文章编号:1671-4598(2019)03-0225-06 DOI:10.16526/j.cnki.11—4762/tp.2019.03.046 中图分类号:TN918.4 文献标识码:A

基于可验证 SM2 门限算法的移动终端签名系统的设计与实现

唐泽严[1]，李文军[2]，黄晓芳[1]
(1. 西南科技大学 计算机科学与技术学院，四川 621010；
2. 北京市公安局 昌平分局警务支援大队，北京 102200)

摘要：随着移动互联网的快速崛起，移动终端作为重要的信息通信及数据载体，其在信息传输及处理过程中发挥了极大作用，但是移动终端密钥的使用、存储不安全性限制了其使用；因此文章在国密算法标准基础上，提出并实现了一种适用于移动终端的基于可验证 SM2 门限签名方案；该方案对密钥进行分散生成、存储，并在密钥生成、数字签名阶段采用验证公式对传递份额进行计算，保证了在移动端的数字签名的安全性，并通过性能测试，表明该系统具有良好的性能并满足签名算法的安全性要求。

关键词：门限签名；SM2；密钥分散；可验证

Design and Achieve of Mobile Terminal Signature System Based on Verifiable SM2 Threshold Scheme

Tang Zeyan[1], Li WenJun[2], Huang Xiaofang[1]
(1. Schoole of Computer Science and Technology, Southwest University of Science and Technology, Mianyang 621010, China; 2. Police Support Branch of Beijing Public Security Bureau Changping District Bureau, Changping 102200, China)

Abstract: With the rapid rise of the mobile Internet, mobile terminals, as important information communication and data carriers, have played an important role in the information transmission and processing, but the use and storage insecurities of mobile terminal keys have limited their use. Therefore, based on the national secret algorithm standard, this paper proposes and implements a verifiable SM2 threshold signature scheme suitable for mobile terminals. The scheme decentralizes and stores the key, and uses the verification formula to calculate the transfer share in the key generation and digital signature phase, which ensures the security of the digital signature on the mobile terminal, and the performance test shows that the system has good performance and meets the security requirements of the signature algorithm.

Keywords: Threshold signature; SM2; Key share; Verifiable

0 引言

伴随着移动互联网的快速发展、智能手机迅速普及，各种移动互联网业务也迅速发展，如移动电子商务、手机支付、手机阅读、手机游戏、手机证券等[1]。移动端作为一个开放性的轻量级设备，在密钥传输、使用过程中所面临的安全威胁更为严峻。因此，如何保证在移动端安全地使用密钥进行数字签名[2]成为了目前移动安全应用所面临的关键问题。

目前为了解决常用的 1024 位 RSA 算法所面临的严重安全威胁，国家密码管理部门发布了以 ECC 算法为基础的改进密码算法 SM2 椭圆曲线公钥密码算法[3]。该方案的正确性、效率和安全性上都优于传统 RSA 和 ECC 算法。为了在移动设备上使用基于 SM2 的数字签名[4]，并且符合《电子签名法》私钥只能由合法用户持有并受此用户控制的规定，本文在尚铭等人[5]提出的 SM2 椭圆曲线门限密码算法和基于多项式的可验证门限密码技术[6]的基础上提出了一种适用于移动终端数字签名系统的可验证 SM2 椭圆曲线门限密码方案并通过实现该方案，证明本文提出的可验证 SM2 门限签名系统方案，在增加可接受时间消耗的同时，能抵御移动终端环境下恶意敌手的欺骗攻击以及合谋攻击，更符合移动数字签名的安全需求。

1 相关研究

1.1 SM2 数字签名算法

SM2 数字签名算法是国家密码安全局于 2010 年 12 月 17 日发布的基于椭圆曲线离散对数问题的数字签名算法。可用于保证身份的真实性、数据完整性和行为的不可否认性等。该算法中由一个签名者对数据产生数字签名，并由一个验证者验证数字签名的可靠性。在使用该算法之前，各参与方需先设定相同的公开参数，包括 p、q、E 和 G，

收稿日期:2018-12-10； **修回日期**:2019-01-03。
基金项目:四川省组织部(17sjjg02)；四川省教育厅(17zd1119)；四川省军民融合研究院(18sxb022)。
作者简介:唐泽严(1993-)，男，重庆人，硕士研究生，主要从事数字签名相关方向的研究。

