密码学报 ISSN 2095-7025 CN 10-1195/TN
Journal of Cryptologic Research,2014,1(2):155–166

E-mail: jcr@cacrnet.org.cn
http://www.jcr.cacrnet.org.cn
Tel/Fax: +86-10-81033101

SM2 椭圆曲线门限密码算法*

尚 铭[1], 马 原[1,2,3], 林璟锵[2,3], 荆继武[2,3]

1. 中国科学院大学, 北京 100049
2. 中国科学院数据与通信保护研究教育中心, 北京 100093
3. 中国科学院信息工程研究所, 北京 100093
通讯作者: 马原, E-mail: yma@dacas.cn

摘 要: 在门限密码学中, 私钥信息被分享给独立的多个参与者, 每一次私钥计算都需要多个参与者同意, 从而提高算法安全性; 而且当少量参与者发生故障、不可用时,不影响私钥的可用性. 一个安全的 (t,n) 门限密码算法应当满足: (1)任意多于 t 个参与者可以计算最终的签名、交换的密钥或明文, 而 t 个或少于 t 个参与者不能得到关于以上结果的任何信息; (2)在算法执行过程中不泄露关于私钥和参与者的子私钥的任何信息. 相比于其他密码体制, 椭圆曲线密码体制在达到相同安全性的条件下所需要的密钥更短, 因此具有优越性. 本文基于最近发布的 SM2 椭圆曲线公钥密码算法, 提出了安全有效的门限密码方案, 包括门限签名算法、门限密钥交换协议和门限解密算法, 同时分析了上述算法的安全性和效率。本文提出的门限密码算法可支持有可信中心和无可信中心的不同情况, 并且具有较小的通信复杂度. 安全分析表明, (1)在 $n\geqslant 2t+1$ ($n\geqslant 3t+1$)情况下, 提出的门限签名方案可容忍对 t 个成员的窃听(中止)攻击, (2)在 $n\geqslant t+1$ ($n\geqslant 2t+1$)情况下, 提出的门限密钥交换和门限解密算法可以容忍对 t 个成员的窃听(中止)攻击.

关键词: SM2 椭圆曲线密码算法; 门限密码学
中图法分类号: TP309. 7 **文献标识码**: A

中文引用格式: 尚铭, 马原, 林璟锵, 荆继武. SM2 椭圆曲线门限密码算法[J]. 密码学报, 2014, 1(2): 155–166.
英文引用格式: Shang M, Ma Y, Lin J Q, Jing J W. A threshold scheme for SM2 elliptic curve cryptographic algorithm[J]. Journal of Cryptologic Research, 2014, 1(2): 155–166.

A Threshold Scheme for SM2 Elliptic Curve Cryptographic Algorithm

SHANG Ming[1], MA Yuan[1,2,3], LIN Jing-Qiang[2,3], JING Ji-Wu[2,3]

1. University of Chinese Academy of Sciences, Beijing 100049, China
2. Data Assurance and Communication Security Research Center, Chinese Academy of Sciences, Beijing 100093, China
3. Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
Corresponding author: Ma Yuan, E-mail: yma@dacas.cn

Abstract: In threshold cryptography, a private key is shared among multiple participants, and any private-key computation involves a threshold number of participants, hence to improve the security. When a small number of participants are unavailable, the shared private key is still available. A secure threshold cryptographic algorithm should satisfy that, (1) any t players can figure out the signature, the exchanged

* 基金项目: 国家重点基础研究发展项目(973 计划)(2013CB338001)
收稿日期: 2013-12-15 定稿日期: 2014-03-14

key or the plaintext, and t or less than t players cannot obtain any available information of the above results, and (2) the execution of the algorithm must not leak any information about the key or the subkeys. Compared with other cryptosystems, elliptic curve cryptosystem uses a much shorter key to achieve an equivalent level of security, thus is superior. In this paper, we design a threshold scheme for the SM2 elliptic curve cryptographic algorithm, consisting of a threshold signature scheme, a threshold key exchange protocol and a threshold decryption algorithm. In addition, we analyze the security and efficiency of the proposed SM2 threshold schemes. Our schemes can work with or without a trusted dealer, and have a small communication load. The security analysis indicates that, (1) the proposed threshold signature algorithm is secure in the presence of t eavesdropping (halting) faults if the total number of players is $n \geqslant 2t+1$ ($n \geqslant 3t+1$), (2) the proposed threshold key exchange protocol and threshold decryption algorithm are secure in the presence of t eavesdropping (halting) faults if the total number of players is $n \geqslant t+1$ ($n \geqslant 2t+1$).

Key words: SM2 elliptic curve cryptographic algorithm; threshold cryptography

1 引言

对于一个 (t,n) 秘密分享方案[1], 任意多于 t 个参与者可以恢复出秘密, t 个或少于 t 个参与者不能得到关于秘密的任何信息; 门限密码算法是在秘密分享方案的基础上构建而来. 门限密码算法中的私钥信息被分享给独立的多个参与者, 每一次私钥计算都需要多个参与者同意, 从而提高算法安全性和健壮性; 当少量参与者发生故障、不可用时, 不影响私钥的可用性. 一个合理的 (t,n) 门限密码算法应当满足: (1)任意多于 t 个参与者可以计算最终的签名、交换的密钥或明文, 而 t 个或少于 t 个参与者不能得到关于以上结果的任何信息; (2)在算法执行过程中不泄露关于私钥和参与者的子私钥的任何信息.

目前的门限密码一般可以分为需要可信中心和不需要可信中心两类. 当可信中心存在时, 可以方便地实现秘密分发, 减少小组成员之间的通信量和计算量; 但一个被小组内所有成员信任的可信中心并不是一直存在的, 此时需要小组成员联合实现对秘密的分享, 即无可信中心方案.

