对一个基于细胞自动机的分组密码变形的分析*

张文涛[1+], 卿斯汉[2,3], 吴文玲[2]

[1](信息安全国家重点实验室(中国科学院 研究生院) 北京 100039)

[2](信息安全国家重点实验室(中国科学院 软件研究所) 北京 100080)

[3](中国科学院 信息安全技术工程研究中心,北京 100080)

Cryptanalysis of a Variant of a Cellular Automata Based Cryptosystem

ZHANG Wen-Tao[1+], QING Si-Han[2,3], WU Wen-Ling[2]

[1](State Key Laboratory of Information Security (Graduate School, The Chinese Academy of Sciences), Beijing 100039, China)

[2](State Key Laboratory of Information Security (Institute of Software, The Chinese Academy of Sciences), Beijing 100080, China)

[3](Engineering Research Center for Information Security Technology, The Chinese Academy of Sciences, Beijing 100080, China)

+ Corresponding author: Phn: +86-10-88256433 ext 24, E-mail: wentaozhang1@sina.com.cn

Received 2003-03-12; Accepted 2003-04-28

Zhang WT, Qing SH, Wu WL. Cryptanalysis of a variant of a cellular automata based cryptosystem. *Journal of Software*, 2004,15(5):767~771.

http://www.jos.org.cn/1000-9825/15/767.htm

Abstract: Subhayan Sen et al. have proposed a block cipher system CAC (cellular automata based cryptosystem) based on cellular automata theory, but they have not given a detailed description of some of its modules. From the point of view of application, one module of CAC is fixed, and the variant is called SMCAC (same major-CA CAC). The analysis of SMCAC is given, and the results show that the variant of this cipher is very insecure under chosen-plaintext attacks. Using cryptanalysis of the variant for reference, attacks on the cipher itself may be found when some of the design details of the cipher are known.

Key words: block cipher system; cellular automata; security analysis; chosen-plaintext attack

摘 要: Subhayan Sen 等人提出了一个基于细胞自动机的分组密码系统(cellular automata based cryptosystem,简称 CAC),但并没有给出 CAC 的某些构造模块的细节描述,从应用角度考虑,将其中的一个模块固定得到 CAC 的变形——SMCAC(same major-CA CAC).对 SMCAC 进行密码分析,结果表明,CAC 的这种变形在选择明文攻击下是极不安全的.对 SMCAC 进行分析的意义在于,知道 CAC 的具体设计细节以后,借鉴对 SMCAC 的分析,有可能对 CAC 密码系统本身的安全性造成威胁.

关键词: 分组密码系统;细胞自动机;安全性分析;选择明文攻击

* Supported by the National Natural Science Foundation of China under Grant Nos. 60373047, 90304007 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.G1999035802 (国家重点基础研究发展规划(973))

作者简介: **张文涛**(1977－),女,陕西丹凤人,博士生,主要研究领域为分组密码设计与分析;**卿斯汉**(1939－),男,研究员,博士生导师,主要研究领域为信息安全理论和技术;**吴文玲**(1966－),女,博士,副研究员,主要研究领域为分组密码设计与分析.

中图法分类号: TP309 文献标识码: A

一个细胞自动机(cellular automata,简称 CA)可以被看作一个模拟离散动态系统的并行机,CA 中细胞单元并行的简单性及其交互的局部性使得它易于硬件实现.人们已经设计了许多基于 CA 的密码系统,比如,文献[1]中提出了一个基于 CA 的伪随机序列生成器,文献[2]构造了一个基于 CA 的公钥密码系统,文献[3]给出了基于 CA 的分组密码系统和流密码系统.

Subhayan Sen 等人在文献[4]中提出了一个新的基于 CA 的分组密码系统 CAC(cellular automata based cryptosystem),设计者称 CAC 密码系统的安全性与高级加密标准 AES 相当,并且 CAC 的加、解密的速度要优于 AES.然而,在文献[4]中,设计者并没有给出 CAC 的某些构造模块的细节描述.从应用角度考虑,我们将其中的一个模块固定得到CAC的一个变形——SMCAC(same major-CA CAC).下面我们将对SMCAC用两种方法进行密码分析,结果表明,CAC 的这种变形在选择明文攻击下是极不安全的.

