基于 Blowfish 和 MD5 的混合加密方案 *

尚华益[a]，姚国祥[a]，官全龙[b]
（暨南大学 a. 信息科学技术学院；b. 网络教育技术中心，广州 510632）

摘　要：针对 Blowfish 算法在实际应用中存在的等价密钥、重复初始化等问题，提出 Blowfish 与 MD5 算法相结合，将 MD5 算法的生成值作为密钥的一部分对 Blowfish 算法进行初始化，使用概率统计方法证明该方案弥补了 Blowfish 算法的实际应用缺陷。计算结果表明，混合加密方案具有更高的安全性、更快的速度。

关键词：Blowfish 算法；MD5 算法；等价密钥；重复初始化；概率统计
中图分类号：TP309　　**文献标志码**：A　　**文章编号**：1001-3695(2010)01-0231-03
doi:10.3969/j.issn.1001-3695.2010.01.068

Hybrid encryption scheme based on Blowfish and MD5

SHANG Hua-yi[a], YAO Guo-xiang[a], GUAN Quan-long[b]
(*a. College of Information Science & Technology, b. Network & Educational Technology Center, Jinan University, Guangzhou* 510632, *China*)

Abstract: In order to deal with the practical application problems of the Blowfish algorithm such as equivalent key, repeat initialization and so on, this paper proposed to combine with MD5 algorithm, and used MD5 output as a part of key to initialize Blowfish algorithm, used probability statistics to prove that the scheme could make up for the Blowfish algorithm drawbacks in practical application. The research shows that the hybrid encryption scheme has a higher security and faster speed than Blowfish.

Key words: Blowfish algorithm; MD5 algorithm; equivalent key; repeat initialization; probability statistics

随着网络技术的不断进步和全球信息化的飞速发展，信息安全问题已成为当前发展亟待解决的重要问题。当前，主要采用数据加密技术对计算机中的信息资源进行保护，保证即使信息失窃也不至泄密。数据加密技术的核心是加密算法，好的算法能显著增强系统的安全性。Blowfish 算法[1~3]以其出色的性能被广泛应用于众多的加密软件，目前如国外的 Shellfish、SplashID 以及国内的华夏科技宽带网络计费系统等核心加密部分都是使用 Blowfish 算法。但是单纯使用 Blowfish 算法在实际应用中存在一些不足，如等价密钥、重复初始化等，所以当前 Blowfish 算法也多与其他算法结合使用。文献[4]中为了实现安全加密，采用 AES-Blowfish-AES-Blowfish 的加密模式。文献[5]中采用 Blowfish、RSA、MD5 结合保障金融系统安全，将 Blowfish 算法与 CBC 模式相结合，加密前对明文数据块进行变换；使用 RSA 算法将 MD5 生成值用发送方的私钥加密后得到数字签名来确保消息的完整性和来源认证。文献[6]中将 Blowfish 与 MD5 结合，Blowfish 算法对多媒体内容进行加/解密，MD5 算法对修改后的用户密钥生成 hash 值。上述文献中 Blowfish 和 MD5 算法应用都是相互独立的，并未解决 Blowfish 算法存在的实际应用问题。因此，在对 Blowfish 算法研究的基础上，为了弥补单纯使用 Blowfish 算法存在的不足，本文将 Blowfish 与 MD5 算法结合，并提出将 MD5 算法的生成值作为 Blowfish 算法初始化密钥一部分的混合加密方案。

1　Blowfish 算法应用缺陷

1.1　Blowfish 算法简介

Blowfish 算法是由著名密码学家 Bruce Schneier 设计的基于 64 位分组及可变密钥长度的对称加密算法。Blowfish 算法核心在于子密钥生成，它将变长密钥扩展成总长 4 168 Byte 的子密钥数组。算法中使用了大量的子密钥，而子密钥又依赖于用户密钥[7]，实际加/解密过程中使用的是更新后的子密钥数组。子密钥即 P 数组和 S 盒：

