文章编号：2095-2163(2024)03-0001-09 中图分类号：TP309 文献标志码：A

基于 CL-LRSS 的区块链电子病历方案

李昌辉，刘 亚
（上海理工大学 光电信息与计算机工程学院，上海 200093）

摘 要：为使得基于区块链的电子病历系统在共享数据时能够保护隐私，同时尽量保证去中心化的特点，提出了一种结合隐身地址的无证书可链接环签名算法。首先通过无证书签名体系弱化可信第三方权利避免单点故障，同时舍弃双线性对映射降低运算成本；然后结合单密钥对隐身地址技术，将数据传入接收方的隐身地址，保护接收方的隐私；结合智能合约和 IPFS 实现拥有访问控制的数据加密与存储，再记录于区块链上。通过模拟实验证明，所提方案能够高效安全地实现电子病历的传递与访问。

关键词：区块链；环签名；隐身地址；电子病历

Electronic health record scheme based on CL-LRSS and blockchain

LI Changhui, LIU Ya
(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: In order to protect the privacy of the blockchain-based electronic health record system with decentralizations when users share their data, a certificateless linkable ring signature algorithm combined with stealth addresses (CL-LRSS) is proposed. First of all, through using the certificateless signature system, the rights of trusted third parties are weakened to avoid single point of failure. At the same time, CL-LRSS adopts the ECC to reduce computing costs, not using bilinear pairing maps. Then, combined with the single-key pair stealth address technology, the data is transmitted to the receiver's stealth address to protect the receiver's privacy. In addition, smart contracts and IPFS are applied to realize data encryption with access control and storage which are recorded on the blockchain. Finally, the security proof and simulation experiments of the scheme are presented. The results shows that the proposed scheme can efficiently and safely realize the transmission and access of electronic health records.

Key words: blockchain; ring signatures; stealth address; electronic health record

0 引 言

随着医学科学的进步，电子病历系统(Electronic Health Record, EHR)[1]也被越来越多的医疗机构用来记录患者相应的医疗信息，这些信息能够及时便捷地辅助医生做出正确的医疗判断。然而，患者的电子病历通常是由医疗机构统一进行中心化的管理与存储，若发生单点故障，很容易导致电子病历系统无法正常运作并引发数据泄露[2]。此外，患者大多无法直接访问和控制自己的病历，且患者大多会在不同的医院进行诊治，不同机构间 EHR 的数据共享，可能会面临信息孤岛[3]。由此可见，电子病历系统需要具备去中心化和不可篡改的特点，而区块链[4]同时具有了这些特点，因此基于区块链的电子病历的发展成为了当前承载电子病历系统较为理想的方案[5]。然而，在区块链网络中，由于其存在可追溯性使得所有人都可以通过聚类算法或交易图等方式推导出地址与地址之间的关系[6]。对于简单基于区块链的电子病历系统，人们很容易通过区块链分析出患者对应的链上地址以及就诊时间、医疗情况和居住地址等敏感信息，因此基于区链块的电子病历中如何隐藏敏感信息尤为重要。

近年来，与区块链相结合的电子病历方案被众多研究人员提出。2017 年，薛腾飞等学者[7]提出了一个区块链医疗数据分享模型，通过将医疗机构划分等级，再结合改进的代理权益证明共识机制来实现高效共享。该方案虽然在效率上表现良好，但在数据存储方面使用的分布式存储数据库还存在一些

基金项目：上海市自然科学基金项目(22ZR1481000, 20ZR1419700)。

作者简介：李昌辉(1997-)，男，硕士研究生，主要研究方向：区块链。
通讯作者：刘 亚(1983-)，女，博士，副教授，主要研究方向：密码学、区块链。Email：liuya@ usst.edu.cn

