基于 J2ME 的移动支付安全方案研究*)

许 峰[1,2] 崔 隽[1] 黄 皓[1]
(南京大学计算机科学与技术系 南京 210093)[1] (河海大学计算机及信息工程学院 南京 210098)[2]

摘 要 安全方案对移动支付系统的安全性起着决定性作用,其中无线环境中的安全和对用户即手持设备的认证,更是系统成败的关键。借鉴国外已有的移动支付系统,结合宏支付的特点及安全要求,并考虑到 J2ME 平台本身提供的安全性,提出了一个基于 J2ME 的移动支付安全方案,重点解决无线环境下的用户的认证问题,来保证针对宏支付的移动支付系统的安全。分析测试验证了该安全方案的安全性及可行性。

关键词 移动支付,安全方案,密码体制,J2ME 平台

Research on Security Protocol for Mobile Payment Based on J2ME

XU Feng[1,2] CUI Jun[1] HUANG Hao[1]
(Department of Computer Science and Technology, Nanjing University, Nanjing 210093, China)[1]
(College of Computer & Information Engineering, Hehai University, Nanjing 210098, China)[2]

Abstract The security protocols which are crucial to the mobile payment system (MPS), especially to the authentication of the user (mobile hand-held devices) in the wireless environment, determine the security properties or even the fate of whole system. We investigate the existent mobile payment systems, study the characteristic and security requirement of the macro-payment, and consider the inherent security of the J2ME platform, and hereby propose a security protocol for mobile payment based on J2ME, emphasizing the authentication to the user. Then we analyze and test the security and feasibility of the security protocols and the conclusion we have drawn is that security protocol achieves its security goals, has the ability to resist usual attacks and enjoys high efficiency.

Keywords Mobile payment, Security protocol, Cryptography, J2ME platform

1 引言

移动支付(mobile payment)是在现有技术(如 wireless LAN(IEEE 802.11)、Bluetooth 等)的基础上提出的用手持设备如手机、PDA 等(有时称手持设备)作为一个新的终端进行交易的支付方法。从产生过程来讲,移动支付是支付方式演变的结果,从传统支付到电子支付,再发展到移动支付。电子支付克服了传统支付中用户必须携带现金的缺点;而移动支付有着任何时间、任何地点、任何方式的独特优势,它克服了电子支付在固定网络上支付的缺陷。

移动支付系统按照交易额的数量分为宏支付($10 以上)和微支付($10 以下)[1]。现存的移动支付系统大部分都是微支付,主要提供信息类服务,交易类服务很少,真正使用手持设备进行交易的用户就更少。在这种微支付系统中,交易的费用是从用户的话单中扣除的,不直接涉及到银行的参与,而是采用宏支付系统,用户不需携带信用卡,用手持设备即可在商场购物。在这种系统中,银行是参与者之一,用户的交易费用是直接从与用户手持设备绑定的银行账户中扣除的。从长远来看,使用手持设备进行随时随地交易是一个发展趋势。一些发达国家已经实现宏支付,如日本的 I-Mode,但它是针对日本通信技术设计的,且技术不公开。由于交易额较大,对安全性要求较高,安全性问题是阻碍宏支付发展的主要问题。因此,设计出一种安全的移动支付方案对移动支付特别是宏支付的发展具有重要意义。

本文试图从密码学的角度在应用层为宏支付设计出一种安全方案来解决宏支付的安全问题,重点解决用户即手持设备的认证,以及数据保密性、完整性等问题,最后在 J2ME 平台上实现该安全方案。

2 移动支付系统

2.1 一般的移动支付系统(MPS)框架

一般的 MPS 有前端和后台之分,就像客户-服务器系统一样。系统有两个前端,即商家的前端和客户的前端。客户的前端是运行在手持设备上的软件和应用程序而后台负责处理支付请求和账户处理。在一个简单的 MPS 中,一般有 3 个部分和 MPS 交互:终端用户、商家、FSP(金融服务处理)。图 1 是简单的 MPS 的抽象模型[2]。