门限签名作为门限密码学的重要研究内容, 最早由 Desmedt[2]等人提出. 1991 年 Desmedt 和 Frankel 提出了基于 RSA 的 (t,n) 门限签名方案[3], Wang 等人给出了基于离散对数的门限签名方案[4], 此后 Harn 在 1994 年提出了基于 ElGamel 的、不需要可信中心的门限签名方案[5]. Gennaro 等人解决了 DSS(Digital Signature Standard)签名方案的构造[6], 提出了秘密乘积、求逆运算的分享方法, 并基于多种分布式可验证秘密分享方案来构造一个健壮安全的门限 DSS 签名. Miyazaki[7]在 2001 年提出了一个基于椭圆曲线 EIGamal 的门限签名方案. 门限加解密算法广泛应用在在分布式系统中以保护数据安全. 门限解密可以用来保证存储系统中的数据机密性和完整性[8], 以及保证密钥管理系统的安全性[9].

因为基于椭圆曲线上离散对数问题的困难性要高于一般乘法群上的离散对数问题的困难性, 且椭圆曲线所基于的域的运算位数要远小于传统离散对数的运算位数, 椭圆曲线密码体制比原有的密码体制(如 RSA 和 DSA)更具优越性. 2010 年底, 我国国家密码管理局发布了 SM2 椭圆曲线公钥密码算法[10], 包括数字签名算法、密钥交换协议和公钥加解密算法. 作为我国的首个发布的公钥密码算法标准, SM2 算法旨在保障各类信息系统的安全, 对我国商用密码产品和信息安全体系建设意义重大.

本文首先分析了 SM2 数字签名算法, 并针对存在可信中心和不存在可信中心的情况, 分别提出了门限签名方案; 接着, 设计了门限密钥交换协议和门限解密算法; 然后分析了上述门限密码算法的安全性. 安全分析表明, 在 $n > 2t$ 条件下, 任何 $2t+1$ 个参与者集合可以生成一个有效的数字签名; 我们的门限签名方案可以抵抗 $n/2$ 的窃听攻击和 $n/3$ 中止攻击. 最后, 对方案的效率和安全性进行了分析.

2 安全模型与定义

2.1 敌手模型

假设敌手 A 可以至多攻击 n 个成员中的 t 个, 我们定义了如下三种敌手

(1) 窃听(eavesdropping)敌手, 可以获得用户节点的存储信息以及窃听到所有广播的消息;

(2) 中止(halting)敌手, 可以在每轮通信开始的时候让用户中止发送消息;

(3) 恶意(malicious)敌手, 除了具有窃听能力, 还可以迫使用户修改协议

我们假设敌手的计算能力是在概率多项式图灵机模型下的, 因此是无法求解椭圆曲线上的离散对数问题的. 敌手类型又可以分为静态的(static)和自适应的(adaptive), 静态敌手在协议开始前选定要攻击的用户, 而自适应敌手可以在计算中选择. 本文针对窃听和中止敌手, 而且只考虑静态敌手的情况. 对于恶意敌手, 可以在方案中引入可验证的密码分享技术来容忍恶意敌手; 对于自适应敌手, 可以采用 Canetti 等人[11]提出的技术对方案进行改进来容忍自适应敌手, 这将是本文的下一步工作.

2.2 安全性定义

本文定义的门限方案的安全性包括不可伪造性/保密性和健壮性, 定义如下.

定义 1((t,n)门限签名方案的不可伪造性) 给定系统参数, 敌手 A 最多可以破坏 t 个成员, 可以拥有交互运行中的视图, 可以进行 k 次适应性的选择消息 $m_1,\cdots,m_k$ 的门限签名查询, 而最终敌手 A 能产生一个新消息 m 的门限签名的伪造的概率是可忽略的.

定义 2((t,n)门限密钥交换方案的保密性) 给定系统参数, 敌手 A 最多破坏 t 个成员, 可以拥有交互运行中的视图, 可以进行 k 次适应性的选择协商密钥 $K_1,\cdots,K_k$ 的查询, 而最终敌手 A 能成功冒充成员进行密钥协商的概率是可忽略的.

定义 3((t,n)门限解密方案的保密性) 给定系统参数, 敌手 A 最多可以破坏 t 个成员, 可以拥有交互运行中的视图, 可以进行 k 次适应性的选择密文 $C_1,\cdots,C_k$ 的门限解密结果查询, 而最终敌手 A 能解密新密文 C 的概率是可忽略的.

定义 4((t,n)门限签名/密钥交换/解密方案的健壮性) 在敌手 A 最多可以破坏 t 个成员的情况下, 方案仍然能够成功的运行.

3 SM2 椭圆曲线公钥密码算法

SM2 椭圆曲线公钥密码算法包括了数字签名算法、密钥交换协议和公钥加密算法 3 部分. 3 部分使用相同的公钥和私钥, 但设计了不同的算法来实现数字签名、密钥交换和公钥加解密功能.

在使用 SM2 算法之前, 各通信方先设定相同的公开参数, 包括 p、q、E 和 G: p 是大素数, E 是定义在有限域 F_p 上的椭圆曲线, $G=(x_G,y_G)$ 是 E 上 q 阶的基点.

3.1 数字签名算法

密钥产生:

(1) 随机选取秘密 d, $d\in[1,\ q-1]$;

(2) 计算 $P=dG$, 并将 P 作为公钥公开, d 作为私钥保存.

