 的 32 比特，逆推得到倒数第二轮的 16 比特中间状态：

$$\widetilde{A}_{38}=SC^{-1}(PB^{-1}(RK_{39}\oplus\widetilde{Y}))。$$

攻击者将逆推得到的倒数第二轮中间状态 $\widetilde{A}_{38}$ 与存储在映射矩阵中的 A_{38} 相匹配。若匹配一致，则取矩阵中存储的 $(a_{37}^0\,a_{37}^1\,a_{37}^2\,a_{37}^3)$，作为倒数第三轮的中间状态候选值，同时选取所对应的 RK_{37} 的 2 比特、RK_{38} 的 8 比特以及 RK_{39} 中的 32 比特作为候选密钥值。值得注意的是，由故障模型可知，倒数第三轮的中间状态候选值不可能为 0b1111，因此产生相错路径：

$$(\tilde{a}_{37}^0\tilde{a}_{37}^1\tilde{a}_{37}^2\tilde{a}_{37}^3)\oplus 0\text{b}1111\neq 0,$$

利用此条相错路径可以提前筛除部分错误候选密钥值，缩小候选密钥空间，提高分析效率。

步骤 3. 区分器筛选。攻击者可以在 4.4 节介绍的六种区分器中选择合适的进行候选密钥筛选。具体地，利用步骤 2 中所获得的中间状态候选值 $(a_{37}^0\,a_{37}^1\,a_{37}^2\,a_{37}^3)$，结合不同区分器的数学特征，可以计算出对应的统计值，选择最值相应的密钥值，即为正确密钥。

步骤 4. 原始密钥恢复。以在倒数第三轮第一个半字节注入随机故障为例，根据故障传播路径，可以恢复倒数第三轮的 rk_{37}^8 和 rk_{37}^{48}，倒数第二轮的 rk_{38}^8、rk_{38}^{10} 和 rk_{38}^{12} 等，倒数第一轮的 rk_{39}^0、rk_{39}^1 和 rk_{39}^2 等，共 42 比特。攻击者更改故障注入的位置，并重复进行步骤 1～3，直到获得 RK_{38} 和 RK_{39} 的全部 128 比特，再使用主密钥恢复算法即可恢复主密钥 K。算法 3 展示了主密钥的具体求解过程。TweGIFT-128 的主密钥 K 为 128 比特，轮密钥为 64 比特。根据算法 3 可知，恢复主密钥 K，需要轮密钥 RK_{38} 和 RK_{39}，共计 128(=2·64)比特。每注入半字节故障时，可恢复 RK_{38} 中的 8 比特和 RK_{39} 中的 32 比特。因此，分别在编号 0,1,2,…,31 的半字节注入故障时，注入故障位置与可恢复密钥位置共有 8(=64/8)种对应关系。图 4 展示了上述 8 种对应关系，覆盖恢复主密钥 K 所需 RK_{38} 和 RK_{39} 的全部比特。

算法 3.　TweGIFT-128 主密钥恢复算法

输入：轮密钥 RK_{39}、RK_{38}

输出：主密钥 K

1. $sk_3\,\|\,sk_2\,\|\,sk_7\,\|\,sk_6=RK_{38}$；
2. $sk_3=sk_3\ggg 4$；
3. $sk_2=sk_2\ggg 8$；
4. $sk_7=sk_7\ggg 2$；
5. $sk_6=sk_6\ggg 12$；
6. $sk_5\,\|\,sk_4\,\|\,sk_1\,\|\,sk_0=RK_{39}$；
7. $sk_5=sk_5\ggg 4$；
8. $sk_4=sk_4\ggg 8$；
9. $sk_1=sk_1\ggg 4$；
10. $sk_0=sk_0\ggg 8$；
11. $K=sk_7\,\|\,sk_6\,\|\,sk_5\,\|\,sk_4\,\|\,sk_3\,\|\,sk_2\,\|\,sk_1\,\|\,sk_0$；
12. RETURN K.

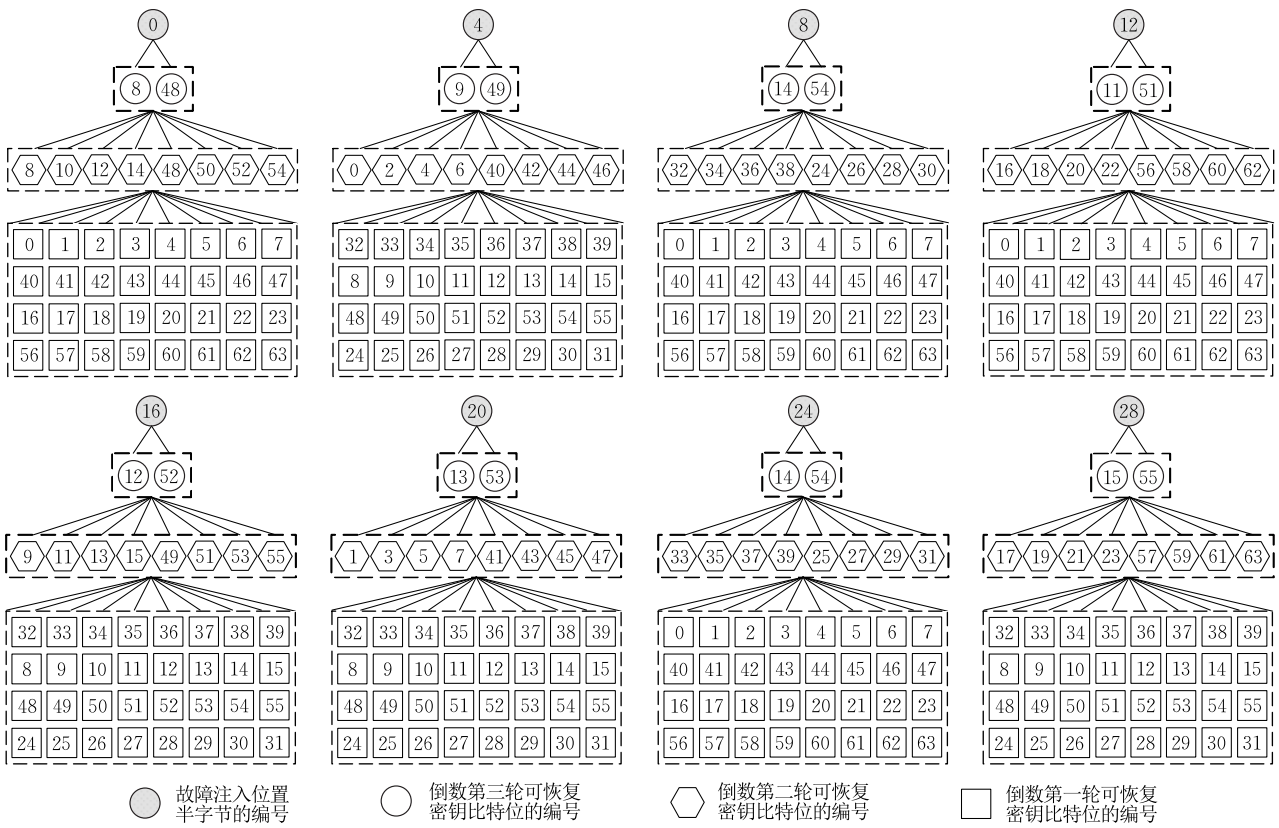

图 4　注入故障位置与可恢复密钥位置的对应关系

4.4 区分器

2013 年,Fuhr 等学者[26]分别对 AES 算法第 6～9 轮进行唯密文分析,并使用汉明重量、极大似然估计和平方欧氏距离作为统计学分析工具,成功恢复全部密钥。本文在极大似然估计和汉明重量经典区分器的基础上,进一步引入余弦距离的统计学概念,设计了两种新型双重区分器:余弦距离-极大似然估计和余弦距离-汉明重量。

4.4.1 经典区分器

(1)平方欧氏距离(Square Euclidean Imbalance,SEI)

