严格可证明安全的两方协同SM2签名协议

程一帆[1,2],刘擎宇[2],梁泽宇[2],于 昇[1,2*]
(1. 云海链控股股份有限公司,海南澄迈 571924;2. 牛津(海南)区块链研究院有限公司密码学实验室,海南澄迈 571924)

摘 要: SM2签名算法自提出后得到了广泛的应用,其中电子合同是一个典型的应用场景. 用户在使用电子合同服务签约时,由于单个用户抗攻击能力较弱,存在严重的私钥泄露风险,因此往往将私钥托管在服务商的云端服务器上. 但是这又涉及对服务商的信任问题,甚至直接影响电子合同的合法性. 为了解决这个两难问题,我们基于同态加密的思想提出了一种两方协同SM2签名协议,用户和服务商协同生成并保存各自的私钥分片,在使用时通过线上交互的方式合作生成签名,从而同时解决安全和信任问题. 我们发现,现有的两方协同SM2签名协议的安全性都存在问题或者错误,就我们所知,本协议是第一个严格可证明安全的两方协同SM2签名协议.

关键词: SM2协同签名;可证明安全;电子合同;同态加密;安全多方计算
基金项目: 海南省重大科技计划(No.ZDKJ2020009)
中图分类号: TP309 **文献标识码:** A **文章编号:** 0372-2112(2024)02-0540-10
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20220027

A Two-Party SM2 Signing Protocol with Strict Provable Security

CHENG Yi-fan[1,2], LIU Qing-yu[2], LIANG Ze-yu[2], YU Sheng[1,2*]
(1. *SSC Holding Company Ltd.* , *Chengmai*, *Hainan* 571924, *China*;
2. *Laboratory of Cryptography*, *Oxford-Hainan Blockchain Research Institute*, *Chengmai*, *Hainan* 571924, *China*)

Abstract: Since it was first proposed, the SM2 signature algorithm has become increasingly popular. A typical application scenario is the electronic contract service. Due to the inadequate anti-attack capability of a single user and the high risk of private key leakage, users who use electronic contract services to sign contracts frequently host the private key on the service provider's cloud server. However, this calls for consumers to have faith in service providers, and it will even impact the contract's legitimacy. We suggest a two-party SM2 signing protocol based on the concept of homomorphic encryption to address this conundrum. In order to simultaneously address the issues of security and trust, users and service providers work together to create and save their own private key fragments as well as generate signatures through online interaction. We discover that the two-party SM2 signing protocols currently in use have flaws or security mistakes. This protocol is the first strictly proven secure two-party SM2 signature protocol that we are aware of.

Key words: SM2 signature; proven secure; digital contract; homomorphic encryption; secure multi-party computation
Foundation Item(s): Finance Science and Technology Project of Hainan province (No.ZDKJ2020009)

1 引言

数字签名算法自从被提出以来在各领域得到了广泛的应用,其中椭圆曲线密码体制(Elliptic Curve Cryptography)具有一系列优点,得到了密码学研究者的深入研究. 1992年,椭圆曲线数字签名算法(Elliptic Curve Digitial Signature Algorithm)首次被提出[1],之后被国际化标准组织采纳为国际标准算法. 我国从2001年开始研究具有自主产权的椭圆曲线密码体制,在广泛吸收国内外相关的研究成果后,国家商用密码管理办公室于2010年12月正式推出了《SM2椭圆曲线公钥密码算法》[2],随后它成为中国商用密码标准. SM2算法在2016年8月成为中国国家密码标准[3],并于2018年11月正式成为国际标准[4]. 这些标志性事件意味着SM2的应用范围将进一步扩大.

电子合同是SM2签名算法的一个典型的应用场

收稿日期:2022-01-03;修回日期:2023-01-31;责任编辑:王天慧
*通讯作者:于昇

景,国内外市场上涌现了一批快速成长的电子合同服务商[5~7]. 市场主体使用数字签名算法在网络上对电子文本签约,不受时空的限制,令存储查询更加便捷,极大地提高了相关工作效率,节省了签约成本. 在这种场景下,签名私钥等同于用户身份,进而与用户在现实世界中的法律权利义务绑定. 因此,私钥的安全性十分重要. 而单个用户的防护能力非常有限,操作系统漏洞、木马病毒、恶意软件等安全隐患非常容易导致私钥泄露. 为了保证私钥的安全,某些电子合同服务商将用户的私钥托管在其云端,但是这种托管意味着用户必须信任服务商不会滥用其私钥. 事实上,这种行为不仅不安全,也不符合《中华人民共和国电子签名法》等相关法律规定,甚至会影响电子合同的合法性. 为了解决这个问题,本文提出了一种两方协同SM2签名协议,用户和电子合同服务商可以各持一份私钥分片,通过算法协议交互后生成签名. 这样既可以保护用户私钥的安全,也可以防止服务商滥用用户的私钥.

可证明安全理论是现代密码学的基础[8],据悉,目前已有的SM2两方协同签名都不是严格可证明安全的[9,10],有的甚至存在严重的错误[9]. 本文给出第一个严格可证明安全的两方协同SM2签名协议.

2　SM2协同签名相关工作

MacKenzie和Reiter[11]首次基于同态加密的思想提出两方ECDSA协同签名协议. Lindell[12]在此基础上进行了改进,大幅减少了零知识证明的使用,极大地提高了协议的效率. Doerner等人[13]首次基于不经意传输协议实现了两方ECDSA协同签名,并将其有效扩展为(2,n)的门限签名协议. 与Lindell[12]的方案相比, Doerner等人[13]的方案具有计算效率高的优点,但也有传输数据量大的缺点. 此后,一些(t,n)门限ECDSA协议[14~18]在这两种主要框架的基础上得以提出.

关于SM2算法,尚铭等人[19]提出了基于秘密分享方案的(t,n)SM2门限签名协议,但是在签名阶段需要2t+1个成员共同操作,不适用于两方协同签名的情况. 侯红霞等人[9]基于Lindell[12]的思想提出了一种SM2两方协同签名协议,但是其协议及证明存在错误,可以被恶意攻击者恢复私钥. 冯琦等人[10]创新性地提出了一种计算效率较高的SM2两方协同签名协议,但是其协议的安全证明存在问题,无法严格地归约到底层困难问题. 针对以上两种协议,本文将在下面具体指出其问题所在. 本文参考Lindell[12]的思想,提出了一种严格可证明安全的两方协同签名协议. 据悉,这是同类SM2协同签名协议中第一个严格可证明安全的.

2.1　侯红霞等学者[9]中存在的错误

与本文类似,侯红霞等人[9]参考了Lindell[12]的主要思想,使用同态加密算法将签名s的计算在两个参与方之间拆分,由第二个参与方首先计算签名s中与其掌握的参数相关部分的密文并发送给第一个参与方,再由第一参与方利用Paillier算法的同态性质在密文上进行相关计算从而生成签名. 但是其构造的协议省略了对Pailier公钥长度的要求,也没有使用零知识证明协议证明有关密文是按照协议规定生成的. 下面我们将详细指出这种缺失导致的严重后果.

(1)Paillier公钥长度

该协议的主要思想是利用Paillier加密体制的同态性质,由其中一个参与方生成Paillier公私钥对(pk,sk),并且使用Paillier公钥对所掌握的私钥相关信息加密并传输给另一方,由另一方利用加法同态的性质在密文上进行操作,最终得到密文$s'=\mathrm{Enc}_{pk}(d(k+r))$(文献[9]第3节). 此时,生成Paillier公私钥对的一方计算$s=\mathrm{Dec}_{sk}(s')-r \bmod q$得到最终的签名.