签名生成:

(3) 签名者选取随机数 $k\in[1,\ q-1]$, 计算 $kG=(x_1,y_1)$;

(4) 计算 $r=\left(\mathrm{Hash}\left(m\right)+x_1\right)\bmod q$，其中 m 是待签名的消息，$\mathrm{Hash}(\cdot)$ 为单向哈希函数；若 $r=0$ 或 $r+k=q$，则重新选取随机数 k；

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

签名验证:

(6) 验证者接收到 m 和 (r,s) 后，先检查是否满足 $r,s\in[1,\,q-1]$ 且 $r+s\neq q$；然后计算 $\left(x_1',y_1'\right)=sG+\left(r+s\right)P$；

(7) 计算 $r'=\left(\mathrm{Hash}\left(m\right)+x_1'\right)\bmod q$；判断 r' 与 r 是否相等，若二者相等则签名验证通过，否则验证失败.

3.2 密钥交换协议

P_{A}、P_{B} 和 d_{A}、d_{B} 分别表示用户 A 和 B 的公钥和私钥，Z_{A} 和 Z_{B} 分别表示 A 和 B 的唯一标识，$\|$ 表示数据拼接，& 表示比特与运算，$\mathrm{KDF}(\mathrm{ks},\mathrm{klen})$ 是密钥派生函数：以 ks 为种子、产生 klen 比特的伪随机序列，记 $\omega=\left\lceil\left(\left\lceil\log_2(q)\right\rceil/2\right)\right\rceil-1$.

用户 A

(A.1) 选取随机数 $r_{\mathrm{A}}\in[1,\,q-1]$，计算 $R_{\mathrm{A}}=r_{\mathrm{A}}G=\left(x_2,y_2\right)$ 并发送给用户 B;

用户 B

(B.1) 选取随机数 $r_{\mathrm{B}}\in[1,\,q-1]$，计算 $R_{\mathrm{B}}=r_{\mathrm{B}}G=\left(x_3,y_3\right)$ 并发送给用户 A;

(B.2) 计算 $x_{\mathrm{B}}=2^w+(x_3\ \&\ (2^w-1))$ 和 $t_{\mathrm{B}}=\left(d_{\mathrm{B}}+x_{\mathrm{B}}r_{\mathrm{B}}\right)\bmod q$；

(B.3) 验证接收到的 R_{A} 是椭圆曲线 E 上的点，验证通过后计算 $x_{\mathrm{A}}=2^w+(x_2\ \&\ (2^w-1))$；

(B.4) 计算 $V=t_{\mathrm{B}}(P_{\mathrm{A}}+x_{\mathrm{A}}R_{\mathrm{A}})=\left(x_V,y_V\right)$；若 V 是椭圆曲线 E 上的无穷远点，则重新选取 r_{B}、重新协商;

(B.5) 计算 $K_{\mathrm{B}}=\mathrm{KDF}\left(x_V\|y_V\|Z_{\mathrm{A}}\|Z_{\mathrm{B}},\mathrm{klen}\right)$；

用户 A

(A.2) 计算 $x_{\mathrm{A}}=2^w+(x_2\ \&\ (2^w-1))$ 和 $t_{\mathrm{A}}=\left(d_{\mathrm{A}}+x_{\mathrm{A}}r_{\mathrm{A}}\right)\bmod q$；

(A.3) 验证接收到的 R_{B} 是椭圆曲线上的点，验证通过后计算 $x_{\mathrm{B}}=2^w+(x_3\ \&\ (2^w-1))$；

(A.4) 计算 $W=t_{\mathrm{A}}(P_{\mathrm{B}}+x_{\mathrm{B}}R_{\mathrm{B}})=\left(x_W,y_W\right)$；若 W 是椭圆曲线 E 上的无穷远点，则重新选取 r_{A}、重新协商;

(A.5) 计算 $\mathrm{KDF}\left(x_W\|y_W\|Z_{\mathrm{A}}\|Z_{\mathrm{B}},\mathrm{klen}\right)$；

3.3 公钥加解密算法

M 是比特长度为 mlen 的明文，$\oplus$ 表示比特异或.

加密算法:

(1) 选取随机数 $l\in[1,\,q-1]$，分别计算 $C_1=lG=\left(x_4,y_4\right)$ 和 $lP=\left(x_5,y_5\right)$；

(2) 计算 $e=\mathrm{KDF}\left(x_5\|y_5,\mathrm{mlen}\right)$；

(3) 计算 $C_2=M\oplus e$ 和 $C_3=\mathrm{Hash}\left(x_5\|M\|y_5\right)$；

(4) 输出密文 $C=C_1\|C_3\|C_2$.

解密算法:

(1) 验证 C_1 是否在椭圆曲线上, 计算 $dC_1=(x_5,y_5)$;

(2) 计算 $e=\mathrm{KDF}(x_5\|y_5,\mathrm{mlen})$;

(3) 计算 $M=C_2\oplus e$;

(4) 计算 $C_3'=\mathrm{Hash}(x_5\|M\|y_5)$, 并验证 $C_3'=C_3$ 是否成立, 若不成立则报错退出;

(5) 输出明文 M.

4 门限密码预备知识

4.1 Shamir秘密分享方案(SS)

在 Shamir (t,n) 门限秘密分享方案中, 可信中心把秘密 d 拆分成 n 份, 分发给 n 个参与者 $U_1,U_2,\cdots,U_n$. 任意 $t+1$ 个或 $t+1$ 个以上的参与者一起可恢复秘密 d, 任意 t 个参与者一起不能得到秘密 d 的任何信息. 详细方案如下:

(1) 可信中心秘密构造 t 阶多项式 $f(x)=\sum_{i=0}^{t}a_ix^i$, 其中秘密 $d=f(0)=a_0$;

(2) 可信中心计算 $d_i=f(i)$, 并秘密地把 d_i 分别发送给参与者 U_i, $1\leqslant i\leqslant n$;

(3) U_i 将 d_i 作为份额保存.

上述过程称为 t 阶 SS, 并称多项式 $f(x)$ 为分享多项式. 此时, 任意 $t+1$ 个参与者集合 Q 可通过拉格朗日插值公式 $d=\sum_{i\in Q}(d_i\prod_{j\in Q,j\neq i}\frac{j}{j-i})$ 恢复出秘密.