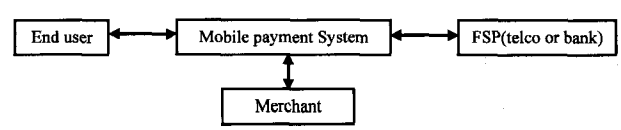

图 1 一般的移动支付系统框架

移动支付过程和基于 Internet 的电子支付过程类似,见表 1。

*)本课题得到国家自然科学基金(60473091)和国家"863"高技术研究发展计划项目基金(2007AA01Z409)资助。许 峰 博士研究生,主要研究方向为信息安全和分布式计算;崔 隽 硕士,主要研究方向为网络安全;黄 皓 教授,博士生导师,主要研究领域为网络安全。

表 1　移动支付过程

Action	Parties
1) Purchase initiation	End-user⇔Merchant
2) User and/or Account identification	End-user ⇔FSP
3) Authorize Payment	FSP ⇔FSP
4) Payment authorized	FSP⇔Merchant
5) Recipient + content	Merchan⇔End- user
6) Payment capture	Merchant ⇔FSP

2.2　现有的 Mobile payment 解决方案分析

随着移动支付技术的发展，目前已经有很多较为成熟的系统，如 Paybox[3]、Simpay[4]、NTT DoCoMo 等系统。从技术角度来看，目前比较有代表性的移动支付系统大致有 4 类：基于 SMS(Short Message Service)的系统、基于 WAP(Wireless Application Protocol)的系统、基于 I-Mode 的系统和基于 J2ME 的系统。其中，基于 SMS 的系统、基于 WAP 的系统适用于微支付系统，基于 I-Mode 的系统适用于宏支付和微支付，是移动支付的成功案例[5]，但是只在日本境内使用且技术不公开，缺乏国际标准。基于 J2ME 的宏支付系统现在还在研究阶段，并且技术不公开，也没有什么标准可循。但是随着 J2ME 的安全性能的逐渐提高、无线信道的加宽以及手持设备的升级，基于 J2ME 的宏支付系统必将是移动支付的发展方向。

2.3　基于 J2ME 的系统

2.3.1　J2ME 的安全性

J2ME(Java 2 Micro Edition)是美国 Sun 公司为小型资源受限终端设备的应用程序开发、提供使用的 JAVA 平台。J2ME 平台分为两个配置(Configuration)：CLDC(Connected Limited Device Configuration)[6]联网的受限设备配置和 CDC(Connected Device Configuration)联网的设备配置。其中 CLDC 是为严格受资源约束的设备而设立的，这种设备如蜂窝电话、PDA 等等，为此它在每个方面都做了优化。而 CDC 是针对机顶盒等这类设备的。

MIDP(Mobile Information Device Profile)[7]移动信息设备配置文件是目前为止可供使用的用于小设备的框架，它遵循了 CLDC 的宗旨，尽可能使用尽量少的资源。MIDlet Suite 把多个 MIDlet 关联到一起。在 J2ME 平台上开发的程序即 MIDlet Suite 打包后下载到支持 MIDP 的真机上即可运行。目前，绝大多数品牌手持设备都支持 MIDP 规范 。MIDP 从 MIDP1.0 发展到 MIDP2.0，安全性能逐渐加强。和 MIDP1.0 相比，MIDP2.0 开始支持 https 从而保护传输层的安全，大大加强了对用户界面、多媒体和游戏功能、网络连接功能的支持，同时将 OTA 应用程序下载包括到规范中来，另外还为无线信息设备提供了端到端(end-to-end)的安全机制，MIDP2.0 还支持服务器 Push 体系架构[8]。

2.3.2　J2ME 适合移动支付系统特别是宏支付的开发

WAP、I-Mode 等都是基于微浏览器的。微浏览器架构过分依赖在服务器和手持设备端之间传递数据的网络，如果网络出现故障或暂时瘫痪，会不可避免地对移动互联产生毁灭性影响，而且微浏览器架构所不具备的高交互性和安全性也成为其走向企业领域的软肋。而 J2ME 为移动互联引入了一种新的模型，即允许手持设备可以从互联网上下载各种应用程序，并在手持设备创造可执行环境离线运行这些程序。同时定义了可执行程序下载的标准，并在手持设备上创立了可执行环境和程序开发语言。

