文章编号:1001 - 9081(2008)08 - 1912 - 04

一种基于演化计算的序列密码分析方法

陈连俊,赵 云,张焕国
(武汉大学 计算机学院,武汉 430072)
(francky123@sina.com)

摘 要:序列密码是一类重要的密码,演化计算是一种重要的智能计算。在研究利用演化计算进行序列密码分析方法的基础上,具体给出了一种利用演化计算对非线性滤波型序列密码体制进行分析的方法。分别在移位器初态未知和抽头位置未知两种情况下,对滤波流密码体制进行了密码分析。实验结果表明,该算法的攻击复杂度远远小于穷举攻击的复杂度。

关键词:序列密码;密码分析;演化计算
中图分类号: TP309 **文献标志码**:A

Cryptanalysis for stream cipher based on evolutionary computation

CHEN Lian-jun, ZHAO Yun , ZHANG Huan-guo
(*College of Computer Science , Wuhan University , Wuhan Hubei* 430072, *China*)

Abstract: It is well known that stream cipher is an important cipher and evolutionary computing is an important intelligent computing. A new analytical method for filter model stream cipher based on evolutionary computing was proposed. A particular analytical method was given for steam cipher in two cases with the initial state of shift register and the positions of drawing out points of shift register unknown. The results of practical experiments indicate that the computational complexity of the attack method is far lower than that of exhaustive attack.

Key words: stream cipher; cryptanalysis; evolutionary computation

0 引言

演化计算是借鉴生物进化规律(优胜劣汰,适者生存)的一种通用问题求解方法,具有高度并行、自适应、自学习等特征,实践证明演化计算是解决许多困难问题的有效方法[1]。本文借鉴生物进化的思想,将密码学与演化计算结合起来,提出演化密码概念和利用演化计算设计密码的方法,并在演化 DES 密码体制、S 盒的演化设计、Bent 函数和 HASH 函数的分析与演化设计方面获得实际成功[2-4]。

序列密码又称为流密码,是密码学的一个重要分支,具有实现容易、效率高等特点,已经成为许多重要应用领域的主流密码。序列密码的基本原理就是通过明(密)文和移位寄存器产生的密钥序列模 2 加来完成加(解)密的过程,因此序列密码的关键是产生密钥序列的算法。根据产生密钥序列的移位寄存器的不同可分为线性移位寄存器和非线性移位寄存器。一方面,对于单纯的线性移位寄存器产生(设其级数为 n)的密钥序列,如果已知一段长为 $2n$ 的明密对,可以在多项式时间复杂度 $O(n^3)$ 对其时行破译,使得算法的安全性大大降低;另一方面,由于缺少得力的数学工具,对于非线性移位寄存器序列的分析和研究都比较困难。因此实际应用中的流密码一般是对线性移位寄存器序列进行非线性组合[5]。

本文研究利用演化计算对滤波模型流密码进行密码分析的方法,并在移位寄存器初态未知和抽头位置未知两种情况下进行了具体的密码分析,实验表明此方法是有效的,其分析复杂度远远低于穷举攻击的复杂度。

1 演化计算的基本概念

演化计算主要包含遗传算法、演化规划、演化策略和演化程序设计等理论与技术。

1.1 遗传算法概要

遗传算法是近几年发展起来的一种崭新的全局优化算法。通过维持一组可行解,并对可行解进行重新组合,最终走向最优解。它克服了传统优化方法容易陷入局部极值的缺点。其主要特点是直接对结构对象进行操作,不存在求导和函数连续性的限定,具有鲁棒性、随机性、全局性以及适于并行处理的特点,已广泛应用于许多科学技术领域,并且取得了巨大成功。

1.2 遗传算法的步骤

1) 定义一个目标函数;

2) 将可行解群体在一定的约束条件下初始化,每一个可行解用一个向量 $\boldsymbol{x}$ 来编码,称为一条染色体,向量的分量代表基因,对应可行解的某一决策变量;

3) 计算群体中每条染色体 $x_i(i = 1,2,\cdots,n)$ 所对应的目标函数值,并以此计算适应值 $f(x_i)$,按 $f(x_i)$ 的大小来评价该可行解的好坏;

4) 以优胜劣汰的机制,将适应值差的染色体淘汰掉,对幸存的染色体根据其适应值的好坏,按概率随机选择,形成新的群体;