4.2 联合Shamir随机秘密分享(Joint-RSS)

在联合 Shamir 随机秘密分享中, n 个参与者 $U_1,U_2,\cdots,U_n$ 分别选择任意的秘密值, 并各自按照“可信中心”的步骤来分发所选秘密值的份额, 从而实现对一个随机值的秘密分享. 该随机值等于所有被分享的秘密值之和, 整个过程不需要可信中心的参与. 详细过程如下:

(1) 每个参与者 U_i 将自己作为“可信中心”, 选取随机的秘密值 $a_0^{(i)}$, 构造多项式为 $f_i(x)=\sum_{j=0}^{t}a_j^{(i)}x^j$, 执行 t 阶 SS;

(2) $U_j(1\leqslant j\leqslant n)$ 收到其余 $n-1$ 个参与者 $U_i(1\leqslant i\leqslant n,i\neq j)$ 发送给自己的 $f_i(j)$, 计算 $\sigma_j=\sum_{i=1}^{n}f_i(j)$ 作为 U_j 的份额.

上述过程称为 t 阶 Joint-RSS, 此时参与者们分享的秘密为 $\sigma=\sum_{i=1}^{n}a_0^{(i)}$, 相应的分享多项式为 $f(x)=\sum_{j=0}^{t}a_jx^i$, 其中 $a_j=\sum_{i=1}^{n}a_j^{(i)}$.

4.3 联合Shamir零秘密分享(Joint-ZSS)

联合 Shamir 零秘密分享与联合 Shamir 随机秘密分享类似; 不同的是, 此时每个参与者 U_i 选取的

秘密 $a_0^{(i)}=0$，所以参与者分享的秘密 a_0 也为 0.

4.4 秘密和/差的分享

假设参与者 U_i 已经得到了 u 和 v 的 t 阶 *SS* 分享份额：$u_i=f_u(x)$、$v_i=f_v(x)$，其中 $u=f_u(0)$、$v=f_v(0)$，$f_u(x)$ 和 $f_v(x)$ 是不同的 t 阶多项式.

容易看出，每个参与者 U_i 分别计算 $z_i=u_i\pm v_i$ 即可得到秘密和/差 $z=u\pm v$ 的秘密分享份额，z 的分享多项式是 $f_z(x)=f_u(x)\pm f_v(x)$，$f_z(x)$ 也是 t 阶的.

4.5 秘密乘积的分享[6](Mul-SS)

与秘密和/差的分享类似，参与者 U_i 将自己的分享份额 u_i 与 v_i 相乘，就可实现对乘积 $h=uv$ 的秘密分享. 此时，分享多项式 $f_h(x)=f_u(x)\times f_v(x)$ 的阶数为 $2t$，也就是说需要至少 $2t+1$ 个参与者才能恢复秘密 uv. 需要注意的是，$f_h(x)$ 是由两个 t 阶多项式相乘得到，所以 $f_h(x)$ 不是不可约多项式，例如 $f_h(x)$ 不可能是 x^2+1. 因此 $f_h(x)$ 的系数并不是完全随机的，这就会降低安全性. 为此，需要对 $f_h(x)$ 进一步“随机化”：加上随机的 $2t$ 阶多项式，使其系数完全随机. 详细过程如下：

(1) 参与者执行 $2t$ 阶 Joint-ZSS，U_i 的分享份额为 α_i，作为随机化因子；

(2) U_i 计算 $h_i=u_iv_i+\alpha_i$ 作为对秘密 $h=uv$ 的分享份额.

此时参与者们分享的秘密为 uv，至少 $2t+1$ 个参与者 U_i 广播其 h_i，通过插值公式可以恢复出 $h=uv$.

4.6 秘密逆的分享[6](Inv-SS)

设 U_i 已经分享了 u，份额是 u_i. 逆分享的基本思想为首先分享一个随机秘密 β，在计算 $(\beta u)^{-1}$ 的基础上得到 $c=u^{-1} \bmod q$ 的分享份额. 逆的分享过程如下：

(1) 参与者集合执行 t 阶 Joint-RSS 分享随机秘密 β，分享份额为 β_i；

(2) 至少 $2t+1$ 个参与者执行 3. 5 节中 u 与 β 乘积的分享过程 Mul-SS，并广播自己乘积的分享份额 $(\beta u)_i$；

(3) U_i 记录广播的 $(\beta u)_j\ (1\leqslant j\leqslant n)$，通过插值公式计算出 βu，并计算 $c_i=(\beta u)^{-1}\beta_i \bmod q$ 得到 u^{-1} 的分享份额.

需要注意的是，广播 $(\beta u)_i$ 不会影响机密性，因为 β_i 随机并且保密，所以 u_i 的信息不会泄露. 由于分享随机数 β 的多项式是 t 阶，所以 c 的分享多项式也是 t 阶，也就是说虽然需要至少 $2t+1$ 个参与者才能得到分享份额 c_i，但 $t+1$ 个 c_i 的持有者就可以恢复出 c.

4.7 点乘的分享(PM-SS)

Shamir 秘密分享方案可以直接应用在椭圆曲线上点乘运算的分享中. 设 U_i 已经分享了 u，份额是 u_i，任意 $t+1$ 个参与者的集合 Q 计算通过分享 u_iG 计算 uG 的过程如下：

(1) 至少 $t+1$ 个 U_i 计算点乘 u_iG，并广播；

(2) U_i 通过插值公式计算

$$uG=\left(\sum_{i\in Q}u_i\prod_{j\in Q,j\neq i}\frac{j}{j-i}\right)G=\sum_{i\in Q}(u_iG)\prod_{j\in Q,j\neq i}\frac{j}{j-i}$$

5　SM2 椭圆曲线门限密码算法

5.1　SM2门限签名方案(SM2-Sign-Thresh)

门限签名方案一般包括三个部分: 密钥产生、签名生成和签名验证. 在密钥产生阶段, 参与者们(通过可信中心, 或不通过可信中心)对私钥分享并公开公钥; 签名生成时, 每个参与者首先通过秘密分享的方式计算得到 r, 而后得到 s 的分享份额 s_i, 超过门限个的参与者广播其 s_i 则可通过插值公式计算出签名 s; 在签名验证时, 直接对签名结果 (r,s) 进行验证.