收稿日期：2023-03-06

不足。之后,Dagher 等学者[8]提出了 Ancile 结构,能够通过智能合约控制不同角色的访问权限,并使用代理重加密确保用户不需要在本地存储密钥。然而,代理重加密存在代理更改密文的可能,因此方案容易遭受贿赂攻击或合谋攻击。2021 年,Li 等学者[9]使用基于属性的同态密码系统,来解决 EHR 的访问控制和隐私保护问题。通过改进的密文策略属性基加密方案(Ciphertext policy attribute based encryption, CP-ABE)[10],实现基于部分密文的半策略隐藏和动态权限的改变,从而实现 EHR 的细粒度访问控制,同时用同态密码系统使得保险公司可以通过密文的方式解决医疗理赔,保护患者的隐私免受非医疗机构的侵害。然而,CP-ABE 存在中心化证书颁发机构,且同态加密的使用会导致计算开销过大影响效率。同年,为了保护隐私,Wang 等学者[11]通过将区块链与群签名[12]相结合,以医疗机构为群体成员和共识节点,对 EHR 进行签名并打包上传至区块链,但是群签名存在群管理员这一中心化角色,容易遭受合谋攻击。

本文提出了基于结合隐身地址的无证书可链接环签名(CL-LRSS)的区块链电子病历方案。该方案虽也存在可信第三方,但通过引入无证书签名体系降低其权利,并且避免了第三方遭受攻击后的密钥泄露问题,同时也避免了单点故障与密钥分发问题。CL-LRSS 算法不采用运算量大的双线性对映射,大大降低了环签名的运算成本,通过其可连接性保证了电子病历传输过程的完整性与真实性。其次,CL-LRSS 算法结合隐身地址技术将电子病历的接收者地址进行隐藏,防止恶意行为者通过链上分析得到患者的敏感信息,并将隐身地址生成所需的双密钥对简化为单密钥对,从而在保护患者隐私的情况下,减少密钥生成的数量,缓解用户计算和管理密钥的压力。该算法结合智能合约实现电子病历的加密与访问控制,并利用星际文件系统(InterPlanetary File System, IPFS)实现电子病历的链上存储与共享,有效减少存储空间的消耗以及持有者与医疗机构的频繁交互。

1 预备知识

1.1 环签名

环签名属于数字签名的一种,最初由 Rivest 等学者[13]提出,允许一个用户代表一个群组签名消息,并且他人接收签名时无法判断签名者是群组内的哪一名成员。环签名中只有环成员,没有管理者也不需要其余成员的交互,能够实现无条件匿名。在签名过程中,签名者可以随机选择其余成员形成签名者集,通过自身的私钥对消息进行签名。一个有效的环签名满足正确性、无条件匿名性、不可伪造性等安全特性。

1.2 多授权属性加密

基于属性的加密方案存在可信第三方 CA,来负责属性的分发与授权,同时无法对不同类型的属性进行分散管理,只能集中统一地进行一对多加密。因此,Chase[14]提出了多授权属性加密算法(Multi-Authority Attribute-Based Encryption, MA-ABE)。该算法允许多个权威机构分别管理不同的属性信息,并通过多方参与的方式生成用户的私钥,提高了用户密钥的安全性,同时也实现了在分布式管理系统中的访问控制,避免了普通属性基加密算法的单点故障。

1.3 IPFS

IPFS 是一种分布式文件系统和网络协议[15],旨在创建一个全球性、去中心化的文件系统,可以让所有连接到互联网的设备都能够共享和访问同一个文件系统,同时提供更快、更安全、更可靠的文件传输服务。就是基于内容寻址并保证文件内容的不可篡改,通过内容计算出唯一的短哈希值。

2 CL-LRSS 算法

2.1 算法描述

CL-LRSS 算法假设签名者为 Alice,接收者为 Bob。

2.1.1 系统初始化(CL-LRSS_Setup)

输入安全性参数 $\lambda \in N$。KGC 选取素数阶为 p 的循环加法群 G,P 是 G 的生成元。KGC 随机选择主密钥 $M_{sk} \in Z_p^*$,计算系统主公钥 $M_{pk} = M_{sk}P$,并将 $H_1:\{0,1\}^* \times G \times G \rightarrow Z_p^*$、$H_2:\{0,1\}^* \rightarrow G$、$H_3:\{0,1\}^* \rightarrow Z_p^*$ 作为 3 个系统哈希函数。最后公布公共参数 $pp = (G, p, H_1, H_2, H_3, P, M_{pk})$,并且将主私钥 M_{sk} 秘密保存。

2.1.2 部分密钥生成算法(CL-LRSS_PSKGen)

(1)用户输入本人身份 ID_i;