5) 通过杂交和变异的操作,产生子代。杂交是随机选择两条染色体(双亲),将某一点或多点的基因互换而产生两个

收稿日期:2008 - 04 - 25;**修回日期**:2008 - 06 - 05。
基金项目:国家自然科学基金资助项目(60673071);国家 863 计划项目(2006AA01Z442,2007AA01Z441)。
作者简介:陈连俊(1965 -),男,四川成都人,高级工程师,博士研究生,主要研究方向:密码学; 赵云(1980 -),男,河北武安人,工程师,硕士研究生,主要研究方向:演化密码; 张焕国(1945 -),男,河北元氏人,教授,博士生导师,主要研究方向:信息安全、可信计算、容错计算。

新个体,变异是基因中的某一点或多点发生突变;

6) 对子代群体重复步骤 3) ~5)的操作,进行新一轮遗传进化过程,直到迭代收敛(适应值趋于稳定)即找到了最优解或准最优解。

2　非线性滤波模型流密码

在流密码系统中,密钥流的生成是一个关键问题,而前馈网络模型则是一种非常重要的密钥流生成器,它用一个非线性前馈函数(也称前馈逻辑)对一个或几个线性反馈移位寄存器的适当抽头进行非线性变换,产生出具有较好密码学性质的输出序列。人们通常将前馈网络模型(这里只考察单输出的情况)按其具体形式分为 3 类进行讨论:

1) 滤波模型,它由一个线性反馈移位寄存器和一个非线性滤波函数组成;

2) 组合模型,它由多个线性反馈移位寄存器和一个非线性组合函数组成;

3)滤波—组合模型,它是滤波模型和组合模型的混合模型。

作为一种重要的密钥流生成器,人们对前馈网络模型进行了大量研究,给出了许多设计准则和攻击方法[6-8]。

本文分析的是滤波模型的流密码,即由一个线性移位寄存器(Line Shift Register,LSR)和一个非线性函数 $f(x)$ 组成的滤波模型密码,其结构如图 1 所示。

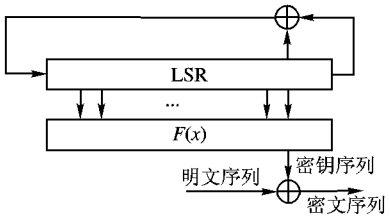

图 1　滤波模型流密码

非线性函数中某个变量的输入值和函数输出值相同的个数占这个变量总输入个数的比例称为该变量与函数输出的相关优势。例如在 $f(x)$ 中变量 x_i 的相关优势为 $p_i = p(f(x) = x_i)$,其中 $p(f(x) = x_i)$ 表示 $f(x)$ 与 x_i 相等的概率。

非线性函数的相关优势和由此函数构成的流密码的抗相关攻击能力是密切相关的,这在后面的实验中也得到了证实。

3　移存器初态未知的相关分析

3.1　密码分析的前提和假定

假设攻击方对整个序列密码的结构都清楚,只是不知道移位寄存器的初始状态(以下简称初态),即攻击者知道移位寄存器的连接多项式 $g(x)$,非线性函数 $f(x)$,非线性函数在移位寄存器中抽取抽头位置以及截获的一定长度的密钥序列。这样攻击者可以对密码进行选择明文攻击。此时,移位寄存器的初态就是这个算法的初始密钥,攻击者的目的就是找到这个初态,从而完成对密码的破译。

3.2　序列密码演化分析的具体实现

1) 个体编码方式和初始种群的产生。

以寄存器初态为个体,采用二进制编码,则 29 级移位寄存器的初态可以用 29 比特的二进制数来表示。初始种群采用计算机随机数的方式产生,产生一定数量 29 比特的二进制数,这里的数量就是种群的规模,二进制数就是假定的初态。

2) 适应度函数。

由于移位寄存器不同的初态产生的密钥序列是不一样的,初态差别越大,其产生的密钥序列的差别也越大(这里的差别就是指二进制序列中不相同的比特数),因此可以用两个密钥序列之间的差别大小来衡量产生这两个密钥序列的两个初态之间的接近程度。

由汉明距离的定义可知,两个密钥序列汉明距离越大则它们之间差别越大,即产生密钥序列的两个初态相差越大,反之亦然。我们把每个个体产生的密钥序列和截获的真正初态产生的密钥序列进行比较,如果两者差别大,说明个体与真实初态的差距越大,应该赋予一个较小的适应值;如果两者差别小,说明个体与初态差别小,应该赋予一个较大的适应值。

