SMS4 密码算法的差分故障攻击

张 蕾 吴文玲

(中国科学院软件研究所信息安全国家重点实验室 北京 100080)

(中国科学院研究生院信息安全国家重点实验室 北京 100049)

摘 要 SMS4 是用于 WAPI 的分组密码算法,是国内官方公布的第一个商用密码算法.由于公布时间不长,关于它的安全性研究尚没有公开结果发表.该文研究 SMS4 密码算法对差分故障攻击的安全性.攻击采用面向字节的随机故障模型,并且结合了差分分析技术.该攻击方法理论上仅需要 32 个错误密文就可以完全恢复出 SMS4 的 128 比特种子密钥.因为实际中故障发生的字节位置是不可能完全平均的,所以实际攻击所需错误密文数将略大于理论值;文中的实验结果也验证了这一事实,恢复 SMS4 的 128bit 种子密钥平均大约需要 47 个错误密文.文章结果显示 SMS4 对差分故障攻击是脆弱的.为了避免这类攻击,建议用户对加密设备进行保护,阻止攻击者对其进行故障诱导.

关键词 SMS4 密码算法;差分分析;差分故障攻击;故障模型;差分表

中图法分类号 TP309

Differential Fault Analysis on SMS4

ZHANG Lei WU Wen-Ling

(*State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing* 100080)

(*State Key Laboratory of Information Security, Graduate University of Chinese Academy of Sciences, Beijing* 100049)

Abstract SMS4 is the block cipher used in WAPI, and it is also the first commercial block cipher disclosed by the government. Since it was disclosed only a short time ago, on its security, there has been no published paper at present. In this paper the strength of SMS4 against the differential fault attack is examined. The authors use the byte-oriented fault model, and take advantage of the differential analysis as well. Theoretically, the 128bit master key for SMS4 can be obtained by using 32 faulty ciphertexts. But in practice, for the fact that the byte position where the fault happens isn't equally distributed, the number of faulty ciphertexts needed will be a little bigger than the theoretical value. The attack experiment result validates this fact too. The result shows that only need average 47 faulty ciphertexts to recover the 128bit keys for SMS4. So SMS4 is vulnerable to differential fault attack. To avoid this kind of attack, the authors suggest that the encryption device should be protected to prevent the adversary from deducing faults.

Keywords SMS4; differential analysis; differential fault attack; fault model; difference distribution table

---

收稿日期:2006-03-28;**修改稿收到日期**:2006-06-06.**本课题得到国家自然科学基金**(60373047,90604036,60503014)**和国家"九七三"重点基础研究发展规划项目基金**(2004CB318004)**资助.****张 蕾**,女,1981 年生,博士研究生,目前主要从事分组密码的安全分析研究.E-mail:zhanglei1015@is.iscas.ac.cn.**吴文玲**,女,1966 年生,博士,研究员,博士生导师,目前主要从事分组密码的设计与分析、分组密码工作模式和可证明安全理论的研究.

1 引　言

SMS4 密码算法[1]作为 WAPI 相关密码算法，是国内官方公布的第一个商用密码算法. 该密码算法的公布有利于对其安全状态的了解，同时通过密码分析将更好地理解该算法的密码特性. 本文研究 SMS4 密码算法对差分故障攻击的安全性.

“故障攻击”的概念是 1996 年由 Boneh 等人首次提出的，是对 RSA 公钥密码体制的新型攻击方法[2]，该方法利用了密码计算过程中的错误. 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 和 Shamir 将这种攻击方法应用于对称密码体制，首次提出了“差分故障攻击”的概念[3]，并成功地攻击了 DES 算法. 此后研究人员提出了各种不同的故障攻击方法，成功攻击了多种密码体制，如 ECC 公钥体制[4]、AES 算法[5~8]、3DES 算法[9]以及 RC4 算法[10,11]等.

本文给出的 SMS4 算法的差分故障攻击就是利用面向字节的随机故障模型，结合差分分析实现的. 面向字节的随机故障模型是指当对设备存储中间值的存储单元进行故障诱导时，将得到任意的单字节错误. 利用该攻击方法，理论上仅需要 32 个错误密文就可以完全恢复出 SMS4 的 128bit 加密密钥.

本文第 2 节简单介绍 SMS4 算法；第 3 节描述差分故障攻击的基本思想；第 4 节详细介绍对 SMS4 的差分故障攻击；第 5 节列出攻击实验及实验结果；第 6 节总结全文.

2 SMS4 算法简介

SMS4 算法的分组长度和密钥长度均为 128bit. 加密算法与密钥扩展算法都采用 32 轮非线性迭代结构. 解密算法与加密算法的结构相同，只是轮密钥的使用顺序相反，解密轮密钥是加密轮密钥的逆序. 限于篇幅，我们下面仅简单介绍加密算法.

