改进型多态性密码的安全机制研究

尹毅峰[1,2] 胡予濮[1]
(西安电子科技大学CNIS重点实验室 西安 710071)[1]
(陕西国防工业职业技术学院电子工程系 西安 710300)[2]

摘 要 和大多数通常已知的对称性加密算法(包括AES的代表算法,例如Rijndael和Twofish)相比较,多态性密码可以抵御差分密码攻击。这种算法主要用于磁盘文件的加密。要解决的问题是在P2P网络中改进多态性密码机制,为通信双方提供大量安全的会话密钥。构造了满足严格雪崩准则和输出位独立性的强单向函数,并对其进行了安全性分析,用实验数据和相关理论进行了论证。

关键词 多态性密码,P2P,严格雪崩准则,强单向函数
中图法分类号 TP309 **文献标识码** A

Security Analysis of Mechanisms for the Improved Polymorphic Cipher

YIN Yi-feng[1,2] HU Yu-pu[1]
(Key Laboratory of Computer Network & Information Security, Xidian University, Xi'an 710071, China)[1]
(Electronic Engineering Department, Shanxi Institute of Technology, Xi'an 710300, China)[2]

Abstract In contrast to most or all commonly known symmetric encryption algorithm designs (including the AES candidates such as Rijndael and Twofish), the Polymorphic Cipher (PMC) can immunize differential power attack. The algorithm is mostly used to encrypt disk files. The problem that would be solved is to improve the Polymorphic Cipher in P2P, and to provide mass-produced session keys for two parties across a communication channel. A strong one-way function satisfing the Strict Avalanche Criterion (SAC) was constructed. The security of the function was analyzed by some experimental results and correlated theories.

Keywords Polymorphic cipher, Bit independence criterion, Strong one-way function

设计流密码的主要问题是如何设计一个能够产生足够随机数据位的密钥流发生器。为了对抗差分密码分析和穷举攻击,构造复杂的迭代密码,C. B. Roellgen于2002年提出了一种全新的密码机制——多态性密码(PMC)理论[1],该理论提供了可实现的、自编译的密码通用模式。PMC的构造思想是提供一组简单单向函数$\{f_1, f_2, \cdots, f_n\}$,利用攻击方攻击所产生的参数启动自编译系统,进行函数的交叉组合,经过多轮迭代后,获得满足密码学特性的输出序列,以此抵抗差分密码分析和穷举攻击。

因为不便于携带大量的密钥流序列,流密码要求加密算法必须能够利用短密码来产生密钥流序列。多态性密码是一种对称型加密算法,基本上属于流密码的范畴,在具体实现上又是分组和流密码的混合,为自编译类的加密算法提供了一种全新的密码反馈模式。多态性密码采用一个伪随机序列发生器[2]产生一个置乱的序列,此序列直接和明文位进行XOR运算,加解密速度也明显高于AES。该理论已经被用于磁盘文件的加密。

本文提出将多态密码机制应用于P2P网络,解决双方通信中会话密钥的产生问题。这样,可以将该密码机制的安全性和高速性两个优点从单机的磁盘文件加密移植到网络环境。我们从输出位独立性和满足严格雪崩准则[3]两个方面分析这种改进算法的安全性。

1 网络节点的随机特性

在网络中,通信双方的节点之间可以构成一个序列,每个节点能够同时完成上传数据以及下载数据的操作。每个节点既是服务器又是客户机[4];利用节点i的特点设置相关的矢量$v_i(u_i, d_i)$,u_i描述该节点状态的上传状态,d_i描述该节点的下载状态。由于每个节点在整个网络中承担相邻节点的上传和下载任务,可能处于多条通信链路中,因此第i个节点的v_i具有很强的随机性。其次,根据实际的路由算法,网络中的通信链路的路由变化性较强,所以从发送方到接收方节点之间的路由的变化以及通信链路中每个节点的随机性很强的v_i进行组合,必然构造出一个具备物理上随机性的随机通信链路序列。再者,每个节点可能处于两个或两个以上的不同通信链路中。如图1所示,通信链路1经过由$Peer_{11}, \cdots, Peer_{1i}, \cdots, Peer_{1n}$节点构成的链路,假设从所有包含$Peer_{1i}$节点的通信链路组中任选另一通信链路2,通信链路2由

