基于改进混沌 Hash 函数的一次签名方案

李 晨,王世红
LI Chen,WANG Shi-hong

北京邮电大学 理学院,北京 100876
College of Sciences,Beijing University of Post and Telecommunication,Beijing 100876,China
E-mail:johnsun222@sohu.com

LI Chen,WANG Shi-hong.New one-time signature scheme based on improved chaos Hash function.Computer Engineering and Applications,2007,43(35):133-136.

Abstract: A Hash function based on chaos has been analyzed and some collisions are found.A new one-time digital signature scheme based on the improved chaos hash function is issued.The statistical properties and simple safety of the scheme are analyzed.

Key words: hash function;chaos;one-time digital signature

摘 要:分析了一种基于混沌构造的 Hash 函数方法,发现其中存在着碰撞。提出了一种基于改进 Hash 函数的一次数字签名方案,并对此方案的统计性和安全性进行了分析。

关键词:Hash 函数;混沌;一次签名
文章编号:1002-8331(2007)35-0133-04 **文献标识码**:A **中图分类号**:O415.5

随着信息技术的不断发展,电子商务日趋走进人们的生活,数字签名作为一种特殊的安全机制已成为电子商务安全性的重要组成部分。数字签名是一种以电子形式给一个消息签名的方法,实际上是手工签名的电子实现[1],是只有信息发送方才能进行的签名,是其他人无法伪造的字符串,这段特殊的字符串同时也是对签名真实性的一种证明。一次签名是数字签名中的一种,指对于一个密钥仅有一个消息能被签名,当然,该签名能被验证任意多次。使用单向函数来提交密钥的方式所构造的一次签名方案已经被许多人研究[2,3]。目前一次签名主要存在的问题是签名长度太长,同时私钥不能重复使用。

数字签名中常用到 Hash 函数。关于 Hash 函数的概念最早是在 1968 年由 Wilkes 提出,Hash 函数是密码学的一个基本工具,在密码学中有许多应用,特别是在数字签名和消息的完整性检测方面有重要的应用。Hash 函数的特点是正向计算简单、反向计算复杂,而且很难找到两个不同的输入对应于同一个输出值的一种函数。自 2004 年、2005 年王小云发现 MD5 和 SHA-1 的碰撞问题以来,关于 Hash 函数的研究成为一个热点。

混沌是一种复杂的动力学行为。混沌表现为确定性的非线性系统中的伪随机性,混沌信号是非周期的,并且对初始状态敏感。混沌的这些性质使利用混沌构造 Hash 函数成为一个新的研究方向[4,5]。

本文分析了目前提出的一种基于混沌映射的 Hash 函数构造方法[6],找到其中存在的碰撞,并提出了改进的方案。最后在改进的 Hash 函数基础之上,提出了一种一次数字签名方案。

1 一种混沌 Hash 函数的改进方案

Hash 函数有一个很重要的要求——无碰撞,即给定算法 h,要找两个不同的消息 $x_1 \neq x_2$,其杂凑值 $h(x_1)=h(x_2)$ 是困难的(计算不可行)。文献[6]提出了一种基于混沌的 Hash 函数构造方案。该方案相对于 MD5 构造较为简单,运算速度更快。但该方案存在着碰撞。

1.1 碰撞分析

首先简单介绍一下文献[6]提出的 Hash 函数构造方案。其原理如图 1 所示:

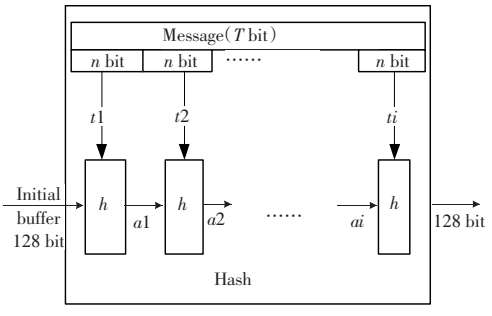

图 1 Hash 函数原理图

其中 h 函数由高维 Amold cat 映射来构造。即:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_m(k+1) \end{bmatrix} = A \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_m(k) \end{bmatrix} \bmod 2^L \quad (1)$$

