离线匿名电子现金系统的设计与实现*)

彭建新[1] 王常吉[2]

(广东警官学院计算机系 广州 510230)[1] (中山大学计算机科学系 广州 510275)[2]

摘 要 本文利用 Java Applet 和 Java Network Launch Protocol 等技术设计和实现了一个基于 RSA 盲数字签名的离线匿名的电子现金系统。系统采用 B/S 和 C/S 相结合的架构,可以方便进行 Web 在线支付和点对点支付。系统通过数字证书提供对用户和银行的强身份认证,用户在银行网站进行存取款和在商家网站进行支付,通过 SSL 协议来保证通信数据的安全,同时用户数据以加密的方式保存在本地磁盘,并可方便地移植到智能卡设备。

关键词 电子现金,盲数字签名,切割选择

The Design and Implementation of Offline Anonymous Electronic Cash System

PENG Jian-Xin[1] WANG Chang-Ji[2]

(Department of Computer Science, Guangdong Police Officer College, Guangzhou 510230)[1]

(Department of Computer Science, Sun Yat-sen University, Guangzhou 510275)[2]

Abstract An off-line anonymous electronic cash system based on RSA blind signature is designed and implemented with the technologies such as Java Applet and Java Network Launch Protocol in this paper. The system framework combines B/S with C/S, which is convenient for online payment through the Web and point-to-point payment. The bank and the client are authenticated strongly by digital certificate, the interactive information between the client and the bank is secured by SSL protocol, and the client's data can be saved in the local disk in an encrypted way and also can be transferred to the smartcard devices.

Keywords Electronic cash, Blind signature, Cut and choose

1 引言

随着互联网应用日趋成熟,电子商务以其低成本、全球化市场、超越时空、多媒体手段与个性化服务等优点,正在成为研究和应用的热点。电子支付是电子商务活动中最核心和关键的环节,是电子商务得以进行的基础条件,没有支付系统,整个商务活动就不能实现。电子现金是一种非常重要的电子支付方式,它可以看作是现实货币的电子或数字模拟。电子现金以数字信息形式存在,通过互联网流通,理想的电子现金具有独立性、安全性(不可伪造性、不可重复使用性)、匿名性、便捷性、可传递性、可分性、鲁棒性和原子性等特性[1]。D. Chaum 于 1982 年最早提出一个在线的、匿名的电子现金方案[2],又于 1988 年最早提出一个离线的、匿名的电子现金方案[3]。电子现金已经经历了 20 多年的发展,研究者主要集中于设计满足诸如公平性和可分性等电子现金方案,很少有文章讨论电子现金系统的实现。迄今为止,满足所有理想特性的电子现金方案是不存在的,设计电子现金系统模型不可一概而论,而应该根据不同需求或应用背景来设计。本文主要针对可应用于 Internet 之上的小额支付,重点考虑实现电子现金的匿名性、离线性和便捷等最基本的属性。可以预见,电子现金在未来的电子商务中将会拥有非常广阔的市场和前景。

2 模型设计

D. Chaum 于 1982 年最早提出了盲数字签名的概念[2]。一个盲数字签名协议是用户和签名者之间的密码协议,用户得到签名者对他所提供的消息的签名,而签名者事后并不知道他所签名的消息的内容以及签名,即签名者得到信息与产生的消息签名对是统计无关的。盲签名在需要考虑匿名性的领域,如电子现金和电子选举等有着广泛的应用。

切割选择[4]使得盲数字签名技术成功应用于电子现金中。在本文提出的系统中,切割选择的思想主要用于如下两个方面:

1)用户对创建的 n 个相同的电子现金进行盲化,银行随机选择其中 $n-1$ 个要求用户揭示出盲因子,如果通过验证,则对剩下的那个电子现金进行签名。

2)用户的身份信息被切割成两部分,每个电子现金中都仅包含用户的部分身份信息。当用户在商家进行消费时,商家产生随机挑战数,用户应答商家的挑战信息,相当于去除盲因子。商家把电子现金、随机挑战数和用户的应答信息发送给银行进行兑换。如果用户重复花费,对于同一个电子现金,银行将以较大概率揭示出作弊用户的完整身份信息。

电子现金在一次完整的商务活动进程中至少涉及用户(U)、商家(S)、银行(B)等三方,其基本模型如图 1 所示。

基于图 1 所示的 3 个进程,取款时既可采用 B/S 架构(用户与银行之间通过 SSL 协议建立的安全通道进行通信),也可采用 C/S 架构(用户通过其电子现金管理软件直接和银行进行安全连接,用户首先从银行网站下载一个电子现金管理软件,并获得银行的公钥)。