我们认为,Paillier加密体制的明文取值范围是$0\leqslant m<N$,所有的加解密操作均在模N下进行,在解密之后才能进行模q的运算. 因此,如果N达不到应有的长度,很可能出现$\mathrm{Dec}_{sk}(s')=d(k+r)+\eta q \bmod N\neq d(k+r) \bmod q$的情况. 这样即使参与协议的双方诚实地执行了协议,最终也无法获得正确的签名值.

(2)未证明密文按照规定正确生成

在该协议的协作签名阶段(4.c),U_2计算$c_1=\mathrm{Enc}_{pk}\left(d_2^{-1}\cdot k_2 \bmod q\right)$,$c_2=\mathrm{Enc}_{pk}\left(d_2^{-1}+\eta q\right)$. 在这里,$U_2$没有被要求证明$c_1$和$c_2$是正确生成的. 在此,我们构想一个恶意的攻击者$U_2'$将$c_1$设置为整数0的密文,$c_2$按原协议规定生成. 即$c_1=\mathrm{Enc}_{pk}(0)$,$c_2=\mathrm{Enc}_{pk}\left(d_2^{-1}+\eta q\right)$. 按照原协议步骤(5.$b$),$U_1$计算$s'=\left(\left(d_1^{-1}\cdot k_1\right) \bmod q\right)\odot c_1\oplus\left(\left(d_1^{-1}\cdot r\right) \bmod q\right)\odot c_2=\mathrm{Enc}_{pk}\left(d_1^{-1}\cdot r\cdot\left(d_2^{-1}+\eta q\right)\right)$. U_2'收到s'后即可计算$\mathrm{Dec}_{sk}(s') \bmod q=d_1^{-1}\cdot r\cdot d_2^{-1} \bmod q$. 对于$U_2'$,$d_2$是其本来就掌握的SM2私钥分片,$r$是步骤(5.$a$)中已经计算出的签名参数,显然$U_2'$可以利用这种攻击方式轻而易举地恢复出$U_1$手中的SM2私钥分片$d_1$,从而恢复出整个私钥$d$.

显然,一个不安全的协议不可能有正确的安全证明,因此其安全证明部分也是错误的. 比如,当U_2腐化时,在c_1和c_2的正确性无法保证的情况下,模拟器不可能模拟出一个与真实视图不可分辨的s'. 其证明的错误

之处在于，在真实世界中，当U_2腐化没有正确生成c_1和c_2时，U_1仍然会按照要求计算s'并返回给U_2. 但是在模拟世界中，$\mathcal{S}$则直接模拟U_1退出协议了. 这两种视图显然不是不可分辨的.

2.2 冯琦等学者[10]中存在的问题

冯琦等人[10]提出了一种非常具有创新性的协议构造方式，该协议的密钥生成部分与文献[9]类似，但是不需要生成Paillier密钥对. 其创新点在于在签名协议阶段生成R时引入私钥分片的逆元，因此不需要使用同态加密算法即可计算签名，同时减少了计算开销较大的零知识证明协议的使用，大大提升了签名协议的效率. 但是，我们不得不指出，该协议的安全证明部分存在错误，其安全性基础是不稳固的.

在该协议安全证明部分（第6.2节），当U腐化时，模拟器s在模拟U的视图时随机选择了一个K_s模拟发送给U（步骤2.3）. 在其不可区分性分析部分写道，“K_s在模拟执行环境中是随机生成的，而在真实执行环境中是由随机数k_s计算而来$K_s=k_s\cdot G$，由于敌手无从得知关于k_s的信息，因此二者均满足均匀随机分布，敌手视角不可区分”.

我们认为，这种对视图不可区分性的理解是存在问题的. 在安全多方计算理论中，模拟视图和真实视图的不可区分指的是视图整体变量分布的不可区分，而非单个变量的不可区分[20,21]. 只有单个变量在整个分布中是独立变量时，这种论证才是有道理的. 在这个协议中，U在真实世界中的视图是：$\{k_U, d_S^{-1}\cdot G, k_S\cdot G, r, s'\}$，其中$r=x_1+e \bmod n$，$(x_1,y_1)=d_S^{-1}\cdot(K_S+K_U)$，$s'=d_S\times r+k_S \bmod n$. U在模拟世界中的视图是$\{k_U, d_S^{-1}\cdot G, k_S^{\mathrm{sim}}\cdot G, r, s'\}$，其中$k_S^{\mathrm{sim}}$是在$\mathbb{Z}_q^*$中所取的随机变量. 记$R=(x_1,y_1)$，则$K_S=k_S\cdot G=d_S\cdot R-k_U\cdot G$. 因此，在真实视图中，当$(k_U, d_S^{-1}\cdot G, r)$这三个变量确定时，$k_S\cdot G$是一个确定值而不是一个均匀分布，而在模拟视图中，随机选择的$k_S^{\mathrm{sim}}\cdot G$是一个均匀分布，因此这两种视图的分布是完全不同的. 如果要声称两种视图不可区分的话，按照现代密码学的可证明安全理论，必须引入额外的困难问题假设，并将这两种视图的不可区分性归约到底层的困难问题假设上. 简单地声称敌手无从得知关于k_s的信息就意味着视角不可区分，这种论证方式是不严密的.

3 基础知识

3.1 SM2签名算法

令$\mathbb{G}$为椭圆曲线上的循环群，其生成元为G，阶为q. 取私钥为$d\leftarrow[1,q-2]$，相应的公钥为$Q=d\cdot G$. 对于消息$M\in\{0,1\}^*$，SM2签名算法定义如下：

①计算$e=\mathrm{hash}(Z\|M)$，其中Z为用户的身份标识符，椭圆曲线参数和公钥坐标的哈希值；

②选择随机数$k\leftarrow\mathbb{Z}_q^*$；

③计算$R=k\cdot G$，记为$R=(r_x,r_y)$；

④计算$r=(r_x+e) \bmod q$，若$r=0$或$r+k=q$则重新选择随机数k；

⑤计算$s=(1+d)^{-1}\cdot(k-rd) \bmod q$，若$s=0$则重新选择随机数$k$，否则输出签名$(r,s)$.

在实际使用过程中，经常使用一种变体形式，令$Q=(d^{-1}-1)\cdot G$，则$s=d\cdot(k+r)-r \bmod q$. 在这种形式下，在计算s的过程中不出现求逆操作，从而简化了计算. 本文在后续计算中采用这种变体形式，以后不再说明.

3.2 理想的承诺功能$\mathcal{F}_{\mathrm{com}}$

理想的承诺功能的参与方为P_1和P_2，具体内容如下：

在接收到参与方$P_i(i\in\{1,2\})$发送的指令$(\mathrm{commit},\mathrm{sid},x)$后，记录$(\mathrm{commit},\mathrm{sid},x)$并且向$P_{3-i}$发送$(\mathrm{receipt},\mathrm{sid})$. 若已经存在某个$(\mathrm{commit},\mathrm{sid},x)$，则忽略该消息.

在接收到参与方P_i发送的指令$(\mathrm{decommit},\mathrm{sid})$后，若记录中存在$(\mathrm{sid},i,x)$，则向$P_{3-i}$发送$(\mathrm{decommit},\mathrm{sid},x)$.