其中 p 是大素数，E 是定义在有限域 F_p 上的椭圆曲线，$G=(x_G, y_G)$ 是 E 上 q 阶的基点。其签名过程如下[5]：

(1) 密钥生成阶段，签名者随机选取秘密 d，$d\in[1, q-1]$，计算 $P=dG$ 并将 P 作为公钥公开，d 作为私钥保存。

(2) 签名生成阶段，签名者随机选取随机数 $k\in[1, q-1]$，计算 $kG=(x_1, y_1)$；随后计算 $r=(Hash(m)+x_1)\ mod\ q$，其中 m 为待签名的消息，其中 Hash(·) 是单向哈希函数；若 $r=0$ 或者 $r+k=q$，则需要重新选取随机数 k。最后计算签名值 $s=(1+d)^{-1}(k-rd)\ mod\ q$，若 $s=0$，也必须重新选择随机数 k；否则将 (r, s) 作为签名结果。

(3) 签名验证阶段，验证者接收到 m 和 (r, s) 后，首先检查是否满足 $r, s\in[1, q-1]$ 且 $r+s\neq q$；然后计算 $(x'_1, y'_1)=sG+(r+s)P$；$r'=(Hash(m)+x'_1)\ mod\ q$，判断 r'和 r 是否相等，若二者相等则签名验证通过，否则验证失败。

1.2 可验证门限密码方案

门限密码方案[6]是指采用秘密分享技术将基本的密码机制生成的结果（密钥或者数字签名等）分布于一定数量的参与者集合中，只有有效的参与者子集进行联合，才能恢复正确的密钥或者发布有效的数字签名，而不合法的参与者子集则无法通过伪造参数恢复出正确的密钥或者产生有效数字签名。门限密码方案需要满足以下两个条件：

(1) 每个参与者对密文或者消息应使用自己的子密钥进行解密或签名操作可以得到对应的子明文或者子签名消息，t 个子明文或者子签名消息可以恢复出原始明文或者数字签名，而少于 t 分得子明文或者子签名消息不能恢复出原始数据；

(2) 已知子明文或者子签名消息不能获取主密钥与子密钥之间的任何信息。

可验证门限密码方案[6]是在通常的门限密码方案的基础上额外添加验证操作，从而解决不诚实分发者或参与者的问题。在可验证门限密码方案中，分发者不但需要向其他参与者发布自己的秘密份额，还需要广播自己对该秘密份额的证明，当各个成员获取其秘密份额时，需要通过公开的证明对秘密份额的正确性进行验证；在秘密重构阶段，每个参与者也需采用同样的验证方法来对其他成员发布的秘密份额的正确性进行校验。可验证门限密码方案可以抵御以下两种攻击：

(1) 恶意的分发者在份额分发协议中向其他参与者发送错误的秘密份额；

(2) 恶意的参与者在秘密重构协议中向其他参与者发布错误的秘密份额。

2 改进的可验证 SM2 门限签名方案

2.1 密钥生成协议

(1) 对于参与者集合 $\{U_1, U_2, \cdots, U_i\}$ 每个参与者 U_i $(i=1, 2, \cdots n)$ 选择唯一的身份表示 $ID_i\in\{0, 1\}^*$，计算身份代码 $x_i=H_1(ID_i)\in Z_q^*$ 并公开 x_i；根据门限值 t，每个参与者 U_i 选定一个 t 阶多项式 $f_i(x)=a_{i0}+a_{i1}x+a_{i2}x^2+\cdots+a_it\ x^t\in Z_q[x]$，其中 $a_ik\in Z_q^*$ $(k=0, 1, 2\cdots t)$；参与者 U_i 计算 $a_ik\ G$ 和分散份额 $\lambda_{ij}=f_i(x_j)\ mod\ q$，$j=1, 2, \cdots, n$，公开广播 $a_{ik}G$，将 λ_{ij} 发送给 U_j $(j\neq i)$；

(2) 参与者 U_j 接收到由其他 $n-1$ 个参与者发来的 λ_{ij}，验证 $\lambda_{ij}G=\sum_{k=0}^{t-1}a_{ik}Gx_j^k$ 是否成立。若成立，U_j 接收 λ_{ij}，否则拒绝；

(3) 参与者 U_j 根据接收到的 λ_{ij} 计算自己得到的秘密份额 $\lambda_j=\sum_{j=0}^{n}\lambda_{ij}\ mod\ q=\sum_{i=1}^{n}f_i(x_j)\ mod\ q$ 以及公开份额信息 $R_i=\lambda_iG$；