为度量空间中两点之间的距离,古希腊几何学家欧几里得提出欧氏距离。平方欧氏距离在欧氏距离基础上对其平方,用于衡量向量间的相似性和差异性,广泛应用于密码学、模式识别和机器学习中。Fuhr 等学者[26]通过比较中间状态值的实际概率和理论概率,计算它们的平方欧氏距离,并依据最大值来筛选密钥,从而确定正确候选密钥值。平方欧氏距离区分器的表达式为

$$\mathrm{SEI}=\sum_{\delta=0}^{\omega-1}\left(\frac{n(\delta)}{f_n}-\frac{1}{\omega}\right)^2,$$

其中,f_n表示故障数,ω 表示 4 比特所有可能取值的个数,δ 表示受故障影响后的中间状态值,$n(\delta)$表示出现中间状态 δ 的次数,由本文所使用的故障模型易知,$\delta\in[0,15]$。

(2)汉明重量(Hamming Weight,HW)

1954 年,Reed[39]提出汉明重量的概念,起初应用于纠错码理论。现在汉明重量也广泛用于密码学、信息论和编码理论中,用于评估信息块的变化和相似性。在故障分析中,Fuhr 等学者[26]计算 AES 算法中间状态的汉明重量,选最小汉明重量对应的密钥作为候选密钥值。汉明重量区分器的表达式为

$$\mathrm{HW}=\sum_{\alpha=1}^{f_n}hw(\delta_\alpha),$$

其中,f_n表示故障数,α 表示注入故障数量,δ_α表示注入第 α 个故障后对应的中间状态值,$hw(\delta_\alpha)$表示中间状态值 δ_α的汉明重量,由半字节故障模型,$\delta_\alpha\in[0,15]$。

(3)极大似然估计(Maximum Likelihood Estimation,MLE)

极大似然估计作为一种参数估计方法,最早由 Gauss 等学者于 1821 年提出,随后由 Fisher 等学者[40]补充并发展,目前广泛用于统计学、机器学习和信号处理。极大似然估计首次被应用于分组密码的唯密文故障分析,攻击者通过选出最大似然估计值,得出候选密钥[26]。极大似然估计区分器的表达式为

$$\mathrm{MLE}=\prod_{\delta=0}^{15}(P_\delta)^{Q_\delta\cdot f_n},$$

其中,f_n表示故障数,P_δ和 Q_δ分别为中间状态值 δ 出现的理论概率和实际概率,由半字节故障模型,$\delta\in[0,15]$。

(4)Wasserstein 距离-极大似然估计(Wasserstein Distance-Maximum Likelihood Estimation,WD-MLE)

源于最优传输理论的 Wasserstein 距离,广泛应用于图像检索、计算机视觉、机器学习和金融学等应用领域,来衡量两个概率分布的相似性[41]。2024 年,Li 等学者[33]提出一种 Wasserstein 距离-极大似然估计新型区分器,用于 Pyjamask 密码的不可能统计故障分析,并成功恢复全部 128 比特主密钥[33]。该区分器结合 WD 和 MLE 的优点,以 Wasserstein 距离为基础,计算中间状态值出现的理论概率和实际概率间的相似性:

$$\mathrm{WD}=\inf_{\gamma\in\Gamma(\xi_\delta,\theta_\delta)}E_{(P_\delta\cdot f_n,Q_\delta\cdot f_n)\in\gamma}\langle P_\delta\cdot f_n-Q_\delta\cdot f\rangle,$$

其中,f_n表示故障数,δ 表示注入故障后中间状态值,P_δ和 Q_δ分别为中间状态 δ 出现的理论概率和实际概率,ξ_δ和 θ_δ分别表示中间状态 δ 不导入故障的理论分布和导入故障后的实际分布,$\Gamma(\xi_\delta,\theta_\delta)$表示 ξ_δ和 θ_δ所有联合分布的集合,γ 表示属于 $\Gamma(\xi_\delta,\theta_\delta)$集合的分布,$E(\cdot)$表示数学期望,$\langle\cdot\rangle$表示距离,inf 表示取下界,筛选出 Wasserstein 距离较小的密钥值,再进一步使用极大似然估计,进一步筛选候选密钥,缩小密钥空间。

4.4.2 新型区分器

余弦距离又称余弦相似度,这一概念起源于空间向量模型,最早由信息检索领域提出,被广泛应用于文本挖掘、自然语言处理、推荐系统和聚类分析等领域[42]。余弦距离的表达式为

$$\mathrm{COS}=\frac{\sum_{\delta=0}^{\omega-1}\xi_\delta\theta_\delta}{\sqrt{\sum_{\delta=0}^{\omega-1}\xi_\delta^2\sum_{\delta=0}^{\omega-1}\theta_\delta^2}},$$

其中,ω 表示 4 比特所有可能取值的个数,δ 表示受故障影响后的中间状态值,ξ_δ和 θ_δ分别表示中间状态 δ 不导入故障的理论分布和导入故障后的实际分布。ξ_δ和 θ_δ的夹角及其余弦值,用于表示量分布的相似性。夹角越小,越趋于 0°,余弦值越接近 1,实际分布越吻合理论分布;反之,夹角越大,越趋于

180°，余弦值越接近−1，实际分布越偏离理论分布，因此 COS 值最接近 1 时，中间状态对应密钥即为候选密钥值。

(1) 余弦距离-极大似然估计(Cosine Distance-Maximum Likelihood Estimation，COS-MLE)

COS-MLE 区分器在 MLE 区分器的基础上，结合余弦距离，继承 MLE 区分器和 COS 区分器二者的优点，首先对于导入故障后的中间状态值及其分布律，攻击者计算实际分布与理论分布的余弦距离，并筛选出最大余弦距离对应的密钥作为第一轮候选密钥。然后对初筛结果计算其极大似然估计值，具有最大极大似然估计值的密钥即为最终候选密钥值，极大似然估计区分器表达式见第 4.4.1 节。

(2) 余弦距离-汉明重量(Cosine Distance-Hamming Weight，COS-HW)

COS-HW 区分器在 HW 区分器的基础上，结合余弦距离，继承 HW 区分器和 COS 区分器二者的优点。首先对于导入故障后的中间状态值及其分布律，攻击者计算实际分布与理论分布的余弦距离，并筛选出最大余弦距离对应的密钥作为第一轮候选密钥。然后计算初筛结果的汉明重量，选择具有最小汉明重量的密钥作为最终候选密钥值。汉明重量区分器表达式见第 4.4.1 节。

表 5 列出了本文使用的各区分器的基本信息。

表 5　各区分器的取值范围和筛选过程

区分器	取值范围	筛选过程
SEI[26]	最大值	评估中间状态的平方欧氏距离，筛选最大值对应的候选密钥
HW[26]	最小值	评估中间状态的汉明重量，筛选最小值对应的候选密钥
MLE[26]	最大值	评估中间状态的极大似然估计值，筛选最大值对应的候选密钥
WD-MLE[33]	WD最小值 MLE最大值	先选出 Wasserstein 距离最小值对应的部分候选密钥，再筛选极大似然估计值最大值对应的候选密钥
COS-MLE	COS最大值 MLE最大值	先选出余弦距离最大值对应的部分候选密钥，再筛选极大似然估计值最大值对应的候选密钥
COS-HW	COS最大值 HW最小值	先选出余弦距离最大值对应的部分候选密钥，再筛选汉明重量最小值对应的候选密钥

5　TweGIFT 算法的实验分析

本文通过计算机模拟编程实现 TweGIFT 算法的统计故障分析和中间相错统计故障分析，使用 PC 端设备配置为 Intel(R) Core(TM) i7-8565U CPU，内存为 8 GB，使用 C 语言模拟注入故障、预计算状态数组、推导中间状态、区分器筛选密钥、恢复主密钥等过程。进行 10000 次实验，并以 TweGIFT-128 为例分析实验结果，分析指标包括故障数、成功率、耗时及复杂度。实验数据见附表 A1～A4。

