计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.181202414

利用基于身份的密码算法+短信验证码的移动安全支付方案

刘亚强　李晓宇
郑州大学信息工程学院　郑州 450001
(zzulyq@126.com)

摘　要　针对移动支付过程中短信验证码被盗导致资金失窃,以及在基于证书的密码体制下建立移动支付系统时移动设备和移动网络面临巨大压力的问题,文中提出了利用基于身份的密码算法+短信验证码的移动安全支付方案。该方案中,用户和银行服务器加入一个基于身份的密码系统,它们不再需要基于数字证书的身份认证,这将大大减小移动设备以及移动网络的存储和计算开销。用户首先到银行柜台注册开通手机银行业务,设置用户名、密码,预留安全问题,在银行工作人员的帮助下完成手机银行APP的首次安装和初始化。登录时,银行服务器对用户进行身份认证,保证用户合法。支付时,手机银行APP利用用户的私钥生成对短信验证码的数字签名,并用银行服务器的公钥对数字签名和短信验证码的组合加密后发送给银行服务器以进行验证,只有银行服务器验证通过后才允许用户支付。在本方案中,短信验证码和数字签名将共同为用户提供安全保证,即使验证码泄露,攻击者也不可能根据验证码生成数字签名,从而保证了移动支付的安全。理论分析和实验结果表明,本方案不但能够大大提高移动支付的安全性,而且随着移动终端的增加,系统的平均响应时间也不会急剧增长,因此所提方案具有较好的健壮性和可行性。

关键词:移动支付;短信验证码;基于身份的密码算法;支付安全;数字签名
中图法分类号　TP399

Mobile Secure Payment Scheme Using Identity-based Cryptographic Algorithm+SMS Verification Code

LIU Ya-qiang and LI Xiao-yu
School of Information Engineering,Zhengzhou University,Zhengzhou 450001,China

Abstract　Aiming at the problem of stolen funds caused by stolen SMS verification code in mobile payment process,as well as the mobile device and the mobile network are under great pressure when establishing a mobile payment system under the certificate-based cryptosystem,a mobile secure payment scheme based on identity-based cryptographic algorithm+SMS verification code was proposed. In this scheme,users and bank servers join an identity-based cryptosystem,so they no longer need digital certificate-based identity authentication,which will greatly reduce the storage and computational overhead of mobile devices and mobile networks. Users need to go to the bank counter to register and open mobile banking services,set the user name,password and reserved security issues,and complete the first installation and initialization of mobile banking APP with the help of bank staff. When logging in,the bank server authenticates the user's identity to ensure that the user is legal. In payment,the user's private key is used to generate the digital signature of SMS verification code,and the combination of digital signature and SMS verification code is encrypted with the bank server's public key and sent to the bank server for verification,the bank server will not allow the user to pay until the verification is passed. In this scheme,the SMS verification code and the digital signature will jointly provide security guarantee for the user. Even if the verification code is leaked,the attacker cannot generate a digital signature according to the verification code,thus ensuring the security of the mobile payment. Theoretical analysis and experimental results show that this scheme not only can greatly improve the security of mobile payment,but also the average response time of the system will not increase sharply with the increase of mobile terminals,so it has better robustness and feasibility.

Keywords　Mobile payment,SMS verification code,Identity-based cryptographic algorithm,Payment security,Digital signature

到稿日期:2018-12-25　返修日期:2019-04-30　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金项目(61472412);河南省教育厅自然科学基金项目(14A520012)
This work was supported by the National Natural Science Foundation of China (61472412) and National Natural Science Foundation of Henan Educational Committee (14A520012).
通信作者:李晓宇(iexyli@zzu.edu.cn)

1 引言

随着互联网技术和电子商务的快速发展，移动支付已经成为了人们主流的支付方式。截至2018年6月，中国网民总数已经达到了8.02亿，其中手机网民的人数达到了7.88亿，中国已经成为了互联网大国[1]。在我国手机网民中，有高达71.9%的用户采用移动支付的方式进行日常消费，中国市场已经成为了最大的移动支付市场。

移动支付[2]，又称为手机支付，指交易双方在手机上使用移动网络通信实现的商业交易，它具有移动性、及时性、定制化和集成性的优点。移动支付作为一种支付方式，其最重要的就是移动支付方案的安全性问题[3-4]。目前，大多数安全性强的移动支付方案都是采用双因素认证模型[5-6]，“口令+短信验证码”是双因素认证模型中成本最低、最易实施的验证方式。第42次《中国互联网络发展状况统计报告》统计数据显示，我国网民通过手机接入互联网的比例高达98.3%，因此，采用手机短信验证码的验证方式具有用户绑定性强、学习成本低和用户广泛拥有的优势。但是，目前采用短信验证码进行二次身份信息认证的移动支付方案仍存在很大的安全问题[7]，即短信验证码容易被攻击者窃取，从而导致用户遭受财产损失。另一方面，目前传统的移动支付方案都是建立在基于证书的公钥密码体制的基础上，但是将基于证书的公钥密码应用在移动网络环境中时会出现很多问题，例如，移动设备需要大量的存储空间来存储用户的数字证书，并且在用户通信时还需要大量的计算来验证数字证书。随着移动网络用户的增加，移动设备和无线网络必将面临巨大的压力[8]。因此，在基于证书的公钥密码体制上设计移动支付方案显然存在着弊端。

针对以上问题，本文提出了一种利用基于身份的密码算法+短信验证码的移动安全支付方案。本方案在基于身份的密码算法下建立系统，由银行服务器生成系统主密钥、系统公开参数和用户的公私钥，其中，用户的公钥可由用户的身份和公开算法计算得到，因此本方案不再需要基于数字证书的身份认证，这将减小移动设备和移动网络所面临的压力。在支付过程中，手机银行APP利用用户的私钥对接收到的短信验证码生成数字签名，然后用银行服务器的公钥将数字签名和短信验证码的组合加密后发送给银行服务器进行验证。短信验证码和数字签名将共同为本方案提供安全保证，即使验证码泄露，攻击者也不可能生成对验证码的数字签名，从而保证了移动支付过程的安全性和用户的资金财产安全。