对于 SM2 门限签名方案, 需要分享的秘密为私钥 d. 但是, 对于随机数 k, 因为在已知签名 (r,s) 和 k 的情况下, 参与者可以推导出私钥 d, 所以在签名过程中 k 也必须保密.

为了降低复杂度, 我们对 SM2 签名方案中 s 计算式进行如下等价变形:

$$s=(1+d)^{-1}(k-rd)=(1+d)^{-1}(k-(1+d)r+r)=\left((1+d)^{-1}(k+r)-r\right)\bmod q$$

可以看出, 上式只用到了 $(1+d)^{-1}$ 而没有单独用到 d. 接下来, 在以上讨论和分析的基础上, 我们针对存在和不存在可信中心的情况, 分别提出了安全有效的 SM2 算法门限签名方案. 可信中心只在密钥产生时参与, 签名生成和验证时并不需要可信中心参与.

对于存在可信中心的情况, 密钥产生比较简单, 可信中心可以直接将 $(1+d)^{-1}$ 作为秘密, 通过 t 阶 SS 分享给参与者; 对于不存在可信中心的情况, 参与者们需要联合分享秘密, 每个参与者 U_i 独立生成秘密 $a_0^{(i)}$, 通过 t 阶 Joint-RSS 分享私钥 d, ,再执行 Inv-SS 得到 $(1+d)^{-1}$ 的分享份额用于门限签名.

签名生成阶段, 参与者首先通过 Joint-RSS 分享随机数 k, 执行 PM-SS 计算 r, 在分享 $(1+d)^{-1}$ 和 $(k+r)$ 的基础上, 执行 Mul-SS, 再减去 r 即可得到子签名 s_i, 最后, 超过 $2t+1$ 个参与者广播其子签名就可以通过插值得到签名 (r,s).

门限签名详细过程如下.

5.1.1　密钥产生

A.　存在可信中心

给定门限 t, 参与者集合 $\{U_1,U_2,\cdots,U_n\}$, $n\geqslant 2t+1$. 可信中心随机选取一个整数 d 作为私钥保密, 计算 $P=dG$ 作为公钥公开.

秘密分发时, 计算 $d'=(1+d)^{-1}\bmod q$ 并执行 t 阶 SS, 将 d' 分享给参与者 U_i.

B.　不存在可信中心

对于参与者集合 $\{U_1,U_2,\cdots,U_n\}$, $n\geqslant 2t+1$. 每个参与者 U_i 首先执行 t 阶 Joint-RSS, 分享秘密为私钥 d, 分享份额为子私钥 d_i; 参与者集合执行 PM-SS 得到公钥 $P=dG$.

由于在签名时需要用到 $d'=(1+d)^{-1}$, 因此参与者需要将 d' 分享, 执行 Inv-SS 得到 d' 的分享份额 d_i'. 具体过程如下:

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

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

(3)　U_i 记录广播的 $\gamma_j(1\leqslant j\leqslant n)$, 通过插值公式恢复出 $\gamma=\beta(1+d)$;

(4)　U_i 计算 $d_i'=\gamma^{-1}\beta_i$ 得到 $(1+d)^{-1}$ 的分享份额.

5.1.2　签名生成

至少 $2t+1$ 个参与者 U_i 执行以下过程对消息进行签名生成:

(1) 执行 $2t$ 阶 Joint-ZSS, 分享份额 μ_i;

(2) 执行分享随机秘密 k 的 t 阶 Joint-RSS, 分享份额为 k_i;

(3) 执行 PM-SS 得到点乘 $kG=(x_1,y_1)$, 并计算 $r=\mathrm{Hash}(m)+x_1 \bmod q$;

(4) U_i 计算 $s_i=d_i'(k_i+r)+\mu_i-r$, 得到签名 s 的分享份额 s_i;

(5) 至少 $2t+1$ 个参与者 U_i 广播其 s_i;

(6) U_i 通过插值公式得到签名结果 s.

5.1.3 签名验证

门限签名验证过程和 3.1 节中的签名验证是一致的.

5.2 SM2门限密钥交换协议(SM2-Exch-Thresh)

对于 SM2 门限密钥交换协议, 由于不需要秘密乘积或秘密逆的分享, 因此在密钥产生时与 5.1 节略有不同. 当存在可信中心时, 可信中心执行 t 阶 SS 将私钥 d_A 和 d_B 分别分享给参与者集合 A 和 B; 当不存在可信中心时, A 和 B 分别执行 t 阶 Joint-RSS 分享私钥 d_A 和 d_B, 并执行 PM-SS 得到公钥 P_A 和 P_B.

密钥交换时, 至少 $t+1$ 个 B 的参与者, 进行如下过程进行门限密钥交换(参考 3.2 节, 这里只列举了 B 需要用到门限密码的部分 B.1、B.2 和 B.4):

(B.1.1) 执行 t 阶 Joint-RSS, 共享随机数 r_B, 共享份额为 $r_{B,i}$;

(B.1.2) 执行 PM-SS, 计算 $R_B=r_BG=(x_3,y_3)$;

(B.2) U_i 计算 $x_B=2^w+(x_3 \,\&\, (2^w-1))$ 和 $t_{B,i}=(d_{B,i}+x_B r_{B,i}) \bmod q$, 秘密保存 $t_{B,i}$;

(B.4.1) U_i 计算 $V_i=t_{B,i}(P_A+x_A R_A)=(x_{V,i},y_{V,i})$;

(B.4.2) 通过插值公式恢复出点 V; 若 V 是椭圆曲线 E 上的无穷远点, 则重新选取 $r_{B,i}$、重新协商.