a) 18 个 32 位的 P 数组：$P_1, P_2, \cdots, P_{18}$；

b) 4 个 32 位的 S 盒（具有 256 个入口）：

$S_{10}, S_{11}, S_{12}, S_{13}, \cdots, S_{1255}$

$S_{20}, S_{21}, S_{22}, S_{23}, \cdots, S_{2255}$

$S_{30}, S_{31}, S_{32}, S_{33}, \cdots, S_{3255}$

$S_{40}, S_{41}, S_{42}, S_{43}, \cdots, S_{4255}$

子密钥提前计算并存储在缓冲区，使用中不需要多次执行推导过程，提高了算法速度。由于子密钥 P 数组和 S 盒依赖于用户密钥，使用户密钥与子密钥数组关系复杂化，增强了破解难度，到目前为止其安全性仍未被破解[8~10]。

1.2　Blowfish 应用缺陷

Blowfish 算法采用变长密钥，给用户的使用带来很大便利；同时，由于算法的子密钥 P 数组和 S 盒依赖于用户密钥，算法加/解密核心在于密钥的选择和保密，但实际应用中经常使用一些弱密钥[11,12]对信息资源加密，致使实际应用中存在很大的安全隐患。

1）等价密钥　Blowfish 算法中用户密钥长度（最多 448 bit）小于 P 数组总长度（$32 \times 18 = 576$ bit），用户密钥与 P 数组异或过程中必定存在密钥循环，这种情况下就可能存在等价密钥。即使用不同的密钥 K_1、K_2 与 P 数组异或得 $P \oplus K_1$ 和 $P \oplus$

收稿日期：2009-03-27；**修回日期**：2009-04-29　**基金项目**：国家自然科学基金资助项目（60773083）；省部产学研资助项目（2008B090500201）
作者简介：尚华益（1986-），男，河南南阳人，硕士，主要研究方向为信息安全（pingwow@163.com）；姚国祥（1959-），男，安徽桐城人，教授，博导，主要研究方向为网络安全、信息系统；官全龙（1981-），男，广东韶关人，工程师，主要研究方向为网络技术、信息系统.

K_2,经过 Feistel 加密结构,得到更新后的子密钥数组 P_1 和 S_1、P_2 和 S_2。假设存在 $P \oplus K_1 = P \oplus K_2$,则更新后 P 数组和 S 盒对应部分必有 $P_1 \equiv P_2$,$S_1 \equiv S_2$,可见 K_1 和 K_2 初始化效果相同。本文将 K_1 和 K_2 定义为等价密钥。从图 1 可知判断等价密钥的关键在于用户密钥与 P 数组异或后结果是否惟一。

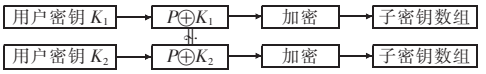

图 1　等价密钥判断

例如分别输入密钥 *a* 和 *aa* 进行算法初始化。将输入的用户密钥循环扩展后与 P 数组异或,异或后得到的 P 数组完全相同,导致更新后对应的 P 数组与 S 盒完全相同。这就存在一个问题:不同的用户密钥可得到相同子密钥数组。

2)重复初始化子密钥　Blowfish 算法不保存用户密钥,直接使用输入的用户密钥进行算法初始化,将更新后的子密钥数组与存储介质中的结果进行匹配来校验密钥。因此,每次输入用户密钥必定进行算法初始化,浪费了大量时间在重复初始化子密钥数组上,降低了算法效率。

3)密码截断　Blowfish 算法中子密钥数组依赖于用户密钥,用户密钥的安全性直接影响系统的安全性。选择密钥时,56 bit 和更低的密钥长度被认为是非常脆弱的。而算法中用户密钥最大长度为 56 bit。当用户密钥长度超过 56 bit 时,超出字符被截断,即仅使用 56 bit 与 P_{14} 异或后密钥进行循环,使超出部分密钥未被充分利用。