5.1　故障数

故障数是评估密码算法抗攻击能力的重要指标，指攻击者破译密码主密钥需注入的故障总数，也代表破译密码所需代价。故障数越少，破译密码所需代价越低，在实际情况中越容易实施攻击。在本文中，故障数特指，运用不同区分器，通过统计故障分析和中间相错统计故障分析，以超过 99%的概率有效恢复 TweGIFT 算法主密钥的最少注入数量。如表 2 所示，经典统计故障分析在使用 SEI、HW、MLE、WD-MLE、COS-MLE 和 COS-HW 区分器时，所需故障数分别为 1600、1120、1056、928、864 和 768 个。而中间相错统计故障分析在使用 SEI、HW、MLE、WD-MLE、COS-MLE 和 COS-HW 区分器时，所需故障数分别为 544、392、376、336、192 和 184 个。故障注入轮数加深一轮，故障数分别减少 66.00%、65.00%、64.39%、63.79%、77.77%及 76.04%。易知，中间相错统计故障分析使用各区分器的所需故障数均小于统计故障分析，且新型区分器均优于经典区分器。图 5 展示了两种方法下使用不同区分器破译 TweGIFT-128 的所需故障数，两种方法在使用新型区分器时最少以 768 和 184 个故障就能筛选出 128 比特主密钥，说明本文提出的新区分器在有效降低攻击代价方面表现优异。

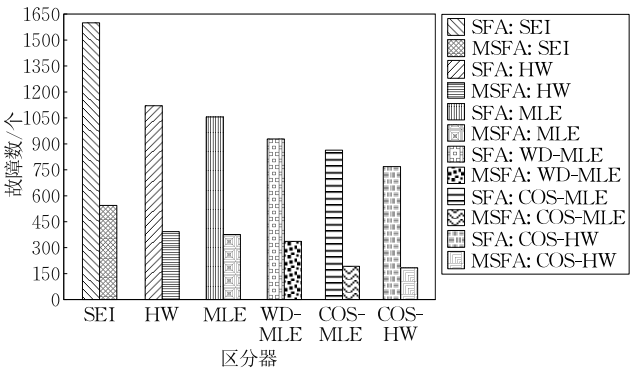

图 5　各区分器恢复 TweGIFT-128 主密钥的所需故障数

当故障注入轮数均为第 39 轮时，经典统计故障分析在使用 SEI、HW、MLE、WD-MLE、COS-MLE 和 COS-HW 区分器时，所需故障数分别为 1600、1120、1056、928、864 和 768 个。而中间相错统计故障分析在使用 SEI、HW、MLE、WD-MLE、COS-MLE 和 COS-HW 区分器时，所需故障数分别为 1536、960、992、864、832 和 736 个。实验数据表明，

在相同故障注入轮数下，中间相错统计故障分析方法在故障数指标上优于传统统计故障分析。

5.2 成功率

成功率指注入指定数量故障后，攻击者在一定攻击次数或攻击时间内成功破译主密钥的概率。图 6 和图 7 展示了在注入指定数量故障时，统计故障分析和中间相错统计故障分析使用不同区分器破译主密钥的成功率。图中，横轴表示不同故障数，纵轴表示有效破译算法主密钥的概率，不同标记的折线展示了使用不同区分器破译主密钥时的成功率。实验结果表示 SEI、HW、MLE、WD-MLE、COS-MLE 和 COS-HW 六种区分器成功率均可达到 99%以上。特别地，在保证成功率的前提下，新型区分器 COS-MLE 和 COS-HW 使用故障数明显少于经典区分器，效果更佳。

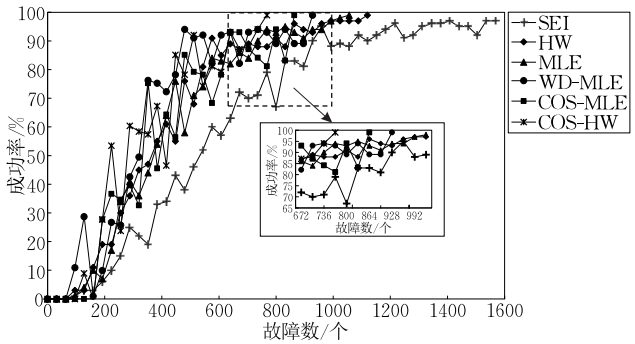

图 6 统计故障分析方法破译主密钥的成功率

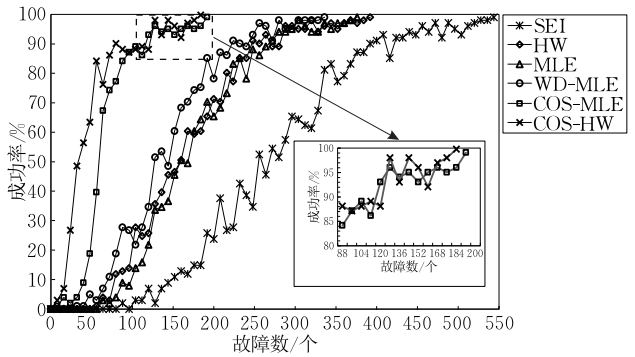

图 7 中间相错统计故障分析方法破译主密钥的成功率

5.3 耗 时

耗时是指完成故障注入到恢复 128 比特主密钥完整过程所需的时间。主要步骤有注入随机故障、中间相错分析、使用区分器筛选密钥及恢复原始密钥。图 8 和图 9 分别展示了统计故障分析方法和中间相错统计故障分析方法，在导入不同故障数情况下，分别使用不同区分器破译主密钥的耗时累加。其中，横轴表示注入故障个数，纵轴表示耗时累加。实验数据表明，通过统计故障分析和中间相错统计故障分析方法，在 TweGIFT-64 版本下，使用经典区分器耗时分别为 2.84 s、2.11 s、2.16 s、2.03 s 以及 132.05 h、112.39 h、117.85 h、58.92 h，使用新型区分器耗时分别为 1.94 s、1.94 s 以及 56.30 h、53.02 h；在 TweGIFT-128 版本下，使用经典区分器耗时分别为 8.11 s、7.92 s、8.78 s、6.71 s 以及 426.97 h、419.66 h、385.31 h、377.44 h，新型区分器耗时分别为 6.38 s、5.97 s 以及 169.83 h、167.28 h。可知，本文提出的新型区分器明显优于经典区分器。

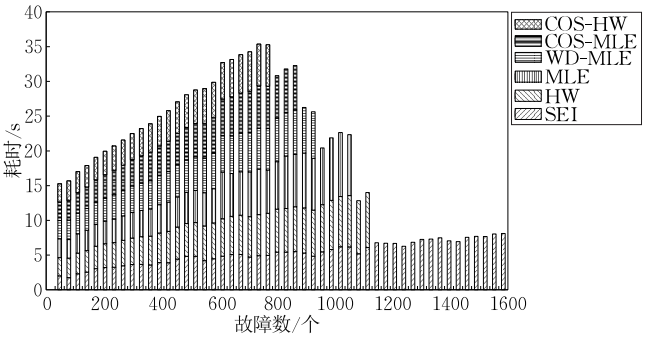

图 8 统计故障分析方法破译主密钥的耗时累加

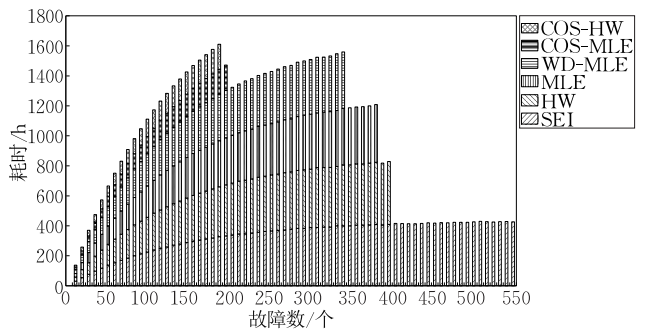

图 9 中间相错统计故障分析方法破译主密钥的耗时累加

5.4 复杂度

时间复杂度、数据复杂度和空间复杂度分别代表算法在运行过程中所需的时间量级、需要处理的数据量级和所需的内存空间量级，计算公式分别为

$$r\cdot f_n+2^u\cdot f_n\cdot q,\ 2^u\cdot f_n,\ 2^{u+w}+4\cdot j\cdot f_n,$$

其中，r 为迭代轮数，$r\in\{28,40\}$，f_n 为故障数，u 为需枚举密钥比特位数，q 为区分器复杂度系数，w 为半字节中间状态值所有可能取值个数，$4\cdot j$ 为分组长度，$j\in\{16,32\}$，半字节故障模型下 $w=16$。表 6 展示了分别使用统计故障分析和中间相错统计故障分析，破译 TweGIFT 算法所有版本的实验数据。中间相错统计故障分析最深故障注入轮数为倒数第三轮，比统计故障分析更深一轮，需要进行中间相错分析并存储大小为 $2^{26}(=2^{16}\cdot 2^{10})$ 的映射矩阵，因此复杂度高于统计故障分析。