基金项目:教育部重点项目资助(No.105034)。
作者简介:李晨(1983-),男,硕士,研究方向为混沌保密通信;王世红(1966-),女,博士,教授,硕导。

其中,k=1,2,…,i,$\boldsymbol{A}$ 是一个 m 维矩阵。

本文以 32 bit 位(即 L=32)运算为例介绍一下具体的实现步骤(按照图 1):

(1)设输入为的 128 bit 的初始值,将其分组为 4 组,每组 32 bit。分别对应方程(1)中的 $x_1(1)$、$x_2(1)$、$x_3(1)$、$x_4(1)$。

(2)将消息分为 i 块,得到 $t1$,$t2$,…,ti 每一块 n bit,n=(m−4)×32,首先将 $t1$ 分为 m−4 组,每组 32 bit,分别对应方程(1)中的 $x_5(1)$,$x_6(1)$,…,$x_m(1)$。

(3)得到 $x_1(1)$,$x_2(1)$,…,$x_m(1)$后,带入方程(1)进行一次 cap 映射。得到 $x_1(2)$,$x_2(2)$,…,$x_m(2)$。

(4)$x_1(2)$、$x_2(2)$、$x_3(2)$、$x_4(2)$保持不变,将 $t2$ 分为 m−4 组,每组 32 bit,分别代入 $x_5(2)$,$x_6(2)$,…,$x_m(2)$。再进行一次 cap 映射。

(5)依此类推,直到最后一次 cap 映射结束。最后取出 $x_1(i+1)$、$x_2(i+1)$、$x_3(i+1)$、$x_4(i+1)$作为输出值。

在该算法中用到了模运算。模运算作用是将一个大数转化在一定的取值范围之内。对于两个大数来说,如果低位不同,经过模运算后低位也不同。所以模运算对低位的变化是很敏感的。但是对于高位来说,特别是超过模范围的位,如果发生变换,经过模运算后有可能相同。举个简单的例子,129 对应的二进制数为 10000001,经过模 2^7 运算结果为 00000001,对应十进制数为 1。如果将高位取反,即 00000001,然后经过模 2^7 运算,对应十进制数也为 1。从中可以看出模运算对高比特位的变化是不敏感的。由于模运算的这个性质,导致文献[6]提出的 Hash 函数构造方案不可避免的产生碰撞。即,给定算法 h,对于两条消息 $x_1 \neq x_2$,如果两条消息使输入值的低比特位相同,只是高比特位不同,其杂凑值 $h(x_1)$、$h(x_2)$很可能相同。

$$A=\begin{bmatrix}1 & a_{12} & 0 & \cdots & 0\\ b_{12} & 1+a_{12}b_{12} & 0 & \cdots & 0\\ 0 & 0 & 1 & \cdots & 0\\ \vdots & \vdots & \vdots & & 0\\ 0 & 0 & 0 & \cdots & 1\end{bmatrix}\begin{bmatrix}1 & 0 & a_{13} & \cdots & 0\\ 0 & 1 & 0 & \cdots & 0\\ 0 & b_{13} & 1+a_{13}b_{13} & \cdots & 0\\ \vdots & \vdots & \vdots & & 0\\ 0 & 0 & 0 & \cdots & 1\end{bmatrix}\cdots$$
$$\begin{bmatrix}1 & \cdots & 0 & 0 & a_{1m}\\ \vdots & & \vdots & \vdots & \vdots\\ 0 & \cdots & 1 & 0 & 0\\ 0 & \cdots & 0 & 1 & 0\\ b_{1m} & \cdots & 0 & 0 & 1+a_{1m}b_{1m}\end{bmatrix}\times\begin{bmatrix}1 & 0 & 0 & \cdots & 0\\ 0 & 1 & a_{23} & \cdots & 0\\ 0 & b_{23} & 1+a_{23}b_{23} & \cdots & 0\\ \vdots & \vdots & \vdots & & 0\\ 0 & 0 & 0 & \cdots & 1\end{bmatrix}\quad(2)$$
$$\cdots\begin{bmatrix}1 & \cdots & 0 & 0 & 0\\ \vdots & & \vdots & \vdots & \vdots\\ 0 & \cdots & 1 & 0 & 0\\ 0 & \cdots & 0 & 1 & a_{m-1,m}\\ 0 & \cdots & 0 & b_{m-1,m} & 1+a_{m-1,m}b_{m-1,m}\end{bmatrix}$$