(4) 参与者 U_i 通过公开信息 $a_{io}G$ $(i=1, 2, \cdots n)$ 计算系统公钥 $Q=\sum_{i=1}^{n}a_{io}G=dG$。

2.2 签名生成协议

(1) 参与者集合执行 t 阶 J－RSS 和 2t 阶 J－ZSS 分享份额分别为 β_i 和 α_i；

(2) 参与者 U_i 计算并广播 $\gamma_i=\beta_i(1+d_i)+\alpha_i$；

(3) 参与者 U_i 记录 γ_j $(1\leqslant j\leqslant n)$，并通过插值公式恢复出 $\gamma=\beta(1+d)$；

(4) 参与者 U_i 计算 $(1+d)^{-1}$ 的分享份额 $d'_i=\gamma^{-1}\beta_i$，并验证参数广播 $D_i(-1)=\gamma\beta_i^{-1}G=(d'_i)^{-1}G$；

(5) 设从参与者集合 U 中选择 $2t-1$ 个签名者，设签名者集合为 $S=\{i_1, i_2, \cdots, i_{2t-1}\}$，签名者 U_i $(i\in S)$ 执行 t 阶 J－RSS，分享份额为 k_i，并广播 $K_i=k_iG$；

(6) 签名者 U_i 执行 PM－SS 得到 $kG=(x_1, y_1)$，并计算 $r=(\mathrm{Hash}(m)+x_1)\ mod\ q$；

(7) 签名者 U_i 计算签名份额 $s_i=d'_i(k_i+r)-r$，得到签名 s 的分享份额 s_i；

(8) 至少 $2t-1$ 个签名者 U_i 通过广播发布其签名份额 s_i；

(9) 签名者 U_j 接收到签名份额 s_i 通过验证公式 $s_iD_i^{-1}+rD_i^{-1}-rG=K_i$ 对签名者 U_i 的签名份额进行验证，正确则接收，错误则拒绝；

(10) 签名者 U_i 通过对接收到的 $2t-1$ 个正确的签名份额进行插值计算，得到签名结果 $s=\sum_{i\in S}s_i$，输出 (r, s) 作为签名者集合 S 对消息 m 的数字签名。

2.3 签名验证协议

(1) 签名验证者接收 m 和 (r, s) 后；

(2) 验证者检查是否满足 $r, s\in[1, q-1]$ 且 $r+s\neq q$；

(3) 计算 $(x'_1, y'_1)=sG+(r+s)P$；$r'=(Hash(m)+x'_1)\ mod\ q$，判断 r'和 r 是否相等，若二者相等则签名验证通过，否则验证失败。

3 基于可验证 SM2 移动签名系统的实现

3.1 系统架构设计

基于可验证 SM2 门限移动数字签名系统结构主要由移

动终端客户端、通信服务队列和密钥管理服务器构成，其各自作用如下。

（1）移动终端：移动终端作为客户端，提供客户端软件，支持对客户端密钥的管理，包括：密钥和签名的生成、存储、备份；为提高移动终端应用的安全性，防止 java 层代码被反编译，从而导致密钥份额等内容泄漏。移动终端应用代码基于 Android Studio NDK 开发编译生成可安装 APK 文件，从而增强移动客户端的安全性。

（2）通信服务队列：采用基于 AMQP 的消息通信服务 RabbitMQ，提供分布式消息传递和消息队列服务，利用其高效可靠的消息传递机制为移动终端和密钥管理服务器提供快速稳定的数据交换、份额分发、进程间通信等服务；

（3）密钥管理服务器：配合移动终端设备完成基于 SM2 的可验证数字签名的生成、保存、备份等操作的服务器。密钥管理服务代码采用 C++语言进行开发，其算法执行内容与移动终端执行内容相同。

3.2　功能模块划分

可验证 SM2 门限签名系统为能够移动终端提供安全可靠的签名服务，根据改进的可验证 SM2 门限签名方案中的不同协议抽象出不同功能模块，对应分别是：（1）公私密钥生成模块；（2）数字签名模块；（3）数字签名验证模块。各个模块的功能为：公私钥密钥生成模块通过可验证门限密码方案生成移动终端签名群的公私钥；数字签名模块通过可验证门限签名方案生成有效的 SM2 数字签名；签名验证模块可对数字签名模块生成的 SM2 数字签名的正确性、有效性进行验证。各个模块相互独立，可以实现移动终端在不同阶段调用的需求。此系统中包含的算法的实现类图如图 1 所示。