表 6　统计故障分析和中间相错统计故障分析恢复 TweGIFT-64 和 TweGIFT-128 主密钥的关键指标对比

区分器	统计故障分析					中间相错统计故障分析				
	注入轮数	故障数	时间复杂度	数据复杂度	存储复杂度	注入轮数	故障数	时间复杂度	数据复杂度	存储复杂度
SEI	27/39	656/1600	$2^{18.94}/2^{24.72}$	$2^{19.35}/2^{20.64}$	$2^{14.29}/2^{14.98}$	26/38	280/544	$2^{41.88}/2^{45.16}$	$2^{40.12}/2^{41.84}$	$2^{26.34}/2^{26.46}$
HW	27/39	352/1120	$2^{18.91}/2^{24.20}$	$2^{18.45}/2^{20.12}$	$2^{14.17}/2^{14.83}$	26/38	184/392	$2^{41.86}/2^{44.59}$	$2^{39.64}/2^{40.61}$	$2^{26.22}/2^{26.66}$
MLE	27/39	368/1056	$2^{18.93}/2^{24.12}$	$2^{18.52}/2^{20.04}$	$2^{14.16}/2^{14.88}$	26/38	200/376	$2^{40.84}/2^{44.57}$	$2^{39.52}/2^{40.55}$	$2^{26.19}/2^{26.55}$
WD-MLE	27/39	320/928	$2^{17.92}/2^{23.84}$	$2^{18.32}/2^{19.85}$	$2^{14.19}/2^{14.89}$	26/38	176/336	$2^{40.89}/2^{44.48}$	$2^{39.45}/2^{40.39}$	$2^{26.22}/2^{26.32}$
COS-MLE	27/39	288/864	$2^{17.81}/2^{23.73}$	$2^{18.16}/2^{19.75}$	$2^{14.09}/2^{14.85}$	26/38	148/192	$2^{40.78}/2^{43.56}$	$2^{39.20}/2^{39.58}$	$2^{26.13}/2^{26.51}$
COS-HW	27/39	272/768	$2^{17.76}/2^{23.57}$	$2^{18.08}/2^{19.58}$	$2^{14.08}/2^{14.98}$	26/38	132/184	$2^{40.73}/2^{43.51}$	$2^{39.04}/2^{39.52}$	$2^{26.03}/2^{26.34}$

在 TweGIFT-128 版本中，当故障注入轮数均为第 39 轮时，以 COS-HW 区分器为例，使用统计故障分析的时间、数据、空间复杂度分别为 $2^{23.73}$、$2^{19.75}$ 和 $2^{14.85}$，使用中间相错统计故障分析的时间、数据、空间复杂度分别为 $2^{17.57}$、$2^{17.70}$ 和 $2^{12.00}$，各复杂度降低值分别为 $2^{6.16}$、$2^{2.05}$ 和 $2^{2.85}$。可知，故障注入轮数相同时，中间相错统计故障分析的时间、数据和存储复杂度皆远低于统计故障分析。因此，本文提出的中间相错统计故障分析比统计故障分析的故障注入轮数更深；新型双重组合区分器 COS-HW 和 COS-MLE 比经典区分器的复杂度更低。本文以更深的故障注入轮数、更低的实现代价破译了 TweGIFT 算法的所有版本。

6　结束语

本文针对轻量级密码 TweGIFT 算法所有版本进行唯密文故障分析，结合中间相错策略提出中间相错统计故障分析，与传统统计故障分析相比，故障注入轮数更深一轮。本文还基于余弦距离的思想构建 COS-HW 和 COS-MLE 两种新型区分器，与经典区分器相比，故障数、成功率以及耗时等指标表现均有优化，不仅破译成功率达到 99%以上而且攻击代价明显减少，恢复 128 比特主密钥所需故障数减少 77.77%。研究结果表明，在使用 TweGIFT 算法时要注意抵御中间相错统计故障分析的安全威胁。该研究聚焦于新型唯密文分析，有助于推动轻量级密码算法的进一步研究。未来将结合量子密码等新型密码的安全分析方法，对其他密码算法进行故障分析。

参　考　文　献

[1] Ding Z, He D, Qiao Q, et al. A lightweight and secure communication protocol for the IoT environment. IEEE Transactions on Dependable and Secure Computing, 2024, 21(3): 1050-1067

[2] Hadipour H, Derbez P, Eichlseder M. Revisiting differential-linear attacks via a boomerang perspective with application to AES, Ascon, CLEFIA, SKINNY, PRESENT, KNOT, TWINE, WARP, LBlock, Simeck, and SERPENT//Proceedings of the Advances in Cryptology. California, USA, 2024, 14923: 38-72

[3] Andreeva E, Bogdanov A, Datta N, et al. The COLM authenticated encryption scheme. Journal of Cryptology, 2024, 37(2): 15

[4] Shepherd C, Kalbantner J, Semal B, et al. A side-channel analysis of sensor multiplexing for covert channels and application profiling on mobile devices. IEEE Transactions on Dependable and Secure Computing, 2024, 21(4): 3141-3152

[5] Esser A, Mukherjee A, Sarkar S. Memory-efficient attacks on small LWE keys. Journal of Cryptology, 2024, 37(4): 36

[6] Guo Yi-Min, Zhang Zhen-Feng, Xiong Ping, et al. PUF-Based lightweight authentication protocols for fog assisted IoT. Chinese Journal of Computers, 2022, 45(7): 1412-1430(in Chinese)
(郭奕旻，张振峰，熊平等. 基于 PUF 的轻量级雾辅助物联网认证协议. 计算机学报，2022，45(7)：1412-1430)

[7] Chakraborti A, Datta N, Jha A, et al. Elastic-tweak: A framework for short tweak tweakable block cipher//Proceedings of the Progress in Cryptology. Jaipur, India, 2021: 114-137

[8] Chakraborti A, Datta N, Jha A, et al. tHyENA: Making HyENA even smaller//Proceedings of the Progress in Cryptology. Jaipur, India, 2021: 26-48

[9] Chakraborti A, Datta N, Jha A, et al. INT-RUP secure lightweight parallel AE modes. IACR Transactions on Symmetric Cryptology, 2020, 2019(4): 81-118

[10] Chakraborti A, Datta N, Jha A, et al. ESTATE: A lightweight and low energy authenticated encryption mode. IACR Transactions on Symmetric Cryptology, 2020, 2020(S1): 350-389

[11] Zhang R, Zhang M, Yan J, et al. Differential cryptanalysis of TweGIFT-128 based on neural network//Proceedings of the International Conference on Data Science in Cyberspace. Shenzhen, China, 2021: 529-534

[12] Chakraborti A, Datta N, Jha A, et al. Short tweak TBC and its applications in symmetric ciphers. Computer Science, 2023

[13] Liu S, Guan J, Hu B. Fault attacks on authenticated encryption modes for GIFT. IET Information Security, 2021, 16(1): 51-63

[14] Kocher P C. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems//Proceedings of the Annual International Cryptology Conference. California, USA: Springer, 1996, 1109: 104-113

[15] Wang Yong-Juan, Fan Hao-Peng, Dai Zheng-Yi, et al. Advances in side channel attacks and countermeasures. Chinese Journal of Computers, 2023, 46(1): 202-228(in Chinese)
(王永娟, 樊昊鹏, 代政一等. 侧信道攻击与防御技术研究进展. 计算机学报, 2023, 46(1): 202-228)

