SHACAL-2 算法的差分故障攻击

魏悦川[①] 李 琳[②④] 李瑞林[②] 李 超[①②③]
[①](国防科技大学计算机学院 长沙 410073)
[②](国防科技大学理学院 长沙 410073)
[③](中国科学院软件所信息安全国家重点实验室 北京 100039)
[④](西安陆军学院 西安 710108)

摘 要：该文采用面向字的随机故障模型，结合差分分析技术，评估了SHACAL-2算法对差分故障攻击的安全性。结果显示：SHACAL-2算法对差分故障攻击是不免疫的。恢复出32 bit子密钥的平均复杂度为8个错误密文，完全恢复出512 bit密钥的复杂度为128个错误密文。

关键词：分组密码；SHACAL-2；差分故障攻击
中图分类号：TN918 **文献标识码**： A **文章编号**：1009-5896(2010)02-0318-05
DOI: 10.3724/SP.J.1146.2008.01575

Differential Fault Analysis on SHACAL-2

Wei Yue-chuan[①] Li Lin[②④] Li Rui-lin[②] Li Chao[①②③]
[①](*College of Computer Science of National University of Defense Technology*, *Changsha* 410073, *China*)
[②](*Science College of National University of Defense Technology*, *Changsha* 410073, *China*)
[③](*State Key Laboratory of Information Security*, *Graduate University of Chinese Academy of Sciences*, *Beijing* 100039, *China*)
[④](*Xi'an Army Command College*, *Xi'an* 710108, *China*)

Abstract: By using word-oriented fault model and the technique of differential cryptanalysis, the security of SHACAL-2 against differential fault analysis is evaluated. Result shows that SHACAL-2 is not immune to such kind of attack. 8 faulty ciphertexts can recover a sub key of 32 bit on average and 128 faulty ciphertexts are needed to recover all the 512 bit keys.

Key words: Block cipher; SHACAL-2; Differential fault analysis

1 引言

2003年，欧洲NESSIE(New European Schemes for Signatures, Integrity and Encryption)计划宣布了新的分组密码标准算法：Rijndael算法，MISTY1算法，Camellia算法和SHACAL-2算法，其中SHACAL-2算法的分组和密钥长度最长，安全性被认为最高，它是由标准Hash函数SHA-256演变而来的，分组长度为256 bit，密钥长度为512 bit，迭代次数为64轮。

故障攻击是侧信道攻击方法的一种，由Boneh等人于1996年首次提出[1]，他们利用密码计算过程中的错误，来攻击基于RSA-CRT实现的签名方案。一般而言，硬件设备均能正确地执行各种密码运算，但在外界干扰的情况下，密码模块的运算过程中可能出现硬件故障或运算错误，利用这些故障行为或错误信息恢复密钥的攻击即是故障攻击。这种攻击方法一经提出立即引起了人们的广泛关注，并展示出了其对密码体制安全性的极大破坏性。1997年，Biham 等人在故障攻击的基础上结合差分分析技术，提出了差分故障攻击的概念[2]，并成功地攻击了DES算法，显示了DES类密码所采用的Feistel结构对该攻击的不免疫性。此后，差分故障攻击又成功地攻击了椭圆曲线加密体制、3DES算法、SMS4算法、AES-128算法、ARIA算法、CLEFIA算法、KeeLoq算法和SHACAL-1算法等[3–10]。由于诱导故障多是针对硬件设备(如智能卡等)，所以故障攻击大多应用于密码算法的硬件实现，且由于其简单易行又十分有效，故障攻击已成为目前最有效的侧信道攻击方法之一。

诱导故障的本质其实就是引入差分，可以说差分故障攻击是一种特殊的差分攻击，只是差分的选取可以出现在加密的中间状态，但攻击原理仍然是

2008-11-27收到，2009-11-04改回
国家自然科学基金(60803156)和信息安全国家重点实验室开放基金(01-07)资助课题
通信作者：魏悦川 wych004@163.com