(2)KGC 随机选择 $r_i \in Z_p^*$,并计算 $R_i = r_iP$,$Q_i = H_1(ID_i, R_i, M_{pk})$,$S_i = r_i + M_{sk}\, Q_i \bmod p$;

(3)KGC 将用户的部分私钥 $ppk_i = (S_i, R_i)$ 通过安全渠道发送给用户。

2.1.3 密钥生成算法(CL-LRSS_KeyGen)

用户通过检验 $R_i + Q_i M_{pk} = S_iP$,判断 KGC 发送

的部分密钥正确性。验证完成后，随机选取 $x_i \in Z_p^*$ 作为秘密值。用户私钥最终为 $sk_i = (x_i, S_i)$，公钥为 $pk_i = (x_i + S_i)P$。

2.1.4　隐身地址生成算法（CL-LRSS_SAGen）

若已知 Bob 公钥为 $pk_b = (x_b + S_b)P$，其隐身地址可由 Alice 通过如下方式构造：

（1）Alice 随机选取 $r_{txb} \in Z_p^*$，并计算 $R_{txb} = r_{txb}P, z_b = H_2(r_{txb}\, pk_b)$；

（2）计算出 Bob 的隐身地址为 $SA_b = H_2(z_bP + pk_b)$，将其与随机数 R_{txb} 一起输出。

2.1.5　签名生成算法（CL-LRSS_Sign）

Alice 公私钥对为 (pk_a, sk_a)，作为签名者随机选取一个公钥集 $PK = (pk_1, pk_2, \cdots, pk_a, \cdots, pk_n)$，输入参数集合 $(event, n, PK, sk_a, m)$。其中，$event$ 为事件描述；n 为环的大小；m 为消息。Alice 生成环签名的步骤如下：

（1）计算 $e = H_2(event)$，签名标签即为 $t = x_a e$；

（2）随机选择 $r_x, r_y, c_1, \cdots, c_{a-1}, c_{a+1}, \cdots, c_n \in Z_p^*$，计算 $K = (r_x + r_y) \cdot P + \sum_{i=1, i\neq a}^{n} c_i \cdot pk_i$，$K' = r_x e + \sum_{i=1, i\neq a}^{n} c_i t$；

（3）计算 c_a 使其满足等式 $(c_1 + \cdots + c_n) \bmod p = H_3(PK \| event \| t \| m \| K \| K')$；计算 $\tilde{x} = (r_x - c_a x_a) \bmod p$ 和 $\tilde{y} = (r_y - c_a S_a) \bmod p$；

（4）最终输出签名 $\sigma = \{t, \tilde{x}, \tilde{y}, c_1, \cdots, c_n\}$。

2.1.6　隐身地址验证算法（CL-LRSS_SAVer）

Bob 输入参数集合 (R_{txb}, sk_b, SA_b)，计算隐身地址私钥 $sSA_b = H_2(R_{txb}\, sk_b) + sk_b$，其隐身地址公钥则为 $pSA_b = (H_2(R_{txb}\, sk_b) + sk_b)P$，验证 $SA_b = H_2(pSA_b)$ 是否成立，成立输出 1，否则输出 0。

2.1.7　签名验证算法（CL-LRSS_SVer）

输入参数集合 $(event, n, PK, \sigma, m)$，计算 $e' = H_2(event)$ 和 $C = H_3(PK \| event \| t \| m \| (\tilde{x} + \tilde{y})P + \sum_{i=1}^{n} c_i \cdot pk_i \| \tilde{x}\, e' + \sum_{i=1}^{n} c_i t)$，验证 $\sum_{i=1}^{n} c_i \bmod p = C$ 是否成立，成立输出 1，否则输出 0。

2.1.8　签名可链接性验证算法（CL-LRSS_LVer）

输入参数集合 $(event_1, n_1, PK_1, m_1, \sigma_1)$ 以及 $(event_2, n_2, PK_2, m_2, \sigma_2)$，验证两者签名的有效性，若均为有效签名，且 2 个签名中的 $t_1 = t_2$，那么 2 个签名关联输出 1，否则输出 0。