其中 a_{ij},$b_{ij}\in[0,2^L-1]$是扩展密钥。

本文对文献[6]提出的 Hash 函数方案进行了分析。选择了一个 10 维的 cat 映射,即 m=10,每个数 32 bit 长,L=32,共 320 bit。输出取 4 个 32 bit 数 $x_1(i+1)$、$x_2(i+1)$、$x_3(i+1)$、$x_4(i+1)$,共 128 bit。分析输入 320 bit 中每个 bit 的变化对输出结果的影响。具体做法是:如果要测试的是输入中第 i bit 对输出结果的影响,那么对同一条消息第 i bit 取反,然后观察取反前后输出结果变化 bit 占总 bit 的百分比。对于每 bit,本文进行了 10 000 次的统计平均。分析结果如图 2 所示。

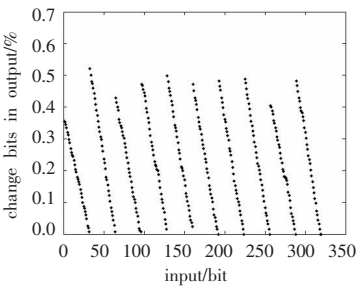

图 2　改进前的统计

图 2 横轴表示输入中发生变化的 bit 位,纵轴表示输出结果发生变化 bit 数占总 bit 数(128)的百分比。由图 2 可以看出,对于输入的每个 32 bit 数,输出结果对每个输入值的低比特位的变化非常敏感,对高比特位则不明感,甚至某些位,如 192 bit、256 bit、288 bit、319 bit、320 bit 的变化对输出没有影响。因此在文献[6]提出的方案中,如果输入的两条消息只是每个输入值不敏感的高比特位不同,其杂凑值 $h(x_1)$、$h(x_2)$就会相同,从而发生碰撞。

1.2　改进方案

在找出文献[6]方案存在碰撞基础上,提出了一种改进的 Hash 函数方案。

为了避免该碰撞,本文希望输出结果对于输入值高位的变化同样很敏感。在文献[6]中 h 函数是一个 m 维的 cat 映射。这里对 h 函数的构造进行了修改,在经过第一次映射后,将输出中的每个数对称互换。将互换后的数再进行一次映射。即将步骤 3 改为:(1)得到 $x_1(1)$,$x_2(1)$,…,$x_m(1)$后,代入方程(1)进行一次 cap 映射,得到$x_1'(1)$,$x_2'(1)$,…,$x'_m(1)$。

(2)将 $x_i(1)$,i=1,2,…,m 的比特位对称互换,即第 k 比特与第 33−k bit 互换,得到 $x_i^0(1)$,然后与 $x'_i(1)$,i=1,2,…,m 异或:

$$x''_i(1)=x_i^0(1)\oplus x'_i(1),\ i=1,2,\cdots,m \qquad (4)$$

(3)将得到的 $x''_1(1)$,$x''_2(1)$,…,$x''_m(1)$,代入方程(1)再进行一次映射,得到 $x_1(2)$,$x_2(2)$,…,$x_m(2)$。

本文对改进后的 Hash 函数进行了分析。同样将输入设为 10 个 32 bit 的数,对每一 bit,本文进行了 10 000 次的统计。分析结果如图 3 所示。