2 相关知识

2.1 基于身份的密码算法

1984年，Shamir提出了基于身份的密码学(IBC)的思想[9]，该思想将用户的公钥和用户的身份直接联系起来，即将用户的身份信息作为用户的公钥。因此，在基于身份的公钥密码体系中不再需要数字证书，这将节省移动设备的存储空间，减小移动设备的计算开销。同时，系统不再需要证书授权机构等所需的一系列的设施，这也将为移动支付系统的建立节省开支。在基于身份的公钥密码体系中[10]，椭圆曲线密码体制ECC具有巨大的优势[11-12]。ECC中160 bits的密钥所具有的安全性与RSA中1024 bits的密钥所具有的安全性相当，这意味着在同等安全性要求的情况下，ECC所需的计算量、存储空间、数据流量都更小[13-14]，也就意味着用户能够以更低的成本获得更好的用户体验，这对移动支付是非常有利的。因此，本文将采用椭圆曲线密码体制设计移动支付方案[15-16]。

在本文提出的基于身份的密码算法中，由银行服务器充当密钥分发中心KGC，其中基于身份的加密方案由系统建立、私钥提取、加密和解密4个功能模块组成。

设E是有限域F_q上的椭圆曲线，G_1和G_2分别是$E(F_q)$中阶为n的加法群和乘法群，P是G_1的一个基点，e为$G_1\times G_1\rightarrow G_2$的双线性对，定义$H_1$和$H_2$为两个单向强碰撞攻击的Hash函数：$H_1:\{0,1\}^*\rightarrow G_1$；$H_2:\{0,1\}^*\rightarrow Z_q^*$。

1)系统建立(Setup)：银行服务器随机选取一个秘密值$SK\in Z_q^*$作为系统私钥。计算系统公钥的公式为：

$$SPK=SK\times P$$

密钥分发中心向外公开参数：

$$params=(G_1,G_2,q,P,SPK,e,H_1,H_2)$$

2)私钥提取(Extract)：给定用户ID，通过手机银行APP将用户ID发送给银行服务器，银行服务器接收并保存ID。银行服务器收到用户ID后计算用户私钥：

$$S_{ID}=SK\times H_1(ID,T)$$

并将(S_{ID},T)安全地发送给手机银行APP，其中T为银行服务器获取的当前时间戳。手机银行APP接收到(S_{ID},T)后，计算：

$$CPK=H_1(ID,T)$$

$$e(S_{ID},P)=e(CPK,SPK) \tag{1}$$

当且仅当式(1)成立时，手机银行APP判定银行服务器生成的用户私钥S_{ID}合法，其中CPK为用户公钥。

3)加密(Encrypt)：对于任意的明文数据m，用户随机选取$r\in Z_q$，计算点：

$$(x_1,y_1)=rP$$

$$(x_2,y_2)=rSPK$$

若$x_2=0$，则重新选取r值进行计算；若$x_2\neq 0$，计算密文：

$$C=m\times x_2$$

手机银行APP发送加密数据(x_1,y_1,C)给银行服务器。

4)解密(Decrypt)：银行服务器接收到密文(x_1,y_1,C)后，计算：

$$(x_2,y_2)=SK(x_1,y_1)$$

从而计算出在F_q中x_2^{-1}的值，然后通过计算$m=Cx_2^{-1}$恢复出明文数据m。

本方案采用的基于身份的数字签名方案由系统建立、私钥提取、签名生成和签名验证4个功能模块组成，涉及手机银行APP、银行服务器两个实体。其中，系统建立与私钥提取功能模块与加密方案的功能模块相同，不再详细描述，下面对

签名生成和签名验证过程进行详细描述。

1）签名生成（Sign）：对任意的消息 m，手机银行 APP 随机选取$Y\in Z_q$，并计算：

$U=YP$

$h=H_2(ID,m,U)$

$V=S_{ID}+hrSPK$

从而得到消息 m 的数字签名(V,U)。

2）签名验证（Verify）：银行服务器接收到数字签名(V,U)后，计算：

$CPK=H_1(ID,T)$

$h=H_2(ID,m,U)$

当且仅当等式 $e(P,V)=e(SPK,CPK+hU)$成立时，银行服务器判定数字签名有效。

2.2 可信执行环境

可信执行环境[17-19]（Trusted Execution Enviroment，TEE）是 Global Platform（GP）提出的安全概念。它是由处理器、内存和存储功能组成的一种安全的、完整性保护的处理环境。如今，几乎所有的智能手机和平板电脑都包含基于硬件的可信执行环境（TEE）。

TEE 的架构分为两大部分：1）REE（Rich Execution Environment，一般指通用操作系统）端接口，包含 TEE 功能性接口和 TEE 客户端接口，主要供一般用户端应用软件接入 TEE 环境使用；2）TEE 端的可信应用软件、可信操作系统与硬件外围设备。REE 与 TEE 相隔离，REE 端应用程序可通过 TEE 功能性接口或 TEE 客户端接口与可信应用程序互通，以存取 TEE 的特定信息与交换信息。TEE 的层次架构如图 1 所示。

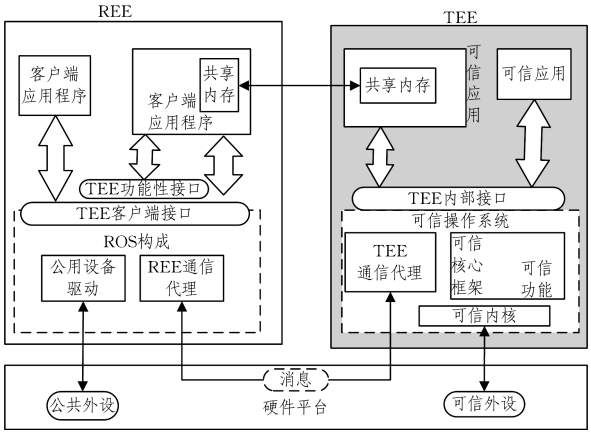


图 1 TEE 层次结构图

Fig. 1 TEE hierarchy diagram

TEE 技术能够为手机提供 3 个基本的安全功能。

1）硬件隔离的安全处理环境：TEE 提供了基于硬件隔离的安全系统，以保证敏感数据的安全和程序的正确执行。

2）平台完整性：为了保证整个系统的安全，TEE 设备加电启动开始逐步验证以保证 TEE 平台的完整性。

3）安全存储：TEE 支持 DES，AES-256，RSA 等加密算法，并且能够依靠部署在硬件中的根密钥、加密算法和完整性保护技术来加密保护用户的敏感信息。

3 利用基于身份的密码算法＋短信验证码的移动安全支付方案

在传统的基于证书的公钥密码体制中，用户的公钥和其身份并没有直接的联系，为了防止攻击者伪造、篡改和替换用户的公钥，必须由证书授权中心颁发的数字证书来建立用户的公钥与其身份之间的联系，并且需要在用户双方通信时利用数字证书进行身份认证。而本方案在基于身份公钥密码体制上建立的，用户的公钥可以直接由身份 ID 和公共参数计算得到，因此银行服务器与手机银行 APP 通信时不再需要基于数字证书的身份认证。