任意UC-安全的承诺体制均可用以实现理想承诺功能，如文献[22~24]. 在随机谕言机安全模型下，可以简单地使用$\mathrm{Com}(x)=H(x,r)$；$r\leftarrow\{0,1\}^n$实现.

3.3 理想的零知识功能$\mathcal{F}_{\mathrm{zk}}^R$

关系R的理想零知识功能参与方为P_1和P_2，具体内容如下：

在接收到参与方$P_i(i\in\{1,2\})$发送的$(\mathrm{prove},\mathrm{sid},x,w)$指令后，验证$(x,w)\in R$，并向$P_{3-i}$发送$(\mathrm{proof},\mathrm{sid},x)$. 若$(x,w)\notin R$或者该sid之前已被使用过，则忽略该消息.

我们在此指出，该功能需要零知识的知识证明协议（与零知识证明协议中只需要证明$x\in R_L$不同，零知识的知识证明协议还需要证明者证明其“知道”w. 详见文献[21]第6.3节和第6.5节.）而非单纯的零知识证明协议实现. 我们使用Fiat-Shamir转换[25]实现非交互式的零知识的知识证明协议.

3.4 理想的承诺非交互式零知识功能$\mathcal{F}_{\mathrm{com-zk}}^R$

理想的承诺非交互式零知识功能$\mathcal{F}_{\mathrm{com-zk}}^R$参与方为$P_1$和$P_2$，具体内容如下：

在接收到参与方$P_i(i\in\{1,2\})$发送的$(\mathrm{com-prove},\mathrm{sid},x,w)$指令之后，如果sid之前已被使用，则忽略该消息，否则向P_{3-i}发送$(\mathrm{comproof-receipt},\mathrm{sid})$. 如

果$(x,w)\in R$,记录(sid,i,x)*.

在接收到参与方P_i发送的指令$(\text{decom}-\text{proof},\text{sid})$之后,如果$(\text{sid},i,x)$记录存在,则向$P_{3-i}$发送$(\text{decom}-\text{proof},\text{sid},x)$.

在具体协议中,我们使用承诺功能对非交互的零知识知识证明进行承诺,以实现该功能.

3.5 Paillier加密

Paillier加密体制是由Paillier于1999年提出的一个非对称加密体制[26],包括如下算法:

(1)密钥生成:

随机生成两个大随机素数p,q,并且满足$\gcd\left(p\cdot q,\left(p-1\right)\left(q-1\right)\right)=1$.

计算$N=p\cdot q,\lambda=\text{lcm}(p-1,q-1)$.

随机选择$g\leftarrow Z_{N^2}^*$

检查$\mu=(L(g^\lambda \bmod N^2))^{-1}\bmod N$的逆元是否存在,若不存在,返回上一步. 其中$L(x)=\dfrac{x-1}{n}$.

公钥为$\left(N,g\right)$;私钥为(λ,μ).

(2)加密:

令m为待加密的消息,其中$0\leqslant m<N$.

选取随机的$r\in Z_N^*$.

计算密文$c=g^m r^N \bmod N^2$

(3)解密:

计算$m=L\left(c^\lambda \bmod N^2\right)\cdot\mu \bmod N$.

本文中我们主要使用其加法同态的性质:记公私钥为$\left(\text{pk},\text{sk}\right)$,相应的加解密算法分别为$\text{Enc}_{\text{pk}}(\cdot)$和$\text{Dec}_{\text{sk}}(\cdot)$. 密文上的"加法"记为$c_1\oplus c_2=\text{Enc}_{\text{pk}}(m_1+m_2)$,标量乘法记为$a\odot c=\text{Enc}_{\text{pk}}(a\cdot m)$.

本文在实际使用时将g设置为$N+1$,此时$\mu=\lambda^{-1}\bmod N$总是存在. 在这种选取方式下,公钥为N,以后不再说明.

4 零知识证明

本节中,我们介绍两方ECDSA协议中用到的零知识证明,具体包括一个零知识知识证明和两个零知识证明.

4.1 椭圆曲线离散对数的零知识知识证明

椭圆曲线上的点与其离散对数构成的二元关系为:$R_{\text{DL}}=\{(\mathbb{G},G,q,P,w)|P=w\cdot G\}$. 我们使用Fiat-Shamir转换后的非交互式Schnorr证明[27]实现$\mathcal{F}_{\text{zk}}^{R_{\text{DL}}}$功能.

4.2 Paillier公钥正确生成的零知识证明

Paillier加密体制要求其公钥$N=p\cdot q$,其中p和q为两个大随机素数,且$\gcd\left(N,\phi(N)\right)=1$. 我们在这里只证明$L_P=\{N|\gcd\left(N,\phi(N)\right)=1\}$,因为满足该性质的公钥完全可以实现签名协议对Paillier加密体制的需求[12]. 具体的证明协议如文献[28]中第3.3节所述.

有必要指出的是,这里的证明协议是零知识证明协议而非零知识知识证明协议,故不能用$\mathcal{F}_{\text{zk}}^{\text{R}}$功能描述[21].

4.3 Paillier加密密文的零知识证明

我们还需要使用零知识证明协议证明Paillier密文对应的明文是椭圆曲线上的点的离散对数,且其数值在$\mathbb{Z}_q$范围内:

$L_{\text{PDL}}=\{(c,pk,R_1,\mathbb{G},G,q)|\exists\left(k_1,r\right),$

使$c=\text{Enc}_{pk}\left(k_1;r\right)$且$R_1=k_1\cdot G$且$k_1\in\mathbb{Z}_q\}$

具体协议实现参考文献[12]第六节**. 在文献[12]中,为了改善效率,其范围证明的完备性要求x_1的取值范围是$x_1\in\left[\dfrac{2q}{3},q\right]$而非$x_1\in\mathbb{Z}_q$. 为了安全证明的严格性和叙述逻辑的简洁性,我们使用文献[29]中的范围证明(第1.2节),而非文献[12]中改进后的范围证明.

5 SM2两方协同签名协议

5.1 密钥协同生成协议$(\mathbb{G},G,q)$

(1)P_1的操作:

(a)P_1选择随机数$d_1\leftarrow\mathbb{Z}_q^*$并计算$Q_1=d_1\cdot G$.

(b)P_1向$\mathcal{F}_{\text{com}-\text{zk}}^{R_{\text{DL}}}$发送指令$(\text{com}-\text{prove},1,Q_1,d_1)$.

(2)P_2的操作:

(a)P_2接收$\mathcal{F}_{\text{com}-\text{zk}}^{R_{\text{DL}}}$返回的消息$(\text{comproof}-\text{receipt},1)$.

(b)P_2选择随机数$d_2\leftarrow\mathbb{Z}_q^*$并计算$Q_2=d_2\cdot G$.

(c)P_2向$\mathcal{F}_{\text{zk}}^{R_{\text{DL}}}$发送指令$(\text{prove},2,Q_2,d_2)$.

(3)P_1的操作:

(a)P_1接收$\mathcal{F}_{\text{zk}}^{R_{\text{DL}}}$返回的消息$(\text{proof},2,Q_2)$. 如果没有接收到,则中止协议.

(b)P_1向$\mathcal{F}_{\text{com}-\text{zk}}^{R_{\text{DL}}}$发送指令$(\text{decom}-\text{proof},1)$.

(c)P_1生成一组Paillier密钥对(pk,sk),记pk为N,要求N的长度至少为$\max(3\log|q|+1,n)$.(这里的n是安全参数)

(d)P_1将公钥N发送给P_2.