除了上面讲的 J2ME 的安全性，J2ME 的可移植性、改善了的 UI 用户体验、更低的网络资源消耗与服务器负载、MIDlet 中的动态事件处理、事务保护、密码术等[9]特点都适合于移动支付特别是宏支付系统的开发。

3　安全方案的设计

3.1　移动支付系统模型

该系统的总体框架如图 2。该系统由用户(手持设备)Client、商家 Merchant、移动支付平台(MPP)、银行端处理设备(Settlement)组成，这里移动运营商起到传媒作用，为了简化系统，不作为移动支付的组成部分。

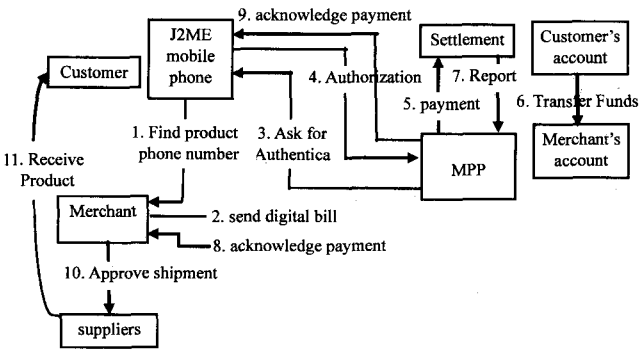

图 2　移动支付系统模型

整个交易过程如下：

① Client 挑选商品后，由商家的服务人员录入所买商品的详细信息，按固定格式形成 Order。选择完毕后，告诉商家手持设备 ID(如手机号)。

② 商家对该 Order 和手持设备 ID(如手机号)加密、签名后通过安全 Internet 通道(如 SSL)发送给 MPP。

③ MPP 收到消息后确认消息的来源。如果消息确实来自指定商家，则对消息处理(如加密签名)后发送给移动用户，即 Client。

④ Client 收到 welcome 消息后输入 PIN 码，同意使用移动支付系统，然后确认所买的商品、消费额、商家标示及消息来源。如果消息正确，则同意支付。消息处理后传送给 MPP。

⑤ MPP 确认消息正确后向银行发起转账请求。

⑥ 银行处理支付。

⑦ MPP 收到转账成功的消息。

⑧ 商家收到支付成功的通知。

⑨ Client 收到电子发票或收据。

⑩ 商家为客户提供服务。

其中③、④两步是手持设备和支付平台间的无线环境下的通信，并且必须保证客户对此次交易支付的确认信息的安全性。移动支付平台对商家的认证也很重要，以防假冒商家。但这是在基于 Internet 的有线环境下，因此很容易做到。

从上面的移动支付系统我们可以看出，对移动支付系统的安全威胁主要存在以下两个方面：①冒充用户即手持设备。用户 A 在选购商品完毕后，冒充用户 B，希望商品的费用从用户 B 在银行的账户中支付；②冒充商家。商家 B 冒充商家 A，希望用户支付给商家 A 的费用转到自己的账户中。而实施这些威胁所采用手段一般是窃听、重放等方法。

3.2　安全方案描述

(1)商家→MPPI：$\{SHA(Order)_{Smerchnant}\}\backslash\backslash Order\backslash\backslash ID\backslash\backslash$

N_{RAND}

ID 为手持设备的 ID(如手机号)。NRAND 为大的随机数,作用是为抵制重放攻击。SHA 函数可以采用杂凑码,以防止黑客用字典进行穷举法的进攻。消息 I 通过安全通道由商家发送到 MPP。因为商家和 MPP 之间是基于 Internet 的有线通道,所以保证消息 I 的安全性是不成问题的。其中 Order 中应包含商家信息、用户的购物单、用户的 ID 等信息,MPP 会检查 Order 中商家和签名的商家是否一致。