差分分析。

本文分析了SHACAL-2算法的加密特性，采用面向字的故障诱导模型，结合差分分析技术，对该算法进行了攻击。攻击结果表明：SHACAL-2不抵抗差分故障攻击。平均需要8个错误密文就可以恢复出1个32 bit子密钥，平均需要128个密文就可以完全恢复出512 bit密钥。

2　SHACAL-2算法

SHACAL-2 密码[11]提案基于单向Hash函数SHA-256中压缩函数的加密模式设计。该算法以32比特的字为处理单位，分别用 $+$，$-$ 表示模2^{32}加和模2^{32}减；用 $\oplus$ 和 $\&$ 表示两个32 bit操作数的异或运算和与运算，用 $\neg$ 表示取反运算，用 $R_i(X)$ 和 $S_i(X)$ 表示将32 bit字 X 向右移动 i 比特和向右循环移动 i 比特。算法中的基本函数有以下4个，定义如下，其中Ch和Maj分别被称为选择函数和主函数。

$$\Sigma_0(X)=S_2(X)\oplus S_{13}(X)\oplus S_{22}(X)$$

$$\Sigma_1(X)=S_6(X)\oplus S_{11}(X)\oplus S_{25}(X)$$

$$\mathrm{Ch}(X,Y,Z)=(X\,\&\,Y)\oplus(\neg X\,\&\,Z)$$

$$\mathrm{Maj}(X,Y,Z)=(X\,\&\,Y)\oplus(X\,\&\,Z)\oplus(Y\,\&\,Z)$$

在如图1所示的一轮加密过程中，明文 P 被存储在8个32比特变量 A_0，B_0，C_0，D_0，E_0，F_0，G_0 和 H_0 中，随后进行64次迭代，相应的密文 C 由 A_{64}，B_{64}，C_{64}，D_{64}，E_{64}，F_{64}，G_{64} 和 H_{64} 级联组成。一轮加密的函数更新过程如下：

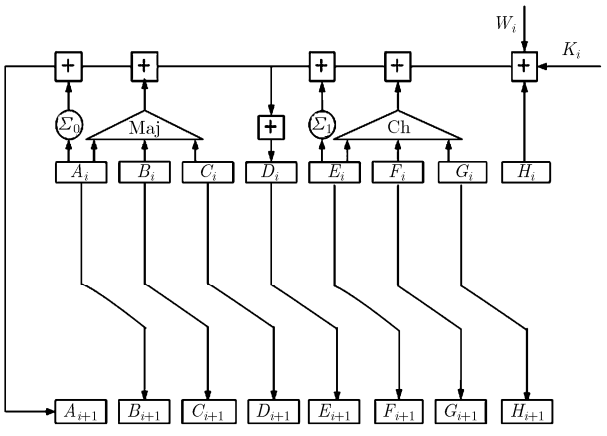

图1 SHACAL-2算法的一轮加密结构

$$T1_{i+1}=H_i+\Sigma_1(E_i)+\mathrm{Ch}(E_i,F_i,G_i)+K_i+W_i$$

$$T2_{i+1}=\Sigma_0(A_i)+\mathrm{Maj}(A_i,B_i,C_i)$$

$$H_{i+1}=G_i;G_{i+1}=F_i;F_{i+1}=E_i;E_{i+1}=D_i+T1_{i+1}$$

$$D_{i+1}=C_i;C_{i+1}=B_i;B_{i+1}=A_i;A_{i+1}=T1_{i+1}+T2_{i+1}$$

这里 W_i 是常数；Ch，Maj，Σ_0，Σ_1 都是具有 32 bit 字输入，32 bit 字输出的函数。

SHACAL-2算法的512 bit种子密钥 K 被存储成16个32 bit密钥字 $K_0,K_1,\cdots,K_{15}$，以下过程将这16个原始密钥字扩展成64个子密钥字（$i=16,17,\cdots,63$）：

