安全的两方协作 SM2 签名算法

侯红霞[1,2,3],杨 波[1,3],张丽娜[1,3,4],张明瑞[1,3]
(1. 陕西师范大学计算机科学学院,陕西西安 710119;2. 西安邮电大学网络空间安全学院,陕西西安 710121;
3. 中国科学院信息工程研究所信息安全国家重点实验室,北京 100093;4. 西安科技大学计算机科学与技术学院,陕西西安 710054)

摘 要: 在签名算法中,一旦签名私钥被窃取,敌手就可以随意伪造合法用户的签名,从而致使合法用户的权益受到侵害. 为了降低签名私钥泄露的风险,本文提出了一种安全的两方协作 SM2 数字签名算法,该算法将签名私钥拆分成两个部分,分别交由两方来保管,通过采用零知识证明、比特承诺、同态加密等密码学技术保证了只有合法的通信双方才能安全地协作产生完整的 SM2 签名,任何一方都不能单独恢复出完整的签名私钥,方案的安全性在通用可组合安全框架下被证明,与已有的 SM2 协作签名方案相比,本文方案具有交互次数少、协作签名效率高等优势.

关键词: 数字签名; 零知识证明; 比特承诺; 同态加密; 可证明安全
中图分类号: TP309 **文献标识码:** A **文章编号:** 0372-2112 (2020)01-0001-08
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2020.01.001

Secure Two-Party SM2 Signature Algorithm

HOU Hong-xia[1,2,3], YANG Bo[1,3], ZHANG Li-na[1,3,4], ZHANG Ming-rui[1,3]
(1. *School of Computer Science, Shaanxi Normal University, Xi'an, Shaanxi* 710119, *China*;
2. *School of Cyberspace Security, Xi'an University of Posts & Telecommunications, Xi'an, Shaanxi* 710121, *China*;
3. *State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100093, *China*;
4. *Department of Computing Science and Technology, Xi'an University of Science and Technology, Xi'an, Shaanxi* 710054, *China*)

Abstract: In the signature algorithm, once the private key of the signature is stolen, the adversary can forge the signature of the legal user arbitrarily, which will cause the rights of legal users to be infringed. In order to reduce the risk of signature private key leakage, a secure two-party SM2 digital signature algorithm is proposed in this paper. The private key of the signature is divided into two parts and each part of the private key is handed over to the different parties separately. The cryptographic techniques such as zero-knowledge proof, bit commitment and homomorphic encryption are used to ensure that only the legal users can generate the integrated SM2 signature. The integrated private key cannot be recovered individually. The security of the proposed scheme is proved under the universally composable security framework. Compared with the existing SM2 cooperative signature schemes, the proposed scheme has the advantages of fewer interactions and higher efficiency.

Key words: digital signature; zero-knowledge proof; bit commitment; homomorphic encryption; provable security

1 引言

2010 年 12 月 17 日,为了满足电子认证服务系统等应用需求,国家密码管理局基于椭圆曲线密码体制发布了 SM2 数字签名算法,它能够满足多种密码应用中的身份认证和数据完整性、真实性的安全需求,已被制定为国家标准 GB/T 32918.2-2016[1],现已成为国际标准 ISO/IEC 14888-3/AMD1[2]. 该算法的安全性依赖于求解椭圆曲线上离散对数问题的困难性,与 RSA 和 DSA 签名算法相比,SM2 数字签名具有安全性高、存储空间小、运算复杂度低的优点,尤其在资源受限的环境中更具优越性,目前在国内商密应用领域已经广泛使用多年.

自从 SM2 数字签名算法公布以来,引起国内外学者的广泛关注[3~7],一些针对 SM2 数字签名的攻击方法已在相关文献中给出. 2013 年,Liu 等人[4]分别用部

收稿日期:2018-10-09;修回日期:2019-06-03;责任编辑:孙瑶
基金项目:国家重点研发计划(No. 2017YFB0802000);国家自然科学基金(No. 61572303, No. 61772326, No. 61802241, No. 61802242);"十三五"国家密码发展基金 (No. MMJJ20180217);中国科学院信息工程研究所信息安全国家重点实验室开放课题(No. 2017-MS-03)

分已知临时值攻击和错误注入攻击方法成功地对 SM2 数字签名算法实施了攻击. 他们表明对于 256 位的 SM2 签名算法,给定 100 个签名和签名时所用临时值的 3 个最低有效位,就可以在数小时之内恢复出签名私钥. 2015 年,Chen 等人[5]首次实现了针对 SM2 数字签名的格点攻击,在这种攻击下,即使临时值仅有几个比特的泄露也会导致签名私钥被恢复. 2017 年,Zhang 等人[6]表明虽然 SM2 数字签名能够防止简单能量分析(Simple Power Analysis,SPA)和差分能量分析(Differential Power Analysis,DPA)攻击,但仍无法抵抗格点攻击. 2018 年,Tuveri 等人[7]对 OpenSSL 代码库中集成的 SM2 算法进行了安全性分析,他们表明在数字签名生成过程中,通过执行远程定时、缓存定时等边信道攻击,可以获取签名密钥的部分信息. 由此可见,随着各种攻击技术的发展,攻击者会通过各种各样的攻击方法窃取用户签名私钥. 一旦签名私钥被窃取,攻击者就可以随意伪造合法用户的签名,在一些应用中,由此带来的损失可能是灾难性的.

为了降低签名私钥泄露的风险,常见的解决方案都是基于(t,n)门限密码学的思想[8],将私钥分割为 n 份,每一份称为一个秘密份额,分别存储在不同的物理设备中,签名时至少需要提供 $t+1$ 个秘密份额才能产生有效的签名. 但通常情况下,门限签名方案[9,10]中往往需要借助一个可信中心来产生和分发秘密份额,而实际情况中,这样的可信中心很难找到或者会成为系统的安全瓶颈. 2016 年,Yan 等人[11]基于 Joint-Shamir-RSS 算法[12]设计了一种无可信中心的 SM2 门限签名方案,但该方案在密钥生成阶段,每个用户都需要将秘密份额通过安全信道秘密地传送给其他用户,在签名生成阶段,至少需要 $2t+1$ 个参与者广播其秘密份额才能得到完整的签名结果,交互复杂,通信次数繁多,效率太过低下,不能满足实际应用需求. 2017 年,Lindell 等人[13]提出一个快速安全的两方 ECDSA 签名方案. 该方案的设计思想为后续的研究工作[3,14,15]提供了一种很好的思路.