本文在第 1 节给出 CAC 密码系统的描述,然后从应用角度考虑,固定 CAC 密码系统的一个模块,将 CAC 的这种变形记为 SMCAC.第 2 节给出对 SMCAC 的密钥恢复的一个选择明文攻击.第 3 节给出对 SMCAC 的一个部分明文恢复的选择明文攻击.第 4 节总结全文.

1 CAC 密码系统的描述

1.1 细胞自动机的一些基本概念

细胞自动机由一系列细胞单元按一定方式排列而成,每个细胞单元状态的演变依赖于与它邻近的一些细胞单元的状态.

在 CAC 中用到的细胞自动机的每一个细胞单元的下一个状态依赖于自身及相邻的两个细胞单元的当前状态.用 q_i^t 表示第 i 个细胞单元在时刻 t 的状态,该细胞单元在下一个时刻 $t+1$ 时的状态由 $q_i^{t+1}=f(q_{i-1}^t,q_i^t,q_{i+1}^t)$ 给出,称 f 为该细胞自动机的规则.将细胞单元的状态在 $GF(2^p)$ 中取值的CA称作 $GF(2^p)$ -CA.如果一个CA的每个细胞单元的状态在经过有限次演化后都可以恢复到原来的初始状态,就称该 CA 为一个群 CA.

1.2 CAC密码系统的描述

CAC 是一个基于细胞自动机的分组密码系统.它的分组长度和密钥长度都是 128 比特,密码系统基于两类具有 16 个细胞单元的 $GF(2^p)$ -CA,分别称作 Major CA 和 Minor CA.Major CA 是一个周期为 32 的群 CA,即经过 32 次演化后,每个细胞单元的状态都会恢复到初始状态.Minor CA 也是一个群 CA,它的周期达到了可能的最大值($2^{128}-1$).文献[4]中指出,可以把 Minor CA 看作一个性能好的伪随机序列生成器.

将明文消息划分成 128 比特的明文块,分别对每一个明文块进行加密.下面用变量 Γ 表示待加密的明文块;用 K 表示 128 比特长的用户选择密钥,在给第一个明文块加密时,令 $S_0=K$.

解密过程由加密过程惟一确定,这里不再给出解密过程的详细描述,只在每一步中给出这一步的逆运算.

Step 0. 预计算子密钥:

将 S_0 作为 Minor CA 的初始输入状态, S_0 经过 d 次演化后的状态记为 S_N , S_N 即为加、解密过程中要用到的子密钥.将这个变换用 F 表示,即 $S_N=F(S_0)$;

Minor CA 的周期为 $2^{128}-1$,因此只需将 S_N 作为 Minor CA 的输入经过($2^{128}-1-d$)次演化后就得到了 S_0 ,将这个逆变换记作 $S_0=F^{-1}(S_n)$.

Step 1. 线性变换:

从 S_N 中获取 δ ,将 128 比特的明文块 Γ 的每个字节循环移位 δ 比特,结果记为 Γ_1 ;

解密时,将相应状态的每个字节朝相反方向循环移位 δ 比特.

Step 2. 仿射变换:

从 S_N 中获取 Δ;

以 S_N 作为参数,用特定的有效算法合成一个 Major CA.将 Γ_1 作为该 Major CA 的初始输入状态,经过 Δ 次演化后的状态记为 Γ_2;

Major CA 的周期为 32,因此在解密时,只需将合成的 Major CA 经过 $(32-\Delta)$ 次演化即可.

Step 3. 非线性变换:

先定义非线性函数 CMN.CMN 由 $y=x\oplus\{(c_1\cdot c_2)\oplus(c_2\cdot c_3)\oplus(c_1\cdot c_3)\}$ 给出,其中,x 是输入比特,y 是输出比特,"$\cdot$"是逻辑运算"AND";c_1,c_2,c_3 是 3 个控制比特,由 S_N 决定.

从 Γ_2 中抽取 5 个比特(抽取的方式由 S_N 决定),称它们为固定比特.

如果这 5 个固定比特中取值为 1 的个数大于 2,则除了这 5 个固定比特保持不变以外,Γ_2 中其余的每个比特经过 CMN 变换(要强调的是,由 S_N 生成两组控制比特,得到两组 CMN 变换,在这里交替使用),所得结果记为 Γ_3;

如果这 5 个固定比特中取值为 1 的个数小于或等于 2,则令 $\Gamma_3=\Gamma_2$,即保持 Γ_2 不变.