[16] Reinbrecht C, Aljuffri A, Hamdioui S, et al. GRINCH: A cache attack against GIFT lightweight cipher//Proceedings of the Design, Automation & Test in Europe Conference & Exhibition. Grenoble, France, 2021: 549-554

[17] Mozipo A T, Acken J M. Residual vulnerabilities to power side channel attacks of lightweight ciphers cryptography competition finalists. IET Computers and Digital Techniques, 2023, 17(3-4): 75-88

[18] Joshi P, Mazumdar B. Semi-permanent stuck-at fault injection attacks on Elephant and GIFT lightweight ciphers. ACM Transactions on Design Automation of Electronic Systems, 2024, 29(4): 1-32

[19] Li W, Li J, Gu D, et al. Statistical fault analysis of the Simeck lightweight cipher in the ubiquitous sensor networks. IEEE Transactions on Information Forensics and Security, 2021, 16: 4224-4233

[20] Xiao H, Wang L. Differential fault analysis on the lightweight block cipher plug-in plug-out. Security and Privacy, 2023, 6(3): 1-13

[21] Le D-P, Yeo S L, Khoo K. Algebraic differential fault analysis on SIMON block cipher. IEEE Transactions on Computers, 2019, 68(11): 1561-1572

[22] Kang Y, Yu Q, Qin L, et al. Meet-in-the-middle differential fault analysis on ITUbee block cipher. Symmetry, 2023, 15(6): 1196

[23] Zhang F, Lou X, Zhao X, et al. Persistent fault analysis on block ciphers. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2018, (3): 150-172

[24] Beigizad A A, Soleimany H, Zarei S, et al. Linked fault analysis. IEEE Transactions on Information Forensics and Security, 2024, 19: 632-645

[25] Gao Y, Zhang Z, Zhang Z. Differential fault attack of lightweight cipher GIFT based on byte model. IEEE Internet of Things Journal, 2025, 12(1): 435-444

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

[27] Nozaki Y, Yoshikawa M. Statistical fault analysis for a lightweight cipher Midori//Proceedings of the IEEE International Conference on Information and Automation. Macau, China, 2017: 236-241

[28] Ramezanpour K, Ampadu P, Diehl W. A statistical fault analysis methodology for the Ascon authenticated cipher//Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust. McLean, USA, 2019: 41-50

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

[30] Salam I, Alawatugoda J, Madushan H. Statistical fault analysis of TinyJambu. Discover Applied Sciences, 2024, 6(2): 55

[31] Ulitzsch V Q, Marzougui S, Tibouchi M, et al. Profiling side-channel attacks on Dilithium: A small bit-fiddling leak breaks it all//Proceedings of the Selected Areas in Cryptography. Ontario, Canada, 2024, 13742: 3-32

[32] Ege B, Swinkels B, Toprakhisar D, et al. Practical improvements to statistical ineffective fault attacks//Proceedings of the Constructive Side-Channel Analysis and Secure Design. Gardanne, France, 2024, 14595: 59-75

[33] Li Wei, Gao Jian-Ning, Gu Da-Wu, et al. Impossible statistical fault analysis of the Pyjamask and SUNDAE-GIFT lightweight cryptosystems. Chinese Journal of Computers, 2024, 47(5): 1010-1029(in Chinese)
(李玮, 高建宁, 谷大武等. 轻量级密码 Pyjamask 和 SUNDAE-GIFT 的不可能统计故障分析. 计算机学报, 2024, 47(5): 1010-1029)

[34] Biham E, Biryukov A, Shamir A. Miss in the middle attacks on IDEA and Khufu//Proceedings of the Fast Software Encryption. Rome, Italy, 1999: 124-138

[35] Biham E, Biryukov A, Shamir A. Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. Journal of Cryptology, 2005, 18(4): 291-311

[36] Sadeghi S, Bagheri N. Improved zero-correlation and impossible differential cryptanalysis of reduced-round SIMECK block cipher. IET Information Security, 2018, 12(4): 314-325

[37] Xie H, Yang L. Quantum miss-in-the-middle attack. arXiv preprint arXiv: 1812.08499, 2021

[38] Wang X, Wu B, Hou L, et al. Searching for impossible subspace trails and improved impossible differential characteristics for SIMON-like block ciphers. Cybersecurity, 2021, 4(1): 17

[39] 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

[40] Fisher R A. On the mathematical foundations of theoretical statistics. Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 1922, 222(594-604): 309-368

[41] Panaretos V M, Zemel Y. Statistical aspects of Wasserstein distances. Annual Review of Statistics and Its Application, 2019, 6(1): 405-431

[42] Qian G, Sural S, Gu Y, et al. Similarity between Euclidean and cosine angle distance for nearest neighbor queries//Proceedings of the ACM Symposium on Applied Computing. New York, USA, 2004: 1232-1237

附录 A.　实验数据

明文：随机生成

128 比特主密钥：0x0123456789abcedf

表 A1　统计故障分析方法恢复 TweGIFT-128 版本 10 比特轮密钥的成功率及耗时

故障数	成功率/%						耗时/ms					
	SEI	HW	MLE	WD-MLE	COS-MLE	COS-HW	SEI	HW	MLE	WD-MLE	COS-MLE	COS-HW
1	0	0	0	0	0	0	62.82	82.50	84.53	82.35	82.82	82.04
2	0	0	0	0	0	0	55.47	87.97	82.81	87.65	89.22	87.03
3	1.05	3.41	1.00	10.89	0	0	72.81	92.50	86.72	92.50	93.59	93.44
4	3.23	3.05	4.19	28.71	0	8.91	79.07	97.50	91.72	97.50	96.41	96.88
5	3.33	11.18	10.19	0.99	0	0	95.00	101.72	96.72	100.78	100.78	100.93
6	6.17	19.19	7.29	9.90	27.72	27.72	100.93	106.56	100.46	105.00	105.31	105.16
7	10.03	19.20	17.28	26.73	36.63	53.47	102.19	110.47	105.79	109.69	109.53	109.06
8	15.29	30.45	34.55	25.74	34.65	23.76	108.75	114.53	109.53	114.06	113.75	112.97
9	25.25	36.36	41.41	42.57	39.60	60.40	113.60	119.38	115.00	118.91	118.91	117.65
10	22.22	45.45	36.36	49.50	32.67	58.42	114.84	123.91	118.43	122.97	122.81	122.34
11	19.19	47.47	44.44	76.24	75.25	57.43	112.19	129.37	122.19	127.50	127.66	127.50
12	33.33	55.56	54.55	75.25	45.54	67.33	122.50	133.44	127.04	132.34	132.35	132.03
13	34.34	61.62	64.65	72.28	64.36	46.53	122.03	140.79	131.71	137.19	137.18	136.57
14	43.43	55.56	76.77	78.22	56.44	85.15	138.75	143.28	136.41	144.37	141.41	141.72
15	38.38	76.77	58.59	94.06	85.15	78.22	150.31	147.81	140.15	145.78	146.25	146.41
16	46.46	68.69	71.72	91.09	79.21	92.08	150.78	152.35	144.38	150.78	150.16	150.78
17	52.53	81.82	74.75	92.08	78.22	74.26	130.94	156.88	149.69	155.00	156.40	155.94
18	60.61	91.92	84.85	82.18	68.32	81.19	139.84	161.25	154.22	159.68	159.69	158.90
19	57.58	85.86	83.84	92.08	78.22	79.21	151.88	167.50	210.47	163.91	164.37	163.91
20	63.64	89.90	82.83	93.07	93.07	93.07	158.90	171.72	192.50	169.37	175.79	167.96
21	72.73	87.88	86.87	82.18	93.07	87.13	159.22	175.94	196.10	173.75	179.84	173.13
22	70.71	89.90	84.85	93.07	87.13	88.12	147.97	180.62	201.87	177.34	185.31	177.50
23	71.72	88.89	90.91	94.06	84.16	94.06	154.06	184.53	204.37	183.59	191.10	187.66
24	79.80	88.89	94.95	93.07	81.19	99.80	155.00	188.91	195.00	186.56	190.46	186.87
25	67.68	91.92	93.94	89.11	94.06	—	169.38	193.60	213.13	195.63	192.03	—
26	83.84	88.89	95.96	94.06	83.17	—	169.06	196.80	235.62	195.63	196.25	—
27	83.84	96.97	93.94	89.11	99.01	—	172.34	201.70	234.84	200.15	199.53	—
28	81.82	94.95	91.92	89.11	—	—	165.63	203.10	246.56	205.32	—	—
29	90.91	93.94	94.95	99.09	—	—	152.03	206.70	232.97	209.68	—	—
30	95.96	96.97	94.95	—	—	—	170.94	212.40	254.69	—	—	—
31	88.89	97.98	97.98	—	—	—	182.03	220.70	280.93	—	—	—
32	89.90	97.98	98.99	—	—	—	193.91	225.70	287.66	—	—	—
33	88.89	97.98	99.23	—	—	—	193.12	230.50	274.68	—	—	—
34	92.93	97.98	—	—	—	—	162.66	238.80	—	—	—	—
35	90.91	99.15	—	—	—	—	190.15	247.60	—	—	—	—
36	92.93	—	—	—	—	—	211.72	—	—	—	—	—
37	94.95	—	—	—	—	—	210.00	—	—	—	—	—
38	96.97	—	—	—	—	—	208.75	—	—	—	—	—
39	91.92	—	—	—	—	—	195.47	—	—	—	—	—
40	92.93	—	—	—	—	—	214.37	—	—	—	—	—
41	95.96	—	—	—	—	—	226.88	—	—	—	—	—
42	96.97	—	—	—	—	—	228.28	—	—	—	—	—
43	96.97	—	—	—	—	—	234.37	—	—	—	—	—
44	97.98	—	—	—	—	—	220.63	—	—	—	—	—
45	95.96	—	—	—	—	—	216.72	—	—	—	—	—
46	95.96	—	—	—	—	—	236.25	—	—	—	—	—
47	92.93	—	—	—	—	—	240.78	—	—	—	—	—
48	97.98	—	—	—	—	—	239.69	—	—	—	—	—
49	97.98	—	—	—	—	—	251.71	—	—	—	—	—
50	99.25	—	—	—	—	—	253.60	—	—	—	—	—
50	99.25	—	—	—	—	—	253.60	—	—	—	—	—

