华南师范大学学报(自然科学版)
Journal of South China Normal University(Natural Science Edition)
2022,54(4):121-128
doi:10.6054/j.jscnun.2022066

高效的两方 ECDSA 门限方案

颜　萌,马昌社*
(华南师范大学计算机学院,广州 510631)

摘要:针对现有的门限 ECDSA 方案存在的计算开销过大、签名效率不高以及通信开销过大的问题,提出了一种高效的两方 ECDSA 门限方案。该方案将签名私钥拆分成 2 个部分,分别由两方保管;利用同态加密技术,每一次协同签名都需要双方用户同时参与签名过程,其中任意一方都无法掌握完整的签名私钥;将签名阶段分为了离线预计算阶段以及在线签名阶段,在离线预计算阶段提前完成了绝大部分计算量,在线签名阶段高效且快速,提高了签名效率。随后,对该方案给出正确性分析、安全证明及效率对比。研究结果表明:高效的两方 ECDSA 门限方案的在线签名阶段有效地避免了花销高昂的同态操作,具有签名效率高、通信代价低和交互轮数少等优势,实用性更高。

关键词:ECDSA; 同态加密; 门限签名方案
中图分类号:TP309　　**文献标志码**:A　　**文章编号**:1000-5463(2022)04-0121-08

An Efficient Threshold Scheme for Two-party ECDSA

YAN Meng, MA Changshe*
(School of Computer Science, South China Normal University, Guangzhou 510631, China)

Abstract: An efficient two-party ECDSA threshold scheme is proposed to fix the problems of existing threshold ECDSA schemes, e.g., some signature protocols having too much computation overhead or too many interaction rounds, leading to low signature efficiency, and some signature protocols having OT (oblivious transfer) to replace the Paillier homomorphic encryption technology, increasing the communication cost by thousands of times. The scheme divides the signature private key into two parts to be kept by two parties respectively. Using the homomorphic encryption technology, each collaborative signature requires both users to participate in the signature process at the same time. In addition, the signature phase is divided into the offline precomputation phase and the online signature phase. Most of the computation is completed in advance in the offline precomputation phase. The online signature phase is efficient and fast, which improves the signature efficiency. The correctness analysis and security proof of the scheme are given, and the two ECDSA schemes proposed by Lindell and this current scheme are compared in terms of theoretical analysis. The results show that the scheme avoids the expensive homomorphic operation in the online signature phase and has the advantages of high signature efficiency, low communication cost, less interaction rounds and higher practicability.

Keywords: ECDSA; homomorphic encryption; threshold signature scheme

在公钥密码技术领域中,密钥对于密码算法来说至关重要,私钥的安全决定着系统以及敏感信息的安全,一旦公钥系统中的私钥被泄露或者丢失,不仅会造成系统出现单点故障,而且在恶意攻击者获得用户私钥之后就可以轻松地获取并篡改敏感信息。由此,SHAMIR[1]提出门限方案,又名秘密共享方案,该方案利用密码技术将需要保护的明文信息进行分割并安全地由不同的参与者存储。随后,DESMEDT 和 FRANKEL[2]正式提出门限签名的概念。随着云计算以及区块链技术的高速发展,系统终端更容易遭受恶意攻击。为了防止权力过度集中,提升计算系统抵抗安全风险的能力,学者们针对

收稿日期: 2021-04-23　　《华南师范大学学报(自然科学版)》网址:http://journal-n.scnu.edu.cn
基金项目: 国家自然科学基金项目(61672243)
*通信作者:马昌社,chsma@163.com

不同的应用场景提出了不同的门限签名方案[3-5]。

1992年,JOHNSON等[6]为了响应NIST对数字签名标准的要求而提出了椭圆曲线数字签名算法(Elliptic Curve Digital Signature Algorithm,ECDSA)。随后,ECDSA算法作为数字签名算法的一种,被广泛用于移动电子商务领域。由于网上交易为现在的主流消费方式,基于ECDSA算法设计出高效率的门限方案势在必行。

现阶段,围绕ECDSA算法的门限化签名工作出现了大量研究。1996年,GENNRAO等[7]提出(t,q)门限ECDSA签名方案,该方案的门限值$t\leqslant q/2$,且签名的计算和通信开销高。2001年,MACKENZIE和REITER[8]提出了第1个两方ECDSA门限签名方案,该方案在密钥生成过程中利用乘法秘密分享以及Paillier加法同态加密技术来解决ECDSA门限化工作中的难点,使得2个签名参与方能协同生成有效签名。GENNARO等[5]为了解决比特币钱包安全的问题,设计了基于门限加法同态加密技术的(t,q)门限方案,并在恶意模型下给出了安全证明。随后,BONEH等[9]优化了文献[5]的方案,提出了level-1全同态加密的门限签名方案。但是,这些方案都采用了计算开销和通信开销都非常高的分布式同态加密密钥生成技术,在实际应用中受限。

LINDELL[10]提出的两方方案由于不需要执行分布式Paillier密钥算法,减少了部分的计算开销,与以前普通的DSA签名的门限方案相比,效率有一定的提升。相反,DOERNER等[11]没有采取同态加密方案而引入了不经意传输技术,提出了满足安全差异性需求的$(2,n)$门限ECDSA签名方案。虽然文献[11]的方案未利用同态加密技术,提高了协同签名的计算性能,但是该方案引入了不经意传输技术[12-13],与文献[10]的方案相比,增加了近千倍的通信开销。近些年,国内学者在ECDSA门限化的研究上也取得了一些进展,如:王婧等[14]利用了Beaver三元组,提出了一种安全高效的两方协同ECDSA签名方案。