结合文献[13]的思想,本文提出了一种安全的两方协作 SM2 签名算法,该算法将签名私钥拆分成两个部分,分别交由第一通信方和第二通信方保管,在密钥生成阶段和签名阶段需要通信双方协作计算才能产生验证公钥和完整的 SM2 签名,任何一方都不能得到完整的私钥独自输出一个完整的签名. 为了保证方案的安全性,我们采用零知识证明技术[16]确保通信双方身份的真实性,采用承诺技术[17]确保输出签名的正确性,采用 Paillier 同态加密技术[18]确保通信双方在不知道对方部分私钥的情况下,仍能够协作计算产生完整的 SM2 签名. 我们在通用可组合(Universally Composable,UC)安全框架[19]中的混合模型下,将本文方案的安全性规约到国家标准 GB/T 32918.2-2016 中的 SM2 数字签名算法[1]的安全性上. 与 Yan 等人[11]方案相比较,我们的方案交互次数少,且不需要建立秘密信道传递秘密份额,在协作签名阶段的效率具有明显优势.

.

2 预备知识

2.1 符号说明

本文中,κ 表示安全参数,$\mu(\kappa)$ 表示一个 κ 的可忽略函数. 用$[*]$表示椭圆曲线点乘运算,用$[-]$表示椭圆曲线点减运算. 用$\odot$表示标量乘同态运算,即 $a\odot b$ 表示 b 对应的明文与 a 做乘法运算;$\oplus$表示加法同态运算,即 $a\oplus b$ 表示 a 对应的明文与 b 对应的明文做加法运算. $Enc_{pk}()$表示在同态公钥 pk 下 Paillier 方案[18]的加密运算,$Dec_{sk}()$表示在同态私钥 sk 下 Paillier 方案[18]的解密运算.

2.2 SM2 数字签名

该数字签名算法由以下几个算法组成.

系统建立 输入安全参数 κ,输出系统公开参数 params $=\{p,q,E,G\}$,其中 p 表示有限域的规模,为大素数或为 2 的幂次;E 表示定义在有限域 F_p 上的椭圆曲线;G 表示椭圆曲线 E 上阶为 q 的生成元点.

密钥产生 输入系统公开参数 params,用户随机选取秘密值 $d\in[1,q-1]$,计算 $P=d[*]G$,将 P 作为公钥公开,d 作为私钥秘密保存.

签名生成 输入系统公开参数 params,私钥 d 和待签名消息 M,签名者按如下步骤生成签名:

令 $\overline{M}=Z\parallel M$,其中 Z 表示签名者身份标识与系统参数的杂凑值;

计算 $e=\mathrm{hash}(\overline{M})$,将 e 转换为整数,hash()为单向哈希函数;

产生随机数 $k\in[1,q-1]$,计算 $k[*]G=(x_1,y_1)$;

计算 $r=(x_1+e)\bmod q$,若 $r=0$ 或 $r+k=q$,则重新选取随机数 k;

计算 $s=(1+d)^{-1}(k-rd)\bmod q$,若 $s=0$,则重新选取随机数 k;否则,将(r,s)作为签名结果输出.

签名验证 输入系统公开参数 params,公钥 P 以及收到的消息 M'的签名(r',s'),验证者按如下步骤验证签名:

(1)检验 $r'\in[1,q-1]$是否成立,若不成立,则验证不通过;

(2)检验 $s'\in[1,q-1]$是否成立,若不成立,则验证不通过;

(3)令$\overline{M'}=Z\parallel M'$;

(4)计算 $e'=\mathrm{hasd}(\overline{M'})$;

(5)计算 $t=(r'+s')\bmod q$,若 $t=0$,则验证不通过;

(6)计算椭圆曲线点 $(x'_1,y'_1)=s'G+tP$;

(7)计算 $r''=(x'_1+e')\bmod q$,检验 $r''=r'$是否成立,若成立,则验证通过;否则验证不通过.

2.3 安全性定义

本节给出数字签名方案 π 和两方协作数字签名方案 Π 的安全性定义.

定义 1 一个签名方案 π 在选择消息攻击下是存在性不可伪造的,如果对于实验 1 中的任意概率多项式时间敌手 $\mathcal{A}$,存在一个可忽略的函数 μ 使得对于每个 κ,有

$$\Pr[\mathrm{Expt\text{-}Sign}_{\mathcal{A},\pi}(1^\kappa)=1]\leqslant\mu(\kappa).$$

定义 2 一个方案 Π 是由签名方案 π 派生的安全两方协作签名方案,如果对于实验 2 中的任意概率多项式时间敌手 $\mathcal{A}$ 和 $b\in\{1,2\}$,存在一个可忽略的函数 μ 使得对于每个 κ,有

$$\Pr[\mathrm{Expt\text{-}DistSign}^{b}_{\mathcal{A},\Pi}(1^\kappa)=1]\leqslant\mu(\kappa).$$

实验 1 和实验 2 的具体描述分别见图 1 和图 2.

实验 1 $\mathrm{Expt\text{-}Sign}_{\mathcal{A},\pi}(1^\kappa)$

1. $(vk,sk)\leftarrow\mathrm{Gen}(1^\kappa)$;
2. $(M^*,\sigma^*)\leftarrow A^{\mathrm{Sign}_{sk}}(1^\kappa,vk)$;
3. 令 Q 是敌手 $\mathcal{A}$ 向预言机询问的所有消息 M 的集合,则实验输出 1 当且仅当 $M^*\notin Q$ 且 $\mathrm{Verify}_{vk}(M^*,\sigma^*)=1$.

注:$\pi=(\mathrm{Gen},\mathrm{Sign},\mathrm{Verify})$表示一个数字签名方案.

图 1　实验 1

实验 2 $\mathrm{Expt\text{-}DistSign}^{b}_{\mathcal{A},\Pi}(1^\kappa)$

1. $(M^*,\sigma^*)\leftarrow\mathcal{A}^{\Pi_b(\cdot,\cdot)}(1^\kappa)$;
2. 令 Q 是敌手 $\mathcal{A}$ 向预言机询问的所有消息 M 的集合,则实验输出 1 当且仅当 $M^*\notin Q$ 且 $\mathrm{Verify}_{vk}(M^*,\sigma^*)=1$,其中,Verify 与方案 π 中一致.