2.1 SMS4 的加密算法

明文输入为 $(X_0,X_1,X_2,X_3)\in(F_2^{32})^4$，密文输出为 $(Y_0,Y_1,Y_2,Y_3)\in(F_2^{32})^4$，轮密钥为 $rk_i\in F_2^{32}$ $(i=0,1,\cdots,31)$. 该算法的加密变换为

$$X_{i+4}=F(X_i,X_{i+1},X_{i+2},X_{i+3},rk_i)=X_i\oplus T(X_{i+1}\oplus X_{i+2}\oplus X_{i+3}\oplus rk_i),\quad i=0,1,\cdots,31.$$

$$(Y_0,Y_1,Y_2,Y_3)=R(X_{32},X_{33},X_{34},X_{35})=(X_{35},X_{34},X_{33},X_{32}).$$

轮函数：SMS4 算法使用的轮函数 F 定义为

$$F(X_i,X_{i+1},X_{i+2},X_{i+3},rk_i)=X_i\oplus T(X_{i+1}\oplus X_{i+2}\oplus X_{i+3}\oplus rk_i).$$

合成置换 T：$F_2^{32}\to F_2^{32}$ 是一个可逆变换，由非线性变换 τ 和线性变换 L 复合而成，即 $T(\cdot)=L(\tau(\cdot))$. 其中：

非线性变换 τ 由 4 个并行的 S 盒构成. 设输入为 $\boldsymbol{A}=(a_0,a_1,a_2,a_3)\in(F_2^8)^4$，输出为 $\boldsymbol{B}=(b_0,b_1,$

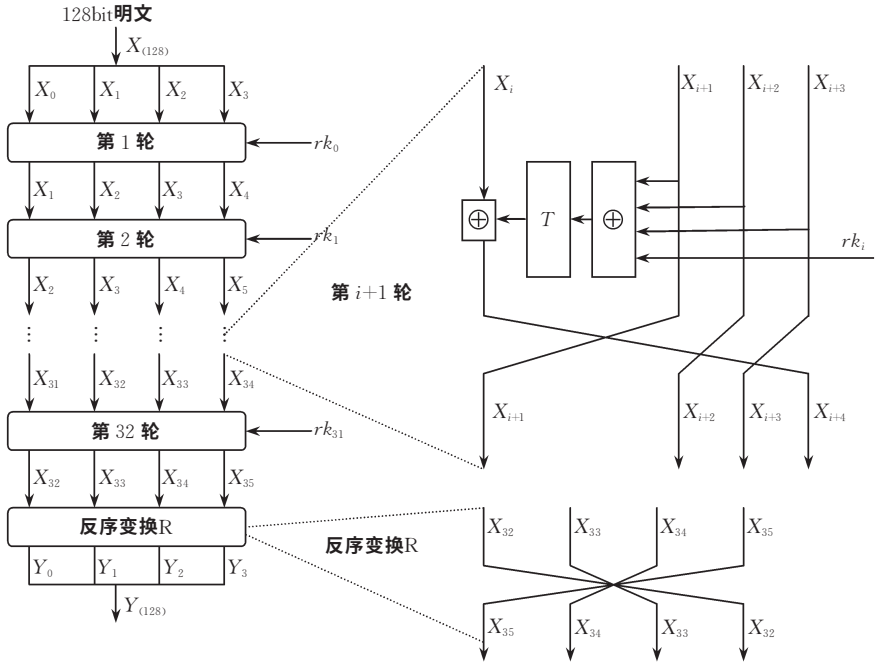

图 1　SMS4 加密算法整体结构

$b_2,b_3)\in(F_2^8)^4$,则$(b_0,b_1,b_2,b_3)=\tau(\boldsymbol{A})=(S(a_0),S(a_1),S(a_2),S(a_3))$.

线性变换 L. 以非线性变换 τ 的输出作为输入. 设输入为 $B\in F_2^{32}$,输出为 $C\in F_2^{32}$,则

$$C=L(B)=B\oplus(B\lll 2)\oplus(B\lll 10)\oplus(B\lll 18)\oplus(B\lll 24).$$

反序变换 R.

$$R(A_0,A_1,A_2,A_3)=(A_3,A_2,A_1,A_0),\quad A_i\in F_2^{32}(i=0,1,2,3).$$

SMS4 算法的加密过程如图 1 所示.

2.2 密钥扩展算法

加密算法的轮密钥由加密密钥通过密钥扩展算法生成. 加密密钥 $\boldsymbol{MK}=(MK_0,MK_1,MK_2,MK_3)$,$MK_i\in F_2^{32}(i=0,1,2,3)$;令 $K_i\in F_2^{32}(i=0,1,\cdots,35)$,轮密钥为 $rk_i\in F_2^{32}(i=0,1,\cdots,31)$,则轮密钥生成方法如下:

首先,$(K_0,K_1,K_2,K_3)=(MK_0\oplus FK_0,MK_1\oplus FK_1,MK_2\oplus FK_2,MK_3\oplus FK_3)$;

然后,对 $i=0,1,\cdots,31$:$rk_i=K_{i+4}=K_i\oplus T'(K_{i+1}\oplus K_{i+2}\oplus K_{i+3}\oplus CK_i)$.

其中 T' 变换与轮函数中的 T 变换基本相同,只将其中的线性变换 L 修改为

$$L'(B)=B\oplus(B\lll 13)\oplus(B\lll 23);$$

$\boldsymbol{FK}=(FK_0,FK_1,FK_2,FK_3)$ 为系统参数,其值用 16 进制表示为

$FK_0=$(A3B1BAC6), $FK_1=$(56AA3350),

$FK_2=$(677D9197), $FK_3=$(B27022DC).

