基于 MD5 与 RSA 算法的数字认证研究与实现

廖莎莎
(黄山学院 现代教育技术中心,安徽 黄山 245041)

摘 要:分析数字认证技术,结合基于 MD5 和 RSA 算法的安全认证体制,给出详细的实现方法与测试结果,为网络信息安全提供了一套实用的参考方案。

关键字:信息安全;MD5;RSA;数字认证
中图分类号:TP391 **文献标识码**:A **文章编号**:1672-447X(2010)04-0080-03

1 引 言

2000 年 6 月,美国国会在"全球和国家商务法(E-SIGN 法)"中批准了电子签名。这一立法避免了对仅仅基于电子形式的合同或者签名的争论,从而赋予电子签名以新的合法性。2005 年 4 月 1 日《电子签名法》在我国开始正式实施,这部法律规定,可靠的电子签名与手写签名或盖章具有同等的法律效力。[1]电子签名技术有很多种,但目前比较成熟以及普遍被使用的电子签名技术是基于 MD5 与 RSA 算法数字签名技术。

2 理论基础

2.1 MD5 算法

MD5 算法可以简要的描述为 MD5 以 512 位分组来处理输入的信息,且每一分组又被划分为 16 个 32 位子分组,经过了一系列的处理后,算法的输出由四个 32 位分组组成,将这 4 个 32 位分组串联后将生成一个 128 位散列值。[2]

MD5 算法步骤:

首先要计算分片信息,计算具体过程如下:

1.假设存在明文 *Context*,现在要对其进行处理,求取长度 $Len_{org}=Length(Context)$;

2. 求得填充字符串 *PadContext* 长度 $Len_{pad}=Length(PADcontext)$,其中 $(Len_{pad}+Len_{org})Mod512=448$,$PadContext=Context+String(0)+1$,即以 1 结尾,前面接着 N 给个 0。接下来,为最后 64*Bit* 填充上 *PacContext* 字符串的长度数值,这样就得到 $N+1$ 个 512 位新明文 *PaddedContext*,作为最终的明文参加运算;

3. 重复上面的 1,2 得到 $N+1$ 个 512 位的分组,计算公式为 $GroupContext[i]=(PaddedContext)Mod$;

4.对上面的每个小组进行进一步划分,得到 16 个 32 位的字符串存储到下面的数组中

$Arr[i]=(GroupContext[i])Mod16$,其中 $0<i<16$。

接着进行单向加密计算,在进行下一步运算之前,先定义 4 个链接变量、4 个非线性函数和 4 个主循环加密函数如下。

链接变量:

$A=0x01234567$,$B=0x89abcdef$,

$C=0xfedcba98$,$D=0x76543210$;

非线性函数:

