一种基于 Hash 函数和分组密码的消息认证码

徐 津[1),2)] 温巧燕[1)] 王大印[3)]

[1)](北京邮电大学网络与交换技术国家重点实验室 北京 100876)

[2)](北京电子科技学院 北京 100070)

[3)](中国科学院信息工程研究所 北京 100093)

摘 要 基于 Hash 函数的 HMAC 是一种应用最为广泛的消息认证码,但最近的研究指出 HMAC 不仅易受到相关密钥攻击,在多用户环境下也易受到攻击.为了避免这些问题,我们对 HMAC 进行了改进,基于 Hash 函数和分组密码设计了一种新的消息认证码 HBMAC.在分组密码是伪随机置换和 Hash 函数所使用的压缩函数是伪随机函数的基本假设下,使用共享随机函数模型证明了 HBMAC 的安全性.同时,还提出了 HBMAC 和 HMAC 的算法实现,并基于典型数据对两种算法的性质和效率进行了分析.结果表明,与 HMAC 相比,HBMAC 在安全性和效率上取得了更好的折衷.

关键词 消息认证码;压缩函数;分组密码;伪随机置换;可证明安全;密码学

中图法分类号 TP309 **DOI 号** 10.3724/SP.J.1016.2015.00793

A New Message Authentication Code Based on Hash Function and Block Cipher

XU Jin[1),2)] WEN Qiao-Yan[1)] WANG Da-Yin[3)]

[1)](*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing* 100876)

[2)](*Beijing Electronic Science and Technology Institute, Beijing* 100070)

[3)](*Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100093)

Abstract HMAC on the base of Hash function is the most popular MAC, but recent studies have pointed out the vulnerabilities of HMAC both to related-key attack and in multi-user environment. To avoid these problems we have modified HMAC and built a new HBMAC based on Hash function and block cipher. Security proof for HBMAC has been given on the basic assumptions that block cipher is pseudo-random permutation and the compression function used by Hash function is a pseudo-random one, which uses the shared random model. Also, realization of HBMAC and HMAC algorithm is introduced. This paper analyses the quality and efficiency of these two algorithms based on some typical data. Compared with HMAC, HBMAC achieves a satisfying tradeoff between safety and efficiency.

Keywords message authentication code; compression function; block cipher; pseudo-random permutation; provable security; cryptography

1 引 言

网络技术的迅猛发展给人们带来方便之际,也带来巨大的危机,例如重要信息的泄漏、篡改等.事实上,在网络通讯中不仅存在无法避免的噪声,还存在着某些主动攻击者的恶意攻击.在现代密码学研究中,有一个基本的假设:对于开放的通讯网络,所

收稿日期:2013-08-29;最终修改稿收到日期:2014-10-13.本课题得到国家自然科学基金(61272057,61202434,61170270,61100203)、中央高校基本科研业务费专项资金(2012RC0612)资助.**徐 津**,女,1977 年生,博士研究生,主要研究方向为密码学、消息认证码和加密模式.E-mail: xujinhappy@163.com. **温巧燕**,女,1959 年生,博士,教授,主要研究领域为密码学、信息安全和量子计算. **王大印**,男,1977 年生,博士,主要研究方向为密码学、信息安全和消息认证码.

有的通讯都会受到攻击者的攻击.

2002 年美国公布的旨在确保国家信息安全的《联邦信息安全管理法案》中,定义“信息安全”是保护信息和信息系统,以避免未授权的访问、使用、泄漏、破坏、修改或者销毁,确保信息的完整性、保密性和可用性.其中,完整性是指防止不恰当的信息修改和破坏,也确保信息的不可否认性和可认证性;保密性是指对信息访问和公开的授权限制,包括对个人隐私和私有信息的保护;可用性是指对信息的及时和可靠的访问.

鉴于信息安全受到的威胁,需要密码学提供一种机制,来保证信息的私密性、完整性和不可否认性等.其中使消息的接收者可以验证该消息确实是来自所声称的主体,且在传输的过程中未曾受到未经授权的截取、重发、修改、伪造或插入消息,即为保证消息的完整性.

密码学中用来保证消息完整性的重要工具就是消息认证码(Message Authentication Codes,MAC).在网络通信和电子商务中,很多时候数据的完整性是至关重要的,因此消息认证码得到了快速的发展,现已广泛应用于各类 Internet 协议,如 IP Security (IPsec)①、Secure Sockets Layer(SSL)②/Transport Layer Security (TLS)③、Secure Shell (SSH)④、Simple Network Management Protocol(SNMP)⑤等.此外,很多标准化组织也开始了消息认证码的标准化工作.日本、英国和中国采用的 MAC 标准是国际标准 ISO/IEC 9797 中的 Cipher-Block Chaining Message Authentication Code(简记为 CBC-MAC)和 Hash-based Message Authentication Code(简记为 HMAC),韩国的标准等同于国际标准 RFC 4493.

MAC 的保护机制:首先在参与通讯的双方间通过秘密信道共享一个密钥 K,双方通过公开信道通讯时(这里使用 Alice 和 Bob 分别代表参与通讯的两方),Alice 如果要传送一个消息给 Bob,首先将这一消息使用 MAC 生成算法和共享密钥 K 计算出一个认证标记,称为 MAC 值,然后将此标记附加在这一消息之后一起传送给 Bob.接收后,Bob 使用 MAC 验证算法和共享密钥 K 计算接收到消息的认证标记,并和他所接收到的标记进行比较.如果两个标记相同,MAC 验证算法输出为 1,Bob 就认为消息在传送过程中没有被未经授权的篡改;如果两个标记不相同,MAC 验证算法输出为 0,Bob 就认为消息在传送过程中被篡改了.MAC 的保护机制过程如图 1 所示.

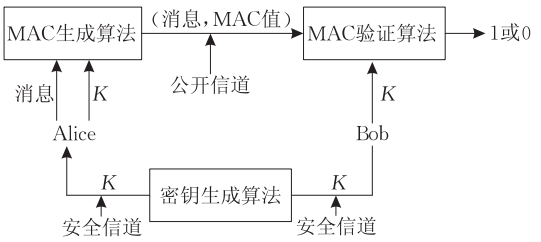

图 1 MAC 的保护机制

MAC 的攻击类型:根据攻击者的攻击对消息完整性安全造成的危害,针对 MAC 的攻击分为下面几种类型.

(1)存在性伪造(Existential Forgery).攻击者构造了一个消息,并伪造了该消息的认证标记.如果该消息和伪造的认证标记以几乎为 1 的概率通过验证,即使这个消息有可能没有意义,但是在存在性伪造下也称对消息认证码的攻击成功.

(2)选择性伪造(Selective Forgery).攻击者可以选取某个消息伪造其认证标记,并以几乎为 1 的概率通过验证.

(3)通用性伪造(Universal Forgery).攻击者可以选取任意消息伪造其认证标记,并以几乎为 1 的概率通过验证.

(4)密钥恢复(Key Recovery).攻击者能恢复密钥,这样一来攻击者和 Alice 一样,可以生成任意消息的合法的认证标记.类似于加密机制,这种攻击是最具破坏性的.

此外,区分攻击也是一类攻击方法,有专门针对使用前缀的消息认证码的区分攻击,如文献[1].还有将消息认证码和随机函数进行区分的 R-型区分攻击(Distinguishing-R Attack),以及将消息认证码内部的具体密码元件和随机函数进行区分的 H-型区分攻击(Distinguishing-H Attack).详细定义见文献[2].