基于以上思想,分别计算每个个体产生的密钥序列与真正初态产生的密钥序列之间的汉明距离,然后用密钥序列的长度减去此汉明距离作为适应度函数。具体表示如式(1)。

$$sfitness(x_i) = stringlength - d(string(x_i), aimstring) \tag{1}$$

其中,x_i 是一个个体(即一个假定的初态),$string(x_i)$ 是 x_i 产生的密钥序列,$aimstring$ 是真正的初态产生的密钥序列(也就是截获到的序列),$d(string(x_i), aimstring)$ 是两个密钥序列的汉明距离,$stringlength$ 是密钥序列的长度,$sfitness(x_i)$ 是个体 x_i 的适应度函数。显然 $sfitness(x_i)$ 的取值范围是[0, $stringlength$]。

实验过程中发现用上式作为适应度函数,种群个体逼近真正初态的速度不够快,这是因为每个个体适应度之间的差别比较小,造成了与真正初态更接近的个体不能以很高的概率遗传到下一代。为了加快算法的收敛性,提高算法效率,我们对函数 $sfitness(x_i)$ 进行了取幂运算,从而扩大不同个体之间的适应度差距,在实验中取的个体 x_i 适应度函数如下。

$$fitness(x_i) = (sfitness(x_i))^s \tag{2}$$

其中,s 为幂定标,此时适应度函数的取值范围为[0, $stringlength^s$]。

3) 遗传算子的设计。

我们在算法中采用一般的遗传算子。选择策略是通过轮盘赌选择的方式。杂交是对种群中相邻两个个体以 Pc 的概率采用单点杂交,变异算子是对种群中每个个体以 Pm 的概率采用单点均匀变异。每一代的最佳个体都被保留到下一代。

4) 分析成功的判定。

由公式 $fitness(x_i) = (sfitness(x_i))^s$,所以当 x_i 为真正的初态时,有 $fitness(x_i) = (stringlength)^s$。设 $currentbest$ 表示当前最好的个体,因此等式 $fitness(currentbest) = (stringlength)^s$ 成立可以作为算法成功的判定条件,满足此条件时,算法终止,并判定找到了初态。

注意,如果截获的真正初态产生的密钥序列比较短,则存在一定的误判概率(即虽然满足 $fitness(currentbest) = (stringlength)^s$,但 $currentbest \neq aimplant$。也就是虽然以当前找出的最好个体为初态得到的密钥序列和真正的初态产生的密钥序列一样,但这个最好个体却不是真正的初态)。在穷尽攻击和别的攻击算法中也会遇见同样的问题,这是由密钥序列的长短决定的,与攻击算法无关,在实验数据中我们给出了这样的误判次数。同时在实验中设定了一个最大演化代数。如果达到这个代数还没有找到初态,就判定算法失败。实验中判定成功次数这个指标是指 N 次实验中,以满足等式 $fitness(currentbest) = (stringlength)^s$ 为判定条件的成功次数。因为存在误判,用成功次数减去误判次数就是真正找到初态的次数,除以实验次数,就得到实验成功的概率。

3.3 实验数据及分析

通过编程,利用遗传算法对初态未知的滤波模型密码进行分析,其中线性移位寄存器的连接多项式为29次本原函多项式 $g(x) = x^{29} + x^2 + 1$。密钥序列的长度取64比特,种群规模为100,最大演化代数为1 000。非线性函数分别采用 $f_1(x) = x_1x_2 \oplus x_2x_3 \oplus x_3$, $f_2(x) = x_1x_4 \oplus x_2x_3$, $f_3(x) = x_1x_4 \oplus x_2x_5 \oplus x_3x_6 \oplus x_1x_2x_3$。对于每一个非线性函数都进行了多组实验,其中选择概率 Pc 变化范围0.5~1.0,变异概率 Pm 变化范围0.01~0.10,幂定标 s 变化范围1~10。

表1给出了具体的实验数据,其中演化算子 $Pc = 0.75$, $Pm = 0.04$, $s = 5$。

表1 初态未知的实验数据

非线性函数	实验次数	判定成功的次数	误判次数	成功概率/%	平均演化代数
$f_1(x)$	1000	942	8	93.4	74
$f_2(x)$	1000	528	0	52.8	197
$f_3(x)$	1000	268	0	26.8	230