$F(x,y,z)=(x\&y)|(\sim x)\&z$,$G(x,y,z)=(x\&z)|(y\&(\sim z))$,

$H(x,y,z)=x\hat{}y\hat{}z$,$I(x,y,z)=y\hat{}x|(\sim z))$,其中 & 表示二进

收稿日期:2010-07-03
作者简介:廖莎莎(1983-),安徽祁门人,黄山学院现代教育技术中心助理实验师,研究方向为信息安全。

制位与运算，|位或运算，~是位非运算，^ 位异或运算。

循环加密函数：

$FF(a,b,c,d,M[i],s,t_i)$表示 $a=b+((a+F(b,c,d))+M[i]+t_i\ll s$

$GG(a,b,c,d,M[i],s,t_i)$表示 $a=b+((a+G(b,c,d))+M[i]+t_i\ll s$

$HH(a,b,c,dM[i],s,t_i)$表示 $a=b+((a+H(b,c,d))+M[i]+t_i\ll s$

$II(a,b,c,d,M[i],s,t_i)$表示 $a=b+((a+I(b,c,d))+M[i]+t_i\ll s$，

$t_i=2^{32}*abs(Sin(i))$

5.进行循环计算

将连接变量赋值到定义的临时变量 a,b,c,d 中，每个函数运算得到一个新的 a,b,c 或者 d；第一轮

$FF(a,b,c,d,M[0],3,2^{32}*abs(Sin(1)))$，产生 a

$FF(d,a,b,c,M[1],7,2^{32}*abs(Sin(2)))$，产生 b

$FF(c,d,a,b,M[2],11,2^{32}*abs(Sin(3)))$，产生 c

$FF(b,c,d,a,M[3],19,2^{32}*abs(Sin(4)))$，产生 d

$FF(a,b,c,d,M[4],3,2^{32}*abs(Sin(5)))$，产生 a

$FF(d,a,b,c,M[5],7,2^{32}*abs(Sin(6)))$，产生 b

$FF(c,d,a,b,M[6],11,2^{32}*abs(Sin(7)))$，产生 c

$FF(b,c,d,a,M[7],19,2^{32}*abs(Sin(8)))$，产生 d

$FF(a,b,c,d,M[8],3,2^{32}*abs(Sin(9)))$，产生 a

$FF(d,a,b,c,M[9],7,2^{32}*abs(Sin(10)))$，产生 b

$FF(c,a,a,b,M[10],11,2^{32}*abs(Sin(11)))$，产生 c

$FF(b,c,d,a,M[11],19,2^{32}*abs(Sin(12)))$，产生 d

$FF(a,b,c,d,M[12],3,2^{32}*abs(Sin(13)))$，产生 a

$FF(d,a,b,c,M[13],7,2^{32}*abs(Sin(14)))$，产生 b

$FF(c,d,a,b,M[14],11,2^{32}*abs(Sin(15)))$，产生 c

$FF(b,c,d,a,M[15],19,2^{32}*abs(Sin(16)))$，产生 d

第二、三、四轮分别使用 $GG(a,b,c,d,M[i],s,t_i)$，$HH(a,b,c,d,M[i],s,t_i)$和 $II(a,b,c,d,M[i],s,t_i)$，其中 i 自增。

6.每执行完上述四轮一次后，执行下面运算

$A=A+a,B=B+b,C=C+c,D=D+d$；

7.对每个 *GroupContest*[*i*]重复 5,6 操作，直到$N+1$个 *GroupContest* [*i*] 都完成，最终输出 128 位字符串 *ABCD*作为数字指纹。

2.2　RSA 算法

尽管公钥加密算法的种类有很多，但 RSA 对目前为止已知的所有密码攻击几乎都具备抵抗力。RSA 算法的安全性基于分解大整数的困难性。计算两个大素数的乘积是容易的，但是把一个大合数分解成两个素数的积却是特别困难的。[9]利用目前已经掌握的知识和理论，RSA 算法分解 2048bit 的大整数已经超过了 64 位计算机的运算能力，因此在目前和预见的将来，它是足够安全的。[9]

