网络入侵检测中序列模式挖掘技术研究*)

赵 峰 李庆华 赵彦斌

(国家高性能计算中心 武汉430074) (华中科技大学计算机科学与技术学院 武汉430074)

摘 要 面向入侵检测的数据挖掘是目前国际上网络安全和数据库、信息决策领域的最前沿的研究方向之一。入侵检测中进行序列模式挖掘时,由于频繁网络模式和频繁系统活动模式只能在网络或操作系统的单个审计数据流中获得,因而传统从事件流数据中获取单序列模式的算法,以及从不同多数据序列中获取多个序列模式的算法都不再适用。本文研究了入侵数据的特性,提出了网络入侵检测中序列模式挖掘框架和实时序列模式挖掘模型,并设计了一种新的面向入侵检测,基于轴属性、参考属性、相关支持度的序列模式挖掘算法 SPM-ID(Sequential Patterns Mining for Intrusion Detection)。最后在 KDD Cup99数据集的基础上实现算法及分析算法的性能。

关键词 入侵检测,序列模式,轴属性,参考属性,相关支持度

Sequential Patterns Mining Approach for Network Intrusion Detection

ZHAO Feng LI Qing-Hua ZHAO Yan-Bin

(National High Performance Computing Center(WuHan), Wuhan 430074)

(School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074)

Abstract One of the most advace research problems of network security, database and information decision is Data Mining based on Instrusion Detection. Since in sequential patterns mining for IDS, frequent network patterns and system activity patterns are got from operation system and single audit stream, the old sequential patterns mining algorithms are not fit for ID, which include mining single pattern from event stream and mining patterns from data sequences. We put forward the framework and the realtime mining model of sequential patterns mining in IDS , and more, we design a new algorithm based-on axis-attributes, reference-attributes and relative support for intrusion dectection whose name is SPM-ID(Sequential Patterns Mining for Intrusion Detection). What more, we implemente SPM-ID on enviorment in KDD Cup99 data set.

Keywords Intrusion detection, Sequential patterns, Axis attribute(s), Reference attribute(s), Relative support

1 引言

伴随着网络的发展,产生了各种各样的问题,其中安全问题尤为突出。入侵检测系统[1](IDS, Intrusion Detection System)是保护网络安全的关键技术和重要手段,它是一种主动保护自己免受攻击的网络安全技术,是对入侵行为的发觉。操作系统的日益复杂和网络数据流量的急剧增加,导致了审计数据以惊人速度剧增,如何在海量的审计数据中提取出具有代表性的系统特征模式,以对程序和用户行为作出更精确的描述,是实现入侵检测的关键。将数据挖掘技术用于入侵检测领域,是实现 IDS 智能化的重要手段,其主要思想是利用数据挖掘中的关联分析、序列模式分析、分类分析、元-分类分析等算法提取相关的用户行为特征,并根据这些特征生成安全事件的分类模型,应用于安全事件的自动鉴别。

序列模式挖掘是时态数据挖掘中的一个重要组成部分。入侵检测中,通过学习网络事件的频繁序列模式来掌握网络攻击的本质,从而发现新的入侵模式,是入侵检测智能化的一个重要手段。同时也是 IDS 研究中的一种新的趋势。

美国哥伦比亚大学的 Wenke Lee 和他的同事们提出了一个用于 IDS 的数据挖掘框架[2],企图通过对网络审计数据进行挖掘,发现新的模式以提高 IDS 的自适应性。Mississippi 州立大学的 Susan M. Bridges 等人采用模糊数据挖掘技术对 Wenke Lee 的框架进行改进,以使 IDS 具有某种程度的智能性。Massachusetts 的 MITRE 研究小组探索用数据挖掘处理入侵检测超载[3],他们用仿型(profiling)和分类分别处理误用检测和异常检测。国内对 IDS 中数据开采技术的研究不多,中国科学院研究生院信息安全国家重点实验室对于基于模式挖掘的异常检测有一定的研究。但目前国内外对 IDS 中序列模式挖掘研究尚有欠缺。

本文首先建立了入侵检测中序列模式挖掘框架,其后讨论入侵检测中序列模式挖掘算法,然后提出入侵检测中实时序列模式挖掘模型,最后通过基于 KDD Cup 99的实验来验证本文建立的模型。