本文系统采用 Java 的网络启动协议(Java Network Launch Protocol 简称 JNLP)。JNLP 同时具备 Java Applet 和 Java Application 的优点。JNLP 程序运行时将动态地从服务器下载资源,当用户在线时,可以实现自动检查并获取最新版本。

*)基金项目:国家自然科学基金项目(60503005);广东省自然科学基金重点资助项目(05200302)。**彭建新** 硕士,助教,主要研究方向为信息与网络安全。**王常吉** 博士,副教授,硕士生导师,主要研究方向为信息与网络安全。

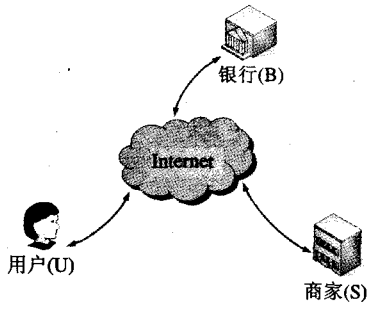

图 1　电子现金基本模型

首先，需要在 Tomcat 中配置 web. xml，添加如下代码，以支持对. jnlp 的解析。

```
〈mime-mapping〉
    〈extension〉jnlp〈/extension〉
    〈mime-type〉application/x-java-jnlp-file〈/mime-type〉
〈/mime-mapping〉
```

然后，生成和签发 JAR 文件。

第一步，创建 JAR 文件，如：

```
jar cf JCash. jar *. class
```

第二步，利用 keytool 创建一个私钥到 keystore 中，系统将提示要求输入私钥相关身份信息，如：

```
keytool -genkey -keystore jcashKeys -alias jcash
```

第三步，用刚才建立的密钥文件来签发 JAR 文件，如：

```
jarsigner -keystore jcashKeys JCash. jar jcash
```

最后，新建一个 xml 格式的 jnlp 文件，如 JCash. jnlp：

```
〈? xml version="1. 0" encoding="gb2312"?〉
〈jnlp spec="1. 0+" codebase="http://localhost:8080/
 test/"〉
  〈information〉
    〈title〉中山大学电子现金项目组〈/title〉
    〈homepage href="/test" /〉
   〈/information〉
  〈resources〉
    〈j2se version="1. 2+" /〉
     〈jar href = "http://localhost: 8080/test/JCash.
      jar"/〉
  〈/resources〉
  〈application-desc main-class="webcash. purse. JCash" /〉
〈/jnlp〉
```

这样，在浏览器输入 http://localhost:8080/test/JCash. jnlp，就可以通过浏览器执行 JCash. jar 这个应用程序。

用户和商家接口设计如图 2 所示。

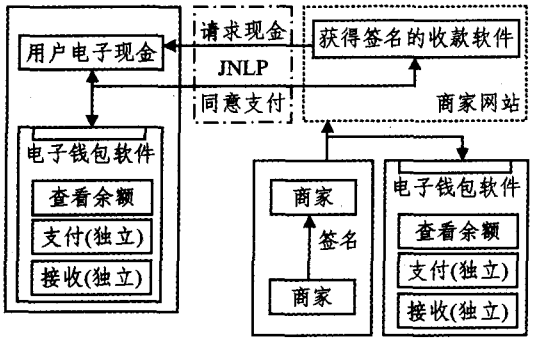

图 2　用户和商家接口设计

当用户确定接受商家网站提供的收费服务(比如，商家提供的收费软件下载服务)之后，用户浏览器将提示用户下载用 Java 实现的支付程序来进行电子现金的支付。由于支付程序将和用户本地的电子钱包软件进行交互，出于安全性考虑，用户可以验证商家使用的接收软件是否经过电子现金发行银行签名，输入支付密码之后，系统将自动完成支付进程。

3　协议流程

3.1　取款协议

首先，银行产生一个 RSA 公私密钥对，公钥为(e,n)，私钥为(d,n)。用户随机构造 p 对大整数组 $x_1,x_2,\cdots,x_p$ 和 $x'_1,x'_2,\cdots,x'_p$，满足 $x_i \oplus x'_i = ID_U (1 \leqslant i \leqslant p)$，$ID_U$ 代表用户的身份信息。

通过哈希函数 H 计算得到：$y_i = H(x_i)$，$y'_i = H(x'_i)$，用户得到的电子现金形式为 $M=(amount, y_1, y'_1, y_2, y'_2, \cdots, y_p, y'_p)$，其中 *amount* 代表要取的金额。由于哈希函数 H 的单向性，银行无法从电子现金 M 中揭示出用户的身份信息 ID_U。用户随机产生一个大整数 k 作为盲因子，同时计算出 $m=H(M)$，利用银行的公钥 e，把对 m 进行盲化之后的信封发送给银行，信封形式为$(m \times k^e) \bmod n$。