2.2　安全性分析

本节根据无证书签名方案考虑的 2 类敌手[16]以及椭圆曲线离散对数困难问题[17]给出 CL-LRSS 算法的安全性分析。

2.2.1　正确性

定理 1　CL-LRSS 算法是正确可验证的。

证明

（1）在隐身地址验证阶段，对于隐身地址的合法性需要满足 $SA_b = H_2(pSA_b)$，证明如下：

$$\begin{aligned} H_2(pSA_b) &= H_2((H_2(R_{txb}\, sk_b) + sk_b)P) = \\ &H_2(H_2(R_{txb}\, sk_b)P + sk_bP) = \\ &H_2(H_2(r_{txb}\, sk_bP)P + pk_b) = \\ &H_2(H_2(r_{txb}\, pk_b)P + pk_b) = \\ &SA_b \end{aligned}$$

（2）在签名验证阶段，对于签名的合法性需要满足 $\sum_{i=1}^{n} c_i \bmod p = c_0$，证明如下：

$$\begin{aligned} &(\tilde{x} + \tilde{y})P + \sum_{i=1}^{n} c_i \cdot pk_i = \\ &(r_x - c_a x_a + r_y - c_a S_a)P + \sum_{i=1}^{n} c_i \cdot pk_i = \\ &(r_x + r_y)P - c_a(x_a + S_a)P + \sum_{i=1}^{n} c_i \cdot pk_i = \\ &(r_x + r_y)P - c_a\, pk_a + \sum_{i=1}^{n} c_i \cdot pk_i = \\ &(r_x + r_y)P + \sum_{i=1, i\neq a}^{n} c_i \cdot pk_i = K; \end{aligned}$$

$$\begin{aligned} &\tilde{x}\, e' + \sum_{i=1}^{n} c_i t = (r_x - c_a x_a) H_2(event) + \sum_{i=1}^{n} c_i x_a e = \\ &(r_x - c_a x_a)e + \sum_{i=1}^{n} c_i x_a e = \\ &r_x e + x_a e \sum_{i=1, i\neq a}^{n} c_i = \\ &r_x e + \sum_{i=1, i\neq a}^{n} c_i t = K' \end{aligned}$$

因此，

$$\begin{aligned} C &= H_3(PK \| event \| t \| m \| (\tilde{x} + \tilde{y})P + \\ &\sum_{i=1}^{n} c_i \cdot pk_i \| \tilde{x}\, e' + \sum_{i=1}^{n} c_i t) = \\ &H_3(PK \| event \| t \| m \| K \| K') = \\ &\sum_{i=1}^{n} c_i \bmod p \end{aligned}$$

2.2.2　匿名性

定理 2　CL-LRSS 算法满足匿名性。

证明　假设存在挑战者 B 通过运行系统初始化算法得到公共参数 $pp = (G, p, H_1, H_2, H_3, P, M_{pk})$，并将其发送给敌手 A。对于任一有效环签

名,已知 H_2 是一个映射在 G 上的安全的哈希函数,那么 $e = H_2(event)$ 在 G 上是均匀分布的,因此 $t = x_a e$ 在 G 上也是均匀分布的。同样,由于签名者所选取的 $r_x, r_y, c_i \in Z_p^*(i = 1,2,\cdots,n, i \neq a)$ 是独立选择且均匀分布在 Z_p^* 上的,因此 $K = (r_x + r_y) \cdot P + \sum_{i=1,i\neq a}^{n} c_i \cdot pk_i$ 和 $K' = r_x e + \sum_{i=1,i\neq a}^{n} c_i t$ 分别在 G 和 Z_p^* 上是均匀分布的。由于 $H_3(PK \| event \| t \| m \| K \| K')$ 是映射在 Z_p^* 上的安全哈希函数 H_3 的输出,因此 $c_a = H_3(PK \| event \| t \| m \| K \| K') - \sum_{i=1,i\neq a}^{n} c_i$ 以及 $\tilde{x} = (r_x - c_a x_a) \bmod p$ 和 $\tilde{y} = (r_y - c_a S_a) \bmod p$ 都是均匀分布在 Z_p^* 上的。

综上,以上提到的所有参数都是独立且均匀分布的,与真实签名者的身份无关,因此相对于随机猜测敌手 A 在识别真实签名者方面没有优势,且猜中概率不超过 $\frac{1}{n}$,满足匿名性。