2 序列模式挖掘框架

2.1 相关理论

为使本文能在概念上自包含,现给出所有可能涉及的定义。

定义1 非空集合 $I=\{i_1,i_2,i_3,\cdots,i_m\}$称为项集,其中 i_k 称为项[4]。

定义2 序列是项集的有序表,记为 $a=a_1\rightarrow a_2\rightarrow\cdots\rightarrow a_n$,其中 $a_k\subset I(k=1,\cdots,n)$。含有 k 个项的序列长度为 k,称为 k

*)基金项目:国家自然科学基金(60273075)。

序列($k=\Sigma|a_i|$)[4]。

定义3 序列模式也称序列关联，可表示成如下形式[5]：

when A occurs ⇒

B occurs within some certain time

定义4 判断序列模式是否有效的参数称为模式浓度(Pattern's strength)。

模式浓度通常基于在给定数据下模式发生的频率来计算。模式发生的频率有5种计算方式[5]：COBJ、CWIN、CWINMIN、CDIST_O、CDIST。

模式浓度通常基于在给定数据下模式发生的频率来计算。如果模式出现得足够频繁，我们就称它为有效模式。决定模式是否有效的参数有：①支持度；②可信度；③重要性；④覆盖度。

对于给定的一个有时戳标记的事件记录集，每个记录是一些项的集合，时距$[t_1,t_2]$表示事件序列从t_1开始，t_2结束，时距的宽度定义为$w=t_2-t_1$。以X表示是项集，那么一个时距就表示包含X的一次最小出现，也就是说该时距的任何子时距都不包含X的出现。

定义5 频繁有效事件(frequent episode)可表示为如下形式[6]：

$X,Y \rightarrow Z[c,s,w]$

其中X,Y,Z是项集；s是支持度，$s=\text{support}(X\cup Y\cup Z)$；c是可信度，$c=\frac{\sup\ port(x\cup y\cup z)}{\sup\ port(x\cup y)}$；w表示规则每次出现都必须在w范围内。

定义6 顺序有效事件规则(serial episode rule)指X,Y,Z在事务发生过程中遵循局部时间顺序，比如说Z在Y之后，并且Y在X之后。

2.2 数据收集

在IDS中进行序列模式挖掘的第一步就是数据收集，数据收集的内容包括系统、网络、数据及用户活动的状态和行为。数据采集主要有两个来源：从网络流中获取数据和从系统的日志文件和审计数据中获取信息。

2.3 预处理

预处理主要是理解并接受用户的发现需求，确定发现任务，抽取并发现与任务有关的知识源，根据背景知识中的约束规则对数据进行合法性检查，通过清理和归纳等操作，生成供序列模式挖掘核心使用的目标数据，即知识基，知识基是原始数据库经汇集后得到的二维表，横向为Tuples或Records，纵向为属性(Attributes或Fields)。在面向IDS的序列模式挖掘中，我们从网络数据包和系统日志及审计数据中获取信息时，数据量非常庞大，并且存在杂乱性、重复性和不完整性等问题，因而有必要对数据进行预处理。

从网络流上截获的数据主要包括网段内部主机之间，以及网络内部主机和外部主机之间通信的数据包头及数据内容。对这类数据的预处理主要集中在提取几个方面的特征值：网络连接特性、连接的内容特性、连接的统计特征。

对系统日志和审计数据预处理的方式有：数据集成、数据清理、数据简化等。

2.4 序列分析

IDS中进行序列模式挖掘时由于频繁网络模式和频繁系统活动模式只能在网络或操作系统的单个审计数据流中获得，因而传统从事件流数据中获取单序列模式的算法，以及从不同多数据序列中获取多个序列模式的算法都不再适用。我们针对IDS中数据的特殊性，设计了一种新的面向入侵检测的序列模式挖掘算法，我们称之为SPM-ID算法。

SPM-ID算法的基本思想是：由于频项集的子集也是频繁的，我们可以充分利用关联规则挖掘算法中的数据结构和库函数，来挖掘长度>=2的频繁有效事件；关联规则算法中的原始矢量(raw vector)被用作时间间隔矢量，它的每对值是时间间隔的临界值；最小、无过载的时间连接函数(temporal join)用于从两个长度为k-1的频繁项集时间间隔矢量来创建长度为k的侯选项集时间间隔矢量。计算频繁序列模式算法可分为两个步骤：①通过"轴"特征找到频繁关联；②在已有的频繁关联的基础上生成频繁序列模式。

