文章编号：1004-4353(2015)03-0249-05

基于哈希链的序列密码算法

姜璇， 李永珍*
（延边大学工学院 计算机科学与技术学科 网络与安全研究室，吉林 延吉 133002）

摘要：为了设计出能应用于无线移动通信的序列密码算法，提出了一种将单分组散列函数应用于哈希链方法的序列密码算法(SC-SBH). 该算法首先用哈希链的方法对单分组散列函数进行循环运算，将运算的每一次结果值输出后连接成序列密码的密钥序列，然后将明文与密钥序列进行按位异或运算得到密文. 实现 SC-SBH 算法后，对其安全性和随机性进行了测试，并将其运行效率与 SC-MD5 和 SC-SHA 算法进行了比较. 实验结果表明，SC-SBH 算法在加密方面不仅能够保证安全性，而且其运行效率明显高于 SC-MD5 和 SC-SHA 算法.

关键词：序列密码；单分组散列函数；哈希链；运行效率；安全性
中图分类号：TP309.7 **文献标识码**：A

Stream cipher algorithm based on Hash chain

JIANG Xuan， LI Yongzhen*
(*Network and Information Security Lab.*, *Dept. of Computer Science & Technology*, *College of Engineering*, *Yanbian University*, *Yanji* 133002, *China*)

Abstract: We proposed a new stream cipher in this article to apply in the wireless mobile communications, new stream cipher implemented by Single-Block hash function based on Hash chain method. In this new algorithm, the Single-Block hash function processes loop operation adopted Hash chain method firstly, then links the output of each operation into the Stream Cipher key sequence, finally cipher text is obtained by bit XOR operation of plaintext and key sequence. After the realization of SC-SBH, we have tested its randomness and security, and have made comparison with SC-MD5 and SC-SHA in efficiency. The experimental results showed that the SC-SBH algorithm not only can ensure safety, but also have high efficiency relative to SC-MD5 and SC-SHA.

Key words: stream cipher; Single-Block hash function; Hash chain; running efficiency; security

序列密码因运行速度快，硬件需求低，且具有安全性高、易同步、无错误传输等优异特性，在无线移动通信中有着广泛的应用，特别是在高速通讯、高噪声信道传输等领域中[1]. 为了能更好地将序列密码应用于无线移动通信领域，在现代 Estream 计划中，许多新的序列密码设计方法被采用，如移位寄存器与其他装置的结合、基于状态表驱动、利用分组密码的部件或思想；一些单纯的基于一种装置来设计序列密码的特殊算法也被提出，如基于神经网络的算法、基于层叠跳转控制序列的算法、基于伪随机数生成器的算法等. 但这些算法目前仍很难做出系统的安全分析，且运行速度也有待提高[2]. 本文通过用哈希链方法对单分组散列函数进行迭代运算，将中间结果输出，连接中间结果作为序列密码的密钥序列，以此实现新的序列密码模型，并且通过实验证明了本文方法的安全性和高效性.

收稿日期：2015-06-26

* **通信作者**：李永珍(1971—)，男，博士，副教授，研究方向为网络安全、无线网络协议.

1 序列密码算法的原理与结构

1.1 序列密码算法的原理及原则

序列密码的构成可用一个6元组来描述,即P,C,K,Z,E_k,D_k,其中P代表明文空间,C代表密文空间,K代表密钥空间,Z代表密钥流生成算法,E_k和D_k代表密钥序列与明文的加密和密文的解密规则,通常为异或运算.对密钥$k \in K$,Z确定一个密钥序列[3].序列密码算法加解密的原理如图1所示.

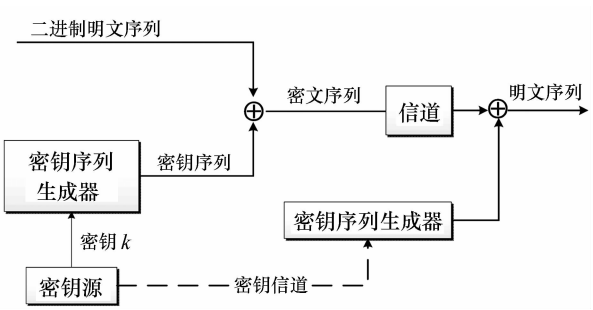

图1 序列密码的工作原理

加密过程可表示为$C_i = P_i \oplus K_i$,解密操作可表示为$P_i = C_i \oplus K_i$,$i \geqslant 1$.其中:二进制序列$P_0,P_1,\cdots,P_i$是明文序列,$P_i \in \mathrm{GF}(2)$,$i \geqslant 1$;$K_0,K_1,\cdots,K_i$是密钥序列,$K_i \in \mathrm{GF}(2)$,$i \geqslant 1$;$\oplus$表示按位异或运算,即模2加运算.