5.3 SM2门限解密算法(SM2-Decry-Thresh)

与门限密钥交换中的密钥产生过程一致, 在门限解密算法中, 参与者集合 $\{U_1,U_2,\cdots,U_n\}$ 分享私钥 d, 分享份额为 d_i. 在这里, 我们对 3.3 节解密过程中的第(1)步进行扩展, 其他步骤没有用到门限密码的知识. 至少 $t+1$ 个参与者:

(1.1) U_i 计算 d_i 和点 P 点乘 $d_iC_1=(x_{5,i},y_{5,i})$;

(1.2) 通过插值公式恢复出点 $dC_1=(x_5,y_5)$.

6 安全性和效率分析

6.1 安全性

根据文献[12]有:

引理 1[12] 如果签名方案是不可伪造的, 而且门限签名方案是可模拟的, 那么这个门限签名方案也是不可伪造的.

引理 2 对于 t 个成员的的窃听攻击, 如果 $n\geqslant 2t+1$, 那么 (t,n) 门限签名方案 SM2-Sign-Thresh 是健壮的; 对于 t 个成员的中止攻击, 如果 $n\geqslant 3t+1$, 那么 (t,n) 门限签名方案 SM2-Sign-Thresh 是健壮的.

证明: 根据式 $s_i=d_i'(k_i+r)+\mu_i-r$, 由于分享 d' 和 k 的多项式都是 t 阶, 那么分享 s 的多项式是 $2t$ 阶, 所以 $2t+1$ 以上个参与者广播其子签名 s_i, 能通过插值公式恢复出签名 s, 也即是说, 需要至少

$2t+1$ 个参与者就能完成签名. 因此, 对于 t 个成员的中止攻击, 需要保证 $n\geq 3t+1$ 才能完成签名过程. 证毕.

引理 3　门限签名方案 SM2-Sign-Thresh 在 t 个成员的窃听攻击下是不可伪造的.

证明: 首先证明门限签名方案是可模拟的. 方案的模拟过程如图 1 所示. 输入公钥 P, 消息 m, 签名 (r,s), 不失一般性, 设一个敌手可以控制前 t 个参与者: 对他们进行窃听攻击或者中止攻击, 其余参与者为诚实参与者. 因此控制的份额为 $d_1', d_2', \cdots, d_t'$, 容易证明通过私钥份额 d_i 执行 Inv-SS 得到 d_i' 的过程是安全的.

从图 1 不难看出, 模拟协议 SIM 与签名方案 SM2-Sign-Thresh 中的变量是一致的, 接下来我们证明他们具有相同的概率分布.

(1) 因为 Shamir 的秘密分享方案是信息论安全的, 因此所有的分享份额具有相同的概率分布. 因此 $\hat{\mu}_i$ 的分布与 μ_i 一致, 都是 $[1, q-1]$ 上的均匀随机分布.

(2) $\hat{k}_1, \hat{k}_2, \cdots, \hat{k}_t$ 是由 Joint-RSS 产生的, 因此 $r_i^* = \hat{k}_i G\ (1\leqslant i\leqslant t)$ 满足均匀分布, 而剩下的 $r_i^*\ (t+1\leqslant i\leqslant 2n)$ 是由 $r_i^*\ (1\leqslant i\leqslant t)$ 和 r^* 所确定的. 因此 $r_i^*\ (t+1\leqslant i\leqslant 2n)$ 与 $r_i^*\ (1\leqslant i\leqslant t)$ 具有相同的概率分布.

(3) 如上所证, $\hat{s}_i$ 也满足在 $[1, q-1]$ 上的均匀随机分布, 并且与 $s_i = d_i'(k_i + r) + \mu_i - r$ 一致, $\hat{s}_i$ 也满足该等式.

结合引理 1, 可证明门限签名方案 SM2-Sign-Thresh 具有不可伪造性. 证毕.

SIM

输入: 公钥 P, 消息 m, 签名 (r,s), 控制份额 $(d_1', d_2', \cdots, d_t')$

(0) 计算 $r^* = sG + (r+s)P$;

(1) 诚实参与者联合执行 Joint-RSS, 分享份额为 $\hat{\mu}_i\ (1\leqslant i\leqslant n)$, 由于 SIM 可以听到敌手的份额, 因此 SIM 可以得到这些值(因为 SIM 知道 $\hat{\mu}_1, \hat{\mu}_2, \cdots, \hat{\mu}_t$ 和 $\hat{\mu}=0$);

(2) 诚实参与者联合执行 Joint-RSS, 敌手分享份额为 $\hat{k}_1, \hat{k}_2, \cdots, \hat{k}_t$;

(3) 记 $r_i^* = \hat{k}_i G\ (1\leqslant i\leqslant n)$, 且满足任 $t+1$ 个 r_i^* 能够恢复出 r_i^*. 由于 SIM 已知 t 个 $\hat{k}_i$, 因此 SIM 可以计算出其余的 $r_i^* = \hat{k}_i G\ (t+1\leqslant i\leqslant n)$. 向诚实参与者广播 $r_i^*\ (t+1\leqslant i\leqslant n)$, 则可通过插值公式恢复出 $r_i^* = (x_1, y_1)$, 而 $r = (\mathrm{Hash}(m) + x_1) \bmod q$;

(4) 计算 $\hat{s}_i = d_i'(\hat{k}_i + r) + \hat{\mu}_i - r\ (1\leqslant i\leqslant t)$, 对于 $t+1\leqslant i\leqslant 2t$ 随机选取 $\hat{s}_i$. 由 $\hat{s}_i\ (1\leqslant i\leqslant 2t)$ 和 $\hat{s}_0 = s$ 可以唯一确定一个 $2t$ 阶的分享多项式, 因此可以确定其余 $\hat{s}_i\ (2t+1\leqslant i\leqslant n)$;

(5) 向诚实参与者广播 $\hat{s}_i\ (2t+1\leqslant i\leqslant n)$.