注:Π 表示一个两方分布式数字签名方案.

图 2　实验 2

2.4 通用可组合安全模型

通用可组合(Universally Composable,UC)安全是由 Canetti[19]提出的用于定义密码协议安全性的框架. 它最优秀的性质就是模块化的设计思想:可以单独设计密码协议,只要各个子协议满足 UC 安全,就可以保证与其他协议组装、并行运行的安全性. UC 安全框架的核心由三个模型:现实模型,理想模型以及 $\mathcal{F}$-混合模型搭建而成,它的主要证明和技术手段是"模拟".

我们提出的方案是在 $\mathcal{F}_{zk}$ 和 $\mathcal{F}_{com-zk}$ 混合模型下构造的. 在我们的方案中通信方 U_1 和 U_2 使用了三个理想函数:理想承诺函数 $\mathcal{F}_{com}$、理想零知识函数 $\mathcal{F}_{zk}$ 以及承诺的非交互式零知识理想函数 $\mathcal{F}_{com-zk}$.

理想承诺函数 $\mathcal{F}_{com}$ 满足以下定义.

(1)从通信方 $U_i(i\in\{1,2\})$ 接收到(commit,*sid*,*x*),记录(*sid*,*i*,*x*)并发送(receipt,*sid*)给通信方 U_{3-i}. 如果(commit,*sid*,*)已被存储,则忽略该消息.

(2)从通信方 U_i 接收到(decommit,*sid*),如果(*sid*,*i*,*x*)已被记录,则发送(decommit,*sid*,*x*)给通信方 U_{3-i}.

对于一个关系 R,理想零知识函数用 $\mathcal{F}^R_{zk}$ 来表示,满足如下定义.

从通信方 $U_i(i\in\{1,2\})$ 接收到(prove,*sid*,*x*,*w*):如果$(x,w)\notin R$ 或 *sid* 已经被使用过了,则忽略该消息;否则,发送(proof,*sid*,*x*)给通信方 U_{3-i}.

对于一个关系 R,承诺的非交互式零知识理想函数用 $\mathcal{F}^R_{com-zk}$ 来表示,满足如下定义.

(1)从通信方 $U_i(i\in\{1,2\})$ 接收到(com-prove,*sid*,*x*,*w*):如果$(x,w)\notin R$ 或 *sid* 已经被使用过了,则忽略该消息;否则,存储(*sid*,*i*,*x*)并发送(proof-receipt,*sid*)给通信方 U_{3-i}.

(2)从通信方 U_i 接收到(decom-proof,*sid*):如果(*sid*,*i*,*x*)已被记录,则发送(decom-proof,*sid*,*x*)给通信方 U_{3-i}.

本文方案所使用的理想零知识函数 $\mathcal{F}^{R_P}_{zk}$,$\mathcal{F}^{R_{DL}}_{zk}$ 分别基于以下两个关系.

(1)Paillier 算法[18]公钥正确性证明关系:

$$R_P=\{(N,(p_1,p_2))\mid N=p_1\cdot p_2\text{ 且 }p_1,p_2\text{ 是素数}\}$$

(2)椭圆曲线点离散对数知识证明关系:

$$R_{DL}=\{(E,G,p,W,w)\mid W=w[*]G\}$$

3 安全的两方协作 SM2 数字签名算法

本节在 UC 安全框架下描述安全的两方协作 SM2 数字签名算法,该算法由以下四个算法组成.

系统建立 该算法与 2.2 节中描述一致.

密钥生成 令 U_1 表示第一通信方,U_2 表示第二通信方,输入系统公开参数 params 后,两方执行以下步骤.

(1)U_1 的第一条信息:

(*a*)U_1 随机选取子私钥 $d_1\in[1,q-1]$,计算子公钥 $P_1=d_1[*]G$;

(*b*)U_1 发送(com-prove,1,P_1,d_1)给理想函数 $\mathcal{F}^{R_{DL}}_{com-zk}$.

(2)U_2 的第一条信息:

(*a*) U_2 从 $\mathcal{F}^{R_{DL}}_{com-zk}$ 接收到(proof-receipt,1);

(*b*)U_2 随机选取子私钥 $d_2\in[1,q-1]$,计算子公钥 $P_2=d_2[*]G$;

(*c*)U_2 生成 Paillier 同态加密算法的密钥对(*pk*,*sk*);

(*d*)U_2 发送(prove,2,P_2,d_2)给理想函数 $\mathcal{F}^{R_{DL}}_{zk}$;

(*e*)U_2 发送(prove,2,*pk*,*sk*)给理想函数 $\mathcal{F}^{R_P}_{zk}$.

(3)U_1 的第二条信息:

(a)U_1 从 $\mathcal{F}_{zk}^{R_{UL}}$ 接收到(proof,2,P_2),从 $\mathcal{F}_{zk}^{R_P}$ 接收到(proof,2,pk);否则,退出;

(b)U_1 发送(decom-proof,1)给理想函数 $\mathcal{F}_{com\text{-}zk}^{R_{UL}}$.

(4)U_2 的验证:U_2 从 $\mathcal{F}_{zk}^{R_{UL}}$ 接收到(decom-proof,1,P_1),否则退出.

(5)输出:

(a)U_1 计算签名公钥 $P=d_1[*]P_2[-]G$ 并保存(d_1,P);

(b)U_2 计算签名公钥 $P=d_2[*]P_1[-]G$ 并保存(d_2,P).

协作签名 令 sid 表示唯一的会话标识符,M 表示待签名消息. 假设通信双方都已验证 sid 之前没有被使用过. 两方执行以下步骤.

(1)U_1 的第一条信息:

(a)U_1 计算 $\bar{M}=Z\|M$ 以及 $e=\mathrm{hash}(\bar{M})$,其中 Z 表示通信双方身份标识与系统参数的杂凑值;

(b)U_1 选取一个随机数 $k_1\in[1,q-1]$,计算 $Q_1=k_1[*]G$;

(c)U_1 发送(com-prove,$sid\|1$,Q_1,k_1)给理想函数 $\mathcal{F}_{com\text{-}zk}^{R_{UL}}$.