RSA 算法步骤：

1.产生 2 个随机的互异的素数P和Q；

2.计算 $N=P*Q$；

3.随机产生一个的数 E，且 $1<E<M$，

其中 $M=(P-1)*(Q-1)$，E 与 M 互素；

4.根据 E 产生 F，其中 F 由公式$(E*F)\bmod(E)=1$ 计算得出；

到此，RSA 的 2 对密钥就产生了，产生密钥对(N,E)和(N,F)，使用者可以选取其中一个做公钥，一个做私钥，进行加密运算。

5.用私钥(N,E)对消息 *Context* 产生数字签名，用于身份验证，产生数字签名的公式如 $DS=(Context)^E Mod(N)$；

6.接收数字签名方可以根据发生签名方提供的公钥进行解密，还原原来的信息 *Context*，验证公式为 $Context'=(DS)^F Mod(N)$。

如果 $Context=Context'$，则表明数字签名是有效的，否则认为网络安全存在隐患，数字认证过程如图 1。

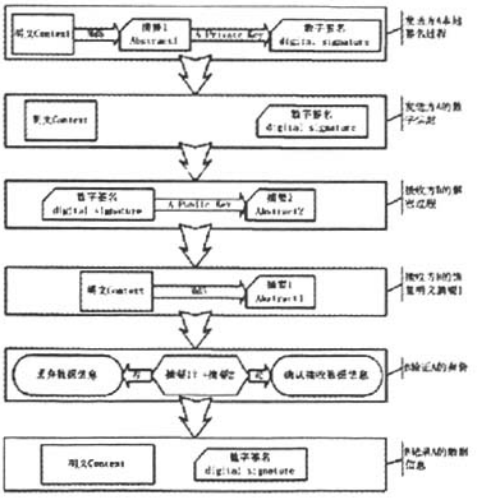


图 1　数字认证过程

3　算法的实现与测试

本文定义了 MD5 和 RSA 两个类对上述算法进行描述并实现。

RSA 算法类如下：

```
class RSA
{
public:
    int m_Len; //密钥长度
    CBigInt Pe,Qe,Ne,Fe,Ee; //定义 P,D,E 等参数
  CString m_F,m_E,m_N; //xml 字段属性
```

```
    CString m_CodeStr;
public:
    RSA(int); //生成密钥
    RSA(CString,int); //利用密钥对摘要进行加解密、
    void SetPublicKey(); //写公钥
    void SetPrivateKey(); //写私钥
    CString RSAEncrypt(); //RSA 加密
    CString RSADecryption(); //RSA 解密
    };
```

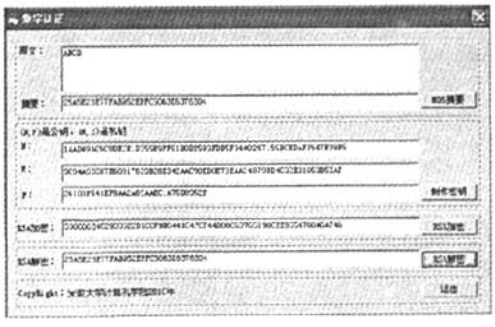
假设现私钥文件 PrivateKey.xml 内容如下：

```
<PrivateKey
N="1AAD49AC6C8DECE1D355E9FF81B0D2593FDB5F34
40247159BCEDAF1547E39B5"
E="9C94A9EC87E60817639B28E342AAC90ED0E7BEAA3
4879DD4C32E31053B5EAF"></PrivateKey>
```

公钥密钥文件 PublicKey.xml 内容如下：

```
<PublicKey
N="1AAD49AC6C8DECE1D355E9FF81B0D2593FDB5F34
40247159BCEDAF1547E39B5"
F = " 26103F541EF8AA2A69AAEC1A75D8952F " >
</PublicKey>
```

数字认证效果如图 2，在实际使用中，用户需要先点击制作密钥，点击后会在系统 C 盘根目录下生成 PrivateKey.xml 和 PublicKey.xml 文件。密钥和用户一一对应，每个用户拥有自己的密钥，除非用户申请更换，否则永远有效。PrivateKey.xml 发放给其用户，PublicKey.xml 由 CA 服务器持有。



2 CA 认证软件效果图

4 结 论

数字签名是通过一个单向函数对要传送的报文进行处理得到的用以认证报文来源并核实报文是否发生变化的一个字母数字串。数据加密是保护数据的最基本方法，它可以防止他人对传输的文件进行破坏。而数字签名则可以解决否认、伪造、篡改及冒充等问题，保护数据的完整性。对网络信息安全方面，本系统采用基于 RSA 和 MD5 的数字签名体制，充分地发挥了 SSH 体系结构的优势，为当前的网络信息安全提供了有力的技术保障。

参考文献：

[1]李景.基于数字签名技术的公文收发系统的设计与实现[D].北京：国防科学技术大学,2008:2.

[2]陆琳琳.MD5 算法的技术研究及性能优化[D].长春：吉林大学,2006:13.

[3]廖思周.MD5 算法防穷举(冲撞)破译的设计及其实现[J].五邑大学学报(自然科学版),2006,(04):33-37.

[4]游新娥.考试系统的身份认证研究与实现[D].长沙：湖南大学,2008:7-8.

[5]高静敏.几种常用的网络安全加密算法[J].黑龙江科技信息,2007,(15):92.

责任编辑：胡德明

Digital Certification Research and Implementation Based on MD5 and RSA Algorithm

Liao Shasha

(Modern Educational Technology Center, Huangshan University, Huangshan 245041, China)

Abstract: In the paper, the digital certification technology is discussed. Combined with secure authentication system based on MD5 and RSA algorithm, detailed implementation methods and testing results are given, thus a useful reference scheme for network information security is provided.