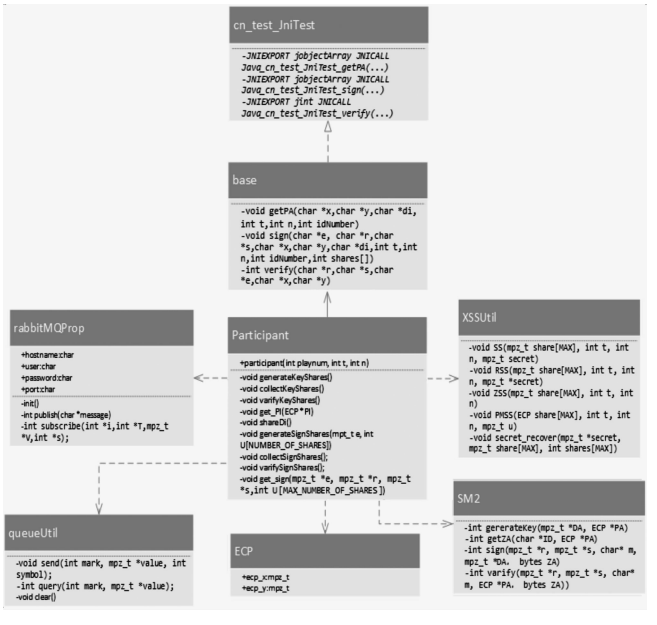


图 1　可验证门限分散算法类图

其中：cn _ test _ JniTest 为移动终端应用采用 NDK 实现的 JNI 接口声明，包括密钥生成、签名生成和签名验证 3 个本地接口声明，密钥管理服务端则不包含此函数；base 为上述本地接口的对应实现；Participant 为基于 SM2 的可验证门限签名方案的具体实现，包括门限密钥份额生成、验证、重构、数字签名份额生成、验证、重构以及数字签名认证等方法；XSSUtil 为 Shamir 秘密分享的实现，包括随机秘密分享、零秘密分享、乘积秘密分享等过程的实现；ECP 为椭圆曲线点类；SM2 为 SM2 密码算法的相关函数，包括 SM2 密钥生成、SM2 数字签名生成、SM2 数字签名验证等方法；RabbitMQProp 为消息传递服务组件 RabbitMQ 的基本配置，包括服务端地址、端口等信息、以及对通信队列建立通信连接等方法；queueUtil 为消息传递服务组件对消息队列的处理方法，包括消息的收发等操作。

3.3　系统模块设计与实现

3.3.1　密钥生成模块

在本阶段模块的主要工作是产生数字签名系统的公私钥对，主要过程为通过多项式秘密分享的方式分享门限签名密钥份额，并获取通过广播多项式承诺值对密钥份额的有效性进行判断。具体步骤为：

（1）移动终端签名应用发起获取公钥申请，密钥管理服务器获得申请后根据门限值（t，n）创建 $n-1$ 个密钥服务进程，并初始化数字签名算法涉及的公开参数、创建通信服务队列。

（2）移动终端应用与 $n-1$ 个密钥管理服务进程同时执行改进 SM2 可验证门限密码生成协议获得系统公钥以及各自持有的私钥份额。主要函数过程如下：

```
void getPA(char * x,char * y,char * di, int t,int n,int playerno)
{
randomize();
Participant player=Participant(playerno,t,n);;
mpz_set_ui(init,playerno);
ECP PA;
player. generteKeyShares();
player. collectKeyShares();
player. varifyKeyShares();
player. getPA(&PA);
}
```

3.3.2　签名生成模块

在本阶段模块中主要实现数字签名的生成。主要过程为首先通过多项式秘密分享过程分享签名生成参数 $(1+d_i)^{-1}$、k_i，并广播其验证参数 $D_i^{-1}=(1+d_i)G$、$K_i=k_iG$。签名参与者各自通过验证公式 $s_iD_i^{-1}+rD_i^{-1}-rG=K_i$ 验证签名参数份额的正确性，得到 $2t-1$ 个正确的签名份额后通过差值公式即可得到签名结果（r，s）。具体步骤为：