(4)零知识证明:

*在文献[9,10,12]中均使用了该功能,其定义基本一致,只有在$(x,w)\in R$时才向P_{3-i}发送(proof-receipt,sid). 我们认为,这种定义是错误的,因为在真实协议的承诺阶段,无论$(x,w)\in R$是否成立,另一方都会收到承诺.

**在Lindell[12]的原始版本中,该协议存在错误. 经我们指出后,Yehuda Lindell教授修补了这个错误,详见该论文最新版本https://eprint.iacr.org/2017/552.pdf.

(a) P_1通过零知识证明协议向P_2证明$N \in L_P$.

(5) P_2的验证:

当下列条件全部成立时, P_2继续执行协议,否则中止协议:

(a) P_2接收到$\mathcal{F}_{\text{com-zk}}^{R_{\text{DL}}}$发送的消息(decom-proof, 1, Q_1).

(b) 公钥N的长度大于等于$\max(3\log|q|+1, n)$.

(c) P_2接受零知识证明$N \in L_P$.

(6) 输出:

(a) P_1计算$Q = d_1 \cdot Q_2 - G$并储存(d_1, Q).

(b) P_2计算$Q = d_2 \cdot Q_1 - G$并储存(d_2, Q, N).

5.2 协同签名协议(sid, M)

(1) P_1的操作:

(a) P_1选择随机数$k_1 \leftarrow \mathbb{Z}_q^*$并计算$R_1 = k_1 \cdot G$.

(b) P_1向$\mathcal{F}_{\text{com-zk}}^{R_{\text{DL}}}$发送指令(com-prove, sid||1, R_1, k_1).

(2) P_2的操作:

(a) P_2接收$\mathcal{F}_{\text{com-zk}}^{R_{\text{DL}}}$返回的消息(comproof-receipt, *sid*||1).

(b) P_2选择随机数$k_2 \leftarrow \mathbb{Z}_q^*$并计算$R_2 = k_2 \cdot G$.

(c) P_2向$\mathcal{F}_{\text{zk}}^{R_{\text{DL}}}$发送指令(prove, sid||2, R_2, k_2).

(3) P_1的操作:

(a) P_1接收$\mathcal{F}_{\text{zk}}^{R_{\text{DL}}}$返回的消息(proof, sid||2, R_2). 如果没有接收到,则中止协议.

(b) P_1向$\mathcal{F}_{\text{com-zk}}^{R_{\text{DL}}}$发送指令(decom-proof, sid||1).

(c) P_1计算$R = k_1 \cdot R_2$, 记$R = (r_x, r_y)$, 计算$r = r_x + e \bmod q\ (e = \text{hash}(Z\|M))$.

(d) 计算$c_k = \text{Enc}_{\text{pk}}(k_1)$并发送给$P_2$.

(4) 零知识证明:

(a) P_1调用零知识证明协议向P_2证明$(c_k, \text{pk}, R_1) \in L_{\text{PDL}}$.

(5) P_2的操作:

(a) P_2接收到$\mathcal{F}_{\text{com-zk}}^{R_{\text{DL}}}$发送的消息(decom-proof, sid$\|1, R_1|$. 如果没有接收到,则中止协议.

(b) P_2接受零知识证明$(c_k, \text{pk}, R_1) \in L_{\text{PDL}}$,否则中止协议.

(c) P_2计算$R = k_2 \cdot R_1$和$r = r_{\text{x}} + e \bmod q$.

(d) P_2计算$C_1 = (k_2 \cdot {d_2}^{-1} \bmod q) \otimes c_k$; P_2选择随机数$\rho \leftarrow Z_{q^2}, \tilde{r} \leftarrow Z_N^*$并计算$\text{C}_2 = \text{Enc}_{\text{pk}}(\rho \cdot q + [{d_2}^{-1} \cdot r \bmod q]; \tilde{r})$; P_2计算并向P_1发送$C_3 = C_1 \oplus C_2$.

(6) P_1产生输出:

(a) P_1计算$s' = \text{Dec}_{\text{sk}}(C_3)$和$s = {d_1}^{-1} \cdot s' - r \bmod q$.

(b) P_1验证签名(r, s)是关于公钥Q的正确签名,输出(r, s),否则中止协议.

若一方在任何时间点中止协议,则停止所有的签名操作.

6 基于规约的安全证明

6.1 安全定义

首先我们回顾一下传统数字签名的安全定义[8]. 定义一个实验$\text{Expt-Sign}_{\mathcal{A},\Pi}(n)$,其中$\mathcal{A}$是概率多项式时间的攻击者,$\Pi = (\text{Gen}, \text{Sign}, \text{Verify})$是数字签名体制,$n$是安全参数. 具体内容如下:

(a) 调用密钥生成算法$\text{Gen}(1^n)$生成公私钥对(pk, sk).

(b) 攻击者$\mathcal{A}$被授予公钥pk和一个谕言机$\text{Sign}_{\text{sk}}(\cdot)$,攻击者可以使用该谕言机查询任何消息的签名.

(c) 攻击者输出消息签名对(m, σ). 令$\mathcal{Q}$为$\mathcal{A}$查询过的所有消息的集合

(d) 当且仅当$m^* \notin \mathcal{Q}$且$\text{Verify}_{\text{pk}}(m^*, \sigma^*) = 1$时,实验输出1.

当数字签名体制$\Pi = (\text{Gen}, \text{Sign}, \text{Verify})$满足下述条件时,我们称之为在自适应性选择消息攻击下存在性不可伪造的:如果对于任意概率多项式时间攻击者$\mathcal{A}$,都存在一个可忽略函数$\text{negl}(n)$使得

$$\Pr\left[\text{Expt-Sign}_{\mathcal{A},\Pi}(n) = 1\right] \leqslant \text{negl}(n)$$

仿照上述经典定义形式,我们从博弈的角度定义两方协同签名协议的安全性.

首先我们参照文献[12],定义一个状态谕言机$\Pi_b(\cdot,\cdot)$:

(a) 在第一次收到形为$(0,0)$的查询后,谕言机初始化一个机器M,执行P_{3-b}在密钥生成协议部分的指令. 如果在密钥生成协议中, P_{3-b}是首次发送消息的一方,则该消息即为谕言机的回复.

(b) 在接收到形为$(0, m)$的查询后,若密钥生成阶段尚未结束,谕言机向机器M发送m,作为其下一接收消息,并返回M的回复. 若密钥生成阶段已经结束,谕言机返回$\perp$.

(c) 若接收到形为(sid, m)的查询,并且$\text{sid} \neq 0$,但是密钥生成阶段尚未结束,谕言机返回$\perp$.

(d) 若接收到形为(sid, m)的查询,其中$\text{sid} \neq 0$,并且密钥生成阶段已经结束,sid是首次使用的标识符,则谕言机初始化一个新的机器M_{sid},运行P_{3-b}在协同签名阶段的指令,并将m输入为需要签名的消息. M_{sid}继承M在密钥生成阶段所获得的密钥分片和在该阶段结束时的状态. 如果P_{3-b}是首次发送消息的一方,则将该消息作为谕言机的回复.

(e) 若接收到形为(sid, m)的查询,其中$\text{sid} \neq 0$, 并

且密钥生成阶段已经结束,sid是已经使用的标识符,则谕言机调用已有的机器$\mathrm{M}_{\mathrm{sid}}$并将$m$作为下一接收消息发送给M,并返回$\mathrm{M}_{\mathrm{sid}}$返回的消息. 如果$\mathrm{M}_{\mathrm{sid}}$结束协议,返回$\mathrm{M}_{\mathrm{sid}}$的输出.