(2)MPP → Client

首先 MIDP2.0 规范下 MPP 和 Client 之间通过 https 保证端到端的安全[10](详见 MIDP2.0 规范)。

MPP 收到消息 I 后验证商家的签名来判断商家的身份并验证数据 Order 的完整性。如果正确,则提取出 ID 和 NRAND,否则返回错误信息给商家。接着从本地数据库中检索 ID 来验证 ID 的合法性。如合法,则运行 InitServlet,产生会话密钥(SessionKey)和随机数 Challenge。会话密钥有两个:一个用于 MPP 端加密而 Client 解密,一个用于 Client 加密而 MPP 端解密。同时将用 PIN 用自己填充过的密钥加密过 SessionKey 后伴随着 Welcome 界面发送给 ID 用户并要求用户输入 6 位 PIN 码,同意使用支付系统,同时存储 NRAND。

II:$\{[SHA(Order\backslash\backslash challenge)]_{SMPP}\backslash\backslash Order\}_{SessionKey}$

III:$(SessionKey)_{PIN+PIN}$ 此加密采用 128bit AES

其中 InitServlet 的功能包括以下几个方面:

产生 128bit 的随机数 challenge;

为客户创建新的 http session;

存储 challenge 到会话变量里;

产生 128bit 的加密/解密会话密钥(SessionKey);

存储(d)产生的会话密钥到数据库 client 指定条目里;

从数据库检索 client 的 PIN 码;

用 PIN 码自己填充后的结果来加密会话密钥;

将(g)的结果伴随着 Welcome 界面发送给 ID 用户并要求用户输入 6 位 PIN 码同意使用支付系统。

尽管每个会话中会话密钥的不同能保证每次加密后认证数据的密文不同,从而防止重放的攻击,但有的用户习惯于长期使用相同的会话密钥而不愿在每次建立会话时都改变它。在这种情况下,challenge 就很有用。

InitServlet 产生两个会话密钥用于单向加密。这样做的好处是在不增加加密算法运算负担的情况下防止密钥被窃取,从而增加安全性。这种方法弥补了对称加密的不足和弱点。对称加密的算法有很多种,如 DES、3DES、AES 等,在这里我们选用安全性能比较高的 AES[11] 算法作为对称加密的算法。

(3)client→ MPP

Client 输入 PIN 码后通过解密 III 得到 SessionKey,用 SessionKey 初始化 AES,解密 II 得到 $\{[SHA(Order\backslash\backslash challenge)]_{SMPP}\backslash\backslash Order\}$ 并验证 MPP 的签名及 Order 的数据完整性以及是否来自预定的商家。

如正确,则确认消息确实来自于移动支付平台。用户检查 Order 后,如同意支付,则生成 Confirm 确认信息后将其签名并附上 challenge、ID 和 PIN 信息用会话密钥加密后发送给移动支付平台;如 Client 输入 PIN 码不正确或得到消息不完整则返回相应信息。

Client 向 MPP 发送消息如下:

IV:$\{[SHA(Confirm)]_{Sclient}\backslash\backslash confirm\backslash\backslash challenge+1\backslash\backslash ID\backslash\backslash PIN\}_{SessionKey}$ 其中 ID 和 PIN 码是为了下一步的进一步验证;challenge+1 是为了防止重放攻击。

(4)MPP→Settlement

MPP 收到 IV 后进入认证阶段。这是客户端认证的关键。

① 首先从数据库中取出用于 IV 的解密 SessionKey 来初始化 AES,解密 IV 得到 $\{[SHA(Confirm)]_{Sclient}\backslash\backslash confirm\backslash\backslash challenge+1\backslash\backslash ID\backslash\backslash PIN\}$。如果解密不成功,则返回 client 相应的错误信息。

② 取 Client 的公钥验证签名的有效性。如无效,则返回 client 相应的错误信息。

③ 提取 confirm 进行 SHA 算法,用结果和收到的摘要进行比较,验证数据(confirm)完整性。如相符,则返回 client 相应的错误信息。

④ 提取出 ID\\Challenge+1\\PIN。依次和数据库取出 PIN 和 ID 比较,验证 PIN 和 ID 的一致性,同时验证 client 发来的 challenge 和会话变量里存储的 challenge 的一致性。