方案一共分为两个阶段。

1）注册阶段。用户在银行柜台注册开通手机银行业务并安装手机银行 APP。在这个阶段中，用户首先签署开通手机银行协议，然后在银行工作人员的帮助下设置用户名、密码，预留信息验证问题，完成手机银行 APP 的首次安装。在首次安装过程中，手机银行 APP 向用户请求其身份 ID 并发送给银行服务器，银行服务器接收用户身份 ID 并验证，在确认用户为首次安装后，通过手机银行 APP 向用户请求其银行卡的预留手机号，然后通过短信验证码的方式对用户进行身份验证。验证通过后，银行服务器根据用户身份 ID 生成用户私钥。为保证安全传输用户私钥，本方案将在银行服务器中配置 AES-256 密码算法，并利用该算法通过密钥协商的方式将私钥安全地发送给手机银行 APP。手机银行 APP 收到银行服务器生成的私钥后会对其进行验证，验证通过后，手机银行 APP 调用手机 TEE 中的 AES-256 加密算法将私钥加密后进行安全存储，完成手机银行 APP 的首次安装。当用户启动手机银行 APP 时，APP 调用 TEE 中 AES-256 解密算法解密私钥，并将解密后的私钥加载至内存。在移动支付系统运行期间，用户私钥和银行服务器私钥都会定期进行更新。

2）支付阶段。用户须登录手机银行 APP 后才能进行支付操作。在登录过程中，用户输入用户名和密码并点击登录后，手机银行 APP 会利用当前时间戳和用户的登录信息生成身份认证信息，将身份认证信息用银行服务器的公钥加密后发送给银行服务器进行验证，银行服务器验证通过后确认用户合法，然后允许其登录并与手机银行 APP 建立安全通道。建立安全通道后，手机银行 APP 会根据用户支付操作生成支付请求，并用银行服务器公钥将其加密后发送给银行服务器，银行服务器验证通过后会发送一个短信验证码到用户的手机，用户将短信验证码输入到手机银行 APP 后，手机银行 APP 会利用用户的私钥生成验证码的数字签名，然后用银行服务器公钥将数字签名和短信验证码加密后发送给银行服务器进行验证，验证通过后允许用户支付。

用户并非首次安装手机银行 APP 时，会进入重装过程：当用户重装 APP 或在手机上安装手机银行 APP 时，手机银行 APP 向用户请求其身份 ID 并发送给银行服务器，银行服务器收到用户身份 ID 并确认为重装手机银行 APP 时，银行服务器会根据用户是否是在新的手机上安装手机银行 APP 决定是重新为用户生成私钥还是使用原来的私钥，然后将私钥安全地发送给手机银行 APP，手机银行 APP 验证通过后

将私钥加密保存在程序安装目录中，完成 APP 的重新安装。本方案中使用的符号定义如表 1 所列。

表 1 相关符号描述

Table 1 Description of related symbols

符号	相关描述
S_pu	银行服务器公钥
S_pr	银行服务器私钥
C_pu	用户公钥
C_pr	用户私钥
Sk_i	AES 对称密钥
M	登录信息(用户名、密码)
REQ	支付请求
$Answer$	预留安全问题答案
Str	短信验证码
$List\langle ID,DI\rangle$	安全设备列表
DI	设备信息(IMEI,ICCID)
$Data_i$	第 i 个加密数据
$params$	系统公开参数
$E\{K,P\}$	使用密钥 K 对数据 P 加密
$D\{K,P\}$	使用密钥 K 对数据 P 解密
$S\{K,P\}$	使用密钥 K 生成对数据 P 的数字签名
$V\{K,Str,P,params\}$	利用密钥 K,Str,$params$ 验证数字签名 P

3.1 注册

用户携带有效身份证件和银行卡到银行柜台，告知银行工作人员要注册开通手机银行业务。银行工作人员在核实了用户的身份证件和银行卡信息后，打印开通手机银行协议并让用户签字。用户签字后，在银行工作人员的帮助下设置用户名、登录密码，预留信息验证问题，并同步存储在银行服务器的数据库上。银行服务器同时在数据库中为该用户账户建立一个列表 $List\langle ID,DI\rangle$，以存储用户的安全设备信息。最后，银行工作人员将业务办理单据交予用户，注册结束；用户在银行工作人员的帮助下进行手机银行 APP 的首次安装。

在首次安装过程中，手机银行 APP 会获取手机的设备信息 DI^* 并弹出对话框提示用户输入身份 ID，DI^* 包括国际移动设备识别码 IMEI 和 SIM 卡识别码 ICCID。用户输入身份 ID 后，手机银行 APP 用银行服务器的公钥将用户 ID 和设备信息 DI^* 加密后发送给银行服务器。银行服务器解密接收到的数据得到用户 ID 和设备信息 DI^*，当银行服务器根据用户 ID 检索到的用户安全设备列表 $List\langle ID,DI\rangle$ 为空时，表示用户首次安装手机银行 APP，银行服务器会通过手机银行 APP 向用户请求其银行卡预留的手机号码。然后，银行服务器向用户发送短信验证码，用户收到并输入短信验证码后，手机银行 APP 调用 TEE 中的 AES-256 密码算法接口生成对称密钥 $SK1$，并用服务器公钥 S_pu 将 $SK1$ 和短信验证码加密后发送给服务器。银行服务器解密数据并确认短信验证码无误后，将当前的用户设备信息存储到安全设备列表 $List\langle ID,DI\rangle$ 中，并根据用户身份 ID 生成用户私钥 C_pr，最后利用解密得到的对称密钥 $SK1$ 将 C_pr 加密后发送给手机银行 APP。手机银行 APP 解密数据后对私钥进行验证，验证通过后，手机银行 APP 调用 TEE 中的 AES-256 加密算法接口，将私钥加密后进行安全存储，完成手机银行 APP 的首次安装。