(f) 如果任何查询(sid, m)导致$\mathrm{M}_{\mathrm{sid}}$发送abort指令,谕言机返回abort并且终止所有运行中的机器.

然后,我们定义一个实验$\mathrm{Expt}-\mathrm{DistSign}_{\mathcal{A},\Pi}^{b}$,其中$\mathcal{A}$控制两方协同签名协议$\Pi$中的参与方$P_b$,$\Pi_b(\cdot,\cdot)$执行参与方$P_{3-b}$的指令. 在博弈实验中,攻击者$\mathcal{A}$可以任意选择想要签名的消息,通过查询$\Pi_b(\cdot,\cdot)$获得相应的签名. 其中密钥生成协议只执行一次,协同签名协议执行多次. 如果$\mathcal{A}$可以伪造一个没有查询过的消息的签名,即为获胜. 实验$\mathrm{Expt}-\mathrm{DistSign}_{\mathcal{A},\Pi}^{b}$的正式定义如下:

令$\pi=(\mathrm{Gen}, \mathrm{Sign}, \mathrm{Verify})$是一个数字签名体制.

(a) $(m^*,\sigma^*)\leftarrow\mathcal{A}^{\Pi_b(\bullet,\bullet)}(1^n)$.

(b) 令$\mathcal{Q}$为$\mathcal{A}$查询过的(sid, m)中所有m的集合,其中$\mathrm{sid}\neq 0$且(sid, m)是首次查询.

当且仅当$m^*\notin\mathcal{Q}$且$\mathrm{Verify}_{pk}(m^*,\sigma^*)=1$时,实验输出1.

下面我们给出协议安全的定义.

定义1　协议Π在满足下述条件时是一个关于π的安全两方协同签名协议:对于任意概率多项式时间攻击者$\mathcal{A}$和$b\in\{1,2\}$, $\Pr\left[\mathrm{Expt}-\mathrm{DistSign}_{\mathcal{A},\Pi}^{b}(1^n)=1\right]\leqslant \mathrm{negl}(n)$.

6.2　安全证明

定理2　若Paillier加密体制在选择明文攻击下是不可分辨的,SM2签名体制在自适应性选择消息攻击下是存在性不可伪造的,那么在$(\mathcal{F}_{\mathrm{com-zk}}^{R},\mathcal{F}_{\mathrm{zk}}^{R})$的混合模型下,本文构造的协议是一个关于SM2的安全两方协同签名协议.

证明:我们将构造一个概率多项式时间攻击者$\mathcal{S}$, $\mathcal{S}$通过调用协议攻击者$\mathcal{A}$的方式完成SM2的伪造签名实验. 我们将证明对任意的概率多项式时间攻击者$\mathcal{A}$和$b\in\{1,2\}$,$\mathcal{S}$在SM2的伪造签名实验中获胜的概率与$\mathcal{A}$在伪造协同签名实验中获胜的概率近似相同,即

$$\left|\Pr\left[\mathrm{Expt}-\mathrm{Sign}_{\mathcal{S},\Pi}^{b}(1^n)=1\right]-\Pr\left[\mathrm{Expt}-\mathrm{DistSign}_{\mathcal{A},\Pi}^{b}(1^n)=1\right]\right|\leqslant\mathrm{negl}(n)\qquad(1)$$

如果SM2签名体制在自适应性选择消息攻击下存在性不可伪造的,即$\Pr\left[\mathrm{Expt}-\mathrm{Sign}_{\mathcal{S},\Pi}^{b}(1^n)=1\right]\leqslant\mathrm{negl}(n)$,那么从上式可以推出$\Pr[\mathrm{Expt}-\mathrm{DistSign}_{\mathcal{A},\Pi}^{b}(1^n)=1]\leqslant\mathrm{negl}(n)$,即$\Pi$是一个关于SM2的安全两方协同签名协议. 我们针对$b=1$和$b=2$的情况分别给予证明.

P_1腐化的情形$(b=1)$:$\mathcal{S}$模拟状态谕言机$\Pi_b(\cdot,\cdot)$与$\mathcal{A}$互动,具体算法如下:

(1)在实验$\mathrm{Expt}-\mathrm{Sign}$中,$\mathcal{S}$收到$(1^n, Q)$,其中$Q$是SM2的公钥.

(2)$\mathcal{S}$调用$\mathcal{A}$并模拟$\mathrm{Expt}-\mathrm{DistSign}$中的状态谕言机$\Pi_b(\cdot,\cdot)$

(a) 在$\mathcal{A}$查询$(0,0)$之前,$\mathcal{S}$对$\mathcal{A}$的所有查询均返回$\perp$.

(b) 在$\mathcal{A}$查询$(0,0)$之后,$\mathcal{S}$接收到P_1在密钥生成协议中发送的第一条消息$(0, m_1)$, $\mathcal{S}$用以下方式计算谕言机的回复:

(i) $\mathcal{S}$将m_1解码为$(\mathrm{com}-\mathrm{prove}, 1, Q_1, d_1)$. $\mathcal{S}$验证$Q_1=d_1\cdot G$,如果成立,$\mathcal{S}$计算$Q_2=d_1^{-1}\cdot(Q+G)$;如果不成立,$\mathcal{S}$随机挑选一个Q_2. $\mathcal{S}$将谕言机回复设置为$(\mathrm{proof}, 2, Q_2)$并返回给$\mathcal{A}$.

(ii) $\mathcal{S}$将下一步接收到的消息$(0, m_2)$中的m_2解码为$(\mathrm{decom}-\mathrm{proof}, 1)$.

(iii) $\mathcal{S}$将下一步接收到的消息$(0, m_3)$中的m_3解码为公钥N.

(iv) $\mathcal{S}$将接下来接收到的消息$(0, m_i)$视为关于$N\in L_P$的零知识证明并作为一个诚实的验证者参与完成零知识证明协议.

(v) 如果$Q_1\neq d_1\cdot G$或者N的长度不符合要求,或者$N\in L_P$的零知识证明无法通过验证,$\mathcal{S}$中止协议,实验结束. 在这种情形下,$\mathcal{S}$没有任何输出;如果$\mathcal{S}$没有中止协议(即上述条件都成立),那么$\mathcal{S}$储存(Q, N). 密钥生成协议结束.

(c) $\mathcal{S}$收到形为(sid, m)的查询之后,如果sid是一个新的会话id, $\mathcal{S}$查询$\mathrm{Expt}-\mathrm{Sign}$实验中的谕言机$\mathrm{Sign}_{\mathrm{sk}}(\cdot)$以获得$m$的签名$(r, s)$. $\mathcal{S}$通过所得的r计算出椭圆曲线上的点R. 接下来,$\mathcal{S}$将从$\mathcal{A}$接收到的查询以如下方式处理:

(i) $\mathcal{S}$将在协同签名子协议中收到的第一条消息(sid, m_1)解码为$(\mathrm{com}-\mathrm{prove}, \mathrm{sid}\|1, R_1, k_1)$. 如果$R_1=k_1\cdot G$,$\mathcal{S}$计算$R_2=k_1^{-1}\cdot R$;否则,$\mathcal{S}$随机选择一个$R_2$. $\mathcal{S}$将谕言机的回复设置为$(\mathrm{proof}, \mathrm{sid}\|2, R_2)$并返回给$\mathcal{A}$.