根据攻击者的能力大小,针对 MAC 的攻击又可以分为下面几种类型.

(1)无消息攻击.攻击者无法获得任何消息及

① IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap [EB/OL]. http://www.rfc-editor.org/rfc/rfc6071.txt 2011,2

② The Secure Sockets Layer (SSL) Protocol Version 3.0 [EB/OL]. http://www.rfc-editor.org/rfc/rfc6101.txt 2011,8

③ The Transport Layer Security (TLS) Protocol Version 1.1 [EB/OL]. http://www.rfc-editor.org/rfc/rfc4346.txt 2006,4

④ The Secure Shell (SSH) Connection Protocol [EB/OL]. http://www.rfc-editor.org/rfc/rfc4254.txt 2006,1

⑤ A Simple Network Management Protocol(SNMP) [EB/OL]. http://www.rfc-editor.org/rfc/rfc1157.txt 1990,5

其对应的合法认证标记.

(2) 已知消息攻击(known message attack,简记为 kma). 攻击者能知道一些消息及其相应的合法认证标记.

(3)选择消息攻击(chosen message attack,简记为 cma). 攻击者能选取一些消息并得到其相应的合法认证标记.

其中,选择消息攻击是最强的,它赋予攻击者最大的能力,而存在性伪造则是攻击成功中相对最弱的,对消息的完整性构成的危害也最小. 本文在讨论消息认证码的安全性时,主要考虑其在选择消息攻击下的不可伪造性.

本文第 1 节是引言,介绍消息认证码的重要性和现实意义;第 2 节介绍本文中用到的基本符号;第 3 节给出消息认证码的安全模型、HMAC 的定义及安全性;第 4 节给出一种全新的消息认证码 HBMAC 的定义及其安全性证明;第 5 节对 HBMAC 进行优势分析;第 6 节为总结和展望.

2　基本记号

下面给出本文中用到的密码学基本符号和定义.

二进制串(string)是指一个由 0 和 1 组成的有限序列,也称为比特串.

一个特定对象的上标 n 则意味着该对象的长度为 n. 例如 0^d 意味着长度为 d 的 0 串,$\{0,1\}^n$ 则表示长度为 n 所有的二进制串.

块(block)是一个长度固定的二进制串,该长度称为块的长度.

$|X|$ 表示串 X 的比特位长度. 如果 X 是一个空串,则 $|X|=0$. 在对比特串的处理过程中,经常将其分成块,例如串 X 可以按 n 比特分成 $m=[|X|/n]$ 块,其中最后一块可能少于 n 比特,其余的块都是 n 比特. $\|M\|$ 表示所分成的块的个数.

$X\|Y$ 表示串 X 和 Y 的连接,例如 $x\|\sigma$ 表示串 x 和 σ 的连接.

$F(\cdot,\cdot)$ 表示函数 F 具有两个输入,有时使用如下写法:$F(k,x)=F_k(x)$.

攻击者(Adversary)又称敌手,是指攻击密码方案的算法,这个算法通常是随机的,简记为 A.

预言机(Oracle)是指攻击者 A 能访问的资源,例如加密算法、解密算法等. 攻击者 A 用到的 Oracle O 一般放在 A 的右上角,用 A^O 表示. $A^O=1$ 表示能访问 O 的攻击者 A 最后输出 1 的事件,$A^O=0$ 表示能访问 O 的攻击者 A 最后输出 0 的事件.

$O(\cdot)$ 表示 Oracle O 有一个输入,如果需要表示 A^O 输出的值,用 $x\leftarrow A^{O(\cdot)}$ 表示,其中 x 的类型同 A^O 输出的类型一致.

$Rand(a,b)$ 表示所有从集合 a 到集合 b 的函数. 如果 m 和 n 是整数,那么 $Rand(m,n)$ 表示所有从集合 $\{0,1\}^m$ 到 $\{0,1\}^n$ 的函数的集合.

$Perm(a)$ 表示所有在集合 a 上的置换的集合. $Perm(n)$ 表示 $\{0,1\}^n$ 上所有置换的集合. 这个集合也可以看成是所有的由集合 k 中元素唯一确定的置换的集合.

$x\xleftarrow{\$}B$ 表示从一个集合 B 中随机地选取一个元素. 例如符号 $\rho\xleftarrow{\$}Rand(a,b)$ 表示随机地从集合 $Rand(a,b)$ 中选取一个函数. $\pi\xleftarrow{\$}Perm(n)$ 表示随机地从集合 $Perm(n)$ 中选取一个置换.

$Pr[K\leftarrow\{0,1\}^n:A^{F(\cdot)}=1]$ 表示在密钥 K 随机选择的情况下,拥有 Oracle $F_K(\cdot)$ 的攻击者 A 返回 1 的概率.

分组密码 E_k 可以看成是这样一个函数:

$$E:k\times\{0,1\}^n\rightarrow\{0,1\}^n,$$

其中,k 是有限集并且每个 $E_k(\cdot)$ 是一个在集合 $\{0,1\}^n$ 上的置换,通常也可写作 $E(k,\cdot)$.

3　相关工作

3.1　可证明安全性

早期的密码算法是没有安全性证明的,其安全性都是基于安全性分析的,在理论上不能完全保证其安全性. 密码研究者们提出一个设想:定义一个主安全属性,算法只要满足这一属性就能完全保证它的安全性. 基于这一设想密码研究者们提出了可证明安全理论.

对应于 Shannon[3] 的完善安全性,1984 年,Goldwasser 和 Micali 在文献[4]中提出了语义安全(semantic security)的定义,首次从计算角度给出了加密方案的安全性定义,这篇文章标志着现代密码学可证明安全研究的开始. 这里的可证明安全性是指:在讨论整个密码方案的安全性的过程中,通过归约(reduction)的方法,将方案的安全性归结为底层密码学原语(primitive)的安全性,即证明如果原语是安全的,那么方案也是安全的,或者证明如果存在攻破方案安全性的算法,那么利用此算法,可以构造出攻破原语安全性的算法. 这里的原语是密码学中

最基本的研究单位，例如困难问题、分组密码、Hash函数等. 一般来讲，由公钥密码构造的算法归约到困难问题，由对称密码构造的算法则归约到分组密码，由Hash函数构造的算法归约到Hash函数. 一般来说，根据构造方法的不同，消息认证码的安全性可以归约到分组密码或Hash函数. 而Hash函数和分组密码的安全性由其伪随机性来定义. Bellare等人使用图灵测试的思想，定义了Advantage(优势)函数，简记为Adv函数，一般会配合上标、下标以及括弧来使用，其完整形式如$Adv_F^{\mathrm{prf}}(t,q,\mu)$，该式表示了函数$F$区分于prf(Pseudo-Random Functions)，也就是伪随机函数的概率. 括弧内的内容既可以是某个攻击者，也可以是攻击者所耗费的资源，如时间、查询次数等. 优势函数通常由一个攻击者发起的实验来定义，下面给出一个具体的例子.

攻击者A能访问理想的随机函数族g，以及函数族F，但访问均在黑盒状态下进行，攻击者A能且只能根据输入及输出来判断是随机函数g还是F. 使用数学语言可描述如下.