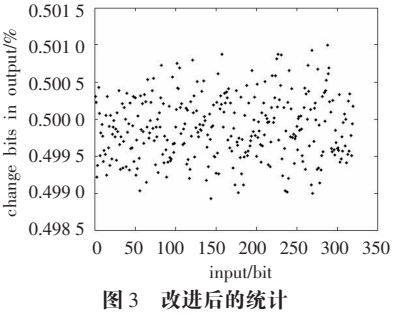

图 3　改进后的统计

从图 3 可以看出每一比特位的变化都会导致输出结果有 1/2 比例的变化,体现出系统良好的随机性,消除了原算法的碰撞。同时,对原算法不敏感比特位进行了进一步的统计分析,如图 4 所示。对于第 256 bit 位的变化,在 10 000 次统计中,输出 128 bit 中每比特变化的概率稳定在 50%左右。相比改进前,不仅不敏感位变得敏感,同时其对输出的影响也体现出良好的统

计性(其中的涨落来源于统计的样本数)。

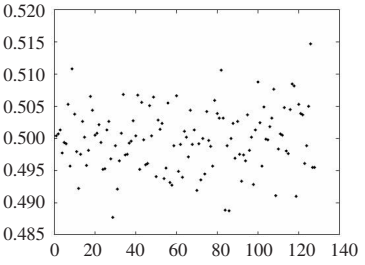

图 4　第 256 bit 变化对输出影响的统计

2　一次签名方案

在对文献[6]提出的 Hash 函数进行改进后，将根据这种改进的 Hash 函数提出一种一次数字签名方案。

通常为了减少签名长度，同时增加签名抵抗攻击的能力，在数字签名前，先对消息进行压缩，获得一个摘要[7]，如 MD4、MD5 将消息压缩为 128 bit 的摘要。本文设计的签名方案是对 128 bit 的摘要进行签名。

2.1　算法描述

具体签名方案描述如下：

系统参数：设 T 是一个 128 bit 的摘要。y 是长度为 128 的随机数组，$y=\{y_i,i=1,2,\cdots,128,y_i\in[0,2^{128}-1]\}$；$m$ 是 128 bit 的二进制随机数组，$m=\{m_j,j=1,2,\cdots,128,m_j\in[0,1]\}$；$z$ 是长度为 128 的数组，$z_i=H(y_i)i=1,2,\cdots,128$。

私有密钥：y

公开密钥：m,z

签名算法：$Sig(T)=Op(mid)$

$mid_i=y_i\times(T_i\oplus m_i),i=1,2,\cdots,128$

验证算法：

$Ver(T,Sig)$=TRUE⇔

Whether $Op(mid')=H(Sig)$

$mid'_i=y_i\times(T_i\oplus m_i),i=1,2,\cdots,128$

方案中通过筛选私钥 y 产生签名，用公钥来验证签名是否有效。原理如图 5 所示，如果等式成立则证明签名有效，否则签名无效。其中，$y=Op(x)$是取出数组 x 中的非零元素组成一个新的数组 y。

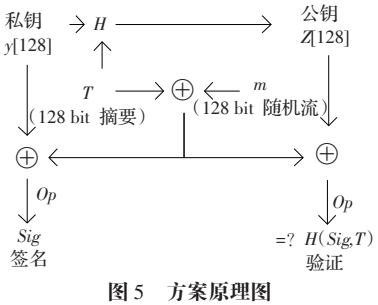

图 5　方案原理图

函数 H 是一个 Hash 函数。根据签名设计的实际需要，用前面提出的改进 Hash 函数来构造。下面具体描述一下实现方法。

2.2　改进 Hash 函数构造 H

签名中 $z_i=H(y_i),i=1,2,\cdots,128$，私钥 y_i 通过函数 H 得到相应的公钥 z_i。设 $y_i\in[0,2^{128}-1]$，对应 $z_i\in[0,2^{128}-1]$。用 6 维 cat 映射来构造图 5 中的 H 函数，方程(1)中 $m=6,L=32$，具体步骤如下：