无论怎样,固定比特在这一步骤中都保持不变,从而保证了解密算法的可行.

Step 4. 子密钥"加"运算:

将 Γ_3 与 S_N 异或,所得结果 Γ_{encr} 即为 Γ 所对应的密文.

至此也就完成了对一个 128 比特的明文块的加密,将加密过程中的 Γ_2 存储起来,作为加密下一个明文块时要用到的 Minor CA 的初始输入状态 S_0.

Step 5. 如果还有未加密的明文块,则返回 Step 0,对下一个明文块进行加密操作.

1.3 对CAC的初步分析和CAC的变形SMCAC

在 CAC 中,加密不同的明文块所使用的子密钥也不同,子密钥不仅与用户选择密钥有关,还与上一次所加密的明文块有关.

加密过程中对明文块进行的循环移位运算、所使用的 Major CA、Major CA 演化的次数、两组控制比特、固定比特位置的选择都与当前使用的子密钥有关.加密不同的明文块,因为子密钥的不同,导致了加密过程中使用的这些参数(或模块)也不相同.设计者称这样做的目的是使得 CAC 的加密行为接近于"一次一密系统",从完善保密性的角度提升 CAC 的安全性.而传统中,使用分组密码对一个明文消息加密时,对于各个明文块,所使用的子密钥都是相同的,比如使用常见的 CBC,CTR 等工作模式利用 DES 或 AES 算法构造分组密码系统;并且,在文献[4]中,CAC 的设计者并没有描述这些参数或模块是怎样通过子密钥来选取的,这就为算法的分析带来了很大的困难.

另外,传统中在设计分组密码时,是将一个密码学性质不是很好的函数(轮函数)经过多次迭代来形成一个密码学性质好的分组算法.CAC 密码没有采取迭代的设计方式,设计者声称在加密不同的明文块时,选用不同的密钥就能够在很大程度上保证算法的安全性.而正是由于 CAC 没有采用迭代结构,才使得下面我们对 CAC 的一个变形的分析变得简单.

在 CAC 加密过程的第 3 个步骤中,需要在子密钥的控制下随机生成一个 Major CA.文献[4]中指出,满足周期为 32 的 Major CA 的个数大于 2^{128}(这个数字正是 128 比特明文空间中明文块的个数);在假设每次加密中所使用的子密钥随机的前提下,所使用的 Major CA 也具有随机性.然而,文献[4]并没有给出利用子密钥合成 Major CA 的具体算法.从应用角度考虑,对 CAC 软、硬件实现时利用特定的合成算法生成的不同 Major CA 之间不可能相互独立,攻击者完全有可能从合成算法中获取大量的冗余信息.

假定在加密过程中,CAC 始终使用相同的 Major CA,下面的分析结果表明,在这样的假设下,CAC 是非常不安全的.将 CAC 的这种变形算法记为 SMCAC.这样做的一个好处是,如果知道利用子密钥合成 Major CA 的具体算法,利用选择明文攻击,通过寻找和揭示不同明文块所对应的 Major CA 间的关系,借鉴下面对 SMCAC 的分析方法,就可以直接对 CAC 进行分析,从而有可能对 CAC 密码系统本身的安全性造成威胁.因此,对 SMCAC 的分析是有意义的.

2 对 SMCAC 在选择明文攻击下的一个密钥恢复攻击

这一节利用选择明文攻击,目的是恢复出 SMCAC 加密系统的用户选择密钥 K.

在 SMCAC 的加密算法中,δ,Δ 的取值都与所使用的子密钥有关.对于每一个明文块,δ 的取值有 8 种可能性,即 $\delta\in\{0,1,2,...,7\}$;Δ 的取值有 32 种可能性,即 $\Delta\in\{0,1,2,...,31\}$,因此,在不知道子密钥的情况下,可以将一个明文块经过 Step 2 以后的结果限定在 $8*32=2^8$ 个可能的取值中.