（1）移动终端签名应用发起数字签名申请，密钥管理服务器获得数字签名申请之后，创建 $n-1$ 个密钥服务签名进程，并载入对应的私钥份额；

(2) 移动终端与各个签名服务进程执行改进可验证 SM2 门限签名协议 1～4 步，通过多项式秘密分享过程分享签名生成参数 $(1+d_i)^{-1}$，并广播其验证参数 $D_i^{-1}=(1+d_i)$，主要函数过程如下：

```
void sign(char *e, char *r,char *s,char *x,char *y,char
*di,int t,int n,int playerno,int U[])
{
randomize();
Participant player=Participant(playerno,t,n);
player.shareDi()
}
```

(3) 选取其中 $(2t-2)$ 个签名进程配合移动终端应用，作为签名参与者执行可验证 SM2 门限协议 5～10 步，生成并检验数字签名份额，在获得 $2t-1$ 个正确的数字签份额后，移动终端应用通过差值公式生产对应的数字签名 (r, s)，主要函数过程为：

```
void sign(char *e, char *r,char *s,char *x,char *y,char
*di,int t,int n,int playerno,int U[])
{
mpz_t r0,s0,di0,x0,y0;
mpz_init(r0);
mpz_init(s0);
mpz_init(di0);

EC_point PA;
PA.init();

mpz_set_str(PA.x,x,16);
mpz_set_str(PA.y,y,16);
mpz_set_str(di0,di,16);

player.generteSignShares(e0,U);
player.collectSignShares();
player.varifySignShares();
player.get_sign(&e0,&r0,&s0,U);
}
```

3.3.3 签名验证模块

在此阶段模块中主要实现对签名结果的验证，验证过程由密钥管理服务器执行。具体步骤为：

(1) 移动终端应用向密钥管理服务器发起激活签名验证进程请求，密钥管理服务器获取验证请求之后创建签名验证进程，并载入相关参数；

(2) 密钥管理服务器签名验证进程检查 (r, s) 是否满足 $r, s\in[1, q-1]$ 且 $r+s\neq q$；若满足参数条件，则计算基点 $(x'_1, y'_1)=sG+(r+s)P$ 以及验证签名值 $r'=(Hash(m)+x'_1)\ mod\ q$，若 r' 与 r 是否一致，若二者相等则表明签名没有被篡改，否则签名失效，并将结果返回至移动终端应用。主要函数过程为：

```
int verify(char *r,char *s,char *e,char *x,char *y)
{
mpz_t r0,s0,e0;;
mpz_init(r0);
mpz_init(s0);
mpz_init(e0);
EC_point PA;
PA.init();
mpz_set_str(PA.x,x,16);
mpz_set_str(PA.y,y,16);
mpz_set_str(r0,r,16);
mpz_set_str(s0,s,16);
mpz_set_str(e0,e,16);
if(SM2.verify(&r0,&s0,&e0,&PA)){
cout<<"签名验证:通过!"<<endl;
return 1;
}
else{
cout<<"签名验证:未通过!"<<endl;
return 0;
}
}
```

4 系统分析与讨论

4.1 正确性分析

通常来讲，门限签名的正确性指的是通过签名算法所生成的数字签名结果必须通过对应的签名验证算法，而本文所提出的改进的 SM2 可验证门限签名方案中涉及密钥份额、子签名份额以及门限签名，下面对这 3 个结果的正确性进行论证。

定理 1：在 SM2 可验证门限签名方案的密钥生成阶段，参与者 U_j 能够验证参与者 U_i 所发送的 λ_{ij} 的真伪，从而保证密钥生成阶段产生的参与者各自的私钥份额是正确的。

证明：参与者 U_i 在密钥生成阶段会公开自己份额对应的校验信息 $a_{ik}G$ $(k=0, 1, \cdots, t)$，$\lambda_{ij}G$ 可以写成：

$$\lambda_{ij}G=f_i(ID_j)G=a_i0\ G+a_i1\ ID_jG+a_i2\ ID_j^2\ G+\cdots+a_it\ ID_j^t G$$

根据参与者 U_i 公开的证明信息，参与者 U_j 可通过验证上式是否成立来验证参与者 U_i 所发送的份额参数 λ_{ij} 的是否正确，从而保证获取的份额参数是正确的，进而保证参与者 U_j 通过差值计算获得正确的私钥份额。

