软件学报 ISSN 1000-9825, CODEN RUXUEW
*Journal of Software* [doi: 10.13328/j.cnki.jos.006978]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

# SM2 数字签名算法的两方门限计算方案框架*

刘振亚[1], 林璟锵[1,2]

[1](中国科学技术大学 网络空间安全学院, 安徽 合肥 230027)

[2](中国科学技术大学北京研究院, 北京 100193)

通信作者: 林璟锵, E-mail: linjq@ustc.edu.cn

**摘　要**: 近年来, 已有多种 SM2 数字签名算法的两方门限计算方案被提出, 这些方案能够有效地增强 SM2 数字签名算法的私钥安全性. 根据不同的密钥拆分方法, 已有公开方案可以分为两类, 分别基于乘法和加法拆分. 再根据不同的签名随机数构造方法, 衍生出多种两方门限计算方案. 提出 SM2 数字签名算法的两方门限计算方案框架, 所提框架给出安全的两方门限计算基本过程, 又可以引入不同构造的签名随机数. 利用提出的框架, 结合随机数的不同构造, 完成所提框架的多种实例化, 即得到 SM2 数字签名算法多种不同的两方门限计算方案. 所提框架的实例化, 包括现有已知的 23 种两方门限计算方案, 也包括多种新的方案.

**关键词**: SM2 签名算法; 两方门限签名; 框架

**中图法分类号**: TP309

中文引用格式: 刘振亚, 林璟锵. SM2数字签名算法的两方门限计算方案框架. 软件学报. http://www.jos.org.cn/1000-9825/6978.htm

英文引用格式: Liu ZY, Lin JQ. Framework of Two-party Threshold Schemes for SM2 Digital Signatures. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/6978.htm

## Framework of Two-party Threshold Schemes for SM2 Digital Signatures

LIU Zhen-Ya[1], LIN Jing-Qiang[1,2]

[1](School of Cyber Science and Technology, University of Science and Technology of China, Hefei 230027, China)

[2](Beijing Research Institute of University of Science and Technology of China, Beijing 100193, China)

**Abstract**: There are a lot of two-party threshold schemes for SM2 digital signatures proposed in recent years, which can significantly enhance the security of private keys for SM2 digital signatures. According to different methods of key splitting, public schemes can be divided into two types: multiplicative key splitting and additive key splitting. Further, these public schemes can be subdivided into various two-party threshold schemes according to different constructions of the signature random number. This study proposes the framework of two-party threshold schemes for SM2 digital signature, which provides a safe basic calculation process of two-party threshold schemes and introduces the signature random number that can be constructed variously. With the proposed framework and various constructions of the random number, this study achieves the instantiation of the framework, obtaining a variety of two-party threshold schemes for SM2 digital signature. The instantiation includes 23 known two-party threshold schemes, as well as a variety of new schemes.

**Key words**: SM2 digital signature; two-party threshold signature; framework

基于公钥密码学的数字签名技术已经被广泛应用, 成为保证信息安全的重要工具, 而私钥的安全性是实施数字签名算法的基础. 如果用户的私钥泄露, 攻击者就可以任意地执行数字签名计算, 窃取用户的隐私信息或者伪造用户身份. 因此, 增强签名算法的私钥安全具有重要的应用意义.

1979 年, Shamir[1]和 Blakley[2]分别独立提出了秘密分享. 在秘密分享的基础上, Desmedt 等人[3,4]引入了门限密

* 基金项目: 国家重点研发计划 (2020YFB1005803)
收稿时间: 2023-01-20; 修改时间: 2023-04-17; 采用时间: 2023-06-04; jos 在线出版时间: 2023-10-11

码的概念, 开启了门限密码学的研究. 门限密码算法将私钥进行拆分, 得到若干个部分私钥, 保护在不同的设备中, 避免了完整私钥的直接存储和使用. 密码计算需要门限数量的部分私钥的参与才能完成. 门限密码方案一方面提高了系统的可用性, 即使少量部分私钥丢失, 也不会影响密码系统的可用性. 另一方面, 恶意敌手即使窃取了少量部分私钥, 也难以破坏密码系统的机密性. 数字签名算法的两方门限计算方案的特点在于: 2 个参与方分享签名私钥, 分别持有部分私钥共同计算数字签名. 数字签名算法的两方门限方案在移动互联网中应用广泛, 由移动终端和半可信的中心服务器共同掌握用户的私钥, 每一次数字签名计算同时需要移动终端和中心服务器参与, 从而提升攻击者窃取密钥的难度; 中心服务器也不能单独地使用该私钥来完成数字签名计算.

SM2 算法[5]是国家密码管理局于 2010 年发布的标准化椭圆曲线公钥密码算法, 包含了数字签名算法, 密钥交换协议和公钥加密算法, 其安全性基于求解有限域上椭圆曲线离散对数问题的困难性. SM2 算法已成为我国公钥算法标准 GM/T 0003.2-2012, 并进入国际标准 ISO/IEC 14888-3:2016 中. SM2 数字签名算法与大量的椭圆曲线签名算法类似, 签名时先生成签名随机数, 再计算签名随机数对应的椭圆曲线点, 根据椭圆曲线点的横坐标及消息摘要计算出签名的第 1 部分, 最后根据签名私钥、签名随机数以及签名第 1 部分, 计算得到签名第 2 部分, 与签名第 1 部分共同组成消息签名. 在上述签名过程中, 签名随机数和私钥都需要被保护.

针对 SM2 签名算法的私钥安全问题, 已有很多两方门限计算方案被提出. 不同的方案区别在于: (1) 不同的密钥生成方式, 例如基于乘法或加法拆分密钥; (2) 不同的签名随机数构造方式, 例如使用不同数量的随机数完成签名随机数的构造; (3) 不同的签名计算方式, 例如使用不同的交互轮数完成数字签名的协作计算. 在此基础上, 本文提出了 SM2 签名算法两方门限计算方案框架: 首先根据乘法或加法密钥拆分, 两个参与方完成协作式的密钥生成, 分别获得一个部分私钥; 然后双方协作计算签名随机数对应的椭圆曲线点, 签名随机数由双方各自构造的随机数共同组成; 最后根据双方分别持有的部分私钥、随机数以及消息摘要和椭圆曲线点, 完成数字签名的协作计算.

在协作签名的过程中, 两个参与方使用独立生成的随机数, 结合签名私钥以及各自的部分私钥, 协作构造符合安全要求的签名随机数, 完成该框架的多种实例化, 得到不同的 SM2 签名门限计算方案. 本文的实例化, 包括了现有已知的 23 种两方门限计算方案, 也包括多种新的方案.

本文第 1 节介绍门限签名的相关工作. 第 2 节介绍本文所需基础知识, 包括 SM2 签名算法和 MtA (multiplicative-to-additive share conversion protocol) 协议. 第 3 节介绍本文提出的两方门限签名计算框架并给出安全性说明. 第 4 节对提出的框架进行性能分析. 第 5 节总结全文.

## 1 相关工作

1991 年, Desmedt 等人[6]针对 RSA 算法设计了门限签名方案, 此后, 针对不同密码算法的门限签名方案也相继被设计. DSA 和 ECDSA 签名算法的门限方案需要所有签名参与方共同协商随机数, 并计算随机数的逆, 因此门限方案挑战很大. 针对该问题, 1996 年, Gennaro 等人[7]设计了计算乘积和求逆的多方安全计算方法, 完成 DSA 算法门限方案. 但是该方案在合成签名阶段, 各用户需要计算私钥与随机数的乘积以及随机数的逆, 当门限值是 $t$ 时, 至少需要 $2t+1$ 个用户才能合成签名, 通信消耗非常大. 2016 年, Gennaro 等人[8]针对门限数字签名算法的多项式扩张问题, 基于 Paillier 同态加密方案的门限版本和陷门承诺 (trapdoor commitment), 构造了最优化的门限 DSA 签名方案, 该方案也适用于 ECDSA 签名算法. 随后, Gennaro 等人[9]对上述方案进行优化, 使用了多方安全计算技术, 避免了使用复杂的零知识证明方法, 设计了原理更简单的 ECDSA 门限方案. 但是该方案依旧使用了 Paillier 同态加密. 针对该问题, 2017 年 Lindell 等人[10]使用指数 ElGamal 的加法同态算法代替 Paillier 算法, 设计了更高效率的门限 ECDSA 分布式密钥生成算法和签名算法.

SM2 签名算法与 DSA/ECDSA 签名算法的计算过程略有不同, 虽然 SM2 签名中包含私钥的逆运算, 但是可以将私钥 $d$ 的表达式 $(1+d)^{-1}$ 整体视为一个变量, 对其进行乘法或加法密钥拆分获得多个部分私钥, 避免数字签名过程中的求逆运算. 各参与方使用部分私钥以及各自生成的随机数, 通过简单的点乘、点加、数乘、数加等运算, 交互得到 SM2 签名结果 $(r,s)$. 相较于使用多方安全计算、加法同态密码算法等技术, 该方法在性能方面具有明显

的优势. 基于上述观察, 林璟锜等人[11]在 2014 年公开了一个基于 SM2 的两方协作签名协议, 不需要私钥的求逆计算; 同年, 尚铭等人[12]基于 Shamir 秘密分享的思想, 分别针对存在可信中心和不存在可信中心的情况, 设计了门限的 SM2 签名算法, 但是效率较低; 张永强等人[13]在 2017 年提出了一个具有盲签名特性的 SM2 两方协作签名方案. 多方安全计算等技术也可以用于设计 SM2 签名算法门限方案. 2015 年, Jie 等人[14]基于秘密分享和多方安全计算技术, 提出了一种无可信中心的 SM2 门限签名方案, 该方案相较于使用相同秘密分享技术的 ECDSA 门限签名方案拥有更高的效率.

## 2 基础知识

本文所提方法主要基于 SM2 签名算法[5], MtA 协议[15]也略有涉及, 下面就相关概念和基本知识予以介绍.

### 2.1 SM2 签名算法

SM2 签名算法是基于椭圆曲线密码的数字签名算法, 椭圆曲线$\mathbb{E}$参数包括$a$、$b$、$p$、$q$和$G$: $\mathbb{E}$是定义在有限域$F_p$上的椭圆曲线, 曲线上的点$(x,y)$满足$y^2 \equiv (x^3+ax+b) \bmod p$, $G=(x_G,y_G)$是曲线上$q$阶的基点.

设待签名的消息为$m$, 为了获取消息$m$的数字签名$(r,s)$, 签名者 Signer 应执行以下运算步骤.

● 密钥生成:

(1) 随机生成$d \in [1,q-1]$.

(2) 计算$P=[d]G$, 将$P$作为公钥公开, $d$作为私钥保存.

● 签名生成:

(1) 置$m'=m\|z$, $z$由用户 Signer 标识和 SM2 椭圆曲线的部分参数共同组成.

(2) 计算消息摘$e=Hash(m')$.

(3) 随机生成$k \in [1,q-1]$.

(4) 计算椭圆曲线点$(x_1,y_1)=[k]G$.

(5) 计算$r=(e+x_1) \bmod q$, 若$r=0$或$r+k=q$则返回步骤 (3).

(6) 计算$s=(1+d)^{-1}(k-rd) \bmod q$, 若$s=0$则返回步骤 (3), 否则签名结果即为$(r,s)$.

验证者 Verifier 收到签名$(r,s)$后, 通过如下步骤验证签名.

● 签名验证:

(1) 检查$r$和$s$是否满足$r,s \in [1,q-1]$; 计算$(x_1',y_1')=[s]G+(r+s)P$.

(2) 计算$r'=(Hash(m')+x_1') \bmod q$; 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.

签名过程中, $s$的计算可以等效为如下形式:

$$s=(1+d)^{-1}(k-rd)=(1+d)^{-1}(k-rd+r-r)=(1+d)^{-1}(k-(1+d)r+r)=(1+d)^{-1}(k+r)-r,$$

由上式可知, 签名私钥$d$的表达式$(1+d)^{-1}$在签名计算中可被整体视为一个变量. 因此, 本文对$(1+d)^{-1}$进行拆分, 避免求逆运算, 并记椭圆曲线点$[k]G$为$Q$.

### 2.2 MtA 协议

MtA 协议[15]能够将两方乘法份额分享转变成加法份额分享: Alice 和 Bob 分别持有乘法份额$z_A$和$z_B$, 满足$z=z_Az_B$. Alice 和 Bob 将$z_A$和$z_B$作为 MtA 协议的输入, 分别得到输出$z_A'$和$z_B'$, 满足$z=z_Az_B=z_A'+z_B'$. 该协议可以使用加法同态加密或不经意传输协议来实现.

#### 2.2.1 基于加法同态密码算法的 MtA 协议

加法同态密码算法定义了 2 个空间, 3 个算法和 2 个运算. 2 个空间包括明文空间$\mathcal{M}$和密文空间$\mathcal{E}$; 3 个算法包括密钥生成算法$KGen$、加密算法$Enc_{pk}$和解密算法$Dec_{sk}$: 密钥生成算法$KGen$用于生成一对公私钥$(pk,sk)$, 加密算法$Enc_{pk}$是$\mathcal{M} \to \mathcal{E}$的概率性算法, 解密算法$Dec_{sk}$是$\mathcal{E} \to \mathcal{M}$的确定性算法; 2 个运算分别是$\oplus: \mathcal{E} \times \mathcal{E} \to \mathcal{E}$运算和$\odot: \mathbb{Z} \times \mathcal{E} \to \mathcal{E}$运算, 定义如下:

$$\begin{cases} m_1+m_2=Dec_{sk}(Enc_{pk}(m_1)\oplus Enc_{pk}(m_2)) \\ k\cdot m=Dec_{sk}(k\odot Enc_{pk}(m)). \end{cases}$$

常用加法同态密码算法有 Paillier 算法[16]. 对于 Paillier 算法, $\oplus$和$\odot$操作定义如下:

$$Enc_{pk}(m_1)\oplus Enc_{pk}(m_2)=Enc_{pk}(m_1)\cdot Enc_{pk}(m_2).$$

$$k\odot Enc_{pk}(m)=Enc_{pk}(m)^k.$$

Alice 和 Bob 分别持有乘法份额$z_A$和$z_B$, 满足$z=z_Az_B$. Alice 和 Bob 通过如下步骤使用加法同态密码算法实现 MtA 协议, 从而获得加法份额分享.

Alice.1 计算$c_A=Enc_{pk}(z_A)$, 发送$c_A$给 Bob.

Bob.1 随机生成$z'_B$.

Bob.2 计算$c_B=z_B\odot c_A$, $c'_B=c_B\oplus Enc_{pk}(-z'_B)$, 发送$c'_B$给 Alice.

Alice.2 计算$z'_A=Dec_{sk}(c'_B)$.

Alice 和 Bob 分别获得加法份额$z'_A$和$z'_B$, 满足$z=z_Az_B=z'_A+z'_B$.

2.2.2 基于不经意传输的 MtA 协议

1-out-of-2 OT 协议[17]允许发送方输入 2 个比特串$(\alpha_0,\alpha_1)$, 接收方输入$\beta\in\{0,1\}$, 结果是接收方获得比特串$\alpha_\beta$. OT 协议保证接收方不知道$\alpha_{\bar{\beta}}$, 发送方不知道$\beta$.

Alice 和 Bob 分别持有乘法份额$z_A$和$z_B$, 满足$z=z_Az_B$, $z_A,z_B\in F_p$, 令$\rho=\log_2|p|$. Alice 和 Bob 通过如下步骤使用 1-out-of-2 OT 协议实现 MtA 协议, 从而获得加法份额分享.

Bob.1 随机生成$\rho$个随机数, 分别为$\mu_0,\ldots,\mu_{\rho-1}\in F_p$.

Bob.2 对于$i\in[0,\rho-1]$. 计算$t_i^0=\mu_i$, $t_i^1=2^iz_B+\mu_i$, 得到$\rho$个比特串对$(t_0^0,t_0^1),\ldots,(t_{\rho-1}^0,t_{\rho-1}^1)$.

Alice.1 记$z_A$的二进制形式为$z_{A_{\rho-1}}\ldots z_{A_0}$.

Alice 和 Bob 执行$\rho$次 1-out-of-2 OT 协议, 第$i$次执行时, Alice 从比特串对$(t_i^0,t_i^1)$中获得$t_i^{z_{A_i}}$.

Alice.2 计算$z'_A=\sum_{i=0}^{\rho-1}t_i^{z_{A_i}}$.

Bob.3 计算$z'_B=-\sum_{i=0}^{\rho-1}\mu_i$.

Alice 和 Bob 分别获得加法份额$z'_A$和$z'_B$, 满足$z=z_Az_B=z'_A+z'_B$.

## 3 SM2 数字签名算法的两方门限计算方案框架

两方门限计算意味着 2 个参与方共同完成 SM2 数字签名的计算过程, 任意一方无法独立完成签名的计算. 因此, 两个参与方应分别持有 1 个部分私钥, 用于协作计算签名. 在签名计算的过程中, 为了保护各自的部分私钥, 2 个参与方应该通过特定的方式协作生成签名随机数, 使得任意一方都无法获取完整的签名随机数, 从而进一步计算出完整的签名私钥. SM2 数字签名算法的两方门限计算方案的主要步骤如下.

(1) 2 个参与方完成协作式的密钥生成, 分别得到 1 个部分私钥. 2 个部分私钥与签名私钥满足特定的关系, 并对应地公开公钥.

(2) 2 个参与方协作计算, 得到一个椭圆曲线点$Q=[k]G$, 其中$k$为签名随机数.

(3) 2 个参与方根据步骤 2 所得椭圆曲线点的$x$轴坐标以及待签名消息的摘要$e$, 计算得到签名的第 1 部分$r$.

(4) 2 个参与方使用步骤 2 中生成或计算的随机数、签名第 1 部分$r$以及各自的部分私钥协作计算, 得到签名的第 2 部分$s$.