需要说明的是，本方案采用了基于身份的密码算法，能够根据用户的身份 ID 使用公开的函数计算出其公钥。用户的私钥由银行服务器生成并由手机银行 APP 加密后保存在 APP 的安装目录下，即使合法用户也不能从安装目录下直接得到私钥，因此，攻击者即使侵入手机，也不可能获取用户私钥来伪造用户的数字签名；并且，在用户使用手机银行 APP 期间，APP 会定期向银行服务器申请新的私钥用于更新用户私钥，这将大大降低用户私钥被窃取的概率。在数据传输过程中，APP 利用银行服务器公钥对数据进行加密，保证了数据传输的机密性，从而保证了移动支付过程的安全。

算法 1 首次安装算法

Begin

1. 用户在手机上下载手机银行 APP，并点击安装；
2. 手机银行 APP 弹出对话框，提示用户输入其 ID；
3. 手机银行 APP 获取当前的设备信息 DI^*；
4. $Data=E\{S\_pu,(ID,DI^*)\}$；
5. 银行服务器接收到 APP 发送的数据 Data；
6. $Data^*=D\{S\_pr,Data\}=(ID,DI^*)$；
7. IF (List⟨ID,DI⟩=NULL)
8. Data1={Request,CellphoneNumber}；
9. 手机银行 APP 收到银行服务器的消息 Data1；
10. APP 弹出对话框，提示用户输入手机号码；
11. Data2=E{S_pu,CellphoneNumber}；；
12. 银行服务器接收到 APP 发送的数据 Data2；
13. CellphoneNumber=D{S_pr,Data2}；
14. 发送短信验证码 Str 到用户手机；
15. 银行服务器在数据库中保存：$Str^*=Str$；
16. APP 接收到用户输入的验证码；
17. APP 调用 TEE 中 AES-256 密码算法接口生成对称密钥 SK1；
18. Data3=E{S_pu,(Str,SK1)}；
19. 银行服务器接收到 APP 发送的数据 Data3；
20. D{S_pr,Data3}；
21. IF (Str=Str^*)
22. 银行服务器生成用户私钥 C_pr；
23. 银行服务器调用 AES-256 算法加密，Data4=E{SK1,C_pr}；
24. APP 接收到用户的私钥 Data4；
25. APP 调用 TEE 中的 AES-256 解密算法接口解密，C_pr=D{SK1,Data4}；
26. IF(e(C_pr,P)=e(C_pu,S_pu))
27. APP 调用 TEE 中 AES-256 加密算法将用户私钥 C_pr 加密后进行安全存储；
28. ELSE
29. 返回 false 给银行服务器；
30. 返回步骤 22；
31. END IF
32. ELSE
33. 返回 false 给 APP；
34. END IF
35. ELSE
36. 进入重装 APP 过程算法步骤 8；
37. END IF

End

3.2 支付阶段

3.2.1 身份认证流程

身份认证又称实体认证，它作为网络安全机制的基础，在

网络信息安全中具有举足轻重的地位[20-22]。只有保证身份认证的有效性，才能保证访问控制、安全审计、入侵防范等安全机制的有效性[23-25]。在本方案中，当用户登录手机银行APP时，银行服务器会对用户的身份认证信息进行验证，以确认用户身份。

当用户启动手机银行APP时，APP首先会调用TEE中的AES-256解密算法来解密私钥，并将解密后的私钥加载至内存。而后，手机银行APP会获取由用户的用户名和登录密码组成的登录信息M，以及当前的时间戳T，然后用银行服务器公钥将身份认证信息(M,T)加密后发送给银行服务器，银行服务器验证通过后，用户即可成功登录APP。当服务器记录用户登录失败3次时，服务器会设置此账号在一定时间内不能登录，并通过手机银行APP、短信或者电子邮箱等方式告知用户其账号存在安全风险，建议其更改登录密码。

算法2 身份认证算法

Begin

1. 用户启动手机银行APP；
2. APP调用TEE中的AES-256解密算法解密私钥，并将解密后的私钥加载至内存；
3. 用户输入用户名和登录密码；
4. 手机银行APP获取当前的时间戳T；
5. Data5＝E{S_pu,(M,T)}；
6. 银行服务器在T^*时刻接收到APP发送的数据Data5；
7. (M,T)＝D{S_pr,Data5}；
8. IF ($T^*-T\leqslant\triangle T$)
9. Bool＝Verify(M)；
10. IF (Bool＝true)
11. 返回true给APP；
12. 手机银行APP显示主界面；
13. ELSE
14. 转至步骤17；
15. END IF
16. ELSE
17. IF(ErrorTimes＜3)
18. 返回false给APP；
19. 手机银行APP显示失败信息；
20. ELSE
21. 银行服务器限制此账号登录；
22. 通知用户其账户存在安全风险，建议用户修改账号密码；
23. END IF
24. END IF

End

在本方案中，手机银行APP利用银行服务器的公钥对登录信息进行加密，而银行服务器的私钥也只由银行服务器自身持有，因此，攻击者无法从密文中窃取用户的用户名和登录密码，也无法利用截获的数据$E\{S_pu,(M,T)\}$来伪造用户的认证信息，保证了认证信息过程的有效性，也保证了网络中不同实体间传递信息的可靠性和保密性。

3.2.2 支付流程

在支付过程中，手机银行APP首先将用户操作生成的支付请求REQ发送给支付业务银行服务器，银行服务器对REQ验证通过后会生成短信验证码Str，并将其发送到用户手机。用户收到短信验证码后将其输入到手机银行APP中，接着手机银行APP利用用户的私钥生成对短信验证码的数字签名，然后用银行服务器公钥将数字签名和短信验证码加密后发送给支付业务银行服务器。银行服务器收到密文后用其私钥进行解密，得到数字签名和验证码，最后银行服务器对验证码进行验证并利用用户的公钥和短信验证码对数字签名进行验证，验证通过后允许用户支付。支付过程如图2所示。