$\boldsymbol{CK}=(CK_0,CK_1,\cdots,CK_{31})$ 为 32 个固定参数,其值请参见相关标准文献[1].

3 差分故障攻击概述

3.1 故障模型和基本假设

本文采用的故障模型是面向字节的随机故障模型,其基本假设为

(1) 攻击者每次可以诱发存储中间值的存储单元发生任意的单字节错误,但是攻击者不知道错误发生的字节位置以及具体的错误值.

(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 攻击基本思想

对密码体制进行故障攻击的基本思想如下:首先选择明文,对加密过程进行故障诱导,分别获得该明文对应的正确密文和错误密文(此阶段称为故障诱导与收集数据),最后对收集到的数据进行分析,恢复出密钥.

本文所给攻击方法的基本思想与上述相同,其基本过程如下:

1. 选择明文,获得该明文对应的正确密文.

2. 从算法的最后一轮开始,对加密过程进行随机故障诱导,获得所需要的错误密文;利用差分分析,恢复出该轮子密钥的部分字节信息;重复这一过程,直至完全恢复出该轮子密钥.

3. 对算法的倒数第 2 轮进行随机故障诱导,获得所需的错误密文. 利用步 2 中已经恢复出的最后一轮子密钥,对最后一轮进行解密;由解密得到的中间值,结合差分分析,恢复出倒数第 2 轮子密钥的部分字节信息;重复这一过程,直至完全恢复出倒数第 2 轮的轮密钥.

4. 同样的,使用上述相同方法进行随机故障诱导,依次攻击该算法的倒数第 3 轮、倒数第 4 轮,恢复出这些轮的轮密钥.

5. 使用已经恢复出的后 4 轮子密钥,根据密钥扩展算法,逆向计算出各轮子密钥以及加密密钥的值.

4 攻击过程详述

4.1 基本记号和符号

记 $(X_0,X_1,X_2,X_3)\in(F_2^{32})^4$ 为明文输入,$(Y_0,Y_1,Y_2,Y_3)\in(F_2^{32})^4$ 为密文输出,$rk_i\in F_2^{32}(i=0,1,2,\cdots,31)$ 为第 i 轮子密钥.

记 $(X_i,X_{i+1},X_{i+2},X_{i+3})\in(F_2^{32})^4(i=0,1,\cdots,31)$ 为第 $i+1$ 轮的输入;$(X_{i+1},X_{i+2},X_{i+3},X_{i+4})\in(F_2^{32})^4(i=0,1,\cdots,31)$ 为第 $i+1$ 轮的输出.

记 $\boldsymbol{A}_i=(a_{0,i},a_{1,i},a_{2,i},a_{3,i})\in(F_2^8)^4(i=1,2,\cdots,32)$ 为第 i 轮 S 盒的输入;$\boldsymbol{B}_i=(b_{0,i},b_{1,i},b_{2,i},b_{3,i})\in(F_2^8)^4(i=1,2,\cdots,32)$ 为第 i 轮 S 盒的输出,同时也是第 i 轮线性变换 L 的输入;$\boldsymbol{C}_i=(c_{0,i},c_{1,i},c_{2,i},c_{3,i})\in(F_2^8)^4(i=1,2,\cdots,32)$ 为第 i 轮线性变换 L 的输出.

记 $\Delta\boldsymbol{A}_i=(\Delta a_{0,i},\Delta a_{1,i},\Delta a_{2,i},\Delta a_{3,i})(i=1,2,\cdots,32)$ 为第 i 轮 S 盒的输入差分;$\Delta\boldsymbol{B}_i=(\Delta b_{0,i},\Delta b_{1,i},\Delta b_{2,i},\Delta b_{3,i})(i=1,2,\cdots,32)$ 为第 i 轮 S 盒的输出差分,同时也是第 i 轮线性变换 L 的输入差分;$\Delta\boldsymbol{C}_i=(\Delta c_{0,i},\Delta c_{1,i},\Delta c_{2,i},\Delta c_{3,i})(i=1,2,\cdots,32)$ 为第 i 轮线性变换 L 的输出差分.

记 $e_i\in F_2^8(i=1,2,\cdots,32)$ 为在第 i 轮加密之前诱发 $X_i\in F_2^{32}(i=1,2,\cdots,32)$ 中发生的任意单字节故障.

对于 SMS4 算法中第 i 轮的 S 盒变换,定义:

$$IN(\Delta a_{j,i},\Delta b_{j,i})=\{z_{j,i}\mid z_{j,i}\in F_2^8,S(z_{j,i})\oplus S(z_{j,i}\oplus\Delta a_{j,i})=\Delta b_{j,i}\},$$

$j=0,1,2,3.$

4.2　攻击过程细节

本部分将详细介绍攻击过程. 为简单记，本方案假设对同一个明文 X 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.