上述过程应该满足如下安全要求: 任一参与方无法从中获取完整签名私钥或另一部分私钥的任何信息. 这要求:

(1) 在密钥生成阶段, 任一参与方不会泄露有关签名私钥或部分私钥的任何信息, 使得另一方无法获取完整的签名私钥.

(2) 在签名计算阶段, 任一参与方不会泄露有关签名私钥或部分私钥的任何信息, 使得另一方无法获取完整的

签名私钥.

(3) 在签名计算阶段, 任一参与方无法获取完整的签名随机数或签名随机数的任何信息, 从而不会泄露完整的签名私钥.

根据部分私钥与签名私钥的特定关系, 本文提出 2 种两方门限计算方案框架, 分别基于乘法和加法的密钥拆分.

### 3.1 基于乘法密钥拆分的两方门限计算方案框架

#### 3.1.1 密钥生成

根据第 2.1 节的分析, 本文签名私钥 $d$ 的表达式对 $(1+d)^{-1}$ 进行乘法拆分, 使得 Alice 和 Bob 分别持有部分私钥 $d_1$ 和 $d_2$, $d$、$d_1$、$d_2$ 满足如下关系:

$$(1+d)^{-1}=d_1d_2,$$

由上式可知, 签名私钥 $d=d_1^{-1}d_2^{-1}-1$, 即公钥 $P=[d]G=[d_1^{-1}d_2^{-1}]G-G$.

为了实现上述乘法密钥拆分, Alice 和 Bob 通过如下步骤[10]进行协作式的密钥生成.

Alice.1 随机生成 $d_1\in[1,q-1]$, 计算 $P_1=[d_1^{-1}]G$, 发送 $P_1$ 给 Bob.

Bob.1 随机生成 $d_2\in[1,q-1]$, 计算 $P_2=[d_2^{-1}]G$, 发送 $P_2$ 给 Alice.

Bob.2 计算公钥 $P=[d_2^{-1}]P_1-G$.

Alice.2 计算公钥 $P=[d_1^{-1}]P_2-G$.

上述密钥生成过程应满足如下安全要求: 任一参与方无法获取完整签名私钥.

安全性说明如下. 在密钥生成过程中, Alice 能够获取如下信息.

(1) Bob 的公钥 $P_2=[d_2^{-1}]G$.

(2) 签名验证公钥 $P=[d_1^{-1}d_2^{-1}-1]G$.

由于椭圆曲线离散对数问题 (elliptic curve discrete logarithm problem, ECDLP)[18], Alice 无法从 $P_2$ 获取 $d_2^{-1}$, 从而无法计算完整签名私钥; 也无法从 $P$ 获取完整签名私钥 $d_1^{-1}d_2^{-1}-1$. 因此, 在 ECDLP 假设下, Alice 无法通过上述过程获取完整签名私钥. 同理可得, Bob 也无法在密钥生成过程中获取完整签名私钥.

#### 3.1.2 签名生成

在每一次签名计算过程中, Alice 和 Bob 需要协作计算一个随机的椭圆曲线点 $Q=[k]G$. 由于 $s=(1+d)^{-1}(k+r)-r$, 其中 $r$ 和 $s$ 是公开的数字签名, 若签名随机数 $k$ 仅由一个参与方生成, 则该参与方能够计算得到签名私钥 $d=(s+r)^{-1}(k+r)-1$, 违背了上述安全要求. 因此, 在协作计算椭圆曲线点的过程中, Alice 和 Bob 都需要生成各自的随机数, 共同组成签名随机数 $k$, 使得任一参与方无法获取完整的签名随机数, 从而无法计算得到签名私钥.

本文先讨论 $s$ 的协作计算过程, 再根据 $s$ 的构造讨论椭圆曲线点 $Q$ 的协作计算过程. 已知:

$$s=(1+d)^{-1}(k+r)-r=d_1d_2(k+r)-r=d_1d_2k+d_1d_2r-r.$$

为了实现 $s$ 的协作计算, Bob 应发送 $d_2r$ 给 Alice. 由于 $r$ 是公开的信息, 为了防止 Alice 计算得到 Bob 的部分私钥 $d_2$, Bob 应生成或计算一个随机数 $w_2$ 来保护自己的部分私钥, 于是 Bob 计算 $s_1=d_2(r+w_2)$ 并发送给 Alice. Alice 接收到 $s_1$ 后也应生成或计算一个随机数 $w_1$ 来保护自己的部分私钥; 否则, 攻击者可以从 $s$ 和 $s_1$ 得到 $d_1$. 于是 Alice 计算:

$$s=d_1(s_1+w_1)-r=d_1d_2w_2+d_1w_1+d_1d_2r-r=d_1d_2(w_2+d_2^{-1}w_1)+d_1d_2r-r,$$

这是一个有效的 SM2 签名, 签名随机数 $k=w_2+d_2^{-1}w_1$.

综上, Alice 和 Bob 协作计算 SM2 签名的过程如下, 如图 1 所示.

Alice.1 置 $m'=m\|z$, 计算 $e=Hash(m')$.

Alice.2 随机生成或计算 $w_1\in[1,q-1]$, 计算 $Q_1=[w_1]G$, 发送 $Q_1$ 和 $e$ 给 Bob.

Bob.1 随机生成或计算 $w_2\in[1,q-1]$, 计算 $Q=(x_1,y_1)=[w_2]G+[d_2^{-1}]Q_1$.

Bob.2 计算 $r=(x_1+e)\bmod q$.

Bob.3 计算 $s_1 = d_2(r + w_2) \bmod q$, 发送 $s_1$ 和 $r$ 给 Alice.

Alice.3 计算 $s = (d_1(s_1 + w_1) - r) \bmod q$, 签名即为 $(r, s)$.

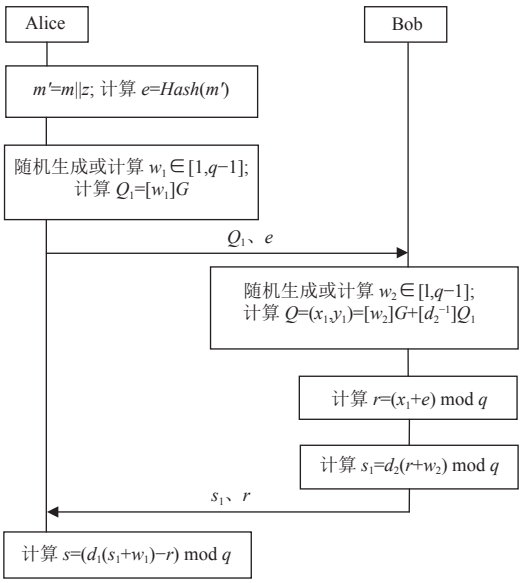


图 1 基于乘法密钥拆分的 2 随机数两方门限计算方案框架

在协作计算 SM2 数字签名的过程中, Alice 和 Bob 能够通过不同的方式构造各自的随机数, 对上述门限计算方案框架完成实例化, 得到不同的两方门限计算方案. 随机数的构造应满足如下安全要求.

(1) 随机数 $w_1$ 至少包含 1 个独立随机生成的随机数 $k_1$, 且 $w_1$ 与 $k_1$ 的取值分布都是 $[1, q-1]$ 均匀分布, 使得方程 $s = d_1 d_2 (w_2 + d_2^{-1} w_1 + r) - r$ 对于 Bob 来说至少包含两个未知数 $d_1$ 和 $w_1$, 从而无法解出 Alice 的部分私钥 $d_1$.

(2) 随机数 $w_2$ 至少包含 1 个独立随机生成的随机数 $k_2$, 且 $w_2$ 与 $k_2$ 的取值分布都是 $[1, q-1]$ 均匀分布, 使得方程 $s_1 = d_2(r + w_2)$ 对于 Alice 来说至少包含两个未知数 $d_2$ 和 $w_2$, 从而无法解出 Bob 的部分私钥 $d_2$.

本文基于随机数 $k_1$ 和 $k_2$, 结合签名私钥 $d$, Alice 和 Bob 的部分私钥 $d_1$、$d_2$ 以及它们的逆 $d_1^{-1}$、$d_2^{-1}$, 进行加或乘的组合, 实现签名随机数的不同构造. 例如, $w_i\ (i = 1, 2)$ 可以是 $k_i$, 这是一个独立随机生成的随机数; 可以是 $d_i k_i$, 这是部分私钥与独立随机生成随机数的乘积; 可以是 $d_i^{-1} k_i$, 这是部分私钥的逆与独立随机生成随机数的乘积; 也可以是 $1 + d_i k_i$, $1 + d_i^{-1} k_i$, $d_i + k_i$, $d_i^{-1} + k_i$ 等.

在具体的实例化方案中, $s$ 按照上文相应的步骤计算即可.

进一步, $w_i$ 还可以使用签名私钥 $d$ 进行构造. 虽然 Alice 和 Bob 无法获取签名私钥 $d$, 但是可以通过一定的形式转换, 使得 Alice 和 Bob 在不知道 $d$ 的情况下, 基于 $P$ 和 $G$ 完成签名随机数构造. 即随机数 $k$ 可进一步变化为如下等价形式:

$$\begin{aligned} k &= w_2 + d_2^{-1} w_1 = d_1^{-1} d_2^{-1} (d_1 d_2 w_2 + d_1 w_1) = ((d_1^{-1} d_2^{-1} - 1) + 1) d_1 d_2 w_2 + d_1 w_1 \\ &= (1 + d) d_1 d_2 w_2 + d_1 w_1 = (1 + d) d_1 d_2 w_2 + (1 + d) d_1 w_1. \end{aligned}$$

令 $(1 + d) d_1 d_2 w_2 = w_2'$, $(1 + d) d_1 w_1 = w_1'$, 使得签名随机数 $k = d_1 w_2' + w_1'$.

此时, Alice 和 Bob 能够通过如下步骤协作计算椭圆曲线点 $Q = [k]G$.

Alice.1 随机生成或计算 $w_1 \in [1, q-1]$, 计算 $Q_1 = [d_1](P + G)$, $Q_2 = [d_1 w_1](P + G)$, 发送 $Q_1$ 和 $Q_2$ 给 Bob.

Bob.1 随机生成或计算 $w_2 \in [1, q-1]$, 计算 $Q = [d_2 w_2] Q_1 + Q_2$.

虽然上述椭圆曲线点 $Q$ 的协作计算过程相较于不使用公钥 $P$ 的计算过程有所不同, 但签名随机数的构造相同, 都是 $k = w_2 + d_2^{-1} w_1$. 因此, $s$ 仍然按照上文相应的步骤计算即可.

由于 $w_i'=(1+d)d_iw_i$, 根据 $w_i$ 的取值, $w_i'$ 可以相应取值为: $(1+d)d_ik_i$, $(1+d)d_i^2k_i$, $(1+d)k_i$, $(1+d)(d_i+d_i^2k_i)$, $(1+d)(d_i+k_i)$, $(1+d)(d_i^2+d_ik_i)$, $(1+d)(1+d_ik_i)$ 等.

限于篇幅, 本文列出如下几种实例化方案. 表 1 展示了不包含签名私钥 $d$ 的签名随机数构造形式.

表 2 展示了包含签名私钥 $d$ 的签名随机数构造形式. 序号 1.*.d 旨在表明该方案与表 1 中的方案 1.*为相同方案, 不同之处在于椭圆曲线点 $Q$ 的协作计算过程是否使用公钥 $P$, 相当于签名随机数是否使用私钥 $d$. 而最终计算的签名第 2 部分 $s$ 相同, 在表 2 中省略.

表 1　基于乘法密钥拆分的 2 随机数框架的实例化

| $w_1$ | $w_2$ | $k=w_2+d_2^{-1}k_1$ | $s=(1+d)^{-1}(k+r)-r$ | 序号 |
|---|---|---|---|---|
| $k_1$ | $k_2$ | $k_2+d_2^{-1}k_1$ | $d_1d_2k_2+d_1k_1+(d_1d_2-1)r$ | 1.1 |
| | $d_2k_2$ | $d_2k_2+d_2^{-1}k_1$ | $d_1d_2^2k_2+d_1k_1+(d_1d_2-1)r$ | 1.2 |
| | $d_2^{-1}k_2$ | $d_2^{-1}(k_2+k_1)$ | $d_1(k_2+k_1)+(d_1d_2-1)r$ | 1.3 |
| | $1+d_2k_2$ | $1+d_2k_2+d_2^{-1}k_1$ | $d_1d_2+d_1d_2^2k_2+d_1k_1+(d_1d_2-1)r$ | 1.4 |
| | $1+d_2^{-1}k_2$ | $1+d_2^{-1}(k_2+k_1)$ | $d_1d_2+d_1(k_2+k_1)+(d_1d_2-1)r$ | 1.5 |
| | $d_2+k_2$ | $d_2+k_2+d_2^{-1}k_1$ | $d_1d_2^2+d_1d_2k_2+d_1k_1+(d_1d_2-1)r$ | 1.6 |
| | $d_2^{-1}+k_2$ | $d_2^{-1}+k_2+d_2^{-1}k_1$ | $d_1+d_1d_2k_2+d_1k_1+(d_1d_2-1)r$ | 1.7 |
| $d_1^{-1}k_1$ | $k_2$ | $k_2+d_1^{-1}d_2^{-1}k_1$ | $d_1d_2k_2+k_1+(d_1d_2-1)r$ | 1.8 |
| | $d_2^{-1}k_2$ | $d_2^{-1}(k_2+d_1^{-1}k_1)$ | $d_1k_2+k_1+(d_1d_2-1)r$ | 1.9 |

表 2　基于签名私钥的两随机数框架签名随机数构造

| $w_1'$ | $w_2'$ | $k=d_1w_2'+w_1'$ | 序号 |
|---|---|---|---|
| $(1+d)d_1k_1$ | $(1+d)d_2k_2$ | $(1+d)(d_1d_2k_2+d_1k_1)$ | 1.1.d |
| | $(1+d)d_2^2k_2$ | $(1+d)(d_1d_2^2k_2+d_1k_1)$ | 1.2.d |
| | $(1+d)k_2$ | $(1+d)(d_1k_2+d_1k_1)$ | 1.3.d |
| | $(1+d)(d_2+d_2^2k_2)$ | $(1+d)(d_1d_2+d_1d_2^2k_2+d_1k_1)$ | 1.4.d |
| | $(1+d)(d_2+k_2)$ | $(1+d)(d_1d_2+d_1k_2+d_1k_1)$ | 1.5.d |
| | $(1+d)(d_2^2+d_2k_2)$ | $(1+d)(d_1d_2^2+d_1d_2k_2+d_1k_1)$ | 1.6.d |
| | $(1+d)(1+d_2k_2)$ | $(1+d)(d_1+d_1d_2k_2+d_1k_1)$ | 1.7.d |
| $(1+d)k_1$ | $(1+d)d_2k_2$ | $(1+d)(d_1d_2k_2+k_1)$ | 1.8.d |
| | $(1+d)k_2$ | $(1+d)(d_1k_2+k_1)$ | 1.9.d |

所列方案旨在展示如何通过随机数的构造对该框架进行实例化, 得到相应两方门限计算方案, 其中也包含了该框架下的已有公开方案. 方案 1.1, 1.3, 1.8, 1.9 为已公开方案[19−31], 其余均为新方案. 对随机数 $w_1$ 和 $w_2$ 进行相应的不同替换即可得到本文提出的所有实例化方案, 完整的实例化方案表格可见附录 A 中的表 A1. 值得说明的是, 以上列出的实例化方案并非所有可能方案, 凡是满足随机数安全要求的构造, 均可用于实例化得到两方门限计算方案.

3.1.3　安全性分析

第 3.1.2 节所描述的两方门限签名计算过程应满足如下安全要求: 任一参与方无法获取完整的签名私钥或另一部分私钥的任何信息.

安全性说明如下.

在一次协作签名中, Alice 能够获取如下信息.

(1) 签名第 1 部分 $r=(x_1+e) \bmod q$.

(2) 签名第 2 部分 $s$ 的中间值 $s_1=d_2(r+w_2) \bmod q$.

其中, $r$ 是公开的信息且不包含任何 Bob 的私密信息, $s_1$ 包含 Bob 的私密信息 $d_2$ 和 $w_2$. Alice 试图从方程 $s_1-rd_2-w_2d_2=0$ 中解出 $d_2$ 和 $w_2$, 该方程为二元一次方程. Alice 令 $d_2$ 在 $[1,q-1]$ 区间上遍历取值, 并计算与之对应的 $w_2=d_2^{-1}s_1-r$, 得到该方程在 $[1,q-1]$ 区间上的 $(q-1)$ 组解. Alice 从这 $(q-1)$ 组解中获取 $d_2$ 和 $w_2$ 的难度等同于穷举. 对于 Alice 而言, 相较于仅掌握签名验证公钥以及签名结果, 掌握上述信息并没有降低获取 Bob 的部分私钥 $d_2$ 以及随机数 $w_2$ 的难度. 假设 Alice 与 Bob 进行多次协作签名, 每次签名 Alice 都能够获取 1 个方程. 记第 $i$ 次签名 Alice 获取的方程为 $s_1^{[i]}-r^{[i]}d_2-w_2^{[i]}d_2=0$, 其中包含 2 个未知数, 分别为 $d_2$ 和 $w_2^{[i]}$. 每次签名 Bob 都会使用相同的部分私钥 $d_2$, 并独立随机生成或计算 1 个随机数 $w_2^{[i]}$. 因此 Alice 通过与 Bob 的 $n$ 次协作签名, 能够获取一个由 $n$ 个方程组成的方程组, 其中包含 $(n+1)$ 个未知数. Alice 令 $d_2$ 在 $[1,q-1]$ 区间遍历取值, 并计算与之对应的 $w_2^{[i]}=d_2^{-1}s_1^{[i]}-r^{[i]}$. 得到该方程组在 $[1,q-1]$ 区间上的 $(q-1)$ 组解. Alice 从这 $(q-1)$ 组解中获取 $d_2$ 和 $w_2^{[i]}$ 的难度等同于穷举. 对于 Alice 而言, 相较于仅掌握签名验证公钥以及签名结果, 掌握上述信息并没有降低获取 Bob 的部分私钥 $d_2$ 以及随机数 $w_2^{[i]}$ 的难度. 进一步, 根据签名随机数构造的安全要求, $w_2^{[i]}$ 总是包含至少 1 个独立随机生成的随机数, 因此在具体的两方门限计算方案中, Alice 也无法求解所得的方程组. 综上, Alice 无法从协作签名的交互过程中获取任何 Bob 的私密信息, 从而无法获取完整的签名私钥或另一部分私钥的任何信息.