为了进一步提升两方ECDSA门限方案[10]的效率,本文提出了一个高效的两方ECDSA门限方案。该方案主要有以下优势:(1)在每一次对消息M进行签名时,签名双方在保证不泄露自己签名私钥份额的前提下共同生成有效签名;(2)不依赖待签名消息完成签名预处理,减少了签名过程中所产生的计算量,提升了签名效率。

1 预备知识

在本文中,循环群G的阶为素数q,有限域F_p有p个元素,由其中的元素可定义有限域F_p上的一条椭圆曲线E,P为椭圆曲线E上的一个基点;$x\leftarrow X$表示从集合X中(或者是从X分布中)随机均匀采样x;$H(\cdot)$表示哈希杂凑函数,其表现形式为$H:\{0,1\}^*\rightarrow Z_q$;对于一个整数$N$,$\mathbb{Z}_N=\{1,2,\cdots,N-1\}$,$\mathbb{Z}_N^*=\{x\mid 0<x<N \text{ 且 } \gcd(x,N)=1\}$。

1.1 ECDSA算法标准

ECDSA算法是DSA算法的变体,利用了椭圆曲线加密算法(Elliptic Curve Cryptography,ECC)对DSA算法进行模拟。与普通的离散对数问题和大整数分解问题相比,因为椭圆曲线密码是目前唯一无法用亚指数算法破解的公钥密码,所以椭圆曲线密码的单位比特强度高于其他公钥密码体制。1999年,ECDSA算法成为ANSI标准,是目前应用最广泛的签名算法之一。以下给出对ECDSA算法的形式化定义。

定义1[6] (ECDSA)设$H(\cdot)$为哈希杂凑函数,待签名消息为M,所采用的椭圆曲线参数$D=(E,G,P,p,q,h)$,对应的密钥对为(x,Q),其中,$Q=x\cdot P$为公钥,x为私钥。

签名步骤:

(1)选择一个随机数$k\leftarrow Z_q$;

(2)计算$R\leftarrow k\cdot P$,并令$R=(r_x,r_y)$;

(3)计算$r=r_x \bmod q$,若$r=0$,则重新从第(1)步开始执行;

(4)计算待签名消息M的哈希值$H(M)$;

(5)计算签名$s=k^{-1}(H(M)+xr) \bmod q$,若$s=0$,则重新从第(1)步开始执行;

(6)输出对消息M的签名(r,s)。

验证步骤:验证方在接收到消息M和签名(r,s)之后,进行如下运算:

(1)计算$sP+H(M)Q=(x_1,y_1)$;

(2)当且仅当$x_1 \bmod q==r$时,验证成功。

1.2 Paillier同态加密方案

Paillier同态加密方案PC=(PK,PE,PD)是基于复合剩余类的困难问题来保证加密方案的安全性的概率公钥加密算法[15]。该方案的描述如下:

(1)密钥生成算法PK:任选2个长度相同且满足$\gcd(pq,(p-1)(q-1))=1$的大素数p和q,然后计算$N=pq$,令$\lambda(N)=\mathrm{lcm}(p-1,q-1)$为$N$的卡迈可尔函数,并且任意选择整数$g\in\mathbb{Z}_{N^2}^*$。令$L(x)=(x-1)/N$,计算$\mu=(L(g^{\lambda(N)} \bmod N^2))^{-1} \bmod N$。生成Paillier加密方案的公钥$\mathrm{ppk}=(N,g)$,私钥$\mathrm{psk}=(\lambda(N),\mu)$。

(2)加密算法PE:选择随机数$r\in\mathbb{Z}_N^*$,然后计算密文$C=E(m,r)=g^m r^N \bmod N^2$,其中$m\in\mathbb{Z}_N$为待加

密信息。

(3)解密算法 PD：针对密文 C，对其进行解密得到明文 $m=L(C^{\lambda} \bmod N^2)\times\mu \bmod N$。

Paillier 加密方案满足加法同态加密性质。对于任意 2 个明文 $a,b \in \mathbb{Z}_N$，其对应的密文分别为 $e_a=\mathrm{PE}(\mathrm{ppk},a)=g^a r_1^N \bmod N^2$ 和 $e_b=\mathrm{PE}(\mathrm{ppk},b)=g^b r_2^N \bmod N^2$，其中随机数 $r_1,r_2 \in \mathbb{Z}_n^*$。定义 $e_a \otimes e_b=(e_a \times e_b) \bmod N^2=g^a(r_1)^N \otimes g^b(r_2)^N=g^{a+b}(r_1 r_2)^N \bmod N^2=\mathrm{PE}(a+b)$，则 $e_a \otimes e_b$ 为 $a+b$ 的密文。定义 $(e_a)^b=\underbrace{e_a \otimes e_a \otimes \cdots \otimes e_a}_{b\text{个}}$。