银行用自己的私钥 d 对信封进行签名，然后把签名了的信封返还给用户，签名之后的信封形式为$(m \times k^e)^d \bmod n$。

用户通过去除盲化因子的计算可以得到经由银行签名的合法电子现金，计算方法如下：

$$\frac{(m \times k^e)^d}{k} \bmod n = \frac{m^d \times k^{ed}}{k} \bmod n$$

$$= \frac{m^d \times k}{k} \bmod n = m^d \bmod n = signedNote$$

其中 *signedNote* 代表由银行签名了的电子现金。

3.2　支付协议

用户把$(M, signedNote)$发送给商家，商家首先利用银行的公钥 e 来验证电子现金的合法性，验证过程即判断 $H(M) = signedNote^e \bmod n$ 是否成立。如果等式成立，则证明电子现金确实是银行签名的合法电子现金。然后商家发送一串随机挑战数 r(假设为 p 比特：$r_1, r_2, \cdots, r_p$))给用户用来验证用户的合法身份。用户根据商家的随机挑战数，给出自己产生的 *RISpart*(部分身份信息)，*RISpart* 足够验证其身份的合法性，但不至于把全部的身份信息暴露给商家，具体过程为：

用户根据商家发送来的挑战数 r，把自己的应答信息 *RISpart* 送回给商家，*RISpart* 要求满足下式：

$$RISpart = \begin{cases} x_i & r_i = 0 \\ x'_i & r_i = 1 \end{cases} \quad \forall i \in (1,2,\cdots,p)$$

商家再根据对应于挑战数 r 的每一个比特位，来验证用户应答数 *RISpart* 是否全部满足下式：

$$\begin{cases} H(x_i) = y_i & r_i = 0 \\ H(x'_i) = y'_i & r_i = 1 \end{cases}$$

如果上式全部成立，则证明用户发来的电子现金和用户的身份一致。

3.3　存款协议

商家把$(M, signedNoet, RISpart, r)$发送给银行，银行首先验证电子现金的合法性和商家发来的挑战数和应答数是否一致，然后再检测其是否被重复消费或者重复存款。

银行通过判断两个 *RISpart* 是否相同可以很容易地检测出商家是否重复存款。如果 *RISpart* 不同，则对于商家随

机产生的两个挑战数：$r_1(r_{11}, r_{12}, \cdots, r_{1p})$和 $r_2(r_{21}, r_{22}, \cdots, r_{2p})$，用户相应地构造出了两个合法的 *RISpart*。银行可以通过如下办法把用户的身份揭示出来。

对于随机生成的 2 个 p 比特的随机串 r_1 和 r_2，其所有位完全一样的概率为$(1-2^{-p})$，因此很大概率上存在至少一位不相等，假设为第 i 位不相等，即 $r_{1i} \neq r_{2i}$。根据异或函数，银行可以获得用户的身份信息 ID_U 为 $x_{1i} \oplus x'_{2i}$或者 $x'_{1i} \oplus x_{2i}$。

4 安全性分析

本系统基于 RSA 盲数字签名的安全性，具有匿名性、不可伪造性、不可重复花费、不可重复存款等安全特性。

从电子支付系统的角度来看，它的安全性体现在：保证消费者的敏感数据的安全，保证电子商店的安全、支付网关的安全以及它们之间链路的安全（他们两两之间的传输数据不会被监听、篡改或伪造）。用户在取款时可能存在如下两种作弊可能。

1）声称其所取金额（该数目将由银行从用户的账户上扣除）比实际电子现金所含的价值要小。

2）在盲化的电子现金中包含了错误的用户身份信息。

NB_ENVELOPE 代表用户创建的信封内所含相同电子现金的数目。如前所述，通过切割选择协议，可以以一定的概率防止这种作弊行为。当 ENVELOPE 取 10 时，银行要求用户把 9 个构建电子现金的信息揭示出来以检验用户身份，那么用户作弊成功的可能性为十分之一，如果 ENVELOPE 取 100，那么用户成功作弊的可能性只有百分之一。显然，用户不会在小额支付上冒险。

NB_RIS 代表电子现金中用户的随机身份串信息的配对数目。检测出用户作弊的概率跟 RIS 配对数成几何级数增长。对于两次不同的商家挑战，能够发现作弊用户的概率为两个 RIS 配对数中至少有一个位不同，即发现用户重复消费的概率为 $p=1-\frac{1}{2^n}$。因此，如果 NB_RIS 取 10，则 $p_{10}=1-\frac{1}{2^{10}} \cong 0.999=99.9\%$