到稿日期:2008-12-15 返修日期:2009-03-02 本文受国家自然科学基金项目(60473029)资助。
尹毅峰(1971-),男,副教授,主要研究领域为网络安全、P2P网络,E-mail:yinyifeng@yeah.net;胡予濮(1955-),男,教授,博士生导师,主要研究领域为网络与信息安全。

$Peer_{21}$,…,$Peer_{2j-1}$,$Peer_{2j}$,$Peer_{2j+1}$,…,$Peer_{2n}$ 节点构成，其中 $Peer_{1i}$ 和 $Peer_{2j}$ 两个节点重合，每个瞬间 v_{1i} 的值都不同，该节点用户也无法重现前一瞬间的值。

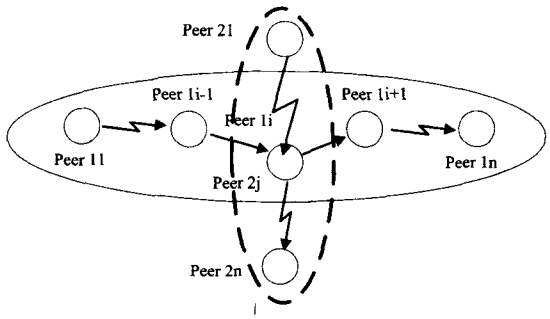

图 1　处于两个不同通信链路中的节点

网络中每个节点的网络数据包到达次数具有很大的随机性，因为每个节点对应一个上网的计算机用户，该用户可能做任何操作，可能上传资料，也可能下载文件。而且，当该节点用户上网使用软件时，该用户就成为通信链路上的一个“种子”，承担了上传和下载的任务。由于该用户所处的网络环境中带宽固定，同时上网的用户数量具有随机性，因此该用户的通信速率是随机变化的，并且该用户的上网时间也具备随机性，由此某个瞬间，此节点的上传字节数和下载字节数具有很强的随机性，人力无法预测。

2　改进多态性密码

改进型多态密码的核心部件是 PRNG[5]。PRNG 存在的充要条件是强单向函数的存在[6]，所以本文提出了利用网络中节点捕获的网络数据包字节数作为随机种子来构造一个具备密码学特性的强单向函数，用这个强单向函数来替换多态密码原有的自编译器。

提供一组简单的单向函数，如下所示：

$f_1(s_{r-1},u_i)=s_{r-1}+a\cdot 2^{32h(u_i)}\cdot u_i$

$f_2(s_{r-1},d_i)=s_{r-1}+d_i\cdot 2^{32h(d_i)}+$进位

$f_3(s_{r-1},u_i\oplus d_i)=2^{32h(u_i\oplus d_i)}\cdot(u_i\oplus d_i)\cdot s_{r-1}+$进位

$f_4(s_{r-1},u_i+d_i)=(2^{32h(u_i+d_i)}\cdot(u_i+d_i))\oplus s_{r-1}$

构造方法如图 2 所示。设 u_i 和 d_i 分别为节点 i 的上传数据包字节数和下载数据包字节数，两者都为 32 位。由此构造 4 个 32 位的随机数据块 u_i+d_i，$u_i\oplus d_i$，d_i 和 u_i，作为自编译器 Hash 函数的输入，得到两个结果：一个 32 位的序列 $htbuf$ 和一个整数 j。利用整数 j 从 16 个候选单向函数集合中包含的 4 个子函数 g_1，g_2，g_3 和 g_4 的结果组合成该节点的 128 位输出数据。其中：s_0 为发送方 *Alice* 所选择的不具备密码学安全的 32 位随机数，s_{r-1} 为通信链路中节点前一个状态所产生的 128 位输出，s_r 为第 r 轮后 Alice 所得到的 128 位的随机序列。

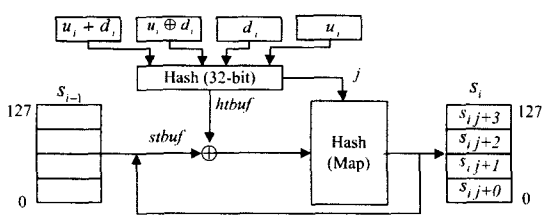

图 2　强单向函数构造图

1)输入

· 随机种子：发送方 Alice 选取一个 32 比特的数作为 PRNG 的初始种子，该数可以为从网络截获的数据包字节数。为了满足算法的需要，s_0 可以设为 128 比特，不具备密码学的安全性，其中的低 32 比特被填充，其余比特位为 0。