1.3　门限签名算法

1991 年，DESMEDT 和 FRANKEL[16] 提出了第一个真正的门限加密以及签名方案。一个 (t,n) 门限签名方案中有 n 个成员参与分布式签名，至少需要 $t+1$ $(t+1\leqslant n)$ 个成员共同参与来生成签名，如成员人数少于或者等于门限数量 t 则无法生成有效签名。该方案通过将私钥信息分割并由多个用户分散保存，提高了系统的鲁棒性及安全性。1 个 (t,n) 门限方案可以分为如下 3 个子协议：

(1)分布式密钥生成协议 Thresh-KeyGen。该协议通过输入安全参数 1^{λ}，每个参与签名的成员 P_i 会获得公钥 Pk 以及对应的私钥份额 sk_i，则 sk_1，$\mathrm{sk}_2,\cdots,\mathrm{sk}_n$ 是关于私钥 sk 的 (t,n) 门限秘密共享。

(2)分布式签名协议 Thresh-Sig。此协议将待签名的信息 M 作为公共输入，同时将参与签名成员的私钥份额 sk_i 作为私有输入，σ_i 为每个签名成员对信息 M 的签名。该协议结束后，将所有的签名份额 σ_i 合并后输出签名 $\sigma \in \{\mathrm{Sig}(\mathrm{sk},m)\}$。

(3)中心化验证算法 Ver。输入待签名消息 M、公钥 Pk 和签名 σ，以检查 σ 是否正确。若验证算法 Ver 输出 1，则签名验证成功。

1.4　安全模型和定义

1.4.1　*敌手模型*　根据 GENNARO 等[7] 对敌手模型进行的描述可知，假定恶意敌手 $\mathcal{A}$ 在 (t,n) 门限方案签名阶段至少可以攻击 n 个成员 $P_1,P_2,\cdots,P_n$ 中的 t 个成员，然后根据攻击能力的大小将敌手分为以下 3 种类型：

(1)窃听敌手：能够获取被攻击成员所存储的信息以及接收其通信信道的广播信息。

(2)中止敌手：不但拥有窃听敌手的能力，而且可以促使被攻击成员在每轮签名开始时停止发送消息。

(3)恶意敌手：不但拥有窃听敌手的能力，而且可以促使被攻击成员修改协议。

假设敌手 $\mathcal{A}$ 的计算能力是可以被概率多项式时间(Probabilistic Polynomial Time，P.P.T)图灵机所模拟的，这就意味着敌手解决椭圆曲线上的离散对数困难问题是不可行的。敌手类型又可分为静态敌手和自适应敌手，二者的区别在于在不同的时间点选择要攻击的用户。下面在讨论本文门限签名方案时，只考虑静态敌手，即在协议开始执行前就选择好要攻击的用户，在协议执行中角色不会转变。

1.4.2　*安全定义*　本文定义的门限签名方案的安全性仅考虑不可伪造性，其定义如下：

定义 2　((t,n) 门限签名方案的不可伪造性)令 $\varepsilon=(\mathrm{Thresh-KeyGen},\mathrm{Thresh-Sig},\mathrm{Ver})$ 为 (t,n) 门限签名方案，称其是不可伪造的，如果敌手 $\mathcal{A}$ 可以自适应地选择 k 次待签名信息 $M_1,M_2,\cdots,M_k$ 进行门限签名查询之后，能够在新的待签名信息 M' ($M' \in \{M_1,M_2,\cdots,M_k\}$)上伪造有效的门限签名的概率是可忽略的。

2　高效的两方 ECDSA 门限方案

本文提出的高效的两方 ECDSA 数字签名方案共包括 3 个部分，分别为密钥生成算法 TPKG、两方签名算法 TPsign 和签名验证。

2.1　密钥生成算法 TPKG

在密钥生成阶段，由两方共同生成数字签名算法中用于验证签名的公钥和各方的部分签名私钥片，同时用户 1 调用同态加密方案，将其私钥的密文发送给用户 2。如图 1 所示，令 U_1、U_2 分别表示为用户 1、用户 2，两方分别执行以下步骤：

Step 1. 首先，用户 U_1 选择随机数 $x_1 \leftarrow Z_q$ 作为子私钥，并且计算子公钥片 $P_1=x_1 \cdot P$，其中 P 是 ECDSA 椭圆曲线的基点；然后，用户 U_1 调用 1.3 节 Paillier 同态加密方案 PC=(PK,PE,PD)，该同态加密方案的公钥、私钥分别为 ppk、psk，将其所拥有的子私钥利用 Paillier 同态加密方案进行加密，其表示为 $\mathrm{ex}_1=\mathrm{PE}(\mathrm{ppk},x_1)$。

Step 2. 用户 U_1 将子公钥片 P_1 和子私钥同态加密密文 ex_1 发送给用户 U_2。

Step 3. U_2 同样随机选择 $x_2 \leftarrow Z_q$ 作为子私钥，并计算子公钥片 $P_2=x_2 \cdot P$。

Step 4. 用户 U_2 将子公钥片 P_2 发送给用户 U_1。

Step 5. 用户 U_1 在收到用户 U_2 发送的公钥份额 P_2 后，计算公钥 $\mathrm{Pk}=P_2+P_1$，并存储参数 $\{\mathrm{Pk},P_2,$