(ii) $\mathcal{S}$将下一步接收到的消息(sid, m_2)中的m_2解码为$(\mathrm{decom}-\mathrm{proof}, \mathrm{sid}\|1)$. 如果$R_1\neq k_1\cdot G$,$\mathcal{S}$中止协议,实验结束;否则,继续执行以下操作.

(iii) $\mathcal{S}$将下一步接收到的消息(sid, m_3)中的m_3解码为$c_k=\mathrm{Enc}_{\mathrm{pk}}(k_1)$.

(iv) $\mathcal{S}$将接下来接收到的消息$(0, m_i)$视为关于$(c_k, \mathrm{pk}, R_1)\in L_{\mathrm{PDL}}$的零知识证明并作为一个诚实的验证

者参与完成零知识证明协议. 如果验证不通过,$\mathcal{S}$中止协议,实验结束;否则,继续执行以下操作.

(v) $\mathcal{S}$选择一个随机的$\rho \leftarrow Z_{q^2}$,计算密文$C_3=\mathrm{Enc}_{\mathrm{pk}}(x_1(s+r) \bmod q+\rho \cdot q)$并将其作为谕言机回复发送给$\mathcal{A}$. 其中$(r,s)$是从Expt – Sign实验谕言机中收到的签名.

6.3 当$\mathcal{A}$停止并输出签名对(m^*,σ^*)时,$\mathcal{S}$输出(m^*,σ^*)并且停止

首先我们认为,$\mathcal{S}$在实验Expt – Sign中收到的公钥Q和$\mathcal{S}$通过模拟谕言机与$\mathcal{A}$交互产生的公钥Q是完成一致的. 因为$\mathcal{S}$模拟的$Q_2=d_1^{-1}\cdot(Q+G)$,所以$\mathcal{A}$计算$d_1\cdot Q_2-G$得到的一定是$\mathcal{S}$接收到的公钥. 其次,容易看到,$\mathcal{A}$在与$\mathcal{S}$模拟出的谕言机交互过程中所查询的签名消息集合$\mathcal{Q}$与$\mathcal{S}$在实验Expt – Sign中所查询的签名消息集合完全一致. 因此,我们只需要证明$\mathcal{S}$模拟出来的$\mathcal{A}$的视图与$\mathcal{A}$在真实协议中的视图不可分辨,即可证明式(1)成立.

在密钥生成子协议阶段,$\mathcal{A}$的模拟视图与真实视图的唯一区别是Q_2的生成方式不同(在$(\mathcal{F}_{\mathrm{com-zk}}^{R},\mathcal{F}_{\mathrm{zk}}^{R})$混合模型下的零知识证明的知识证明中,$\mathcal{A}$的模拟视图与真实视图完全相同). 在真实协议中. Q_2由随机选择$d_2 \leftarrow \mathbb{Z}_q^*$并计算$d_2\cdot G$的方式得到. 在模拟视图中,$Q_2$由随机选择$Q$并计算$d_1^{-1}\cdot(Q+G)$的方式得到. 因为$Q$是完全随机的,所以两者的分布完全相同.

在协同签名子协议阶段,容易看到,R_2的分布在两种视图中完全相同. $\mathcal{A}$的模拟视图与真实视图唯一区别在于C_3的生成方式不同. 在真实视图中,$C_3=\mathrm{Enc}_{pk}\left(\rho\cdot q+[d_2^{-1}\cdot r \bmod q]+k_1\cdot[k_2\cdot d_2^{-1} \bmod q]\right)$;在模拟视图中,$C_3=\mathrm{Enc}_{pk}(d_1(s+r) \bmod q+\rho\cdot q)$. 容易看到,$[d_2^{-1}\cdot r \bmod q]+k_1\cdot[k_2\cdot d_2^{-1} \bmod q]\equiv x_1(s+r) \bmod q$,故有$[d_2^{-1}\cdot r \bmod q]+k_1\cdot[k_2\cdot d_2^{-1} \bmod q]=d_1(\mathrm{s+r}) \bmod q+1\cdot q$. 其中$0\leqslant l<q$,因为等式左边相较右边增大的值显然小于$q^2$. 于是我们得到了$C_3$所对应的明文在两个视图下的分布:

真实视图:$X=d_1(s+r) \bmod q+l\cdot q+\rho\cdot q$

模拟视图:$Y=d_1(s+r) \bmod q+\rho\cdot q$

下面我们证明,对于任意的$l\in\mathbb{Z}_q$,两个分布是统计不可分辨的. 我们注意到,X在区间$[l\cdot q,(l+q+1)\cdot q)$上均匀分布,$Y$在区间$[0,(q+1)\cdot q)$上均匀分布. 令$T=\{\lambda\cdot q|0\leqslant\lambda<l\}$,$X$和$Y$的统计距离为[30]

$$\Delta(X,Y)=\sum_{v\in \mathrm{T}}\left(\Pr[Y=v]-\Pr[X=v]\right)=\sum_{v\in \mathrm{T}}\left(\Pr[Y=v]\right)<\frac{1}{q}$$

其中,$\frac{1}{q}$是关于安全参数n的可忽略函数. 因此,分布X和Y是统计不可分辨的.

综上所述,我们证明了$\mathcal{A}$的模拟视图和真实视图不可分辨. 故当$b=1$时,式(1)成立.

P_2腐化的情形($b=2$):我们想要采用与以上类似的方法进行模拟,但是与之前的情形有所不同的是,当P_2腐化的时候,在协同签名的最后阶段,$\mathcal{S}$无法得知接收到的密文C_3是否被正确生成,因此无法决定最后是否输出所得签名. $\mathcal{S}$可以使用Paillier私钥对密文解密从而验证,但是一旦使用私钥,$\mathcal{A}$的模拟视图和真实视图的不可分辨性就无法归约到Paillier密文在选择明文攻击下的计算不可分辨性. 因为在选择明文攻击实验中,挑战者是不可能获得私钥的. 为了解决这个问题,我们记$\mathcal{A}$查询谕言机的上限次数为$p(n)$,假设$\mathcal{A}$在第i次查询时没有正确生成C_3,其中$i\in\{1,2,\cdots,p(n)+1\}$,当$i=p(n)+1$时意味着$\mathcal{A}$没有进行此项攻击. 那么$\mathcal{S}$有$\frac{1}{p(n)+1}$的概率猜对正确的$i$,从而模拟出正确的视图. 因此,$\mathcal{S}$伪造出正确签名的概率是$\mathcal{A}$伪造成功概率的$\frac{1}{p(n)+1}$. 下面我们介绍具体的模拟方法:

(1) 在实验Expt – Sign中,$\mathcal{S}$收到$(1^n,Q)$,其中Q是SM2的公钥.

(2) 记$\mathcal{A}$查询谕言机的上限次数为$p(n)$,$\mathcal{S}$随机选择$i\in\{1,2,\cdots,p(n)+1\}$.

(3) $\mathcal{S}$调用$\mathcal{A}$并模拟Expt – Sign的状态谕言机$\Pi_b(\cdot,\cdot)$.

(a) 在$\mathcal{A}$查询$(0,0)$之前,$\mathcal{S}$对$\mathcal{A}$的所有查询均返回$\perp$.

(b) 在$\mathcal{A}$查询$(0,0)$之后,$\mathcal{S}$模拟谕言机的回复为(comproof - receipt, 1). 对于之后接收到的消息,用以下方式回复:

(i) $\mathcal{S}$收到消息$(0,m_1)$并将m_1解码为$(\mathrm{prove},2,Q_2,d_2)$. 如果$Q_2\neq d_2\cdot G$或者$Q_2$是椭圆曲线上的零点,则中止协议. 否则,$\mathcal{S}$计算$Q_1=d_2^{-1}\cdot(Q+G)$并将谕言机回复设置为(decom - proof, 1, Q_1).