Key words: information security; MD5; RSA; digital certification

基于MD5与RSA算法的数字认证研究与实现

作者： 廖莎莎， Liao Shasha
作者单位： 黄山学院,现代教育技术中心,安徽,黄山,245041
刊名： 黄山学院学报
英文刊名： JOURNAL OF HUANGSHAN UNIVERSITY
年，卷(期)： 2010，12(5)
被引用次数： 0次

参考文献(5条)

1. 李景 基于数字签名技术的公文收发系统的设计与实现 2008

2. 陆琳琳 MD5算法的技术研究及性能优化 2006

3. 廖思周 MD5算法防穷举(冲撞)破译的设计及其实现 2006(4)

4. 游新娥 考试系统的身份认证研究与实现 2008

5. 高静敏 几种常用的网络安全加密算法 2007(15)

相似文献(10条)

1. 学位论文 陈少晖 Hash函数MD5攻击技术研究 2010

Hash函数是信息安全领域中一个非常重要的基本工具，它在数字签名、身份认证等领域中有着广泛的应用。MD5算法作为Hash函数大家庭中的一员，是MD结构的典型代表，广泛应用于信息安全领域。因此，通过对MD5算法的解析可以掌握Hash函数的基本分析方法，对其他Hash函数的分析有着重要的参考意义。
本论文对MD5算法的攻击技术进行了深入的分析和研究。首先系统总结了对该算法进行碰撞攻击的整体思想，运用了比特跟踪技术对差分路径进行分析和控制，利用消息修改技术提高搜索到产生碰撞的明文对的概率；其次通过介绍初始结构和过渡结构的构建，详细解析了对MD5进行原像攻击的技术和方法；最后对Hash函数进行了归纳总结和进一步的研究与探索。

2. 期刊论文 吴旭凡.胡晨.田渊.丁黄胜 HMAC-MD5算法的硬件实现 -电子器件2003,26(1)

信息安全体系中的消息验证是一个非常重要的方面.采用以散列函数为基础的消息验证编码是其中的一种重要方法.现提出了硬件实现一种以MD5算法为基础的消息验证编码(HMAC-MD5)的电路结构.该电路结构通过对MD5核心运算模块的复用,缩小了电路规模,达到了较高的处理速度.用VerilogHDL描述电路结构,并且在FPGA上验证了该结构的正确性.

3. 学位论文 梁杰 MD5-Hash函数的安全性分析 2007

Hash函数是信息安全与密码学领域中一个非常重要的基本工具。近几年关于Hash函数的碰撞攻击取得了举世注目的成果。我国学者王小云等提出的基于模减的差分分析方法有效地攻破了MD4, MD5, RIPMD,HAVAL和SHA-0等一系列Hash函数，其攻击算法复杂度都低于次压缩函数运算，在一般的个人电脑上运算都可以找到碰撞实例。本文正是针对MD5-Hash函数的碰撞攻击做进一步分析的最新研究成果。

基于王小云等在2005年欧密会上发表的关于MD5-Hash函数的2-block 的碰撞差分链，本文给出了一种快速攻击MD5-Hash函数的算法。研究发现，在王小云等发表的关于MD5-Hash函数碰撞攻击的文章里给出的保持MD5碰撞差分链的导出条件不是充分的即这些条件不足以保持碰撞差分链，并且一些导出条件可以放宽从而扩大碰撞集合；为了给出一个充分的条件集合，本文研究了二进制加法进位的特性，通过反馈计算导出了一些附加条件，从而第一次给出了关于MD5-Hash函数的碰撞差分链的充分条件集合。还特别证明了保持这条差分链的第一个block差分链成立的4个新添加的附加条件：a2,27 = 0, a2,29 = 0, a2,30 = 0 和 a2,31 = 0 是充分必要条件。在这基础上，总结了推导Hash函数的模减碰撞差分链的充分条件集合的一般方法。基于充分条件集合，本文还改进了MD5的碰撞攻击搜索算法，使得其计算复杂度比原来的算法降低了至少16倍；在Pentium4 1.70GHZ CPU 的个人电脑上实验，该攻击算法可以在5个小时内找到碰撞实例。