ppk,psk}。

Step 6. 用户U_2在收到用户U_1发送的公钥份额P_1后,计算公钥 $Pk=P_1+P_2$,并存储参数{Pk,P_1,ppk,ex_1}。

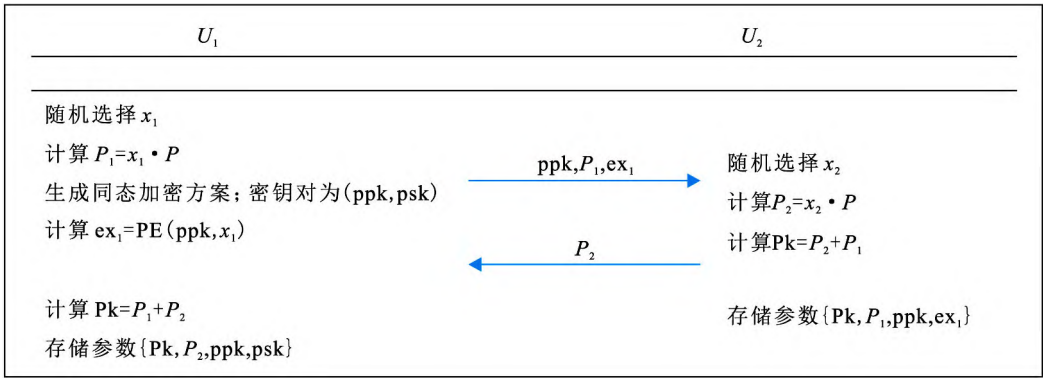

图1 密钥生成算法 TPKG

Figure 1 The key generation algorithm TPKG

2.2 两方签名算法 TPsign

签名生成阶段包括2个步骤:离线预处理过程和在线签名过程(图2、图3)。离线预处理过程不依赖待签名消息,在正式签名之前就可以生成所必需的数据,从而提高签名效率。正式签名时,一旦接收到待签名消息 M,签名者可以高效地生成对消息 M 的签名。两方签名算法 TPsign 的详细过程如算法1和算法2。

算法1 离线阶段的预处理算法

Step 1. 用户U_1生成随机数$k_1 \leftarrow Z_q$,并计算$R_1=k_1 \cdot P$;

Step 2. 用户U_1将R_1发送给用户U_2;

Step 3. 用户U_2选择随机数$k_1 \leftarrow Z_q$,$b \leftarrow Z_q$和$\rho \leftarrow Z_{q^2}$;

Step 4. 用户U_2计算$R_2=k_2 \cdot P$;

Step 5. 用户U_2计算 $t=k_2^{-1} \bmod q$;

Step 6. 用户U_2利用在密钥生成阶段从用户U_1获得的同态加密公钥来计算$e_b=PE(ppk,b+\rho q)$,此处为了让传递的信息更加安全,用户U_2将$\rho \cdot q$与b一起进行同态加密;

Step 7. 用户U_2进行同态操作$\overline{ex}_1=(e_b \otimes x_1)^t$;

Step 8. 用户U_2利用从用户U_1获得的R_1计算$(x,y)=k_2 \cdot R_1$;

Step 9. 计算 $r=x \bmod q$;

Step 10. 用户U_2将$\overline{ex}_1$、R_2发送给用户U_1;

Step 11. 用户U_1利用从用户U_2获得的R_2计算$(x,y)=k_1 \cdot R_2$;

Step 12. 计算 $r=x \bmod q$;

Step 13. 用户U_1利用同态加密方案中的私钥进行解密,获得$\bar{x}=PD(psk,\overline{ex}_1)$;

Step 14. 用户U_1存储参数{x_1,Pk,ppk,psk,r,$\bar{x}$,k_1};

Step 15. 用户U_2存储参数{x_2,Pk,ppk,ex_1,k_2,r,b}。

算法2 在线签名算法

Step 1. 用户U_2收到待签名消息 M 后,计算待签名消息 M 的哈希值 $h=H(M)$;

Step 2. 用户U_2在本地计算部分签名片 $ps=k_2^{-1}(h+x_2 \cdot r-b \cdot r) \bmod q$;

Step 3. 用户U_2将 ps 发送给用户U_1;

Step 4. 用户U_1计算对消息的正式签名 $s=k_1^{-1}(ps+\bar{x} \cdot r) \bmod q$;