定义1. $g:\{0,1\}^l\rightarrow\{0,1\}^L$是一个随机函数族，$F:\{0,1\}^k\times\{0,1\}^l\rightarrow\{0,1\}^L$是一个函数族，攻击者$A$可以访问从两个函数族中随机选取的一个函数，通过询问最后作出判断，如下定义两个实验：

$Exp_{F,A}^{\mathrm{prf\text{-}0}}$

$g\xleftarrow{\$}Rand(D,R)$

$d\leftarrow A^g$

return d

$Exp_{F,A}^{\mathrm{prf\text{-}1}}$

$K\xleftarrow{\$}Keys(F)$

$d\leftarrow A^{F_K}$

return d

A区分F和随机函数g的prf-advantage被定义为

$$Adv_F^{\mathrm{prf}}(A)=Pr[Exp_{F,A}^{\mathrm{prf\text{-}1}}=1]-Pr[Exp_{F,A}^{\mathrm{prf\text{-}0}}=1].$$

对于t,q,μ，定义F的prf-advantage为

$$Adv_F^{\mathrm{prf}}(t,q,\mu)=\max_A\{Adv_{F,A}^{\mathrm{prf}}\},$$

其中，最大值是在所有具有时间复杂度为t，查询次数为q，查询的总位长为μ的攻击者A中取.

经过实际检验，分组密码E和随机置换区分的优势Adv_E^{prp}是可忽略的，也就是二者是不可区分的，所以说分组密码是一个伪随机置换(Pseudo-Random Permutations，PRP). 而Hash函数H和随机函数区分的优势Adv_H^{prf}也是可忽略的，同样也是不可区分的，所以说Hash函数是一个伪随机函数，后面将给出具体的定义.

1994年，文献[5-6]中Bellare、Kilian和Rogaway首次将CBC-MAC的安全性归约为分组密码的伪随机性，即证明了如果分组密码是伪随机置换，那么CBC-MAC是安全的消息认证码. 此后，出现了大量的论证消息认证码安全性的文章，这些文章都是证明如果存在攻破要论证的消息认证码的安全性的算法，那么利用此算法，可以构造出攻破分组密码安全性的算法. 例如基本加密模式的安全性[7]、OCB模式[8]的安全性、CWC模式[9]的安全性等. 现在几乎所有提出来的消息认证码都有安全性证明，是否是可证明安全的已经成为评价消息认证码最基本的指标之一.

3.2 MAC的安全模型

在Alice和Bob采用MAC算法通讯的过程中，攻击者可以自由地窃听Alice和Bob之间的通讯信道，并且可以自由地修改或创建信道中传送的消息. 如果攻击者能够成功地使Bob相信一个经过篡改的消息(该消息不曾被Alice传送过)来自于Alice，就称攻击者进行了成功的伪造.

下面在讨论消息认证码的安全性时，允许攻击者在选择消息攻击下进行攻击，即攻击者能够得到任何他想得到的消息和标记. 在这种情况下，如果攻击者不能进行成功的伪造，那么称消息认证码在选择消息攻击下是安全的.

消息认证码在选择消息攻击(cma)下的不可伪造(unforgeability，简记为uf)安全性可以用下面的实验定义：

$Exp_{\mathrm{MAC},A}^{\mathrm{uf\text{-}cma}}$

$k\xleftarrow{\$}K$

$(M,\sigma)\leftarrow A^{F_k(\cdot)}$

IF $F_k(M)=\sigma$ and M was not a query of A to its oracle

THEN return 1 else return 0

A是攻击者，可以访问oracleF_k. A选择一系列消息向oracleF_k发起询问，获得他想得到的消息认证标记. 这个阶段可以看成是A的学习阶段，然后A进行伪造，给出一对(M,σ). 如果$F_k(M)=\sigma$，并且M从未被查询过，则攻击者攻击成功，实验返回1；否则攻击失败，返回0.

一个好的消息认证码应该使得被伪造的概率是可忽略的，一般用符号$Adv_{\mathrm{MAC}}^{\mathrm{uf\text{-}cma}}$来表示这一概率，其中$Adv_{\mathrm{MAC}}^{\mathrm{uf\text{-}cma}}=Pr[Exp_{\mathrm{MAC},A}^{\mathrm{uf\text{-}cma}}\text{ return }1]$.

3.3 HMAC的定义

HMAC是效率很高、应用很广、性质很好、安全性很强的一种MAC，目前已被采纳为IP安全协议强制执行的认证算法.

HMAC是由Bellare等在文献[10]中提出的，

其要求所使用的 Hash 函数具有迭代结构(如 MD5[①]、SHA1[②]、SHA2[③] 等). 所谓迭代结构就是反复地使用压缩函数 f 将长消息映射为短消息. 这个压缩函数 f 具有两个输入：一个是长度为 l 的链变量 k，一个是长度为 b 的数据块 x，表示为 $f_k=f(k,x)$. 以 SHA1 为例，$b=512$，$l=160$.

举例来说，如果处理消息 $x=(x_1,x_2,\cdots,x_n)$，其中每个 x_i 均是长度为 b 的块，$i=1,\cdots,n$，n 是总块数，那么由压缩函数 f 构造的 Hash 函数的结构如图 2 所示.

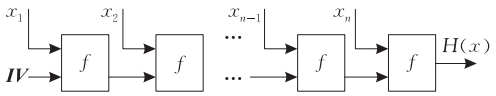

图 2　由压缩函数 f 构造的 Hash 函数

图 2 中 ***IV*** 为初始向量，是一个固定的值，用于处理第一块消息. 但实际上由于消息的长度是任意的，而压缩函数能处理的数据块长度只能是 b，这就必须对消息进行填充，使其成为 b 的整数倍. 填充算法必须是一个一一映射，以保证两个不同的消息在填充后仍然不相同. 一个典型的填充函数是填入 $|x|$ 的值，并填充一些额外的比特(比如说 0 或 1)使得所得到的比特串变成 b 的整数倍.

因为涉及到算法的实现，这里借鉴 Hash 函数 SHA1 的填充方法给出填充算法 $pad(x)$[11]，具体过程如下：

$pad(x)$
comment：$|x|\leq 2^{64}-1$
$d\leftarrow(447-|x|)\bmod 512$
$t\leftarrow|x|$ 的二进制表示
$M=x\|1\|0^d\|t$

如果让 $H(\cdot)$ 代表初始向量固定为 ***IV*** 且具有迭代结构的 Hash 函数，定义 $H^*(k,x)=H(k\|x)$，那么 HMAC(k,x) 的构造方法如下：

HMAC(k,x)
$t\leftarrow 0$，$\bar{k}\leftarrow\overline{pad}(k)$
$k_1=\bar{k}\oplus ipad$，$k_2=\bar{k}\oplus opad$
$t\leftarrow H^*(k_1,pad(x))$
$\sigma\leftarrow H^*(k_2,pad(t))$
return σ

其中，$\overline{pad}(k)$ 表示把函数的输入也就是密钥 k，通过补 0 的方式填充为长度为 b 的二进制串. 同时为了避免使用两个密钥，减少密钥开销，Bellare 等建议通过分别异或一个常值来生成 k_1 和 k_2，其中 $opad$ 和 $ipad$ 为两个 b 比特的常数，$ipad$ 表示二进制数 00110110 重复 $b/8$ 次，$opad$ 表示二进制数 01011010 重复 $b/8$ 次. 算法如图 3 所示.