进一步的，本文从Dobbertin 95年欧密会时给出的关于MD5 Hash函数的半自由起始碰撞攻击的一个碰撞实例出发，按照模减差分分析方法推导出了一条1-block 的MD5模减碰撞差分链。虽然目前还不能够根据这条碰撞差分链构造出高效的攻击算法，但是它至少说明了模减差分分析方法在分析Hash函数方面是非常有效的。

本文最后讨论MD结构的致命缺陷，基于这一缺陷甚至可以构造出在实际应用中的Hash值碰撞实例，如X.509证书的Hash值碰撞；并分析最新提出的构造Hash函数的各种改进方案的优缺点。

4. 期刊论文 陈淳鑫.阎光 MD5算法在B/S结构下口令验证中的应用 -微型机与应用2005,24(2)

提出了一种新型的、可应用于B/S结构的口令验证机制.该机制应用了MD5算法,克服了传统的B/S结构在应用中的用户口令完全是明码传输的重大安全隐患.

5. 学位论文 田渊 嵌入式系统中信息安全的研究与加密算法的优化设计 2003

随着计算机技术和网络通信的发展，计算机系统信息安全的重要性日益突出。嵌入式系统是应用最广泛的一种计算机系统，特别是“个人数字助理”(PDA)被广泛应用于电子商务、移动办公等领域。所以，嵌入式系统的信息安全研究有重要的现实意义。

本文以通用计算机系统的信息安全含义为基础，结合嵌入式系统“面向应用”的特点，以PDA为目标系统，详细讨论了嵌入式系统中信息安全的含义、目标和实现技术。考虑到嵌入式系统在性能上的局限性，我们以密码机制和加解密算法为基础，使PDA从安全机制角度上初步实现了信息安全，能够基本满足保密性要求、身份验证要求、数据完整性要求、不可否认性要求和访问控制的要求。

论文的主体是加解密算法MD5和RSA在PDA中的实现。目标系统的安全机制是建立在密码机制和加解密算法基础上的，所以，我们首先需要在PDA中实现加解密算法。由于加解密算法的复杂性，使之需要软硬件协同工作，完全由软件或者完全由硬件实现都是不合适的。因此，在设计阶段就需要划分软硬件功能，软硬件协同设计。软硬件功能划分是设计的难点，不同的划分方案，会导致不同的设计结果，最后产生巨大的性能差异。

本论文所提出的“基于任务流图的划分方法”是一种相对简单的软硬件划分方法。它适合于加解密算法的特点——算法主体是一个多重循环运算，循环核是相对简单的算术运算或者逻辑运算。在论文中我们以代价函数值作为软硬件划分的依据，并且提出了代价函数的简化表达式。由于篇幅的局限性，在论文中我们只详细讨论了MD5和RSA两个算法用“基于任务流图的划分方法”的软硬件协同设计，以及它们以代价函数值为依据，在多种设计方案之间优化。

论文最后探讨了PDA中，在密码算法的基础上实现“数字签名”、“数字信封”等安全技术的思路，以及最终实现全部安全机制的可行性。

6. 学位论文 黎琳 Hash函数RIPEMD-128和HMAC-MD4的安全性分析 2007

目前，我们生活在信息社会里。随着信息和通信技术的不断发展，这些技术正日益走入越来越多人的生活。这些发展和应用给我们带来了实质性的利益，但是同时也带了诸多危害，它使我们的私人信息易于泄露，身份被他人盗用以及数据被他人篡改．为了避免这些危害，我们需要建立和发展值得信赖的信息体制。这些可信赖的体制及建立它们的基石是密码学研究的主要课题．

密码学[1]是研究数学技术及相关的保密性、数据完整性、实体认证、数据源认证等信息安全各领域的综合性的学科。它运用了数论、概率论、统计学、组合学等数学知识，并涉及信息论、计算复杂性、编码理论等学科知识．它不仅为信息安全提供服务和途径，而且还带来了一系列技术革新。