Step 5. 用户U_1输出签名 $\sigma=(r,s)$。

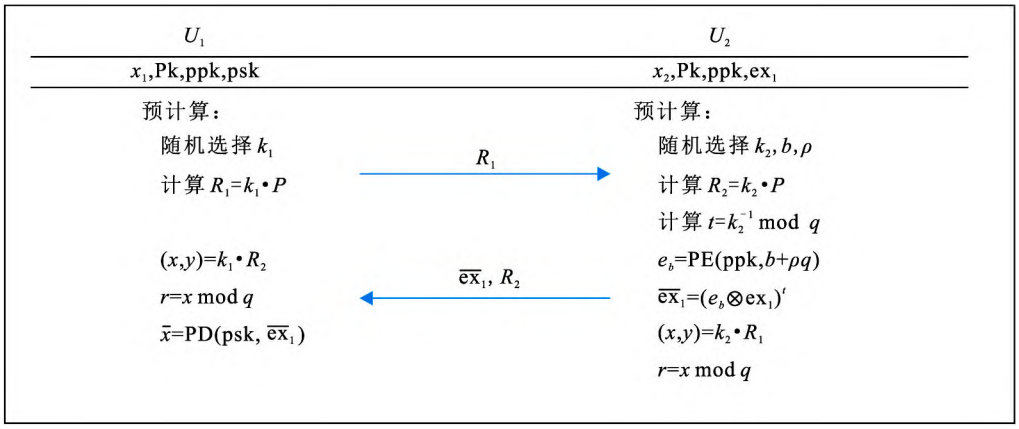

图2 签名离线预处理步骤图

Figure 2 Step diagram of signature offline preprocessing

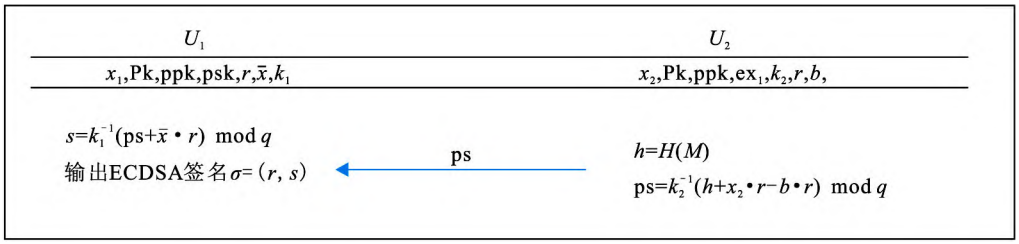

图 3　在线签名步骤图

Figure 3　The steps of online signature

2.3　签名验证

本方案的签名验证过程详细表述如下：

Step 1. 用户U_1利用签名验证算法对得到的签名$\sigma=(r,s)$进行验证：通过计算$sP+H(M)\mathrm{Pk}=(x,y)$来验证是否满足$x \bmod q==r$。若验证失败，则终止协议。

Step 2. 用户U_2收到签名$\sigma=(r,s)$后，采用与用户U_1相同的计算方式来验证签名。如果 2 个通信方的验证都通过，则表明此次两方 ECDSA 签名有效，正常退出，否则终止操作。

3　方案分析

3.1　正确性分析

根据分布式密钥生成算法，可得签名验证公钥：

$$\mathrm{Pk}=P_1+P_2=x_1\cdot P+x_2\cdot G=(x_1+x_2)\cdot P。$$

根据两方协同签名算法，可得

$$R=R_1\cdot k_2=R_2\cdot k_1=k_1\cdot k_2\cdot P。$$

此外，$\mathrm{sk}=x_1+x_2$，$k=k_1\cdot k_2$，可得

$$\mathrm{Pk}=\mathrm{sk}\cdot P,R=k\cdot P=(x,y),r=x \bmod q,$$

$$s=k_1^{-1}(\mathrm{ps}+\bar{x}\cdot r+\rho\cdot q) \bmod q=k^{-1}(h+\mathrm{sk}\cdot r)。$$

由此可知，本文提出的两方 ECDSA 签名方案的输出签名(r,s)和验证公钥 Pk 与 ECDSA 签名方案的输出签名(r,s)和验证公钥 Q 一致。所以，本文方案满足设计目标要求的正确性。

3.2　安全性分析

引理 1[7]　若签名方案是不可伪造的，并且它的门限签名方案是可模拟的，则该门限签名方案也是不可伪造的。

下面给出可模拟的概念。

定义 3　一个(t,n)门限签名方案需要满足以下条件才可以说是可模拟的：

(1) 门限签名方案的密钥生成协议 Thresh-KeyGen 是可模拟的。保证在输入公钥 Pk 和被破坏的 $t-1$ 个成员的私钥份额 $\mathrm{sk}_1,\mathrm{sk}_2,\cdots,\mathrm{sk}_{t-1}$ 的条件下，存在一个能够模拟其他人在 Thresh-KeyGen 协议输出公钥 Pk 的交互视图的模拟器。

(2) 门限签名方案的分布式签名协议 Thresh-Sig 是可模拟的。在输入公钥 Pk，待签名消息 M 以及对它的数字签名 σ，还有 $t-1$ 个成员的私钥份额 $\mathrm{sk}_1,\mathrm{sk}_2,\cdots,\mathrm{sk}_{t-1}$ 的条件下，存在一个能够模拟其他人在 Thresh-Sig 协议中输出数字签名 σ 的交互视图的模拟器。

定理 1　如果 ECDSA 是 EUF-CMA 安全的，则本文的两方 ECDSA 签名方案是不可伪造的。

证明　根据引理 1，只需要证明两方 ECDSA 门限签名方案是可模拟的。由于本方案只存在 2 个用户U_1和U_2，所以，仅考虑用户U_1被攻击和用户U_2被攻击 2 种情况，并分别展示如何模拟密钥生成和签名生成协议。

情形 1：用户U_1被攻击。假设敌手$\mathcal{A}_1$攻击并控制了用户U_1，再构造模拟器$\mathcal{S}im_1$来模拟用户U_2，通过提取用户U_1的输入，生成一个敌手不可区分的交互视图。

(1) 模拟密钥生成。假设敌手$\mathcal{A}_1$破坏了用户U_1，则由于模拟器$\mathcal{S}im_1$知道用户U_1的私钥份额 x_1 及系统的公钥 Pk，模拟器可以通过计算$P_2=\mathrm{Pk}-x_1\cdot P$，模拟出用户$U_2$在密钥生成算法 TPKG 中输出公钥 Pk 的视图。由于模拟器知道私钥份额 x_1，所以也可以计算出 $\mathrm{ex}_1=\mathrm{PE}(\mathrm{ppk},x_1)$。因此，密钥生成算法 TPKG 是可模拟的。