4.2.1　攻击 SMS4

攻击过程如下：

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.**

2. **攻击最后一轮（第 32 轮），步骤如下（图 2 为其示意图）：**

2.1. **对明文 X 在密钥 K 作用下加密，当进行第 32 轮加密之前，诱导 X_{32} 中产生单字节的随机故障，并记由此得到的错误密文为 $\boldsymbol{Y}^*=(Y_0^*,Y_1^*,Y_2^*,Y_3^*)$.**

2.2. **根据反序变换 R 的定义，由密文逆向计算出进入 R 变换的输入值，即第 32 轮的输出值：**

$(X_{32},X_{33},X_{34},X_{35})=R^{-1}(Y_0,Y_1,Y_2,Y_3)=(Y_3,Y_2,Y_1,Y_0)$，

$(X_{32}^*,X_{33}^*,X_{34}^*,X_{35}^*)=R^{-1}(Y_0^*,Y_1^*,Y_2^*,Y_3^*)=(Y_3^*,Y_2^*,Y_1^*,Y_0^*)$.

2.3. **由于只对 X_{32} 进行了单字节故障诱导，所以有 $\Delta X_{31}=\Delta X_{33}=\Delta X_{34}=0$，$\Delta X_{32}=(Y_3\oplus Y_3^*)$ 中只有一个非零字节. 由 Y_3 和 Y_3^* 的值即可确定出该非零字节的位置及其值. 记该非零字节的值为 e_{32}，其对应的字节位置为 $j(0\le j\le 3)$，表示为 $(\Delta X_{32})_j=e_{32}$.**

2.4. **计算 $\Delta C_{32}=\Delta X_{35}\oplus\Delta X_{31}=\Delta X_{35}=Y_0\oplus Y_0^*$.**

2.5. **由 $\Delta A_{32}=\Delta X_{32}\oplus\Delta X_{33}\oplus\Delta X_{34}\oplus\Delta rk_{32}=\Delta X_{32}$ 知 ΔA_{32} 中只包含有一个非零字节 e_{32}，其对应的字节位置为 j，即有 $\Delta a_{j,32}=e_{32}$.**

2.6. **经过 S 盒变换后 ΔB_{32} 中只包含有一个非零字节，其位置为 j，即为 $\Delta b_{j,32}$.**

经过 L 变换后有

$$\begin{aligned}\Delta C_{32}=&(\Delta b_{0,32}\Delta b_{1,32}\Delta b_{2,32}\Delta b_{3,32})\oplus\\&((\Delta b_{0,32}\Delta b_{1,32}\Delta b_{2,32}\Delta b_{3,32})\lll 2)\oplus\\&((\Delta b_{0,32}\Delta b_{1,32}\Delta b_{2,32}\Delta b_{3,32})\lll 10)\oplus\\&((\Delta b_{0,32}\Delta b_{1,32}\Delta b_{2,32}\Delta b_{3,32})\lll 18)\oplus\\&((\Delta b_{0,32}\Delta b_{1,32}\Delta b_{2,32}\Delta b_{3,32})\lll 24)\\=&\Delta X_{35}.\end{aligned}$$

分析 L 变换中的左移运算，可见有

$$\begin{aligned}\Delta C_{32}=&(\Delta b_{0,32}\Delta b_{1,32}\Delta b_{2,32}\Delta b_{3,32})\oplus\\&((\Delta b_{0,32}\Delta b_{1,32}\Delta b_{2,32}\Delta b_{3,32})\lll 2)\oplus\\&((\Delta b_{1,32}\Delta b_{2,32}\Delta b_{3,32}\Delta b_{0,32})\lll 2)\oplus\\&((\Delta b_{2,32}\Delta b_{3,32}\Delta b_{0,32}\Delta b_{1,32})\lll 2)\oplus\\&(\Delta b_{3,32}\Delta b_{0,32}\Delta b_{1,32}\Delta b_{2,32})\end{aligned}\tag{1}$$

考虑 ΔC_{32} 的第 $(j-1)\bmod 4$ 字节和第 $(j-2)\bmod 4$ 字节，分别记为 $(\Delta C_{32})_{(j-1)\bmod 4}$ 和 $(\Delta C_{32})_{(j-2)\bmod 4}$. 由上述表达式 (1) 可见有

$$(\Delta C_{32})_{(j-1)\bmod 4}=(\Delta C_{32})_{(j-2)\bmod 4}=\Delta b_{j,32}\lll 2;$$

再结合 $\Delta C_{32}=\Delta X_{35}=Y_0\oplus Y_0^*$，可见密文差分的第一个字 ΔY_0 中必有两个相等的字节，其位置分别为 $(j-1)\bmod 4$ 和 $(j-2)\bmod 4$，其值为 $\Delta b_{j,32}\lll 2$.