决定算法效率的有两个指标:1)成功概率,即在实验中成功找到真正的初态的概率,此值越大越好;2)找到真正初态所需的平均演化代数,此值越小越好。因此引入算法复杂度来将这两个指标综合考量,复杂度计算如式(3)。

$$\text{算法复杂度} = (\text{种群规模} \times \text{平均演化代数}) \,/\, \text{成功概率} \tag{3}$$

对于一个已知算法结构的29级序列密码进行攻击,进行穷举攻击平均需要试探 2^{28} 次初态(密钥),即计算复杂度为 2^{28}。用本文算法进行攻击的实验中,当非线性函数取 $f_1(x) = x_1x_2 \oplus x_2x_3 \oplus x_3$ 时,成功概率为93.4%,平均演化代数为74代。种群规模为100,因此平均需要试探密钥的次数为 $(100 \times 74)/93.4\% \approx 7922$,因为 $2^{12} < 7922 < 2^{13}$,算法计算复杂度大约为 $O(2^{13})$,这是穷举攻击复杂度的 $2^{13}/2^{28} = 1/2^{15}$,也就是说只是穷举攻击所需试探密钥次数的 2^{15} 分之一。同样计算可得非线性函数取 $f_2(x)$ 和 $f_3(x)$ 时,计算复杂度分别为 $O(2^{15})$ 和 $O(2^{17})$,都远远小于穷举攻击的 $O(2^{28})$。

由第2章引入的相关优势的概念,可知相关优势 $f_1(x) > f_2(x) > f_3(x)$,并且 $f_2(x)$, $f_3(x)$ 中各变量与输出之间的相关优势是相同(也叫平衡)的。实验中成功概率 $f_1(x) > f_2(x) > f_3(x)$,平均演化代数 $f_1(x) < f_2(x) < f_3(x)$。因此算法的复杂度 $f_1(x) < f_2(x) < f_3(x)$,这也从实验的角度验证了序列密码的强度在连接多项式等其他因素相同的情况下和采用的非线性函数的相关优势是成反比的。

在实验中发现,适应度函数对算法复杂度的影响最大,适应度函数选择不好,算法可能会变得效率不高。同时其他遗传算子的选取对于算法也有较大影响,大量实验数据表明,在实验前提下,算法对于变异算子的依赖度最高。也就是说实验结果的好坏受变异算子的影响最大,幂定标次之,而杂交概率对于实验结果的影响最小。因此如何选择适应度函数、选择算子、交叉算子、变异算子是遗传算法的关键。

4 抽头位置未知的相关分析

4.1 密码分析的前提和假定

假设攻击者知道移位寄存器的连接多项式 $g(x)$,非线性函数 $f(x)$ 和移位寄存器的初态,以及截获了一定长度的密钥序列,只是不知道移位寄存器中抽头的位置。此时,抽头位置就是密码算法的密钥,攻击者的目的就是找到抽头位置,从而对算法进行破译。

4.2 序列密码演化分析的具体实现

基本思想和前一节的实验大体相同,只是在编码、杂交、变异操作的具体实现上有所不同。以下主要说明这些不同的地方,相同地方省略。

1)个体编码方式和初始种群的产生。采用二进制编码,以29比特的二进制数作为个体,其中有抽头的位置对应比特为1,没有抽头的位置对应为0。由于 n 元布尔函数 $f(x)$ 有 n 个抽头,因此每个个体的29比特中有 n 个1。初始种群采用计算机随机数的方式产生,具体到每个个体,即随机产生 n 个0~28之间的互不相同的整数,让个体的相应位置为1,其余位置为0,这里的二进制数中比特数为1的位置就是假定的抽头位置。

2)杂交算子。对种群中相邻两个个体以 Pc 的概率杂交,先产生一个0~1的随机数 p,如果 $p < Pc$ 则进行杂交,否则不进行杂交。具体的方法如下:

将两个个体分别异或,记为Xornum,这也是一个29比特的二进制数。

① 当两个个体中的抽头位置都相同时,不做交换。

② 当两个个体中的所有抽头位置都不相同时,Xornum中的1的个数为 $2n$,此时从这 $2n$ 个1中随机选取 n 个,这 n 个1位置不变,别的位置填0,作为一个新的个体。剩下的 n 个1位置不变,别的位置填0,作为另一个新的个体。

③ 当两个个体中有部分抽头位置相同时,设位置相同的个数为 m,则两个个体中抽头相同的位置保持不变,不同的位置进行杂交。此时Xornum中的1的个数为 $2(n-m)$,从中随机选取 $n-m$ 个1,以这些1为抽头位置和两个个体相同的 m 个抽头位置作为一个新的个体,剩下的 $n-m$ 个1为抽头位置和相同的 m 个抽头位置作为另一个新的个体。