(1)输入 y_i 为128 bit，将其分为 4 组，每组 32 bit，分别作为 $x_1(1)$、$x_2(1)$、$x_3(1)$、$x_4(1)$将 128 bit 的摘要分为两块，得到 $t1$，$t2$，每一块 64 bit。首先将 $t1$ 分为 2 组，每组 32 bit，分别对应于方程(1)中的 $x_5(1)$、$x_6(1)$。

(2)得到 $x_1(1),x_2(1),\cdots,x_6(1)$后，带入方程(1)进行一次 cap 映射，得到 $x_1'(1),x_2'(1),\cdots,x_6'(1)$。

(3)将 $x_i(1),i=1,2,\cdots,6$ 的比特位对称互换，即第 k bit 与第 33−k bit 互换，$k=1,2,\cdots,16$，得到 $x_i^0(1)$，然后与 $x_i'(1)$异或：$x_i''(1)=x_i^0(1)\oplus x_i'(1),i=1,2,\cdots,6$。

(4)将 $x_1''(1),x_2''(1),\cdots,x_6''(1)$带入方程(1)再进行一次映射，得到 $x_1''(2),x_2''(2),\cdots,x_6''(2)$。

(5)将 $x_1''(2),x_2''(2),\cdots,x_6''(2)$作为输入，重复(2)~(4)十次，得到 $x_1(2),x_2(2),\cdots,x_6(2)$。

(6)$x_1(2)$、$x_2(2)$、$x_3(2)$、$x_4(2)$不变。将 $t2$ 分为 2 组，每组 32 bit，代入 $x_5(2),x_6(2)$。

(7)重复(3)~(6)，最后取出 $x_1(3)$、$x_2(3)$、$x_3(3)$、$x_4(3)$作为输出值 z_i。

2.3　用 Chen 氏系统产生私钥 y

对于一次签名，算法 y 中不能重复使用。这里用 Chen 氏混沌系统来产生私有密钥 y。Chen 氏混沌系统由是 Chen G 于 1999 年首先提出。Chen 氏系统与 Lorenz 系统非常相似。但是它们的拓扑结构不同，使得 Chen 氏系统具有大量新的动力学特性[8]。所以，Chen 氏系统的动力学特性比 Lorenz 系统更复杂，表示如下：

$$\begin{cases}\dot{w}_1=a(w_2-w_1) & (5)\\ \dot{w}_2=(c-a)w_1-w_1w_3+cw_2 & (6)\\ \dot{w}_3=w_1w_2-bw_3 & (7)\end{cases}$$

其中，a、b、c 是系统参数。如果 $a=35,b=3,20\leqslant c\leqslant 28.4$，系统就会产生混沌[9]。下面介绍如何通过 Chen 氏系统来产生私有密钥 y。

本文首先选择 Chen 氏系统中的参数 a、b、c 和初值 w_1、w_2、w_3。产生私要 y 的方法如下：

(1)设置 Chen 氏系统初值，采用 Runge-Kutta 法，选用 0.001 步长，迭代 100 次。

(2)Chen 氏系统继续迭代，w_1 迭代 1 024 次，每迭代一次取出 w_1，得到 $w_1(i),i=0,1,\cdots,1\,024$。

(3)取出 $w_1=\{w_1(i),i=0,1,\cdots,1\,024\}$。对每一个 $w_1(i)$有：

$w_i(i)=\text{int}(((|w_i(i)|-\text{int}(w_i(i)))\times 2^{50})\bmod 2^{16})$

(4)将每 8 个 $w_1(i)\in[0,2^{16}-1]$共 128 bit 对应于一个 y_j，得到长度为 128 的十进制数组。

这样就得到私有密钥 y。

3　安全性分析

数字签名要有很好的真实性，即签名是不可伪造的。本文设计的是一次签名方案，对于一个密钥仅有一个摘要能被签名。本文将从以下几个方面对方案的真实性进行分析：

首先，签名 T 是由摘要筛选出来的私钥 y 组成，如果摘要中 1 的 bit 位过多，签名中暴露出大量或全部的私钥 y，这样就很容易伪造签名。为了避免以上的缺陷，引入了一个二进制比特流 m，它与摘要异或后将 1 的比重控制在 50%左右，这样签