2.7. **由上述分析可知，第 j 个 S 盒的输入差分为 $\Delta a_{j,32}=e_{32}=(\Delta X_{32})_j$，输出差分为 $\Delta b_{j,32}=((\Delta X_{35})_{(j-1)\bmod 4}\ggg 2)$，**

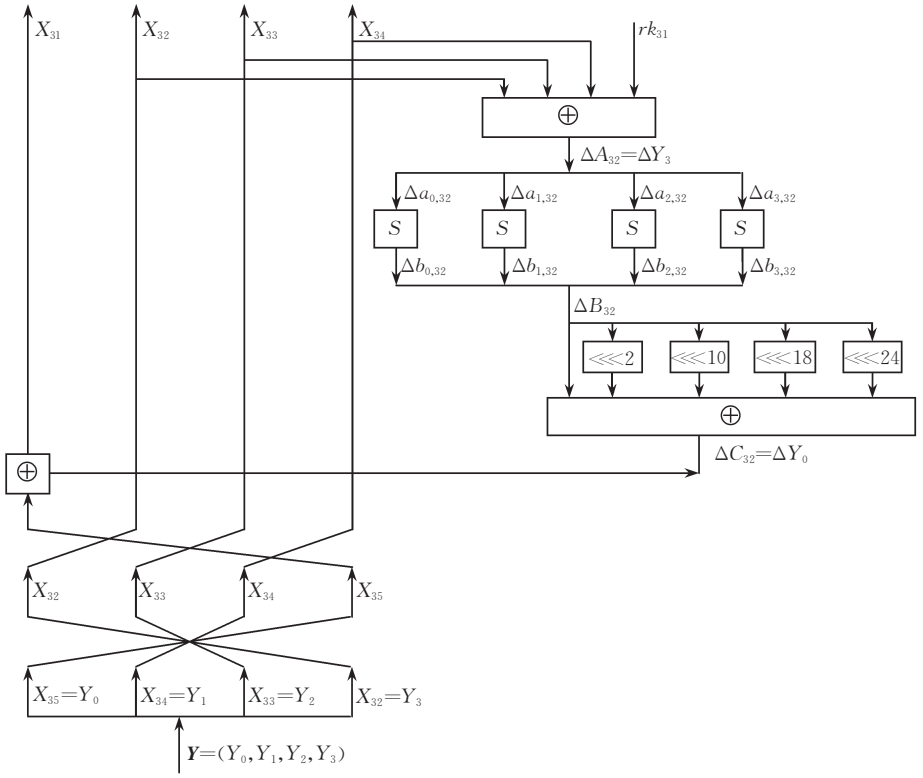

图 2　对最后一轮（第 32 轮）的攻击示意图

则根据 S 盒的差分表可以确定出 rk_{31} 的第 j 个字节值满足

$$(X_{32}\oplus X_{33}\oplus X_{34}\oplus rk_{31})_j \in IN(\Delta a_{j,i},\Delta b_{j,i}),$$

即

$$(rk_{31})_j \in ((X_{32}\oplus X_{33}\oplus X_{34})_j \oplus IN(\Delta a_{j,32},\Delta b_{j,32})).$$

2.8. 重复上述过程,直至恢复出 rk_{31} 的所有字节.

3. 攻击倒数第 2 轮(第 31 轮)的步骤如下:

3.1. 确定出 rk_{31} 后,可以对正确密文解密第 32 轮得到第 31 轮的输出值 $(X_{31},X_{32},X_{33},X_{34})$.

3.2. 对明文 X 在密钥 K 作用下加密,当进行第 31 轮加密之前,诱导 X_{31} 中产生单字节的随机故障,并对由此得到的错误密文解密第 32 轮,得到相应的第 31 轮的输出值 $(X_{31}^*,X_{32}^*,X_{33}^*,X_{34}^*)$.

3.3. 使用与步 2 完全类似的方法,恢复出 rk_{30} 的全部字节.

4. 攻击倒数第 3 轮(第 30 轮)的步骤如下:

4.1. 确定出 rk_{31} 和 rk_{30} 后,可以对正确密文 $\boldsymbol{Y}$ 解密第 32 和 31 轮得到第 30 轮的输出值 $(X_{30},X_{31},X_{32},X_{33})$.

4.2. 对明文 X 在密钥 K 作用下加密,当进行第 30 轮加密之前,诱导 X_{30} 中产生单字节的随机故障,并对由此得到的错误密文解密第 32 和 31 轮,得到相应的第 30 轮的输出值 $(X_{30}^*,X_{31}^*,X_{32}^*,X_{33}^*)$.

4.3. 使用与步 2 完全类似的方法,恢复出 rk_{29} 的全部字节.

5. 用类似的方法攻击倒数第 4 轮(第 29 轮),恢复出 rk_{28} 的全部字节.

6. 利用密钥编排算法,由上述各步得到的轮密钥恢复出加密密钥.

4.3 复杂度分析

由 $IN(\Delta a_{j,i},\Delta b_{j,i})$ 的定义可知,如果 $(X_{32}\oplus X_{33}\oplus X_{34}\oplus rk_{31})_j \in IN(\Delta a_{j,i},\Delta b_{j,i})$,则有 $(X_{32}\oplus X_{33}\oplus X_{34}\oplus (rk_{31}\oplus \Delta a_{j,i}))_j \in IN(\Delta a_{j,i},\Delta b_{j,i})$. 即若 rk_{31} 为一个候选密钥,则 $rk_{31}\oplus\Delta a_{j,32}$ 必定也为满足条件的另一个候选密钥. 又由 SMS4 算法 S 盒子的差分分布特性知,如果 $IN(\Delta a_{j,i},\Delta b_{j,i})$ 非空,则其中包含 2 个元素的概率为 99.2%(=32130/32385),包含 4 个元素的概率仅为 0.8%(=255/32385). 综合上述分析可知,要成功恢复出 rk_{31} 的某 1 个字节,则平均大约只需要 2 个不同的错误密文(要求故障发生在同一字节位置,但其错误值可以是随机的)即可. 因此,要想唯一地恢复出 rk_{31} 的全部字节,理论上只需要 8 个不同的错误密文即可. 以此类推,要恢复出 rk_{30}(或者 rk_{29},rk_{28})的全部字节,理论上也只需要 8 个错误密文. 也就是说,要完全恢复 SMS4 的密钥,理论上只需要 32 个错误密文. 如在恢复密钥的过程中结合使用穷举策略,则可以减少所需要的错误密文的数量.