Step 3 的非线性变换是 SMCAC 中惟一的非线性模块,在这一步中,如果 5 个固定比特中取值为 1 的比特个数小于或等于 2,则这一步相当于恒等变换;在另一种情况下,即当 5 个固定比特中取值为 1 的比特个数大于 2 时,因为有 1/2 的概率使得控制比特 c_1,c_2,c_3 满足

$$(c_1\cdot c_2)\oplus(c_2\cdot c_3)\oplus(c_1\cdot c_3)=0 \tag{1}$$

那么,在两组控制比特独立选取的假设下,将有 1/4 的概率使得两组控制比特都满足式(1),这时,Step 3 也相当于恒等变换.总共有 1/2+1/2*1/4=5/8 的概率使得 Step 3 相当于恒等变换.

下面给出对 SMCAC 在选择明文攻击下的一个密钥恢复攻击的步骤:

(a) 随机选取一个 256 比特长的比特串作为明文消息,记为 P_1P_2 (P_1,P_2 的长度都是 128 比特),设明文块 P_1,P_2 经过 SMCAC 加密后所对应的密文分别为 C_1 和 C_2;

(b) 让 δ 和 Δ 遍历每一个可能的取值,计算出 P_1 和 P_2 经过 Step 2 以后的 256 个可能的取值,分别记为 $M_1(0),M_1(1),...,M_1(255)$ 和 $M_2(0),M_2(1),...,M_2(255)$.设实际上 P_1 经过 Step 2 后的结果为 $M_1(i)$,P_2 经过 Step 2 后的结果为 $M_2(j)$;

(c) 如果 P_1,P_2 经过 Step 3 时的变换都是恒等变换,由 Step 4,有

$$M_1(i)\oplus F(K)=C_1 \tag{2}$$

$$M_2(j)\oplus F(M_1(i))=C_2 \tag{3}$$

其中,$F(K)$ 是加密 P_1 时所使用的子密钥,$F(M_1(i))$ 是加密 P_2 时所使用的子密钥.

对每一个 $0\le n\le 255$,计算 $F(M_1(n))$ 和 $C_2\oplus M_2(n)$,如果集合 $\{F(M_1(n))\}$ 中的某一个元素和集合 $\{C_2\oplus M_2(n)\}$ 中的某一个元素相等,就说明有碰撞发生.如果 P_1,P_2 经过 Step 3 时的变换不全是恒等变换,由于 P_1 和 P_2 选取的随机性以及 Minor CA(F 函数)良好的伪随机特性,从而在这种情况下有碰撞发生的概率非常小,可以忽略不计.因此,我们可以近似地认为有碰撞发生当且仅当 P_1,P_2 经过 Step 3 时的变换都是恒等变换.

(c.1) 当有碰撞发生时,设 $F(M_1(u))=C_2\oplus M_2(v)$,$0\le u,v\le 255$.由式(3),则有 $i=u,j=v$.再由式(2),有 $K=F^{-1}(M_1(i)\oplus C_1)$,这样就恢复出了用户选择密钥 K.

(c.2) 如果没有碰撞发生,则返回步骤(a),重新选取两个明文块,按照上述步骤运行,直到有碰撞发生为止.

Step 3 为恒等变换的概率为 5/8,在明文消息和子密钥独立、均匀分布的假设下,式(2)和式(3)同时成立的概率为 5/8*5/8=25/64.因此,上述的攻击步骤平均运行 3 次,就会有碰撞发生,从而恢复出用户选择密钥.

该攻击的数据复杂度为 3*2=6 个选择明密文对;忽略异或运算,时间复杂度大约为 $3*2^9$ 次(Step 1+Step 2)运算再加上 $3*2^9$ 次 F 运算.

上述事实表明,SMCAC 在选择明文攻击的情况下,利用很小的计算代价就能够恢复出它的用户选择密钥.

3 对 SMCAC 在选择明文攻击下的一个部分明文恢复攻击

本节讨论的明文恢复攻击是针对选择明文攻击而言的.攻击者的任务是:给定一个随机选取的明文所对应的密文,解密这个密文,恢复出它所对应的明文;在攻击过程中,攻击者与加密系统有下面的交互:攻击者可以选择很多明文,加密系统给出它们所对应的密文,即攻击是指选择明文攻击.有关明文恢复攻击的定义和讨论参见文献[5]中的第 6 节.