(2)U_2 的第一条信息:

(a)U_2 计算 $\bar{M}=Z\|M$ 以及 $e=\mathrm{hash}(\bar{M})$;

(b)U_2 从 $\mathcal{F}_{com\text{-}zk}^{R_{UL}}$ 接收到(proof-receipt,$sid\|1$);

(c)U_2 选取一个随机数 $k_2\in[1,q-1]$,计算 $Q_2=k_2[*]G$;

(d)U_2 发送(prove,$sid\|2$,Q_2,k_2)给理想函数 $\mathcal{F}_{zk}^{R_{UL}}$.

(3)U_1 的第二条信息:

(a)U_1 从 $\mathcal{F}_{zk}^{R_{UL}}$ 接收到(proof,$sid\|2$,Q_2);否则,退出;

(b)U_1 发送(decom-proof,$sid\|1$)给理想函数 $\mathcal{F}_{com\text{-}zk}^{R_{UL}}$.

(4)U_2 的第二条信息:

(a)U_2 从 $\mathcal{F}_{com\text{-}zk}^{R_{UL}}$ 接收到(decom-proof,$sid\|1$,Q_1);否则,退出;

(b)U_2 计算$(x_1,y_1)=Q=k_2[*]Q_1$,$r=(x_1+e)\bmod q$;

(c)若 r 不等于0,则 U_2 计算 $c_1=Enc_{pk}((d_2^{-1}\cdot k_2)\bmod q)$,$c_2=Enc_{pk}(d_2^{-1}+\eta q)$,其中 $\eta\in[1,q^2]$ 是随机选取的;

(d)U_2 发送 c_1,c_2 给 U_1.

(5)U_1 的第三条信息:

(a)U_1 计算$(x_1,y_1)=Q=k_1[*]Q_2$,$r=(x_1+e)\bmod q$;

(b)若 r 不等于0,U_1 计算 $s'=((d_1^{-1}\cdot k_1)\bmod q)\odot c_1\oplus((d_1^{-1}\cdot r)\bmod q)\odot c_2$.

(6)U_2 输出签名:

(a)U_2 计算 $s=(Dec_{sk}(s')-r)\bmod q$;

(b)若 s 不等于0且不等于 $q-r$,U_2 用签名公钥 P 验证签名(r,s),若验证通过,则将(r,s)作为完整签名输出;否则,退出.

签名验证 该算法与2.2节中描述一致.

4 安全性证明

定理1 若Paillier同态加密方案在选择明文攻击下是不可区分安全的,SM2签名方案在选择消息攻击下是存在性不可伪造的,则第3节给出的签名方案一个安全的两方协作SM2签名方案.

证明 我们在 $\mathcal{F}_{com\text{-}zk}$,$\mathcal{F}_{zk}$ 混合模型下证明本文方案的安全性. 证明整体思路为:对任意攻击协议的敌手 $\mathcal{A}$,我们构造一个敌手 $\mathcal{S}$(模拟器),$\mathcal{S}$ 在实验1中伪造签名的概率接近于 $\mathcal{A}$ 在实验2中伪造签名的概率. 也就是说,要证明的是如果同态加密算法在选择明文攻击下是不可区分安全的,则对于每个PPT敌手 $\mathcal{A}$ 和每个 $b\in\{1,2\}$,存在一个PPT敌手 $\mathcal{S}$ 和一个可忽略的函数 μ 使得对于每个 κ,有

$$\begin{aligned}&|\Pr[\text{Expt-Sign}_{S,\pi}(1^\kappa)=1]\\&-\Pr[\text{Expt-DistSign}_{\mathcal{A},\Pi}^{b}(1^\kappa)=1]|\leqslant\mu(\kappa)\end{aligned}\tag{1}$$

其中 Π 表示第3节给出的签名方案,π 表示SM2签名算法.

分析:由SM2的安全性可知,存在一个可忽略的函数 μ' 使得对于对于每个 κ,有 $\Pr[\text{Expt-Sign}_{S,\pi}(1^\kappa)=1]\leqslant\mu'(\kappa)$. 结合等式(1),有以下结论成立:

$$\begin{aligned}&\Pr[\text{Expt-DistSign}_{\mathcal{A},\Pi}^{b}(1^\kappa)=1]\\&\leqslant\mu(\kappa)+\mu'(\kappa)\end{aligned}\tag{2}$$

由等式(2)和定义2可得 Π 是安全的.

因此,只需证明等式(1)成立即可. 分以下两种情况展开证明.

$b=1$,即 U_1 被收买. 令 $\mathcal{A}$ 是实验 $\text{Expt-DistSign}_{\mathcal{A},\Pi}^{b}(1^\kappa)$ 中的PPT敌手,构造一个实验 $\text{Expt-Sign}_{S,\pi}(1^\kappa)$ 中的PPT敌手 $\mathcal{S}$ 如下.

(1)在实验Expt-Sign中,敌手 $\mathcal{S}$ 收到(1^κ,P),其中 P 是SM2的公开验证密钥.

(2)$\mathcal{S}$ 模拟实验 $\text{Expt-DistSign}_{\mathcal{A},\Pi}^{b}(1^\kappa)$ 中的预言机 Π,回答 $\mathcal{A}$ 的询问如下.

(a)在密钥生成子协议结束之前,对于 $\mathcal{A}$ 的所有询问(sid,·),$\mathcal{S}$ 均回答 $\perp$ 直到它收到询问(0,0).

(b)在 $\mathcal{A}$ 发送(0,0)给 Π 后,$\mathcal{S}$ 收到 U_1 在密钥生成子协议中的第一条消息(0,M_1),$\mathcal{S}$ 模拟预言机回答如下.

(i)S 解析 M_1 为混合模型中 U_1 发送给 $\mathcal{F}_{com\text{-}zk}^{R_{DL}}$ 的信息(com-prove,1,P_1,d_1).

(ii)$\mathcal{S}$ 验证 $P_1 = d_1[*]G$ 是否成立. 如果成立,则计算 $P_2 = (d_1)^{-1}[*]P$;否则,$\mathcal{S}$ 随机选取 P_2.

(iii)$\mathcal{S}$ 生成同态加密算法的密钥对(pk,sk).

(iv)$\mathcal{S}$ 设置预言机 Π 的回答为(proof,2,P_2),(proof,2,pk)并提交给 $\mathcal{A}$.