5 攻击实验及实验结果

我们在一台普通的 PC 机器(CPU 为 Celeron 2.53GHz,内存 256MB)上使用 C 语言(Visual C++ 6.0)编程实现了本文给出的攻击方法,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.

我们进行了 10 次攻击实验,其结果如表 1 所示. 可见,该攻击实际中平均大约只需要 47 个错误密文(和一个正确密文)即可成功恢复出 128bit 的 SMS4 密钥. 其耗时均不超过 1ms.

表 1 攻击实验结果

序号	攻击第 32 轮所用错误密文数	攻击第 31 轮所用错误密文数	攻击第 30 轮所用错误密文数	攻击第 29 轮所用错误密文数	所用错误密文总数
1	16	9	9	12	46
2	11	12	14	12	49
3	8	10	11	12	41
4	14	10	11	11	46
5	14	11	14	13	52
6	15	10	12	13	50
7	11	10	14	11	46
8	15	9	14	9	47
9	14	14	13	12	53
10	15	9	13	10	47

该平均值略大于前面的理论分析结果,这是因为:假设中故障发生的字节位置是随机的,则只需要 8 次故障诱导就可以保证每个字节位置均发生两次故障. 即 8 个错误密文即可以完成对一个轮密钥的恢复,完成该攻击只需要 $4\times 8=32$ 个错误密文. 而实际中故障发生的字节位置是不可能完全平均的,所以就需要略大于 8 次故障诱导才能保证每个字节位置均至少发生两次故障. 所以攻击中恢复每一轮的轮密钥都需要略大于 8 个错误密文,该攻击成功恢复出 128bit 加密密钥就需要略大于 32 个错误密文,本次实验的结果是 47 个错误密文. 附录中给出了一组实际的攻击实验数据及其结果.

6 总 结

本文给出了一种对 SMS4 算法的面向字节的差分故障攻击,理论上只需要 32 个错误密文就可以恢复出 SMS4 的 128bit 密钥. 但由于故障诱导过程中故障发生的字节位置是不可能完全平均的,所以实际攻击中需要的错误密文数略大于 32 个. 上述攻击实验结果也正好说明了这一点.

该攻击方法除了利用故障诱导和差分分析外,还利用了加密算法中线性变换 L 的特点. 由于线性

变换 L 只包含了简单的移位和异或运算，通过对其移位量和异或结果的分析，可以很容易地找出部分输出值与部分输入值之间的简单关系，从而成功实现该攻击.

通过本文的分析可见，差分故障攻击对 SMS4 算法是十分有效的. 为了避免这类攻击，需要对加密设备进行保护，阻止攻击者对其进行故障诱导.

参 考 文 献

1 Office of State Commercial Cipher Administration. Block Cipher for WLAN Products — SMS4. http://www.oscca.gov.cn/UpFile/200621016423197990.pdf(in Chinese)
(国家商用密码管理办公室. 无线局域网产品使用的 SMS4 密码算法. http://www.oscca.gov.cn/UpFile/200621016423197990.pdf)

2 Boneh D., DeMillo R. A., Lipton R. J.. On the importance of checking cryptographic protocols for faults. In: Proceedings of the EUROCRYPT'97, Konstanz, Germany, 1997, 37～51

3 Biham E., Shamir A.. Differential fault analysis of secret key cryptosystems. In: Proceedings of the CRYPTO'97, Santa Barbara, California, USA, 1997, 513～525

4 Biehl I., Meyer B., Müller V.. Differential fault attacks on elliptic curve cryptosystems. In: Proceedings of the CRYPTO 2000, Santa Barbara, California, USA, 2000, 131～146

5 Blömer J., Seifert Jean-Pierre. Fault based cryptanalysis of the advanced encryption standard (AES). In: Wright R. N. ed. Financial Cryptography-FC 2003. Lecture Notes in Computer Science 2742. Berlin: Springer-Verlag, 2003, 162～181

6 Giraud C.. DFA on AES. In: Dobbertin H., Rijmen V., Sowa A. eds. Advanced Encryption Standard 4-AES 2004. Lecture Notes in Computer Science 3373. Berlin: Springer-Verlag, 2005, 27～41

7 Chen Chien-Ning, Yen Sung-Ming. Differential fault analysis on AES key schedule and some countermeasures. In: Proceedings of the Australasian Conference on Information Security and Privacy-ACISP 2003, Wollongong, Australia, 2003, 118～129