2.2.3 不可伪造性

定理 3 在随机预言机模型下,如果椭圆曲线离散对数问题是困难的,那么 CL-LRSS 算法对于第一类攻击者 A_I 具有不可伪造性。

证明 假设第一类攻击者 A_I 能够以不可忽略的优势 ε_1 攻破提出的方案,构建一个挑战者 B 能够在多项式时间内利用 A_I 来解决椭圆曲线离散对数问题。

(1)初始化阶段:挑战者 B 通过运行系统初始化算法得到公共参数 $pp = (G, p, H_1, H_2, H_3, P, M_{pk})$,并将其发送给第一类攻击者 A_I。

(2)询问阶段:在该阶段中,A_I 会向挑战者 B 发起一系列的问题查询,B 则会自适应地回答并将查询结果与答案存储进相关表中。

(3) H_1 查询:B 维护一个关于 (ID_i, R_i, M_{pk}, Q_i) 元组的表 L_1,当 A_I 发起关于 $H_1(ID_i, R_i, M_{pk})$ 的查询时,如果表 L_1 存在这一元组数据,那么 B 返回 Q_i,如果不存在,B 随机选择 $Q_i \in Z_p^*$,并将其返回给 A_I,令 $H_1(ID_i, R_i, M_{pk}) = Q_i$ 并在表 L_1 中插入 (ID_i, R_i, M_{pk}, Q_i)。

(4) H_2 查询:B 维护一个关于 (α_i, e_i) 元组的表 L_2,当 A_I 发起关于 $H_2(\alpha_i)$ 的查询时,如果表 L_2 中存在这一元组数据,那么 B 返回 e_i。否则 B 随机选择 $e_i \in G$ 并将其返回给 A_I,令 $H_2(\alpha_i) = e_i$ 并在表 L_2 中插入 (α_i, e_i)。

(5) H_3 查询:B 维护一个关于 (β_i, C_i) 元组的表 L_3,当 A_I 发起关于 $H_3(\beta_i)$ 的查询时,如果表 L_3 中存在这一元组数据,那么 B 返回 C_i。否则 B 随机选择 $C_i \in Z_p^*$ 并将其返回给 A_I,令 $H_3(\beta_i) = C_i$ 并在表 L_3 中插入 (β_i, C_i)。

(6)部分密钥查询:B 维护一个关于 (ID_i, R_i, S_i) 元组的表 L_{psk},当 A_I 发起关于身份 ID_i 的部分密钥查询时,如果 $ID_i = ID^*$,其中 ID^* 代表目标身份标识,那么挑战者 B 失败终止并输出 $\perp$,其中 $\perp$ 表示未知。否则如果表 L_{psk} 中存在元组,则返回对应值,若不存在,B 随机选取 $s_i, q_i \in Z_p^*$,并令 $S_i = s_i$, $H_1(ID_i, R_i, M_{pk}) = q_i$, $R_i = s_i P - q_i M_{pk}$,随后 B 将 (R_i, S_i) 返回给 A_I 并在表 L_{psk} 中插入 (ID_i, R_i, S_i),在表 L_1 中插入 (ID_i, R_i, M_{pk}, q_i)。

(7)公钥查询:B 维护一个关于 (ID_i, x_i, S_i, pk_i) 元组的表 L_{pk},当 A_I 发起关于身份 ID_i 的公钥查询时,B 首先会搜索表 L_{pk},若存在相关元组,则输出对应 pk_i。否则 B 会进行如下操作:

① 如果 $ID_i = ID^*$,那么 B 会随机选择 $k_i, q_i, x_i \in Z_p^*$,并计算 $R_i = k_i P$, $q_i = H_1(ID_i, R_i, M_{pk})$, $S_i P = R_i + q_i M_{pk} \bmod p$,从而得到公钥 $pk_i = (x_i + S_i)P$。随后 B 将 $(ID_i, x_i, \perp, pk_i)$ 插入表 L_{pk}。

② 如果 $ID_i \neq ID^*$,那么 B 会从表 L_{psk} 检索 (ID_i, R_i, S_i),并随机选择 $x_i \in Z_p^*$,输出公钥 $pk_i = (x_i + S_i)P$ 并返回给 A_I,同时在表 L_{psk} 中插入 (ID_i, x_i, S_i, pk_i)。