在一次协作签名中, Bob 能够获取如下信息.

(1) 椭圆曲线点 $Q_1=[w_1]G$.

(2) 待签名消息的摘要 $e$.

(3) 公开的签名第 2 部分 $s=(d_1(s_1+w_1)-r) \bmod q$.

由于 ECDLP 和哈希函数的安全性, Bob 无法从 $Q_1$ 和 $e$ 中获取任何信息. 而 $s$ 包含 Alice 的私密信息 $d_1$ 和 $w_1$, 且 $s_1$ 对于 Bob 来说是已知的, 因此 Bob 试图从方程 $s-d_1s_1-d_1w_1+r=0$ 中解出 $d_1$ 和 $w_1$, 该方程为二元一次方程. Bob 令 $d_1$ 在 $[1,q-1]$ 区间上遍历取值, 并计算与之对应的 $w_1=d_1^{-1}(s+r)-s_1$, 得到该方程在 $[1,q-1]$ 区间上的 $(q-1)$ 组解. Bob 从这 $(q-1)$ 组解中获取 $d_1$ 和 $w_1$ 的难度等同于穷举. 对于 Bob 而言, 相较于仅掌握签名验证公钥以及签名结果, 掌握上述信息并没有降低获取 Alice 的部分私钥 $d_1$ 以及随机数 $w_1$ 的难度. 假设 Bob 与 Alice 进行多次协作签名, 每次签名 Bob 都能够获取 1 个方程. 记第 $i$ 次签名 Bob 获取的方程为 $s^{[i]}-s_1^{[i]}d_1-w_1^{[i]}d_1+r^{[i]}=0$, 其中包含 2 个未知数, 分别为 $d_1$ 和 $w_1^{[i]}$. 每次签名 Alice 都会使用相同的部分私钥 $d_1$, 并独立随机生成或计算 1 个随机数 $w_1^{[i]}$. 因此 Bob 通过与 Alice 的 $n$ 次协作签名, 能够获取一个由 $n$ 个方程组成的方程组, 其中包含 $(n+1)$ 个未知数. Bob 令 $d_1$ 在 $[1,q-1]$ 区间遍历取值, 并计算与之对应的 $w_1^{[i]}=d_1^{-1}(s^{[i]}+r^{[i]})-s_1^{[i]}$, 得到该方程在 $[1,q-1]$ 区间上的 $(q-1)$ 组解. Bob 从这 $(q-1)$ 组解中获取 $d_1$ 和 $w_1^{[i]}$ 的难度等同于穷举. 对于 Bob 而言, 相较于仅掌握签名验证公钥以及签名结果, 掌握上述信息并没有降低获取 Alice 的部分私钥 $d_1$ 以及随机数 $w_1^{[i]}$ 的难度. 进一步, 根据签名随机数构造的安全要求, $w_1^{[i]}$ 总是包含至少 1 个独立随机生成的随机数, 因此在具体的两方门限计算方案中, Bob 也无法求解所得的方程组. 综上, Bob 无法从协作签名的交互过程中获取任何 Alice 的私密信息, 从而无法获取完整的签名私钥或另一部分私钥的任何信息.

假设存在一个外部敌手 Adv 能够获取公开信息以及签名生成过程的通信信息, 包括:

(1) 椭圆曲线点 $Q_1=[w_1]G$.

(2) 待签名消息的摘要 $e$.

(3) 签名第 2 部分 $s$ 的中间值 $s_1=d_2(r+w_2) \bmod q$.

(4) 签名第 1 部分 $r=(x_1+e) \bmod q$.

(5) 公开的签名第 2 部分 $s=(d_1(s_1+w_1)-r) \bmod q$.

(6) 公开的签名验证公钥 $P=[d_1^{-1}d_2^{-1}-1]G$.

由于 ECDLP 和哈希函数的安全性, Adv 无法从 $Q_1$、$P$ 和 $e$ 中获取任何信息. Adv 试图从方程 1: $s_1-rd_2-$

$w_2d_2=0$ 中解出 $d_2$ 和 $w_2$, 并从方程 2: $s-d_1s_1-d_1w_1+r=0$ 中解出 $d_1$ 和 $w_1$. 根据上述安全性分析可知, Alice 和 Bob 无法分别求解方程 1 和方程 2, 且 Adv 所掌握的信息少于 Alice 和 Bob. 因此, Adv 无法从方程 1 和方程 2 中解出 $d_1$、$d_2$、$w_1$、$w_2$. 进一步, 根据签名随机数构造的安全要求, $w_1$ 和 $w_2$ 总是包含至少 1 个独立随机生成的随机数, 因此在具体的两方门限计算方案中, 敌手 Adv 也无法求解所得的方程. 综上, Adv 无法从协作签名的交互过程中获取任何 Alice 和 Bob 的私密信息, 从而无法获取完整的签名私钥或任一参与方的部分私钥.

3.1.4 多随机数两方门限计算方案框架

第 3.1.2 节所描述的 Alice 与 Bob 协作计算椭圆曲线点 $Q$ 的过程中, Alice 使用随机数 $w_1$、Bob 使用随机数 $w_2$ 以及自己部分私钥 $d_2$ 计算 $Q=[w_2+d_2^{-1}w_1]G$. 由于此处 $d_2^{-1}$ 被用于计算 $Q$ 而非 $s$, 对 $d_2^{-1}$ 进行替换并不会影响最终签名的有效性. 进一步, Bob 可以不使用 $d_2^{-1}$, 而是替换成一个新的随机数 $w_3$, 使得 $Q=[w_2+w_1w_3]G$, 即签名随机数 $k=w_2+w_1w_3$. 这使得 $s$ 的协作计算过程发生一些变化, 此时有:

$$s=d_1d_2(w_2+w_1w_3+r)-r=d_1d_2w_2+d_1d_2w_1w_3+d_1d_2r-r=d_1(d_2r+d_2w_2)+d_1d_2w_1w_3-r=d_1s_1+(d_1w_1)(d_2w_3)-r.$$

为了实现 $s$ 的协作计算, Bob 不仅需要发送 $s_1$ 给 Alice, 还应发送 $s_2=d_2w_3$ 给 Alice. 然后 Alice 计算 $s=d_1(s_1+w_1s_2)-r$.

进一步, 本文对上述 $s$ 中的 $(d_1w_1)(d_2w_3)$ 部分进行扩展. 即 Bob 在协作计算 $s$ 时可以使用更多的随机数, 记为 $w_{\gamma+2},w_{\gamma+3},\ldots,w_{2\gamma+1}$, 将它们分别与自己的部分私钥 $d_2$ 相乘, 并发送给 Alice. Alice 也使用相应个数的随机数与之相乘, 记为 $w_1,w_2,\ldots,w_\gamma$, 从而计算得到 $s$. 综上, Alice 和 Bob 协作计算 SM2 签名的过程如下, 如图 2 所示.

Alice.1 置 $m'=m\|z$, 计算 $e=Hash(m')$.

Alice.2 随机生成或计算 $w_1,w_2,\ldots,w_\gamma\in[1,q-1]$, 计算 $Q_1=[w_1]G,\ldots,Q_\gamma=[w_\gamma]G$, 发送 $Q_1,\ldots,Q_\gamma$ 和 $e$ 给 Bob.

Bob.1 随机生成或计算 $w_{\gamma+1},w_{\gamma+2},\ldots,w_{2\gamma+1}\in[1,q-1]$, 计算 $Q=(x_1,y_1)=[w_{\gamma+1}]G+[w_{\gamma+2}]Q_1+\ldots+[w_{2\gamma+1}]Q_\gamma)$.

Bob.2 计算 $r=(x_1+e)\bmod q$.

Bob.3 计算 $s_1=d_2(r+w_{\gamma+1})\bmod q$, $s_2=(d_2w_{\gamma+2})\bmod q,\ldots,s_{\gamma+1}=(d_2w_{2\gamma+1})\bmod q$, 发送 $s_1,\ldots,s_{\gamma+1}$ 和 $r$ 给 Alice.

Alice.3 计算 $s=(d_1(s_1+w_1s_2+\ldots+w_\gamma s_{\gamma+1})-r)\bmod q$, 签名即为 $(r,s)$.

此时签名随机数 $k=w_{\gamma+1}+w_1w_{\gamma+2}+\ldots+w_\gamma w_{2\gamma+1}$.

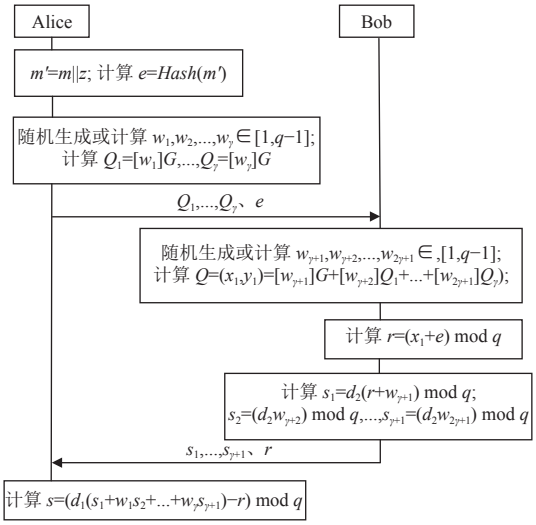


图 2 基于乘法密钥拆分的多随机数两方门限计算方案框架

在协作计算 SM2 数字签名的过程中, Alice 和 Bob 能够通过不同的方式构造各自的随机数, 对上述门限计算方案框架进行实例化. 随机数的构造应满足下述安全要求.

(1) 为了避免相关性, 随机数 $w_1,w_2,\ldots,w_\gamma$ 均包含至少 1 个独立随机生成的随机数 $k_i$, 且其与 $k_i$ 的取值分布都

是 $[1,q-1]$ 均匀分布, 也使得 Bob 无法从方程 $s=d_1(s_1+w_1s_2+\ldots+w_\gamma s_{\gamma+1})-r$ 中解出 Alice 的部分私钥 $d_1$.

(2) 随机数 $w_{\gamma+1}$ 包含至少 1 个独立随机生成的随机数 $k_i$, 且其与 $k_i$ 的取值分布都是 $[1,q-1]$ 均匀分布, 使得方程 $s_1=d_2(r+w_{\gamma+1})$ 对于 Alice 来说至少包含两个未知数 $d_2$ 和 $w_{\gamma+1}$ 从而无法解出 Bob 的部分私钥 $d_2$.

(3) 随机数 $w_{\gamma+2},\ldots,w_{2\gamma+1}$ 均包含至少 1 个独立随机生成的随机数 $k_i$, 且其与 $k_i$ 的取值分布都是 $[1,q-1]$ 均匀分布, 使得 Alice 无法从 $s_2,\ldots,s_{\gamma+1}$ 中获取 Bob 的部分私钥 $d_2$.

本文基于随机数 $k_i\ (i=1,2,\ldots)$, 结合签名私钥 $d$, Alice 和 Bob 的部分私钥 $d_1$、$d_2$ 以及它们的逆 $d_1^{-1}$、$d_2^{-1}$, 进行加或乘的组合, 实现签名随机数的不同构造. 例如, $w_1,\ldots,w_\gamma$ 可以是: $k_i$, $d_1k_i$, $d_1^{-1}k_i$, $1+d_1k_i$, $1+d_1^{-1}k_i$, $d_1+k_i$, $d_1^{-1}+k_i$, $k_i\ldots k_j$ ($k_i\ldots k_j$ 表示多个随机数相乘) 等; $w_{\gamma+1},\ldots,w_{2\gamma+1}$ 可以是: $k_i$, $d_2k_i$, $d_2^{-1}k_i$, $1+d_2k_i$, $1+d_2^{-1}k_i$, $d_2+k_i$, $d_2^{-1}+k_i$, $k_i\ldots k_j$ 等.

在具体的实例化方案中, $s$ 按照上文相应的步骤计算即可.

进一步, $w_i$ 还可以使用签名私钥 $d$ 进行构造. 虽然 Alice 和 Bob 无法获取签名私钥 $d$, 但是可以通过一定的形式转换, 使得 Alice 和 Bob 在不知道 $d$ 的情况下, 基于 $P$ 和 $G$ 完成签名随机数构造. 即随机数 $k$ 可进一步变化为如下等价形式:

$$\begin{aligned}k&=w_{\gamma+1}+w_1w_{\gamma+2}+\ldots+w_\gamma w_{2\gamma+1}\\&=d_1^{-1}d_2^{-1}\left(d_1d_2(w_{\gamma+1}+w_1w_{\gamma+2}+\ldots+w_\gamma w_{2\gamma+1})\right)\\&=(1+d)\left(d_1d_2(w_{\gamma+1}+w_1w_{\gamma+2}+\ldots+w_\gamma w_{2\gamma+1})\right)\\&=(1+d)d_1d_2w_{\gamma+1}+(1+d)d_1d_2w_1w_{\gamma+2}+\ldots+(1+d)d_1d_2w_\gamma w_{2\gamma+1}.\end{aligned}$$

令 $(1+d)d_2w_i=w'_i\ (i=\gamma+1,\ldots,2\gamma+1)$, $d_1w_i=w'_i\ (i=1,\ldots,\gamma)$, 使得签名随机数 $k=d_1w'_{\gamma+1}+w'_1w'_{\gamma+2}+\ldots+w'_\gamma w'_{2\gamma+1}$.

此时, Alice 和 Bob 能够通过如下步骤协作计算椭圆曲线点 $Q=[k]G$.

Alice.1 随机生成或计算 $w_1,w_2,\ldots,w_\gamma\in[1,q-1]$, 计算 $Q_1=[d_1](P+G)$, $Q_2=[d_1w_1](P+G),\ldots,Q_{\gamma+1}=\left[d_1w_\gamma\right](P+G)$, 发送 $Q_1,\ldots,Q_{\gamma+1}$ 给 Bob.

Bob.1 随机生成或计算 $w_{\gamma+1},w_{\gamma+2},\ldots,w_{2\gamma+1}\in[1,q-1]$, 计算 $Q=\left[d_2w_{\gamma+1}\right]Q_1+\left[d_2w_{\gamma+2}\right]Q_2+\ldots+\left[d_2w_{2\gamma+1}\right]Q_{\gamma+1}$.

虽然上述椭圆曲线点 $Q$ 的协作计算过程相较于不使用公钥 $P$ 的计算过程有所不同, 但签名随机数 $k$ 的构造相同, 都是 $k=w_{\gamma+1}+w_1w_{\gamma+2}+\ldots+w_\gamma w_{2\gamma+1}$. 因此, $s$ 仍然按照上文相应的步骤计算即可.

由于 $w'_i=d_1w_i\ (i=1,\ldots,\gamma)$, 根据 $w_1,\ldots,w_\gamma$ 的取值, $w'_1,\ldots,w'_\gamma$ 可以相应取值为: $d_1k_i$, $d_1^2k_i$, $k_i$, $d_1+d_1^2k_i$, $d_1+k_i$, $d_1^2+d_1k_i$, $1+d_1k_i$, $d_1k_i\ldots k_j$ 等.

由于 $w'_i=(1+d)d_2w_i\ (i=\gamma+1,\ldots,2\gamma+1)$, 根据 $w_{\gamma+1},\ldots,w_{2\gamma+1}$ 的取值, $w'_{\gamma+1},\ldots,w'_{2\gamma+1}$ 可以相应取值为: $(1+d)d_2k_i$, $(1+d)d_2^2k_i$, $(1+d)k_i$, $(1+d)\left(d_2+d_2^2k_i\right)$, $(1+d)(d_2+k_i)$, $(1+d)\left(d_2^2+d^2k_i\right)$, $(1+d)(1+d_2k_i)$, $(1+d)d_2k_i\ldots k_j$ 等.

限于篇幅, 本文列出如下几种实例化方案.

表 3 展示了取 $\gamma=3$ 时, 不包含签名私钥 $d$ 的签名随机数构造形式.

表 4 展示了取 $\gamma=3$ 时, 包含签名私钥 $d$ 的签名随机数构造形式. 序号 3.*.d 旨在表明该方案与表 3 中的方案 3.*为相同方案, 不同之处在于椭圆曲线点 $Q$ 的协作计算过程是否使用公钥 $P$, 相当于签名随机数是否使用私钥 $d$. 而最终计算的签名第 2 部分 $s$ 相同, 在表 4 中省略.

所列方案旨在展示如何通过随机数的构造对该框架进行实例化, 得到相应两方门限计算方案, 其中也包含了该框架下的已有公开方案. 方案 3.1, 3.9, 3.10, 3.11 为已公开方案[32−38], 其余均为新方案. 对随机数 $w_1,w_2,\ldots,w_{2\gamma+1}$ 进行相应的不同替换即可得到本文所提出的所有实例化方案. 值得说明的是, 以上实例化方案并非所有可能方案, 凡是满足随机数安全要求的构造, 均可进行实例化.