8 Dusart P., Letourneux G., Vivolo O.. Differential fault analysis on AES. In: Zhou J., Yung M., Han Y.. eds.. Applied Cryptography and Network Security-ACNS 2003. Lecture Notes in Computer Science 2846. Berlin: Springer-Verlag, 2003, 293～306

9 Hemme L.. A differential fault attack against early rounds of (Triple-)DES. In: Joye Marc, Quisquater Jean-Jacques eds. Cryptographic Hardware and Embedded Systems-CHES 2004. Lecture Notes in Computer Science 3156. Berlin: Springer-Verlag, 2004, 254～267

10 Hoch Jonathan J., Shamir A.. Fault analysis of stream ciphers. In: Joye Marc, Quisquater Jean-Jacques eds. Cryptographic Hardware and Embedded Systems-CHES 2004. Lecture Notes in Computer Science 3156. Berlin: Springer-Verlag, 2004, 240～253

11 Biham E., Granboulan L., Nguyên P. Q.. Impossible fault analysis of RC4 and differential fault analysis of RC4. In: Gilbert Henri, Handschuh Helena eds. Fast Software Encryption-FSE 2005. Lecture Notes in Computer Science 3557. Berlin: Springer-Verlag, 2005, 359～367

附录. 一组攻击实验数据及其结果.

任意选择明文和加密密钥如下：

明文：00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff

加密密钥：ff ee dd cc bb aa 99 88 77 66 55 44 33 22 11 00

攻击实验数据如下：

正确密文：35 9e 30 65 87 9e 63 2b 56 47 8a 56 64 be 10 62

对第 32 轮的输入 X_{32} 进行故障诱导得到的错误密文如下：

序号	错误密文
1	04 39 a6 f3 87 9e 63 2b 56 47 8a 56 f3 be 10 62
2	c9 51 03 56 87 9e 63 2b 56 47 8a 56 e1 be 10 62
3	7e 04 e1 2e 87 9e 63 2b 56 47 8a 56 64 c9 10 62
4	65 da 24 35 87 9e 63 2b 56 47 8a 56 64 db 10 62
5	d5 7e e8 5d 87 9e 63 2b 56 47 8a 56 64 be 46 62
6	7e 89 6c 39 87 9e 63 2b 56 47 8a 56 2c be 10 62
7	8c 48 5f dc 87 9e 63 2b 56 47 8a 56 64 84 10 62
8	32 99 f5 a7 87 9e 63 2b 56 47 8a 56 64 be 3b 62
9	03 27 bf 53 87 9e 63 2b 56 47 8a 56 64 79 10 62
10	f9 52 cf 56 87 9e 63 2b 56 47 8a 56 64 be a8 62
11	75 df 31 64 87 9e 63 2b 56 47 8a 56 ca be 10 62
12	60 9f 64 30 87 9e 63 2b 56 47 8a 56 64 21 10 62
13	ce 98 cd 9e 87 9e 63 2b 56 47 8a 56 64 d8 10 62
14	68 ef 41 49 87 9e 63 2b 56 47 8a 56 64 be 10 39
15	fb e1 81 ab 87 9e 63 2b 56 47 8a 56 64 92 10 62
16	1a 22 8c f6 87 9e 63 2b 56 47 8a 56 64 be 10 43

（续　表）

恢复出的第 32 轮轮密钥 rk[31]为

rk[31]=0f 50 21 39；

对第 31 轮的输入 X_{31} 进行故障诱导得到的错误密文如下：

序号	错误密文
1	25 f5 e0 de 95 67 88 c0 56 47 8a 56 64 be 10 62
2	95 f6 42 c9 2c dc 8a 80 56 47 8a 56 64 be 10 62
3	f4 fb b9 11 e5 13 ee c4 56 47 8a 56 64 be 10 62
4	0d 5f 15 23 40 f4 ce 86 56 47 8a 56 64 be 10 62
5	a9 69 bc ed 8f 96 69 29 56 47 8a 56 64 be 10 62
6	e6 99 33 9a 45 f5 ca 82 56 47 8a 56 64 be 10 62
7	fa c9 75 e2 97 8a 67 3b 56 47 8a 56 64 be 10 62
8	95 2a f8 cb ba 6a 97 e2 56 47 8a 56 64 be 10 62
9	f3 a2 2b 7f b8 a1 90 e7 56 47 8a 56 64 be 10 62

恢复出的第 31 轮轮密钥 rk[30]为

rk[30]=1a 64 ce c9；

对第 30 轮的输入 X_{30} 进行故障诱导得到的错误密文如下：

序号	错误密文															
1	2f	6c	c7	87	b2	5d	5f	b8	7e	6f	a8	5c	64	be	10	62
2	77	8a	00	2a	b6	8e	6d	ea	ab	ba	09	28	64	be	10	62
3	00	8c	f2	0a	ab	86	29	76	ff	99	fd	21	64	be	10	62
4	11	aa	55	87	7a	29	31	f2	57	46	ca	17	64	be	10	62
5	49	71	bd	ac	93	0f	ce	78	aa	b8	75	55	64	be	10	62
6	34	b7	1d	60	88	c1	6f	61	2e	21	94	2e	64	be	10	62
7	5c	cf	a1	b0	55	e1	7d	1c	16	42	8f	13	64	be	10	62
8	fb	d6	79	c6	b6	9d	26	7d	2c	f7	40	9c	64	be	10	62
9	73	8b	e7	0a	d6	97	6d	ff	d2	e2	ab	d2	64	be	10	62