名 Sig 就只会暴露出 50%左右的私钥 y，控制暴露私钥 y 的数量，也控制签名的长度。

在控制签名长度后，本文假设一个攻击者知道一个摘要 $T1$ 以及其有效签名 $Sig1$，同时知道公钥 m,z，他想以此来伪造另一篇不同的摘要 $T2\neq T1$ 的签名。那么可能情况有以下两种：

（1）$T1_i\oplus m_i, T2_i\oplus m_i, i=1,2,\cdots,128$ 中，存在 $a\in[1,128]$，使 $T1_a\oplus m_a=0, T2_a\oplus m_a=1$。这种情况下攻击者无法从签名 Sig 中得到 y_a，因此签名者将无法伪造签名。

（2）$T1_i\oplus m_i, T2_i\oplus m_i, i=1,2,\cdots,128$ 中，存在 $a\in[1,128]$，使 $T1_a\oplus m_a=1, T2_a\oplus m_a=0$。此时由于 $Sig2\in Sig1$，攻击者似乎能够由已知的签名 $Sig1$ 来伪造摘要 $T2$ 的签名 $Sig2$。然而由于该签名方案中，公钥 z 是由私钥 y 和摘要 T 产生，因此对于不同的摘要，其公钥也不会相同，这样 $H(Sig2,T2)\neq Op(mid1')$，其中 $mid1_i'=z_i\times(T2_i\oplus m_i), i=1,2,\cdots,128$，签名 $Sig2$ 验证无效。

因此，在同一个的公钥下，想伪造其它摘要的签名是不可能的。如果想穷举私钥 y 在计算上也是不可能的。以签名长度是私钥长度的 50%为例，如果要穷举出私钥 y，需要进行 $2^{128\times(128\times50\%)}=2^{8\,192}$ 次签名运算。同时函数 H 是 Hash 函数，具有很好地单向性，因此想通过公钥 z 推算 y 是非常困难的。

签名中私钥 y 不能重用。本文设计的方案中，采用了 Chen 氏系统来设计私钥 y。由于 Chen 氏系统对初值非常敏感，微小变化的初值导致截然不同的结果。所以只要选用不同的初值，即使是微小的不同，私钥 y 将不同。避免了私钥 y 重复使用。

数字签名要是可靠的。即对于一条摘要和私钥，只能有一个有效签名。在本文设计的方案中，采用了改进的 Hash 函数，消除了其中存在的碰撞，这样对于不同的私钥 $y_1\neq y_2$，经过 Hash 函数处理后，产生的公钥 $z_1\neq z_2$。所以可以保证签名的可靠性。

4 结束语

本文首先对一种 Hash 函数提出了改进，同时根据改进的 Hash 函数提出一种一次数字签名方案。本文设计的签名方案其签名长度比以往的一次签名方案要短，相对于 Lamport 签名算法，本方案产生的签名的长度只是 Lamport 的一半。方案中运用了改进的混沌 Hash 函数，以及 Chen 氏系统的伪随机性、对初值的敏感性等特点使签名足够安全。同时由于混沌系统的结构相对简单，提高了运算速度且易于实现。本文在理论上对该方案进行了描述，同时作出了一定的安全性分析。下一步将考虑如何对方案的安全性进行更严密的论证。

（收稿日期：2007 年 5 月）

参考文献：

[1] 冯登国.计算机通讯网络安全[M].北京：清华大学出版社，2001：43-45.

[2] Lamport Leslie.Constructing digital signatures from a one way function[R].Technical Report CSL-98，SRI International，1979.

[3] Michael O Rabin.Digitalized signatures[C]//Richard A D，David P D，Anita K J，et al.Foundations of Secure Computation：Academic Press，1978：157-162.

[4] 彭飞，丘水生，龙敏.基于二维超混沌映射的单向 Hash 函数构造[J].物理学报，2005，54(10)：4562-4568.