(3)变异算子。对种群中每个个体以 Pm 的概率采用单点变异。具体对于个体中的每个抽头,先产生一个0~1的随机数 p,如果 $p < Pm$ 则进行变异,否则不进行变异。$p < Pm$ 时随机产生一个0~28的整数 q,如果 q 与这个个体中的某个抽头位置相同则重新产生 q,否则将这个抽头由当前位置变到位置 q(即将原来抽头位置的1置为0,并将第 $q+1$ 比特的数置为1)。

4.3 实验数据及分析

利用遗传算法对抽头位置未知的密码算法进行分析,其中线性移位寄存器的连接多项式为 $g(x) = x^{29} + x^2 + 1$。密钥序列的长度取64比特,种群规模为100,最大演化代数为100。非线性函数分别为 $f_2(x) = x_1x_4 \oplus x_2x_3$, $f_3(x) = x_1x_4 \oplus x_2x_5 \oplus x_3x_6 \oplus x_1x_2x_3$, $f_4(x) = x_1x_4 \oplus x_2x_3 \oplus x_1x_5 \oplus x_2x_6 \oplus x_3x_7 \oplus x_4x_8$,对于每一个非线性函数都进行了多组实验。

表2给出了具体的实验数据,其中遗传算子 $Pc = 0.95$, $Pm = 0.07$, $s = 5$。

表2 抽头位置未知的实验数据

非线性函数	实验次数	判定成功的次数	误判次数	成功概率/%	平均演化代数
$f_2(x)$	10000	9586	87	94.99	7
$f_3(x)$	10000	5134	28	51.06	22
$f_4(x)$	10000	1643	16	16.27	40

对于一个已知算法结构和初态,抽头位置未知的29级序

列密码进行穷举攻击,平均需要试探 $\frac{C_{29}^{n}}{2}$ 次,其中 n 为抽头的个数。此时抽头的位置就可以认为是这个序列密码的密钥。

当非线性函数取 $f_2(x) = x_1x_4 \oplus x_2x_3$ 时,穷举攻击平均试探密钥次数为 $\frac{C_{29}^{4}}{2} = 11\,876$,而由式(3)可知本文算法平均需要试探密钥的次数为 $(100 \times 7)/94.99\% \approx 737$,此次数是穷举攻击平均试探次数的 $737 / 11\,876 \approx 1/2^4$,也就是说计算复杂度是穷举攻击的 2^4 分之一。同样当非线性函数分别取 $f_3(x)$ 和 $f_4(x)$ 时,计算复杂度分别是穷举攻击的 $1/2^6$ 和 $1/2^7$。可见本算法对抽头位置未知条件下的滤波模型流密码进行攻击的复杂度比穷举攻击有很大的减少。

对于实验二,遗传算子依然是 $Pc = 0.95$, $Pm = 0.07$, $s = 5$ 的情况下,当攻击者截获的密钥序列由 64 比特变为 128 比特时,实验结果见表 3。与表 2 相应的数据对比可以发现成功概率有了较大提高,同时平均演化代数也变小。当抽头数量较多时,这种提高更明显。例如当非线性函数取 $f_4(x)$,截获的密钥序列为 64 比特时平均试探密钥次数为 $(100 \times 40)/16.27\% \approx 24\,585$,截获的密钥序列为 128 比特的情况下平均试探密钥次数为 $(100 \times 30)/40.29\% \approx 7\,446$,后者只是前者的 30% 左右。可见实际密码攻击中,攻击者掌握的密钥序列越长越有助于提高此攻击算法的效率。

表 3　截获的密钥序列为 128 比特时的实验数据

非线性函数	实验次数	判定成功的次数	误判次数	成功概率/%	平均演化代数
$f_2(x)$	10000	9977	118	98.59	5
$f_3(x)$	10000	8558	19	85.29	15
$f_4(x)$	10000	4045	16	40.29	30

5　实验结果分析

通过大量实验数据说明,无论是对于初态未知还是抽头位置未知的滤波模型序列密码进行分析,通过选取适当地适应度函数,并且在实验过程中对演化算子进行调整和择优,本文提出的基于遗传算法的分析方法都有较高的成功概率和较小的演化代数,可以有效减少密码分析过程中试探密钥的次数,提高了攻击效率。特别是当非线性函数的输入和输出之间的相关优势较大时,这种提高更明显。从实验数据可以看出,相关优势越大,成功概率越高,平均演化代数越小,计算复杂度越小。这也从实验的角度解释了,为什么在实际的序列密码设计中要求尽量采用相关优势接近 0.5 的非线性函数,并且要求非线性函数中各变量与输出之间的相关优势相差比较小的原因。