· 集合 $X_i\{u_i,d_i,u_i\oplus d_i,u_i+d_i\}$，被用于哈希函数的输入，可以产生 4 组 32 比特的序列。

2)输出

s_i：第 i 次迭代所产生的临时 128 比特数。

3)内部数据结构

· $stbuf$：32 比特的缓存变量，初始值为 s_0 的低 32 比特。$stbuf$ 为 s_{i-1} 中的一组 32 比特的数。仿照输出反馈模式(OFB)[8]，在得到最终结果前，哈希映射函数为之提供 32 比特的反馈序列。

· htbuf：算法执行后，在固定时间间隔依次产生 4 组 32 比特的序列。$htbuf$ 数组中的内容(128 比特)可以用于和 $stbuf$ 依次进行 XOR 操作。

· Hash 函数：哈希函数的目的是产生每组数据的“指纹”[9]，该算法由 16 个简单 PRNG 组成，令 u_i，d_i，$u_i\oplus d_i$ 和 u_i+d_i 依次作为该函数的输入参数，从而每次获得不同的 $htbuf$ 和 j 的值。

· Map 映射：用于产生 stbuf 的反馈值和每组 s_i. s_i 中的 32 比特的数据位置依赖于 j。

图 2 描述的算法归纳为如下 8 个步骤：

步骤 1　[初始化随机种子]

网络节点中嵌入的算法获得 s_i.，发送方 *Alice* 为 $stbuf$[0…3]选取 32 比特的随机数。令 $stbuf$[4…7]为 0，此步骤得到 $stbuf$ 的初值 s_0。

步骤 2　[从发生源采集随机数]

u_i←数据上传所产生的即时数据包字节数。

d_i←数据下载所产生的即时数据包字节数。

$flags$[0…3]←0

步骤 3　[产生 $htbuf$ 比特]

a)在固定时间间隔内重复步骤 3—7 四次，每次哈希函数利用 16 个 PRNG 的组合输出一个值 j，并依次将该值送给 Map 映射函数。

b) $flags[j]$ ←1，设置标志位。

步骤 4　[判断输出序列是否有效]

若用户认可输出，则转去执行步骤 8 或者步骤 5，接收者 Bob 在此步骤可随时决定数据是否有效。

步骤 5　[产生一组新的伪随机序列]

将 $stbuf\oplus htbuf$ 作为 Map 映射函数的输入。

步骤 6　[每次从 Map 传送 32 比特数据给 s_i]

a)映射 $stbuf\oplus htbuf$ 的值给 $s_{i,j}$；

b) $stbuf$←$stbuf\oplus htbuf$。

步骤 7　[是否复位 j?]

若 $flags$[0..3]全为 1，则转去执行步骤 2(重新采集)或者步骤 3。

步骤 8　[结果]

得到 s_i。

在步骤 3 中，j 的值是哈希函数的输出，依赖于 s 向量 v_i (u_i,d_i)的值，即上传数据和下载数据的状态，而且步骤 4 的执行时间是不确定的，由接收方随机控制。

3 位独立性准则

Webster 和 Tavares 提出了输出位独立(BIC)准则[7]。对于函数 $f:GF(2)^n \to GF(2)^n, i,j,k \in \{1,2,3,\cdots,n\}$ 且 $j \neq k$,当输入序列中的第 i 位取反时,会引起输出序列中第 j 位和第 k 位各自独立地改变,这样称函数 f 满足位独立性准则(BIC)[8]。

为了衡量函数的位独立性,需要为输出序列的第 j 位和第 k 位设置相关性系数 $BIC^{e_i}(a_j,a_k)$,可以描述两个随机变量之间的相关程度的指标[9],也称为平衡向量 A^{e_i},其中 e_i 为单位向量,$a_j^{e_i} \in \{0,1\}$。当第 i 位发生改变时,平衡向量可定义为:

$$A^{e_i} = f(X) \oplus f(X \oplus e_i) = [a_1^{e_i} a_2^{e_i} \cdots a_n^{e_i}]$$