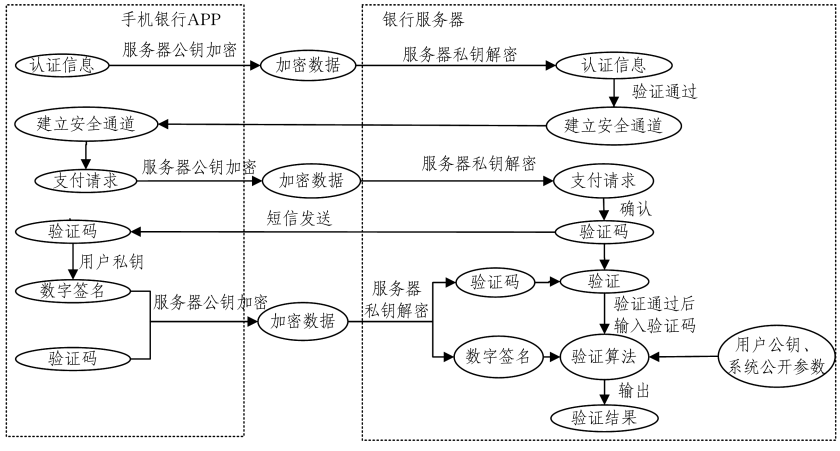


图2 支付过程

Fig. 2 Payment process

算法3 支付过程算法

Begin

1. 用户操作手机银行APP进入支付页面；
2. 手机银行APP生成支付请求REQ；
3. Data6＝E{S_pu,REQ}；
4. 手机银行APP将Data6发送给银行服务器；
5. 手机银行APP进入到输入验证码界面；
6. 银行服务器接收到APP发送的数据Data6；
7. REQ＝D{S_pr,Data6}；
8. IF (CHECK(REQ)＝true)
9. 银行服务器发送Str到用户手机；
10. 银行服务器在数据库中保存：Str^*＝Str；
11. APP收到用户输入的短信验证码Str；
12. DS＝S{C_pr,Str}；
13. Data7＝E{S_pu,(DS,Str)}；
14. APP将Data7发送给银行服务器；

15. 银行服务器收到 APP 发送的数据 Data7；
16. (DS,Str)=D{S_pr,Data7}；
17. IF(Str=Str*)
18. Bool=V{C_pu,Str,DS,params}；
19. IF(Bool=true)
20. 允许支付,返回给 APP 扣款信息,支付完成；
21. ELSE
22. 不允许支付,返回 APP 验证失败；
23. END IF
24. ELSE
25. 不允许支付,返回 APP 验证失败；
26. END IF
27. ELSE
28. 返回 APP 账户余额不足信息；
29. END IF
End

3.3 重装 APP 过程

用户在使用的过程中,有可能在移动设备上重新安装 APP,也有可能在更换移动设备后重新安装 APP。当用户名和密码都失窃后,攻击者便会利用 APP 重装机制,试图在自己的设备上安装 APP 并登入用户账户。为了支持用户重装 APP 并阻止攻击者通过安装 APP 入侵,本文方案会在用户注册时设置预留安全问题,当用户更换移动设备时,需要正确回答预留安全问题后才能成功安装 APP。

在用户第一次安装使用手机银行 APP 时,手机银行 APP 会收集设备信息并上传至银行服务器。银行服务器会将收到的用户设备信息 DI^* 和用户的身份 ID 保存在安全设备列表 $List\langle ID,DI\rangle$ 中。当用户再次安装手机银行 APP 时,手机银行 APP 将当前的设备信息 DI^* 和用户身份 ID 发送给银行服务器进行信息比对,如果 $(ID,DI^*)\in List\langle ID,DI\rangle$,表明此设备是用户的安全设备,可以正常安装;如果 $(ID,DI^*)\notin List\langle ID,DI\rangle$,表明用户更换了新的移动设备,用户需要正确回答预留的安全问题才可以成功安装,然后银行服务器会将 (ID,DI^*) 添加到用户的安全设备列表 $List\langle ID,DI\rangle$ 中。若是新设备登录,手机银行 APP 会在安装过程中就向银行服务器重新申请私钥 C_pr,申请成功后,原来的私钥就会被作废,且不能再用于支付过程的验证。如果用户在移动设备上重新安装或者升级手机银行 APP,手机银行 APP 仍使用原来的私钥 C_pr 通信,不用申请新的私钥。

攻击者如果窃取了用户的用户名和密码,就会在自己的设备上安装 APP 并登录该账户,由于其无法正确回答预留安全问题,因此会导致身份认证失败。当 3 次回答预留问题均错误时,手机银行 APP 会默认该安装者是非法用户,此时银行服务器记录“用户账户存在安全风险”,并通过手机银行 APP、短信或者电子邮箱等方式告知用户其账号存在安全风险,建议其更改登录密码。为了避免攻击者对预留的安全问题进行穷举攻击,银行服务器在一段时间内会禁止所有新设备使用该用户 ID 安装 APP。

算法 4 重装 APP 过程算法

Begin
1. 用户重新下载并安装手机银行 APP；
2. 手机银行 APP 弹出对话框,提示用户输入身份 ID；
3. 手机银行 APP 获取当前的设备信息 DI*；
4. APP 调用 TEE 中的 AES-256 密码算法接口生成对称密钥 SK2；
5. Data8=E{S_pu,[(ID,DI*),SK2]}；
6. 银行服务器接收到 APP 发送的数据 Data8；
7. D{S_pr,Data8}；
8. IF(List⟨ID,DI⟩≠Null AND (ID,DI*)∉List⟨ID,DI⟩)
9. 银行服务器返回预留的安全问题给 APP；
10. 用户输入安全问题的答案 Answer；
11. APP 获取当前时间戳 T；
12. Data9=E{S_pu,(Answer,T)}；
13. 银行服务器在 T* 时刻接收到 Data9；
14. (Answer,T)=D{S_pr,Data9}；
15. IF(Check(Answer)=true and T*−T≤ΔT)；
16. 银行服务器(ID,DI*)添加到列表 List⟨ID,DI⟩中；
17. 银行服务器生成新私钥 C_pr；
18. 银行服务器调用 AES-256 算法将用户私钥 C_pr 加密后发送给用户,Data10=E{SK2,C_pr}；
19. APP 接收到用户的私钥 Data10；
20. APP 调用 TEE 中的 AES-256 解密算法接口解密,C_pr=D{SK2,Data10}；
21. IF(e(C_pr,P)=e(C_pu,S_pu))
22. APP 调用 TEE 中的 AES-256 加密算法将用户私钥 C_pr 加密后进行安全存储；
23. ELSE
24. 返回 false 给银行服务器；
25. 返回步骤 17；
26. END IF
27. ELSE
28. IF(ErrorTimes<3)
29. 返回 false 给 APP；
30. ELSE
31. 禁止使用此 ID 安装 APP；
32. 通知用户其账户存在风险；
33. END IF
34. END IF
35. ELSE
36. 银行服务器调用 AES-256 算法将用户原来的私钥 C_pr 加密后发送给用户,Data11=E{SK2,C_pr}；
37. APP 接收到用户的私钥 Data11；
38. APP 调用 TEE 中的 AES-256 解密算法接口解密,C_pr=D{SK2,Data11}；
39. IF(e(C_pr,P)=e(C_pu,S_pu))
40. APP 调用 TEE 中的 AES-256 加密算法将用户私钥 C_pr 加密后进行安全存储；
41. ELSE
42. 返回步骤 36；
43. END IF
44. END IF
End