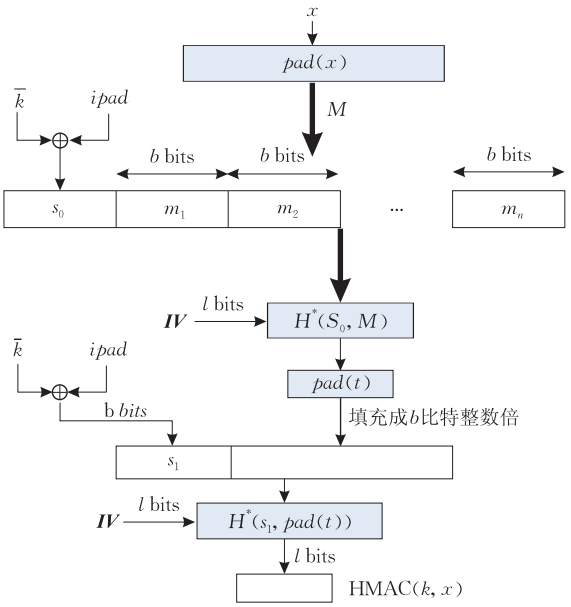

图 3　HMAC 消息认证码算法结构

这种设计有一个典型的优点就是算法不依赖于任何具体的 Hash 函数，底层的 Hash 函数就相当于黑盒，可以根据需要方便地选择任何的 Hash 算法，这一特性称之为黑盒.

3.4　HMAC 的安全性

在可证明安全性理论出现之前，密码研究者们用各种方式对 HMAC 进行攻击，如碰撞攻击、扩展攻击、分割征服攻击[10]等，但这些方法都没有好的效果，对 HMAC 最好的攻击仍然是生日攻击和穷尽密钥搜索攻击. 后来 Bellare 等人基于 HMAC 中两个密钥 k_1 和 k_2 相互独立的假设，在文献[12]中给出了 HMAC 的严格安全性证明，将其安全性同底层 Hash 函数的安全性关联起来. 这一分析考虑了对于 HMAC 任何形式的攻击，而不是有限的可能的攻击，很好地量化了 Hash 函数的安全性和基于该 Hash 的 HMAC 的安全性之间的紧密关系，实现了 HMAC 安全性的归约. 换句话说，如果发现 HMAC 安全性存在问题，那么底层的 Hash 函数必然存在问题. 因此，在理论上证明了 HMAC 的安全性，也验证了之前的攻击实践.

但随着研究的进展，出现了新的攻击方法即相关密钥攻击，而且它已经成为现代密码学安全性中

① The MD5 message-digest algorithm [EB/OL]. http://www.rfc-editor.org/rfc/rfc1321.txt 1992,4
② US secure hash algorithm 1 (SHA1) [EB/OL]. http://www.rfc-editor.org/rfc/rfc3174.txt 2001,9
③ US secure hash algorithm 2 (SHA2) [EB/OL]. http://www.rfc-editor.org/rfc/rfc6668.txt 2012,7

一个重要准则[13]. 因此,文献[14]注意到需调用两次 Hash 函数 H,但两次所使用的密钥间存在相互关联,不相互独立的关系. 而利用这一关联,即 $k_1 \oplus k_2 = ipad \oplus opad$,可以构造对 HMAC 相关密钥攻击. 鉴于这一特性,Neal Koblitz 等①提出在多用户环境下,HMAC 也易受到攻击.

针对这些情况,本文对 HMAC 进行了改进,将算法中用到的两个密钥减少到一个,提出了一种新的基于 Hash 函数和分组密码的消息认证码,从而消除了两个密钥之间存在的相关性,保证了在多用户环境下用户密钥选取的独立性,有效避免了上述文献中提到的问题.

4 HBMAC 的定义及安全性证明

设计新的消息认证码算法是一个困难的过程,首先算法需要是可证明安全的,并且证明过程还要尽可能的简洁易懂,其次要使得消息认证码的效率尽可能的高,性质尽可能的优良,这几点要求本来就是相互矛盾的.

HBMAC(Message Authentication Code Based on Hash Function and Block Cipher)算法设计充分考虑了安全性、效率和性质. 与 HMAC 相比,HBMAC 在安全性和效率上取得了更好的折衷.

4.1 HBMAC 的定义

HBMAC 是使用 Hash 函数和分组密码的组合来构造的,所以将其称为 HBMAC.

H 的定义同 HMAC 中的 H^* 定义. 假设 H^* 的输出等于分组密码的分组长度,二者都为 l,那么 HBMAC 的构造方法如下:

$$\begin{aligned}&\text{HBMAC}(k,x)\\&M=pad(x)\\&L=E_k(0)\\&\sigma=E_k(H^*(\overline{pad}(L),M))\\&\text{return } (x\,\|\,\sigma)\end{aligned}$$

其中,x 是任意长度的输入消息,M 表示对 x 填充后的消息,$pad(x)$为按照前面算法描述进行填充,k 是长为 l 的密钥,$\overline{pad}(L)$表示为对 L 通过补 0 的方式填充成长度为 b 的二进制串,E 表示分组密码. 算法如图 4 所示.

对于 HBMAC,在 Hash 函数的压缩函数是伪随机函数、分组密码的分组长度等于 Hash 函数输出的长度、分组密码是伪随机置换的基本假设下,本文证明了该消息认证码安全性的界为

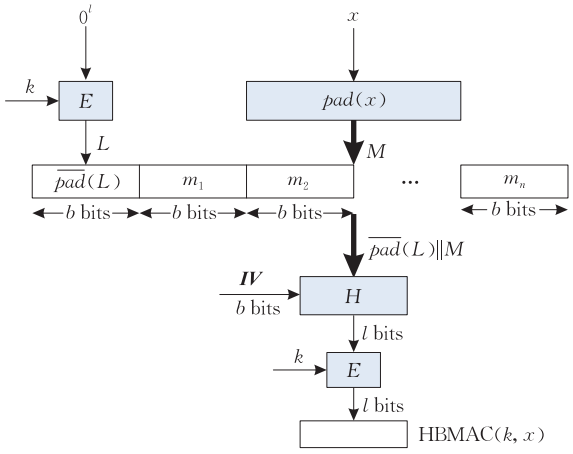

图 4 HBMAC 消息认证码算法结构

$$Adv^{\text{uf-cma}}_{\text{HBMAC}}(A_{\text{HBMAC}}) \le Adv^{\text{prp}}_{E,B_A}(t,q) + \frac{q(q+1)}{2^{l+1}} + \binom{q+2}{2} Adv^{\text{au}}_{H^*}(A_{H^*}) + \frac{1}{2^l}.$$

其中,A_{HBMAC}为针对 HBMAC 的攻击者,共进行了 q 次查询,B_A为针对分组密码的攻击者,进行了 $q+1$ 次查询,A_{H^*} 为针对 Hash 函数的攻击者,共进行了 2 次查询.

4.2 HBMAC 的安全性证明

首先给出可证明安全中的几个基本定义,然后证明 HBMAC 所基于的 Hash 函数是计算上几乎泛的,最后构造了两个实验,证明了 HBMAC 的安全性.