(c)$\mathcal{S}$ 收到的第二条消息(0,M_2)被处理如下.

(i)$\mathcal{S}$ 解析 M_2 为以下消息:$\mathcal{A}$ 发送给 $\mathcal{F}_{com\text{-}zk}^{R_{DL}}$ 的(decom-proof,$sid \| 1$).

(ii) 如果 $P_1 \neq d_1[*]G$ 或 $d_1 \notin [1, q-1]$,$\mathcal{S}$ 生成预言机的回答为 U_2 退出.

(c) 如果 $\mathcal{S}$ 模拟一个退出,则实验终止. 这种情况下 $\mathcal{S}$ 什么都不会输出. 否则,$\mathcal{S}$ 存储(d_1,$P[-]G$),分布式密钥生成阶段完成.

(d) 在收到一个形式为(sid,M)的询问后(其中 sid 为一个新的会话标识符),$\mathcal{S}$ 用 M 询问它在实验 Expt-Sign 中的签名预言机,收到一个返回的签名(r,s). 由 SM2 算法的验证程序,$\mathcal{S}$ 可计算椭圆曲线上的点 $Q = k_1k_2G$. 那么,$\mathcal{S}$ 从 $\mathcal{A}$ 收到的带有会话标识符 sid 的询问被处理如下.

(i)第一条消息(sid,M_1)首先被解析为(com-prove,$sid \| 1$,Q_1,k_1). 如果 $Q_1 = k_1[*]G$,则 $\mathcal{S}$ 令 $Q_2 = (k_1)^{-1}[*]Q$;否则 $\mathcal{S}$ 随机地选取 Q_2. $\mathcal{S}$ 设置预言机的回答为(proof,$sid \| 2$,Q_2),这正是 $\mathcal{A}$ 期望收到的.

(ii) 第二条消息(sid,M_2)被解析为以下消息:$\mathcal{A}$ 发送给 $\mathcal{F}_{com\text{-}zk}^{R_{DL}}$ 的(decom - proof,$sid \| 1$);如果 $Q_1 \neq k_1[*]G$,则 $\mathcal{S}$ 模拟 U_2 退出且实验终止;否则,$\mathcal{S}$ 随机选取 $c_2 \in [1, q^4]$,计算 $c_1 = (d_1 \cdot k_1^{-1} \bmod q) \cdot Enc_{pk}[(s+r) \bmod q] \oplus (-k_1^{-1} \cdot r \bmod q) \odot c_2$,其中 N 为同态加密算法的公开参数,s,r 为从 $\mathcal{F}_{SM2}$ 收到的签名中的值,$\mathcal{S}$ 设置给 $\mathcal{A}$ 的预言机回答为 c_1,c_2.

(iii) 第三条消息(sid,M_2)被解析为 s',$\mathcal{S}$ 设置给 $\mathcal{A}$ 的预言机回答为 s.

(3)当 $\mathcal{A}$ 终止并输出一个对(M^*,σ^*),敌手 $\mathcal{S}$ 也输出(M^*,σ^*)并终止.

接下来证明等式(1)成立.

首先,在模拟环境中 $\mathcal{S}$ 为 $\mathcal{A}$ 生成的公钥 $P[-]G$ 实际上是由它在实验 Expt-Sign 中接收到的公钥 P 得到的,所以 $\mathcal{A}$ 在实验 Expt-DistSign 中输出的有效伪造就是 $\mathcal{S}$ 在实验 Expt-Sign 中的一个有效伪造. 接下来证明真实环境和模拟环境中 $\mathcal{A}$ 的视图分布是统计接近的.

密钥生成阶段中,$\mathcal{A}$ 的模拟视图和真实视图之间的唯一不同是 P_2 的生成方式:真实协议中,是诚实参与方 U_2 选取随机数 d_2,计算 $P_2 = d_2[*]G$;而模拟游戏中是 $\mathcal{S}$ 计算 $P_2 = (d_1)^{-1}[*]P$. 由于 P 是随机选择的,故 $d_2[*]G$ 与 $(d_1)^{-1}[*]P$ 是同分布的. 最后,若 U_2 不退出,则真实协议中和模拟游戏中的公钥都为 $d_1[*]P_2[-]G = P[-]G$. 因此,密钥生成阶段中,$\mathcal{A}$ 的模拟视图和真实视图是同分布的且公钥均为 $P[-]G$.

签名阶段中,$\mathcal{A}$ 的模拟视图和真实视图之间的不同是 c_1,c_2 以及 s 的生成方式. 具体来说,两种情况中的 Q_2 是同分布的,这是因为 Q 是 $\mathcal{F}_{SM2}$ 随机生成的,所以 $(k_1)^{-1}[*]Q$ 和 $k_2[*]G$ 具有相同的分布. 在 $\mathcal{F}_{com\text{-}zk}$,$\mathcal{F}_{zk}$ 混合模型下,零知识证明和验证也是同分布的. 因此,不同之处是 c_1,c_2 以及 s 的生成方式:模拟游戏中,c_2 是随机选取的,c_1 是由

$$((d_1k_1^{-1}) \bmod q) \cdot Enc_{pk}[(s+r) \bmod q] \oplus ((-k_1^{-1}r) \bmod q) \odot c_2$$

计算而来的,s 是实验 Expt-Sign 中签名预言机返回的值;真实协议中,c_1 是对 $d_2^{-1}k_2 \bmod q$ 的加密,c_2 是对 $d_2^{-1} + \eta q$ 的加密,s 是由 $Dec_{sk}(s') - r \bmod q$ 计算而来的.

首先,真实视图中密文 c_2 所对应的明文为 $d_2^{-1} + \eta q$,模拟视图中 c_2 为随机选取,由于 $\eta \in [1, q^2]$ 是随机均匀选取的,因此两种视图中 c_2 是同分布的.