3.1.4.1 安全性分析

第 3.1.4 节所描述的两方门限签名计算的过程应满足如下安全要求: 任一参与方无法获取完整的签名私钥或另一部分私钥的任何信息.

安全性说明如下.

表 3　基于乘法密钥拆分的多随机数框架的实例化

| Alice | | | Bob | | | | $k=w_4+w_1w_5+w_2w_6+w_3w_7$ | $s=(1+d)^{-1}(k+r)-r$ | 序号 |
|---|---|---|---|---|---|---|---|---|---|
| $w_1$ | $w_2$ | $w_3$ | $w_4$ | $w_5$ | $w_6$ | $w_7$ | | | |
| $k_1$ | Null | Null | $k_2$ | $k_3$ | Null | Null | $k_2+k_1k_3$ | $d_1d_2k_2+d_1d_2k_1k_3+(d_1d_2-1)r$ | 3.1 |
| | | | | $d_2k_3$ | | | $k_2+d_2k_1k_3$ | $d_1d_2k_2+d_1d_2^2k_1k_3+(d_1d_2-1)r$ | 3.2 |
| | | | | $d_2^{-1}k_3$ | | | $k_2+d_2^{-1}k_1k_3$ | $d_1d_2k_2+d_1k_1k_3+(d_1d_2-1)r$ | 3.3 |
| | | | | $1+d_2k_3$ | | | $k_2+k_1+d_2k_1k_3$ | $d_1d_2k_2+d_1d_2k_1+d_1d_2^2k_1k_3+(d_1d_2-1)r$ | 3.4 |
| | | | | $1+d_2^{-1}k_3$ | | | $k_2+k_1+d_2^{-1}k_1k_3$ | $d_1d_2k_2+d_1d_2k_1+d_1k_1k_3+(d_1d_2-1)r$ | 3.5 |
| | | | | $d_2+k_3$ | | | $k_2+d_2k_1+k_1k_3$ | $d_1d_2k_2+d_1d_2^2k_1+d_1d_2k_1k_3+(d_1d_2-1)r$ | 3.6 |
| | | | | $d_2^{-1}+k_3$ | | | $k_2+d_2^{-1}k_1+k_1k_3$ | $d_1d_2k_2+d_1k_1+d_1d_2k_1k_3+(d_1d_2-1)r$ | 3.7 |
| | | | | $k_2k_3$ | | | $k_2(1+k_1k_3)$ | $d_1d_2k_2(1+k_1k_3)+(d_1d_2-1)r$ | 3.8 |
| | | | $k_2k_3$ | $k_3$ | Null | Null | $k_3(k_1+k_2)$ | $d_1d_2k_3(k_1+k_2)+(d_1d_2-1)r$ | 3.9 |
| $d_1^{-2}k_1$ | $k_2$ | $k_3$ | $k_4$ | $k_5$ | $d_2^{-2}k_6$ | $k_7$ | $d_1^{-2}k_1k_5+d_2^{-2}k_2k_6+k_3k_7+k_4$ | $d_1^{-1}d_2k_1k_5+d_1d_2^{-1}k_2k_6+d_1d_2k_4+d_1d_2k_3k_7+(d_1d_2-1)r$ | 3.10 |
| $d_1^{-1}k_1$ | $k_2$ | $k_3$ | $k_4$ | $k_5$ | $d_2^{-2}k_6$ | $k_7$ | $d_1^{-1}k_1k_5+d_2^{-2}k_2k_6+k_3k_7+k_4$ | $d_2k_1k_5+d_1d_2^{-1}k_2k_6+d_1d_2k_4+d_1d_2k_3k_7+(d_1d_2-1)r$ | 3.11 |

表 4　基于签名私钥的多随机数框架签名随机数构造

| Alice | | | Bob | | | | $k=d_1w'_4+w'_1w'_5+w'_2w'_6+w'_3w'_7$ | 序号 |
|---|---|---|---|---|---|---|---|---|
| $w'_1$ | $w'_2$ | $w'_3$ | $w'_4$ | $w'_5$ | $w'_6$ | $w'_7$ | | |
| $d_1k_1$ | Null | Null | $(1+d)d_2k_2$ | $(1+d)d_2k_3$ | Null | Null | $(1+d)d_1d_2(k_2+k_1k_3)$ | 3.1.d |
| | | | | $(1+d)d_2^2k_3$ | | | $(1+d)d_1d_2(k_2+d_2k_1k_3)$ | 3.2.d |
| | | | | $(1+d)k_3$ | | | $(1+d)d_1d_2\left(k_2+d_2^{-1}k_1k_3\right)$ | 3.3.d |
| | | | | $(1+d)\left(d_2+d_2^2k_3\right)$ | | | $(1+d)d_1d_2(k_2+k_1+d_2k_1k_3)$ | 3.4.d |
| | | | | $(1+d)(d_2+k_3)$ | | | $(1+d)d_1d_2\left(k_2+k_1+d_2^{-1}k_1k_3\right)$ | 3.5.d |
| | | | | $(1+d)\left(d_2^2+d_2k_3\right)$ | | | $(1+d)d_1d_2(k_2+d_2k_1+k_1k_3)$ | 3.6.d |
| | | | | $(1+d)(1+d_2k_3)$ | | | $(1+d)d_1d_2\left(k_2+d_2^{-1}k_1+k_1k_3\right)$ | 3.7.d |
| | | | | $(1+d)d_2k_2k_3$ | | | $(1+d)d_1d_2k_2(1+k_1k_3)$ | 3.8.d |
| | | | $(1+d)d_2k_2k_3$ | $(1+d)d_2k_3$ | Null | Null | $(1+d)d_1d_2k_3(k_1+k_2)$ | 3.9.d |
| $d_1^{-1}k_1$ | $d_1k_2$ | $d_1k_3$ | $(1+d)d_2k_4$ | $(1+d)d_2k_5$ | $(1+d)d_2^{-1}k_6$ | $(1+d)d_2k_7$ | $(1+d)d_1d_2(d_1^{-2}k_1k_5+d_2^{-2}k_2k_6+k_3k_7+k_4)$ | 3.10.d |
| $d_1k_1$ | $d_1k_2$ | $d_1k_3$ | $(1+d)d_2k_4$ | $(1+d)d_2k_5$ | $(1+d)d_2^{-1}k_6$ | $(1+d)d_2k_7$ | $(1+d)d_1d_2(d_1^{-1}k_1k_5+d_2^{-2}k_2k_6+k_3k_7+k_4)$ | 3.11.d |

在一次协作签名中, Alice 能够获取如下信息.

(1) 签名第 2 部分 $r=(x_1+e) \bmod q$.

(2) 签名第 1 部分 $s$ 的中间值 $s_1=d_2\left(r+w_{\gamma+1}\right) \bmod q$, $s_2=d_2w_{\gamma+2} \bmod q,\ldots,s_{\gamma+1}=d_2w_{2\gamma+1} \bmod q$.

其中, $r$ 是公开的信息且不包含任何 Bob 的私密信息, $s_1,\ldots,s_{\gamma+1}$ 包含 Bob 的私密信息 $d_2$, $w_{\gamma+1},\ldots,w_{2\gamma+1}$. Alice 试图求解下述方程组:

$$\begin{cases} s_1-rd_2-w_{\gamma+1}d_2=0 \\ s_2-d_2w_{\gamma+2}=0 \\ \vdots \end{cases},$$

该方程组包含$\gamma+1$个方程、$\gamma+2$个未知数. Alice 令$d_2$在$[1,q-1]$区间上遍历取值, 并计算与之对应的$w_j$ ($j\in[\gamma+1, 2\gamma+1]$), 其中$w_{\gamma+1}=d_2^{-1}(s_1-rd_2)$, $w_{\gamma+l}=d_2^{-1}s_l$ ($l\in[2,\gamma+1]$), 得到该方程组的$(q-1)$组解. Alice 从这$(q-1)$组解中获取$d_2$, $w_{\gamma+1},\ldots,w_{2\gamma+1}$的难度等同于穷举. 对于 Alice 而言, 相较于仅掌握签名验证公钥以及签名结果, 掌握上述信息并没有降低获取 Bob 的部分私钥$d_2$以及随机数$w_{\gamma+1},\ldots,w_{2\gamma+1}$的难度. 假设 Alice 与 Bob 进行多次协作签名, 每次签名 Alice 都能够获得一个由$\gamma+1$个方程组成的方程组, 记第$i$次签名 Alice 获取的方程组为:

$$\begin{cases} s_1^{[i]}-r^{[i]}d_2-w_{\gamma+1}^{[i]}d_2=0 \\ s_2^{[i]}-d_2w_{\gamma+2}^{[i]}=0 \\ \quad\vdots \end{cases},$$

其中包含$\gamma+2$个未知数, 分别为$d_2$和$w_{\gamma+1}^{[i]},\ldots,w_{2\gamma+1}^{[i]}$. 每次签名 Bob 都会使用相同的部分私钥$d_2$, 并独立随机生成或计算$(\gamma+1)$个随机数$w_{\gamma+1}^{[i]},\ldots,w_{2\gamma+1}^{[i]}$. 因此 Alice 通过与 Bob 的$n$次协作签名, 能够获取一个由$n\gamma+1$个方程组成的方程组, 其中包含$n(\gamma+1)+1$个未知数. Alice 令$d_2$在$[1,q-1]$区间上遍历取值, 并计算与之对应的$w_j^{[i]}$($j\in[\gamma+1, 2\gamma+1]$), 其中$w_{\gamma+1}^{[i]}=d_2^{-1}\left(s_1^{[i]}-r^{[i]}d_2\right)$, $w_{\gamma+l}^{[i]}=d_2^{-1}s_l^{[i]}$ ($l\in[2,\gamma+1]$), 得到该方程组的$(q-1)$组解. Alice 从这$(q-1)$组解中获取$d_2$, $w_{\gamma+1}^{[i]},\ldots,w_{2\gamma+1}^{[i]}$的难度等同于穷举. 对于 Alice 而言, 相较于仅掌握签名验证公钥以及签名结果, 掌握上述信息并没有降低获取 Bob 的部分私钥$d_2$以及随机数$w_{\gamma+1}^{[i]},\ldots,w_{2\gamma+1}^{[i]}$的难度. 进一步, 根据签名随机数构造的安全要求, $w_{\gamma+1}^{[i]},\ldots,w_{2\gamma+1}^{[i]}$都至少包含 1 个独立随机生成的随机数, 因此在具体的两方门限计算方案中, Alice 也无法求解所得的方程组. 综上, Alice 无法从协作签名的交互过程中获取任何 Bob 的私密信息, 从而无法获取完整的签名私钥或另一部分私钥的任何信息.

在一次协作签名中, Bob 能够获取如下信息.

(1) 椭圆曲线点$Q_1=[w_1]G,\ldots,Q_\gamma=[w_\gamma]G$.

(2) 待签名消息的摘要$e$.

(3) 公开的签名第 2 部分$s=(d_1(s_1+w_1s_2+\ldots+w_\gamma s_{\gamma+1})-r) \bmod q$.

由于 ECDLP 和哈希函数的安全性, Bob 无法从$Q_1$和$e$中获取任何信息. 而$s$包含 Alice 的私密信息$d_1$和$w_1,w_2,\ldots,w_\gamma$, 且$s_1,\ldots,s_{\gamma+1}$对于 Bob 来说是已知的, 因此 Bob 试图求解方程$s-s_1d_1-\ldots-s_{\gamma+1}w_\gamma d_1+r=0$. 该方程包含$(\gamma+1)$个未知数. Bob 令$d_1,w_1,\ldots,w_{\gamma-1}$分别在$[1,q-1]$区间上遍历取值, 并计算与之对应的$w_\gamma=d_1^{-1}s_{\gamma+1}^{-1}\left(s-s_1d_1-\ldots-s_\gamma w_{\gamma-1}d_1+r\right)$, 得到该方程在$[1,q-1]$上的$(q-1)^\gamma$组解. Bob 从这$(q-1)^\gamma$组解中获取$d_1$和$w_1,w_2,\ldots,w_\gamma$的难度远大于穷举. 对于 Bob 而言, 相较于仅掌握签名验证公钥以及签名结果, 掌握上述信息并没有降低获取 Alice 的部分私钥$d_1$以及随机数$w_1,w_2,\ldots,w_\gamma$的难度. 假设 Bob 与 Alice 进行多次协作签名, 每次签名 Bob 都能够获取 1 个方程, 记第$i$次签名 Bob 获取的方程组为$s^{[i]}-s_1^{[i]}d_1-\ldots-s_{\gamma+1}^{[i]}w_\gamma^{[i]}d_1+r^{[i]}=0$. 其中包含$(\gamma+1)$个未知数, 分别为$d_1$和$w_1^{[i]},\ldots,w_\gamma^{[i]}$. 每次签名 Alice 都会使用相同的部分私钥$d_1$, 并独立随机生成或计算$\gamma$个随机数$w_1^{[i]},\ldots,w_\gamma^{[i]}$. 因此 Bob 通过与 Alice 的$n$次协作签名, 能够获取一个由$n$个方程组成的方程组, 其中包含$(n\gamma+1)$个未知数. Bob 令$d_1,w_1^{[i]},\ldots,w_{\gamma-1}^{[i]}$分别在$[1,q-1]$区间上遍历取值, 并计算与之对应的$w_\gamma^{[i]}=d_1^{-1}s_{\gamma+1}^{[i]-1}(s^{[i]}-s_1^{[i]d_1}-\ldots-s_\gamma^{[i]}w_{\gamma-1}^{[i]}d_1+r^{[i]})$, 得到该方程组在$[1,q-1]$区间上的$(q-1)^{n(\gamma-1)+1}$组解. Bob 从这$(q-1)^{n(\gamma-1)+1}$组解中获取$d_1$和$w_1^{[i]},w_2^{[i]},\ldots,w_\gamma^{[i]}$的难度远大于穷举. 对于 Bob 而言, 相较于仅掌握签名验证公钥以及签名结果, 掌握上述信息并没有降低获取 Alice 的部分私钥$d_1$以及随机数$w_1^{[i]},w_2^{[i]},\ldots,w_\gamma^{[i]}$的难度. 进一步, 根据签名随机数构造的安全要求, $w_1^{[i]},\ldots,w_\gamma^{[i]}$总是包含至少 1 个独立随机生成的随机数, 因此在具体的两方门限计算方案中, Bob 也无法求解所得的方程组. 综上, Bob 无法从协作签名的交互过程中获取任何 Alice 的私密信息, 从而无法获取完整的签名私钥或另一部分私钥的任何信息.

假设存在一个外部敌手 Adv 能够获取公开信息以及签名生成过程的通信信息, 包括:

(1) 椭圆曲线点$Q_1=[w_1]G,\ldots,Q_\gamma=[w_\gamma]G$.

(2) 待签名消息的摘要$e$.

(3) 签名第 1 部分$s$的中间值$s_1=d_2(r+w_{\gamma+1}) \bmod q$, $s_2=d_2w_{\gamma+2} \bmod q,\ldots,s_{\gamma+1}=d_2w_{2\gamma+1} \bmod q$.

(4) 签名第 2 部分$r=(x_1+e) \bmod q$.

(5) 公开的签名第 2 部分 $s=(d_1(s_1+w_1s_2+\ldots+w_\gamma s_{\gamma+1})-r) \bmod q$.

(6) 公开的签名验证公钥 $P=\left[d_1^{-1}d_2^{-1}-1\right]G$.

由于 ECDLP 和哈希函数的安全性, Adv 无法从 $Q_1,\ldots,Q_{\gamma+1}$、$P$ 和 $e$ 中获取任何信息. Adv 试图从方程组 1:

$$\begin{cases} s_1-rd_2-w_{\gamma+1}d_2=0 \\ s_2-d_2w_{\gamma+2}=0 \\ \quad\vdots \end{cases}$$

中解出 $d_2$ 和 $w_{\gamma+1},\ldots,w_{2\gamma+1}$, 并从方程 2: $s-s_1d_1-\cdots-s_{\gamma+1}w_\gamma d_1+r=0$ 中解出 $d_1$ 和 $w_1,w_2,\ldots,w_\gamma$.

根据上述安全性分析可知, Alice 和 Bob 无法分别求解方程组 1 和方程 2, 且 Adv 所掌握的信息少于 Alice 和 Bob. 因此, Adv 无法从方程组 1 和方程 2 中解出 $d_1$、$d_2$、$w_1,\ldots,w_{2\gamma+1}$. 进一步, 根据签名随机数构造的安全要求, $w_1,\ldots,w_{2\gamma+1}$ 总是包含至少 1 个独立随机生成的随机数, 因此在具体的两方门限计算方案中, 敌手 Adv 也无法求解所得的方程 (组). 综上, Adv 无法从协作签名的交互过程中获取任何 Alice 和 Bob 的私密信息, 从而无法获取完整的签名私钥或任一参与方的部分私钥.

### 3.2 基于加法密钥拆分的门限计算框架

#### 3.2.1 密钥生成

根据第 2.1 节的分析, 本文签名私钥 $d$ 的表达式对 $(1+d)^{-1}$ 进行加法拆分, 使得 Alice 和 Bob 分别持有部分私钥 $d_1$ 和 $d_2$, $d$、$d_1$、$d_2$ 满足如下关系:

$$(1+d)^{-1}=d_1+d_2,$$

由上式可知 $d=(d_1+d_2)^{-1}-1$, 即公钥 $P=[d]G=\left[(d_1+d_2)^{-1}\right]G-G$.

为了实现上述加法密钥拆分, Alice 和 Bob 使用 MtA 协议将乘法份额分享转变成加法份额分享, 并通过如下步骤实现密钥生成.

(1) Alice 和 Bob 实现第 3.1.1 节中基于乘法密钥拆分的密钥生成, 分别得到乘法分享份额 $d_1'$ 和 $d_2'$.