$$K_i=\sigma_1(K_{i-2})+K_{i-7}+\sigma_0(K_{i-15})+K_{i-16}$$

σ_0 和 σ_1 是线性函数：

$$\sigma_0(X)=S_7(X)\oplus S_{18}(X)\oplus R_3(X),$$

$$\sigma_1(X)=S_{17}(X)\oplus S_{19}(X)\oplus R_{10}(X)$$

3　SHACAL-2算法的差分故障攻击

3.1　故障模型与基本假设

根据SHACAL-2算法的结构和轮函数的具体形式，为了充分利用差分传播性质，本文采用面向字的故障诱导模型，其基本假设为：

(1)攻击者可以每次诱导加密过程中间状态的某个存储单元发生故障，但是他不知道具体的错误值；

(2)对于同一个明文 P 而言，攻击者可以获得在同一个密钥 K 作用下的正确密文 C 和错误密文 C^*。

3.2　攻击的基本原理

SHACAL-2算法没有采用传统的Feistel结构，但是可以等价刻划为一类广义非平衡Feistel结构：即每次迭代中仅有两个中间状态进行更新，其他保持不变。SHACAL-2算法是以32 bit字为单位进行运算的，所以本文采用面向字的差分故障诱导模型。在进行攻击时，首先，获得一个明文和相应的正确密文。其次，对上述明文进行加密，并对倒数第2轮进行随机故障诱导，获得所需的错误密文。由于SHACAL-2算法未采用S盒，而是一些简单的算术运算，所以本文在非线性部件中引入差分，根据加密算法的特点，利用差分分析技术恢复出一个未知状态，将未知状态值代入迭代函数中的代数方程，恢复出最后一轮的轮密钥。利用该轮密钥对最后一轮进行解密，由解密获得倒数第2轮的输出值，此时，使用类似方法可依次攻击算法的其他轮，从而恢复出16轮轮密钥。最后，利用SHACAL-2密钥扩展算法的特点恢复出512 bit的种子密钥。

3.3　攻击的详细步骤

本部分将详细介绍攻击过程。本文方案假设对同一个明文 P 进行多次故障诱导。而在实际的攻击方案中攻击者完全可以随机地选择明文。只要他可以得到一个正确密文和一个对应的包含有他所需要故障类型的错误密文即可。

我们注意到，在加密过程中，每一轮的8个字中有两个字被更新，密钥的参与仅仅是在变量 $T1$ 处，将 $T1$ 的最后一轮更新写成代数方程，即：

$$T1_{64} = H_{63} + \Sigma_1(E_{63}) + \mathrm{Ch}(E_{63},F_{63},G_{63}) + K_{63} + W_{63} \tag{1}$$

由于密文的8个字都是已知的，根据第2节中的迭代关系可知，式(1)中只有 H_{63} 和 K_{63} 是未知的，其中

$$T1_{64} = A_{64} - T2_{64} = A_{64} - \Sigma_0(B_{64}) - \mathrm{Maj}(B_{64},C_{64},D_{64}) \tag{2}$$

若 H_{63} 已知，则可将密钥 K_{63} 表示为

$$K_{63} = T1_{64} - H_{63} - \Sigma_1(F_{64}) - \mathrm{Ch}(F_{64},G_{64},H_{64}) - W_{63} \tag{3}$$

而 $H_{63} = G_{62}$ 这启示我们对倒数第2轮的输入进行故障诱导，以求出 H_{63} 的值。

对SHACAL-2算法的一轮攻击过程如下：

步骤1 随机选择一个明文 X 获得其在密钥 K 作用下的正确密文 Y 。

步骤2 对第63轮的输入进行单字的故障诱导，要求故障发生在32 bit字 E 的位置(只有这样的故障诱导才能有效)，不妨假设其值为 δ ，并记录得到的密文 Y^* 。将第63轮 $T1$ 的更新函数表示如下：

$$\begin{aligned} T1_{63} &= H_{62} + \Sigma_1(E_{62}) + \mathrm{Ch}(E_{62},F_{62},G_{62}) \\ &\quad + K_{62} + W_{62} \\ \widehat{T1}_{63} &= H_{62} + \Sigma_1(E_{62}\oplus\delta) + \mathrm{Ch}(E_{62}\oplus\delta,F_{62},G_{62}) \\ &\quad + K_{62} + W_{62} \end{aligned}$$