图 1　门限签名方案的模拟协议

Figure 1 Simulation protocol for the threshold signature scheme

进一步, 根据引理 2 和引理 3, 可以得到:

定理 1　门限签名方案 SM2-Sign-Thresh 是安全的, 即具有不可伪造性和健壮性: 当 $n\geqslant 2t+1$ 时, 可抵抗对 t 个成员的窃听攻击; 当 $n\geqslant 3t+1$ 时, 则可抵抗对 t 个成员的中止攻击.

定理 2　当 $n\geqslant t+1\ (n\geqslant 2t+1)$ 时, SM2 门限密钥交换协议 SM2-Exch-Thresh 和门限解密算法 SM2-Decry-Thresh 可以容忍对 t 个成员的窃听(中止)攻击.

证明: 首先证明健壮性. 对于 SM2-Exch-Thresh 和 SM2-Decry-Thresh, 根据 PM-SS 可知, $t+1$ 个参与者联合即可完成密钥交换和解密.

其次证明机密性, 即协议交互过程中没有额外有效的信息泄露. 对于 SM2-Exch-Thresh, 通过参与者 U_i 的 $V_i=t_{\mathrm{B},i}\left(P_{\mathrm{A}}+x_{\mathrm{A}}R_{\mathrm{A}}\right)=\left(d_{\mathrm{B},i}+x_{\mathrm{B}}r_{\mathrm{B},i}\right)\left(P_{\mathrm{A}}+x_{\mathrm{A}}R_{\mathrm{A}}\right)$, 无法得到 $t_{\mathrm{B},i}$ 的信息, 这是一个椭圆曲线上的离散对数问题. 另外, 类似的, 门限密钥交换过程也只广播了 d_iC_1 信息. 又由于 Joint-RSS 本身的安全性, 所以破坏少于 $t+1$ 个参与者不能获得关于私钥的任何信息. 因此, SM2-Exch-Thresh 和 SM2-Decry-Thresh 是安全的. 证毕.

6.2 效率

6.2.1 通信量

本文以协议中传输的数据量来估算门限密码算法的通信复杂度. 由于门限密钥交换协议和门限解密算法的门限密码部分较为简单, 在这里只讨论门限签名方案的通信量. 在以下说明中, $|x|$ 表示 x 的比特位数.

对于门限签名方案, 在密钥产生阶段, 当不存在可信中心时, 每个参与者首先执行 Joint-RSS, 此时需要向其他参与者发送 $|q|$ 比特信息得到子私钥, 并广播自己的子公钥 $2|p|$ 比特(包括 x 坐标和 y 坐标, 可通过点压缩至 $|p|$ 比特). 此后, 参与者需分享 $(1+d)^{-1}$, 需执行两次 Joint-RSS 和广播 γ_i. 因此在私钥产生阶段共需广播 $2|p|+|q|$ 比特以及秘密传送 $3(n-1)|q|$ 比特. 当可信中心(trusted dealer, TD)存在时, 可信中心将私钥 d 和 $(1+d)^{-1}$ 分享给参与者(若只做签名, 私钥 d 不必分享, 只需分享享 $(1+d)^{-1}$ 即可), 再广播公钥即可. 如表 1 所示, 需要注意的是, 可信中心存在时, 参与者在密钥产生阶段并不需要通信, 表中的数据是可信中心的通信量. 在 SM2 椭圆曲线推荐参数中, p 和 q 的比特长度都是 256 比特, 因此 $|p|=|q|$.

在签名阶段, 假设有 $T(T>2t)$ 个参与者参与签名, 则每个参与者需要执行 t 阶 Joint-RSS 和 $2t$ 阶 Joint-ZSS, 以及广播 $2|p|$ 比特的 k_iG 和 $|q|$ 比特的 s_i. 因此共需要秘密传送 $2(T-1)|q|$ 比特, 广播 $2|p|+|q|$ 比特.

表 1 SM2 门限签名方案的通信量

Table 1 The communication traffic for the SM2 threshold signature scheme

过程	广播(bit)	秘密传送(bit)
密钥产生(不存在 TD)	$2\|p\|+\|q\|$	$3(n-1)\|q\|$
密钥产生(存在 TD)	$2\|p\|$	$2(n-1)\|q\|$
签名生成	$2\|p\|+\|q\|$	$2(T-1)\|q\|$

在 ECDSA 中, 签名 $s=k^{-1}\left(\mathrm{Hash}(m)+rd\right)\bmod q$, $r=kG \bmod q$. 因此, 在使用相同的基础知识来设计基于 ECDSA 的门限签名方案时, 签名生成 s 时需要执行 Inv-SS 分享 k^{-1} 再执行 Mul-SS 分享 s. 而在 SM2 中, 是对私钥 d 求逆, 因此, 我们基于 SM2 门限签名方案的签名通信量更小, 只需在密钥产生时执行 Inv-SS, 接下来的每次签名都只执行 Mul-SS. 此外, 在可信中心存在的情况下, SM2 的门限签名方案的优势就更加明显, 因为只需要可以由可信中心执行一次求 $(1+d)^{-1}$ 的过程即可, 参与者不需要执行 Inv-SS.

6.2.2 计算量

本文以椭圆曲线 $E(F_p)$ 上点加、点乘以及 Z_q 上逆元运算的计算量来估计门限签名方案的计算复

杂度. 相比上述运算, 签名时其他运算的计算量都很小.