恢复出的第 30 轮轮密钥 rk[29]**为**

rk[29]=77 f1 40 0f;

对第 29 轮的输入 X_{29} **进行故障诱导得到的错误密文如下:**

序号	错误密文															
1	0a	67	64	3f	91	68	2b	c9	f2	d7	ba	88	74	fe	50	32
2	dc	72	00	b9	d9	f9	3a	48	ac	38	1c	ae	be	d0	a4	b8
3	ed	bf	7f	99	d9	33	39	c8	4d	77	fb	85	de	5c	f2	3a
4	21	a1	fe	68	29	ce	40	fa	01	99	20	18	80	63	29	86
5	af	cc	e5	6a	ea	63	49	04	8f	bb	e6	66	af	9d	33	8a
6	cc	df	23	8f	3a	87	72	83	2a	a5	d3	75	09	f6	35	47
7	71	2e	03	fa	c3	2e	48	05	be	5b	c1	67	52	07	9f	54
8	7f	70	cf	bf	3d	9b	d3	f6	aa	ff	45	0d	4c	1e	b0	ea
9	1f	9a	a2	f2	55	1f	eb	35	a6	df	3d	2d	90	77	2d	96
10	08	c3	6f	5e	4f	b1	25	10	50	db	f1	6b	8e	54	42	da

（续 表）

序号	错误密文															
11	9b	fc	90	22	93	b3	70	81	a4	f3	f9	e4	43	b5	3c	4e
12	a0	74	b0	10	4c	80	22	43	fc	18	5d	f7	99	43	93	1c

恢复出的第 29 轮轮密钥 rk[28]**为**

rk[28]=bc 16 56 d0;

由上述轮密钥恢复出的全部轮密钥及加密密钥为

rk[27]=95 bb 4f 83; rk[26]=16 a2 f0 d5;

rk[25]=8f 30 1c 7a; rk[24]=43 9b 6c f7;

rk[23]=bd 5d 8e a2; rk[22]=fd 37 15 05;

rk[21]=90 73 f7 a1; rk[20]=40 c2 54 ee;

rk[19]=22 83 e7 dd; rk[18]=4e e8 a4 c5;

rk[17]=d4 09 fe fd; rk[16]=77 34 80 f6;

rk[15]=3c 81 6e 54; rk[14]=2c 55 ea 19;

rk[13]=60 09 c9 fd; rk[12]=f4 06 c1 e5;

rk[11]=79 d5 97 5d; rk[10]=f4 58 c5 63;

rk[9]=6e 8e bb e1; rk[8]=21 b8 1e b6;

rk[7]=30 f9 e1 9d; rk[6]=28 cc 1a 5b;

rk[5]=11 71 2e a5; rk[4]=4a 89 24 f4;

rk[3]=ce 57 ef ab; rk[2]=2c 0a 12 42;

rk[1]=e9 8e bd 81; rk[0]=5d 95 06 75;

加密密钥:ff ee dd cc bb aa 99 88 77 66 55 44 33 22 11 00

ZHANG Lei, born in 1981, Ph. D. candidate. Her research interest is cryptanalysis of block ciphers.

WU Wen-Ling, born in 1966, Ph.D., researcher, Ph.D. supervisor. Her research interests include design and cryptanalysis of block ciphers, modes of operation for block ciphers, and the theory of provable security.

Background

In September 1996 Boneh, DeMillo, and Lipton announced a new type of cryptanalytic attack which exploits computational errors to find cryptographic keys. Their attack is called Fault Attack. In 1997, Biham and Shamir extended this technique to secret key cryptosystem and came up with the concept of Differential Fault Attack (DFA). They successfully analyzed the Data Encryption Standard (DES) using this method. From then on, researchers had come up with many different kinds of differential fault cryptanalytic techniques and successfully attacked many different cryptosystems. For example, in 2000 Biehl, Meyer and Muller presented a paper describing two types of differential fault attacks on elliptic curve cryptosystems. Later many new results have been continuously given by using differential fault attack to analyze various cryptosystems such as AES, (Triple-)DES, and RC4. It is clearly that these attacks are very powerful because they can be made in practice and various techniques have been described to induce faults during cryptographic computations.

In this paper the strength of SMS4 against the differential fault attack is examined. The authors use the byte-oriented fault model, and take advantage of the differential analysis as well. Theoretically, the 128bit master key for SMS4 can be obtained by using 32 faulty ciphertexts. But in practice, for the fact that the byte position where the fault happens isn't equally distributed, the number of faulty ciphertexts needed will be a little bigger than the theoretical value. The attack experiment result validates this fact too. The result shows that only need average 47 faulty ciphertexts to recover the 128 bit keys for SMS4. So SMS4 is vulnerable to differential fault attack. To avoid this kind of attack, the authors suggest that the encryption device should be protected to prevent the adversary from deducing faults.