⑤ 此时根据签名的有效性可以判断 Conform 确实来自指定的客户端,同时根据 SHA 结果可以判断 Conform 在传输过程中没有被篡改过。再次验证 challenge 来判断是不是重放。而 PIN 和 ID 起到再次验证的作用。

MPP 认证 Client 后,要向 Settlement 提出支付处理请求。因为这个过程是在有线环境下完成,所以复杂的密码算法不会在很大程度上影响速度。

V:$\{SHA(Request)\}_{SMPP}\backslash\backslash Request\backslash\backslash Nrand$

MPP 把消息 V 通过安全通道发送给 Settlement。其中 Request 应包含客户 ID、商家 ID、转账金额、交易代码、交易时间戳等等信息。

(5)Settlement → MPP

Settlement 收到消息 V 后:

① 验证 MPP 的签名的有效性。如无效,则返回 PMF 相应的出错信息。

② 用 SHA 散列 Request 后将散列结果和收到的摘要进行比较。如不一致,则返回 PMF 相应的出错信息。

③ 通过 Request 里的交易时间戳判断是否是重放。

经过上面的验证可知:消息 V 确实来自指定的 MPP 且不是重放,这样可以防止假冒 MPP 的情况。

Settlement 根据 Request 提供的信息进行支付的处理。如处理成功,则返回转账成功的消息给 MPP,否则返回 PMP 相应的出错信息。

VI:

$\{SHA(ResMPP)\}_{Sbank}\backslash\backslash ResMPP\backslash\backslash Nrand+1\backslash\backslash$

$\{\{SHA(ResMerchant)\}_{Sbank}\backslash\backslash\backslash ResMerchant\}_{Pmerchant}\backslash\backslash$

$\{\{SHA(Resclient)\}_{Sbank}\backslash\backslash Resclient\}_{Pclient}$

Settlement 通过安全通道把消息 VI 发送给 MPP。其中 ResMPP 应包含客户 ID、商家 ID、账户处理结果、交易代码、交易时间戳等等信息,并用银行的私钥签名。ResMerchant 是银行给 merchant 的收据,用银行的私钥签名后用 merchant 的公钥加密。Resclient 是银行给 client 的收据,用银行的私钥签名后用 client 的公钥加密。这样对 MPP 来讲,MPP 只知道支付成功了而看不到 merchant 和 client 的帐户信息,其中 $\{\{SHA(ResMerchant)\}_{Sbank}\backslash\backslash ResMerchant\}_{Pmerchant}$ 部分只有 merchant 用它的私钥才能打开来验证银行的签名,而

$\{\{SHA(Resclient)\}_{Sbank}\backslash\backslash Resclient\}_{Pclient}$ 部分只有 client 用它的私钥才能打开来验证银行的签名。这样做的目的是：①对 client 和 merchant 和商家来说他们的钱存在银行里，也就是说他们信任银行，只有发给他们银行的收据才说明支付成功。②银行签发的收据是支付的凭证，支付过程中出现什么问题，可以用这个收据来追溯责任甚至法律责任。

Settlement 应存储近几笔交易已备客户查询。

(6)MPP→ Merchant　MPP→ Client

MPP 收到 VII 后：

① 验证 Settlement 的签名的有效性。如无效，则返回 Settlement 相应出错信息。

② 用 SHA 散列 Response 后将散列结果和收到的摘要进行比较。如不一致，则返回 Settlement 相应的出错信息。

③ 通过收到的(Nrand+1)与 Nrand 的差是否为 1 来判断是否是重放。

通过上述 3 步可以确定消息确实来自 Settlement，说明支付已经成功。此时 MPP 需要通知商家和用户，采用的方式是收据的形式。

MPP→Merchant

VII:$\{\{SHA(ResMerchant)\}_{Sbank}\backslash\backslash ResMerchant\}_{Pmerchant}$