表 A2 中间相错统计故障分析方法恢复 TweGIFT-128 版本 42 比特轮密钥的成功率及耗时

故障数	成功率/%						耗时/h					
	SEI	HW	MLE	WD-MLE	COS-MLE	COS-HW	SEI	HW	MLE	WD-MLE	COS-MLE	COS-HW
1	0	0	0	0	0.99	2.97	3.83	3.47	3.08	3.38	1.71	1.73
2	0	0	0	0	3.96	6.93	6.85	6.65	5.93	6.34	3.28	3.28
3	0	0	0	0	1.98	26.73	9.67	9.63	8.47	9.11	4.72	4.73
4	0	0	0	0.99	3.96	48.51	12.26	12.42	10.89	11.59	6.07	6.08
5	0	0	0	0.99	8.91	56.44	14.73	14.98	13.19	13.91	7.36	7.36
6	0	0	0	4.95	18.81	63.37	17.11	17.40	15.36	16.20	8.59	8.59
7	0	0.99	0	2.97	39.60	84.16	19.25	19.66	17.39	18.23	9.72	9.70
8	0	3.96	2.97	6.93	67.33	76.24	21.21	21.74	19.35	20.13	10.76	10.74
9	0	2.97	1.98	10.89	74.26	86.14	23.05	23.83	21.15	22.03	11.78	11.79
10	0	11.88	3.96	18.81	77.23	90.10	25.01	25.72	22.82	23.85	12.68	12.66
11	1.98	12.87	8.91	27.72	84.16	88.12	26.67	27.27	24.32	25.37	13.64	13.60
12	0	13.86	7.92	26.73	87.13	87.13	28.23	28.88	26.06	26.99	14.46	14.32
13	2.97	27.72	13.86	21.78	89.11	88.12	29.87	30.61	27.44	28.33	15.16	15.19
14	2.97	24.75	15.84	27.72	86.14	89.11	31.28	31.91	29.14	29.69	15.99	15.90
15	6.93	25.74	21.78	34.65	93.07	88.12	32.45	33.38	30.40	30.92	16.68	16.61
16	1.98	35.64	33.66	51.49	96.04	98.02	33.80	34.56	31.68	31.97	17.35	17.28
17	6.93	39.60	34.65	53.47	94.06	93.07	34.76	35.83	32.75	33.25	17.86	17.89
18	8.91	45.54	36.63	48.51	95.05	98.02	36.08	36.98	33.95	34.26	18.47	18.51
19	10.89	46.53	45.54	60.40	93.07	96.04	37.14	38.18	34.89	35.28	18.98	19.03
20	12.87	50.50	50.50	68.32	95.05	92.08	38.23	38.93	35.73	36.31	19.52	19.46
21	11.88	60.40	49.50	70.30	96.04	97.03	39.13	39.80	36.62	37.12	20.07	19.95
22	14.85	59.41	60.40	74.26	95.05	98.02	39.89	40.92	37.50	37.77	20.49	20.48
23	14.85	60.40	64.36	75.25	96.04	99.59	40.86	41.67	38.39	38.77	20.69	20.91
24	25.74	65.35	70.30	85.15	99.19	—	41.56	42.57	39.27	39.39	21.24	—
25	23.76	71.29	65.35	78.22	—	—	42.36	43.20	39.79	40.13	—	—
26	37.62	70.30	68.32	87.13	—	—	43.23	43.91	40.50	40.66	—	—
27	26.73	80.20	73.27	86.14	—	—	43.39	44.61	41.21	41.48	—	—
28	27.72	77.23	83.17	91.09	—	—	44.04	45.09	41.47	42.05	—	—
29	42.57	85.15	85.15	90.10	—	—	44.71	45.67	42.38	42.57	—	—
30	38.61	85.15	78.22	89.11	—	—	45.32	45.98	42.72	43.10	—	—
31	34.65	91.09	88.12	93.07	—	—	45.66	46.67	42.92	43.40	—	—
32	52.48	90.10	86.14	97.03	—	—	46.16	46.94	43.57	44.10	—	—
33	45.54	93.07	89.11	96.04	—	—	46.58	47.73	44.10	44.19	—	—
34	54.46	89.11	91.09	91.09	—	—	47.10	47.83	44.31	44.52	—	—
35	51.49	89.11	94.06	98.02	—	—	47.83	48.31	45.05	45.15	—	—
36	57.43	96.04	94.06	95.05	—	—	47.89	48.83	45.30	45.36	—	—
37	65.35	96.04	95.05	95.05	—	—	48.52	48.71	45.77	45.59	—	—
38	64.36	95.05	98.02	98.02	—	—	48.63	49.82	45.77	46.23	—	—
39	62.38	95.05	96.04	98.02	—	—	48.87	49.68	46.19	45.98	—	—
40	61.39	98.02	94.06	98.02	—	—	48.98	49.77	46.52	46.25	—	—
41	67.33	94.06	94.06	98.02	—	—	49.93	49.61	46.59	47.30	—	—
42	81.19	96.04	97.03	99.43	—	—	49.97	50.90	46.94	47.18	—	—
43	83.17	97.03	95.05	—	—	—	49.95	50.86	47.63	—	—	—
44	77.23	95.05	97.03	—	—	—	50.50	51.14	47.47	—	—	—
45	79.21	97.03	97.03	—	—	—	50.52	51.18	47.68	—	—	—
46	83.17	98.02	98.02	—	—	—	50.74	51.52	47.81	—	—	—
47	87.13	97.03	99.37	—	—	—	51.09	51.81	48.16	—	—	—
48	87.13	98.02	—	—	—	—	50.60	51.66	—	—	—	—
49	90.10	99.29	—	—	—	—	51.21	52.46	—	—	—	—
50	91.09	—	—	—	—	—	52.02	—	—	—	—	—
51	93.07	—	—	—	—	—	51.83	—	—	—	—	—
52	85.15	—	—	—	—	—	51.69	—	—	—	—	—
53	92.08	—	—	—	—	—	51.71	—	—	—	—	—
54	92.08	—	—	—	—	—	52.12	—	—	—	—	—
55	94.06	—	—	—	—	—	52.51	—	—	—	—	—

