第34卷 第18期 计 算 机 工 程 2008年9月
Vol.34 No.18 Computer Engineering September 2008
·安全技术· 文章编号：1000—3428(2008)18—0181—03 文献标识码：A 中图分类号：TP393.08

RC4密码的改进方法及其性能分析

李 琴，曾凡平
(中国科学技术大学计算机科学技术系，合肥 230027)

摘 要：针对RC4密码技术在工程应用中存在的弱密钥和相关密钥攻击、不变性弱点、数据流偏向性弱点等安全问题，提出一种将ECC技术与RC4技术相结合的方法。对改进后的RC4的数据处理效率、密钥管理、安全性能进行研究和分析。改进后的RC4技术在保证与RC4数据处理效率相近的同时，对当前针对RC4流密码的密码分析方法具有一定的抗攻击性。该技术较好地解决了密钥的共享和更新问题，具有重要的工程应用意义。

关键词：RC4技术；改进的RC4；密钥协商；并行的类FPK散列算法

Improved RC4 Cipher Method and Its Performance Analysis

LI Qin, ZENG Fan-ping
(Department of Computer, University of Science and Technology of China, Hefei 230027)

【Abstract】When used in application, RC4 cipher technology has some problems, such as weak keys and related key attacks, invariance weakness,, byte bias and so on. This paper presents an improved RC4 cipher technology, which combines the ECC cipher and the RC4 stream cipher. The emphasis is analyzing the efficiency, the management of keys, and the capability of the improved RC4 cipher. The result indicates that the efficiency of the improved RC4 is just a litter lower than RC4 stream cipher while the security is much better than RC4. It has good repellency of the current attacks which aim at RC4 stream cipher. What is more, the improved RC4 cipher has a good way to solve the problems about key management and key update, which is very significant in application.

【Key words】RC4 technology; improved RC4; consultation of key; parallel FPK-like Hash algorithm

1 概述

RC4流密码技术是当前应用最为广泛的一种对称密码技术，以随机置换为基础，是一个可变密钥长度、面向字节操作的流密码。其算法简单、易于实现、加密解密速度快，被广泛用于安全套接字协议/传输层安全协议(SSL/TLS)标准，同时也应用于作为IEEE802.11无线局域网标准一部分的WEP协议。

RC4密码技术自公开以来，针对RC4的研究与分析就越来越多。研究人员在发现RC4诸多优点的同时，也分析出了RC4密码技术在使用中存在的一些问题，特别是在WEP协议的使用中存在很大的安全问题。众多研究表明，这些问题并不能说明RC4本身存在安全问题，而是对RC4的不当使用或者是缺乏有效的管理机制所引起的。这就迫切要求深入地研究如何使用RC4技术，将其优势发挥到最大，同时又不会引入新的安全问题。

2 RC4密码技术

RC4密码技术是由Ron Rivest于1987年提出的[1]。其以随机置换为基础，是一个面向字节操作的流密码。整个RC4算法包含2个算法：一个是密钥安排算法(Key Scheduling Algorithm, KSA)，另一个是伪随机序列生成算法(Pseudo-Random Generation Algorithm, PRGA)[2]。整个代码如下所示。

```
KSA(K)
初始化：
 For i=0,1,…,N-1
    S[i]=i;
 j=0;
打乱(不规则化)：
 For i=0,1,…,N-1
    j=j+S[i]+K[i mod l ] (mod N);
    Swap(S[i],S[j]);
PRGA(K)
初始化：
 i=0;
 j=0;
循环：
   i=i+1;
   j=j+S[i] (mod N);
   Swap(S[i],S[j]);
   输出 z=S[(S[i]+S[j]) (mod N)];
```

其中，$N=2^n$；n是字节的长度(一般取n=8)，l是密钥以字节为度量的长度，S是0~N-1的一个置换。KSA算法把密钥按照一定的算法与初始置换S{0,1,…,N-1}运算得到一个新的置换；PRGA就利用这个新的置换来产生一个伪随机序列。其每输出一个字节的结果仅需8~16条机器操作指令[3]。RC4算法正是由于其简单性，在应用中比较易于软件实现，加密解密的速度非常快。