ResMerchant 应包含 Merchant 的帐户信息和账户处理信息。MPP 把从 Settlement 收到的消息 VI 中提取出 VII 后通过安全通道发送给 Merchant。

MPP→Client

VIII:$\{\{SHA(Resclient)\}_{Sbank}\backslash\backslash Resclient\}_{Pclient}$

Resclient 应包含 client 帐户信息和账户处理信息。MPP 从消息 VI 中提取出 VIII 后用 MPP 和 client 之间的 SessionKey 加密后发给 client，即

VIIII:$\{\{\{SHA(Resclient)\}_{Sbank}\backslash\backslash Resclient\}_{Pclient}\}_{SessionKey}$

(7) Merchant 收到 VII 后用私钥解密后，验证银行签名，查看收据并存储收据。Client 收到 VIIII 后，用会话密钥解密，用私钥解密，验证银行签名，查看收据并存储收据。

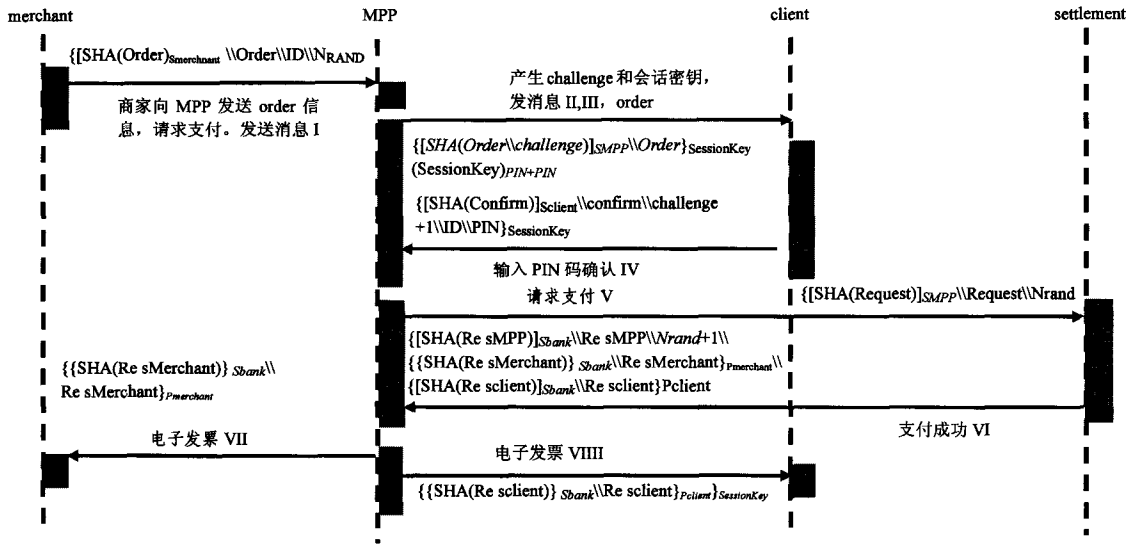

图 3　安全方案时序图

由于 MIDP 只支持 JDK 中的有限个基本包，包括 java.io、java.lang、java.util，也不支持浮点运算。一些在 J2SE 平台下运行的加密算法在 J2ME 平台下由于运算能力、存储能力的限制就不能被识别和运行。本系统采用 Bouncy castle 开发的一种开源的轻量级加密包[10]来完成一系列的加密算法。Bouncy Castle 提供了可以在 J2ME 平台得到支持的 API。

4　系统实现和测试

4.1　系统框架

系统分为 Client 端和 Server 端。在 Client 端是 MIDlets 以及用于加密的类，而在 Server 端是 Servlets 和加密类。如图 4 所示。