（续　表）

故障数	成功率/%						耗时/h					
	SEI	HW	MLE	WD-MLE	COS-MLE	COS-HW	SEI	HW	MLE	WD-MLE	COS-MLE	COS-HW
56	93.07	—	—	—	—	—	52.28	—	—	—	—	—
57	96.04	—	—	—	—	—	52.63	—	—	—	—	—
58	94.06	—	—	—	—	—	52.57	—	—	—	—	—
59	97.03	—	—	—	—	—	53.02	—	—	—	—	—
60	92.08	—	—	—	—	—	52.93	—	—	—	—	—
61	97.03	—	—	—	—	—	52.84	—	—	—	—	—
62	95.05	—	—	—	—	—	53.37	—	—	—	—	—
63	93.07	—	—	—	—	—	53.68	—	—	—	—	—
64	96.04	—	—	—	—	—	53.42	—	—	—	—	—
65	97.03	—	—	—	—	—	53.29	—	—	—	—	—
66	98.02	—	—	—	—	—	53.43	—	—	—	—	—
67	98.02	—	—	—	—	—	53.57	—	—	—	—	—
68	99.19	—	—	—	—	—	53.37	—	—	—	—	—

表 A3　统计故障分析方法恢复 TweGIFT-64 版本 10 比特轮密钥的成功率及耗时

故障数	成功率/%						耗时/h					
	SEI	HW	MLE	WD-MLE	COS-MLE	COS-HW	SEI	HW	MLE	WD-MLE	COS-MLE	COS-HW
1	0	0	0	0	0	0	81.09	80.31	82.97	82.19	80.31	84.07
2	0	0	0	0	0	0	84.06	81.56	82.19	83.44	81.56	83.43
3	0	0	0	0	0	0	86.56	84.84	85.78	85.62	85.31	86.88
4	0	0	0	10.89	23.76	0	89.54	87.82	87.81	87.50	87.35	89.84
5	0	10.89	10.89	10.89	0	10.89	91.56	90.15	90.63	90.16	90.31	91.88
6	10.89	43.56	10.89	10.89	10.89	10.89	93.59	92.35	92.50	91.87	92.03	94.22
7	30.69	3.96	49.50	25.74	20.79	36.63	96.09	94.85	94.84	94.84	94.85	96.41
8	19.80	30.69	57.43	48.51	36.63	50.50	98.75	98.12	97.34	97.34	97.34	98.75
9	39.60	40.59	64.36	59.41	29.70	69.31	101.25	99.69	99.69	99.85	99.68	101.40
10	8.91	28.71	65.35	47.52	69.31	80.20	103.91	101.87	102.34	102.65	102.35	104.22
11	55.45	56.44	31.68	35.64	73.27	80.20	106.40	105.47	105.00	104.85	107.50	106.40
12	27.72	70.30	63.37	53.47	81.19	53.47	109.07	107.50	106.72	107.34	108.44	108.75
13	24.75	63.37	45.54	81.19	61.39	49.50	110.93	110.16	110.15	109.84	112.03	111.41
14	26.73	78.22	55.45	63.37	60.40	90.10	113.75	112.03	112.81	117.66	112.19	113.91
15	8.91	90.10	76.24	81.19	64.36	91.09	115.94	115.00	114.69	115.31	114.68	117.03
16	65.35	90.10	88.12	74.26	89.11	94.31	118.91	117.18	117.19	118.28	116.87	119.84
17	58.42	82.18	90.10	90.10	96.26	99.27	120.47	120.63	119.85	119.69	119.38	121.72
18	21.78	91.09	83.17	83.17	99.19	—	122.66	122.34	122.96	122.50	121.56	—
19	58.42	90.10	91.09	93.27	—	—	124.21	124.21	125.32	125.16	—	—
20	39.60	91.09	91.09	99.33	—	—	127.19	127.35	127.34	127.18	—	—
21	53.47	91.09	91.09	—	—	—	130.00	130.63	129.22	—	—	—
22	63.37	99.49	92.08	—	—	—	131.72	132.03	132.50	—	—	—
23	56.44	—	99.20	—	—	—	134.85	—	135.31	—	—	—
24	49.50	—	—	—	—	—	136.88	—	—	—	—	—
25	77.23	—	—	—	—	—	140.00	—	—	—	—	—
26	78.22	—	—	—	—	—	142.50	—	—	—	—	—
27	78.22	—	—	—	—	—	145.00	—	—	—	—	—
28	79.21	—	—	—	—	—	147.35	—	—	—	—	—
29	90.10	—	—	—	—	—	149.37	—	—	—	—	—
30	57.43	—	—	—	—	—	152.03	—	—	—	—	—
31	80.20	—	—	—	—	—	155.79	—	—	—	—	—
32	82.18	—	—	—	—	—	156.10	—	—	—	—	—
33	87.13	—	—	—	—	—	159.21	—	—	—	—	—
34	93.07	—	—	—	—	—	162.04	—	—	—	—	—
35	91.09	—	—	—	—	—	163.90	—	—	—	—	—
36	87.13	—	—	—	—	—	168.60	—	—	—	—	—
37	93.07	—	—	—	—	—	169.22	—	—	—	—	—
38	93.07	—	—	—	—	—	171.25	—	—	—	—	—
39	93.07	—	—	—	—	—	173.59	—	—	—	—	—
40	97.13	—	—	—	—	—	175.47	—	—	—	—	—
41	99.29	—	—	—	—	—	177.81	—	—	—	—	—