4.2.1 基本定义

在可证明安全理论中,使用优势函数来度量一个算法与一个理想的算法之间的差别. 如果这个差别是可忽略的,就认为该算法是安全的. 这里给出可证明安全中经常用到的优势函数的定义.

定义 2. 假定 A 是一个具有 Oracle 的攻击者,A^o表示 A 可以查询 Oracle O. 不失一般性,假定攻击者从来不查询该 Oracle 定义域之外的值,并且从来不重复查询已经查询过的值,那么在进行了一定数量的查询之后,攻击者 A 输出一个值,这个值要么是 0,要么是 1. 定义优势函数为

$$Adv^{\text{prp}}_E(A) = Pr[k \xleftarrow{\$} K : A^{E_k(\cdot)} = 1] - Pr[\pi \xleftarrow{\$} Perm(n) : A^{\pi(\cdot)} = 1].$$

该式表示在经过了一定数量的查询之后,当给定的 Oracle 为 $E_k(\cdot)$时攻击者 A 输出 1 的概率与当给定的 Oracle 为 $\pi(\cdot)$时攻击者 A 输出 1 的概率之差. 其中 k 随机地从 K 中选择,π 随机地从置换族

① Neal Koblitz, Alfred Menezes. Another look at HMAC. Cryptology ePrint Archive[EB/OL]. http://eprint.iacr.org/data 2012,7,4

$Perm(n)$ 中选择. 该式度量了分组密码 $E:K\times\{0,1\}^n\to\{0,1\}^n$ 和随机置换之间区分的概率. 在安全性证明中一般都默认算法中所使用的分组密码的 $Adv_E^{\mathrm{prp}}(A)$ 可忽略.

类似地,定义一个从 $\{0,1\}^n$ 到 $\{0,1\}^n$ 的函数族 $F:K\times\{0,1\}^n\to\{0,1\}^n$,其中 K 是有限集. 给定 $k\in K$,函数组也可记作 $F_k(\cdot)=F(k,\cdot)$. $Rand(n)$ 表示所有 $\{0,1\}^n\to\{0,1\}^n$ 的函数的集合. 定义:

$$Adv_F^{\mathrm{prf}}(A)=Pr[k\xleftarrow{\$}K:A^{F_k(\cdot)}=1]-Pr[\rho\xleftarrow{\$}Rand(n):A^{\rho(\cdot)}=1].$$

该式度量了函数 $F:K\times\{0,1\}^n\to\{0,1\}^n$ 和随机函数之间区分的概率. 以上这些定义由 Goldwasser 和 Bellare① 给出.

引理 1.　攻击者 A 区分一个 n 比特的伪随机置换 E 和一个随机函数的优势 $Adv_{E,A}^{\mathrm{prf}}$,具有下面的界:

$$Adv_{E,A}^{\mathrm{prf}}\le Adv_{E,A}^{\mathrm{prp}}+q(q-1)/2^{n+1},$$

其中,q 为攻击者 A 对 Oracle 的查询次数,该引理的证明可参见文献[5].

4.2.2　计算上几乎泛的 Hash 函数

泛 Hash 函数族是 1979 年由 Carter 和 Wegman 提出的,它广泛应用于计算机科学的各个领域,包括密码学、复杂性理论、编译器以及数据库.

文献[15]给出了泛 Hash 函数的概念,文献[16-17]在此基础上,给出了基于压缩函数的 Hash 函数 H^* 是计算上几乎泛的函数的定义.

在下面的讨论中,假定 Hash 函数的定义域和值域都是有限的二进制串的集合,并且值域要小于定义域.

定义 3.　固定定义域 D 和值域 R. 如果对于每个 $x,y\in D$,其中 $x\ne y$,

$$Pr_{h\in H}[h(x)=h(y)]=1/|R|,$$

那么称一个有限的 Hash 函数集合 $H=\{h:D\to R\}$ 是泛的.

定义 4.　设 $H^*:\{0,1\}^k\times D\to R$ 是一个函数族,A_{H^*} 是一个攻击 H^* 的攻击者,它返回定义域中的两个消息,它的 au 优势定义为

$$Adv_{H^*}^{\mathrm{au}}(A_{H^*})=Pr[H^*(k,M_1)=H^*(k,M_2)\wedge M_1\ne M_2:(M_1,M_2)\leftarrow A_{H^*};k\leftarrow K],$$

其中,$\wedge$ 表示逻辑"和". 若该优势函数在资源限定的情况下很小,则说该函数是计算上几乎泛的,记作 cAU. 在压缩函数是伪随机函数的假设下,文献[12]中给出了由压缩函数基于迭代结构构造的 Hash 的优势函数的界,指出这一优势是可忽略的,也就证明了由压缩函数构造的 Hash 函数在有密钥的情况下是计算上几乎泛的.

4.2.3　安全性证明

关于 HBMAC 的安全性有如下定理.

定理 1.　设 $q,t\ge1$ 是整数,A 是攻击 HBMAC 的攻击者,E 是长度为 l 比特的 PRP,l_p 表示查询的总的明文比特长度,那么 $Adv_{\mathrm{HBMAC}}^{\mathrm{uf\text{-}cma}}$ 是可忽略的,并有如下不等式成立:

$$Adv_{\mathrm{HBMAC}}^{\mathrm{uf\text{-}cma}}(t,q,l_P)\le Adv_{E,A}^{\mathrm{prp}}(t,q)+\frac{q(q+1)}{2^{l+1}}+\binom{q+2}{2}Adv_{H^*}^{\mathrm{au}}(A)+\frac{1}{2^l}.$$

证明.　本文在选择消息攻击下,使用标准的消息认证码安全模型来考虑 HBMAC 的安全性:攻击者 A 获得 HBMAC 的 Oracle,在进行了 q 次询问之后攻击者发起伪造攻击.

A 的攻击过程如下:

$Exp_{\mathrm{HBMAC},A}^{\mathrm{uf\text{-}cma}}$

$k\xleftarrow{\$}K$

$L=E_k(0)$

When A ask M_i, reply with $E_k(H^*(L,M_i))$

$(M,\sigma)\leftarrow A$

IF $E_k(H^*(L,M))=\sigma$ and M was not a query of A to its oracle

THEN return 1 else return 0

从消息认证码安全性的定义中可以知道 $Adv_{\mathrm{HBMAC},A}^{\mathrm{uf\text{-}cma}}$ 等于实验 $Exp_{\mathrm{HBMAC},A}^{\mathrm{uf\text{-}cma}}$ 返回 1 的概率.

假设 A 是攻击消息认证码 HBMAC 的攻击者,在实验 $Exp_{\mathrm{HBMAC},A}^{\mathrm{uf\text{-}cma}}$ 中共询问了 q 次,A 的运行时间最多是 t,设计一个攻击者 B_A,它是区分伪随机置换 $E:\{0,1\}^l\times\{0,1\}^l\to\{0,1\}^l$ 和随机函数 $Rand(l,l)$ 的一个算法.

B_A 拥有一个 Oracle,该 Oracle 为 $f:\{0,1\}^l\to\{0,1\}^l$. B_A 在运行的过程中调用 A,当 A 询问时,B_A 使用自己的 Oracle 模拟 A 的 Oracle 来回答,最后当 A 输出它的伪造时,B_A 验证,如果验证通过,则返回 1,此时 B_A 认定所使用 Oracle 是一个伪随机置换;否则就返回 0,此时 B_A 认定它自己的 Oracle 是随机函数.