下面给出对 SMCAC 在选择明文攻击下的一个部分明文恢复攻击的步骤.该攻击不直接恢复用户选择密钥,而是恢复第 1 个明文块在经过 Step 2 以后的结果,从而由加密算法,通过链接计算,也就知道了加密第 1 个明文块之后的每一个明文块时所使用的子密钥.

(a) 选取第 1 个明文块为 128 比特的全 0 字符串(即 128 个比特都取 0),记为 P_0;记第 2 个明文块及其对应的密文分别为 P_1 和 C_1;

(b) P_0 经过 Step 1 以后保持不变,经过 Step 2 以后,有 32 个可能的取值,分别记为 $M_0(0),M_0(1),...,M_0(31)$. 对每一个 $0 \le n \le 31$,计算 $F(M_0(n))$,则加密下一个明文块 P_1 时所使用的子密钥在集合 $\{F(M_0(n))\}$ 中;

(c) 让 n 从 0 开始取值,每次增加 1:

以 $F(M_0(n))$ 作为加密 P_1 时所使用的子密钥,对 P_1 进行加密操作.如果以某一个 $F(M_0(i))$ ($0 \le i \le 31$)作为子密钥,加密 P_1 所得的密文与 C_1 相等,则终止操作.并且以几乎为 1 的概率确认,加密 P_1 所用的子密钥为 $F(M_0(i))$, P_0 经过 Step 2 以后的结果为 $M_0(i)$.

恢复出 P_0 经过 Step 2 以后的结果 $M_0(i)$,则由加密过程,可以计算出加密第 2 个明文块时所使用的子密钥,并且通过递推,可以计算出在加密其后的每一个明文块时所使用的子密钥.也就是说,该攻击可以恢复出第 2 个明文块以后(包括第 2 个明文块)的所有明文块的加密过程,从而给定任意一个密文,就可以通过计算加密时所用的子密钥来恢复它所对应的明文.

该攻击的数据复杂度为 2 个选择明密文对,时间复杂度为 32 次(Step 1+Step 2)运算和 32 次加密过程以及 32 次 F 运算.

从上面的结果可以看出,当 SMCAC 对一个明文消息进行加密时,利用选择明文攻击,使用很小的计算代价就能够恢复出第 2 个明文块以后(包括第 2 个明文块)的所有明文块的加密过程.

4 总 结

Subhayan Sen 等人在文献[4]中提出了一个新的基于细胞自动机的分组密码系统(CAC),但保密了其中某些密码模块的细节描述,这就增加了对 CAC 进行密码分析的难度.从应用角度考虑,我们将其中的一个模块固定,得到 CAC 的一个变形——SMCAC.本文用两种方法对 SMCAC 进行密码分析,两种方法分析的结果都表明,CAC 的这种变形在选择明文攻击下是极不安全的.对 SMCAC 进行分析的意义在于,知道 CAC 的具体设计细节以后,借鉴对 SMCAC 的分析,有可能对 CAC 密码系统本身的安全性造成威胁.

本文的分析结果表明,CAC 分组密码系统存在一定的安全隐患.通过初步的调研,我们相信,作为密码系统的一种构造模块,细胞自动机具有许多良好的密码学性质.然而,目前还没有一种公众认可的安全、高效的基于细胞自动机的分组密码系统.那么,如何把细胞自动机用于分组密码系统的设计,构造出安全、高效的分组密码系统,还值得进一步研究.

References:

[1] Wolfram S. Cryptography with cellular automata. In: Williams HC, ed. Advances in Cryptology'85. Santa Barbara: Springer-Verlag, 1986. 429~432.

[2] Guan P. Cellular automaton public-key systems. Complex System, 1987,1(1):51~57.

[3] Nandi S, Kar BK, Chaudhuri PP. Theory and application of cellular automata in cryptograph. IEEE Trans. on Computers, 1994, 43(12):1346~1357.

[4] Sen S, Shaw C. Cellular automata based cryptosystem (CAC). In: Deng RH, Qing SH, Bao F, Zhou JY, eds. Information and Communications Security. LNCS 2513, Berlin: Springer-Verlag, 2002. 303~314.

[5] Goldwasser S, Bellare M. Lecture Notes on Cryptography. 2001. http://www-cse.ucsd.edu/users/mihir