(2) 模拟签名生成。

①模拟器$\mathcal{S}im_1$接受一个关于消息 M 的 ECDSA 签名 $\sigma=(r,s)$；

②模拟器$\mathcal{S}im_1$使用 ECDSA 验证算法来计算 $R=s\cdot P+r\cdot\mathrm{Pk}$；

③若第 1 条会话信息被解析为验证$R_1=k_1\cdot P$，则模拟器$\mathcal{S}im_1$令$R_2=k_1^{-1}\cdot R$，否则模拟器$\mathcal{S}im_1$随机地选取R_2；

④第2条会话信息被解析为R_2、$\overline{\mathrm{ex}}_1$、ps。模拟器$\mathcal{S}im_1$随机选取ps的值并计算$\bar{x}=s\cdot k_1-\mathrm{ps}$,则可以由$\bar{x}$计算出传递的信息$\overline{\mathrm{ex}}_1=\mathrm{PE}(\mathrm{ppk},\bar{x})$;

⑤模拟器$\mathcal{S}im_1$输出正确的ECDSA签名$\sigma=(r,s)$并终止模拟签名过程。

观察以上模拟的交互过程可知,敌手$\mathcal{A}_1$的视图分布与真实协议执行环境下的输出是不可区分的。究其原因为:在交互过程中计算公钥Pk时,真实协议执行中,是由诚实用户U_2随机选择私钥份额x_2,计算公钥片$P_2=x_2\cdot P$;在模拟过程中,模拟器$\mathcal{S}im_1$计算公钥片$P_2=\mathrm{Pk}-x_1\cdot P$。由于Pk是随机选择的,故$x_2\cdot P$和$\mathrm{Pk}-x_1\cdot P$是同分布的。因此,模拟器$\mathcal{S}im_1$的输出与诚实用户$U_2$的输出分布是相同的。

在签名阶段,敌手$\mathcal{A}_1$的视图分布与真实协议执行环境下的不同之处在于ex_1、$\overline{\mathrm{ex}}_1$和ps的生成方式。具体来说:(1)在模拟执行环境中,$R_2=k_1^{-1}\cdot R$,其中k_1是随机数。但是在真实环境中$R_2=k_2\cdot P$,其中,k_2是随机数。因为敌手无法获取k_2的信息,所以二者都满足均匀随机分布,敌手无法有效区分这2种情况。(2)在真实环境下计算$\overline{\mathrm{ex}}_1=(e_b\otimes\mathrm{ex}_1)^t$,其中$t=k_2^{-1}\bmod q$,而模拟器$\mathcal{S}im_1$使用随机选取的ps,然后计算$\bar{x}=(s\cdot k_1-\mathrm{ps})$,便可以得到传递的信息$\overline{\mathrm{ex}}_1=\mathrm{PE}(\mathrm{ppk},\bar{x})$。显然,真实环境与模拟执行环境下生成的签名都为$\sigma=(r,s)$,敌手无法做出区分。综上所述,对于敌手$\mathcal{A}_1$来说,真实的签名协议执行过程和模拟的执行过程是无法区分的。

情形2:用户U_2被攻击。假设敌手$\mathcal{A}_2$攻击并控制了用户U_2,现在构造模拟器$\mathcal{S}im_2$来模拟用户U_1,通过提取用户U_2的输入,生成一个敌手不可区分的交互视图。

(1)模拟密钥生成。与用户U_1被攻击的情况相同,假设敌手$\mathcal{A}_2$破坏了用户U_2,那么由于模拟器知道用户U_2的私钥份额x_2和系统的公钥Pk,则模拟器可以通过计算$P_1=\mathrm{Pk}-x_2\cdot P$来模拟用户$U_1$在密钥生成算法TPKG中输出公钥Pk的视图。所以,密钥生成算法TPKG是可模拟的。

(2)模拟签名生成。

①模拟器$\mathcal{S}im_2$接受一个关于消息M的ECDSA签名$\sigma=(r,s)$。

②模拟器$\mathcal{S}im_2$使用ECDSA算法来计算点$R=s\cdot P+r\cdot Pk$,并向敌手$\mathcal{A}_2$发送指令,指示其开始执行签名协议。

③第1条会话消息被解析为敌手$\mathcal{A}_2$向模拟器$\mathcal{S}im_2$发送R_2,模拟器$\mathcal{S}im_2$验证是否满足$R_2=k_2\cdot P$,且判断R_2是否为椭圆曲线上的非零点。若不成立,模拟器$\mathcal{S}im_2$中止协议;否则,模拟器$\mathcal{S}im_2$计算$R_1=k_2^{-1}\cdot R$并发送给敌手$\mathcal{A}_2$。

④第2条会话信息被解析为:模拟器$\mathcal{S}im_2$计算$\mathrm{ps}=k_2^{-1}(h+x_2\cdot r-b\cdot r)\bmod q$和$\overline{\mathrm{ex}}_1=(e_b\otimes\mathrm{ex}_1)^t$。