(ii) $\mathcal{S}$生成一组Paillier密钥对(pk, sk),记pk为N,N的长度为$\max(3\log|q|+1,n)$. $\mathcal{S}$将公钥N作为谕言机回复发送给$\mathcal{A}$.

(iii) $\mathcal{S}$调用关于L_p的零知识证明的模拟器,其输入为N,以模拟$\mathcal{A}$的视图.

(iv) $\mathcal{S}$储存(x_2,Q),密钥生成阶段结束.

(c) $\mathcal{S}$收到形为(sid, m)的查询之后,如果sid是一个新的会话id,$\mathcal{S}$将谕言机回复设置为(comproof -

receipt, sid||1)并发送给$\mathcal{A}$,然后查询Expt - Sign实验中的谕言机以获得m的签名(r,s). $\mathcal{S}$通过所得的r计算出椭圆曲线上的点R. 接下来,$\mathcal{S}$将从$\mathcal{A}$接收到的查询以如下方式处理:

(i) $\mathcal{S}$收到消息(sid, m_1)并将m_1解码为$(\text{prove}, \text{sid}||2, R_2, k_2)$. 如果$R_2 \neq k_2 \cdot G$或者$R_2$是椭圆曲线上的零点,则中止协议. 否则,$\mathcal{S}$计算$R_1 = k_2^{-1} \cdot R$并将谕言机回复设置为$(\text{decom}-\text{proof}, \text{sid}||1, R_1)$.

(ii) $\mathcal{S}$随机选择一个$\widetilde{k_1} \leftarrow \mathbb{Z}_q^*$并使用Paillier公钥加密得到密文$c_k = \text{Enc}_{\text{pk}}(\widetilde{k_1})$. $\mathcal{S}$将c_k作为谕言机回复发送给$\mathcal{A}$.

(iii) $\mathcal{S}$调用关于L_{PDL}的零知识证明的模拟器,其输入为(c_k, pk, R_1),以模拟$\mathcal{A}$的视图.

(iv) $\mathcal{S}$收到消息(sid, m_2)并将m_2解码为C_3. 如果这是$\mathcal{A}$的第i次查询,$\mathcal{S}$中止协议. 否则,$\mathcal{S}$输出签名(r,s).

6.4 当$\mathcal{A}$停止并输出签名对(m^*, σ^*)时,$\mathcal{S}$输出(m^*, σ^*)并且停止

与P_1腐化时的情况类似,当P_2腐化时,$\mathcal{S}$在实验Expt - Sign中收到的公钥Q和$\mathcal{S}$通过模拟谕言机与$\mathcal{A}$交互产生的公钥Q是完成一致的.

记P_2在其第j次查询消息m_j的签名的时候第一次使用未按协议要求计算的C_3查询谕言机. 那么当$i=j$时,模拟视图和真实视图的唯一区别在于$\text{Enc}_{\text{pk}}(\widetilde{k_1})$和$\text{Enc}_{\text{pk}}(k_1)$的不同. 具体而言,在真实视图中,$R_1 = k_1 \cdot G$而在模拟视图中,$\widetilde{k_1}$只是从$\mathbb{Z}_q^*$中挑选的一个随机数. 但是,$\mathcal{S}$在模拟的过程中从未使用过Paillier私钥. 因此,两者的不可分辨性可以直接归约到Paillier加密体制在选择明文攻击下的不可分辨性. 因此,$\mathcal{S}$有$\frac{1}{p(n)+1}$的概率模拟出与真实视图不可分辨的$\mathcal{A}$的视图. 由此可以得到

$$\left|\Pr\left[\text{Expt - Sign}_{\mathcal{S},\pi}^{b}\left(1^n\right)=1\right]-\frac{1}{p(n)+1}\cdot\Pr\left[\text{Expt - DistSign}_{\mathcal{A},\Pi}^{b}\left(1^n\right)=1\right]\right|\leqslant \text{negl}(n)$$

即

$$\Pr\left[\text{Expt - Sign}_{\mathcal{S},\pi}^{b}\left(1^n\right)=1\right]\geqslant\frac{1}{p(n)+1}\cdot\Pr\left[\text{Expt}-\text{DistSign}_{\mathcal{A},\Pi}^{b}\left(1^n\right)=1\right]-\text{negl}(n).$$

这样我们就证明了,假如存在一个攻击者,其在两方协同签名实验中成功伪造签名的概率不可忽略,那么就一定存在一个在SM2签名实验中的攻击者,其成功伪造签名的概率也是不可忽略的.

7 效率分析

本节对前文所设计的两方SM2签名方案进行实验测试. 测试基于商用密码管理办公室公开的《SM2椭圆曲线公钥密码算法》,选取椭圆曲线"NID_sm2p256v1"作为测试算例,哈希函数则采用SM3算法. 本文测试程序的构建基于商用密码管理办公室的开源密码学库GmSSL,测试环境为一台配置有64位ubuntu操作系统,AMD Ryzen 75800H处理器,8.00 GB内存的联想笔记本电脑.

我们分别针对恶意模型和半诚实模型进行了测试,测试一共进行20次,运行时间取平均值计算. 在恶意模型下,SM2密钥生成部分需要两次非交互的Schnorr证明和一个交互的互素证明,SM2协同签名部分需要两次非交互的Schnorr证明、一次交互的零知识证明以及一个范围证明. 在半诚实模型下,上述证明均取消.

从表1可以看出,相较于恶意模型,本协议在半诚实模型下的效率提升很大. 这说明协议的主要计算开销发生在使用到的证明协议上. 我们经过进一步实验发现,在所有证明协议中,Schnorr非交互式证明协议的开销很小(约为0.2 ms),大量的开销发生在交互式的证明协议上.

表1　不同安全模型下两方协同SM2签名协议运行时间及与两方协同ECDSA协议的对比　　单位: ms

安全模型	密钥生成时间	签名时间	总时间
恶意模型	483	1 440	1 923
半诚实模型	45	35	80
Lindell[12]中的恶意模型	2 435	36.8	2 471.8

本协议与Lindell[12]在恶意模型下的两方协同ECDSA协议表现大体相似. 不同之处在于,我们在签名阶段所需时间较多,Lindell[12]在密钥生成阶段所需时间较多. 这种差异来源于本协议所用到的交互式零知识证明大多分布在签名协议阶段.

总体来看,我们的协议表现与类似的两方协同ECDSA协同签名协议相近,计算开销主要发生在签名阶段的交互式零知识证明协议. 针对这些协议进行的优化是提升整个协议运算效率的突破口.

8 总结

为了解决电子合同应用中出现的私钥托管问题,我们提出了一种两方协同SM2签名协议. 用户和电子合同服务提供商可以独立保管一部分私钥分片,在需要签约时通过线上交互的方式生成合同的签名. 在整个过程中私钥不会出现,而且任何一方都无法独立地生成签名,既保证了私钥的安全性,也保证了服务商不

能滥用用户的私钥. 同时,我们使用基于博弈的方式证明了本协议是可证明安全的,即如果攻击者可以通过执行本协议伪造一个签名,那么他也可以在多项式时间内伪造一个SM2签名. 据悉,这是同类协议中第一个严格的安全证明. 最后,我们实验测试了该协议的运行效率,经过对比,该协议的效率与同类的两方ECDSA协议的耗时是接近的. 此外我们发现开销最大的是交互式的零知识证明和范围证明协议,但是这些协议为了保证协议在恶意模型下的安全性是必不可少的. 下一步的研究重点是在保证安全性的基础上对相关证明的效率进行优化.