2　混合加密方案构建

2.1　混合加密方案设计

鉴于 Blowfish 算法在实际应用中存在等价密钥等问题,使攻击者可以构造等价密钥,获得相同的 P 数组和 S 盒对用户密钥进行猜解。为了消除等价密钥的影响,避免重复初始化,考虑与杂凑函数[13]结合来完善 Blowfish 加密算法。而在杂凑函数中 MD5 算法[14]与 SHA-1、RIPEMD-160 算法相比,具有迭代次数少、运算速度快的优点。因此,本文选择使用 MD5 单向杂凑函数。尽管在 2004 年举行的国际密码学会议上来自山东大学的王小云教授已经找出 MD5 算法存在碰撞,表明算法在理论上存在安全问题,但目前对应用领域来说仍然是足够安全的。混合加密方案结合了 Blowfish 和 MD5 算法的优点。在方案中使用 MD5 值校验的同时将 MD5 值作为 Blowfish 算法初始化密钥的一部分,能够解决 Blowfish 算法在实际应用中存在的问题。如图 2 所示,混合加密方案详细流程如下:

a)用∏的十六进制赋值 P、S 数组。

b)输入用户密钥 K,经 MD5 算法 hash 后得到 $H(K)$。如果是首次使用,将 $H(K)$ 写入注册表,跳转到步骤 d);否则继续下一步。

c)将注册表中存储的 hash 值读出到数组,与用户密钥 hash 后的生成值 $H(K)$ 进行校验,避免了 Blowfish 算法重复初始化。校验不相同跳转到步骤 h);相同则继续下一步。

d)将用户密钥 K 和 $H(K)$ 拼接(用户密钥 K 前置)作为密钥。由于 MD5 算法 hash 后生成值 $H(K)$ 惟一,确保了拼接密钥在密钥循环扩展后惟一,能够消除等价密钥影响,这是将 MD5 算法生成的 hash 值作为初始化密钥一部分的关键原因。然后,将拼接密钥 $K \| H(K)$ 循环移位与 P 数组异或,得 $P = P \oplus [K \| H(K)]$。基于拼接密钥的惟一性,即使使用等价密钥与 P 数组异或得到的结果也不可能完全相同。

e)用 Blowfish 算法加密全零明文分组,使用步骤 a)得到的 S 盒和步骤 d)生成的 P 数组作为密钥。

f)用步骤 e)的输出结果替换 P_1 和 P_2,用 Blowfish 算法加密步骤 e)的输出,使用替换过 P_1 和 P_2 后的 P 数组;输出结果替换 P_3 和 P_4,重复上述操作,经过 521 次迭代后,得到全部更新后的 P、S 数组。

g)使用更新后的 P、S 数组对资源进行加/解密操作。

h)操作完成后退出。

2.2　混合加密方案密钥管理

在介绍了混合加密方案设计的具体流程后,下面简要介绍方案中的密钥管理。在单一 Blowfish 算法中不保存用户密钥,而混合方案中要进行 MD5 校验,就需要考虑密钥存储管理问题。考虑到混合加密方案密钥量小,方案不采用数据库存储用户密钥,而是将用户密钥的 hash 值写入注册表。具体流程如图 3 所示。

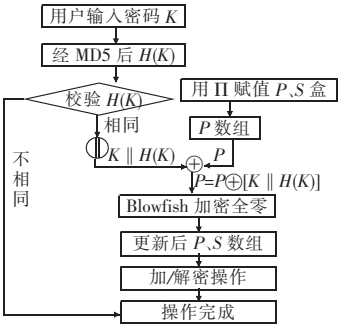

图 2　加密流程

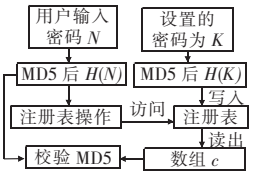

图 3　密钥存储管理

a)输入用户密钥 N,通过 MD5 算法进行 hash 得到 $H(N)$。

b)访问注册表,如果对应键值不存在,表明是首次设置密码,将 $H(N)$ 写入注册表中对应位置;否则,将注册表中对应键值转换后读出到数组 c 中。

c)通过将数组 c 中的值依次与 $H(N)$ 比较,判断是否完全相同,相同则进行下一步,不同则退出程序,检验完成释放数组空间。

d)密钥校验相同后,打开应用程序界面。

e)当丢失密码时,用户可以直接删除注册表中对应的键值,然后重置密码。

2.3　混合加密方案的优点

混合加密方案将 Blowfish 算法与 MD5 算法结合,消除了单纯使用 Blowfish 加密算法存在的应用问题。方案已经在 Shellfish 开源软件上实现,通过将 MD5 源码嵌入到 Shellfish 源码中,实现了上述设想,证明了方案的可行性。下面通过图 4 的流程对比详细阐述混合加密方案具有的优点。