Hash函数在现代密码学中起着重要的作用。它可以将任意长度的输入压缩为固定长度的值，输出值我们称为Hash值。根据其定义和性质，Hash函数可用于保证数据完整性和实体认证，同时也是多种密码体制和协议的安全保障，例如数字签名、消息认证码等。Hash函数用于数据签名可带来诸多好处：可破坏数字签名方案的某种数学结构；可提高数字签名速度；无需泄露签名所对应的消息；可将签名变换和加密变换分开。

目前，标准的Hash函数分为两大类：MDx系列(MD4[11]，MD5[2]，HAVAL[12]，RIPEMD[13]，RIPEMD-128[6]，R[PEMD-160[6]等)和SHA系列(sHA-0[14]，SHA-1[3]，SHA-256，384，512[15]等)．这些Hash算法体现了Hash函数的设计技术。MD4算法是较早出现的Hash函数算法，它使用了基本的算术和布尔运算，其设计原则采用了迭代结构[30，43]的思想。在MD4算法公布后，许多}lash函数算法相继提出来，包括MD5、HAVAL、RIPEMD、RIPEMD-128、RIPEMD-160、SHA-0和SHA-1等，它们的大多数算法也是基于MD4算法的设计原则．RIPEMD算法是欧洲计划RIPE[13]的组成部分．RIPEMD-128算法是1996年由Hans Dobbertin，Antoon Bosselaers和Bart Preneel提出来的，用于替代RIPEMD算法，其输出Hash值为128比特．该算法由两个平行独立的操作部分组成，我们分别称为左路和右路，两部分都由四圈组成，每圈都包括一个圈函数，每个圈函数有不同的函数特性，用于以后的攻击中。两部分操作的输出结果进行运算用于生成Hash函数值．通常，我们讨论Hash函数的安全性，其包括三个特性，分别是抗原根性、抗第二原根性和抗碰撞性．所以对于不同特性在分析中存在多种不同的攻击方法。在近十几年里，对于Hash函数的分析取得了一些进展．生日攻击是一般的攻击方法，其名字来源于“生日问题”，可用于任何类型的Hash函数，其复杂性依赖于Hash值的长度。1996年，H.Dobbertin[16]给出了一个对MD4算法全算法攻击，该攻击以2-<’22>的概率找到一个碰撞。1998年，H.Dobbertin[17]证明了MD4算法的前两圈不是单向函数，这一结论意味着对于寻找原根和第二原根存在有效的攻击。对于RIPEMD算法，H.Dobbertin[18]可以以2<’31>的复杂性找到两圈RIPEMD的碰撞。

随着MDx系列Hash函数的发展，对于这些函数的安全性分析也不断增加。B．den Boer和A.Bosselaers[19]找到了针对MD5算法的一种伪碰撞，即同一明文在不同初始值下的Hash函数值相同．在1996年欧洲密码大会上，H.Dobbertin[20]公开了另一种形式的伪碰撞，即在两个不同的初始值下的不同消息的一个碰撞。在1998年国际密码大会上，F.Chalbaud和A．Joux[21]证明了用差分攻击方法可以以2-<’61>的概率找到SHA-0的一个碰撞。2003年亚密会上，B．V．Rompay[22]等人给出了以2-<’29>的概率针对HAVAL-128算法的碰撞攻击。