在序列密码系统中,加密原则为:1)明文序列与密钥序列逐位进行加密,所以密钥序列一定要与明文的长度相同;2)序列密码的密钥序列随时间变换,加密算法"一次一密",所以在序列密码中相同明文对应的密文也不同.

根据加密原理及原则可知,密钥序列的伪随机性决定了序列密码的安全强度.研究[4]表明,将无记忆的、离散的二进制均匀分布信源产生的随机序列作为密钥序列,在理论上虽然被证实是安全可靠的,但在实际中是不可能的.文献[4]还表明,仅当密钥个数与明文个数至少一样多时,"一次一密"才是完全保密的,即密钥长度至少和明文长度一样长且不重复使用时才是安全的.但由于过长的密钥序列在实际中不便于存储和分配,所以设计序列密码的主要目标就是研究如何用一个短的密钥生成一个周期长且安全性高的密钥序列.

1.2 序列密码中几种构造密钥生成器的方法

1)线性反馈移位寄存器构造密钥序列生成器.该密钥生成器的典型算法是RC4算法[5-6],它是一个密钥长度可变、面向字节操作的序列密码,以随机置换作为基础.由于该算法易于软件实现,而且安全性高,因而得到非常广泛的应用.

2)基于分组密码构造密钥序列生成器.该密钥序列生成器的典型算法是LEX算法[7-8],其密钥序列生成器借鉴了分组密码的成熟设计理念,易于理解和实现.

3)基于布尔函数构造密钥序列生成器.将Bent函数使用在非线性组合序列生成器中,非线性组合函数$f(x)$的选取能够保证非线性组合序列具有较好的随机特性、较高的线性复杂度和较强的抗破译性,但是使用Bent函数具有不平衡、无相关免疫性等缺陷[9-10].

2 基于哈希链的序列密码算法设计

2.1 单分组散列函数

本文提出的基于哈希链方法的序列密码算法,其密钥序列是利用单分组散列函数设计的.单分组散列函数SBH是以哈希函数的设计原理和准则为理论依据来进行设计的,它是一种针对短消息进行处理的散列函数.由于单分组散列函数优化了MD5的运算,减少了不必要的计算,即缩短了运算周期,因此在能够保证安全的同时,还能够提高消息的处理效率.SBH算法的输入和输出均为128位,因此也被称为SBH-128[11].该算法使用4个寄存器,主循环有4轮,每轮4步.

运算时,首先将128位的初始值分成4个组(每组32位),分别与A、B、C、D 4个寄存器常量进行运算,一共运算16步.在每步运算结束时,将当前获得的运算结果替换缓存器保存的前一轮的值,以此来更新缓存内容.算法的输出值是由4个32位缓存寄存器经过16步运算后得到的结果构成的一组128位哈希码.图2为SBH算法主循环部分的逻辑图.在图2中:F,G,H,I是4轮运算

中应用的非线性函数；ρ_2，ρ_3，ρ_4 是 128 bits 分组 $X[i]$ 在第 2、3、4 轮中的使用顺序；K_1，K_2，K_3，K_4 是常量.

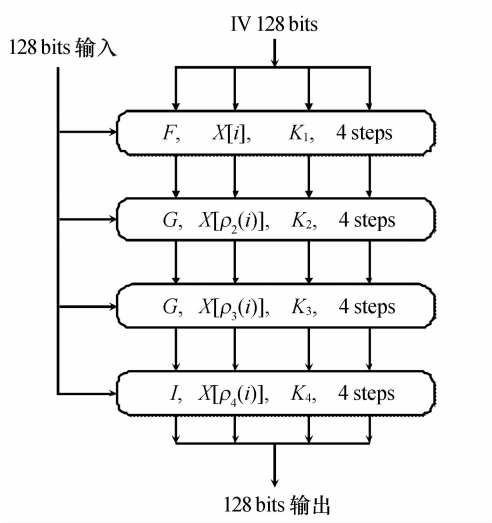

图 2　SBH-128 算法整体逻辑图

2.2　基于哈希链的序列密码算法结构

基于哈希链的序列密码算法的设计思想主要是：将 SBH-128 用哈希链方法进行迭代运算，然后把每次运算得到的中间结果连接成为密钥序列，最后将密钥序列与明文进行按位异或运算得到密文.

一个单分组散列函数输出的是 128 位的散列码. 运算时，首先用哈希链的方法对 SBH-128 的 128 位散列码进行处理，即把当前一次的 SBH-128 输出值作为下一次的 SBH-128 输入值，然后把每一次 SBH-128 的运算结果记录并输出，以此形成 128× n 位的序列作为算法的最终密钥序列. 图 3 为序列密码 SC-SBH 密钥生成器设计逻辑图.