⑤第3条会话消息被解析为$\bar{x}$和ps,模拟器$\mathcal{S}im_2$设置给敌手$\mathcal{A}_2$的预言机回答为s。

若敌手$\mathcal{A}_2$诚实执行协议,则会输出正确的ECDSA签名$\sigma=(r,s)$。

观察以上模拟的交互过程可知,敌手$\mathcal{A}_2$的视图分布与真实协议执行环境下密钥生成阶段的输出是不可区分的,与用户U_2被攻击的情况相似,由于Pk是随机选择的,故$x_1\cdot P$和$\mathrm{Pk}-x_2\cdot P$是同分布的。因此,其密钥生成协议是可模拟的。

在签名过程中,具体来说:(1)在模拟执行环境中,$R_1=k_2^{-1}\cdot R$;在真实环境中,$R_1=k_1\cdot P$,其中k_1为随机数。因为敌手无法获取k_1的信息,所以二者都满足均匀随机分布,敌手无法有效区分这2种情况。因此,其签名生成协议是可模拟的。(2)在真实环境中,$\overline{\mathrm{ex}}_1=(e_b\otimes\mathrm{ex}_1)^t$,其中,$\mathrm{ex}_1=\mathrm{PE}(\mathrm{ppk},x_1)$,$t=k_2^{-1}\bmod q$。而在模拟环境中,由于用户$U_2$的信息都被暴露,包括$b$、$k_2$、$x_2$,所以与真实环境下的执行结果是一样的。由于签名预言机生成的(r,s)一定满足随机均匀的特性,且k_1也是随机选择的,两者的执行结果对于敌手$\mathcal{A}_2$来说都是一样的。

综上所述,根据引理1,无论是用户U_1还是用户U_2被攻击,都可以构造相应的模拟器来模拟其协议的整个运行过程,即可证明ECDSA门限签名方案具有不可伪造性。证毕。

3.3 效率分析

在现有的两方ECDSA方案中,LINDELL提出的方案[10]比DOERNER提出的方案[11](下文分别简称为LINDELL方案、DOERNER方案)的效率更高,究其原因为:DOERNER方案采用OT(Oblivious Transfer)技术来替代Paillier同态加密技术,在实际应用中,一次k比特的OT运算需要$O(k)$次公钥密码操作,与同态加密技术相比,其计算量更大。由于本文提出的方案与LINDELL方案[10]相类似,所以接下来将与LINDELL方案进行效率对比。

本文提出的方案暂为一个基础的两方 ECDSA 门限方案，其安全性只达到被动安全级别，但同样可以利用 LINDELL 方案中所采取的理想函数实现主动安全。在进行效率分析时，忽略零知识证明，在半诚实模型下与本文方案进行比较，并列举出 2 个方案的指数运算次数。

在分析计算量时，本文只讨论点乘运算、Paillier 加密及解密计算的计算量，而忽略其他计算量，并且把一次椭圆曲线上的点乘运算的时间记为T_d、Paillier 加密或解密计算时间记为T_p。由 2 个方案的效率对比结果（表 1）可知本文方案的在线签名效率远远优于 LINDELL 方案：（1）在密钥生成阶段：在本文方案中，用户U_1计算$P_1=x_1\cdot P$ 和$\mathrm{ex}_1=\mathrm{PE}(\mathrm{ppk},x_1)$，计算时间为$1\cdot T_d+1\cdot T_p$；用户$U_2$计算$P_2=x_2\cdot P$，计算时间仅为$1\cdot T_d$。而在 LINDELL 方案中，用户$U_1$计算$Q_1=x_1\cdot P$，$Q=x_1\cdot Q_2$和 Paillier 加密操作$c_{\mathrm{key}}=\mathrm{Enc}_{\mathrm{pk}}(x_1)$，计算时间为$2\cdot T_d+1\cdot T_p$；用户$U_2$计算$Q_2=x_2\cdot P$ 和$Q=x_2\cdot Q_1$，计算时间仅为$2\cdot T_d$。（2）在离线预计算阶段：在本文方案中，用户U_1计算$R_1=k_1\cdot P$ 和$(x,y)=k_1\cdot R_2$，并对$\overline{\mathrm{ex}}_1$进行解密$\bar{x}=\mathrm{PD}(\mathrm{psk},\overline{\mathrm{ex}}_1)$，计算时间为$2\cdot T_d+1\cdot T_p$；用户$U_2$计算$R_2=k_2\cdot P$ 和$(x,y)=k_2\cdot R_1$，并进行 Paillier 加密操作$e_b=\mathrm{PE}(\mathrm{ppk},b+\rho q)$和 Paillier 同态操作$\overline{\mathrm{ex}}_1=(e_b\otimes\mathrm{ex}_1)^t$，计算时间为$2\cdot T_d+2\cdot T_p$。在 LINDELL 方案中，用户$U_1$计算$R_1=k_1\cdot P$ 和$R=k_1\cdot R_2$，计算时间为$2\cdot T_d$；用户U_2计算$R_2=k_2\cdot P$ 和$R=k_2\cdot R_1$，并进行同态操作$C_2=k_2^{-1}\cdot x_2\cdot r\odot c_{\mathrm{key}}$，计算时间共计$2\cdot T_d+1\cdot T_p$。（3）在在线签名阶段：在本文方案中，用户$U_1$、$U_2$都只需要进行线性运算，所以两方在此阶段的点乘运算次数为 0。在 LINDELL 方案中，用户U_1需要对接收的密文C_3进行同态解密，计算时间为$1\cdot T_p$；用户U_2需要对信息加密$C_1=\mathrm{Enc}_{\mathrm{pk}}(\rho\cdot q+[k_2^{-1}\cdot m' \bmod q])$，计算时间为$1\cdot T_p$。综上可知，由于本文方案在离线预计算阶段完成了绝大部分计算量，从而在线签名时仅需要简单的操作，而 LINDELL 方案在线计算还需要一些 Paillier 加密操作。相比之下，本文方案在线签名阶段的计算量是几乎可以忽略的。