(8)秘密值查询:当 A_I 发起关于身份 ID_i 的秘密值查询时,如果 $ID_i = ID^*$,那么 B 失败终止并输出 $\perp$。否则 B 检索表 L_{pk},如果 $x_i \neq \perp$ 则输出 x_i,否则 B 执行公钥查询并将 (ID_i, x_i, S_i, pk_i) 插入 L_{pk},返回 x_i 值。

(9)公钥替换查询:当 A_I 想要将用户 ID_i 的公钥进行替换时,B 会检索表 L_{pk} 中的 (ID_i, pk_i) 的元组并将其替换为 (ID_i, pk_i')。

(10)签名查询:当 A_I 发起对于输入参数集合 $(event, n, PK, sk_a, m_i)$ 的签名查询时,如果 $ID_a \neq ID^*$ 且 $ID_a \notin L_{rpk}$,那么 B 会通过执行签名算法输出一个有效签名 σ。否则,B 将进行下列操作:

① 若 $H_2(event)$ 未赋值则对事件描述 $event$ 进行 H_2 查询,若表 L_2 中存在则输出 $e = H_2(event)$,并计算得到 $t = x_a e$。

② 随机选取 $\tilde{x}, \tilde{y}, c_i \in Z_p^*, i \in (1,2,\cdots,n)$,计算 $H_3(PK \| event \| t \| m \| (\tilde{x} + \tilde{y})P + \sum_{i=1}^{n} c_i \cdot pk_i \| \tilde{x} e' +$

$\sum_{i=1}^{n} c_i t$) 以及 $C = \sum_{i=1}^{n} c_i \bmod p$，判断前者 H_3 值与 C 是否相等。如果发生了碰撞，说明 C 的值已经被添加进 L_3 表中，那么重复进行这一步。

③ 输出有效签名 $\sigma = \{t, \tilde{x}, \tilde{y}, c_1, \cdots, c_n\}$。

(11) 伪造阶段：A_I 最终可以得到一个伪造的有效的签名 $\sigma^* = \{t^*, \tilde{x}^*, \tilde{y}^*, c_1^*, \cdots, c_n^*\}$。为了得到有效签名，$A_I$ 需要进行 H_3 查询，此处假设 A_I 在第 j 次 H_3 查询时得到了一个返回值 C^*。由于已知 H_3 是一个安全的哈希函数，而此处 $C^* = c_1^* + \cdots + c_n^*$，那么至少存在一个 $a \in \{1,2,\cdots,n\}$ 使得 c_a^* 是确定的。根据分叉引理[18]，如果 A_I 能够在多项式时间 T 内以不可忽略的优势 $\varepsilon_1 \geqslant \frac{7 P_{q_{H_3}}^n}{2^\lambda}$ 伪造出一个有效的签名，其中 λ 为系统输入的安全参数，$P_{q_{H_3}}^n$ 表示 q_{H_3} 个元素中取出 n 个元素的排列数，那么通过在时间 $T' \leqslant \frac{16 P_{q_{H_3}}^n T}{\varepsilon_1}$ 内重新定位概率多项式时间图灵机的哈希，A_I 可生成 2 个有效的签名。因此 A_I 可得到另一个有效的伪造签名 $\sigma^{*'} = \{t^{*'}, \tilde{x}^{*'}, \tilde{y}^{*'}, c_1^{*'}, \cdots, c_n^{*'}\}$，其中 $C^{*'} = c_1^{*'} + \cdots + c_n^{*'}$，且 $C^* \neq C^{*'}$。同样地，存在一个 $c_a^{*'}$ 能够由 $C^{*'}$ 确定。由于 $c_a^{*'} \neq c_a^*$ 且 $c_i^{*'} = c_i^*, i \neq a$，因此 $\tilde{x}^* \neq \tilde{x}^{*'}, \tilde{y}^* \neq \tilde{y}^{*'}$。如果 $ID_a^* = ID^*$，那么 $\tilde{x}^* + \tilde{y}^* = r_x^* + r_y^* - c_a^*(x_a + S_a)$，$\tilde{x}^{*'} + \tilde{y}^{*'} = r_x^* + r_y^* - c_a^{*'}(x_a + S_a)$。由这 2 个等式可计算出 $(x_a + S_a) = (c_a^* - c_a^{*'})^{-1}[(\tilde{x}^{*'} + \tilde{y}^{*'}) - (\tilde{x}^* + \tilde{y}^*)]$，即解决了 $pk_a = (x_a + S_a)P$ 这一困难问题。