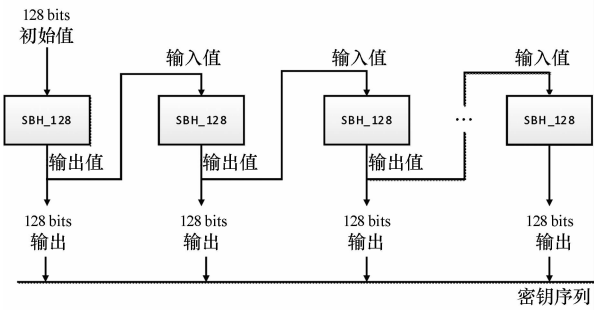

图 3　SC-SBH 密钥生成器设计逻辑图

SC-SBH 密钥生成器的设计步骤为：

1）设计密钥生成器的初始值. 由于序列密码要求加密时“一次一密”，所以本文用 currentTimeMillis 方法生成初始值.

2）运算次数. 在设计密钥生成器时，当明文长度是 128 × n 位时，哈希链进行 n 次哈希运算，其结果直接与明文进行按位异或运算即可；当明文长度不是 128 × n 位时，需要将明文长度与 128 进行求模运算，得到模 m 后，哈希链再进行 $m+1$ 次运算. 这是因为序列密码加密要求明文和密钥要有相同的长度，所以，明文与密钥按位异或运算，密钥多出来的位自动舍弃.

3）组成密钥序列. 把单分组散列函数每一次循环运算的中间结果值输出，连接成 128 × n 位的密钥序列.

4）生成密文. 将明文与得到的密钥序列进行位异或运算得到密文.

由于 SC-SBH 的密钥生成器部件的设计是以单分组散列函数为基础，继承了其针对短消息处理的功能，因此该方法实现了具有非线性特性的新型序列密码算法.

3　实验及结果分析

本实验采用的编程语言是 Java 语言，编译工具是 Eclipse(JUNO 版本)，实验运行环境是 PC 机(配置为 Intel Core 3.20 GHZ，4.00 GB ROM). 根据 Golomb 提出的随机性假设要求，本文从“0/1”平衡和“0/1”游程个数[12-13]的角度分析 SC-SBH 的安全性，并将 SC-SBH 同 SC-MD5(基于 MD5 的哈希链方法序列密码算法)和 SC-SHA(基于 SHA 的哈希链方法序列密码算法)进行效率对比.

3.1　“0/1”平衡随机性测试

Golomb 提出的“序列应满足的随机性假设”中第 1 条要求为：在序列的一个周期内，0 和 1 的个数相差至多为 1，即要求 0 和 1 出现的概率基本相同.

用 SC-SBH 加密算法分别对 50 KB、500 KB、1 MB、5 MB、10 MB 不同大小的明文进行了加密，

并记录密文中“1”的个数所占 2 进制密钥序列长度的比例. 从表 1 的实验结果中可以看出,“1”所占的比例约为 53%,因此本文设计的序列密码算法 SC-SBH 生成的密钥序列的“0”和“1”个数较为平衡,基本符合 Golomb 提出的“序列随机性假设”中的第 1 条要求.

表 1 SC-SBH 的平衡性测试结果

加密明文大小	50 KB	500 KB	1 MB	5 MB	10 MB
密文中 1 所占比例/%	53.17	53.01	53.03	53.04	53.04

3.2 游程随机性测试

Golomb 提出的“序列应满足的随机性假设”中第 2 条要求为:长度为 T 的周期内,任意长度的 0 的游程个数与 1 的游程个数相同.

游程定义:一个二元序列中,形如 100…001 的片段称为该序列的一个 0 游程,形如 0111…110 的片段为该序列的一个 1 游程,并称 0 游程中 0 的个数为该游程的长度,1 游程中 1 的个数为该游程的长度.

约定:0 开头的信号片段 00…01 为一个 0 游程,1 开头的信号片段 11…10 为一个 1 游程,以 0 结束的信号片段 100…0 为一个 0 游程,以 1 结束的信号片段 011…1 为一个 1 游程.

本测试实验中利用单分组散列函数进行 1 000次迭代,累加生成 12.8 万个长度的密钥序列作为一个周期,然后对这个周期内密钥序列中长度为 2、3、4、5 的 0 的游程和 1 的游程的个数进行统计和对比.

用 SC-SBH 加密算法分别对 50 KB、500 KB、1 MB、5 MB、10 MB 不同大小的明文对应生成的密钥序列进行了游程测试,并记录了不同长度密钥序列的不同长度的 0 的游程和 1 的游程的个数,结果如表 2 所示. 从表 2 中可以看出,0 的游程个数和 1 的游程个数基本相同,因此,本文设计的 SC-SBH 算法生成的密钥序列的游程测试基本符合 Golomb 提出的“序列随机性假设”中的第 2 条要求.