定理 2：在签名份额生成过程，签名者 U_j 能够验证合法的签名者 U_i 所发布的签名份额信息 s_i 的真伪，从而保证签名者 U_j 获取的签名份额是正确的。

证明：根据签名参与者 U_i 所公开的校验信息 $D_i^{-1}=(d'_i)^{-1}G$，对于一个诚实的签名者 U_i 所产生的正确的门限签名份额 $s_i=d'_i(k_i+r)-r$ 应满足签名份额验证公式：

$$s_iD_i(-1)+rD_i(-1)-rG=[d'_i(k_i+r)-r)](d'_i)^{-1}G+rD_i^{-1}-rG=(k_i+r-r(d'_i)^{-1})G+rD_i^{-1}-rG=k_iG+$$

$$rG - rD_i^{-1} + rD_i^{-1} - rG = k_i G$$

而 $K_i = k_i G$，故 $K_i = s_i D_i^{-1} + rD_i^{-1} - rG$ 成立，所以若子签名份额能够通过上式的校验，则证明签名参与者 U_j 获取 U_i 签名份额是正确的。

定理 3：在数字签名重构过程中，签名者通过差值公式计算得出的签名值 s 是正确的。

证明：由 SM2 可验证门限签名生成协议过程第（6）可得 $r = (Hash(m) + x_1) \bmod q$，而 $(x_1, y_1) = kG$ 式中 $k = \sum_{i \in S} k_i$，所以 x 即是点 kG 的 x 坐标。由于：

$$(k_1 + r, \cdots, k_t + r) \leftrightarrow k + r,$$

$$((1 + d_1)^{-1}, \cdots (1 + d_t)^{-1}) \leftrightarrow (1 + d)^{-1},$$

根据分散秘密重构定理，在获得 $2t-1$ 个正确的签名份额的条件下：

$s = \sum_{i \in S} s_i = (1 + d)^{-1}(k + r) - r$，因此该签名方案产生的数字签名是正确的。

4.2　安全性分析

在本系统实现方案中通过可验证门限技术保证系统的安全性，以下将从防恶意敌手的欺骗攻击和抗不安全集合的合谋攻击两个方面对该系统方案的系统的安全性能进行论证。

定理 1：本文提出的基于 SM2 的可验证签名方案可以抵御欺骗攻击。

证明：门限数字签名方案中，恶意敌手可以在密钥生成阶段或者数字签名生成阶段进行欺骗攻击。在本可验证系统方案中，如果恶意敌手在执行密钥生成协议过程中时，伪造参与者密码份额进行欺骗，恶意敌手将会在份额验证阶段被密钥生成验证公式 $\lambda_{ij} G = \sum_{k=0}^{t-1} a_{ik} G x_j^k$ 检测出来；如果恶意敌手在执行签名生成协议过程中伪造错误的签名份额进行欺骗，恶意敌手会在签名生成验证阶段被签名份额验证公式。

$s_i D_i^{-1} + rD_i^{-1} - rG = K_i$ 检测出来，因此本签名方案可以抵御来自恶意敌手的欺骗攻击。

定理 2：本文提出的数字签名方案可以抵御合谋攻击。

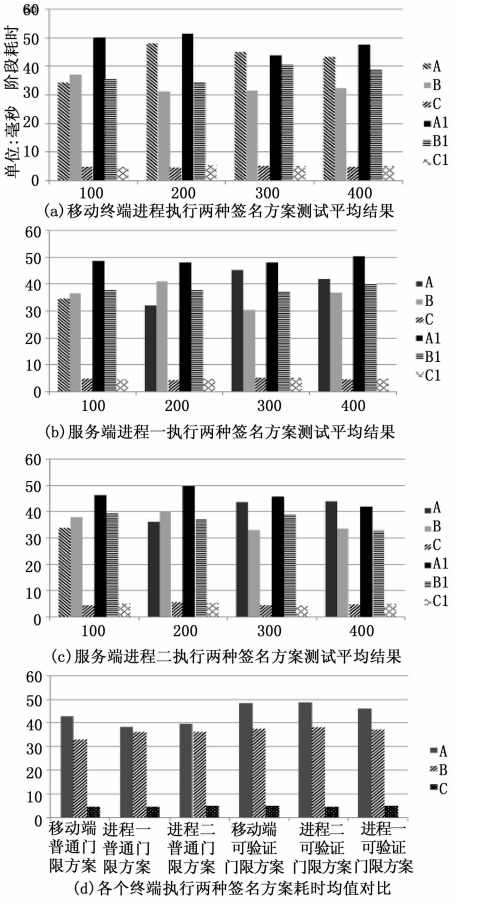
证明：由 Shamir 秘密分享方案可知，在（t，n）门限的条件下，大于等于 $2t-1$ 个成员可以通过合谋重构 Lagrange 恢复出插值多项式 $f(x)$，由于 ID_i 为公开信息，故合谋者可以获得其他群成员 U_i 的秘密份额 λ_i，但是根据子签名 $s_i = (1 + d_i)^{-1}(k_i + r) - r$，合谋者无法获得 U_i 的私钥 d_i，因此不能冒充群成员 U_i 对消息 m 生成对应子签名份额。所以本文提出的门限签名方案是可以抵抗不安全集合的合谋攻击。