(2) Alice 和 Bob 共同运行 MtA 协议, 分别以 $d_1'$ 和 $d_2'$ 作为 MtA 的输入, Alice 得到输出 $d_1$, Bob 得到输出 $d_2$, 满足 $d_1'd_2'=d_1+d_2$. 完成加法密钥拆分.

上述密钥生成过程应满足如下安全要求: 任一参与方无法获得完整签名私钥.

安全性说明如下. 上述密钥生成过程由乘法密钥拆分和 MtA 协议组成, 由于第 3.1.3 节中已说明了乘法密钥拆分的安全性, 且 MtA 协议是安全的, 因此上述密钥生成过程也是安全.

#### 3.2.2 签名生成

在每一次签名计算过程中, Alice 和 Bob 需要协作计算一个随机的椭圆曲线点 $Q=[k]G$. 由于 $s=(1+d)^{-1}(k+r)-r$, 其中 $r$ 和 $s$ 为公开的签名, 若签名随机数 $k$ 仅由一个参与方生成, 则该参与方可以计算得到签名私钥 $d=(s+r)^{-1}(k+r)-1$, 违背了安全要求. 因此, 在协作计算椭圆曲线点的过程中, Alice 和 Bob 都需要构造各自的随机数, 即签名随机数 $k$ 由 Alice 和 Bob 各自构造的随机数共同组成, 任一参与方无法获取完整的签名随机数.

本文先讨论 $s$ 的协作计算过程, 再根据 $s$ 的构造讨论椭圆曲线点 $Q$ 的协作计算过程. 已知:

$$s=(1+d)^{-1}(k+r)-r=(d_1+d_2)(k+r)-r=(d_1+d_2)k+d_1r+d_2r-r.$$

为了实现 $s$ 的协作计算, Bob 应发送 $d_2r$ 给 Alice. 由于 $r$ 是公开信息, 为了防止 Alice 计算得到 Bob 的私钥 $d_2$, Bob 应构造一个随机数 $w_2$ 来保护自己的部分私钥, 于是 Bob 计算 $s_1=d_2(r+w_2)$ 并发送给 Alice. Alice 接收到 $s_1$ 后也应构造一个随机数 $w_1$ 来保护自己的部分私钥, 于是 Alice 计算:

$$\begin{aligned} s&=d_1(r+w_1)+s_1-r=d_1r+d_1w_1+d_2r+d_2w_2-r \\ &=(d_1+d_2)((d_1+d_2)^{-1}(d_1w_1+d_2w_2)+r)-r, \end{aligned}$$

这是一个有效的 SM2 签名, 签名随机数 $k=(d_1+d_2)^{-1}(d_1w_1+d_2w_2)$.

综上, Alice 和 Bob 协作计算 SM2 签名的过程如下, 如图 3 所示.

Alice.1 置 $m'=m\|z$, 计算 $e=Hash(m')$.

Alice.2 随机生成或计算 $w_1\in[1,q-1]$, 计算 $Q_1=[d_1w_1](P+G)$, 发送 $Q_1$ 和 $e$ 给 Bob.

Bob.1 随机生成或计算 $w_2 \in [1, q-1]$, 计算 $Q = (x_1, y_1) = [d_2 w_2](P+G) + Q_1$.

Bob.2 计算 $r = (x_1 + e) \bmod q$.

Bob.3 计算 $s_1 = d_2 (r + w_2) \bmod q$, 发送 $s_1$ 和 $r$ 给 Alice.

Alice.3 计算 $s = (d_1 (r + w_1) + s_1 - r) \bmod q$, 签名对即为 $(r, s)$.

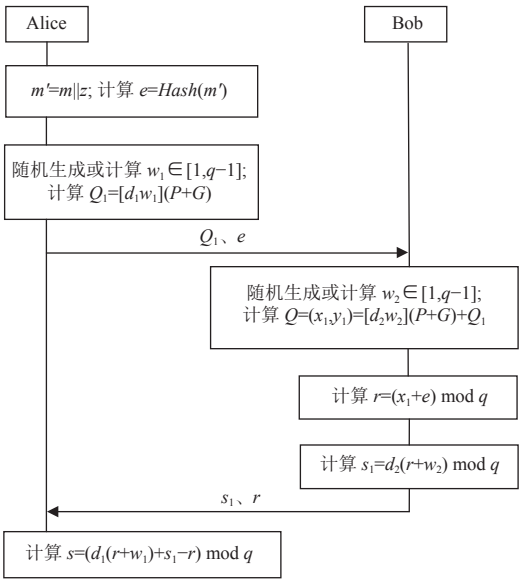


图 3　基于加法密钥拆分的 2 随机数两方门限计算方案框架

在协作计算 SM2 数字签名的过程中, Alice 和 Bob 能够通过不同的方式构造各自的随机数, 对上述门限计算方案框架进行实例化. 随机数的构造应满足下述安全要求.

(1) 随机数 $w_1$ 中至少包含 1 个独立随机生成的随机数 $k_1$, 且 $w_1$ 与 $k_1$ 的取值分布都是 $[1, q-1]$ 均匀分布, 使得方程 $s = d_1 (r + w_1) + s_1 - r$ 对于 Bob 来说包含至少两个未知数 $d_1$ 和 $w_1$, 从而无法解出 Alice 的私钥 $d_1$.

(2) 随机数 $w_2$ 中至少包含 1 个独立随机生成的随机数 $k_2$, 且 $w_2$ 与 $k_2$ 的取值分布都是 $[1, q-1]$ 均匀分布, 使得方程 $s_1 = d_2 (r + w_2)$ 对于 Alice 来说至少包含两个未知数 $d_2$ 和 $w_2$, 从而无法解出 Bob 的私钥 $d_2$.

本文基于随机数 $k_1$ 和 $k_2$, 结合 Alice 和 Bob 的部分私钥 $d_1$、$d_2$ 以及它们的逆 $d_1^{-1}$、$d_2^{-1}$, 进行加或乘的组合, 实现签名随机数的不同构造. 例如, $w_i$ $(i = 1, 2)$ 可以是: $k_i$, $d_i^{-1} k_i$, $d_i^{-2} k_i$, $1 + d_i^{-1} k_i$, $1 + d_i^{-2} k_i$, $d_i^{-1} + k_i$, $d_i^{-2} + k_i$ 等.

限于篇幅, 本文列出如下几种实例化方案, 如表 5 所示.

表 5　基于加法密钥拆分 2 随机数框架的实例化

| $w_1$ | $w_2$ | $k = (d_1 + d_2)^{-1} (d_1 w_1 + d_2 w_2)$ | $s = (1 + d)^{-1} (k + r) - r$ | 序号 |
|---|---|---|---|---|
| | $k_2$ | $(d_1 + d_2)^{-1} (d_1 k_1 + d_2 k_2)$ | $d_1 k_1 + d_2 k_2 + (d_1 + d_2 - 1) r$ | 5.1 |
| | $d_2^{-1} k_2$ | $(d_1 + d_2)^{-1} (d_1 k_1 + k_2)$ | $d_1 k_1 + k_2 + (d_1 + d_2 - 1) r$ | 5.2 |
| | $d_2^{-2} k_2$ | $(d_1 + d_2)^{-1} (d_1 k_1 + d_2^{-1} k_2)$ | $d_1 k_1 + d_2^{-1} k_2 + (d_1 + d_2 - 1) r$ | 5.3 |
| $k_1$ | $1 + d_2^{-1} k_2$ | $(d_1 + d_2)^{-1} (d_1 k_1 + d_2 + k_2)$ | $d_1 k_1 + d_2 + k_2 + (d_1 + d_2 - 1) r$ | 5.4 |
| | $1 + d_2^{-2} k_2$ | $(d_1 + d_2)^{-1} (d_1 k_1 + d_2 + d_2^{-1} k_2)$ | $d_1 k_1 + d_2 + d_2^{-1} k_2 + (d_1 + d_2 - 1) r$ | 5.5 |
| | $d_2^{-1} + k_2$ | $(d_1 + d_2)^{-1} (d_1 k_1 + 1 + d_2 k_2)$ | $d_1 k_1 + 1 + d_2 k_2 + (d_1 + d_2 - 1) r$ | 5.6 |
| | $d_2^{-2} + k_2$ | $(d_1 + d_2)^{-1} \left(d_1 k_1 + d_2^{-1} + d_2 k_2\right)$ | $d_1 k_1 + d_2^{-1} + d_2 k_2 + (d_1 + d_2 - 1) r$ | 5.7 |
| $d_1^{-1} k_1$ | $d_2^{-1} k_2$ | $(d_1 + d_2)^{-1} (k_1 + k_2)$ | $k_1 + k_2 + (d_1 + d_2 - 1) r$ | 5.8 |

所列方案旨在展示如何通过随机数的构造对该框架进行实例化, 得到相应两方门限计算方案, 其中也包含了该框架下的已有公开方案. 方案 5.8 为已有公开方案[39], 其余均为新方案. 对随机数 $w_1$ 和 $w_2$ 进行相应的不同替换即可得到本文提出的所有实例化方案, 完整的实例化方案表格可见附录 A 中的表 A2. 值得说明的是, 以上列出的实例化方案并非所有可能方案, 凡是满足随机数安全要求的构造, 均可用于实例化得到两方门限计算方案.

虽然可以参照第 3.1.4 节中乘法拆分的方式将签名随机数中的 $d_1w_1$ 和 $d_2w_2$ 替换为 $w_1w_3$ 和 $w_2w_4$ , 此时 $k=w_1w_3+w_2w_4=d_1\left(d_1^{-1}w_1w_3\right)+d_2\left(d_2^{-1}w_2w_4\right)$. 但是对于任意形式的 $k=w_1w_3+w_2w_4$ , 均可由对 $k=d_1w_1'+d_2w_2'$ 进行如下取值来实现:

$$w_1'=d_1^{-1}w_1w_3,$$
$$w_2'=d_2^{-1}w_2w_4,$$

即 $k=d_1w_1+d_2w_2$ 已包含了 $k=w_1w_3+w_2w_4$ 的所有可能构造, 因此这种替换并没有实际意义.

3.2.3 安全性分析

第 3.2.2 节所描述的两方门限签名计算过程应满足如下安全要求: 任一参与方无法获取完整的签名私钥或另一部分私钥的任何信息.

安全性说明如下.

在一次协作签名中, Alice 能够获取如下信息.

(1) 签名第 2 部分 $r=(x_1+e) \bmod q$.

(2) 签名第 1 部分 $s$ 的中间值 $s_1=d_2(r+w_2) \bmod q$.

其中, $r$ 是公开的信息且不包含任何 Bob 的私密信息, $s_1$ 包含 Bob 的私密信息 $d_2$ 和 $w_2$. Alice 试图从方程 $s_1-rd_2-w_2d_2=0$ 中解出 $d_2$ 和 $w_2$, 该方程为二元一次方程. Alice 令 $d_2$ 在 $[1,q-1]$ 区间上遍历取值, 并计算与之对应的 $w_2=d_2^{-1}s_1-r$, 得到该方程在 $[1,q-1]$ 区间上的 $(q-1)$ 组解. Alice 从这 $(q-1)$ 组解中获取 $d_2$ 和 $w_2$ 的难度等同于穷举. 对于 Alice 而言, 相较于仅掌握签名验证公钥以及签名结果, 掌握上述信息并没有降低获取 Bob 的部分私钥 $d_2$ 以及随机数 $w_2$ 的难度. 假设 Alice 与 Bob 进行多次协作签名, 每次签名 Alice 都能够获得 1 个方程. 记第 $i$ 次签名 Alice 获取的方程为 $s_1^{[i]}-r^{[i]}d_2-w_2^{[i]}d_2=0$, 其中包含 2 个未知数, 分别为 $d_2$ 和 $w_2^{[i]}$. 每次签名 Bob 都会使用相同的部分私钥 $d_2$, 并独立随机生成或计算 1 个随机数 $w_2^{[i]}$. 因此 Alice 通过与 Bob 的 $n$ 次协作签名, 能够获得一个由 $n$ 个方程组成的方程组, 其中包含 $(n+1)$ 个未知数. Alice 令 $d_2$ 在 $[1,q-1]$ 区间上遍历取值, 并计算与之对应的 $w_2^{[i]}=d_2^{-1}s_1^{[i]}-r^{[i]}$, 得到该方程在 $[1,q-1]$ 区间上的 $(q-1)$ 组解. Alice 从这 $(q-1)$ 组解中获取 $d_2$ 和 $w_2^{[i]}$ 的难度等同于穷举. 对于 Alice 而言, 相较于仅掌握签名验证公钥以及签名结果, 掌握上述信息并没有降低获取 Bob 的部分私钥 $d_2$ 以及随机数 $w_2^{[i]}$ 的难度. 进一步, 根据签名随机数构造的安全要求, $w_2^{[i]}$ 总是包含至少 1 个独立随机生成的随机数. 因此在具体的两方门限计算方案中, Alice 也无法求解所得的方程组. 综上, Alice 无法从协作签名的交互过程中获取任何 Bob 的私密信息, 从而无法获取完整的签名私钥或另一部分私钥的任何信息.

在一次协作签名中, Bob 能够获取如下信息.

(1) 椭圆曲线点 $Q_1=[d_1w_1](P+G)$.

(2) 待签名消息的摘要 $e$.

(3) 公开的签名第 2 部分 $s=(d_1(r+w_1)+s_1-r) \bmod q$.

由于 ECDLP 和哈希函数的安全性, Bob 无法从 $Q_1$ 和 $e$ 中获取任何信息. 而 $s$ 包含 Alice 的私密信息 $d_1$ 和 $w_1$, 且 $s_1$ 对于 Bob 来说是已知的, 因此 Bob 试图从方程 $s-(r+w_1)d_1-s_1+r=0$ 中解出 $d_1$ 和 $w_1$, 该方程为二元一次方程. Bob 令 $d_1$ 在 $[1,q-1]$ 区间上遍历取值, 并计算与之对应 $w_1=d_1^{-1}(s+r-s_1)-r$ 的, 得到该方程在 $[1,q-1]$ 区间上的 $(q-1)$ 组解. Bob 从这 $(q-1)$ 组解中获取 $d_2$ 和 $w_2$ 的难度等同于穷举. 对于 Bob 而言, 相较于仅掌握签名验证公钥以及签名结果, 掌握上述信息并没有降低获取 Alice 的部分私钥 $d_1$ 以及随机数 $w_1$ 的难度. 假设 Bob 与 Alice 进行多次协作签名, 每次签名 Bob 都能够获取 1 个方程. 记第 $i$ 次签名 Bob 获取的方程为 $s^{[i]}-\left(r^{[i]}+w_1^{[i]}\right)d_1-s_1^{[i]}+r^{[i]}=0$, 其中包含 2 个未知数, 分别为 $d_1$ 和 $w_1^{[i]}$. 每次签名 Alice 都会使用相同的部分私钥 $d_1$, 并独立随机生成或计

算 1 个随机数 $w_1^{[i]}$. 因此 Bob 通过与 Alice 的 $n$ 次协作签名, 能够获取一个由 $n$ 个方程组成的方程组, 其中包含 $(n+1)$ 个未知数. Bob 令 $d_1$ 在 $[1,q-1]$ 区间上遍历取值, 并计算与之对应的 $w_1^{[i]}=d_1^{-1}\left(s^{[i]}+r^{[i]}-s_1^{[i]}\right)-r^{[i]}$, 得到该方程在 $[1,q-1]$ 区间上的 $(q-1)$ 组解. Bob 从这 $(q-1)$ 组解中获取 $d_1$ 和 $w_1^{[i]}$ 的难度等同于穷举. 对于 Bob 而言, 相较于仅掌握签名验证公钥以及签名结果, 掌握上述信息并没有降低获取 Alice 的部分私钥 $d_1$ 以及随机数 $w_1^{[i]}$ 的难度. 进一步, 根据签名随机数构造的安全要求, $w_1^{[i]}$ 总是包含至少 1 个独立随机生成的随机数, 因此在具体的两方门限计算方案中, Bob 也无法求解所得的方程组. 综上, Bob 无法从协作签名的交互过程中获取任何 Alice 的私密信息, 从而无法获取完整的签名私钥或另一部分私钥的任何信息.

假设存在一个外部敌手 Adv 能够获取公开信息以及签名生成过程的通信信息, 包括:

(1) 椭圆曲线点 $Q_1=[d_1w_1](P+G)$.

(2) 待签名消息的摘要 $e$.

(3) 签名第 2 部分 $r=(x_1+e) \bmod q$.

(4) 签名第 1 部分 $s$ 的中间值 $s_1=d_2(r+w_2) \bmod q$.

(5) 公开的签名第 2 部分 $s=(d_1(r+w_1)+s_1-r) \bmod q$.

(6) 公开的签名验证公钥 $P=\left[d_1^{-1}d_2^{-1}-1\right]G$.

由于 ECDLP 和哈希函数的安全性, Adv 无法从 $Q_1$、$P$ 和 $e$ 中获取任何信息. Adv 试图从方程 1: $s_1-rd_2-w_2d_2=0$ 中解出 $d_2$ 和 $w_2$, 并从方程 2: $s-(r+w_1)d_1-s_1+r=0$ 中解出 $d_1$ 和 $w_1$. 根据上述安全性分析可知, Alice 和 Bob 无法分别求解方程 1 和方程 2, 且 Adv 所掌握的信息少于 Alice 和 Bob. 因此, Adv 无法从方程 1 和方程 2 中解出 $d_1$、$d_2$、$w_1$、$w_2$. 进一步, 根据签名随机数构造的安全要求, $w_1$ 和 $w_2$ 总是包含至少 1 个独立随机生成的随机数, 因此在具体的两方门限计算方案中, 敌手 Adv 也无法求解所得的方程. 综上, Adv 无法从协作签名的交互过程中获取任何 Alice 和 Bob 的私密信息, 从而无法获取完整的签名私钥或任一参与方的部分私钥.