表 2 SC-SBH 的游程测试

长度	Count 0 的游程个数	Count 1 的游程个数
2	7 063	7 053
3	3 796	3 786
4	1 945	1 930
5	1 006	1 005

3.3 效率性能分析

为验证 SC-SBH 算法的效率性能,将 SC-SBH 算法的效率分别与 SC-MD5 和 SC-SHA 算法的效率进行了对比. 实验中,3 种算法均对 50 KB、500 KB、1 MB、5 MB、10 MB 不同大小的明文进行 100 次加密处理,实验记录数据如表 3 所示. 3 种算法的效率示意图如图 4 所示.

表 3 3 种算法对明文加密运行的平均时间

算法类型	不同大小明文平均加密时间/ms				
	50 KB	500 KB	1 MB	5 MB	10 MB
SC-SBH	44	250	469	2 202	4 428
SC-MD5	63	279	531	2 421	4 674
SC-SHA	47	281	547	2 549	4 999

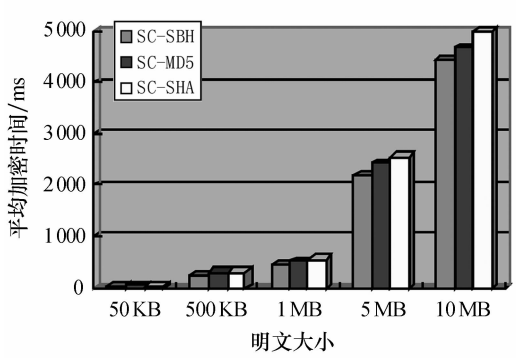

图 4 3 种算法的效率示意图

由表 3 和图 4 可以看出,在相同大小的明文和运行环境下,SC-SBH 算法的运行效率优于 SC-MD5 和 SC-SHA 算法. 为了更准确地显示出 SC-SBH 算法效率的提高率,实验又对 SC-SBH、SC-MD5、SC-SHA 算法的效率进行了再计算,结果如表 4 所示.

表 4　SC-SBH 算法相对于 SC-MD5 和 SC-SHA 算法的运行效率的提高率　　%

	不同大小明文效率提高率				
	50 KB	500 KB	1 MB	5 MB	10 MB
SC-SBH 相对于 SC-MD5	16.13	10.39	11.68	9.05	5.26
SC-SBH 相对于 SC-SHA	6.38	11.03	14.69	13.61	11.42

4　结论

本文提出的 SC-SBH 算法的创新之处在于它将单分组散列函数应用于哈希链方法的序列密码算法. 经实验分析表明，本文提出的方法与 MD5 和 SHA 算法相比，具有安全性好、效率高、硬件消耗资源小、软件易实现等优点，可应用于“保证常规安全性前提下快速加密”的数据保护，尤其是无线移动通信的数据保护. 本文对算法的安全性和效率进行了分析和验证，今后还需对其抗攻击性做进一步验证分析，以期得到更加完善的算法.

参考文献：

[1] William Stallings. 密码编码学与网络安全-原理与实践[M]. 王张宜，杨敏，杜瑞颖译. 5 版. 北京：电子工业出版社，2012.

[2] Halevi S, Coppersmith D, Jutla C. Scream: a software-efficient stream cipher[C]//Fast Software Encryption (FSE) 2002. Lecture Notes in Computer Science, 2002:195-209.

[3] 王相生. 序列密码设计与实现的研究[D]. 上海：中国科学院上海冶金研究所，2001.

[4] 宫大力. 流密码算法的研究与设计[D]. 南京：南京航空航天大学，2011.

[5] 张海纳. eSTREAM 序列密码候选算法的安全性分析[D]. 山东：山东大学，2009.

[6] Singhal N, Raina J P S. Comparative analysis of AES and RC4 algorithms for better utilization[J]. International Journal of Computer Trends and Technology, 2011,2(6):177-181.

[7] 刘依依. eSTREAM 和流密码分析现状[J]. 信息安全与通信保密，2009(12):47-49.

[8] Luo Y, Chai Q, Gong G, et al. A lightweight stream cipher WG-7 for RFID encryption and authentication[C]//Global Telecommunications Conference (GLOBECOM 2010), IEEE, 2010:1-6.

[9] 武传坤，王新梅. Bent 函数在流密码中的应用[J]. 通信学报，1993,4:23-27.

[10] Gupta S S, Chattopadhyay A, Sinha K, et al. High-performance hardware implementation for RC4 stream cipher[J]. IEEE Transactions on Computers, 2013,62(4):730-743.

[11] 王静雅. 单分组散列函数的设计与应用[D]. 延吉：延边大学，2014.

[12] 尤加勇. 现代序列密码的设计与分析[D]. 北京：国防科学技术大学，2007.

[13] Sharif S O, Mansoor S P. Performance analysis of Stream and Block cipheralgorithms [C]//Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on. IEEE, 2010,1:522-525.