4.3　效率分析

基于第 3 章的系统设计模型，采用（t，n）门限集合为（2，3）的情况下，在表 1 所示系统设备上针对可验证 SM2 门限签名系统方案和普通门限签名方案进行系统运行效率测试，测试结果如图 2 所示。其中系统参与者由一个移动终端进程和 2 个密钥服务端进程构成，共同执行改进的可验证 SM2 门限签名方案和普通签名方案。图 2－a，b，c 中纵坐标表示分别运行普通门限签名方案和本文提出的 SM2 可验证门限签名方案在 100、200、300、400 次运行次数下不同进程在门限签名方案的 3 个阶段的平均耗时；横坐标 A、B、C 分别代表使用普通门限数字签名方案的 3 个阶段，分别为密钥生成阶段、签名生成阶段以及签名验证阶段，而 A1、B1、C1 分别代表使用可验证 SM2 门限数字签名方案对应的 3 个阶段。图 2－d 则为 3 个终端分别执行两种方案时各个阶段的总平均时间消耗。

表 1　系统设备配置参数

终端/服务器	设备配置参数
移动终端	小米 MI－MAX；CPU：6 核，1.8GHz；内存：2GB
通信服务器	操作系统：Windows；CPU：X86_X64，8.0GHz 内存：8GB
密钥管理服务器	操作系统：Ubuntu；CPU：X86_X64，8.0GHz 内存：8GB



(a)移动终端进程执行两种签名方案测试平均结果

(b)服务端进程一执行两种签名方案测试平均结果

(c)服务端进程二执行两种签名方案测试平均结果

(d)各个终端执行两种签名方案耗时均值对比

图 2　（2，3）条件下两种方案的效率对比

根据图 2（a），（b），（c）可以看出在各个终端执行两种方案时的时间消耗与 2（d）中的平均结果相差不大，证明本面向移动终端的数字签名系统在良好网络环境下具有良好的稳定性。根据图 2（d）中的数据可知在（2，3）条

件下，未采用可验证方案的密钥生成阶段平均耗时约 40.294 ms，签名生成阶段平均耗时约 35.186 ms，签名验证阶段约为 4.954 ms；采用可验证方案的密钥生成阶段平均耗时约 47.711 ms，签名生成阶段平均耗时约 37.687 ms，签名验证阶段约为 5.148 ms。由此可以推论采用 SM2 可验证门限签名方案在增强系统整体安全性和数字签名正确性的基础上同时也增加了一定的时间，但增加的时间消耗在可接受范围。而面对移动终端的开放性环境，对接收的消息做进一步确认是必要的，因此此方案具有一定的实用性和应用价值。

5 结语

本文提出的基于 SM2 的可验证门限签名方案，每个参与者私钥由参与者自己选择，并根据可验证秘密分享过程对 SM2 数字签名进行可验证秘密分享。安全性分析表明在密钥生成、签名生成阶段，利用公开信息进行验证，可以有效避免恶意敌手或者参与者的欺诈，且可以抵御参与者合谋攻击。因此，该方案具有良好的安全性。通过对系统功能的实现以及相关的测试，证实该系统方案的正确性和可靠性。结果证明，基于可验证 SM2 门限算法的移动终端签名系统满足开放性移动终端安全使用的需求，且相较于同类数字签名方案更贴近实际环境，具有一定的实际应用价值。

参考文献：

[1] 马 骏，马建峰，郭渊博. 可证明安全的智能移动终端私钥保护方案 [J]. 通信学报，2012 (12)：108－115.