1. 基于属性的有效性测量　我们可利用关于审计记录的计划级(schema-level)的信息来知道模式挖掘过程。也就是说，对于一特定的数据分析任务，尽管我们事先不知道哪个包含具体属性值的模式有效，但我们可以知道哪些属性更重要，更有用。

在不考虑任何专业背景的前提下，传统Apriori算法默认用最小支持度和最小可信度来判断模式是否有效。也就是说，假定I是模式p的有效性测量，那么

$I(p)=f(\text{support}(p),\text{confidence}(p))$

其中f是一些分等级函数。

如果要将计划级信息加入到有效性测量中，假设IA是包含特殊属性的模式p的有效性测量，那么

$I_e(p)=f_e(I_A(p),f(\text{support}(p),\text{confidence}(p)))=f_e(I_A(p),I(p))$

其中f_e是考虑了模式中属性的分等级函数。

审计数据计划级特征的表现形式是"哪些属性必须考虑"，也就是说找到哪些属性可作为引导来挖掘相关的特征。在设计算法的时候，我们并不在后处理阶段使用I_A测量通过分等级排序来过滤掉无用的规则，而是在候选项集生成的过程中将I_A作为项的约束条件。

2. 使用轴属性　在审计数据的属性中存在重要性的排序。对于描述数据而言，有些属性是必需的，有些属性则是辅助性质的。

表1　网络连接审计数据

Time stamp	Service	Src_host	Dst_host	Src_bytes	Dst_bytes	Flag
1.1	telnet	A	B	100	2000	SF
2.0	ftp	C	B	200	300	SF
2.3	Smtp	B	D	250	300	SF
3.4	telnet	A	D	200	12100	SF
3.7	Smtp	B	C	200	300	SF
5.2	http	D	A	200	0	REJ
…	…	…	…	…	…	…

表1描述了网络连接的审计数据。每条记录表示了一个网络连接，连接属性除了时戳timestamp之外，连续的属性值可被离散到一个适当的盒子中。一个网络连接可被唯一地表示成：

〈timestamp,src_host,src_port,dst_host,service〉

也即连接的开始时间，源主机，源端口，目的主机，服务。这些属性对于描述网络数据来说是必需的。我们认为有效的关联规则是描述和"必要属性"相联系的模式，仅仅包含"辅助属性"的模式经常是无用的。例如，基本算法可产生如下规则：

src_bytes=200→flag=SF,[0.6,0.2]

这些规则是无用的,在某种意义上甚至是误导。

当"必要属性"在关联规则挖掘算法中项的约束条件时,我们称之为"轴属性"。在候选集生成过程中,一个项集必须包含轴属性的值。我们对非轴属性之间的关联不感兴趣,也就是说:

$$I_A(p)=\begin{cases}1 & \text{如果 } p \text{ 包含轴属性}\\ 0 & \text{其他}\end{cases}$$

在实际应用中,并不是所有的必要属性都是轴属性。有些网络分析任务需要不同网络服务的静态信息,而其他一些任务需要和主机相关的模式。这些时候,我们就要选择适当的轴属性。

对于频繁有效时间来说,用轴属性限制项的生成是至关重要的。为此我们引入如下定理。

定理1 s是关联规则A→B的支持度,N是有效事件规则的总数,这些有效事件规则有如下表现形式:

(A|B)(,A|B)*→(A|B)(,A|B)*

那么N至少是s的指数。

基本算法只能生成包含不重要属性值的顺序有效事件规则,例如:

src_bytes=200,src_bytes=200→

dst_bytes=300,src_bytes=200,[0.4,0.2,2s]

注意这里的src_bytes=200来自不同的连接。如果包含非轴属性的关联规则A→B的支持度很高,根据上面的定理,那么将会产生很多形如(A|B)(,A|B)*→(A|B)(,A|B)*的无用的顺序有效事件。

为避免产生大量无用的事件规则,我们将基本的频繁事件规则算法扩展为频繁序列模式挖掘算法。算法分为两步:①通过"轴"特征找到频繁关联;②在已有的频繁关联的基础上生成频繁序列模式。也就是对第二步而言,事件项集(episode itemsets)建立之后,它的项是有关轴属性的关联,并且属性是有值的。如下是一个规则的例子。