利用平衡变量可以测量当明文中 1 比特发生改变时密文序列位改变的随机性、输入序列对族 A 和 A_i,其中 $A = A_i \oplus e_i$。

令 $a_j, a_k \in A$,则输出序列的第 j 位和第 k 位的相关性系数为:

$$BIC^{e_i}(a_j,a_k) = |corr(a_j^{e_i}, a_k^{e_i})| = \max_{i=1,2,\cdots,n} \left| \frac{\frac{1}{2^n} \sum_{A \in \{0,1\}^n} [a_j(A) \oplus a_j(A \oplus e_i)][a_k(A) \oplus a_k(A \oplus e_i)] - \overline{a_j a_k}}{(\overline{a_j} - \overline{a_j}^2)(\overline{a_k} - \overline{a_k}^2)} \right|$$

其中,$\overline{a_j} = \frac{1}{2^n} \sum_{A \in \{0,1\}^n} a_j(A) \oplus a_j(A \oplus e_i)$。

函数 f 的相关系数即为最大相关系数[10],可定义为:

$$BIC(f) = \max_{j \neq k} BIC(a_j, a_k)$$

其取值范围在 $-1 \sim +1$ 之间。绝对值越接近 1,表示两个变量之间的相关程度越高;接近 0,表示相关性越差,0 表示没有相关性。

如果函数 f 满足位独立性准则,输出序列的第 j 位和第 k 位不相关,第 j 位的变动不会影响第 k 位的变动;$BIC(f)=1$ 时,第 j 位和第 k 位完全相关,第 j 位和第 k 位已经变成了函数关系;$0 < BIC(f) < 1$ 时,即为第 j 位的变动引起第 k 位变动的概率,$BIC(f)$ 的绝对值越大,引起对第 k 位变动的概率就越大;$BIC(f) > 0.8$ 时称为高度相关,$BIC(f) < 0.3$ 时称为低度相关,其他为中度相关。

4 实验分析

为了验证改进方法的严格雪崩准则和输出位独立性,在西安电子科技大学的 CNIS 实验室开发了实验系统,构建了带宽为 100Mbps 的实际可用的 P2P 网络环境[11],进行了大量的实验。为了仿真实际的网络环境,参与实验的网络对等点用户自由随意地进行各种网络访问,包括浏览网页、FTP、VOD 视频点播、聊天、收发电子邮件等。在实验过程中,选择了 6 个不同长度(32-比特、64-比特、128-比特、256-比特、512-比特和 1024-比特)的明文序列,经过 10h 的网络数据包的随机获取,得到近 1000 组 $\{u_i + d_i, u_i \oplus d_i, d_i$ 和 $u_i\}$ 作为 PMC 的参数。由于本文设定 PMC 的 4 个参数为 32 位,而实验所需的 n 的长度是 32 的倍数,因此进行网络参数采集时,要根据 n 的大小设置参数的采集次数。例如当 $n=128$ 时,需要从网络数据包中采集 4 组 $\{u_i + d_i, u_i \oplus d_i, d_i$ 和 $u_i\}$。

在实验中必须对输出进行量化分析,即随着输入参数的变化,得到输出 s_i 的变化程度。可以利用海明距离来观察两个输入和一个输出的改变程度,理想的结果应该是 d_i 和 u_i 位改变很少的情况下,输出 s_i 有一半位发生改变,可以用概率统计来量化输出的改变程度。

图 4 和图 5 显示了对等点上所安装的系统在 30ms 时段内截获的 d_i 和 u_i 的位变化情况。利用海明距离 d_h 和 32 的比来显示两个参数变化的程度,则令经过伪随机数产生器内部的强单向函数处理后,得到图 6 所示的 128 位的输出序列位变化率,即 $\delta d = \frac{d_h(d_i, d_{i-1})}{32}$,$\delta u = \frac{d_h(u_i, u_{i-1})}{32}$,$\delta s = \frac{d_h(s_i, s_{i-1})}{128}$。实验数据表明,128 位输出的位变化率较接近 0.5。也就是说,因为 32 位输入参数的随机变化,引起 128 位输出的每个位较均匀地发生改变。