其次,SM2 签名算法中,当公钥变形为 $P[-]G$,验证算法保持不变时,其签名可变形为

$$\begin{aligned} s &= (kd^{-1} + d^{-1}r - r) = (d^{-1}(k+r) - r) \\ &= d_1^{-1}d_2^{-1}(k_1k_2 + r) - r) \bmod q \end{aligned}$$

故模拟视图中,

$$s = (d_1^{-1}k_1 \cdot d_2^{-1}k_2 + d_1^{-1}d_2^{-1}r - r) \bmod q,$$

而真实视图中,

$$\begin{aligned} s &= (Dec_{sk}(s') - r) \bmod q \\ &= (d_1^{-1}k_1 \cdot d_2^{-1}k_2 + d_1^{-1}r \cdot d_2^{-1} \\ &\quad + d_1^{-1}r \cdot \eta q - r) \bmod q \end{aligned}$$

因此,两种视图中 s 在 $\bmod q$ 下是实际上是相同的.

再次,c_1 在真实视图与模拟视图中的不同是

真实:密文 c_1 所对应的明文为 $d_2^{-1} \cdot k_2 \bmod q$;

模拟:密文 c_1 所对应的明文为

$$[d_1k_1^{-1} \cdot (s+r) \bmod q] - ((k_1^{-1}r) \bmod q) \cdot Dec_{sk}(c_2).$$

真实视图中,

$$s + r = (d_1^{-1}k_1 \cdot d_2^{-1}k_2 + d_1^{-1}r \cdot d_2^{-1} + d_1^{-1}r \cdot \eta q) \bmod q,$$

这意味着存在某个 ℓ,$0 \leqslant \ell < q$,使得

$$\begin{aligned} (s+r) \bmod q = &(d_1^{-1}k_1 \cdot d_2^{-1}k_2) \bmod q \\ &+ (d_1^{-1}r \cdot d_2^{-1}) \bmod q \\ &+ (d_1^{-1}) \bmod q \cdot \eta q + \ell q, \end{aligned}$$

即

$$\begin{aligned} d_2^{-1}k_2 = &[d_1k_1^{-1} \cdot (s+r) \bmod q] \\ &- k_1^{-1}(rd_2^{-1} + r\eta r\eta + d_1\ell) \cdot q \end{aligned}$$

由于 $\eta \in [1, q^2]$ 和密文 $c_2 \in [1, q^4]$ 是随机均匀选

取的,故值 $k_1^{-1}(rd_2^{-1}+r\eta+d_1\ell)\cdot q$ 与值 $(k_1^{-1}r \bmod q)\cdot Dec_{sk}(c_2)$ 是同分布的. 因此,两种视图中 c_1 是同分布的.

综上分析,$b=1$,即 U_1 被收买的情况下等式(1)成立.

$b=2$,即 U_2 被收买. 如同 U_1 被收买的情况,即构造一个模拟器 $\mathcal{S}$ 实验 Expt-Sign 的交互过程中模拟 $\mathcal{A}$ 的视图. 模拟过程中唯一的不同是,U_2 发送给 U_1 的最后一条信息,即密文 c_1,c_2 可能是被 $\mathcal{A}$ 恶意构造的,模拟器不能发现. 我们通过让 $\mathcal{S}$ 在某个时刻模拟 U_1 退出来解决这个问题. 也就是说,$\mathcal{S}$ 选取一个随机的 $i\in\{1,\cdots,p(\kappa)+1\}$,其中 $p(\kappa)$ 是 $\mathcal{A}$ 向预言机 Π 发起询问数量的上界. 如果 $\mathcal{S}$ 选择正确,则模拟是完备的. 因为 $\mathcal{S}$ 选对 i 的概率是 $\frac{1}{p(\kappa)+1}$,这意味着 $\mathcal{S}$ 能以 $\frac{1}{p(\kappa)+1}$ 的概率模拟 $\mathcal{A}$ 的视图. 因此,$\mathcal{S}$ 实验在 Expt-Sign 中伪造一个签名的概率至少是 $\mathcal{A}$ 在实验 Expt-DistSign 中伪造一个签名概率的 $\frac{1}{p(\kappa)+1}$ 倍.

令 $\mathcal{A}$ 是一个概率多项式时间敌手,$\mathcal{S}$ 执行如下.

(1)在 Expt-Sign 中,$\mathcal{S}$ 收到 $(1^\kappa,P)$,其中 P 是 SM2 的公开验证密钥.

(2)令 $p(\cdot)$ 表示 $\mathcal{A}$ 在实验 Expt-DistSign 中向预言机 Π 所做询问个数的上界. 则 $\mathcal{S}$ 选取一个随机的 $i\in\{1,\cdots,p(\kappa)+1\}$.

(3)$\mathcal{S}$ 模拟实验 $\text{Expt-DistSign}^{b}_{\mathcal{A},\Pi}(1^\kappa)$ 中的预言机 Π,回答 $\mathcal{A}$ 的询问如下.

(*a*) 在密钥生成子协议结束之前,对于 $\mathcal{A}$ 的所有询问 $(sid,\cdot)$,$\mathcal{S}$ 均回答 $\perp$ 直到它收到询问 $(0,0)$.

(*b*) 在 $\mathcal{A}$ 发送 $(0,0)$ 给 Π 后,$\mathcal{S}$ 计算预言机回答为 (proof-receipt,1),这正是 $\mathcal{A}$ 所期望收到的.

(*c*) S 收到形式为 $(0,M_1)$ 的消息后,处理过程如下:

(i)$\mathcal{S}$ 解析 M_1 为混合模型中 U_2 发送给 $\mathcal{F}_{zk}^{R_{ot}}$ 的信息 $(\text{prove},2,P_2,d_2)$,U_2 发送给 $\mathcal{F}_{zk}^{R_P}$ 的信息 $(\text{prove},2,pk,sk)$;

(ii)$\mathcal{S}$ 验证 pk 是否正确,P_2 是否为椭圆曲线上的非零点以及 $P_2=d_2[*]P$ 是否成立,如果验证不成功,$\mathcal{S}$ 模拟 U_1 退出并终止;

(iii)$\mathcal{S}$ 设置预言机 Π 的回答为 $(\text{decom-proof},1,P_1)$.

$\mathcal{S}$ 存储 $(d_2,P[-]G)$,密钥生成阶段模拟完毕.

(*d*) 在收到一个形式为 (sid,M) 的询问后(其中 sid 为一个新的会话标识符),$\mathcal{S}$ 计算一个 $\mathcal{A}$ 期望收到的预言机回答 $(\text{proof-receipt},sid\|1)$,并将其发送给 $\mathcal{A}$.