(service=smtp,src_bytes=200,dst_bytes=300,flag=SF),(service=telnet,flag=SF)→(service=http,src_bytes=200),[0.2,0.1,2s]

事件规则中的每个项集,比如(service=smtp,src_bytes=200,dst_bytes=300,flag=SF),是一个关联。我们实际上是要将属性中的关联和记录中的序列模式联合成一个单一的规则,这个规则不仅能除去无用的模式,还能提供审计数据丰富有用的信息。

3. 使用参考属性 除"轴"属性外,另外一个我感兴趣的系统审计数据的特征是:有些属性是另外一些属性的参考。这些"参考属性"经常携带有关某"主题"(subject)的信息,而其他属性描述同一主题的"行为"(action)。

表2表示了访问一个Web站点的日志。Action和request是主题的行为,主题是指remote host。其中有一些remote hosts,每个remote host有一系列的请求:"/images","/images","/shuttle/images/sts-71"。当我们寻找序列"行为"模式时,用"主题"作为参考是很重要的,因为其他主题的行为经常是无用的。这种类型的序列模式可表示成如下形式:

(subject=X,action=a),(subject=X,action=b)→(subject=X,action=c),[confidence,support,window]

表2 Web日志记录

Timestamp	Remote host	Action	Request
1	His.moc.kw	GET	/images
1.1	His.moc.kw	GET	/images
1.3	His.moc.kw	GET	/shuttle/images /sts-71
…	…	…	…
3.1	Taka10.taka.uec.ad.jp	GET	/images
3.2	Taka10.taka.uec.ad.jp	GET	/images
3.5	Taka10.taka.uec.ad.jp	GET	/shuttle/images /sts-71
…	…	…	…
8	Rjenkin.hip.cam.org	GET	/images
8.2	Rjenkin.hip.cam.org	GET	/images
9	Rjenkin.hip.cam.org	GET	/shuttle/images /sts-71
…	…	…	…

需要注意的是在模式的每次出现内,"行为"的值指向相同的"主题",尽管具体的"主题"值没有给出来,这是因为对于真个数据集来说,任何特殊的主题值不一定是频繁的,也就是说"主题"仅仅是"参考",或"变量"。

基本频繁有效事件算法可扩展到考虑参考属性。简单说,当形成一个有效事件时,有一个附加的条件就是,在它最小出现内,组成的项集的记录有相同的参考属性值。也即

$$I_A(p)=\begin{cases}1 & \text{如果项集 } p \text{ 包含相同的参考属性值}\\ 0 & \text{其他}\end{cases}$$

4. level-wise 近似挖掘 在实际应用中,经常有必要包含低频繁模式。在日常网络交通中,有些服务,比如gopher,出现的次数很少,但我们仍需要包含它们的模式到网络交通配置文件中,因为我们需要每个所支持服务的表现。如果我们提供给数据挖掘算法一个低的支持度,我们能得到很多大量的不必要的模式,而这些模式又和高频繁发生的服务相连,如smtp。

我们通过如下所示算法从审计数据中挖掘出序列挖掘模式。

```
Input:database D ,最终最小支持度 s_t,初始最小支持度 s_i,轴属性(s)
Output:频繁有效事件规则(frequent episode rules)Rules
Begin
    (1)R_restricted=∅;
    (2)扫描数据库D形成L={frequent 1-itemsets that meet st};
    (3)s=s_i;
    (4)while (s>=s_t)do begin
    (5)从L中计算有效事件:每个有效事件必须至少包含一个轴属
       性的值不在R_restricted内;
    (6)添加新的轴属性的值到R_restricted中;
    (7)添加新的有效事件规则到规则集Rules中;
    (8)s=s/2;/* use a smaller support value for the next iteration
       */
    end
end
```

该算法的基本思想是:首先找到和高频繁轴属性值相关的有效事件,例如:

(service = smtp;src_bytes = 200);(service = smtp;src_bytes = 200)→(service = smtp;dst_bytes = 300);[0.3;0.3;2s]