4 安全性分析

4.1 支付方案的安全性

1)前向安全

在本方案中,为加强移动支付方案的安全性,用户和银行

服务器的私钥会定期进行更新。银行服务器更新用户私钥的公式为 $S_{ID}=SK\times H_1(ID,T)$，其中 $H_1(ID,T)$即为用户的公钥。由公式可知，用户的公钥和私钥总是成对产生的，每次更新用户私钥时，用户公钥的值 $H_1(ID,T)$也都会被更新。本方案的通信过程未使用用户的公钥加密过任何数据，因此，即使用户正在使用的私钥被攻击者窃取，攻击者也不可能利用该私钥解密得到有用的信息。用户的私钥一旦被更新，旧的私钥将不再具备任何价值。

银行服务器更新私钥时，会随机选取一个秘密值 $SK\in Z_q^*$ 作为系统私钥，计算系统公钥的公式为：$SPK=SK\times P$。同样，服务器的公钥和私钥也是成对产生的，只有当服务器的私钥和公钥是一对时，对于公钥加密的数据，私钥才能够解密出其正确的明文。因此，即使服务器正在使用的私钥被攻击者窃取，攻击者也不可能解密过去银行服务器与手机银行APP通信的加密数据，不可能从中窃取用户的信息，从而证明了本方案是前向安全的。

2)重放攻击

首先，在用户首次安装手机 APP 过程中，由于手机银行APP与银行服务器是首次通信，攻击者不可能实施重放攻击。而当用户首次安装手机银行 APP 成功后，攻击者再利用用户 *ID* 安装手机银行 APP 时就会进入重装 APP 过程算法。此时，攻击者面临回答银行服务器返回的安全问题的难题。即使攻击者成功截获了用户回答安全问题时的通信数据 *Data*9，并通过重放 *Data*9 进行攻击，其也是不可能成功的，因为手机银行 APP 在 *Data*9 中加入了时间戳 *T*，此重放攻击会由于 $T^*-T>\Delta T$ 而验证失败。

其次，在用户登录手机银行 APP 时，手机银行 APP 与银行服务器的通信数据 *Data*5 中同样加入了时间戳 *T*。因此，攻击者也不可能通过重放数据 *Data*5 攻击成功。

最后，攻击者想要在支付过程中通过重放攻击攻击成功，就必须有成对的数据 *Data*6 和 *Data*7。但是，由于用户每次的支付请求 *REQ* 和服务器每次发送的验证 *Str* 是不同的，并且用户和银行服务器的公私钥会定期更新，因此数据 *Data*6 和数据 *Data*7 是一直在变化的。数据 *Data*6 和数据 *Data*7 每次被使用后就会失效，攻击者不可能通过重放数据 *Data*6 和数据 *Data*7 攻破支付验证算法。

综上所述，本方案是不可能被成功重放攻击的。

3)数据篡改

在手机银行 APP 发送给银行服务器的通信数据中，所有的发送数据都使用服务器的公钥进行了加密。攻击者想要通过篡改手机银行 APP 发送给服务器的数据欺骗服务器，就必须解密通信数据，获取明文信息才能修改正确。但本方案采用的基于身份的密码算法是安全可靠的，攻击者不可能攻破。因此，本方案能够防止攻击者通过篡改数据进行攻击。

4)字典攻击

攻击者想要通过字典攻击攻击本方案，只有两个入口。第一个入口是攻击者利用用户 *ID* 在自己的手机上安装手机银行 APP，对安全性问题进行字典攻击。而预留的安全性问题的答案只有用户一个人知道，当攻击者回答错误 3 次时，服务器就会禁止利用用户 *ID* 在此设备上登录，因此，想要利用用户 *ID* 通过字典攻击成功安装手机银行 APP 是不可能的。第二个入口是攻击者在用户登录过程中进行字典攻击。当登录失败 3 次时，服务器就会限制此账号在一定时间内登录。针对以上两种情况，服务器还会通知用户其账户存在风险，建议用户修改密码。因此，本方案是能够防御字典攻击的。

5)私钥机密性

攻击者想要获取用户私钥攻击本方案，有以下途径。①攻击者利用用户 *ID* 在自己的手机上安装手机银行 APP 来窃取用户私钥。但这是困难的，因为攻击者需要回答用户在注册时预留的安全问题，而问题的答案只有用户知道，因此攻击者不可能在自己的手机上成功安装 APP，也就不可能获取用户的私钥。②攻击者在服务器给手机银行分发私钥的时候窃取通信数据，并对其解密以获得用户私钥。这也是困难的，因为服务器在给手机银行 APP 分发私钥时使用了 AES-256 算法对私钥加密传输，攻击者是不可能的破解此加密数据。③攻击者通过入侵用户手机来获取用户私钥。由于本方案结合基于硬件的 TEE 对私钥进行安全存储，只有手机银行 APP 才能够访问用户私钥，因此攻击者不可能获取用户的私钥；并且手机银行 APP 调用 TEE 的中 AES-256 密码算法对私钥进行了加密，攻击者即使获取了保存在手机上的私钥的加密数据，其也不可能破解此加密数据。