假定在实验 $Exp_{\mathrm{HBMAC},A}^{\mathrm{uf\text{-}cma}}$ 中,A 最多询问 q 次,这也就意味着 A 进行了 $q-1$ 次查询,1 次验证. 下面是实验 B_A 的实现代码.

① Goldwasser S, Bellare M. Lecture Notes on Cryptograhpy [EB/OL]. http://www-cse.ucsd.edu/users/mihir 2008

Exp B_A^f

$L=f(0)$

For $i=1,\cdots,q-1$ do

When A asks its oracle some query M_i

generate a nonce non_i

and answer with $f(H^*(L,M_i))$

End for

A outputs (M,σ)

$\sigma' \leftarrow f(H^*(L,M))$

IF $\sigma=\sigma'$ and M was not a query of A to its oracle

THEN return 1 else return 0

在实验的最初阶段，B_A查询它自己的 Oracle 在 0 点的值，然后把返回值赋予 L，当 A 查询 M_1 时，算法 B_A使用它自己的 Oracle 计算 $f(H^*(L,M_i))$，并把 $f(H^*(L,M_i))$返回给 A. A 继续进行查询，直到 $q-1$ 次，然后停止，输出一个伪造对(M,σ). B_A验证这个伪造对，如果正确，就返回 1.

使用符号 D 表示攻击者 B_A^f返回 1 的事件，使用 B 表示事件 $f \xleftarrow{\$} Rand(l,l)$，在实验中 Oracle f 的输入互不相同的事件使用 Z 表示.

在证明之前首先注意到下面两个事实：

事实 1. $Adv^{\text{uf-cma}}_{\text{HBMAC},A}=Pr[D|f \xleftarrow{\$} E]$，根据定义该式显然成立.

事实 2. 对于任何事件 A，B 和 C，C^c表示事件 C 的补事件.

$$\begin{aligned}Pr[A|B]&=\frac{Pr[A\cap B]}{Pr[B]}\\&=\frac{Pr[A\cap B\cap C]}{Pr[B]}+\frac{Pr[A\cap B\cap C^c]}{Pr[B]}\\&=\frac{Pr[A\cap B\cap C]}{Pr[B\cap C]}\frac{Pr[B\cap C]}{Pr[B]}+\\&\quad\frac{Pr[A\cap B\cap C^c]}{Pr[B\cap C^c]}\frac{Pr[B\cap C^c]}{Pr[B]}\\&=Pr[A|B\cap C]Pr[C|B]+\\&\quad Pr[A|B\cap C^c]Pr[C^c|B].\end{aligned}$$

那么根据定义 2 有

$$\begin{aligned}Adv^{\text{prf}}_{E,B_A}&=Pr[B_A^f=1|f\xleftarrow{\$}E]-\\&\quad Pr[B_A^f=1|f\xleftarrow{\$}Rand(128,128)]\\&=Pr[D|f\xleftarrow{\$}E]-Pr[D|B]\\&=Adv^{\text{uf-cma}}_{\text{HBMAC},A}-Pr[D|B]\\&=Adv^{\text{uf-cma}}_{\text{HBMAC},A}-Pr[D|B\cap Z]Pr[Z|B]-\\&\quad Pr[D|B\cap Z^c]Pr[Z^c|B]\\&\geq Adv^{\text{uf-cma}}_{\text{HBMAC},A}-Pr[D|B\cap Z]-\\&\quad Pr[D|B\cap Z^c].\end{aligned}$$

下面分别求 $Pr[D|B\cap Z]$和 $Pr[D|B\cap Z^c]$.

首先考虑 $Pr[D|B\cap Z]$，当事件 B 和 Z 同时发生的时候，也就是说 $f\xleftarrow{\$}Rand(l,l)$并且 f 的输入互不相同. 在这种情况下，攻击者 A 每次查询之后得到的都是一个随机值. 因为 f 是一个随机函数，而且 M 没有被查询过，从而 $\sigma=f(H^*(L,M_i))$的概率最多是 2^{-l}，即 $Pr[D|B\cap Z]\leq 2^{-l}$.

下一步考虑 $Pr[D|B\cap Z^c]$. 在这种情况下，$f\xleftarrow{\$}Rand(l,l)$，该随机函数的输入有碰撞. 不考虑攻击者如何利用这个碰撞来攻击，只要发生碰撞就认为攻击者能够使用这次碰撞成功伪造消息.

下面计算这个碰撞的概率.

设 C_i表示直到第 i 个查询时才发生碰撞的概率. C_0表示第一次查询得到 L 时发生碰撞的事件，显然有 $Pr[C_0]=0$. 如果任意两个消息之间碰撞的概率都是 ε，那么 $Pr[C_i]=i\varepsilon$，这是因为有 i 个不同的值可能与之发生碰撞，从而

$$\begin{aligned}Pr[D|B\cap Z^c]&\leq\sum_0^{q+1}Pr[C_i]\leq\sum_0^{q+1}i\varepsilon\leq\varepsilon\sum_0^{q+1}i\\&=\varepsilon(q+1)(q+2)/2.\end{aligned}$$

由定义 4 知，$\varepsilon=Adv^{\text{au}}_{H^*}(A_{H^*})$是一个可忽略的值，因此有

$$Pr[D|B\cap Z^c]\leq\frac{(q+1)(q+2)}{2}Adv^{\text{au}}_{H^*}(A).$$

从而有

$$\begin{aligned}Adv^{\text{prf}}_{E,B_A}&\geq Adv^{\text{uf-cma}}_{\text{HBMAC},A}-Pr[D|B\cap Z]-Pr[D|B\cap Z^c]\\&\geq Adv^{\text{uf-cma}}_{\text{HBMAC},A}-\frac{1}{2^l}-\binom{q+2}{2}Adv^{\text{au}}_{H^*}(A).\end{aligned}$$

由于攻击者 B_A进行了 $q+1$ 次询问，移项并使用引理 1 得

$$\begin{aligned}Adv^{\text{uf-cma}}_{\text{HBMAC},A}&\leq Adv^{\text{prp}}_{E,B_A}+\frac{q(q+1)}{2^{l+1}}+\\&\quad\frac{1}{2^l}+\binom{q+2}{2}Adv^{\text{au}}_{H^*}(A).\end{aligned}$$

下面继续进行分析：

$$\begin{aligned}&Adv^{\text{uf-cma}}_{\text{HBMAC}}(t,q,l_P)=\max_A\{Adv^{\text{uf-cma}}_{\text{HBMAC},A}\}\\&\leq\max_A\left\{Adv^{\text{prp}}_{E,B_A}+\frac{q(q+1)}{2^{l+1}}+\binom{q+2}{2}Adv^{\text{au}}_{H^*}(A)+\frac{1}{2^l}\right\}\\&\leq\max_A\{Adv^{\text{prp}}_{E,B_A}\}+\frac{q(q+1)}{2^{l+1}}+\binom{q+2}{2}Adv^{\text{au}}_{H^*}(A)+\frac{1}{2^l}\\&\leq Adv^{\text{prp}}_{E,A}(t,q)+\frac{q(q+1)}{2^{l+1}}+\binom{q+2}{2}Adv^{\text{au}}_{H^*}(A)+\frac{1}{2^l}.\end{aligned}$$

上面的第 1 个等式是由定义 1 得来的，第 2 行的不等式则应用了上一步的结果，后面的推导就是简单地利用最大值的性质. 证毕.