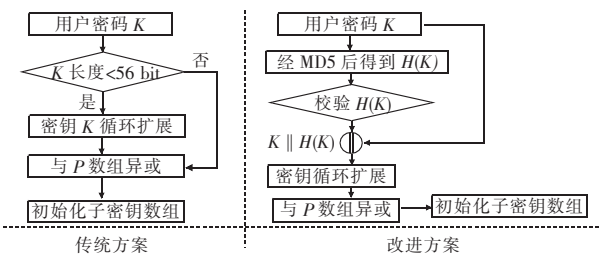

图 4　流程对比

a)完善密钥循环,消除等价密钥。通过图 4 可以看到,单纯使用 Blowfish 加密算法在密钥循环扩展时会受到等价密钥的影响,导致循环扩展后对应密钥完全相同,降低安全性。而

在混合加密方案中,将 K 和 $H(K)$ 拼接作为算法初始化的密钥,由于拼接后密钥惟一,即使对于等价密钥,更新后也不可能存在完全相同的子密钥 P 数组和 S 盒,从根本上消除了等价密钥对算法的影响。表 1 用概率统计方法说明改进前后使用口令穷举攻击所需的平均次数。

表 1　穷举攻击消耗平均次数

平均次数	密钥长度				
	1 bit	2 bit	10 bit	20 bit	30 bit
改进前	48	4513	$3\times10^{19}\sim5\times10^{9}$	$1.8\times10^{39}\sim5\times10^{19}$	$1.08\times10^{59}\sim5\times10^{29}$
基准	48	4560.5	3×10^{19}	1.8×10^{39}	1.08×10^{59}
改进后	48	4560.5	3×10^{19}	1.8×10^{39}	1.08×10^{59}

从表 1 可以看出,改进前由于存在等价密钥,攻击所耗平均次数小于基准次数;改进后消除了等价密钥影响,增强了破解难度(注:密钥空间为 95 个可打印显示的 ASCII 字符,基准是根据密钥空间密码长度不同组合得到的结果)。

b)避免重复初始化,解除密钥限制。单纯的 Blowfish 算法不保存用户密钥,适用于用户密码不经常改变的场合,而且输入错误的用户密钥也会进行初始化,导致无用且重复的初始化。而在混合加密方案中添加 MD5 算法来校验用户密钥,避免无用的初始化,对 Blowfish 算法加/解密速度影响不大;而且由于初始化耗时(大约加密 4 KB 文本的时间)远大于读写注册表和 MD5 函数 hash 时间总和,使混合加密方案速度更快。同时,取消了用户密钥长度和密钥更改频度的限制,方便用户操作。

2.4　安全性分析

Blowfish 加密算法以其可靠的安全性被应用到众多加密软件中。混合加密方案是在 Blowfish 算法的基础上,为了改进 Blowfish 算法应用缺陷而提出的。方案中引入 MD5 算法解决了等价密钥、重复初始化等问题,添加了算法密钥管理,提高了算法效率。下面就混合加密方案中可能存在的安全问题进行假设分析。

1)算法安全性　混合加密方案采用 Blowfish 与 MD5 算法相结合,将 MD5 算法嵌入到 Blowfish 加密算法源码中。在文献[15]中证明从 Blowfish 算法更新后得到的子密钥数组可逆向推导出密钥数组,影响实际应用。而在混合加密方案流程中,攻击一般发生在图 5 标示①的位置,当攻击发生时,攻击者可以从更新后的子密钥 P 数组推导出初始化密钥数组即 $K\parallel H(K)$;鉴于单项杂凑函数逆向推导在计算上是不可行的,且用户密钥长度未知,所以即使攻击者截获更新后的子密钥数组,仍然不能得到用户密钥 K。尽管攻击者可以得到更新后的子密钥数组,但并不表示混合加密方案中算法被攻破。