步骤3 计算模减差分

$$\begin{aligned} \Delta = \widehat{T1} - T1 &= \mathrm{Ch}(E_{62}\oplus\delta,F_{62},G_{62}) - \mathrm{Ch}(E_{62},F_{62}, \\ &\quad G_{62}) + \Sigma_1(E_{62}\oplus\delta) - \Sigma_1(E_{62}) \\ &= \mathrm{Ch}(\widehat{G_{64}},\widehat{H_{64}},\widehat{H_{63}}) \\ &\quad - \mathrm{Ch}(G_{64},H_{64},H_{63}) + \Sigma_1(\widehat{G_{64}}) - \Sigma_1(G_{64}) \end{aligned}$$

依据迭代关系，有：$\widehat{G_{64}} \oplus G_{64} = \delta$ ，$\widehat{H_{64}} \oplus H_{64} = 0$ ，$\widehat{H_{63}} \oplus H_{63} = 0$ ，故

$$\begin{aligned} \Delta = &\ ((G_{64}\oplus\delta)\,\&\,H_{64}) \oplus (\neg(G_{64}\oplus\delta)\,\&\,H_{63}) \\ &- ((G_{64}\,\&\,H_{64}) \oplus (\neg G_{64}\,\&\,H_{63})) \\ &+ (\Sigma_1(G_{64}\oplus\delta) - \Sigma_1(G_{64})) \end{aligned} \tag{4}$$

另一方面，Δ 的值可以通过计算获得。

$$\begin{aligned} T1_{63} &= A_{63} - T2_{63} \\ &= A_{63} - \Sigma_0(B_{63}) - \mathrm{Maj}(B_{63},C_{63},D_{63}) \\ &= B_{64} - \Sigma_0(C_{64}) - \mathrm{Maj}(C_{64},D_{64},E_{64} - T1_{64}) \end{aligned}$$

$T1_{64}$ 可以通过式(2)获得。同理可以得到 $\widehat{T1}_{63}$，进而获得 Δ 值。注意到式(4)中只有 H_{63} 未知。

步骤4 继续对倒数第2轮的输入进行故障诱导，获得相应的 (δ,Δ) 约束关系对，通过搜索来确定 H_{63} 的值，我们期望通过有限次诱导，就可以将 H_{63} 唯一确定下来。将 H_{63} 代入式(2)，即可获得轮密钥 K_{63} 。

上述攻击过程的目标是获得最后一轮的轮密钥，为实现对整个密码算法的攻击，利用 K_{63} 对正确密文进行解密一轮，得到第63轮的输出值

$$(A_{63},B_{63},C_{63},D_{63},E_{63},F_{63},G_{63},H_{63})$$

利用与上述相同的攻击方法在第62轮的输入诱导故障，可以获得第63轮的轮密钥 K_{62} 。将攻击依次进行下去，分别获得64轮的轮密钥。

4 实验结果

在普通的PC机上进行实验，其中通过故障诱导得到错误密文的过程是利用计算机模拟的，在位置 E 上进行诱导错误，对于SHACAL-2算法，平均需要8个错误密文就可以恢复出1个32 bit子密钥，理论上需要64×8个密文就可以完全恢复SHACAL-2变种的所有轮密钥，提取出前16个轮密钥 $W_0,W_1,\cdots,W_{15}$ ，即为原始的512 bit种子密钥。

实验采用512 bit的种子密钥为0x61626380，0，0，0，0，0，0，0，0，0，0，0，0，0，0，0，0x00000018，明文选取为0x6a09e667，0xbb67ae85，0x3c6ef372，0xa54ff53a，0x510e527f，0x9b05688c，0x1f83d9ab，0x5be0cd19。通过模拟，平均进行8次诱导后，能够唯一地确定下密钥。表1是其中一组攻击数据。