接下来,$\mathcal{S}$ 询问它在实验 Expt-DistSign 中的签名预言机并收到返回的签名 (r,s). 由 SM2 算法的验证程序,$\mathcal{S}$ 可计算椭圆曲线上的点 $Q=k_1k_2G$. 那么,$\mathcal{S}$ 从 $\mathcal{A}$ 收到的带有会话标识符 sid 的询问被处理如下.

(i) 第一条消息 (sid,M_1) 首先被解析为 $\mathcal{A}$ 发送给 $\mathcal{F}_{zk}^{R_{ot}}$ 的 $(\text{prove},sid\|2,Q_2,k_2)$. $\mathcal{S}$ 验证 $Q_2=k_2[*]G$ 且 Q_2 为椭圆曲线上的非零点;否则 $\mathcal{S}$ 模拟 U_1 退出. $\mathcal{S}$ 计算 $Q_1=(k_2)^{-1}[*]Q$ 并设置预言机的回答为 $(\text{decom-proof},sid\|1,Q_1)$.

(ii) 第二条消息 (sid,M_2) 被解析为 c_1,c_2. 如果这是第 i 次 $\mathcal{A}$ 向预言机 Π 发起的询问,则 $\mathcal{S}$ 模拟 U_1 退出. 否则,继续.

(4)当 $\mathcal{A}$ 终止并输出一个对 (M^*,σ^*),$\mathcal{S}$ 也输出 (M^*,σ^*) 并终止.

如同 U_1 被收买的情况,模拟过程中 $\mathcal{S}$ 生成的公钥 $P[-]G$ 实际上是由它在实验 Expt-Sign 中接收到的公钥 P 计算而来的. 令 j 是以 (sid,c_1,c_2) 对预言机 Π 的第一次调用,其中 c_1,c_2 是使得 U_1 不能获得一个有效签名 (r,s) 的值. 如果 $j=i$,$\mathcal{A}$ 的真实视图与模拟视图之间的不同就是密文 c_1,c_2. 具体来说,真实环境中,$c_1=Enc_{pk}(d_2^{-1}k_2)$(其中 $Q_2=k_2[*]G$),$c_2=Enc_{pk}(d_2^{-1}+\eta*q)$(其中 $P_2=d_2[*]G$)而模拟环境中,$c_1=Enc_{pk}(d_2^{-1'}k'_2)$(其中 k'_2 是随机的,与 $Q_2=k_2[*]G$ 无关),$c_2=Enc_{pk}(d_2^{-1'}+\eta q)$(其中 $d_2^{-1'}$ 是随机的,与 $P_2=d_2[*]G$ 无关). 因此,模拟视图与真实视图的不可区分性可直接规约到选择明文攻击下同态加密方案的不可区分性上.

故有

$$\left|\Pr[\text{Expt-Sign}_{S,\pi}(1^\kappa)=1\,|\,i=j]-\Pr[\text{Expt-DistSign}^{2}_{\mathcal{A},\Pi}(1^\kappa)=1]\right|\leqslant\mu(\kappa)$$

推得

$$\Pr[\text{Expt-DistSign}^{2}_{\mathcal{A},\Pi}(1^\kappa)=1]\leqslant\frac{\Pr[\text{Expt-Sign}_{S,\pi}(1^\kappa)=1\wedge i=j]}{\Pr[i=j]}+\mu(\kappa)\leqslant\frac{\Pr[\text{Expt-Sign}_{S,\pi}(1^\kappa)=1]}{1/(p(\kappa)+1)}+\mu(\kappa)$$

故

$$\Pr[\text{Expt-Sign}_{S,\pi}(1^\kappa)=1]\geqslant\frac{\Pr[\text{Expt-DistSign}^{2}_{\mathcal{A},\Pi}(1^\kappa)=1]}{(p(\kappa)+1)}-\mu(\kappa)$$

这意味着如果 A 能在 $\text{Expt-DistSign}^{2}_{A,\Pi}(1^\kappa)$ 中以不可忽略的概率伪造一个签名,则 S 就可以在 $\text{Expt-Sign}_{S,\pi}$ 中以不可忽略的概率伪造一个签名,这显然与 SM2 的安全性相矛盾.

综上分析,$b=2$,U_2 被收买的情况下等式(1)成立.

5 效率分析

本节主要将本文方案与文献[11]中 Yan 等人方案

的运行效率进行分析比较. 由于本文方案是两方协作签名,故我们在进行比较时,主要考虑门限值 $t=1$ 时的运行效率.

为了比较两个方案在各个阶段的运行效率,我们使用 Java 语言编程实现两个方案,通过调用 Bouncy Castle 密码库实现椭圆曲线上的计算,基于 PC 端开发,主要运行环境如下.

中央处理器:Intel i5-4200H &2. 8GHz;

内存:8. 00GB;

硬盘:240GB;

操作系统:Windows 10 专业版.

椭圆曲线各参数的具体取值与 GB/T 32918. 2-2016 中的附录 A. 2 中各参数的取值相同.

在没有考虑网络延时的情况下,我们分别执行两个方案 20 次,图 3 表示两个方案在各个阶段的平均运行时间,图 4 表示两个方案执行过程中的总耗时. 如图 3 所示,本文方案在协作签名阶段的运行时间明显低于 Yan 等人方案;如图 4 所示,本文方案的整体运行效率要高于 Yan 等人方案.

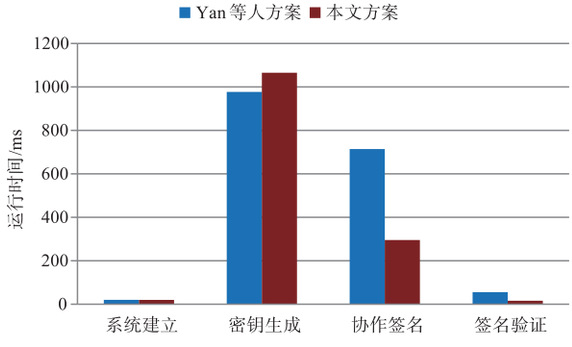

图3　方案各阶段平均运行时间

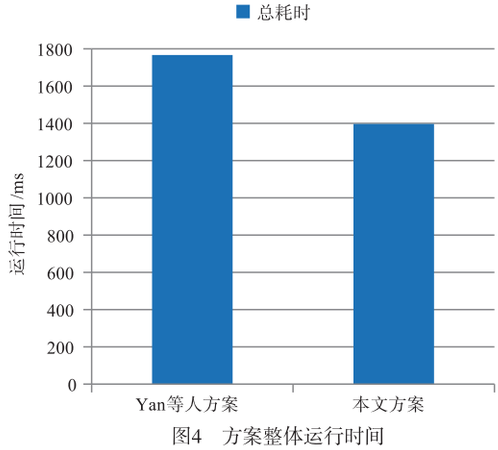

图4　方案整体运行时间

此外,为了实现协作签名,Yan 等人方案至少需要 $2t+1$,即 3 个参与方之间进行交互通信,才能实现协作签名的功能,而且每个参与方均需建立秘密信道,将秘密份额发送给其他参与方;而本文方案中仅需 2 个参与方便可实现安全的协作签名,且各参与方无需建立秘密信道,所有信息均可公开发送.