当前针对RC4的密码分析技术主要利用了RC4的一些弱点，比如相关性弱点、输出流偏向性弱点、相似密钥弱点、不变性弱点、密钥重用性弱点等。文献[2-4]都对这些弱点问题进行了详细说明，由于篇幅限制这里就不再重复分析。

作者简介：李 琴(1982－)，女，硕士研究生，主研方向：信息安全；曾凡平，副教授

收稿日期：2007-10-25 **E-mail：**billzeng@ustc.edu.cn

3 改进的 RC4 密码技术

现代密码技术分为对称加密体制和公开密钥加密体制两类。对称加密体制中通信双方需要采用一种安全的方式来保证密钥的共享；公开密钥加密体制通信双方各有一对密钥，不需要密钥的交互。一般来说，公钥加密体制对数据的处理效率没有对称密码体制效率高，但密钥却更易于管理。因此，在工程应用中，可以考虑引入公开密码体制进行双方密钥的协商，然后利用协商的密钥采用对称密码体制进行数据的加密解密。这样就可以最大程度地发挥两类密码体制的优势。

ECC 作为一种典型的公开密钥加密体制，其应用非常广泛。特别是在处理短消息时，对带宽的要求非常低。处理 100 bit 的短消息时其带宽要求不到 RSA 的 1/3。因此，在对 RC4 进行改进时，引入 ECC 密码技术进行密钥协商，来解决 RC4 使用中的密钥管理问题。改进后的 RC4 密码技术数据处理过程如图 1 所示。

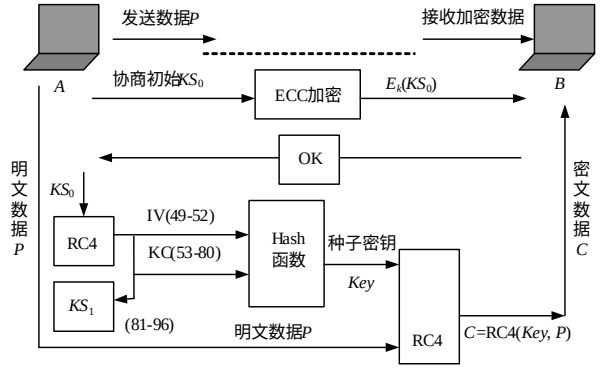

图 1 RC4 密码技术改进后的算法流程

改进后的 RC4 密码技术整个加密过程分为 3 个阶段：会话密钥的协商阶段，种子密钥的生成阶段，数据的处理阶段。3 个阶段的处理过程可用下面的步骤进行描述：

Step1 协商RC4 中将使用初始会话密钥KS_0。由*A*生成一个 16 Byte随机数KS_0，利用*B*的公钥对KS_0采用ECC加密技术进行加密。当*B*接收到这个加密后的信息后，用自己的私钥进行解密，就得到了双方此次通信的共享密钥。然后返回给*A*一个OK消息。至此初始密钥的协商阶段完成。

Step2 *A*利用KS_0通过RC4 算法得到一个伪随机输出流，在舍弃这个输出流的前 48 个字节之后，截取第 49 个字节到第 52 个字节作为初始向量***IV***，截取第 53 个字节到第 80 个字节作为用户共享密钥*KC*。并且截取第 81 个到第 96 个字节作为下一次的协商密钥KS_1。

Step3 *A* 将得到的 ***IV*** 和 *KC* 采用一定的 Hash 函数进行组合，得到种子密钥 *Key*。至此种子密钥的生成阶段完成。

Step4 用种子密钥 *Key* 通过 RC4 算法得到一个新的伪随机数据流，丢弃此数据流的前 48 个字节的数据，然后依次对 *P* 进行加密。

Step5 *A*把加密所得到的密文 *C*, *C*=RC4(*key*, *P*)发送给 *B*。*B* 收到数据以后按照相同的过程进行解密，就能得到消息数据 *P*。