## 4 性能分析

Alice 和 Bob 使用一对公私钥多次进行两方协作签名, 密钥生成过程只执行一次. 因此, 本文只讨论签名生成过程的计算消耗和通信消耗. 签名生成过程的计算消耗主要由点乘、点加、数乘和数加运算组成. 相关文献[40]指出, 点乘运算的计算消耗大约是点加运算的 300 倍, 且远大于数乘和数加运算的计算消耗, 因此本文只讨论签名生成过程中的点乘计算消耗. 以下分别讨论不同 SM2 两方门限计算方案框架的计算消耗以及通信消耗, 如表 6 所示.

表 6 性能分析

| 框架 | 已知点点乘/次 | 未知点点乘/次 | 离线预计算 | 在线计算 | 通信轮数 | 通信量(bit) |
|---|---|---|---|---|---|---|
| 乘法密钥拆分、2随机数 | 2 | 1 | $[w_1]G,[w_2]G$ | $\left[d_2^{-1}\right]Q_1$ | 2 | 1 280 |
| 乘法密钥拆分、多随机数 | $\gamma+1$ | $\gamma$ | $[w_1]G,\ldots,[w_{\gamma+1}]G$ | $[w_{\gamma+2}]Q_1,\ldots,[w_{2\gamma+1}]Q_\gamma$ | 2 | $768\gamma+512$ |
| 加法密钥拆分、服务器/客户端1-to-1 | 2 | 0 | $[d_1w_1](P+G),[d_2w_2](P+G)$ | Null | 2 | 1 280 |
| 加法密钥拆分、服务器/客户端1-to-n | 1 | 1 | $[d_1w_1](P+G)$ | $[d_2w_2](P+G)$ | 2 | 1 280 |

### 4.1 基于乘法密钥拆分的 2 随机数两方门限计算方案框架

该框架所给出的签名生成过程包含 3 次点乘运算, 分别为 Alice 端的一次基点点乘 $[w_1]G$, Bob 端的一次基点点乘 $[w_2]G$ 和一次未知点点乘 $\left[d_2^{-1}\right]Q_1$. 由于基点 $G$ 为已知点, 且随机数 $w_1$ 和 $w_2$ 分别由 Alice 和 Bob 独立生成, 因此点乘 $[w_1]G$ 和 $[w_2]G$ 可以离线预计算, 将点乘结果分别存储, 节省签名生成的计算消耗. 而点乘 $\left[d_2^{-1}\right]Q_1$ 中的 $Q_1$ 是签名生成过程中 Alice 发送给 Bob 的点, 对于 Bob 而言是未知点, 因此只能在线计算.

该框架所给出的签名生成过程包含两次通信, 分别是 Alice 给 Bob 发送椭圆曲线点 $Q_1$ 和消息摘要 $e$, 以及 Bob 给 Alice 发送签名中间值 $s_1$ 和签名第 2 部分 $r$. 根据 SM2 椭圆曲线的标准参数可知, 椭圆曲线点 $Q_1$ 包含两个 256 bit

的坐标, 消息摘要$e$、签名中间值$s_1$和签名第 2 部分$r$的长度均为 256 bit, 因此该签名生成过程的通信量为 1 280 bit.

### 4.2 基于乘法密钥拆分的多随机数两方门限计算方案框架

该框架所给出的签名生成过程的计算消耗与随机数的数量相关, 当随机数数量为$(2\gamma+1)$时, 签名生成过程包含$(2\gamma+1)$个点乘. 分别为 Alice 端的$\gamma$次基点点乘$Q_1=[w_1]G,\ldots,Q_\gamma=\left[w_\gamma\right]G$, Bob 端的一次基点点乘$\left[w_{\gamma+1}\right]G$和$\gamma$次未知点点乘$\left[w_{\gamma+2}\right]Q_1,\ldots,\left[w_{2\gamma+1}\right]Q_\gamma$. 由于基点$G$为已知点, 且随机数$w_1,w_2,\ldots,w_\gamma$和$w_{\gamma+1}$分别由 Alice 和 Bob 独立生成, 因此点乘$Q_1=[w_1]G,\ldots,Q_\gamma=\left[w_\gamma\right]G$和$\left[w_{\gamma+1}\right]G$可以离线预计算, 将点乘结果分别存储, 节省签名生成的计算消耗. 而点乘$\left[w_{\gamma+2}\right]Q_1,\ldots,\left[w_{2\gamma+1}\right]Q_\gamma$中的$Q_1,\ldots,Q_\gamma$是签名生成过程中 Alice 发送给 Bob 的点, 对于 Bob 而言是未知点, 因此只能在线计算.

该框架所给出的签名生成过程包含两次通信, 分别是 Alice 给 Bob 发送椭圆曲线点$Q_1,\ldots,Q_\gamma$和消息摘要$e$, 以及 Bob 给 Alice 发送签名中间值$s_1,\ldots,s_{\gamma+1}$和签名第 2 部分$r$. 每个椭圆曲线点都包含两个 256 bit 的坐标, 消息摘要$e$签名中间值$s_1,\ldots,s_{\gamma+1}$和签名第 2 部分$r$的长度均为 256 bit, 因此该签名生成过程的通信量为$(768\gamma+512)$ bit.

### 4.3 基于加法密钥拆分的两方门限计算方案框架

该框架所给出的签名生成过程包含两次点乘运算, 分别为 Alice 端的一次点乘$[d_1w_1](P+G)$, 和 Bob 端的一次点乘$[d_2w_2](P+G)$. 对于 Alice 而言, $P$和$G$均为已知点, 且$w_1$由 Alice 独立生成. 因此点乘$[d_1w_1](P+G)$可被视为已知点点乘, 且可以离线预计算, 将点乘结果存储在 Alice 端, 节省签名生成的计算消耗. 对于 Bob 而言, $P$和$G$均为已知点. 若 Bob 只与一个 Alice 协作计算签名, 即 Bob 与 Alice 为 1-to-1 的关系时, 点乘$[d_2w_2](P+G)$可被视为已知点点乘, 可以离线预计算, 将点乘结果存储在 Bob 端, 节省签名生成的计算消耗; 若 Bob 作为签名服务器, 掌握多个 Alice 的签名验证公钥, 即 Bob 与 Alice 为 1-to-n 的关系时, 由于存储资源的限制, 无法对每个 Alice 的公钥点进行离线预计算, 此时点乘$[d_2w_2](P+G)$被视为未知点点乘, 只能在线计算.

该框架所给出的签名生成过程包含两次通信, 分别是 Alice 给 Bob 发送椭圆曲线点$Q_1$和消息摘要$e$, 以及 Bob 给 Alice 发送签名中间值$s_1$和签名第 2 部分$r$. 椭圆曲线点$Q_1$包含两个 256 bit 的坐标, 消息摘要$e$、签名中间值$s_1$和签名第 2 部分$r$的长度均为 256 bit, 因此该签名生成过程的通信量为 1 280 bit.

综上所述, 基于加法密钥拆分的两方门限计算方案框架的性能最优. 在已有方案以及本文实例化得到的新方案中, 袁峰等人[39]提出的方案性能最优. 当 Bob 和 Alice 为 1-to-1 关系时, 该方案的签名生成需要 2 次已知点点乘, 且 2 次点乘均可离线预计算; 该方案需要两轮通信, 通信量为 1 280 bit. 当 Bob 和 Alice 为 1-to-n 关系时, 该方案的签名生成需要 1 次已知点点乘和 1 次未知点点乘, 其中已知点点乘可以离线预计算, 未知点点乘需要在线计算; 该方案需要两轮通信, 通信量为 1 280 bit. 值得说明的是, 该方案的密钥生成过程使用了基于 Paillier 算法或 OT 算法的 MtA 协议, 计算消耗较高. 尤其考虑到同时支持 SM2 两方协同解密算法[11], 在解密时需要两方协作计算$[d]C_1=\left[(d_1+d_2)^{-1}-1\right]C_1$, 计算过程较为复杂.

## 5 总 结

本文针对 SM2 签名算法的私钥安全问题, 提出了一个两方门限计算方案框架. 该框架描述了基于不同密钥拆分方法的 SM2 签名协作计算过程, 在签名计算的过程中, 参与方可以通过构造不同的签名随机数对该框架进行实例化, 从而得到不同的 SM2 签名两方门限计算方案. 本文的实例化, 包括了现有的 23 种两方门限计算方案, 也能够得到多种新的方案, 其中方案 [39] 性能最优.

**References:**

[1] Shamir A. How to share a secret. Communications of the ACM, 1979, 22(11): 612–613. [doi: 10.1145/359168.359176]

[2] Blakley GR. Safeguarding cryptographic keys. In: Proc. of the 1979 Int'l Workshop on Managing Requirements Knowledge. New York: IEEE, 1979. 313–313. [doi: 10.1109/MARK.1979.8817296]

[3] Desmedt Y. Society and group oriented cryptography: A new concept. In: Pomerance C, ed. Advances in Cryptology—CRYPTO 1987. Berlin, Heidelberg: Springer, 1988. 120–127. [doi: 10.1007/3-540-48184-2_8]

[4] Desmedt Y, Frankel Y. Threshold cryptosystems. In: Brassard G, ed. Advances in Cryptology—CRYPTO 1989. New York: Springer,

1980. 307–315. [doi: 10.1007/0-387-34805-0_28]

[5] State Cryptography Administration. SM2 elliptic curve cryptographic algorithm. 2010 (in Chinese). https://www.oscca.gov.cn/sca/xxgk/2010-12/17/content_1002386.shtml

[6] Desmedt Y, Frankel Y. Shared generation of authenticators and signatures. In: Feigenbaum J, ed. Advances in Cryptology—CRYPTO 1991. Berlin Heidelberg: Springer, 1992. 457–469. [doi: 10.1007/3-540-46766-1_37]

[7] Gennaro R, Jarecki S, Krawczyk H, Tabin T. Robust threshold DSS signatures. In: Proc. of the 1996 Int'l Conf. on the Theory and Applications of Cryptographic Techniques. Springer, 1996. 354–371. [doi: 10.1007/3-540-68339-9_31]

[8] Gennaro R, Goldfeder S, Narayanan A. Threshold-optimal DSA/ECDSA signatures and an application to bitcoin wallet security. In: Proc. of the 14th Int'l Conf. on Applied Cryptography and Network Security. Guildford: Springer, 2016. 156–174. [doi: 10.1007/978-3-319-39555-5_9]

[9] Gennaro R, Goldfeder S. Fast multiparty threshold ECDSA with fast trustless setup. In: Proc. of the 2018 ACM SIGSAC Conf. on Computer and Communications Security. Toronto: ACM, 2018. 1179–1194. [doi: 10.1145/3243734.3243859]

[10] Lindell Y, Nof A. Fast secure multiparty ECDSA with practical distributed key generation and applications to cryptocurrency custody. In: Proc. of the 2018 ACM SIGSAC Conf. on Computer and Communications Security. Toronto: ACM, 2018. 1837–1854. [doi: 10.1145/3243734.3243788]

[11] Lin JQ, Ma Y, Jing JW, Wang QX, Lei LG, Cai QW, Wang L. Signing and decryption methods and systems based on SM2 scheme suitable for cloud computation: CN, 104243456A. 2014-12-24 (in Chinese).

[12] Shang M, Ma Y, Lin JQ, Jing JW. A threshold scheme for SM2 elliptic curve cryptographic algorithm. Journal of Cryptologic Research, 2014, 1(2): 155–166 (in Chinese with English abstract). [doi: 10.13868/j.cnki.jcr.000015]

[13] Zhang YQ, Liu Q. Collaberative singing and decryption methods, devices and systems of SM2 scheme: CN, 107196763B. 2020-02-18 (in Chinese).

[14] Jie Y, Yu L, Li-Yun C, Wei N. A SM2 elliptic curve threshold signature scheme without a trusted center. KSII Trans. on Internet and Information Systems, 2016, 10(2): 897–913. [doi: 10.3837/tiis.2016.02.025]

[15] Aumasson JP, Hamelink A, Shlomovits O. A survey of ECDSA threshold signing. Cryptology ePrint Archive, 2020.

[16] Paillier P. Public-key cryptosystems based on composite degree residuosity classes. In: Proc. of the 1999 Int'l Conf. on the Theory and Applications of Cryptographic Techniques. Prague: Springer, 1999. 223–238. [doi: 10.1007/3-540-48910-X_16]

[17] Gilboa N. Two party RSA key generation. In: Proc. of the 19th Annual Int'l Cryptology Conf. Santa Barbara: Springer, 1999. 116–129. [doi: 10.1007/3-540-48405-1_8]

[18] Menezes A. Evaluation of security level of cryptography: The Elliptic Curve Discrete Logarithm Problem (ECDLP). Waterloo: University of Waterloo, 2001.

[19] Su YX, Tian HB. A two-party SM2 signing protocol and its application. Chinese Journal of Computers, 2020, 43(4): 701–710 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2020.00701]

[20] Ning HZ, Wang QG, Wei XY, Gong Z, Ma CS. SM2-based collaborative signature calculation method and device: CN, 112632630A. 2021-04-09 (in Chinese).

[21] Li ZH, Chen LM, Li L, Zhou YY, Li FG. Efficient two-party cooperative signature method based on SM2: CN, 112636918B. 2021-04-09 (in Chinese).

[22] Zhao GL, Liao ZY, Liu XP, He J, Peng JH, Liu WZ, Li X, Wei ZG. SM2-based split key signature method and system: CN, 110943826B (in Chinese). 2022-03-31.

[23] Jia WY, Zhang F, Huang NN, Li HL. SM2-based two-party cooperative signature and decryption methods: CN, 111314089A. 2020-06-19 (in Chinese).

[24] He DB, Feng Q, Luo M, Li L, Huang XY. SM2-based two-party cooperative signature method and media for lightweight clients: CN, 111010285B. 2020-04-14 (in Chinese).

[25] Jing JW, Wang PJ, Wang YW, Wang P, Lei LG, Liu LM, Sun SW, Kou CJ. SM2-based cooperative signature and decryption methods for protecting user privacy: CN, 114186251A. 2022-03-15 (in Chinese).

[26] Pan JC, Wang ZH, Liang ZQ. The method, device ande system of SM2 cooperative signature and decryption: CN, 109672539B. 2019-04-23 (in Chinese).

[27] Zhao GL, Liu XP, Liao ZY, Peng JH, Liu CH. A cooperative signature method and system based on SM2: CN, 108989047B. 2018-12-11 (in Chinese).

[28] Han LM, Wang QF. Lightweight cooperative signature method and device based on SM2 algorithm: CN, 110535636B. 2019-12-03 (in Chinese).

[29] Wang H, Zhang Y, Zheng JD. A method and system of two party cooperative signature and decryption based on SM2: CN, 113849831A.

2021-12-28 (in Chinese).
[30] Ma CS, Gong Z, Liu ZJ, Jiang M. SM2 collaborative signature method: CN, 110278088A. 2019-09-24 (in Chinese).
[31] Zhang LT, Wang XF, Pan WL. SM2-based two-party signature method and system: CN, 109450640B. 2019-03-08 (in Chinese).
[32] Yao YF, Fang WM, Li HQ, Qin PC. A method and system of SM2 multi-party cooperative signature based on key factor: CN, 112187469A. 2021-01-05 (in Chinese).
[33] Lin JQ, Ma Y, Wu XY, Chen TY, Jing JW. The method applied to CS architecture for SM2 signature: CN, 108737103B. 2018-11-02 (in Chinese).
[34] Cheng ZH. The method and device based of SM2 digital signature: CN, 107124274B. 2017-09-01 (in Chinese).
[35] Zhang ZF, Tang GF. The actively secure generation method for SM2 digital signature: CN, 111447065B. 2020-07-24 (in Chinese).
[36] Long YH. The segmentation generation method and system for SM2 digital signature: CN, 106603246B. 2017-04-26 (in Chinese).
[37] Liu T, Wang ZB. The method, device and storage medium for SM2 2-party cooperative signature: CN, 109245903B. 2019-01-18 (in Chinese).
[38] Liu T, Wang ZB. The method, device and storage medium for SM2 cooperative signature: CN, 109309569B. 2019-02-05 (in Chinese).
[39] Yuan F, Zhang LY, Feng WD, Zhang XP. The method based on addition key segmentation for SM2 signature: CN, 107623570B. 2018-01-23 (in Chinese).
[40] Hankerson D, Vanstone S, Menezes A. Guide to Elliptic Curve Cryptography. New York: Springer, 2004. 95–113.