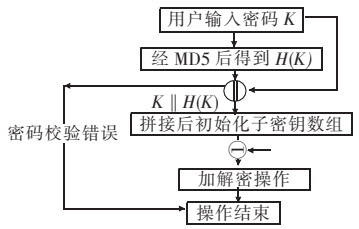

图 5　攻击位置

2)密钥安全性　方案将用户密钥经 MD5 算法后生成的 hash 值写入注册表,注册表中存储的是 hash 后的 32 bit 十六进制数。即使攻击者截获存储的 hash 值,用其他密钥重构得到完全相同的 hash 值,但接下来密钥拼接后初始化子密钥数组时会出错。对于常见的 hash 值,攻击者用较少的穷举次数就可以破解用户密钥,导致算法失效。推荐用户使用尽可能复杂的密码确保安全。另外,将用户密钥的 hash 值写入注册表,使密钥存储管理更加方便快捷,即使丢失或泄露用户密钥,也可以通过删除注册表对应键值快速重置密码,降低安全影响。

3)缓冲存储安全性　方案中使用到数组,数组仅当从注册表读出 hash 值时使用,校验完成后数组空间被释放,防止数组泄密。Blowfish 算法将更新后的子密钥数组存储在只读存储器等存储介质中。如果攻击者可以访问这些存储介质,就能够得到更新后的子密钥数组,据此仅能推出初始化密钥数组,不能推导出用户密钥。

3　结束语

混合加密方案将 Blowfish 加密算法与 MD5 算法结合,解决了单纯使用 Blowfish 算法存在的应用问题,增强了算法抗攻击能力,在安全性方面得到了较大改进。在以往研究的基础上,方案将 MD5 算法的生成值作为密钥的一部分,使两种算法紧密结合。方案进一步设想对计算机中加密资源生成 hash 值,确保资源加密过程的安全性和完整性。

参考文献:

[1] SCHNEIER B. Description of a new variable-length key,64 bit block cipher (Blowfish)[C]//Proc of FastSoftware Encryption, the Cambridge Security Workshop. London:Springer-Verlag,1994:191-204.

[2] STALLINGS W. 密码编码学与网络安全:原理与实践[M]. 刘玉珍,王丽娜,傅建明,译. 北京:电子工业出版社,2006:132-136.

[3] SCHNEIER B. 应用密码学[M]. 吴世忠,译. 北京:机械工业出版社,2000.

[4] MANZANARES A I,SIERRA C J M,MARQUEZ J T. On the implementation of security policies with adaptative encryption [J]. Computer Communications,2006,29(2):2750-2758.

[5] 姚亮,陈克非,朱学正. 一种基于混合密码体制的网络数据安全方案[J]. 计算机工程,2003,29(2):174-176.

[6] 刘波. 基于开放标准的内容交易管理平台 CyberManager 设计与实现[D]. 北京:北京邮电大学,2006:10-13,25-27.

[7] HARRIS S, ADAMS C. Key-dependent S-box manipulations[C]// Proc of the Selected Areas in Cryptography. Berlin: Springer-Verlag, 1999:631-632.

[8] SCHNEIER B. The Blowfish encryption algorithm[EB/OL]. (1995-09). http://www. schneier. com/paper-blowfish-oneyear. html.

[9] BIRYUKOV A,WAGNER D. Slide attacks[C]//Proc of FastSoftware Encryption. Berlin:Springer-Verlag,1999:245-259.

[10] NAKAHARA J. A linear analysis of Blowfish and Khufu[C]//Proc of Information Security Practice and Experience. Berlin:Springer-Verlag, 2007:20-32.

[11] VAUDENAY S. On the weak keys of Blowfish[C]//Proc of the 3rd International Workshop on FastSoftware Encryption. London:Springer-Verlag,1996:27-32.

[12] KARA O,MANAP C. A new class of weak keys for Blowfish[C]// Proc of FastSoftware Encryption. Berlin: Springer-Verlag,2007:167-180.

[13] 杨波. 现代密码学[M]. 2 版. 北京:清华大学出版社,2007:167-181.

[14] DOBBERTIN H. Secure hashing in practice[J]. Information Security Technical Report,1999,4(4):53-62.

[15] 钟黔川,朱清新. Blowfish 密码系统分析[J]. 计算机应用,2007,27(12):2940-2941.