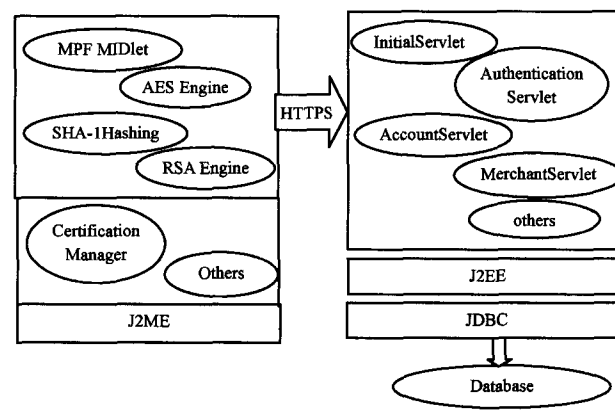

图 4　系统框架

Client 端：在 Client 端主要有 MPPMIDlet、AESEngine、RSAEngine、CertificationManager 等等。其中 MPPMIDlet 是手持设备与后台 J2EE 服务器交互的重要类，也是唯一的类。MPPMIDlet 是个 MIDlet，它负责用户界面的交互，同时调用其它的类，如 AESEngine、RSAEngine、CertificationManager 等等来完成安全算法；AESEngine 主要完成 AES 加解密操作，在系统中使用 BouncyCastle 的轻量级加密算法；同样 RSAEngine 主要完成用户签名操作；CertificationManager 主要完成用户端证书的管理如取证书或证书的 URL 等，同时在 Client 端还有其它的类。

Server 端：在 Server 端主要是 Servlet。每一个 Servlet 代表一种服务，如 AuthenticationServlet 主要完成手持设备端的

(下转第 121 页)

统平台中具有重要的作用。

新型电力网格平台 PGP 的设计和实现是江苏省软件和集成电路业专项经费项目《基于网格技术的调度自动化集成系统》的一个重要部分，目前我们已经完成了这个平台系统的开发工作。本文的研究是该平台服务和信息管理层的核心工作，主要提出了一种对传统树型目录进行改进的基于 TAG 的信息分类机制，使得用户检索具有层次和非层次关系的信息资源都能获得使用上的顺滑感和满意的结果。在此基础上，设计了一种分布式的资源存储和检索机制来实现这种基于 TAG 的分类，在兼顾性能的同时提高了整个平台资源管理服务的可靠性，使整个 PGP 平台能够很好地满足电力系统资源共享和信息集成的要求，对解决大规模电力系统的全网统一调度和分析计算问题具有推动作用。

今后的工作主要集中在两方面：一方面是在电力网格平台上开发以分布式潮流计算为代表的各种电力应用；另一方面是进一步丰富资源管理功能，实现平台级的任务调度、代码切片和计算协同。

参 考 文 献

[1] 徐志伟，冯百明，李伟. 网格计算技术. 北京：电子工业出版社，2004

[2] Foster I, Kesselman C. The Grid: Bluepoint for a Future Computing Infrastructure. San Francisco: Morgan Kaufmann Publishers, 1999

[3] 辛耀中. 电力信息化几个问题的探讨. 电力信息化，2003，1(3)：20-23

[4] 赵遵廉. 中国电网的发展与展望. 中国电力，2004，37(1)：1-6

[5] 张伟，沈沉，卢强. 电力网格体系初探(二)：电力网格体系结构. 电力系统自动化，2004，28(23)：1-5

[6] Draft IEC 61970. Energy Management System Application Program Interface (EMS-API), Part 301. Common Information Model(CIM)-Base, 1970

[7] Foster I, Kesselman C. Globus: A Metacomputing Infrastructure Toolkit. International Journal of Supercomputer Applications, 1997, 11(2): 115-128

[8] Howes T A, Smith M C. LDAP Programming Directory Access Protocol. Macmillan Technical Publishing, 1997

[9] Foster I, von Laszewski G. Usage of LDAP in Globus. http://www.globus.org/research/papers.html

[10] Simple Object Access Protocol (SOAP) 1.1. W3C, Note 8, 2000

[11] Tim O'Reilly. 什么是 Web2.0[J]. 互联网周刊，2005，40：38-40

[12] Tag-Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Tags

[13] 张伟，沈沉，陈颖，等. 电力网格体系初探(三)：原型系统的设计与实现. 电力系统自动化，2004，28(24)：5-8

(上接第 97 页)