因此，本文所提的利用基于身份的密码算法＋短信验证码的移动支付方案，能够有效地防止非法用户的攻击，保证用户账户的安全。

4.2 基于身份密码算法的安全性

本方案采用了基于身份的密码体制 ECC，其是一种基于椭圆曲线离散对数问题的公钥密码算法[26]。

此密码体制中 160 bits 的密钥所提供的安全性与 RSA 中 1 024 bits 的密钥所提供的安全性相当，256 bits 的密钥所提供的安全性与 RSA 中 3072 bits 的密钥所提供的安全性相当，即当椭圆曲线密码体制中的密钥长度与 RSA 中密钥长度相同时，前者所提供的安全性更高[27-28]。因此，通过计算破解本方案采用的基于身份的密码算法是困难的，该密码算法能够有效地保证用户的资金财产安全。

4.3 私有密钥定期更新

为了保证用户私钥的保密性，增强移动支付方案的安全性，手机银行 APP 会定期向银行服务器申请新的私钥用于用户私钥的更新。在更新用户私钥时，手机银行 APP 会调用 TEE 中的 AES-256 算法生成协商密钥 *SK*，并用服务器的公钥对其加密后发送给服务器。服务器收到手机银行 APP 的私钥更新请求后，为用户生成新的私钥并用协商密钥 *SK* 对其加密后发送给手机银行 APP。手机银行 APP 校验通过后，利用 TEE 中的 AES-256 算法将私钥加密并安全存储。

由于手机银行 APP 在更新用户私钥时，使用 AES-256 密码算法通过密钥协商的方式对新私钥加密传输，而不是使用用户旧的公钥对新私钥加密传输，因此，即使用户正在使用的私钥被窃取，也不会影响用户新私钥的安全性；而一旦用户私钥被更新后，旧的私钥就会作废，不能再用于支付过程中数字签名的生成，这将大大降低由于用户私钥被窃取导致的用户财产损失的概率。同时，银行服务器的私钥也会定期进行

更新。其中，私钥更新的频率可以根据银行服务器的性能以及相关因素进行设定。

5 实验结果与分析

5.1 实验环境及性能指标

实验的硬件环境为：CPU 为 Intel(R) Xeon(R) CPU E5-2609 v2@2.50 GHz 2.50 GHz；内存为 8.00 GB；操作系统为 64 位 windows 7 旗舰版；实验的软件环境为：Intellij IDEA 2018.2.5 x64 平台，JDK1.8.0_191 开发包。

5.2 实验结果及分析

在使用 184 bits 登录信息的情况下，不同数量的用户登录其账户时，系统响应用户登录的平均时间如图 3 所示。其中，横坐标表示并发登录账户的用户数量，即移动终端数；纵坐标表示在用不同位密钥对登录信息加密的情况下，系统相应的平均响应时间。从图 3 可以看出，当移动终端数一定时，随着密钥位数的增加，系统登录时的平均响应时间虽有所增加，但增加幅度并不大。当密钥位数一定时，随着移动终端数量的不断增加，系统登录的平均响应时间呈现近似线性增加。这表明随着移动终端数的增加，系统运行稳定，健壮性较好。

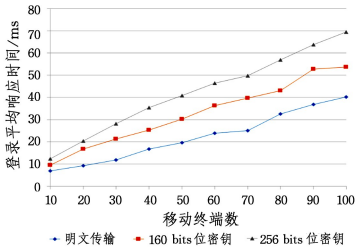


图 3 不同数量的移动终端登录时的平均响应时间

Fig. 3 Average response time when different mobile terminals log in

对不同的密钥采用 4 位和 6 位验证码时，系统响应用户支付的平均时间如图 4 所示。其中，横坐标表示并发支付的移动终端数；纵坐标表示用户提交验证码到收到支付确认消息的支付平均响应时间；160-4，160-6，256-4 和 256-6 曲线分别表示采用 160 bits 密钥 4 位验证码、160 bits 密钥 6 位验证码、256 bits 密钥 4 位验证码和 256 bits 密钥 6 位验证码时的支付平均响应时间曲线。

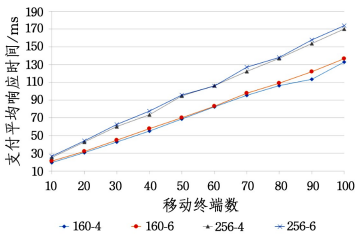


图 4 采用不同位验证码时的支付平均响应时间

Fig. 4 Average payment response time with different bit verification code

从图 4 可以看出，当密钥位数一定时，无论采用 4 位验证码还是 6 位验证码，支付平均响应曲线走势大体相同，支付响应时间差距非常小；当验证码位数一定时，无论是采用 160 bits 密钥还是 256 bits 密钥，随着并发支付的移动终端数的增加，支付平均响应时间都呈现近似线性增加。这表明，采用基于身份的密码算法对验证码签名和对传输数据加密不会影响系统性能，不会成为造成系统性能瓶颈的因素。

在使用 6 位验证码的情况下，采用不同位密钥时，系统响应用户支付的平均时间如图 5 所示。其中，横纵坐标表示并发支付的移动终端数；纵坐标表示当采用不同位密钥对验证码签名和对传输数据加密时的支付平均响应时间。从图 5 可以看出，当密钥位数一定时，随着并发支付的移动终端数的增加，支付平均响应时间呈现近似线性增加；而当移动终端数一定时，增加密钥位数，支付平均响应时间并未出现大幅增加，处于稳定增长的态势。这表明随着并发支付的移动终端数的增多，系统性能不会出现急剧下降的情况，证明本系统安全、高效，具有较好的健壮性。