不存在可信中心时, 在密钥产生和签名生成阶段各执行了一次 PM-SS, 并且在密钥产生时执行了逆元运算; 存在可信中心时, 可信中心只需计算执行逆元运算计算 $(1+d)^{-1}$ 和执行一次点乘计算公钥.

点乘运算是由若干步点加运算组成的, 对于整数 $k \in Z_q$ 和基点 G, 点乘运算 kG 可以转换为 $2(|q|-1)$ 次点加运算. 在 PM-SS 过程中, 每个参与者需要执行 $t+2$ 次点乘和 t 次点加运算, 约为 $2(t+2)(|q|-1)$ 次点加.

门限签名方案的计算量如表 2 所示. 需要注意的是, 可信中心存在时, 参与者在密钥产生阶段并不需要计算, 表中的数据是可信中心的计算量.

表 2 SM2 门限签名算法的计算量

Table 2 The computation quantity for the SM2 threshold signature scheme

过程	$E(F_p)$ 上的点加(次)	Z_q 的逆元(次)
密钥产生(不存在 TD)	$2(t+2)(\|q\|-1)$	1
密钥产生(存在 TD)	$2(\|q\|-1)$	1
签名生成	$2(t+2)(\|q\|-1)$	0

7 结束语

本文提出了 SM2 门限密码算法, 包括门限签名算法、门限密钥交换协议和门限解密算法, 并对它们的安全性和效率进行了分析. 文章对可信中心是否存在的情况分别进行了讨论, 结果表明: 在不存在可信中心的情况下我们的方案具有较小的通信复杂度, 当可信中心存在时该优势更加明显. 安全分析表明: 在 $n>2t$ 条件下, 任何 $2t+1$ 个参与者集合可以生成一个有效的数字签名; 我们的门限签名方案可以抵抗 $n/2$ 的窃听攻击和 $n/3$ 中止攻击. 接下来对于恶意敌手和自适应敌手下安全门限方案的研究是下一步工作的重点.

References

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

[2] Desmedt Y, Frankel Y. Threshold cryptosystems[C]. In: Advances in Cryptology—CRYPTO '89 Proceedings. Springer New York, 1990: 307–315.

[3] Desmedt Y, Frankel Y. Shared generation of authenticators and signatures[C]. In: Advances in Cryptology—CRYPTO '91. Springer Berlin Heidelberg, 1992: 457–469.

[4] Wang C T, Lin C H, Chang C C. Threshold signature schemes with traceable signers in group communications[J]. Computer Communications, 1998, 21(8): 771–776.

[5] Harn L. Group-oriented (t, n) threshold digital signature scheme and digital multisignature[J]. IEE Proceedings-Computers and Digital Techniques, 1994, 141(5): 307–313.

[6] Gennaro R, Jarecki S, Krawczyk H, et al. Robust threshold DSS signatures[C]. In: Advances in Cryptology—EUROCRYPT'96. Springer Berlin Heidelberg, 1996: 354–371.

[7] Miyazaki K, Takaragi K. A threshold digital signature scheme for a smart card based system[J]. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2001, 84(1): 205–213.

[8] Herlihy M P, Tygar J D. How to make replicated data secure[C]. In: Advances in Cryptology—CRYPTO '87. Springer Berlin Heidelberg, 1988: 379–391.

[9] Reiter M K, Franklin M K, Lacy J B, et al. The Ω key management service[C]. In: Proceedings of the 3rd ACM Conference on Computer and Communications Security. ACM, 1996: 38–47.

[10] 国家密码管理局. SM2 椭圆曲线公钥密码算[EB/OL].(2010:12). http://www.oscca.gov.cn/UpFile/2010122214822692.pdf

[11] Canetti R, Gennaro R, Jarecki S, et al. Adaptive security for threshold cryptosystems[C]. In: Advances in Cryptology—CRYPTO '99. Springer Berlin Heidelberg, 1999: 98–116.

[12] Goldwasser S, Micali S, Rivest R L. A digital signature scheme secure against adaptive chosen-message attacks[J]. SIAM Journal on Computing, 1988, 17(2): 281–308.

作者信息

尚铭(1965–), 博士, 高级工程师. 主要研究领域为网络与信息安全.
E-mail: shm@css.com.cn

马原(1988–), 博士. 主要研究领域为网络与信息安全.
E-mail: yma@dacas.cn

林璟锵(1978–), 博士, 副研究员. 主要研究领域为网络与信息安全.
E-mail: linjq@dacas.cn

荆继武(1964–), 博士, 研究员, 中国密码学会理事. 主要研究领域为网络与信息安全.
E-mail: jing@dacas.cn

中国密码学会 2014 年 1–8 月学术会议一览

一、2014 年国产密码算法电子认证国际互操作技术研讨会

主办单位: 中国密码学会电子认证专业委员会
承办单位: 国民技术股份有限公司
会议时间: 2014 年 5 月 20 日
会议地点: 深圳市五洲宾馆
会议网址: http://www.dacas.cn/TCCA
http://www.nationz.com.cn

二、中国密码学会 2014 年量子密码专委会学术会议

主办单位: 中国密码学会量子密码专业委员会
承办单位: 清华信息科学技术国家实验室（筹）
清华大学物理系和赤峰学院物理与电子信息工程学院
会议时间: 2014 年 7 月 19 日–20 日
会议地点: 赤峰学院国际学术报告厅
会议网址: http://www.31huiyi.com/event/211326/

三、中国密码学会 2014 年会

主办单位: 中国密码学会
承办单位: 信息工程大学
会议时间: 2014 年 8 月 27 日–30 日
会议地点: 河南郑州
会议网址: http://cacr2014.cacrnet.org.cn/