然后我们迭代降低支持度的入口,通过限制已在输出有效事件中的旧轴属性的加入,来发现和低频繁轴属性值相关的有效事件。更具体地说,当一个有效事件生成时,它必须至少包含一个新的轴值,也即低频繁轴值。例如,在第二次迭代的过程中,smtp现在是一个旧轴值,我们得到如下的有效事

件规则：

(service = smtp;src_bytes = 200);(service = http;src_bytes = 200)→(service = smtp; src_bytes = 300);[0.4; 0.1; 2s]

当一个足够低的支持度出现时，这个过程就会终止，实际上它可以是所有轴值中的最小值。

需要注意的是，对于一个高频繁的轴值来说，有些非常低的频繁有效事件是运行低支持度得到的，我们认为它们是无效的，因而常常忽略了它们。也就是说，对于每次迭代，有如下公式：

$$I_A(p)=\begin{cases}1 & \text{如果 } p \text{ 至少包含一个“新”的轴属性值}\\ 0 & \text{其他}\end{cases}$$

同时，需要强调的是，我们设计的算法是近似挖掘，因为“旧”轴值对我们获取“新”轴值的序列很重要，所以我们在用“新”(低频繁)轴值的同时要用到所有“旧”(高频繁)轴值来形成有效事件。

5. 在相关支持下挖掘(Mining with Relative Support)

另一个容易处理分布极不平衡的属性值的方式是利用相关支持度(relative support)。不像以往那样使用数据库中记录数，而是用数据库中每个唯一属性值出现的次数用做计算模式支持度值的参考。也就是说，如果相关支持度 s_i 是指定给属性 a_i 的，唯一的值，比如 $a_i=v_{ij}$，在数据库中共有 n_{ij} 次出现，那么如果包含 $a_i=n_{ij}$ 的模式至少出现 $si * n_{ij}$ 次，我们就认为该模式是频繁的。不同的相关支持度可用于不同的属性，下面给出了用相关支持度挖掘频繁模式的算法。

Input：database D，最小支持度 s，每个属性 a_i 的相关支持度为 s_i
Output：频繁模式(frequent patterns (association rules or frequent episodes))
Begin
(1)扫描数据库 D 形成 L={unique attribute values}，count 是每个属性的值；db_size 是 D 中的记录数；
(2)for each $v_{ij}\in$ L do begin
if si is speci_ed then
v_{ij}. rel_support=$s_i * v_{ij}$. count;
else
if $s_i * v_{ij}$. count<db_size then
remove v_{ij} from L;
else
v_{ij}. rel_support=s * db_size;
end/ * L is now the set of frequent 1-itemsets; * /
(3)从 L 中计算关联或频繁有效事件；
当候选集 lk 从 l_{k-1} 和 l_{k-1} 中创建，
l_k. rel_support=min(l_{k-1}. support,l_{k-1}. support)，
lk is frequent if l_k. count > l_{lk}. rel_support;
end

算法中，第一步是找出数据库中所有唯一的属性值；第二步是指定给一个属性的相关支持度，或“全面”支持度，向包含这些属性值的频繁模式设置 res_support 要求；第三步修改关联或频繁有效事件的侯选集的生成过程，当一个候选集包含不同相关度要求的不同属性值时，最小的值用于检查项集是否频繁，对于一个给定的模式 p 有：

$$I_A(p)=\begin{cases}1 & \text{如果 p 的 count 值不小于最小相关支持度}\\ 0 & \text{其他}\end{cases}$$

这个算法对于计算“稀有”事件(属性值分布极不平衡)的模式是很有用的。

2.5 后处理

当序列模式挖掘算法完成后，需要一个特征构造程序，自动从已经产生的模式中构造时间统计特征，用这些统计特征作为附加特征，提高了入侵检测系统的有效性。规则集合中规则的总数会不断增加并逐渐趋于稳定，在增长曲线趋于平缓时，就认为收集到的审计数据已经足够描述用户的正常行为了。在实验中用这种方法收集数据，并用学习到的分类模型检测异常，实验的结果说明用频繁规则集合作为收集审计数据是否充分的标志还是可行的。通过比较正常模式和几种有代表性的网络攻击的攻击模式，可以发现关键属性和参考属性的选择对计算和识别入侵检测模式非常重要，并通过一个构造统计特征的过程，利用关键属性和参考属性从事件序列中构造统计特征。