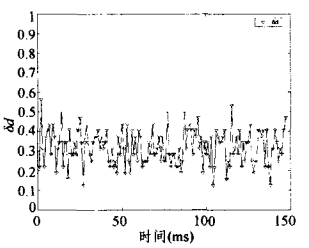

图 4 150ms 时间段内 d_i 的位变化率

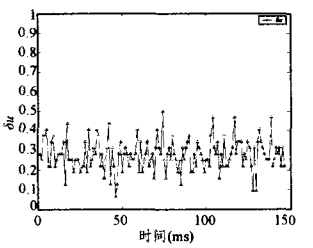

图 5 150ms 时间段内 u_i 的位变化率

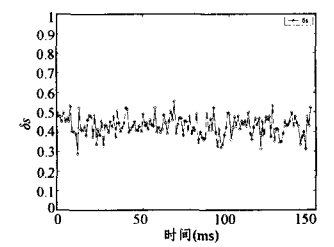

图 6 同时间段内与 d_i 和 u_i 所对应的 128 位输出的位变化率

为了验证改进方法输出位独立性准则,在实验中必须对输出进行量化分析。可以利用海明距离来观察海明距离为 1 的两个输入向量和两个输出向量的改变程度,用概率统计输出序列中两个 Bit 的相关程度。实验中选取了不同长度的输出序列,$n=8,16,32,64,128$,根据网络节点上所安装的系统在一个时段内因所截获的 d_i 和 u_i 的位而引起的输出序列的变化情况,可以统计出不同长度输出序列所得到的相关性系数,随着 n 的增大,相关性系数取值快速下降,根据本文所构造的强单向函数 f 得到的 $BIC(f)$ 来分析改进后的多态密码是否满足位独立性准则。

结束语 本文中主要对改进的多态密码的位独立性进行了研究。由于所构造的强单向函数的 4 个相关参数受到网络节点的直接影响,引进 PMC 的加密机制后,可以快速而且大量地产生符合密码学特性的长周期输出序列,构造安全的 S 盒至少要满足 3 个特性,即严格雪崩准则、位独立性准则和非线性特性,进一步的研究目标是构造出高非线性度的布尔函数。

参 考 文 献

[1] Roellgen C B. Polymorphic Cipher Theory [OL]. http://www.pmc-ciphers.com/technology/roellgen02generalizedPMCmodel.

pdf
[2] Bucci M, Luzzi R. Design of Testable Random Bit Generators in Cryptographic Hardware and Embedded Systems[C]// CHES 2005. Springer-Verlag, 2005: 147-156
[3] Yin Yifeng, Li Xinshe, Hu Yupu. Fast S-box security mechanism research based on the polymorphic cipher[J]. Information Sciences, 2008, 178(6): 603-1610
[4] Ripeanu M, Foster I. Mapping the Gnutella Network: Macroscopic Properties of Large-scale Peer-to-Peer Systems[C]// IPTPS '02. MIT Faculty Club, Cambridge, MA, USA, March 2002
[5] Viega J. Practical Random Number Generation in Software[C]// Proc. 19th Annual Computer Security Applications Conference. Dec. 2003
[6] Goldreich O. Foundations of Cryptography Basic Tools [M]. Press Syndicate of the University of Cambridge, 2001: 75-89
[7] Menezes A, Oorschot P, Vanstone S. Handbook of Applied Cryptography [M]. CRC, 1996: 169-190
[8] 胡予濮, 张玉清, 肖国镇. 对称密码学[M]. 北京: 机械工业出版社, 2002: 56-57
[9] Molina M, Niccolini S, Duffield N G. A Comparative Experimental Study of Hash Functions Applied to Packet Sampling[C]// Proceedings of ITC-19. Beijing, China, 2005
[10] Stallings W. Network Security Essentials: Applications and Standards[M]. New Jersey: Pearson Education, 2004: 164-166
[11] 张三峰, 陈刚, 于坤, 等. P2P 网络可用性关键技术研究进展[J]. 计算机科学, 2008(6)

(上接第 55 页)