5　HBMAC 的优势分析

相关密钥攻击[18-19]是 Biham 和 Knudsen 针对分组密码提出的，用于从区分攻击[20-21]到密码恢复攻击[22]的各种安全性分析，已经成为主流密码学的挑战之一. 对于如何避免这一问题，密码学工作者们做了大量的工作，但只有文献[23-25]给出了少数积极的结果.

Bellare 认识到构造相关密钥安全的伪随机函数和伪随机置换对研究理想密码很有意义，2010 年 Cash 在文献[26]中基于 DDH/DLIN 假设构造了一个相关密钥安全的伪随机置换，这是一个突破性的进展. 虽然该构造已经证明是可行的，但其效率低下，很难实际应用. 2013 年文献[13]虽然对其进行了改进，但要求底层 Hash 函数是基于分组密码构造的，和基于 Mekle-Damgård 结构的 Hash 函数相比，效率上仍然处于劣势. 显然，采用上述方法解决 HMAC 的问题并不理想. 此外，文献[14]也给出了一种解决这一问题的办法，即在消息前增加一个比特或一个字节，但文献[27]的研究则证明了这一改进在密钥较长时面临安全风险.

本文另辟蹊径，为解决 HMAC 易受相关密钥攻击和在多用户环境下的安全问题，同时在效率上取得优势，通过和分组密码相结合的方法，构造了一种新的基于 Hash 函数和分组密码的消息认证码 HBMAC.

表 1 对 HBMAC 和 HMAC 的性质进行了对比.

表 1　HBMAC 和 HMAC 的性质对比

性质	HMAC	HBMAC
需要的密钥个数	2	1
是否抵抗相关密钥攻击	否	是
是否具有黑盒特性	是	是
有无弱密钥	有[14]	无

HBMAC 算法仅使用一个密钥，底层 Hash 函数的密钥由分组密码来生成，保证了不同密钥之间相互的独立性，能够避免相关密钥攻击和多用户环境下的安全风险，这些风险均源自使用了两个密钥. 此外，还避免了文献[27]中提到的存在弱密钥的安全风险的问题. 同时，HBMAC 算法还具备黑盒特性，分组密码及底层的 Hash 函数均可以根据需要进行灵活选择.

在效率方面，HBMAC 较 HMAC 减少了一次消息填充和两次 Hash 迭代运算，增加了两次分组密码计算，而其他计算过程完全一致. 当处理较长消息时，这一改变对性能影响可以忽略. 在处理较短消息时，由于分组密码计算要慢于 Hash 函数，效率会略有下降，事实上在实际应用中可以通过分组密码预计算的方法降低这一影响. 在某些具体的运用环境中，如安全性要求较高的银行认证系统中，此类消息认证算法具有一定的优势.

在 HBMAC 具体实现上，底层 Hash 函数可选择 SHA2-256，其分组长度为 512 bits，输出长度为 256 bits，分组密码选择分组长度和密钥长度均为 256 bits 的 Rijndeal 算法. 在 HMAC 具体实现上，底层 Hash 函数可选择 SHA2-256.

表 2 和表 3 分别列出了在上述实现下，HBMAC 和 HMAC 针对长度分别为 128 bits、512 bits、1024 bits、1536 bits 以及 $n\times512+m$ bits(其中 m 为小于 448 的正整数)消息的计算量.

表 2　HMAC 的计算量

消息分组长度	PC	HC	EC
128 bits	2	4	0
512 bits	2	5	0
1024 bits	2	6	0
1536 bits	2	7	0
$n\times512+m$ bits	2	$n+4$	0

其中，1PC 表示一次对消息的填充计算，1HC 表示一次 Hash 迭代计算，1EC 表示一次分组密码计算.

表 3　HBMAC 的计算量

消息分组长度	PC	HC	EC
128 bits	1	2	2
512 bits	1	3	2
1024 bits	1	4	2
1536 bits	1	5	2
$n\times512+m$ bits	1	$n+2$	2

从表 2 及表 3 中可以看出，随着消息长度的增加，Hash 迭代计算呈线性增长，而消息填充计算和分组密码计算则为一个固定值.

HBMAC 的输出可根据需要选取，最长支持 256 bits 的认证标记，如需要较短认证标记，可直接对其输出进行截断操作. 举例来说，如需要 128 bits 的认证标记，可直接选取分组密码输出的前 128 bits，只是其安全性会随着认证标记的缩短而降低. 此外，在计算消息认证码之前，进行密钥编排和 $E_k(0)$ 的预计算，可有效提升 HBMAC 算法的计算效率.

6　总结和展望

为解决 HMAC 面临的相关密钥攻击的风险问

题,结合当前密码学研究进展,本文另辟蹊径,通过和分组密码相结合,构造了一种新的基于 Hash 函数和分组密码的消息认证码 HBMAC,该算法不仅有效地避免了相关密钥攻击,确保了安全性,通过分组密码加密模式的选择,还很好地保证了算法的效率.此外,本文巧妙地设计了两个区分实验,完全模拟了攻击者伪造 HBMAC 消息认证码的过程,在分组密码是伪随机置换和 Hash 函数所使用的底层压缩函数是伪随机函数的基本假设下,证明了 HBMAC 的安全性.

在消息认证码的研究中,基于 Hash 的消息认证码一直是研究热点,一些人热衷于研究新的攻击方法,一些人热衷于不断完善和提升其安全性,如本文和文献[28],基本理念恰是一攻一防,相互促进,相得益彰.随着研究的不断深入,尤其是新的分析工具的出现和理论的实际运用,这一热点还将长期存在下去.

此外,随着云计算的发展,用户大量的数据将存储在云端,云端数据的完整性保护也是一个巨大的课题.由于云环境的特殊性,数据的修改将在云端由托管方进行修改,用户如何确保托管方在完整的数据上实施了合法授权的修改,这是目前需要研究的问题.Gennaro① 和 Catalano 等人[29]已经在这方面做了一些探讨性工作,提出了同态消息认证码的概念,但离实际使用还有一段距离,需要研究者们不断去改进和完善.

致 谢 感谢参与审稿的专家和编辑老师,他们诚恳地提出了修改意见,这些意见都十分宝贵,对我以后的学习和科研有很好的帮助!