**附中文参考文献**:

[5] 国家密码管理局. 国家密码管理局关于发布《SM2椭圆曲线公钥密码算法》公告. 2010. https://www.oscca.gov.cn/sca/xxgk/2010-12/17/content_1002386.shtml
[11] 林璟锵, 马原, 荆继武, 王琼霄, 雷灵光, 蔡权伟, 王雷. 适用于云计算的基于SM2算法的签名及解密方法和系统: 中国, 104243456A. 2014-12-24.
[12] 尚铭, 马原, 林璟锵, 荆继武. SM2椭圆曲线门限密码算法. 密码学报, 2014, 1(2): 155–166. [doi: 10.13868/j.cnki.jcr.000015]
[13] 张永强, 刘镭. SM2算法协同签名及解密方法、装置与系统: 中国, 107196763B. 2020-02-18.
[19] 苏吟雪, 田海博. 基于SM2的双方共同签名协议及其应用. 计算机学报, 2020, 43(4): 701–710. [doi: 10.11897/SP.J.1016.2020.00701]
[20] 宁红宙, 王启刚, 危学艳, 龚征, 马昌社. 一种基于SM2的协同签名计算方法及装置: 中国, 112632630A. 2021-04-09.
[21] 李正宏, 陈黎明, 李磊, 周雨阳, 李发根. 一种高效的基于SM2两方协同签名方法: 中国, 112636918B. 2021-04-09.
[22] 赵国磊, 廖正赟, 刘熙胖, 何骏, 彭金辉, 刘武忠, 李鑫, 卫志刚. 一种基于SM2算法的拆分密钥签名方法与系统: 中国, 110943826B. 2022-03-31.
[23] 贾文义, 张凡, 黄念念, 李鸿利. 一种基于SM2的两方协同签名方法及解密方法: 中国, 111314089A. 2020-06-19.
[24] 何德彪, 冯琦, 罗敏, 李莉, 黄欣沂. 一种适用于轻量级客户端的SM2两方协同签名方法及介质: 中国, 111010285B. 2020-04-14.
[25] 荆继武, 王平建, 王跃武, 王鹏, 雷灵光, 刘丽敏, 孙思维, 寇春静. 一种保护用户隐私的SM2密码算法协同签名、解密方法: 中国, 114186251A. 2022-03-15.
[26] 潘金昌, 王志辉, 梁珍权. SM2算法协同签名及解密方法、装置及系统: 中国, 109672539B. 2019-04-23.
[27] 赵国磊, 刘熙胖, 廖正赟, 彭金辉, 刘长河. 一种基于SM2算法的通信双方协同签名方法与系统: 中国, 108989047B. 2018-12-11.
[28] 韩留明, 王庆芝. 一种轻量级的基于SM2算法的协同签名方法与装置: 中国, 110535636B. 2019-12-03.
[29] 王慧, 张渊, 郑江东. 一种基于SM2算法的两方协同签名和解密方法及系统: 中国, 113849831A. 2021-12-28.
[30] 马昌社, 龚征, 刘志杰, 姜枚. 一种SM2协同签名方法: 中国, 110278088A. 2019-09-24.
[31] 张立廷, 王现方, 潘文伦. 基于SM2的两方签名方法及系统: 中国, 109450640B. 2019-03-08.
[32] 姚有方, 方伟明, 李红乾, 秦盼春. 一种基于密钥因子的SM2多方协同数字签名方法和系统: 中国, 112187469A. 2021-01-05.
[33] 林璟锵, 马原, 吴鑫莹, 陈天宇, 荆继武. 一种应用于CS架构的SM2算法签名方法: 中国, 108737103B. 2018-11-02.
[34] 程朝辉. 基于SM2的数字签名方法和装置: 中国, 107124274B. 2017-09-01.
[35] 张振峰, 唐国锋. 一种主动安全的SM2数字签名两方生成方法: 中国, 111447065B. 2020-07-24.
[36] 龙毅宏. 一种SM2数字签名分割生成方法及系统: 中国, 106603246B. 2017-04-26.
[37] 刘婷, 王宗斌. 双方协同生成SM2算法的签名方法、装置及存储介质: 中国, 109245903B. 2019-01-18.
[38] 刘婷, 王宗斌. 基于SM2算法的协同签名的方法、装置及存储介质: 中国, 109309569B. 2019-02-05.
[39] 袁峰, 张立圆, 封维端, 张祥攀. 一种基于加法密钥分割的SM2签名方法: 中国, 107623570B. 2018-01-23.

## 附录 A

表 A1　基于乘法密钥拆分的 2 随机数框架的实例化

| $w_1$ | $w_2$ | $w_1'$ | $w_2'$ | $k=w_2+d_2^{-1}w_1$ | $k=d_1w_2'+w_1'$ | $s=(1+d)^{-1}(k+r)-r$ | 序号 |
|---|---|---|---|---|---|---|---|
| | $k_2$ | | $(1+d)d_2k_2$ | $k_2+d_2^{-1}k_1$ | $(1+d)(d_1d_2k_2+d_1k_1)$ | $d_1d_2k_2+d_1k_1+(d_1d_2-1)r$ | 1 |
| | $d_2k_2$ | | $(1+d)d_2^2k_2$ | $d_2k_2+d_2^{-1}k_1$ | $(1+d)\left(d_1d_2^2k_2+d_1k_1\right)$ | $d_1d_2^2+d_1k_1+(d_1d_2-1)r$ | 2 |
| | $d_2^{-1}k_2$ | | $(1+d)k_2$ | $d_2^{-1}(k_2+k_1)$ | $(1+d)(d_1k_2+d_1k_1)$ | $d_1(k_2+k_1)+(d_1d_2-1)r$ | 3 |
| $k_1$ | $1+d_2k_2$ | $(1+d)d_1k_1$ | $(1+d)\left(d_2+d_2^2k_2\right)$ | $1+d_2k_2+d_2^{-1}k_1$ | $(1+d)\left(d_1d_2+d_1k_1+d_1d_2^2k_2\right)$ | $d_1d_2+d_1d_2^2k_2+d_1k_1+(d_1d_2-1)r$ | 4 |
| | $1+d_2^{-1}k_2$ | | $(1+d)(d_2+k_2)$ | $1+d_2^{-1}(k_2+k_1)$ | $(1+d)\left(d_1d_2+d_1k_2+d_1k_1\right)$ | $d_1d_2+d_1(k_2+k_1)+(d_1d_2-1)r$ | 5 |
| | $d_2+k_2$ | | $(1+d)\left(d_2^2+d_2k_2\right)$ | $d_2+k_2+d_2^{-1}k_1$ | $(1+d)\left(d_1d_2^2+d_1k_1+d_1d_2k_2\right)$ | $d_1d_2^{-1}+d_1d_2k_2+d_1k_1+(d_1d_2-1)r$ | 6 |
| | $d_2^{-1}+k_2$ | | $(1+d)(1+d_2k_2)$ | $d_2^{-1}+k_2+d_2^{-1}k_1$ | $(1+d)\left(d_1+d_1d_2k_2+d_1k_1\right)$ | $d_1+d_1d_2k_2+d_1k_1+(d_1d_2-1)r$ | 7 |
| | $k_2$ | | $(1+d)d_2k_2$ | $k_2+d_1d_2^{-1}k_1$ | $(1+d)\left(d_1d_2k_2+d_1^2k_1\right)$ | $d_1d_2k_2+d_1^2k_1+(d_1d_2-1)r$ | 8 |
| | $d_2k_2$ | | $(1+d)d_2^2k_2$ | $d_2k_2+d_1d_2^{-1}k_1$ | $(1+d)\left(d_1d_2^2k_2+d_1^2k_1\right)$ | $d_1d_2^2k_2+d_1^2k_1+(d_1d_2-1)r$ | 9 |
| | $d_2^{-1}k_2$ | | $(1+d)k_2$ | $d_2^{-1}(k_2+d_1k_1)$ | $(1+d)\left(d_1k_2+d_1^2k_1\right)$ | $d_1(k_2+d_1k_1)+(d_1d_2-1)r$ | 10 |
| $d_1k_1$ | $1+d_2k_2$ | $(1+d)d_1^2k_1$ | $(1+d)\left(d_2+d_2^2k_2\right)$ | $1+d_2k_2+d_1d_2^{-1}k_1$ | $(1+d)\left(d_1d_2+d_1^2k_1+d_1d_2^2k_2\right)$ | $d_1d_2+d_1d_2^2k_2+d_1^2k_1+(d_1d_2-1)r$ | 11 |
| | $1+d_2^{-1}k_2$ | | $(1+d)(d_2+k_2)$ | $1+d_2^{-1}(k_2+d_1k_1)$ | $(1+d)\left(d_1d_2+d_1k_2+d_1^2k_1\right)$ | $d_1d_2+d_1(k_2+d_1k_1)+(d_1d_2-1)r$ | 12 |
| | $d_2+k_2$ | | $(1+d)\left(d_2^2+d_2k_2\right)$ | $d_2+k_2+d_1d_2^{-1}k_1$ | $(1+d)\left(d_1d_2^2+d_1^2k_1+d_1d_2k_2\right)$ | $d_1d_2^2+d_1d_2k_2+d_1^2k_1+(d_1d_2-1)r$ | 13 |
| | $d_2^{-1}+k_2$ | | $(1+d)(1+d_2k_2)$ | $d_2^{-1}+k_2+d_1d_2^{-1}k_1$ | $(1+d)\left(d_1+d_1d_2k_2+d_1^2k_1\right)$ | $d_1+d_1d_2k_2+d_1^2k_1+(d_1d_2-1)r$ | 14 |
| | $k_2$ | | $(1+d)d_2k_2$ | $k_2+d_1^{-1}d_2^{-1}k_1$ | $(1+d)(d_1d_2k_2+k_1)$ | $d_1d_2k_2+k_1+(d_1d_2-1)r$ | 15 |
| | $d_2k_2$ | | $(1+d)d_2^2k_2$ | $d_2k_2+d_1^{-1}d_2^{-1}k_1$ | $(1+d)\left(d_1d_2^2k_2+k_1\right)$ | $d_1d_2^2k_2+k_1+(d_1d_2-1)r$ | 16 |
| | $d_2^{-1}k_2$ | | $(1+d)k_2$ | $d_2^{-1}\left(k_2+d_1^{-1}k_1\right)$ | $(1+d)(d_1k_2+k_1)$ | $d_1\left(k_2+d_1^{-1}k_1\right)+(d_1d_2-1)r$ | 17 |
| $d_1^{-1}k_1$ | $1+d_2k_2$ | $(1+d)k_1$ | $(1+d)\left(d_2+d_2^2k_2\right)$ | $1+d_2k_2+d_1^{-1}d_2^{-1}k_1$ | $(1+d)\left(d_1d_2+k_1+d_1d_2^2k_2\right)$ | $d_1d_2+d_1d_2^2k_2+k_1+(d_1d_2-1)r$ | 18 |
| | $1+d_2^{-1}k_2$ | | $(1+d)(d_2+k_2)$ | $1+d_2^{-1}\left(k_2+d_1^{-1}k_1\right)$ | $(1+d)\left(d_1d_2+d_1k_2+k_1\right)$ | $d_1\left(k_2+d_1^{-1}k_1\right)+d_1d_2+(d_1d_2-1)r$ | 19 |
| | $d_2+k_2$ | | $(1+d)\left(d_2^2+d_2k_2\right)$ | $d_2+k_2+d_1^{-1}d_2^{-1}k_1$ | $(1+d)\left(d_1d_2^2+d_1d_2k_2+k_1\right)$ | $d_1d_2^2+d_1d_2k_2+k_1+(d_1d_2-1)r$ | 20 |

表 A1　基于乘法密钥拆分的 2 随机数框架的实例化 (续)

| $w_1$ | $w_2$ | $w_1'$ | $w_2'$ | $k=w_2+d_2^{-1}w_1$ | $k=d_1w_2'+w_1'$ | $s=(1+d)^{-1}(k+r)-r$ | 序号 |
|---|---|---|---|---|---|---|---|
| | $d_2^{-1}+k_2$ | | $(1+d)(1+d_2k_2)$ | $d_2^{-1}+k_2+d_1^{-1}d_2^{-1}k_1$ | $(1+d)\left(d_1+d_1d_2k_2+k_1\right)$ | $d_1+d_1d_2k_2+k_1+(d_1d_2-1)r$ | 21 |
| | $k_2$ | | $(1+d)d_2k_2$ | $k_2+d_2^{-1}+d_1d_2^{-1}k_1$ | $(1+d)\left(d_1d_2k_2+d_1+d_1^2k_1\right)$ | $d_1d_2k_2+d_1+d_1^2k_1+(d_1d_2-1)r$ | 22 |
| | $d_2k_2$ | | $(1+d)d_2^2k_2$ | $d_2k_2+d_2^{-1}+d_1d_2^{-1}k_1$ | $(1+d)\left(d_1d_2^2k_2+d_1+d_1^2k_1\right)$ | $d_1d_2^2k_2+d_1+d_1^2k_1+(d_1d_2-1)r$ | 23 |
| | $d_2^{-1}k_2$ | | $(1+d)k_2$ | $d_2^{-1}k_2+d_2^{-1}+d_1d_2^{-1}k_1$ | $(1+d)\left(d_1k_2+d_1+d_1^2k_1\right)$ | $d_1k_2+d_1+d_1^2k_1+(d_1d_2-1)r$ | 24 |
| $1+d_1k_1$ | $1+d_2k_2$ | $(1+d)\left(d_1+d_1^2k_1\right)$ | $(1+d)\left(d_2+d_2^2k_2\right)$ | $1+d_2k_2+d_2^{-1}+d_1d_2^{-1}k_1$ | $(1+d)\left(d_1d_2+d_1d_2^2k_2+d_1+d_1^2k_1\right)$ | $d_1d_2+d_1d_2^2k_2+d_1+d_1^2k_1+(d_1d_2-1)r$ | 25 |
| | $1+d_2^{-1}k_2$ | | $(1+d)(d_2+k_2)$ | $1+d_2^{-1}k_2+d_2^{-1}+d_1d_2^{-1}k_1$ | $(1+d)\left(d_1d_2+d_1k_2+d_1+d_1^2k_1\right)$ | $d_1d_2+d_1k_2+d_1+d_1^2k_1+(d_1d_2-1)r$ | 26 |
| | $d_2+k_2$ | | $(1+d)\left(d_2^2+d_2k_2\right)$ | $d_2+k_2+d_2^{-1}+d_1d_2^{-1}k_1$ | $(1+d)\left(d_1d_2^2+d_1d_2k_2+d_1+d_1^2k_1\right)$ | $d_1d_2^2+d_1d_2k_2+d_1+d_1^2k_1+(d_1d_2-1)r$ | 27 |
| | $d_2^{-1}+k_2$ | | $(1+d)(1+d_2k_2)$ | $d_2^{-1}+k_2+d_2^{-1}+d_1d_2^{-1}k_1$ | $(1+d)\left(d_1+d_1d_2k_2+d_1+d_1^2k_1\right)$ | $d_1+d_1d_2k_2+d_1+d_1^2k_1+(d_1d_2-1)r$ | 28 |
| | $k_2$ | | $(1+d)d_2k_2$ | $k_2+d_2^{-1}+d_1^{-1}d_2^{-1}k_1$ | $(1+d)\left(d_1d_2k_2+d_1+k_1\right)$ | $d_1d_2k_2+d_1+k_1+(d_1d_2-1)r$ | 29 |
| | $d_2k_2$ | | $(1+d)d_2^2k_2$ | $d_2k_2+d_2^{-1}+d_1^{-1}d_2^{-1}k_1$ | $(1+d)\left(d_1d_2^2k_2+d_1+k_1\right)$ | $d_1d_2^2k_2+d_1+k_1+(d_1d_2-1)r$ | 30 |
| | $d_2^{-1}k_2$ | | $(1+d)k_2$ | $d_2^{-1}k_2+d_2^{-1}+d_1^{-1}d_2^{-1}k_1$ | $(1+d)\left(d_1k_2+d_1+k_1\right)$ | $d_1k_2+d_1+k_1+(d_1d_2-1)r$ | 31 |
| $1+d_1^{-1}k_1$ | $1+d_2k_2$ | $(1+d)(d_1+k_1)$ | $(1+d)\left(d_2+d_2^2k_2\right)$ | $1+d_2k_2+d_2^{-1}+d_1^{-1}d_2^{-1}k_1$ | $(1+d)\left(d_1d_2+d_1+d_1d_2^2k_2+k_1\right)$ | $d_1d_2+d_1d_2^2k_2+d_1+k_1+(d_1d_2-1)r$ | 32 |
| | $1+d_2^{-1}k_2$ | | $(1+d)(d_2+k_2)$ | $1+d_2^{-1}k_2+d_2^{-1}+d_1^{-1}d_2^{-1}k_1$ | $(1+d)\left(d_1d_2+d_1k_2+d_1+k_1\right)$ | $d_1d_2+d_1k_2+d_1+k_1+(d_1d_2-1)r$ | 33 |
| | $d_2+k_2$ | | $(1+d)\left(d_2^2+d_2k_2\right)$ | $d_2+k_2+d_2^{-1}+d_1^{-1}d_2^{-1}k_1$ | $(1+d)\left(d_1d_2^2+d_1+d_1d_2k_2+k_1\right)$ | $d_1d_2^2+d_1d_2k_2+d_1+k_1+(d_1d_2-1)r$ | 34 |
| | $d_2^{-1}+k_2$ | | $(1+d)(1+d_2k_2)$ | $d_2^{-1}+k_2+d_2^{-1}+d_1^{-1}d_2^{-1}k_1$ | $(1+d)\left(d_1+d_1d_2k_2+d_1+k_1\right)$ | $d_1+d_1d_2k_2+d_1+k_1+(d_1d_2-1)r$ | 35 |
| | $k_2$ | | $(1+d)d_2k_2$ | $k_2+d_1d_2^{-1}+d_2^{-1}k_1$ | $(1+d)\left(d_1d_2k_2+d_1^2+d_1k_1\right)$ | $d_1d_2k_2+d_1^2+d_1k_1+(d_1d_2-1)r$ | 36 |
| | $d_2k_2$ | | $(1+d)d_2^2k_2$ | $d_2k_2+d_1d_2^{-1}+d_2^{-1}k_1$ | $(1+d)\left(d_1d_2^2k_2+d_1^2+d_1k_1\right)$ | $d_1d_2^2k_2+d_1^2+d_1k_1+(d_1d_2-1)r$ | 37 |
| $d_1+k_1$ | $d_2^{-1}k_2$ | $(1+d)\left(d_1^2+d_1k_1\right)$ | $(1+d)k_2$ | $d_2^{-1}k_2+d_1d_2^{-1}+d_2^{-1}k_1$ | $(1+d)\left(d_1k_2+d_1^2+d_1k_1\right)$ | $d_1k_2+d_1^2+d_1k_1+(d_1d_2-1)r$ | 38 |
| | $1+d_2k_2$ | | $(1+d)\left(d_2+d_2^2k_2\right)$ | $1+d_2k_2+d_1d_2^{-1}+d_2^{-1}k_1$ | $(1+d)\left(d_1d_2+d_1d_2^2k_2+d_1^2+d_1k_1\right)$ | $d_1d_2+d_1d_2^2k_2+d_1^2+d_1k_1+(d_1d_2-1)r$ | 39 |
| | $1+d_2^{-1}k_2$ | | $(1+d)(d_2+k_2)$ | $1+d_2^{-1}k_2+d_1d_2^{-1}+d_2^{-1}k_1$ | $(1+d)\left(d_1d_2+d_1k_2+d_1^2+d_1k_1\right)$ | $d_1d_2+d_1k_2+d_1^2+d_1k_1+(d_1d_2-1)r$ | 40 |