观察发现，SHACAL-2的密钥扩展算法也是非平衡的Feistel型结构，它是用4个密钥字对一个密钥字进行更新，再加上攻击的顺序为从后至前，这使得我们的攻击复杂度可以继续降低。依据密钥扩展算法，有 $K_{i-16} = K_i - \sigma_1(K_{i-2}) - K_{i-7} - \sigma_0(K_{i-15})$ ，给定连续16个字就可以求解之前的连续16个字，依次类推，只需要恢复出 $K_{48},K_{49},\cdots,K_{63}$ ，就可以解出最初的种子密钥 $K=(K_0,K_1,\cdots,K_{15})$ 。因此，在本文的攻击中，只需要16×8个密文就可以完全恢复出SHACAL-2变种的512 bit种子密钥。表2是通过实验恢复出的密钥。

5 实验结果分析

以上实验结果表明，由于迭代函数中选择函数的使用，SHACAL-2不抵抗差分故障攻击，这是因为在对SHACAL-2的攻击中，依据约束关系对 (δ,Δ) 寻找 H_{63}，实际上是求解满足代数方程：

$$\begin{aligned} \Delta = &\ ((G_{64}\oplus\delta)\,\&\,H_{64}) \oplus (\neg(G_{64}\oplus\delta)\,\&\,H_{63}) \\ &- ((G_{64}\,\&\,H_{64}) \oplus (\neg G_{64}\,\&\,H_{63})) \\ &+ (\Sigma_1(G_{64}\oplus\delta) - \Sigma_1(G_{64})) \end{aligned}$$

如果可以解得 H_{63}，则根据更新函数 $T1_{64}$ 的表达式即可恢复出最后一轮子密钥 K_{63} 。需要多少对 (δ,Δ) 可求解出唯一的 H_{63}，成为攻击复杂度的主要来源。通过计算机模拟，进行有限次诱导，可以唯

表1 SHACAL-2密码差分故障攻击的实验结果(一)

明文	6a09e667bb67ae853c6ef372a54ff53a510e527f9b05688c1f83d9ab5be0cd19
正确密文	506e3058d39a216504d24d6cb85e2ce95ef50f24fb121210948d25b6961f4894
错误密文1	409abb845303252304d24d6cb85e2ce9f5a841fd7a7b15ce760a53ec961f4894
错误密文2	14b9ded343b5daec04d24d6cb85e2ce9b9ad47cb6b2dcb973e0b9af8961f4894
错误密文3	8920e78f 69572ef804d24d6cb85e2ce9e64bea0490cf1fa38d9f51d2961f4894
错误密文4	34f46eb540131a1304d24d6cb85e2ce99e5a672f 678b0abe70e1059e 961f4894
错误密文5	a83d96774bf83cf604d24d6cb85e2ce9c0526fb7 73702da1f86f1d90961f4894
错误密文6	6b6e136f 86a70b8a04d24d6cb85e2ce93b72b7c6ae1efc358af1737d961f4894
错误密文7	efd56e15ceaa052304d24d6cb85e2ce9a7097589f621f5ce d2932cff 961f4894
错误密文8	ccfc36779016e18004d24d6cb85e2ce97fcd1df7b78ed22b197d9281961f4894
中间状态	b21bad3d
恢复的第64轮密钥	12b1edeb
第64轮密钥	12b1edeb

表2 SHACAL-2密码差分故障攻击的实验结果(二)