[5] 刘军宁，谢杰成，王普.基于混沌映射的单向 Hash 函数构造[J].清华大学学报：自然科学版，2000，7：57-60.

[6] KWOK H S，WALLACE K S T.A chaos-based cryptographic hash function for message authentication[J].International Journal of Bifurcation and Chaos，2005，15(12)：4043-4050.

[7] 章照止.现代密码学基础[M].北京：北京邮电大学出版社，2004：174-208.

[8] Chen G，Ueta T.Yet another chaotic attractor[J].int J Bifur Chaos，1999，9(7)：1465-1466.

[9] Guan Zhi-Hong，Huang Fangjun，Guan Wenjie.Chaos-based image encryption algorithm[J].Physics Letters A，2005：393-399.

（上接 92 页）

于 F 的商集 $\forall[x],[y]\in M/F$，定义：$[x]\rightarrow[y]=[x\rightarrow y]$。

命题 11 商集 M/F 中定义的运算“$\rightarrow$”是合理的。

证明 设 $x,y,u,v\in M$ 且 $x\sim u, y\sim v$；要证 $x\rightarrow y\sim u\rightarrow v$。

事实上，由于 $x\sim u\Rightarrow x\rightarrow u\in F, u\rightarrow x\in F; y\rightarrow v\Rightarrow y\rightarrow v\in F, v\rightarrow y\in F$；

由 u-(3) $y\rightarrow v\leqslant(x\rightarrow y)\rightarrow(z\rightarrow v), v\rightarrow y\leqslant(x\rightarrow v)\rightarrow(x\rightarrow y)$；又 $(y\rightarrow v)\in F,(v\rightarrow y)\in F, F$ 为滤子，由命题 7 知 $(x\rightarrow y)\rightarrow(x\rightarrow v)\in F,(x\rightarrow v)\rightarrow(x\rightarrow y)\in F$。即 $x\rightarrow y\sim x\rightarrow v$；再由 u-(3)有 $u\rightarrow x\leqslant(x\rightarrow v)\rightarrow(u\rightarrow v'), x\rightarrow u\leqslant(u\rightarrow v)\rightarrow(x\rightarrow v)$；又 $(u\rightarrow x)\in F,(x\rightarrow u)\in F, F$ 为滤子，由命题 7 知 $(x\rightarrow v)\rightarrow(u\rightarrow v)\in F,(u\rightarrow v)\rightarrow(x\rightarrow v)\in F$。故 $(x\rightarrow v)\sim(u\rightarrow v)$。

$\because\sim$ 具有传递性，$\therefore x\rightarrow y\sim u\rightarrow v$。

证毕。

命题 12 $(M/F,\rightarrow,',[0],[1])$是一个 UB 代数。这里$[a]'=[a']$

证明 直接用定义，略。

定义 7 称$(M/F,\rightarrow,',[0],[1])$为由滤子 F 诱导的(商)UB 代数，[1]为最大元，[0]为最小元。（收稿日期：2007 年 7 月）

参考文献：

[1] Borns D W，Mack J.An angebraic introducyion to mathematical logic[M].Berlin：Springer，1975.

[2] 何华灿.泛逻辑学原理[M].北京：科学出版社，2001.

[3] 罗敏霞，何华灿.一种泛逻辑代数系统[J].计算机工程与应用，2005，41(14)：21-22.

[4] Pavelka J.On Fuzzy Logic I[J]Z Math Logic Grund Math，1979，25：45.

[5] 肖云萍，邹庭荣.N-半单代数与蕴涵代数[J].模糊系统与数学，2006，1：99-102.

[6] Zou T R.The vartity of commutative BCK-algebras is Z-based[J].The Southeast Asian Bulletin of Math，2000，23.

[7] Xiao Yunping，Zou Tingrong.On Fuzzy BCC-ideals and quotient BCC-algebra induced by a Fuzzy BCC-ideal[J].The Southeast Asian Bulletin of Math，2006，30：165-175.

[8] 邹庭荣.关于 M-R 公开问题的注记[J].数学学报，2000，3.