表 A4 中间相错统计故障分析恢复 TweGIFT-64 版本 42 比特轮密钥的成功率及耗时

故障数	成功率/%						耗时/ms					
	SEI	HW	MLE	WD-MLE	COS-MLE	COS-HW	SEI	HW	MLE	WD-MLE	COS-MLE	COS-HW
1	0	0	0	0	0	1.98	1.53	1.23	1.31	0.63	0.64	0.66
2	0	0	0	0	0	2.97	2.62	2.36	2.42	1.24	1.27	1.28
3	0	0	0	0	0	0	3.75	3.41	3.55	1.83	1.84	1.83
4	0	0	0	0	0	9.90	4.78	4.51	4.49	2.46	2.50	2.43
5	0	0.99	0.99	0.99	0.99	21.78	5.93	5.42	5.64	2.89	3.55	2.95
6	0	1.98	0.99	4.96	4.95	37.62	6.85	6.40	6.42	3.47	3.52	3.52
7	0.99	1.98	3.96	5.95	2.97	49.50	7.66	7.51	7.47	3.92	4.36	4.04
8	0.99	10.89	9.90	10.90	11.88	56.45	8.55	8.25	8.23	4.38	4.68	4.48
9	0.99	8.91	17.82	21.79	17.82	60.41	9.52	9.15	9.27	4.90	5.08	5.02
10	3.96	13.86	17.82	21.79	22.77	58.43	10.28	10.16	10.09	5.37	5.86	5.33
11	5.94	23.76	19.80	24.76	17.82	70.31	11.14	10.91	10.88	5.93	6.16	5.88
12	9.90	31.68	36.63	32.68	30.69	77.24	11.76	12.08	11.52	6.16	6.79	6.21
13	5.94	28.71	35.64	44.56	28.71	68.33	12.55	12.49	12.59	6.50	7.17	6.83
14	12.87	33.66	40.59	39.61	30.69	67.34	13.83	13.19	13.03	7.17	7.45	7.47
15	14.85	44.55	47.52	48.52	39.60	79.22	14.17	13.91	13.57	7.60	8.32	7.76
16	11.88	52.48	50.50	39.61	46.53	86.15	15.40	15.11	14.65	7.84	8.26	8.10
17	14.85	45.54	59.41	51.50	44.55	82.19	15.69	15.32	15.37	8.21	8.96	8.63
18	18.81	64.36	53.47	58.43	57.44	87.14	16.74	16.53	15.96	8.64	9.05	9.00
19	15.84	71.29	61.39	60.41	51.50	79.22	16.94	17.15	16.55	9.19	9.48	9.30
20	25.74	63.37	70.30	67.34	65.36	83.18	17.45	17.73	16.68	9.31	9.71	9.31
21	22.77	63.37	62.38	62.39	63.38	88.13	18.57	18.58	17.91	9.58	10.20	9.76
22	23.76	73.27	77.23	74.27	72.29	91.10	19.66	18.57	18.28	9.78	10.31	10.01
23	30.69	68.32	76.24	76.25	78.23	88.13	19.51	19.63	18.90	10.10	10.88	10.57
24	35.64	78.22	79.21	82.19	74.27	95.06	20.08	20.08	18.94	11.03	11.39	10.50
25	28.71	85.15	80.20	72.29	80.21	95.06	21.03	20.40	20.06	10.66	11.82	11.09
26	37.62	82.18	83.17	81.20	83.18	92.09	21.16	21.20	20.37	11.63	11.54	11.07
27	39.60	84.16	84.16	79.22	78.23	91.10	21.90	20.73	21.01	11.29	12.49	11.58
28	39.60	82.18	86.14	94.07	83.18	93.08	21.60	21.90	20.98	11.26	11.86	11.78
29	51.49	79.21	83.17	91.10	91.10	94.07	23.02	21.60	21.72	12.19	13.29	12.01
30	55.45	89.11	85.15	91.10	80.21	96.05	22.34	23.09	22.26	11.97	12.77	12.70
31	60.40	88.12	92.08	80.21	88.13	95.06	23.70	23.78	23.01	12.49	13.07	12.97
32	62.38	89.11	84.16	91.10	89.12	98.03	23.59	23.17	23.63	12.62	13.00	12.41
33	61.39	94.06	87.13	91.10	92.09	99.62	24.07	23.05	22.50	12.85	13.30	13.26
34	72.28	95.05	94.06	93.08	95.06	—	24.14	24.25	24.22	12.90	14.35	—
35	67.33	92.08	94.06	94.07	94.07	—	25.44	24.82	24.37	13.61	14.38	—
36	66.34	94.06	96.04	97.04	98.03	—	23.96	25.35	25.37	13.16	14.16	—
37	68.32	95.05	96.04	98.03	99.59	—	26.32	26.35	25.30	13.35	14.08	—
38	67.33	97.03	95.05	95.06	—	—	25.53	26.73	25.49	13.82	—	—
39	78.22	96.04	97.03	96.05	—	—	26.46	27.42	25.63	14.17	—	—
40	83.17	95.05	96.04	95.06	—	—	27.39	27.23	27.29	14.21	—	—
41	79.21	95.05	97.03	95.06	—	—	26.92	26.97	25.90	14.81	—	—
42	81.19	96.04	98.02	96.05	—	—	27.74	27.07	27.02	14.84	—	—
43	81.19	98.02	97.03	96.05	—	—	27.76	27.18	27.51	14.39	—	—
44	85.15	98.02	96.04	99.31	—	—	27.24	26.50	27.72	14.73	—	—
45	91.09	98.02	98.02	—	—	—	27.84	27.17	28.10	—	—	—
46	87.13	99.25	98.02	—	—	—	28.70	28.10	28.02	—	—	—
47	86.14	—	96.04	—	—	—	28.15	—	27.47	—	—	—
48	88.12	—	98.02	—	—	—	29.20	—	27.88	—	—	—
49	91.09	—	98.02	—	—	—	28.63	—	28.00	—	—	—
50	95.05	—	99.07	—	—	—	30.87	—	29.46	—	—	—
51	97.03	—	—	—	—	—	29.11	—	—	—	—	—
52	94.06	—	—	—	—	—	29.93	—	—	—	—	—
53	91.09	—	—	—	—	—	31.44	—	—	—	—	—
54	92.08	—	—	—	—	—	29.80	—	—	—	—	—
55	95.05	—	—	—	—	—	30.02	—	—	—	—	—
56	95.05	—	—	—	—	—	30.06	—	—	—	—	—
57	96.04	—	—	—	—	—	30.82	—	—	—	—	—

（续　表）

故障数	成功率/%						耗时/h					
	SEI	HW	MLE	WD-MLE	COS-MLE	COS-HW	SEI	HW	MLE	WD-MLE	COS-MLE	COS-HW
58	96.04	—	—	—	—	—	30.90	—	—	—	—	—
59	96.04	—	—	—	—	—	30.79	—	—	—	—	—
60	95.05	—	—	—	—	—	32.43	—	—	—	—	—
61	97.03	—	—	—	—	—	32.81	—	—	—	—	—
62	96.04	—	—	—	—	—	32.52	—	—	—	—	—
63	95.05	—	—	—	—	—	31.24	—	—	—	—	—
64	97.03	—	—	—	—	—	33.00	—	—	—	—	—
65	95.05	—	—	—	—	—	31.47	—	—	—	—	—
66	98.02	—	—	—	—	—	33.12	—	—	—	—	—
67	98.02	—	—	—	—	—	32.73	—	—	—	—	—
68	98.02	—	—	—	—	—	31.84	—	—	—	—	—
69	98.02	—	—	—	—	—	32.25	—	—	—	—	—
70	99.53	—	—	—	—	—	33.01	—	—	—	—	—

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

LIU Yuan, M. S. candidate. Her main research interests include fault analysis of block ciphers.

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

HUANG Jia-Yin, M. S. candidate. Her main research interests include fault analysis of block ciphers.

LU Hai-Ning, M. S., engineer. His main research interests include cryptographic engineering and network security protocols.

Background

Our research focuses on novel ciphertext-only analysis and contributes to advancing further research on lightweight cryptosystems. Fault analysis is a significant method for evaluating the security of ciphers. It allows the attackers to recover secret keys through injecting faults. Since Boneh et al. successfully employed fault analysis to decode the RSA cryptosystem, various fault analyses have been explored, including the statistical fault analysis (SFA), the differential fault analysis (DFA), the algebraic fault analysis (AFA), and the meet-in-the-middle fault analysis (MFA) et al. It is important to note that these methods are based on the ciphertext-only attacking assumption.

The TweGIFT lightweight tweakable cryptosystem, proposed by Chakraborti et al. in 2021, aims to protect the data security of Internet of Things (IoT) devices in smart cities, intelligent manufacturing, and smart agriculture. TweGIFT is the core block cipher for LOTUS/LOCUS and ESTATE, which are candidate algorithms in the International Lightweight Cryptography Competition. Hence, the research on the security of the TweGIFT cipher is significant.

This paper primarily focuses on the miss-in-the-middle strategy and the ciphertext-only attacking assumption, explores the new types of ciphertext-only attack and two innovative composite distinguishers. We present the miss-in-the-middle statistical fault analysis (MSFA) for TweGIFT, rely on the miss-in-the-middle strategy and statistical fault analysis by injecting random nibble-oriented faults, collecting fault-affected ciphertexts, and analyzing statistical properties to recover the secret key. Furthermore, we design two novel distinguishers of Cosine Distance-Hamming Weight (COS-HW) and Cosine Distance-Maximum Likelihood Estimation (COS-MLE). It only requires at least 184 faults to recover the 128-bit secret key of all versions of TweGIFT. Compared to traditional SFA and classical distinguishers such as SEI, HW, MLE, and WD-MLE, the MSFA method, combined with the COS-HW and COS-MLE distinguishers, enables deeper fault injection locations, reduces the number of faults by 77.77% and achieves a success of over 99%.

This research was supported in part by the National Key Research and Development Program of China (Grant No. 2020YFA0712300), the National Natural Science Foundation of China (Grant No. 62472286), the Shanghai Natural Science Foundation (Grant No. 24ZR1401300), the Shanghai Sailing Program (Grant No. 23YF1401000), and the Fundamental Research Funds for the Central Universities (Grant No. 223202D-25).