(12) 概率分析：令 $q_{H_i}(i = 1,2,3)$，q_{psk}，q_s，q_{pk}，q_{rpk}，q_{sign} 分别表示 $H_i(i = 1,2,3)$ 查询、部分密钥查询、秘密值查询、公钥查询、公钥替换查询、签名查询的最大查询次数。按如下定义 3 个事件：

① E_1：A_I 没有针对 ID^* 进行部分密钥查询。

② E_2：ID^* 在用户列表中且为真实签名者。

③ E_3：A_I 成功伪造出有效签名。

那么，$\Pr[\text{成功}] = \Pr[E_1] \cdot \Pr[E_2 \mid E_1] \cdot \Pr[E_3 \mid E_1 \wedge E_2] = (1 - \frac{1}{q_{pk}})^{q_{psk}} \cdot \frac{1}{q_{pk}} \cdot \varepsilon_1$。

综上所述，如果 A_I 能够以不可忽略的优势 ε_1 成功伪造，则 B 能以至少为 $\Pr[\text{成功}]$ 的优势解决椭圆曲线离散对数问题，但这与椭圆曲线离散对数问题的困难假设相矛盾，所以 CL-LRSS 算法在随机预言机模型中是安全的，对于第一类攻击者 A_I 具有不可伪造性。

定理 4　在随机预言机模型下，如果椭圆曲线离散对数问题是困难的，那么 CL-LRSS 算法对于第二类攻击者 A_{II} 具有不可伪造性。

证明过程与定理 3 相同，限于篇幅，在本次研究中不再赘述。

2.2.4　可链接性

在说明 CL-LRSS 算法是可链接的之前，先证明以下引理。

引理 1　在随机预言机模型下，如果椭圆曲线离散对数问题是困难的，在敌手 A 只知道一个私钥 $sk_a = (x_a, S_a)(a \in [1,n])$ 的情况下，能够为事件 $event$ 生成一个有效的环签名 $\sigma = \{t, \tilde{x}, \tilde{y}, c_1, \cdots, c_n)$，并得到 $t = x_a H_2(event)$。

证明　挑战者 B 将一系列公共参数给到敌手 A，敌手 A 执行如定理 3 中所描述的各种查询，然后敌手 A 生成一个有效的环签名 $\sigma = \{t, \tilde{x}, \tilde{y}, c_1, \cdots, c_n\}$，其中 $t = x_\tau H_2(event), x_\tau \in Z_p^*$。接着挑战者 B 通过倒带敌手 A 相同的输入从 H_3 查询中得到另一个签名 $\sigma' = \{t', \tilde{x}', \tilde{y}', c_1', \cdots, c_n'\}$，其中 2 次运行过程中的公钥集 PK，事件描述 $event$，消息 m，私钥 $(x_\tau, S_\tau), S_\tau \in Z_p^*$ 以及 K, K' 都是一样的。由此 B 可以得到 2 个不同的值 u, u'，同样地，也能得知存在 $c_a, c_a', a \in [1,n]$ 能够在得到 u, u' 后分别计算得出。因为 $c_a \neq c_a'$ 且 $c_i = c_i', i \neq a$，所以 $\tilde{x} = (r_x - c_a x_\tau) \neq \tilde{x}' = (r_x - c_a' x_\tau), \tilde{y} = (r_y - c_a S_\tau) \neq \tilde{y}' = (r_y - c_a' S_\tau)$。$B$ 可以计算出 $x_\tau = (c_a - c_a')^{-1}(\tilde{x}' - \tilde{x})$，$S_\tau = (c_a - c_a')^{-1}(\tilde{y}' - \tilde{y})$。根据定理 3 可知，敌手 A 无法伪造有效环签名，同时可由 $t = x_\tau H_2(event) = x_a H_2(event)$ 得知 $x_\tau = x_a$，所以敌手 A 必须知道一个私钥 (x_a, S_a)。