一些针对Hash函数的强力有效的攻击方法及结果出现在2004年。Eli Biham和Rafi[23]提出了对SHA-0算法的几乎碰撞。A.Joux[25]给出了一个由4个明文构成的SHA-0的碰撞．这些结果中，最为显著的是在2004年国际密码学大会上，王小云[7，8，9，10]等人宣布的对于一系列Hash函数的碰撞结果，包括MD4、MD5、H1AVAL-128和RIPEMD算法，其可以找到MD4和RIPEMD算法碰撞的复杂性分别低于2<’8>和2<’18>。同时，王小云提出了一套新的针对MDx系列的Hash函数的分析技术，给出了得到满足差分路线的充分条件的方法，以及如何使用明文修改技术来提高碰撞攻击的成功概率。随后，在2005年，王小云等使用此技术对MD5、SHA-0、SHA-1算法进行攻击，取得了较好的结果，并证明这些技术对于Hash函数分析是十分有效的。此攻击方法还可以用于MD4算法的第二原根攻击[26]，以及HMAC和NMAC的伪造攻击及部分密钥恢复攻击[28]．差分分析是由E.Biham和A．Shamir[34]于1990年提出的，它是针对对称密码体制(分组密码、流密码、Hash函数和MlAC算法)的选择明文(或选择密文)攻击最有效的方法之一．它的主要思想是通过分析特定明文差分对对结果密文差分的影响来获得可能性最大的密钥。王小云等基于差分分析的思想，采用不同于传统异或差分的模减差分(H.Dobbertin也采用过此类差分[16，20，18])，提出了一系列对标准Hash函数算法MD4，RLPEMD，HAVAL，MD5，SHAO，SHA1[7，8，9，10]的攻击。在王小云等人的方法中，经过对Hash函数算法的的明文差分通过每轮圈函数的整数模减差分和异或差分的分析，得到差分特征，以得到更多的信息来寻找碰撞．

在本文中，我们详细介绍了Hash函数，包括Hash函数的基本定义、性质、设计原则、应用以及针对Hash函数的攻击方法等基本知识。本文的主要工作分为两部分．一是给出了针对RIPEMD-128算法后三圈的碰撞攻击．另一个是对HMAC-MD4内部函数的安全性分析．这两部分中的攻击都采用王小云等提出的对于国际通用标准Hash函数的攻击方法．

在第四章中，我们定义整数模减差分为：对于两个输入X和X<’’>，△X=X<’’>-X。在此攻击中，我们采用的差分是带正负标记的整数模2<’32>的模差分和异或差分．其中，我们用带正负标记的比特位置来标记异或差分，对于X中比特值为0的比特位不带标记，X中比特为值1的定义为负标记。例如，对于差分-2<’23>，[24，25，26，27，28，-29]表示整数模减差分X<’’>-X=-2<’23>，其中比特进位从第24比特到第29比特。X中第24比特到第28比特值为0，第29比特值为1，而对于X’，第24比特到第28比特值为1，第29比特值为0。异或差分就是这些由比特进位带来的差分．

对算法的攻击过程包括以下三步：

第一步选择满足后三圈RIPEMD-128算法的明文差分由于RIPEMD-128算法是由两部分平行且独立的操作组成的，所以选择明文差分，使其同时满足两部分操作以产生差分路线是比较困难的。通过分析明文分组在两部分中每圈中位置，以及非线性圈函数的性质，并考虑明文修改技术的操作特点，我们选择两比特的明文差分如下：△H<,0>=0 △H=0，△M=M<’1>-M=(△mo，△m<,1>..，△m<,15>)，对应的输出差分为：△m<,1>=2<’31>;
△m<’14>=2<’23>，△m<,i>=0，0≤i≤15，i≠1，14．△<,α>=△cc+△ddd=0．△b=△dd+△aaa=0，△c=△aa+△bbb=(2<’31>+2<’31>)mod 2<’32>=0，△d=△bb+△ccc=0．我们可以分别得到RIPEMD-128算法后三圈两部分操作的差分路线。

根据算法中布尔函数的性质和差分特性，我们将得到满足差分特征的碰撞路线充分条件，这个过程在我们的攻击中也是至关重要的．它与碰撞路线是密切相关的，若充分条件中有矛盾出现，不能满足满分特征，则表明碰撞路线是错误的，不能产生碰撞。

第三步对于明文M，通过简单明文修改技术，可以减少第一圈的充分条件，再通过高级明文修改技术，可以减少更多条件，并保证差分路线正确，且条件不产生矛盾，可将充分条件分别降为19和36个，总计55个。

根据以上的攻击过程，我们可以以2-<’55>的碰撞概率，找到后三圈RIPEMD-128算法的碰撞，低于生日攻击的2-<’64>的碰撞概率。由于RIPEMD-128算法的由两条平行的操作组成，且其圈函数不同于其他标准Hash函数算法，所以找到该算法的碰撞路线及其满足给路线的充分条件都是比较困难的，本文中给出的攻击结果是首次对RIPEMD-128算法后三圈攻击的结果。