48–63轮密钥	fb3e89cbcc7617dbb9e66c34a999366784badeddc2 1462bc1487472cb20f7a99ef57b9cdebe6b2389fe3 095e78bc8d4ba43fcf15668b2ff8eeaba2cc12b1edeb	由差分故障攻击恢复
32–47轮密钥	93f5997f3b68ba73aff4ffc1f10a5c620a8b399672 af830a9409e33e246415229f47bf 94f0a64f5a3e24 6a7927333ba30c4763f2840abf277a290d5d065c43da	由密钥扩展算法恢复
16–31轮密钥	61626380000f00007da86405600003c63e9d7b7801 83fc0012dcbfdbe2e2c38ec 8215c1ab 73679a2e5bc 390932663c5b9d209d67ec8726cb702138a4d3b7973b	由密钥扩展算法恢复
0–15轮密钥(种子密钥)	6162638000000000000000000000000000000000 0000000000000000000000000000000000000000 0000000000000000000000000000000000000018	由密钥扩展算法恢复

一地确定中间状态 H_{63}。这表明：在设计分组密码的非线性函数时，必须考虑到差分故障攻击诱导的代数方程的解的情况。

6　结论

差分故障攻击在实际中高效可行，这必然对算法的实现提出更高的要求。本文给出了一种对SHACAL-2算法面向字的差分故障攻击，指出SHACAL-2不抵抗差分故障攻击。实验表明：平均需要8个错误密文就可以恢复出1个32 bit子密钥，平均需要128个密文就可以完全恢复出全部的512 bit种子密钥。由此可见，分组密码中非线性函数的性质可以极大地影响整个算法的安全性。

参 考 文 献

[1] Boneh D, DeMillo R A, and Lipton R J. On the importance of checking cryptographic protocols for faults. EUROCRYPT'97, Konstanz, Germany, 1997, LNCS 1233: 37–51.

[2] Biham E and Shamir A. Differential fault analysis of secret key cryptosystems. CRYPTO'97, California, USA, 1997, LNCS 1294: 513–525.

[3] Biehl I, Meyer B, and Müller V. Differential fault attacks on elliptic curve cryptosystems. CRYPTO 2000, California, USA, 2000, LNCS 1880: 131–146.

[4] Hemme L. A differential fault attack against early rounds of (Triple-) DES. Cryptographic Hardware and Embedded Systems-CHES 2004. Boston, 2004, LNCS 3156: 254–267.

[5] 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): 1596–1602.
Zhang Lei and Wu Wen-ling. Differential fault analysis on SMS4. *Chinese Journal of Computers*, 2006, 29(9): 1596–1602.

[6] Li Wei, Gu Da-wu, and Li Juan-ru. Differential fault analysis on the ARIA algorithm. *Information Sciences*, 2008, 178(19): 3727–3737.

[7] 李玮, 谷大武. 基于密钥编排故障的 SMS4 算法的差分故障分析. 通信学报, 2008, 29(10): 135–142.

Li Wei and Gu Da-wu. Differential fault analysis on the SMS4 cipher by inducing faults to the key schedule. *Journal of China Institute of Communications*, 2008, 29(10): 135–142.

[8] Chen Hua, Wu Wen-ling, and Feng Deng-guo. Differential fault analysis on CLEFIA. International Conference on Information and Communication Security-ICICS 2007, Zhengzhou, China, 2007, LNCS 4861: 284–295.

[9] Piret G and Quisquater J J. A differential fault attack technique against SPN Structures, with Application to the AES and KHAZAD. Cryptographic Hardware and Embedded Systems-CHES 2003.Cologne, 2003, LNCS 2779: 77–88.

[10] 李琳, 李瑞林, 谢端强, 李超. KeeLoq 和 SHACAL-1 算法的差分故障攻击. 武汉大学学报, 2008, 54(5): 507–512.
Li Lin, Li Rui-lin, Xie Duan-qiang, and Li Chao. Differential Fault Analysis on Keeloq and SHACAL-1. *Journal of Wuhan University*, 2008, 54(5): 507–512.

[11] NIST. FIPS-180-2: Secure Hash Standard(SHS). 2002.

魏悦川： 女，1982年生，博士生，研究方向为编码密码理论及其应用.

李 超： 男，1966年生，博士生导师，教授，研究方向为编码密码理论及其应用.