3 实时序列模式挖掘模型

入侵检测中，目前主要四类攻击：①DoS，denial-of-service，拒绝服务攻击，比如：ping-of death，teardrop，smruf，syn flood 等。②R2L，来自远程机器的未授权访问，比如：猜测密码。③U2R，本地未经授权的一般用户越权使用超级用户权限，比如：各种缓冲溢出攻击。④PROBING，监视和探测攻击，比如：port-scan，ping-sweep。根据攻击类型的检测难度不同以及网络数据滑动窗口理论，我们提出了基于滑动窗口的实时序列模式挖掘模型，如图1所示。图中粗实线是窗口大小的循环，细实线是数据流，虚线是挖掘模型的循环。

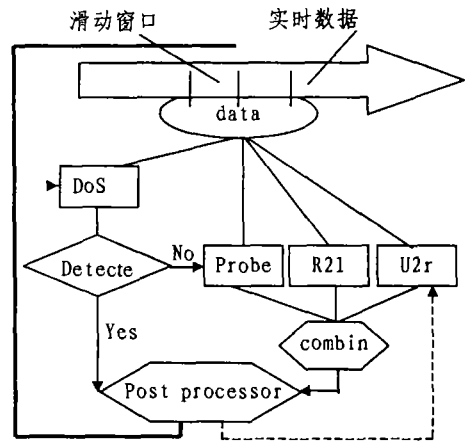

图1 基于滑动窗口的实时序列挖掘模型

4 实验结果及分析

由于 Apriori 算法自身算法特点，对支持度的大小和频繁项的长度变化十分敏感。较小的支持度使更多项目满足用户要求，频繁集元素个数增加，既而增加了候选集的个数，每趟迭代的计算量随之增加，将大大影响算法的性能。Apriori 算法是基于“自底向上”搜索模式。对于可能包含较长项目集的交易数据库，算法需要数量惊人的计算开销。因为假如某个频繁项目集的长度为 l，那么就必须生成它的所有 2^l 个子集(当然除了空集合以外)，因为它的每一个子集必然也是频繁的。因此这一成指数增长的复杂度基本上决定了 Apriori 算法只能应用于发现相对较短的频繁项目集。

以 Aprior 算法为基本算法的面向入侵检测的序列模式挖掘算法，克服了传统 Aprior 算法的局限，将数据属性分成几个等级，侧重于基于属性的序列模式，算法可应用于比较长的频繁项集，大大提高了入侵检测的精确性。我们以 KDD Cup99数据集为实验基础，从算法精确性、扩展性和适应性方面对算法进行了分析。

表3实验数据表示在可信度为2%，滑动窗口为2秒的情况下，随着支持度的不同，算法有不同的精确性。

表4中的实验结果表明响应时间随着数据库大小的增加而增加。可以看出 SPM-ID 算法有较好的可扩展性。

表3 算法精确性的比较

支持度	3%	2%	1.5%	1%
Aprior 算法	47%	57%	79%	92%
GSP 算法	44%	57%	78%	85%
SPM-ID 算法	54%	67%	90%	98%

表4 可扩展性分析(1%,2%,2s)

数据量	71.4M	71.4*2M	71.4*4M
Aprior 算法	154s	337s	613s
GSP 算法	143s	335s	550s
SPM-ID 算法	134s	210s	374s

我们在 AIX4.0,ReadHat7.2,Win2000Server 下的实验结果如表5所示,结果数据表明 SPM-ID 算法在不同的环境下有较好适应性。

表5 可适应性分析(数据库大小71.4M,(1%,2%,2s))

环境	Unix	Linux	Win2000
Aprior 算法	154s	187s	188s
GSP 算法	143s	145s	177s
SPM-ID 算法	134s	134s	137s

参 考 文 献

1 Denning D E. An intrusion detection model. IEEE Trans on Software Engineering, 1987, 13(2): 222～232

2 Lee W, Stoifo S J. Data mining approaches for intrusion detection. In: Proc. of the 7th USENIX Security Symposium. San Antonio, TX. Jan. 1998

3 http://www.cs.msstate.edu/～secrutiy/iids/publications/citss_ids.pdf

4 周斌,吴泉源. 序列模式挖掘的一种渐进算法[J]. 计算机学报, 1999,22(8):882～887

5 Joshi M, Karypis G. A Universal Formulation of Sequential Patterns: [Technical Report No. 99-021]. Department of Computer Science, University of Minnesota, 1999