第五章中，我们的分析基于对MD4算法的攻击。首先，给出差分路线。选择两个明文：M=(m<,0>，m<,1>…m<,15>)则明文差分定义为：

在我们的攻击中，选择明文差分为：

表5.1给出了差分特征和差分路线．

其次，根据圈函数的性质，表5.2给出了满足差分特征的所有充分条件。我们可以以概率为2<’-54>找到一个几乎碰撞．

7.期刊论文 赵合胜 MD5加密算法在企业信息安全中的应用 -科技信息（学术版）2007(1)

企业信息安全是开发人员关注的重要问题,本文结合ASP.NET中的MD5加密算法,介绍了在ASP.NET开发的企业应用程序中,如何利用MD5来保证企业信息的安全.

8.学位论文 孙秋梅 对4圈杂凑函数HAVAL-160的一个攻击 2005

杂凑函数是信息安全中一个非常重要的工具，它对一个任意长度的消息m施加操作，返回一个固定长度的杂凑值h(m)，杂凑函数是公开的，对处理过程不用保密。单向杂凑函数的安全性取决于它的单向性，其输出不依赖于输入。杂凑函数是许多密码算法和协议的安全保证，它广泛用于签名、群签名、MAC码、电子钱币、比特承诺、电子选举等。

目前受到人们广泛关注和青睐的是标准杂凑函数，而标准杂凑函数又可分为两大家族：MDx家族(MD4、MD5、HAVAL、RIPE-MD、RIPE-MD-160)和SHA家族(SHA-0、SHA-1、SHA-256，384，512)。这些杂凑算法揭示了杂凑函数主要的设计技术。

目前，标准杂凑函数的分析技术已经取得了很大的进展。HansDobbertin于1996年对MD4给出了一个攻击，可以以2-22找到一个碰撞；1997年，Kasselman对MD4给出了一个更为有效的攻击。对于MD5，B.denBoer和A.Basselaersobber找到了MD5的一类伪碰撞——在两组不同的初始值下得到同一明文的相同的杂凑值；在1996年欧密会上，Dobbertin给出了MD5的一个碰撞——在另一初始值下得到两组不同的报文；2004年美密会上，王小云对MD5的攻击引起了国际密码界的轰动，王小云利用比特追踪法寻找碰撞路线、推出碰撞发生的必要条件、修改明文提高碰撞发生的概率可很容易找到一个碰撞。2003年，B.V.Rompay等对3圈杂凑函数HAVAL有一个攻击，其计算复杂度为229。而王小云于2004年利用破译MD5的技术同样找到了概率为2-7的3圈杂凑函数HAVAL-128的碰撞。至于SHA家族，2005年一月，王小云对SHA-1的研究又取得巨大进展，其计算复杂度少于269杂凑运算。

杂凑函数HAVAL是由Y.Zheng等在Auscrypto’92提出的，该体制可以在3、4或5圈压缩任意长度的报文并输出长度为128-比特、160-比特、192-比特或224-比特的杂凑值。

本文利用王小云教授提出的比特追踪方法和明文修改技术对四圈的HAVAL-160进行了分析，能够以不超过2-40的概率找到一对Haval碰撞，这个较生

日攻击的2-80的概率是一个很大的提高。

9.期刊论文 毛明.秦志光.陈少晖 破译MD5算法关键技术探索 -计算机应用2009,29(12)

针对Hash函数MD5算法的结构特点,从明文差分的引入、差分路径的控制和充分条件的确立等方面系统总结了该算法破译过程的关键技术及其主要步骤.首先介绍了破译过程中应用的三种差分的概念,分析了MD5算法中非线性函数的性质以及符号差分的扩展、循环左移的特点,然后从整体的分析思想和具体的实践方法两方面对破译MD5算法的关键技术进行了探索,以实例详细解析了消息修改技术,对Hash函数的破译进行了进一步的研究和探索.

10.期刊论文 秦剑波 MD5加密算法在校园网信息安全中的应用 -商场现代化2008(20)

本文介绍MD5加密算法的基本原理,举例说明该算法在用ASP和ASP.NET开发WEB应用程序时安全保护数据库中用户密码这一重要数据的方法.