6　结论

随着移动智能终端的迅速普及,其安全问题也日益凸显. 如果私钥完整地存储在移动智能终端上,攻击者可能会通过各种攻击手段将私钥从移动智能终端导出,从而导致整个系统的不安全. 基于这样的背景,本文提出了一种安全的两方协作 SM2 数字签名算法,并基于 UC 框架在混合模型下证明了方案的安全性. 与现有的基于门限思想的方案[8] 相比,我们的方案在效率方面更具优势. 但本文方案由于采用的同态加密技术,导致在密钥生成阶段中生成同态加密算法的公私钥对耗时太长,影响方案的整体运行效率. 下一步的研究工作是在保持同等安全性的前提下,设计更加高效的两方以及多方协作的 SM2 签名方案.

参考文献

[1] GB/T 32918. 2-2016,信息安全技术 SM2 椭圆曲线公钥密码算法[S].

[2] ISO/IEC 14888-3:2016, Information Technology-Security Techniques-Digital Signatures with Appendix-Part 3: Discrete Logarithm Based Mechanisms[S].

[3] ZHANG Yu-di, HE De-biao, ZHANG Ming-wu, et al. A provable-secure and practical two-party distributed signing protocol for SM2 signature algorithm[OL]. Frontiers of Computer Science, 2018-05-28. DOI:10. 1007/s11704-018-8106-9.

[4] LIU M, CHEN J, LI H. Partially known nonces and fault injection attacks on SM2 signature algorithm[A]. Proceedings of the 9th International Conference on Information Security and Cryptology [C]. Berlin: Springer, 2013. 343 -358.

[5] CHEN Jia-zhe, LIU Ming-jie, LI He-xin, SHI Hong-song. Mind your nonces moving:Template-based partially-sharing nonces attack on SM2 digital signature algorithm[A]. Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security[C]. Singapore: ACM, 2015. 609 -614.

[6] ZHANG Kai-yu, XU Sen, GU Da-wu, et al. Practical partial-nonce-exposure attack on ECC algorithm[A]. Proceedings of the 13th International Conference on Computational Intelligence and Security[C]. New York: IEEE, 2017. 248 -252.

[7] TUVERI N, HASSAN S, et al. Side-channel analysis of SM2:a late-stage featurization case study[A]. Proceedings of the 34th Annual Computer Security Applications Confer-

ence[C]. San Juan:ACM,2018. 147 – 160.
[8] SHAMIR A. How to share a secret[J]. Communications of the ACM,1979,22(11):612 – 613.
[9] 马春光,石岚,等. 属性基门限签名方案及其安全性研究[J]. 电子学报,2013,41(5):1012 – 1015.
MA Chun-guang, SHI Lan, et al. Threshold attribute-based signature and its security[J]. Acta Electronica Sinica, 2013,41(5):1012 – 1015. (in Chinese)
[10] YANG Xiao-dong, WANG Cai-fen, ZHANG Lei, QIU Jian-bin. On-line/off-line threshold proxy re-signatures[J]. Chinese Journal of Electronics, 2014, 23(2):248 – 253.
[11] YAN Jie,LU Yu,CHEN Li-yun,NIE Wei. A SM2 elliptic curve threshold signature scheme without a trusted center[J]. KSII Transactions on International and Information Systems,2016,2(10):897 – 913.
[12] PEDERSEN T P. Distributed provers with applications to undeniable signatures[A]. Proceedings of Advances in Cryptology-EUROCRYPT' 91[C]. Berlin: Springer, 1991. 221 – 242.
[13] LINDELL Y. Fast secure two-party ecdsa signing[A]. Proceedings of Annual International Cryptology Conference[C]. Berlin:Springer,2017. 613 – 644.
[14] HE De-biao, ZHANG Yu-di, et al. Secure and efficient two-party signing protocol for the identity-based signature scheme in the IEEE P1363 standard for public key cryptography[OL]. IEEE Transactions on Dependable and Secure Computing, 2018-07-19. DOI: 10. 1109/TDSC. 2018. 2857775.
[15] ZHANG Yu-di, HE De-biao, et al. Efficient and provably secure distributed signing protocol for mobile devices in wireless networks[J]. IEEE Internet of Things Journal, 2018,5(6):5271 – 5280.
[16] GOLDWASSER S,MICALI S,RACKOFF C. The knowledge complexity of interactive proof system[J]. SIAM Journal on Computing,1989,18(1):186 – 208.
[17] BLUM M. Coin flipping by telephone[A]. Proceedings of Advances in Cryptology-CRYPT' 81[C]. Berlin:Springer,1981. 133 – 137.
[18] PAILLIER P. Cryptosystems based on composite degree residuosity classes[A]. Proceedings of Advances in Cryptology-EUROCRYPT' 99[C]. Berlin:Springer,1999. 223 – 238.
[19] CANETTI R. Universally composable security:a new paradigm for cryptographic protocols[A]. Proceedings of the 42nd IEEE Symposium on the FOCS[C]. New York: IEEE,2001. 136 – 145.

作者简介

侯红霞 女,1980 年生于山西朔州. 陕西师范大学计算机科学学院博士研究生,研究方向为密码学、信息安全.
E-mail:hongxiahou@ snnu. edu. cn

杨 波(通信作者) 男,1963 年生于陕西富平. 陕西师范大学计算机科学学院教授、博士生导师,研究方向为密码学、信息安全.
E-mail:byang@ snnu. edu. cn