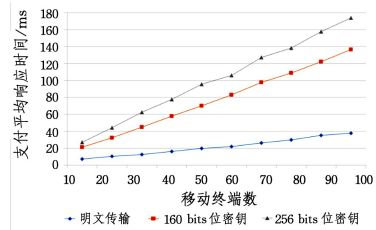


图 5 采用不同位密钥时的支付平均响应时间

Fig. 5 Average payment response time for different bit keys

结束语 针对短信验证码被盗后造成用户财产损失的问题，本文提出了利用基于身份的密码算法＋短信验证码的移动支付方案。不同于传统的基于短信验证码的移动支付方案中单单依靠验证码进行安全验证的方式，在本方案中，当用户在手机银行 APP 输入短信验证码后，手机银行 APP 会利用用户的私钥生成对短信验证码的数字签名，然后用银行服务器公钥对短信验证码和数字签名加密后发送给银行服务器。移动支付方案的安全性由短信验证码和数字签名共同保障，即使短信验证码被攻击者窃取，也不会对用户造成财产损失。而在传统方案中，一旦短信验证码被窃取，便无安全性可言。其次，本方案采用基于身份的密码算法建立系统，因此也就不需要证书授权机构等所需的一系列的设施，这将为移动支付系统的建立节省大量的开销。理论分析和实验结果表明，本文提出的利用基于身份的密码算法＋短信验证码的移动支付方案能够提供短信验证码＋数字签名的有效安全验证，并且随着移动终端数的增加，系统响应时延增幅较小，系统运行稳定，健壮性较好。因此，本方案具有可行性，能够解决短信验证码被窃取对用户造成财产损失的问题。下一步将继续改善本方案，以进一步提高方案的安全性。

参 考 文 献

[1] China Internet Network Information Center. The 42nd Statistical Report on Internet Development in China [R]. Beijing: China Internet Network Information Center, 2018.

[2] DAHLBERG T, GUO J, ONDRUS J. A critical review of mobile payment research[J]. Electronic Commerce Research and Applications, 2015, 14(5): 256-284.

[3] LIU Y L, JIN Z G, GAO T Y. Survey of Security Research in Mobile Payment System [J]. Information Network Security, 2017(2): 1-5.

[4] ISAAC J T,SHERALI Z. Secure Mobile Payment Systems[J]. IT Professional,2014,16(3):36-43.

[5] CAO W,ZHAO Y. Research on the Technology of Mobile Payment Security Based onTwo-Factor Authentication [J]. Information Security and Technology,2014,5(2):10-12,15.

[6] MTAHO A B. Improving Mobile Money Security with Two-Factor Authentication[J]. International Journal of Computer Applications,2015,109(7):9-15.

[7] FAN M,CHEN L. Research on Security Threats of SMS Verification Code Based on Mobile E-commerce[J]. Journal of Hefei University of Technology (Social Science Edition),2017,31(5):37-41.

[8] ZHOU C Y,WANG J W,LI M. Research on Identity—Based Cryptography Application in Internet of Things [J]. Information Security Research,2017,3(11):1040-1044.

[9] SHAMIR A. Identity-based Cryptosystems and Signature Schemes[M]. Germany:Springer-Verlage,1984.

[10] 中国密码学会组. 中国密码学发展报告 2008[M]. 北京:电子工业出版社,2009:1-32.

[11] RAY S,BISWAS G P,DASGUPTA M. Secure Multi-Purpose Mobile-Banking Using Elliptic Curve Cryptography[J]. Wireless Personal Communications,2016,90(3):1331-1354.

[12] LAUTER K. The advantages of elliptic curve cryptography for wireless security[J]. IEEE Wireless Communications, 2004, 11(1):62-67.

[13] JANA B,PORAY J. A performance analysis on elliptic curve cryptography in network security[C]// International Conference on Computer. IEEE,2017.

[14] SINGH S R,KHAN A K,SINGH S R. Performance evaluation of RSA and Elliptic Curve Cryptography[C] // International Conference on Contemporary Computing and Informatics. Noida:India:IEEE,2017:302-306.

[15] LI J F,CUI J S. Elliptic Curve Encryption Algorithm and Case Analysis[J]. Network Security Technology and Application, 2004(11):56-57.

[16] SHIM K A. An ID-based aggregate signature scheme with constant pairing computations[J]. Journal of Systems & Software, 2010,83(10):1873-1880.

[17] EKBERG J E,KOSTIAINEN K,ASOKAN N. The Untapped Potential of Trusted Execution Environments on Mobile Devices [J]. IEEE Security & Privacy,2014,12(4):29-37.

[18] DAI W,JIN H,ZOU D,et al. TEE:A virtual DRTM based execution environment for secure cloud-end computing[J]. Future Generation Computer Systems,2015,49:47-57.

[19] YONGKAI F,SHENGLE L,GANG T,et al. Fine-grained access control based on Trusted Execution Environment[J/OL]. https://doi.org/10.1016/j.future.2018.05.062.

[20] ABDULWAHID A A,CLARKE N,STENGEL I,et al. The Current Use of Authentication Technologies: An Investigative Review[C]// International Conference on Cloud Computing. Riyadh,Saudi Arabia:IEEE,2015.

[21] CRAWFORD H,RENAUD K. Understanding user perceptions of transparent authentication on a mobile device[J]. Journal of Trust Management,2014,1(1).

[22] FAN J S,ZHANG J X. On Identity Authentication Technology in Network Security[J]. Network Security Technology and Application,2018(1).

[23] LI L,LIU Y. Security analysis of mobile payment system [J]. Journal of Electronic Measurement and Instrument,2017(3).

[24] China Communications Standards Association. Technical re-quirements for security capability of smart mobile terminal: YD/T 2407-2013 [S]. Beijing:The People's Posts and Telecommunications Press,2013.

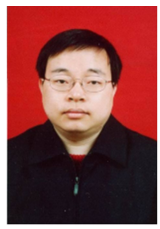
[25] XU Y P,MA Z F,WANG Z H,et al. Survey of security for Android smart terminal[J]. Journal on Communications, 2016, 37(6):169-184.

[26] LAUTER K E. The advantages of elliptic curve cryptography for wireless security[J]. IEEE Wireless Communications,2004, 11(1):62-67.

[27] ABDULLAH K. Comparison between the RSA cryptosystem and elliptic curve cryptography[D]. Hamilton, New Zealand: The University of Waikato,2010.

[28] PAAR C,PELZL J. The RSA Cryptosystem[M]. Understanding Cryptography. Berlin:Springer,2010.

LIU Ya-qiang, born in 1992, postgraduate, is not member of China Computer Federation (CCF). His main research interests include mobile information security, mobile payment.



LI Xiao-yu, born in 1974, Ph.D, associate professor, is member of China Computer Federation (CCF). His main research interests include mobile computing, quantum computing and quantum information.