在网络节点中是分布式的，因而非源节点可能需要增加额外的存储负担。

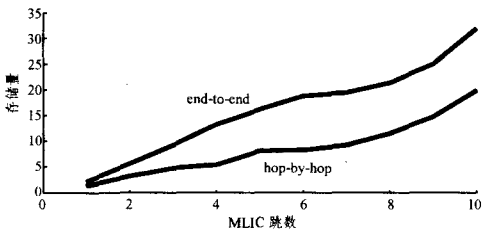

图 6 存储开销比较

结束语 在改善 MLIC 网络性能方面，hop-by-hop 方式的性能通常要高于 end-to-end，但后者也同样伴随着高协议复杂度和额外的内存、处理器需求等不利因素。为了简化问题的复杂度，本文在建模和分析中作出了许多的假设，这使得文章具有一定局限性，这些也正是我们下一步需要继续研究的方向。

· 模型没有考虑从源到目的节点之间的节点移动性及相应的路由变化，而路由变化将导致包传递路径长度的改变，进而对成功递送率、发送次数、缓存使用等产生影响。

· 本文假定每一跳的链路状态改变都是独立的，然而在真实网络中，相邻两跳链路上的相互干扰等因素将导致链路状态具有一定的相关性。

· 文章简单地使用 FIFO 机制来管理缓存，将来的工作将研究 MLIC 环境下更合适的缓存管理机制，以更好地利用分布于全网的缓存。

实际上，减少 MLIC 影响可以分为预先式和响应式两种方法。前者在断开连接之前执行预先的行为，这需要对何时可能发生间歇做出相对准确的预测，如针对卫星等有着周期性可预测的链路，但对于一般空间节点这种预测是很难实现的，因为一般节点很少知道或者没有关于将来网络拓扑的变化的知识，节点移动也是随机的和未知的。后者在连接断开后执行矫正行为。本文主要关注响应方法，针对卫星等可预测链路间歇连通性研究以及与本文所研究的响应式方法的结合等问题，也将作为下一步的研究内容。

参 考 文 献

[1] Aguayo D, Bicket J, Biswas S, et al. Link-level measurements from an 802.11b mesh network[C]// ACM SIGCOMM '04. New York, NY, USA, Aug. 2004
[2] Lenders V, Wagner J, May M. Analyzing the Impact of Mobility in Ad Hoc Networks[C]// ACM REALMAN. Florence, Italy, May 2006
[3] Burleigh S, et al. Delay-Tolerant Networking - An Approach to Interplanetary Internet[J]. IEEE Communications Magazine, July 2003
[4] Cerf V, Burleigh S, et al. Delay-Tolerant Networking Architecture[J]. RFC4838, April 2007
[5] Khan J I Dr. A Commentary on the Nature of Mobility and Connectivity in Space and Implications on the Design of Space Network Protocols[R]. An report to NASA Space Networking Initiative, August 2005
[6] Almeroth K C. Distinguishing between connectivity, intermittent connectivity, and intermittent disconnectivity [J]. Challenged networks, 2007: 1-2
[7] Marshall P. Disruption Tolerant Networking Industry Day BAA 04-13[OL]. http://www.darpa.mil/ato/solicit/DTN/DTN industryday.pdf, Jan. 2004
[8] Akyildiz I F, Zhang X, Fiang J. TCPPeach+: Enhancement of TCP-Peach for Satellite IP Networks[J]. IEEE Communications Letters, 2002, 6: 303-305
[9] Heimlicher S, Karaliopoulos M, Levy H. End-to-end vs. Hop-by-hop Transport under Intermittent Connectivity [C]// Martin May Proceedings of Autonomics 2007. ICST, Rome, Italy, October 2007
[10] Viswanathan R, Li J, Chuah M C. Message Ferrying for Constrainted Scenarios[C]// poster. Proc. of WoWMoM. 2005
[11] Elaarag H. Improving TCP Performance over Mobile Networks [J]. ACM Computer Surveys, 2002, 34(3)
[12] Akan O B, Fang J, Akyildiz I F. TP-Planet: A Reliable Transport Protocol for Interplanetary Interne[J]. IEEE Journal of Selected Areas in Communications (JSAC), Feb. 2004, 22: 384-361
[13] Seligman M. Storage usage of custody transfer in delay tolerant networks with intermittent connectivity[C]// Proceedings of ICWN06. June 2006
[14] Farrell S, Symington S, Weiss H, et al. Delay-Tolerant Networking Security Overview. Internet-Draft[J]. February 2008
[15] 周晓波, 周健, 卢汉成, 等. DTN 网络的延时模型分析[J]. 计算机研究与发展, 2008, 45(6): 960-966