[2] 王 芸，赵长江. 安全的移动终端数字签名方案. 中国科技信息，2015，(22)：52－54.

[3] 国家密码管理局. SM2 椭圆曲线公钥密码算 [EB/OL]. http://www.oscca.gov.cn/UpFile/20101222214822692.pdf，2012.

[4] 汪朝晖，张振峰. SM2 椭圆曲线公钥密码算法综述 [J]. 信息安全研究，2016，2 (11)：972－982.

[5] 尚 铭，马 原，林璟锵，荆继武. SM2 椭圆曲线门限密码算法 [J]. 密码学报，2014，1 (2)：155－166.

[6] 邱卫东，黄 征. 密码协议基础 [M]. 北京：高等教育出版社，2008.

[7] Shamir A. How to share a secret [J]. Communications of the ACM，1979，22 (11)：612－613.

[8] 庞辽军，詹 阳，王育民. 基于 ECC 的门限秘密共享方案及安全性 [J]. 西南电子科技大学学报（自然科学版），2006 (33)：572－575.

[9] 秦志光，张险峰，周世杰，刘锦德. 基于 ECC 的门限数字签名方案及其安全新 [J]. 电子科技大学学报，2005 (34)：109－112.

[10] 林璟锵，马 原，荆继武，等. 适用于云计算的基于 SM2 算法的签名及解密方法和系统 [P]. 中国发明专利，CN104243456A，2014－08－29.

[11] 毛文波著，王继林，等译，现代密码学理论与实践 [M]. 北京：电子工业出版社，2004.

[12] 王婷婷，侯书会. 门限签名方案的研究及其安全性分析 [J]. 计算机工程与应用，2018，54 (13)：123－130.

[13] 黄梅娟. 两个标准模型下门限签名方案的安全性分析 [J]. 计算机与数字工程，2013，41 (8)：1318－1320.

[14] Chung Y F，Chen T S，Chen T L. An efficient threshold signature scheme resistible to conspiracy attack [J]. Applied Mathematics & Information Sciences，2014，8 (6)：3027－3032.

[15] Kaya K，Selcuk A A. Sharing DSS by the Chinese remainder theorem [J]. Journal of Computational and Applied Mathematics，2014，259：4148－4160.

[16] 杨 阳，朱晓玲，丁 凉. 基于中国剩余定理的无可信中心可验证秘密共享研究 [J]. 计算机工程，2015，41 (2)：122－128.

[17] Harn L，Wang F. Threshold signature scheme without using polynomial interpolation [J]. IJ Network Security，2016，18 (4)：710－717.

(上接第 224 页)

4 总结

通过使用人工智能算法，系统有效提高了传统人脸识别的准确性，同时优化了客户端的工作流程，有效减少了识别时间和识别精度，进一步优化了系统。本系统在小区、公司等场合具有广泛的应用前景。

参考文献：

[1] 哈里斯. 人工智能的应用 [M]. 南京：译林出版社，1992.

[2] 杨玉龙. 人脸识别门禁系统的设计与实现 [D]. 重庆：重庆大学，2014.

[3] 陈存强，姚晋丽. 基于人脸识别技术的门禁系统设计与实现 [J]. 内燃机与配件，2018，12：20－24

[4] 施 勇，冯华丽. 人脸识别技术在门禁系统中的研究与应用 [J]. 烟台职业学院学报，2012，18 (4)：83－87.

[5] Xiang J，Zhu G. Joint Face Detection and Facial Expression Recognition with MTCNN [A]. International Conference on Information Science and Control Engineering [C]. IEEE Computer Society，2017，424－427.

[6] 赵荣刚，贺庆民. 计算机人脸识别技术的应用 [J]. 电子技术与软件工程，2018，4：137－137.

[7] Jha S，Agarwal N，Agarwal S. Towards Improved Cartoon Face Detection and Recognition Systems [J]. 2018，1－5.

[8] 王 森，杨克俭. 基于双线性插值的图像缩放算法的研究与实现 [J]. 自动化技术与应用，2008，27 (7)：44－45.

[9] 周飞燕，金林鹏，董 军. 卷积神经网络研究综述 [J]. 计算机学报，2017，40 (6)：1229－1251.

[10] Zhang Z，Sabuncu M R. Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels [J]. 2018，10－15.