认证功能；AccountServlet 主要是模拟银行完成帐户的转帐功能；而 MerchantServlet 则是模拟商家向用户提交订单；InitialServlet 完成会话密钥的生成和随机数的生成。在 Server 端还有其它的类，如 AES、RSA、SHA-1 等等。

4.2 系统测试及结果

根据前面提出的安全方案，我们在 sun one studio mobile edition 5 上实现了使用该安全方案的移动支付系统，并在该系统上用 DefaultColorPhone 作为模拟器进行测试。系统采用的数据库是当前流行的 Oracle 9i，通过 JDBC-ODBC 桥连接数据库。可以看出，该安全方案是完全可以实现的；系统所采用的轻量级密码算法基本不会影响到系统的速度。此外，系统基本上满足了秘密性、完整性、可认证性、不可否认性、可用性等要求，并且具备常用攻击如重放性攻击、中间人攻击、身份欺骗、窃听、数据篡改等)的防御能力。

系统的优点：①与信用卡有关的敏感信息不用在有线和无线通道上传输，保证了信用卡信息的安全性。②会话密钥的解密过程在手持设备上进行，运算出的结果用 PIN 码加密后存放到手持设备上，即使手持设备被偷窃，该信息也不会泄露。③系统采用轻量级的密码算法，提高了系统的运行速度。④该系统具有很强的移植性，凡是支持 MIDP2.0 的手持设备把该系统下载到本机上即可运行。

结束语 移动支付在技术上涉及到多方面的安全性，尤其是无线安全，并且没有现存的基于 J2ME 的系统可参考。本文充分利用开放的安全算法源代码资源，在比较国外已有的移动支付系统的基础上，结合宏支付的特点及安全要求，并考虑到 J2ME 平台本身提供的安全性，精心选择了一套包括分组密码、公钥密码、Hash 函数和数字签名的密码算法。在此基础上提出了一个基于 J2ME 的移动支付安全方案，来保证针对宏支付的移动支付系统的安全，重点解决无线环境下的用户即手持设备端的认证问题。经分析及测试，该安全方案实现了其安全目标，达到了认证移动设备的目的，能抵抗目前已知的攻击，且运行性能较优。

参 考 文 献

[1] mobile payment forum, Ltd. mobile payment forum white paper. http://www.mobilepaymentforum.org

[2] Ashley P, Hinton H, Vandenwauver M. Wired versus Wireless Security: the Internet, WAP and iMode for E-Commerce. http://www.acsac.org/2001/abstracts/ the-1030-b-ashley.html, 2001.12

[3] Cervera A. Analysis of J2ME™ for developing Mobile Payment Systems. Master's Thesis. IT University of Copenhagen, 2002.8

[4] 周展飞，周典萃，王贵林，等. 电子商务协议的公平性. 电子学报，2000，28(9)：13-15

[5] 田建波，王育民. 一种改进的认证逻辑. 电子学报，1998，26(7)

[6] Forum Nokia. What's in MIDP 2.0: A Guide for Java™ Developers, Version 1.0. http://www.forum.nokia.com, 2003.9

[7] Forum Nokia. MIDP 2.0 Introduction, Version 1.0, 2003.3

[8] lcrypto-j2me-127.tar.gz. http://www.bouncycastle.org/latest_releases.html. United Arab Emirates, paybox features world of mobile commerce in Dubai. http://www.ameinfo.com/news/Detailed/46255.html

[9] http://www.chinaepayments.com/system/shownews.asp?id=390

[10] Daemen J, Rijmen V. AES Proposal: Rijndael. AES Algorithm Submission, National Institute of Standards and Technology (NIST), 1999

[11] Housley R, Ford W. Internet X.509 Public Key Infrastructure Certificate and CRL Profile. IETF RFC 2459. http://www.faqs.org/rfcs/rfc2459.html, 1999.1

[12] Freudenthal M, Heiberg S, Willemson J. Personal Security Environment on Palm PDA// IEEE Proceedings, 2000. http://citeseer.nj.nec.com/freudenthal00personal.html