参 考 文 献

[1] Wang Xiao-Yun, Wang Lei, Jia Ke-Ting, Wang Mei-Qin. New distinguishing attack on MAC using secret-prefix method //Proceedings of Fast Software Encryption 2009. Leuven, Belgium, 2009: 363-374

[2] Leurent G, Peyrin T, Wang Lei. New generic attacks against hash-based MACs//Proceedings of Advances in Cryptology-ASIACRYPT 2013, Part II. Bengaluru, India, 2013: 1-20

[3] Shannon C. Communication theory of secrecy systems. Bell System Technical Journal, 1949, 28(4): 656-715

[4] Goldwasser S, Micali S. Probabilistic encryption. Journal of Computer and System Sciences, 1984, 28(2): 270-299

[5] Bellare M, Kiliany J, Rogaway P. The security of the cipher block chaining message authentication code. Journal of Computer and System Sciences, 2000, 61(3): 362-399

[6] Bellare M, Kilian J, Rogaway P. The security of cipher block chaining//Proceedings of Advances in Cryptology CRYPTO 1994. Santa Barbara, USA, 1994: 341-358

[7] Bellare M, Desai A, Jokipii E, Rogaway P. A concrete security treatment of symmetric encryption: Analysis of the DES modes of operation//Proceedings of the 38th Symposium on Foundations of Computer Science. Miami Beach, USA, 1997: 394-403

[8] Rogaway P, Bellare M, Black J, Krovetz T. OCB: A block-cipher mode of operation for efficient authenticated encryption //Proceedings of the 8th ACM Conference on Computer and Communications Security. Philadelphia, USA, 2001: 196-205

[9] Kohno T, Viega J, Whiting D. CWC: A high-performance conventional authenticated encryption mode//Proceedings of Fast Software Encryption 2004. Delhi, India, 2004: 408-426

[10] Bellare M, Canetti R, Krawczyk H. Keying hash functions for message authentication//Proceedings of Advances in Cryptology-CRYPTO 1996. Santa Barbara, USA, 1996: 1-15

[11] Stinson D R. Cryptography Theory and Practice. 2nd Edition. London, UK: Chapman & Hall/CRC, 2003

[12] Bellare M. New proofs for NMAC and HMAC: Security without collision-resistance//Proceedings of Advances in Cryptology-CRYPTO 2006. Santa Barbara, USA, 2006: 602-619

[13] Bhattacharyya R, Roy A. Secure message authentication against related-key attack//Proceedings of Fast Software Encryption 2013. Singapore, Singapore, 2013: 305-324

[14] Peyrin T, Sasaki Y, Wang Lei. Generic related-key attacks for HMAC//Proceedings of Advances in Cryptology-ASIACRYPT 2012. Beijing, China, 2012: 580-597

[15] Carter L, Wegman M. Universal hash functions. Journal of Computer and System Sciences, 1979, 18(2): 143-154

[16] Stinson D. Universal hashing and authentication codes// Proceedings of Advances in Cryptology-CRYPTO 1991. Santa Barbara, USA, 1991: 74-85

[17] Krawczyk H. LFSR-based hashing and authentication// Proceedings of Advances in Cryptology-CRYPTO 1994. Santa Barbara, USA, 1994: 129-139

[18] Kundsen. L Cryptanalysis of LokI91//Proceedings of Advances in Cryptology-AUSCRYPT 1992. Gold Coast Queensland, Australia, 1992: 196-208

[19] Biham E. New types of cryptanalytic attacks using related keys. Journal of Cryptology, 1994, 7(4): 229-246

[20] Biryukov A, Dunkelman O, Keller N, et al. Key recovery attacks of practical complexity on AES-256 variants with up to 10 rounds//Proceedings of Advances in Cryptology-EUROCRYPT 2010. Riviera, French, 2010: 299-319

[21] Biryukov A, Khovratovich D, Nikolic I. Distinguisher and related-key attack on the full AES-256//Proceedings of Advances in Cryptology-CRYPTO 2009. Santa Barbara, USA, 2009: 231-249

① Fully Homomorphic Message Authenticators Cryptology ePrint Archive[EB/OL]. http://eprint.iacr.org/2012/290 2013,8,26

[22] Biham E, Dunkelman O, Keller N. A related-key rectangle attack on the full KASUMI//Proceedings of Advances in Cryptology-ASIACRYPT 2005. Chennai, India, 2005: 443-461

[23] Applebaum B, Harnik D, Ishai Y. Semantic security under related-key attacks and applications//Proceedings of Innovations in Computer Science. Beijing, China, 2011: 45-60

[24] Bellare M, Cash D, Miller R. Cryptography secure against related-key attacks and tampering//Proceedings of Advances in Cryptology-ASIACRYPT 2011. Seoul, Korea, 2011: 486-503

[25] Bellare M, Paterson K, Thomson S. RKA security beyond the linear barrier: IBE, encryption and signatures//Proceedings of Advances in Cryptology-ASIACRYPT 2012. Beijing, China, 2012: 331-348

[26] Bellare M, Cash D. Pseudorandom functions and permutations provably secure against related-key attacks//Proceedings of Advances in Cryptology-CRYPTO 2010. Santa Barbara, USA, 2010: 666-684

[27] Dodis Y, Ristenpart T, Steinberger J. To hash or not to hash again? (in) differentiability results for H2 and HMAC//Proceedings of Advances in Cryptology-CRYPTO 2012. Santa Barbara, USA, 2012: 348-366

[28] Dodis Y, Steinberger J. Domain extension for MACs beyond the birthday//Proceedings of Advances in Cryptology-EUROCRYPT 2011. Tallinn, Estonia, 2011: 323-342

[29] Catalano D, Fiore D. Practical homomorphic MACs for arithmetic circuits//Proceedings of Advances in Cryptology-EUROCRYPT 2013. Athens, Greece, 2013: 336-352

XU Jin, born in 1977, Ph. D. candidate. Her current research interests include cryptography, message authentication code and encryption mode.

WEN Qiao-Yan, born in 1959, Ph. D., professor, Ph. D. supervisor. Her research interests include cryptography, information security and quantum computing.

WANG Da-Yin, born in 1977, Ph. D. His research interests include cryptography, information security and message authentication code.

Background

Message Authentication Codes (MACs) are basic problems in cryptography, which could provide a way to detect whether a message has been tampered with during transmission. The usual model for authentication includes three participants: a transmitter, a receiver and an adversary. The transmitter sends a message over an insecure channel, where the adversary can introduce new messages as well as alter existing ones. Insertion of a new message by the adversary is called impersonation, and modification of an existing message by the adversary is called substitution. In both cases the adversary's goal is to deceive the receiver into believing that the new message is authentic. In many applications, it is of significant importance that the receiver can verify the data integrity of a message. In some cases this is even more important than encryption. The term MAC first appeared around 1980 in the ANSI X9.9 standard. From then on, HMAC, CBC-MAC XCBC-MAC and so on are proposed in sequence.

HMAC based on Hash function is the most popular MAC, which has been adopted as the standard of many security protocols such as IPSEC and SSL. But it has been pointed out that HMAC is vulnerable to related-key attack in recent research paper published in 2012. There are no methods to solve this problem. To avoid this problem, we have modified HMAC and built a new HBMAC from Hash function and block cipher in this paper. HBMAC algorithm uses only one key, using block cipher to generate the underlying Hash function key, and ensures the independence of each different key, thereby avoiding the risks from related-key attack and in multi-user environments. On the basic assumptions that block cipher is pseudo-random permutation and the compression function used by Hash function is a pseudo-random one, we have given a security proof for HBMAC.

This subject is partially supported by the National Natural Science Foundation of China (Grant Nos. 61272057, 61202434, 61170270, 61100203) and the Fundamental Research Funds for the Central Inveracities (Grant Nos. 2012RC0612, 2011YB01). The significance of the research project is to promote the development of cryptography in China. We have published more than 20 papers in the domestic and foreign various important academic journals and academic conferences in Message Authentication Codes field. This paper is helpful to the researchers who pay attention to the security of Message Authentication Codes in the related-key attack scenarios.

This paper could be helpful to researchers who specialize in the security of Message Authentication Codes in the related-key attack scenarios.