总结 本文所提出的系统基于 RSA 公钥密码体制，可以满足电子现金最基本的特性：匿名性、安全性、点对点支付、在线购物。它不仅可以直接支付电子现金给其他用户，也可以从其他用户处接收电子现金。同时，还可以支持在 Internet 上进行支付，且易于推广到基于蓝牙、SMS、非接触式智能卡等技术的系统上运行，从而更容易被大众接受。

参 考 文 献

1 王常吉，裴定一，蒋文保．一个改进的基于限制性盲签名的电子现金系统．电子学报，2002，29(7)：1083～1085

2 Chaum D. Blind Signature for untraceable payment. In: Advances in Cryptology - CRYPTO'82, 1983. 199～203

3 Chaum D, et al. Untraceable Electronic Cash. In: Advances in Cryptology - CRYPTO '88, 1989. 319～327

4 Manho L, et al. Design and Implementation of Revocable electronic cash system based on Elliptic Curve Discrete Logarithm Problem. Journal of communications and networks, 2002, 4(2): 81～89

（上接第 89 页）

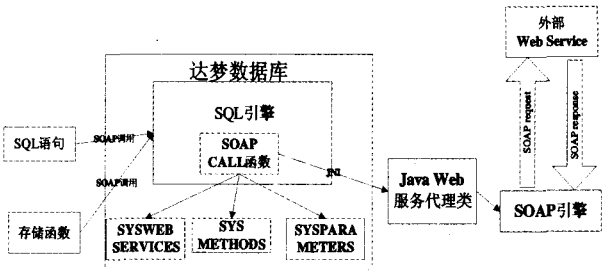

图 3 数据库消费 Web 服务

由于调用外部 Web 服务需要通过网络来传输数据，而网络的好坏和该服务的处理时间直接关系到该事务的处理流程和响应时间。最常用的 HTTP 请求是一种同步请求，在服务器端返回之前，客户端无法继续执行后面的操作。如果我们使用 HTTP 来请求一个外部的 Web 服务是否会影响数据库其它事务的执行呢？

大部分数据库中，每个事务的执行都对应着一个线程。而在达梦数据库中，一个工作线程并不为某个特定的事务服务，当处理事务由于等待而挂起的时候，处理该事务的线程并不会挂起，而是继续到工作队列中查找是否还有用户请求要处理。因此，即使我们使用 HTTP 来请求一个长时间的外部 Web 服务，我们可以使用事件机制来将该事务挂起，而不影响到整个服务器中其它事务的处理。

结论 传统数据库同 Web 应用的隔离造成了要将数据提供给外部访问或者数据库要访问外部数据必须通过编写大量的代码来实现，这一方面造成了大量代码的重复编写，另一方面使得数据库永远只是在后台扮演着数据提供者的角色，而不能直接同外部 Web 应用进行交互。数据库 Web 服务思想的提出，填平了数据库同 Web 应用的鸿沟，使得数据库从后台直接走到了前台，利用 Web 服务成为了面向服务体系结构的重要组成者。本文提出的方法很好地将数据库同 Web 服务技术整合在一起，该方法考虑到在 Web 服务调用中，大部分都是通过 HTTP 协议使用 SOAP-RPC 的形式来使用 Web 服务的，因此专门针对这种类型的调用来实现。另外，为了方便用户在数据库中发布和调用 Web 服务，我们也开发了一些便利的工具来帮助用户使用。例如，通过指定要调用的 Web 服务的 WSDL 文档，自动声明该服务以及该服务下的操作；自动根据用户部署的数据库 Web 服务，生成 WSDL 文档等。

参 考 文 献

1 Hacigumus H, Iyer B, Mehrotra S. Providing Database as a Service. In: 18th International Conference on Data Engineering(ICDE'02)2002 IEEE

2 Simple Object Access Protocal (SOAP). http://www.w3.org/TR/SOAP

3 Universal Description. Discovery and Integration (UDDI). http://www.uddi.org

4 Web Services Activity, W3C Architecture Domain. http://www.w3.org/2002/wsk

5 Web Service Derscription Language (WSDL). http://www.w3.org/TR/wsdl

6 Malaika S, Nelin C J, Qu R. DB2 and Web service. IBM SYSTEMS JOURNAL, 2002, 41(4)

7 Database Web Services. ORACLE White Paper, November 2002

8 Brogden B 著．SOAP 和 Java 编程指南．电子工业出版社

9 IBM Web Services Architecture Tam. Web services architecture overview. The next stage of evolution for e-business. IBM Technical Document, Web Architecture Library, 2000

10 柴晓路，梁宇奇编著．Web Services 技术、架构和应用．电子工业出版社