整个过程中*B*在得到协商密钥以后也要采用同样的算法来产生种子密钥，用于得到密文*C*以后进行数据的解密。在一次连接通信中，后续的每一帧通信数据使用的种子密钥都是对Step2 生成的初始化向量***IV***加 1，然后再和*KC*通过Hash 函数计算得到。并且，*KC*的使用也有一定的时间T_0限制。当通信中*KC*的使用超过时限T_0后，就利用上一轮中在Step2 中生成的KS_1作为新一轮协商密钥，利用Step2 重新生成新的***IV***和*KC*。

4 改进的 RC4 密码技术的分析

在第 3 节中描述了 RC4 密码技术改进后的算法流程。下面将对这种改进后的 RC4 密码技术进行性能分析。主要从数据处理效率、密钥管理、安全性能 3 个角度进行分析。

(1)数据处理效率

在 WEP 协议中引入的 RC4 密码技术选用了初始化向量 ***IV***，其将 ***IV*** 直接与 *KC* 进行前后连接 ***IV***||*KC* 构成种子密钥；然后在之后的数据通信中也是将***IV***不断加 1 来保证密钥的不断变化。改进后的 RC4 密码技术沿用了 WEP 协议中对 RC4 种子密钥的处理过程，增加了密钥协商阶段，并且将 ***IV*** 和 *KC* 生成种子密钥的过程复杂化，引入了 Hash 函数处理过程。为了消除密钥之间的线性关系，引入类似 RSA 公司提出的 Fast Packet Keying 散列算法来对 ***IV*** 和 *KC* 进行处理。计算过程如下文代码段的前 6 行所示。

```
Key[31]=S[KC[27]⊕IV[1]]
Key[30]=S[KC[26]⊕IV[2]⊕Key[31]]
…
Key[i]=S[KC[(i-3)mod 28]⊕IV[4-(i mod 4)]⊕Key[i-1]]
…
Key[0]=S[KC[24]⊕IV[0]⊕Key[1]]
Key[7]=S[KC[6]⊕IV[3]]
Key[6]=S[KC[5]⊕IV[2]⊕key[7]]
Key[5]=S[KC[4]⊕IV[1]⊕key[6]]
……
Key[1]=S[KC[0]⊕IV[1]⊕key[2]]
Key[0]=S[KC[6]⊕IV[0]⊕key[1]]
```

其中，*KC* 长 28 Byte；***IV*** 长 4 Byte；种子密钥 *Key* 长 32 Byte。为了进一步提高 Hash 的速度，提出一种并行的类 FPK 散列算法。将 *KC* 分为 4 部分，每部分 7 Byte，分别与 ***IV*** 进行计算得到种子密钥 *Key* 的 4 个部分，*Key* 每部分 8 Byte，每一部分的计算过程如以上代码段第 7 行~第 12 行所示。原来的 Hash 过程变成 4 个并行的过程，可以使用硬件实现以加快处理的速度，这样就可以加快 RC4 的处理速度。

由于数据处理的核心部分依然是RC4 流密码的数据处理过程，只是每次对于***IV***和*KC*的处理多了Hash 的步骤，每T_0时间段多出一次RC4 的操作，每一次完整的通信多一次使用ECC进行密钥协商的阶段。因此，数据处理速度也只是有了很小的减慢。实验中，在加密 40 万个 1 500 Byte的数据包时，WEP协议中的RC4 需要的平均时间是 1 0291 ms，而改进后的 RC4 需要的平均时间是 10 766 ms。数据证明改进后的RC4 技术的数据处理速率非常接近于RC4。