6　结语

本文研究了应用演化计算分析滤波模型流密码的技术与方法。实验表明演化计算在序列密码相关分析中有重要作用,可以作为密码分析的一个有力工具。其中如何设计好的适应度函数,以及选择、杂交、变异等遗传算子是遗传算法的关键,也是决定遗传算法是否高效的重要因素。本文只是利用遗传算法对滤波模型流密码进行了分析,实际上该方法同样可以应用在非线性组合模型,组合-滤波模型等多种结构的序列密码分析中。另外根据问题实际需要,将演化算法与蚁群算法、模拟退火算法等演化计算方法相结合,会获得更好的分析效果。

参考文献:

[1] 潘正君, 康立山, 陈毓屏. 演化计算[M]. 北京: 清华大学出版社, 1998.

[2] 张焕国, 冯秀涛, 覃中平, 等. 演化密码与 DES 的演化研究[J]. 计算机学报, 2003, 26(12): 1678 - 1684.

[3] 孟庆树, 张焕国, 王张宜, 等. Bent 函数的演化设计[J]. 电子学报, 2004, 32(11): 1901 - 1903.

[4] ZHANG HUAN-GUO, WANG YU-HUA, WANG BANG-JU, *et al*. Evolutionary random number generator based on LFSR[J]. Wuhan University Journal of Natural Science, 2007, 12(1): 179 - 182.

[5] 张焕国, 刘玉珍. 密码学引论[M]. 武汉: 武汉大学出版社, 2003.

[6] 温巧燕, 钮心忻, 杨义先. 现代密码学中的布尔函数[M]. 北京: 科学出版社, 2000: 183 - 245.

[7] 丁存生, 肖国镇. 流密码学及其应用[M]. 北京: 人民邮电出版社, 1994: 106 - 180.

[8] 胡一平. 谱绝对值均匀的前馈流密码的线性复杂度的稳定性研究[J]. 应用数学学报, 1998, 21(2): 288 - 294.

(上接第 1911 页)

TCG 为了能够推广可信计算机,要求 TPM 是低成本的,并且与当前的 PC 架构兼容,这样的要求也影响到了内部密钥管理的合理设计。因此造成在实际应用中随着密钥数量的不断增加,相应的授权数据也大量增加,用户需要维护大量的口令,这就给实际应用造成很大的困难。TCG 为了解决 TPM 的存储空间有限的问题,用父密钥对除 EK、SRK 外的所有密钥进行加密之后保存在 TPM 之外,TCG 内部没有存储密钥的任何信息。如果修改某一密钥的授权数据后,在 TPM 外就存在包含该密钥的两个加密数据块,分别对应老授权数据和新授权数据。如果将含有老授权数据的密钥数据块载入到 TPM 中,该密钥仍然可以正常使用,只要输入老的授权数据即可。而且在 TCG 的密钥管理机制下,也没有办法删除某个密钥。这些显然是 TCG 密码机制中的缺点。

参考文献:

[1] 张焕国, 罗捷, 金刚, 等. 可信计算研究进展[J]. 武汉大学学报: 理学版, 2006, 52(5): 513 - 518.

[2] 沈昌祥, 张焕国, 冯登国, 等. 信息安全综述[J]. 中国科学 E 辑: 信息科学, 2007, 37 (1): 129 - 150.

[3] Trusted Computing Group: TPM specification version 1.2. Part 1 Design Principles[Z].

[4] Trusted Computing Group: TPM specification version 1.2. Part 2 TPM Structures[Z].

[5] Trusted Computing Group: TPM specification version 1.2. Part 3 TPM Commands[Z].

[6] FABREGA F J T, HERZOG J C, GUTTMAN J D. Strand spaces: proving security protocols correct [J]. Journal of Computer Security, 1999, 7(2/3): 191 - 230.

[7] EISENBARTH T, GüNEYSU T, PAAR C. Reconfigurable trusted computing in hardware[C]// Proceedings of 2nd ACM Workshop on Scalable Trusted Computing. New York: ACM Press, 2007: 15 - 20.

[8] GRAWROCK D. The Intel safer computing initiative[EB/OL]. [2007 - 10 - 22]. http://www.intel.com/intelpress/toc-secc.pdf.

[9] ALVES T, FELTON D. TrustZone: Integrated hardware and software security enabling trusted computing in embedded systems[EB/OL]. [2007 - 10 - 22]. http://www.arm.com/pdfs/TZ_Whitepaper.pdf.