表 A1　基于乘法密钥拆分的 2 随机数框架的实例化 (续)

| $w_1$ | $w_2$ | $w_1'$ | $w_2'$ | $k=w_2+d_2^{-1}w_1$ | $k=d_1w_2'+w_1'$ | $s=(1+d)^{-1}(k+r)-r$ | 序号 |
|---|---|---|---|---|---|---|---|
| | $d_2+k_2$ | | $(1+d)\left(d_2^2+d_2k_2\right)$ | $d_2+k_2+d_1d_2^{-1}+d_2^{-1}k_1$ | $(1+d)\left(d_1d_2^2+d_1d_2k_2+d_1^2+d_1k_1\right)$ | $d_1d_2^2+d_1d_2k_2+d_1^2+d_1k_1+(d_1d_2-1)r$ | 41 |
| | $d_2^{-1}+k_2$ | | $(1+d)(1+d_2k_2)$ | $d_2^{-1}+k_2+d_1d_2^{-1}+d_2^{-1}k_1$ | $(1+d)\left(d_1+d_1d_2k_2+d_1^2+d_1k_1\right)$ | $d_1+d_1d_2k_2+d_1^2+d_1k_1+(d_1d_2-1)r$ | 42 |
| | $k_2$ | | $(1+d)d_2k_2$ | $k_2+d_1^{-1}d_2^{-1}+d_2^{-1}k_1$ | $(1+d)\left(d_1d_2k_2+1+d_1k_1\right)$ | $d_1d_2k_2+1+d_1k_1+(d_1d_2-1)r$ | 43 |
| | $d_2k_2$. | | $(1+d)d_2^2k_2$. | $d_2k_2+d_1^{-1}d_2^{-1}+d_2^{-1}k_1$ | $(1+d)\left(d_1d_2^2k_2+1+d_1k_1\right)$ | $d_1d_2^2k_2+1+d_1k_1+(d_1d_2-1)r$ | 44 |
| | $d_2^{-1}k_2$ | | $(1+d)k_2$ | $d_2^{-1}k_2+d_1^{-1}d_2^{-1}+d_2^{-1}k_1$ | $(1+d)(d_1k_2+1+d_1k_1)$ | $d_1k_2+1+d_1k_1+(d_1d_2-1)r$ | 45 |
| $d_1^{-1}+k_1$ | $1+d_2k_2$ | $(1+d)(1+d_1k_1)$ | $(1+d)\left(d_2+d_2^2k_2\right)$ | $1+d_2^{-1}k_2+d_1^{-1}d_2^{-1}+d_2^{-1}k_1$ | $(1+d)\left(d_1d_2+d_1k_2+1+d_1k_2\right)$ | $d_1d_2+d_1k_2+1+d_1k_1+(d_1d_2-1)r$ | 46 |
| | $1+d_2^{-1}k_2$ | | $(1+d)(d_2+k_2)$ | $1+d_2^{-1}k_2+d_1^{-1}d_2^{-1}+d_2^{-1}k_1$ | $(1+d)\left(d_1d_1+d_1k_2+1+d_1k_1\right)$ | $d_1d_2+d_1k_2+1+d_1k_1+(d_1d_2-1)r$ | 47 |
| | $d_2+k_2$ | | $(1+d)\left(d_2^2+d_2k_2\right)$ | $d_2+k_2+d_1^{-1}d_2^{-1}+d_2^{-1}k_1$ | $(1+d)\left(d_1d_2^2+d_1d_2k_2+1+d_1k_1\right)$ | $d_1d_2^2+d_1d_2k_2+1+d_1k_1+(d_1d_2-1)r$ | 48 |
| | $d_2^{-1}+k_2$ | | $(1+d)(1+d_2k_2)$ | $d_2^{-1}+k_2+d_1^{-1}d_2^{-1}+d_2^{-1}k_1$ | $(1+d)\left(d_1+d_1d_2k_2+1+d_1k_1\right)$ | $d_1+d_1d_2k_2+1+d_1k_1+(d_1d_2-1)r$ | 49 |

表 A2　基于加法密钥拆分的 2 随机数框架的实例化

| $w_1$ | $w_2$ | $k=(d_1+d_2)^{-1}\ (d_1w_1+d_2w_2)$ | $s=(1+d)^{-1}(k+r)-r$ | 序号 |
|---|---|---|---|---|
| $k_1$ | $k_2$ | $(d_1+d_2)^{-1}\ (d_1k_1+d_2k_2)$ | $d_1k_1+d_2k_2+(d_1+d_2-1)r$ | 1 |
| | $d_2^{-1}k_2$ | $(d_1+d_2)^{-1}\ (d_1k_1+k_2)$ | $d_1k_1+k_2+(d_1+d_2-1)r$ | 2 |
| | $d_2^{-2}k_2$ | $(d_1+d_2)^{-1}\ (d_1k_1+d_2^{-1}k_2)$. | $d_1k_1+d_2^{-1}k_2+(d_1+d_2-1)r$ | 3 |
| | $1+d_2^{-1}k_2$ | $(d_1+d_2)^{-1}\ (d_1k_1+d_2+k_2)$ | $d_1k_1+d_2+k_2+(d_1+d_2-1)r$ | 4 |
| | $1+d_2^{-2}k_2$ | $(d_1+d_2)^{-1}\ (d_1k_1+d_2+d_2^{-1}k_2)$ | $d_1k_1+d_2+d_2^{-1}k_2+(d_1+d_2-1)r$ | 5 |
| | $d_2^{-1}+k_2$ | $(d_1+d_2)^{-1}\ (d_1k_1+1+d_2k_2)$ | $d_1k_1+1+d_2k_2+(d_1+d_2-1)r$ | 6 |
| | $d_2^{-2}+k_2$ | $(d_1+d_2)^{-1}\left(d_1k_1+d_2^{-1}+d_2k_2\right)$ | $d_1k_1+d_2^{-1}+d_2k_2+(d_1+d_2-1)r$ | 7 |
| $d_1^{-1}k_1$ | $k_2$ | $(d_1+d_2)^{-1}\ (k_1+d_2k_2)$ | $k_1+d_2k_2+(d_1+d_2-1)r$ | 8 |
| | $d_2^{-1}k_2$ | $(d_1+d_2)^{-1}\ (k_1+k_2)$ | $k_1+k_2+(d_1+d_2-1)r$ | 9 |
| | $d_2^{-2}k_2$ | $(d_1+d_2)^{-1}\ (k_1+d_2^{-1}k_2)$ | $k_1+d_2^{-1}k_2+(d_1+d_2-1)r$ | 10 |
| | $1+d_2^{-1}k_2$ | $(d_1+d_2)^{-1}\ (k_1+d_2+k_2)$ | $k_1+d_2+k_2+(d_1+d_2-1)r$ | 11 |
| | $1+d_2^{-2}k_2$ | $(d_1+d_2)^{-1}\ (k_1+d_2+d_2^{-1}k_2)$ | $k_1+d_2+d_2^{-1}k_2+(d_1+d_2-1)r$ | 12 |
| | $d_2^{-1}+k_2$ | $(d_1+d_2)^{-1}\ (k_1+1+d_2k_2)$ | $k_1+1+d_2k_2+(d_1+d_2-1)r$ | 13 |
| | $d_2^{-2}+k_2$ | $(d_1+d_2)^{-1}\left(k_1+d_2^{-1}+d_2k_2\right)$ | $k_1+d_2^{-1}+d_2k_2+(d_1+d_2-1)r$ | 14 |
| $d_1^{-2}k_1$ | $k_2$ | $(d_1+d_2)^{-1}\left(d_1^{-1}k_1+d_2k_2\right)$ | $d_1^{-1}k_1+d_2k_2+(d_1+d_2-1)r$ | 15 |
| | $d_2^{-1}k_2$ | $(d_1+d_2)^{-1}\left(d_1^{-1}k_1+k_2\right)$ | $d_1^{-1}k_1+k_2+(d_1+d_2-1)r$ | 16 |
| | $d_2^{-2}k_2$ | $(d_1+d_2)^{-1}\left(d_1^{-1}k_1+d_2^{-1}k_2\right)$ | $d_1^{-1}k_1+d_2^{-1}k_2+(d_1+d_2-1)r$ | 17 |
| | $1+d_2^{-1}k_2$ | $(d_1+d_2)^{-1}\left(d_1^{-1}k_1+d_2+k_2\right)$ | $d_1^{-1}k_1+d_2+k_2+(d_1+d_2-1)r$ | 18 |
| | $1+d_2^{-2}k_2$ | $(d_1+d_2)^{-1}\left(d_1^{-1}k_1+d_2+d_2^{-1}k_2\right)$ | $d_1^{-1}k_1+d_2+d_2^{-1}k_2+(d_1+d_2-1)r$ | 19 |
| | $d_2^{-1}+k_2$ | $(d_1+d_2)^{-1}\left(d_1^{-1}k_1+1+d_2k_2\right)$ | $d_1^{-1}k_1+1+d_2k_2+(d_1+d_2-1)r$ | 20 |
| | $d_2^{-2}+k_2$ | $(d_1+d_2)^{-1}\left(d_1^{-1}+d_2^{-1}+d_2k_2\right)$ | $d_1^{-1}+d_2^{-1}+d_2k_2+(d_1+d_2-1)r$ | 21 |
| $1+d_1^{-1}k_1$ | $k_2$ | $(d_1+d_2)^{-1}(d_1+k_1+d_2k_2)$ | $d_1+k_1+d_2k_2+(d_1+d_2-1)r$ | 22 |
| | $d_2^{-1}k_2$ | $(d_1+d_2)^{-1}(d_1+k_1+k_2)$ | $d_1+k_1+k_2+(d_1+d_2-1)r$ | 23 |
| | $d_2^{-2}k_2$ | $(d_1+d_2)^{-1}(d_1+k_1+d_2^{-1}k_2)$ | $d_1+k_1+d_2^{-1}k_2+(d_1+d_2-1)r$ | 24 |
| | $1+d_2^{-1}k_2$ | $(d_1+d_2)^{-1}\ (d_1+k_1+d_2+k_2)$ | $d_1+k_1+d_2+k_2+(d_1+d_2-1)r$ | 25 |
| | $1+d_2^{-2}k_2$ | $(d_1+d_2)^{-1}(d_1+k_1+d_2+d_2^{-1}k_2)$ | $d_1+k_1+d_2+d_2^{-1}k_2+(d_1+d_2-1)r$ | 26 |
| | $d_2^{-1}+k_2$ | $(d_1+d_2)^{-1}\ (d_1+k_1+1+d_2k_2)$ | $d_1+k_1+1+d_2k_2+(d_1+d_2-1)r$ | 27 |
| | $d_2^{-2}+k_2$ | $(d_1+d_2)^{-1}\left(d_1+k_1+d_2^{-1}+d_2k_2\right)$ | $d_1+k_1+d_2^{-1}+d_2k_2+(d_1+d_2-1)r$ | 28 |
| $1+d_1^{-2}k_1$ | $k_2$ | $(d_1+d_2)^{-1}\left(d_1+d_1^{-1}k_1+d_2k_2\right)$ | $d_1+d_1^{-1}k_1+d_2k_2+(d_1+d_2-1)r$ | 29 |
| | $d_2^{-1}k_2$ | $(d_1+d_2)^{-1}\left(d_1+d_1^{-1}k_1+k_2\right)$ | $d_1+d_1^{-1}k_1+k_2+(d_1+d_2-1)r$ | 30 |
| | $d_2^{-2}k_2$ | $(d_1+d_2)^{-1}\left(d_1+d_1^{-1}k_1+d_2^{-1}k_2\right)$ | $d_1+d_1^{-1}k_1+d_2^{-1}k_2+(d_1+d_2-1)r$ | 31 |
| | $1+d_2^{-1}k_2$ | $(d_1+d_2)^{-1}\left(d_1+d_1^{-1}k_1+d_2+k_2\right)$ | $d_1+d_1^{-1}k_1+d_2+k_2+(d_1+d_2-1)r$ | 32 |

表 A2 基于加法密钥拆分的 2 随机数框架的实例化 (续)

| $w_1$ | $w_2$ | $k=(d_1+d_2)^{-1}\ (d_1w_1+d_2w_2)$ | $s=(1+d)^{-1}(k+r)-r$ | 序号 |
|---|---|---|---|---|
| | $1+d_2^{-2}k_2$ | $(d_1+d_2)^{-1}\left(d_1+d_1^{-1}k_1+d_2+d_2^{-1}k_2\right)$ | $d_1+d_1^{-1}k_1+d_2+d_2^{-1}k_2+(d_1+d_2-1)r$ | 33 |
| | $d_2^{-1}+k_2$ | $(d_1+d_2)^{-1}\left(d_1+d_1^{-1}k_1+1+d_2k_2\right)$ | $d_1+d_1^{-1}k_1+1+d_2k_2+(d_1+d_2-1)r$ | 34 |
| | $d_2^{-2}+k_2$ | $(d_1+d_2)^{-1}\left(d_1+d_1^{-1}+d_2^{-1}+d_2k_2\right)$ | $d_1+d_1^{-1}+d_2^{-1}+d_2k_2+(d_1+d_2-1)r$ | 35 |
| | $k_2$ | $(d_1+d_2)^{-1}\ (1+d_1k_1+d_2k_2)$ | $1+d_1k_1+d_2k_2+(d_1+d_2-1)r$ | 36 |
| | $d_2^{-1}k_2$ | $(d_1+d_2)^{-1}\ (1+d_1k_1+k_2)$ | $1+d_1k_1+k_2+(d_1+d_2-1)r$ | 37 |
| | $d_2^{-2}k_2$ | $(d_1+d_2)^{-1}\left(1+d_1k_1+d_2^{-1}k_2\right)$ | $1+d_1k_1+d_2^{-1}k_2+(d_1+d_2-1)r$ | 38 |
| $d_1^{-1}+k_1$ | $1+d_2^{-1}k_2$ | $(d_1+d_2)^{-1}\ (1+d_1k_1+d_2+k_2)$ | $1+d_1k_1+d_2+k_2+(d_1+d_2-1)r$ | 39 |
| | $1+d_2^{-2}k_2$ | $(d_1+d_2)^{-1}\left(1+d_1k_1+d_2+d_2^{-1}k_2\right)$ | $1+d_1k_1+d_2+d_2^{-1}k_2+(d_1+d_2-1)r$ | 40 |
| | $d_2^{-1}+k_2$ | $(d_1+d_2)^{-1}\ (1+d_1k_1+1+d_2k_2)$ | $1+d_1k_1+1+d_2k_2+(d_1+d_2-1)r$ | 41 |
| | $d_2^{-2}+k_2$ | $(d_1+d_2)^{-1}\left(1+d_1k_1+d_2^{-1}+d_2k_2\right)$ | $1+d_1k_1+d_2^{-1}+d_2k_2+(d_1+d_2-1)r$ | 42 |
| | $k_2$ | $(d_1+d_2)^{-1}\left(d_1^{-1}+d_1k_1+d_2k_2\right)$ | $d_1^{-1}+d_1k_1+d_2k_2+(d_1+d_2-1)r$ | 43 |
| | $d_2^{-1}k_2$ | $(d_1+d_2)^{-1}\left(d_1^{-1}+d_1k_1+k_2\right)$ | $d_1^{-1}+d_1k_1+k_2+(d_1+d_2-1)r$ | 44 |
| | $d_2^{-2}k_2$ | $(d_1+d_2)^{-1}\left(d_1^{-1}+d_1k_1+d_2^{-1}k_2\right)$ | $d_1^{-1}+d_1k_1+d_2^{-1}k_2+(d_1+d_2-1)r$ | 45 |
| $d_1^{-2}+k_1$ | $1+d_2^{-1}k_2$ | $(d_1+d_2)^{-1}\left(d_1^{-1}+d_1k_1+d_2+k_2\right)$ | $d_1^{-1}+d_1k_1+d_2+k_2+(d_1+d_2-1)r$ | 46 |
| | $1+d_2^{-2}k_2$ | $(d_1+d_2)^{-1}\left(d_1^{-1}+d_1k_1+d_2+d_2^{-1}k_2\right).$ | $d_1^{-1}+d_1k_1+d_2+d_2^{-1}k_2+(d_1+d_2-1)r$ | 47 |
| | $d_2^{-1}+k_2$ | $(d_1+d_2)^{-1}\left(d_1^{-1}+d_1k_1+1+d_2k_2\right)$ | $d_1^{-1}+d_1k_1+1+d_2k_2+(d_1+d_2-1)r$ | 48 |
| | $d_2^{-2}+k_2$ | $(d_1+d_2)^{-1}\left(d_1^{-1}+d_1k_1+d_2^{-1}+d_2k_2\right)$ | $d_1^{-1}+d_1k_1+d_2^{-1}+d_2k_2+(d_1+d_2-1)r$ | 49 |

**刘振亚**(1998－), 男, 硕士, 主要研究领域为密码工程.

**林璟锵**(1978－), 男, 博士, 教授, 博士生导师, 主要研究领域为密码工程.