(2)密钥管理

在WEP协议中引入的RC4 加密算法对密钥没有一个合理的管理方式，因此引起了诸多的安全问题[1]。其虽引入***IV***来变化种子密钥*Key*，但是由于将***IV***作为明文发送，以及将***IV***的计数加 1 与物理层绑定，造成了***IV***的大量重复和可截获，引起了许多问题。并且在WEP中对于用户的*KC*也是长时间才更换一次，这给攻击者足够的时间来破解密钥。

而在这种改进后的RC4 密码技术中，引入ECC进行通信双方的密钥协商。这样避免了在大的网络中需要大量的两两共享密钥来保证对称加密技术的执行。其在后续阶段中不断地将***IV***加 1，并且对于长时间的通信而言，其在每个时间段T_0内，使用的*KC*也不同。整个加密解密过程中密钥更新非常

灵活，而且简单易实现。

(3)安全性分析

通过第一层的 RC4 算法来生成 ***IV*** 和 *KC*，这样就不再需要在网络中用明文传输 ***IV***，攻击者就无法简单地利用 ***IV*** 来选择一些弱密钥进行密码分析。同时也更好地避免了密钥的重复使用，避免了根据部分已知明文推知其他明文。并且对于 ***IV*** 的初始化不再是和网卡绑定，而是由第一层的 RC4 生成的随机流的一部分，这样就对于 ***IV*** 的管理也很安全。并且由于在文献[3]中提到不管是 ***IV***||*RC*，还是 *RC*||***IV*** 得到的密钥都可利用 ***IV*** 的线性变化来分析 *key*。而这里改进后的 RC4 中将 ***IV*** 与 *KC* 通过 Hash 计算得到 *Key*，这样 ***IV*** 的线性变化所引起 *Key* 的变化就不再是线性的，也就避免了利用 ***IV*** 的线性变化来推知 *Key* 的变化。

相关性攻击对于不同长度的***IV***，有不同的分析长度。对于***IV***长度为 4 Byte时，其任意的一个***IV***可用于第一个字节的相关性分析的概率只有 4.50×10^{-5}，要分析这样的密钥中的一个字节的*KC*所需要的弱***IV***的数目为 1 330 000。这在对于***IV***长度自 3 个字节到 16 个字节进行的分析中，是概率最小，需要***IV***数目最多的情况[1]。这就是笔者在RC4 的改进方案中选择用 4 个字节的***IV***的原因。并且为了提高安全性，这里将采用 256 位的密钥。

这种改进方案中，每一层的RC4 算法生成的伪随机流在使用时，需要舍弃前 48 个字节。主要是考虑到，RC4 算法中的KSA部分，对于*S*盒元素的置换，其元素有 37%的概率只会发生一次置换。这就使得*S*盒中元素E(E<32)有 37%的概率只与*Key*中元素 0~E相关[4]。这就使得最后生成的伪随机流的最开始的一些字节有略低于 37%的概率会同由种子密钥预测的可能值相同。

在对 100 000 个 8 bit 的 RC4 伪随机流的分析中，第 1 个输出字节等于近似值的概率为 37%，第 2 个字节为 36.8%，之后依次有减少。到第 48 个字节时，其概率为 0.6%。因此，为了保证密码分析的难度足够，在改进后的 RC4 密码技术中都舍弃了伪随机流的前 48 个字节，这样就避免了利用最初几个字节的输出流的偏向性进行密钥的攻击。同样对于利用第 2 个字节的偏向性进行加速穷举的攻击也具有了抗攻击性。对于舍弃最初的 48 个字节是否是最佳的选择，还有待于进一步的实验分析。