6 Mannila H, Tovonen H. Discovering generalized episodes using minimal occurrences. In: Proc. of the 2nd Intl. Conf. on Knowledge Discovery in Databases and Data Mining, Portland, Oregon, Aug. 1996

(上接第68页)

常吉[15]则通过在 Brands 数字现金方案中嵌入由银行规定的数字现金的有效期来防止银行检察现金重用性的数据库的无限增长。

5 数字现金的应用现状

目前在国外已有一些数字现金系统在实际使用或试用。下面是两种比较有代表性的基于智能卡的数字现金系统:

VisaCash 是一个储值支付品牌,由 Visa 提供,消费者的智能卡跟踪保持在银行的现金帐户的金额,为付款者提供脱线的支付担保。商家收集所有支付,统一在银行进行批存款操作,一旦通过银行内部网络进行资金结算,这些收款者的资金有效。

Mondex electronic cash 与 VisaCash 不同,就像一个现金替代物。智能卡代表现金,资金可以从付款者的卡直接转到收款者,收款者重复接收资金而不需要在银行存放。在 Mondex 和普通货币之间转换需要内部金融网络,但不进行结算存款。

数字现金在欧洲和日本比较普及,主要用于小金额的购买活动,如:自动售货机、Internet 上的微支付等。但是由于数字现金的某些关键技术并未彻底解决,再加上存在一些相互竞争的技术而没有统一的数字现金系统的标准,从而给数字现金的进一步推广应用造成了困难。

结论 由于数字现金具有极为广泛的应用前景,因此对它的研究已成为当前学术界的一个热点。但是,在迄今人们已提出的各种数字现金方案中还没有一个方案可以真正地同时满足理想数字现金的各种性质要求。我们认为,数字现金未来的研究发展方向应该集中在如何解决其可分性、可传递性和如何提高其执行效率上,这样才有可能把数字现金真正推向实用化。

参 考 文 献

1 Schneier B(美). 应用密码学. 北京. 机械工业出版社, 2000

2 杨义先,等. 信息安全新技术. 北京. 北京邮电大学出版社, 2002

3 卿斯汉. 密码学与计算机网络安全. 北京:清华大学出版社, 2001

4 Chaum D, Fiat A, Naor M. Untraceable Electronic Cash. Advances in Cryptology-Crypto'88, 1990. 319～327

5 Okamoto T, Ohta K. Disposable Zero-knowledge Authentication and Their Applications to Untraceable Electronic Cash. Advances in Cryptology, Proceedings of Crypto'89 (Lecture Notes in Computer Science), 1990. 481～496

6 Franklin M, Yung M. Secure and Efficient Off-line Digital Money. In: Proc. ICALP'93 LNCS 700, Springer-verlag, 1993. 265～276

7 Brands S. Untraceable Off-Line Cash in Wallets with Observers. Advances in Cryptology-Crypto'93, 1993. 302～318

8 Okamoto T, Ohta K. Universal Electronic Cash. Advances in Cryptology-Crypto'91. 1992. 324～337

9 Camenisch J, Maurer U, Stadler M. Digital Payment Systems with Passive Anonymity-Revoking Trustees. Esorics'96. Italy. Springer-verlag, 1996. 33～44

10 陈恺,等. 基于概率验证的可分电子现金系统. 计算机研究与发展, 2000.6

11 史国庆,等. 加入概率验证的 Brands 数字现金改进方案算法. 计算机工程与科学, 2001.6

12 陈恺,等. 可撤消匿名性的可分电子现金系统. 西安电子科技大学学报, 2001.1

13 王常吉,等. 一个新的利用 Smart 卡的公正的电子现金系统. 计算机学报, 2001,12

14 谭运猛,等. 针对 Brands 电子现金支付协议的预处理算法. 通信学报, 2002.2

15 王常吉,等. 一个改进的基于限制性盲签名的电子现金系统. 电子学报, 2002,7

16 Yu H-C, His K-H, Kuo P-J. Electronic payment systems: an analysis and comparison of types. Technology in Society, 2002, 24:331～347

17 陈恺,等. 电子现金系统的研究与发展. 西安电子科技大学学报, 2000,4