参考文献

[1] RIVEST R L, HELLMAN M E, ANDERSON J C, et al. Responses to NIST's proposal[J]. Communications of the ACM, 1992, 35(7): 41-54.

[2] 国家密码管理局. SM2椭圆曲线公钥密码算法: GM/T0003-2012[S/OL]. (2012)[2023]. http://www.sca.gov.cn/sca/xxgk/2010-12/17/1002386/files/b791a9f908bb4803875ab6aeeb7b4e03.pdf.

[3] 全国信息安全标准化技术委员会. SM2椭圆曲线公钥密码算法: GB/T 32918.1-2016[S/OL]. (2016)[2023]. https://std.samr.gov.cn/gb/search/gbDetailed?id=71F772D81182D3A7E05397BE0A0AB82A.

[4] IX-IEC. Information technology - Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms: ISO/IEC 14888-3: 2016[S/OL]. (2016)[2023]. https://www.iso.org/standard/64267.html.

[5] Docusign, Inc. The modern chief legal officer[EB/OL]. (2021) [2024]. https://assets. ctfassets. net/3fcisxc3a6xz/3phbTe5Holw8Sxo1IuXiCj/1784ec22b996c661bda6fabea6c5912f/The_Modern_Chief_Legal_Officer.pdf.

[6] 杭州天谷信息科技有限公司.签管一体化电子合同云平台 [EB/OL]. (2021) [2023]. https://www. esign. cn/product/platform/.

[7] 深圳法大大网络科技有限公司.电子合同[EB/OL].(2023)[2023]. https://www.fadada.com/contractnotice/list-19.

[8] KATZ J, LINDELL Y. Introduction to Modern Cryptography[M]. Boca Raton: CRC Press, 2020.

[9] 侯红霞, 杨波, 张丽娜, 等. 安全的两方协作SM2签名算法[J]. 电子学报, 2020, 48(1): 1-8.
HOU H X, YANG B, ZHANG L N, et al. Secure two-party SM2 signature algorithm[J]. Acta Electronica Sinica, 2020, 48(1): 1-8. (in Chinese)

[10] 冯琦, 何德彪, 罗敏, 等. 移动互联网环境下轻量级SM2两方协同签名[J]. 计算机研究与发展, 2020, 57(10): 2136-2146.
FENG Q, HE D B, LUO M, et al. Efficient two-party SM2 signing protocol for mobile Internet[J]. Journal of Computer Research and Development, 2020, 57(10): 2136-2146. (in Chinese)

[11] MACKENZIE P, REITER M K. Two-party generation of DSA signatures[J]. International Journal of Information Security, 2004, 2(3): 218-239.

[12] LINDELL Y. Fast secure two-party ECDSA signing[C]// Annual International Cryptology Conference. Cham: Springer, 2017: 613-644.

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

[14] GENNARO R, GOLDFEDER S. Fast multiparty threshold ECDSA with fast trustless setup[C]//Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2018: 1179-1194.

[15] DOERNER J, KONDI Y, LEE E, et al. Threshold ECDSA from ECDSA assumptions: the multiparty case[C]// 2019 IEEE Symposium on Security and Privacy. San Francisco: IEEE, 2019: 1051-1066.

[16] LINDELL Y, NOF A. Fast secure multiparty ECDSA with practical distributed key generation and applications to cryptocurrency custody[C]//Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2018: 1837-1854.

[17] DAMGÅRD I, JAKOBSEN T P, NIELSEN J B, et al. Fast threshold ECDSA with honest majority[C]//International Conference on Security and Cryptography for Networks. Cham: Springer, 2020: 382-400.

[18] GENNARO R, GOLDFEDER S. One round threshold ECDSA with identifiable abort[EB/OL]. (2020) [2023]. https://eprint.iacr.org/2020/540.pdf.

[19] 尚铭, 马原, 林璟锵, 等. SM2椭圆曲线门限密码算法[J]. 密码学报, 2014, 1(2): 155-166.
SHANG M, MA Y, LIN J Q, et al. A threshold scheme for SM2 elliptic curve cryptographic algorithm[J]. Journal of Cryptologic Research, 2014, 1(2): 155-166. (in Chinese)

[20] LINDELL Y. How to simulate it-A turorial on the simulation proof technique[M]//Tutorials on The Foundations of Cryptography. Berlin: Springer, 2017: 277-346.

[21] HAZAY C, LINDELL Y. Efficient Secure Two-party Protocols: Techniques and Constructions[M]. Berlin:

Springer, 2010.
[22] LINDELL Y. Highly-efficient universally-composable commitments based on the DDH assumption[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2011: 446-466.
[23] BLAZY O, CHEVALIER C, POINTCHEVAL D, et al. Analysis and improvement of Lindell's UC-secure commitment schemes[C]//International Conference on Applied Cryptography and Network Security. Berlin: Springer, 2013: 534-551.
[24] FUJISAKI E. Improving practical UC-secure commitments based on the DDH assumption[J]. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2022, 105(3): 182-194.
[25] FIAT A, SHAMIR A. How to prove yourself: Practical solutions to identification and signature problems[C]// Conference on the Theory and Application of Cryptographic Techniques. Berlin: Springer, 1986: 186-194.
[26] PAILLIER P. Public-key cryptosystems based on composite degree residuosity classes[C]//International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 1999: 223-238.
[27] SCHNORR C P. Efficient identification and signatures for smart cards[C]//Conference on the Theory and Application of Cryptology. New York: Springer, 1989: 239-252.
[28] HAZAY C, MIKKELSEN G L, RABIN T, et al. Efficient RSA key generation and threshold paillier in the two-party setting[J]. Journal of Cryptology, 2019, 32(2): 265-323.
[29] BOUDOT F. Efficient proofs that a committed number lies in an interval[C]// International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2000: 431-444.
[30] SCHOENMAKERS B. Cryptographic protocols[EB/OL]. (2022)[2023]. https://www.win.tue.nl/~berry/CryptographicProtocols/LectureNotes.pdf.

作者简介

程一帆　男，1992年出生，河南南乐人. 2013年获得复旦大学物理学学士学位，2016年获得复旦大学物理学硕士学位. 现为牛津(海南)区块链研究院密码学实验室研究员. 主要研究方向为安全多方计算、数字签名.
E-mail: yifan@ssc-hn.com

刘擎宇　男，1996年出生，湖北天门人. 分别于2019年和2022年获得江南大学理学学士和南开大学理学硕士. 目前在南开大学攻读理学博士学位. 主要研究方向为密码学与编码理论.

梁泽宇　男，1995年出生，辽宁抚顺人. 2018年在辽宁工程技术大学获取电子信息工程学士学位. 同年开始从事于嵌入式安全相关工作. 现为牛津(海南)区块链研究院密码学实验室工程师. 主要研究方向为安全多方计算、可信执行环境.
E-mail: zeyu@oxhainan.org

于　昇　男，1976年出生，天津人. 1998年在天津理工大学获得计算机应用技术学士学位，分别于2006年、2010年在解放军信息工程大学获得军事装备学硕士学位、密码学博士学位. 现为云海链控股股份有限公司密码学实验室负责人. 主要研究方向为密码学、隐私计算、可信区块链.
E-mail: yusheng_24@163.com