在网络流量为 11 Mb/s 的网络中，传输 1 500 Byte 的数据包，在一次通信中单纯的 RC4 技术 5 h 左右就会出现不同包使用了相同的 ***IV*** 的情况：

$$11(\text{Mb/s})/(1\ 500\ \text{Byte/packet}\times 8\ \text{bit/Byte}) = 916.67\ \text{packet/s}$$

$$2^{24} = 1\ 677\ 216$$

$$1\ 677\ 216/916.67 = 5.084\ \text{h}$$

而在改进后的 RC4 密码技术中，即使是在 100 Mb/s 的网络中，传输 1 500 Byte 的数据包，出现不同包使用相同 ***IV*** 的时间间隔会是 5.96 天。即使是 1 000 Mb/s 网络中，也会有 14 个小时的间隔。

$$100\ (\text{Mb/s})/(1\ 500\ \text{Byte/packet}\times 8\ \text{bit/Byte}) = 8\ 333.36\ \text{packet/s}$$

$$2^{32} = 4\ 294\ 967\ 296$$

$$4\ 294\ 967\ 296/8\ 333.36 = 143.165\ \text{h} = 5.96\ 天$$

因此，可以设置T_0为 14 h，这样就避免了密钥的重用问题。如果网络流量很小，为了提高网络的效率，还可以将T_0的设置增大。至于T_0的设置对数据加密整体速度效率产生影响大小是多少，如何设置可以保证效率的更高效，还有待于进一步的试验分析。

5 结束语

改进后的RC4 密码技术引入ECC技术进行密钥的协商，在减少网络密钥管理负担的同时，保证了RC4 密钥的安全性。方案额外引入一层的RC4 算法来生成伪随机流，从中截取部分字节作为***IV***和*KC*，采用并行的类FPK散列算法来复杂化种子密钥的生成过程，从而避免了利用***IV***进行密钥分析的攻击。改进后的RC4 通过舍弃RC4 伪随机流的前 48 字节避免了针对RC4 伪随机流最初一些字节的偏向性的一些攻击。这些都使得改进后的RC4 在传统的针对RC4 的攻击手段具有一定的抵抗能力。改进后的RC4 密码技术从工程应用的角度进行设计，在保证高效的同时，也保证了密码技术的安全性，以及可用性和易于实现性，具有很强的工程应用意义。其关于T_0的最佳设置，以及对于舍弃的位数是否需要再增长，有待于进一步用实验数据分析，这将是后期的研究重点。

参考文献

[1] 耿　嘉, 曹秀英, 毕光国. 一种攻击 RC4-WEP 类密码的改进方法[J]. 通信学报, 2004, 25(1): 11-21.

[2] Mantin I, Shamir A. A Practical Attack on Broadcast RC4[C]//Proc. of the 8th International Workshop on Fast Software Encryption. [S. l.]: Springer-Verlag, 2002.

[3] Mantin I. Analysis of the Stream Cipher RC4[D]. Israel: Weizmann Institute of Science, 2001-11-27.

[4] Mousa A, Hamad A. Evaluation of the RC4 Algorithm for Data Encryption[J]. International Journal of Computer Science & Applications, 2006, 3(2): 44-56.

~~~~~~

(上接第 180 页)

参考文献

[1] Goresky M, Klapper A. 2-adic Shift Registers[C]//Proc. of Cambridge Security Workshop on Fast Software Encryption. Cambridge, England: [s. n.], 1994.

[2] Arnault F, Berger T P. Design and Properties of a New Pseudorandom Generator Based on a Filtered FCSR Automaton[J]. IEEE Trans. on Computers, 2005, 54(10): 1374-1383.

[3] Qi Wenfeng, Xu Hong. Partial Period Distribution of FCSR Sequences[J]. IEEE Trans. on Inform. Theory, 2003, 49(3): 761-765.

[4] Xu Hong, Qi Wenfeng. Autocorrelations of Maximum Period FCSR Sequences[J]. SIAM Journal on Discrete Mathematics, 2006, 20(3): 568-577.

[5] Goresky M, Klapper A. Fibonacci and Galois Representation of Feedback-with-carry Shift Registers[J]. IEEE Trans. on Inform. Theory, 2002, 48(10): 2826-2836.

[6] Klapper A, Goresky M. Feedback Shift Registers, 2-adic Span, and Combiners with Memory[J]. J. Cryptology, 1997, 10(1): 111-147.
~~~~~~