表 1　2 种方案的计算效率

Table 1　The calculation efficiency of the two schemes

签名阶段	本文方案		LINDELL 方案	
	U_1	U_2	U_1	U_2
密钥生成阶段	$1\cdot T_d+1\cdot T_p$	$1\cdot T_d$	$2\cdot T_d+1\cdot T_p$	$2\cdot T_d$
离线预计算	$2\cdot T_d+1\cdot T_p$	$2\cdot T_d+2\cdot T_p$	$2\cdot T_d$	$2\cdot T_d+1\cdot T_p$
在线签名	0	0	$1\cdot T_p$	$1\cdot T_p$

4　结论

现有的两方 ECDSA 签名方案不是存在计算开销过大的问题就是交互轮数过多，导致在实际应用中的效率并不高。为了提高协同签名效率，本文提出了一种高效的两方 ECDSA 门限签名方案，主要是将两方签名算法拆分为离线预计算算法和在线签名算法，并且证明了其不可伪造性。与文献[10]的两方 ECDSA 方案相比，本文提出的方案计算效率高，其离线预计算阶段完成了大部分的计算量，从而在线签名阶段仅仅需要简单的操作。本文方案虽然只具有被动安全，但是在通用可组合安全框架下可以实现主动安全。

参考文献：

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

[2] DESMEDT Y, FRANKEL Y. Threshold cryptosystems[C]// Proceedings of 89th Annual International Cryptology Conference. Berlin: Springer, 1989: 307-315.

[3] SHOUP V, GENNARO R. Securing threshold cryptosystems against chosen ciphertext attack[J]. Journal of Cryptology, 2002, 15(2): 75-96.

[4] FOUQUE P, POUPARD G, STERN J. Sharing decryption in the context of voting or lotteries[C]// Proceedings of Financial Cryptography—FC 2000. Berlin: Springer, 2000: 90-104.

[5] GENNARO R, GOLDFEDER S, NARAYANAN A. Threshold-optimal DSA/ECDSA signatures and an application to

Bitcoin wallet security[C]//Proceedings of 16th International Conference on Applied Cryptography and Network Security. Berlin:Springer,2016:156-174.

[6] JOHNSON D,MENEZES A,VANSTONE S. The elliptic curve digital signature algorithm(ECDSA)[J]. International Journal of Information Security,2001,1(1):36-63.

[7] GENNARO R,JARECKI S,KRAWCZYK H,et al. Robust threshold DSS signatures[C]//Proceedings of 98th International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 1996: 354-371.

[8] MACKENZIE P,REITER M K. Two-party generation of DSA signatures[C]//Proceedings of 21st Annual International Cryptology Conference. Berlin:Springer,2001:137-154.

[9] BONEH D,GENNARO R,GOLDFEDER S. Using level-1 homomorphic encryption to improve threshold DSA signatures for Bitcoin wallet security[C]//Proceedings of the 5th International Conference on Cryptology and Information Security in Latin America. Berlin: Springer, 2017: 352-377.

[10] LINDELL Y. Fast secure two-party ECDSA signing[C]//Proceedings of 37th Annual International Cryptology Conference. Berlin:Springer,2017:613-644.

[11] DOERNER J,KONDI Y,LEE E,et al. Secure two-party threshold ECDSA from ECDSA assumptions[C]//Proceedings of 2018 IEEE Symposium on Security and Privacy (SP). San Francisco:IEEE,2018:980-997.

[12] CHOU T,ORLANDI C. The simplest protocol for oblivious transfer[C]//Proceedings of the 1st International Conference on Cryptology and Information Security in Latin America. Berlin:Springer,2015:40-58

[13] KELLER M,ORSINI E,SCHOLL P. Actively secure OT extension with optimal overhead[C]//Proceedings of 35th Annual Cryptology Conference. Berlin:Springer, 2015: 724-741.

[14] 王婧,吴黎兵,罗敏,等. 安全高效的两方协同ECDSA签名方案[J]. 通信学报,2021,42(2):12-25.
WANG J,WU L B,LUO M,et al. Secure and efficient two-party ECDSA signature scheme[J]. Journal on Communications,2021,42(2):12-25.

[15] PAILLIER P. Public-key cryptosystems based on composite degree residuosity classes[C]//Proceedings of 99th International Conference on the Theory and Applications of Cryptographic Techniques. Berlin:Springer,1999: 223-238.

[16] DESMEDT Y,FRANKEL Y. Shared generation of authenticators and signatures(extended abstract)[C]//Advances in Cryptology—CRYPTO'91. Berlin:Springer,1991:457-469.

【责任编辑:庄晓琼 责任校对:庄晓琼 英文审校:程杰】