定理 5　在随机预言机模型下，如果椭圆曲线离散对数问题是困难的，且敌手 A 只通过一个签名者私钥无法生成 2 个有效的且不链接的环签名，那么 CL-LRSS 算法是可链接的。

证明　如果敌手 A 能够产生 2 个不链接的有效环签名，那就意味着 2 个环签名的签名标签 $t_1 \neq t_2$，其中 $t_1 = x_{a1} H_2(event), t_2 = x_{a2} H_2(event)$。由引理 1 可知，此时敌手 A 一定知道私钥 (x_{a1}, S_{a1}) 和 (x_{a2}, S_{a2})，这与前提假设敌手 A 只知道一个私钥相矛盾，所以 CL-LRSS 算法是可链接的。

2.2.5 隐身地址的安全性

定理 6 在随机预言机模型下,如果椭圆曲线离散对数问题是困难的,那么 CL-LRSS 算法中的隐身地址具有安全性,即除接收方外其余任何角色都无法计算出隐身地址的私钥。

证明 假设敌手 A 能够以不可忽略的优势攻破提出的方案,则构建挑战者 B 能够利用 A 解决椭圆曲线离散对数问题。B 通过运行系统初始化算法得到公共参数 pp 并将其发送给敌手 A,A 适应性的查询见定理 3 以及以下描述的各个预言机。

(1)隐身地址查询:当 A 以 pk_i 为接收方地址使用此预言机时,B 会运行隐身地址生成算法生成此次 pk_i 的隐身地址与随机数并向 A 返回 (SA_i, R_{txi})。

(2)区块链查询:当 A 查询此预言机时,B 会访问区块链中的公开信息获取公开的所有隐身地址集 $\{(SA_i, R_{txi})\}$ 并返回给敌手 A。

假设 A 能够以不可忽略的优势得到目标隐身地址的私钥,那么 B 就可以通过 $sSA_i = H_2(R_{txi}\ sk_i) + sk_i = H_2(r_{txi}\ pk_i) + sk_i$ 计算出 $sk_i = sSA_i - H_2(r_{txi}\ pk_i)$,已知 $R_{txi} = r_{txi}P$ 求 r_{txi},由于前提假设 B 拥有解决此困难问题的能力,因此 B 能够以同样的优势解决这一问题,并能够同样以不可忽略的优势解决 $pk_i = sk_iP$ 这一困难问题。但这与假设椭圆曲线离散对数问题是困难的相矛盾,因此 CL-LRSS 算法中的隐身地址具有安全性。

2.2.6 隐身地址的匿名性

定理 7 在随机预言机模型下,如果椭圆曲线离散对数问题是困难的,那么 CL-LRSS 算法中的隐身地址具有匿名性。

证明 假设敌手 A 能够以不可忽略的优势攻破提出的方案,则构建挑战者 B 能够利用 A 解决椭圆曲线离散对数问题。B 将一系列公共参数给到敌手 A,A 适应性的查询见定理 3 以及定理 6 中描述的各个预言机。

敌手 A 可以通过区块链查询预言机得到隐身地址及其随机数,已知随机数 $R_{txi} = r_{txi}P$,隐身地址 $SA_i = H_2(H_2(r_{txi}\ pk_i)P + pk_i)$,$B$ 通过 H_2 查询预言机置 $\alpha_i = H_2(r_{txi}\ pk_i)P + pk_i$,假设 A 能够以不可忽略的优势得到对应的 pk_i,那么 B 能够计算出 $pk_i = \alpha_i - H_2(r_{txi}\ pk_i)P = \alpha_i - H_2(R_{txi}\ sk_i)P$,根据定理 6 可知,在椭圆曲线离散对数问题是困难的情况下,敌手 A 无法得到目标的私钥,因此 CL-LRSS 算法中的隐身地址具有匿名性。

3 区块链电子病历方案

基于 CL-LRSS 的区块链电子病历方案模型如图 1 所示。整个方案存在医生、患者、区块链、KGC、属性权威机构(Attribute Authority, AA)、IPFS 网络、数据访问者七种角色。整个方案分为 4 步完成,即系统初始化、生成电子病历、上传电子病历和访问电子病历。

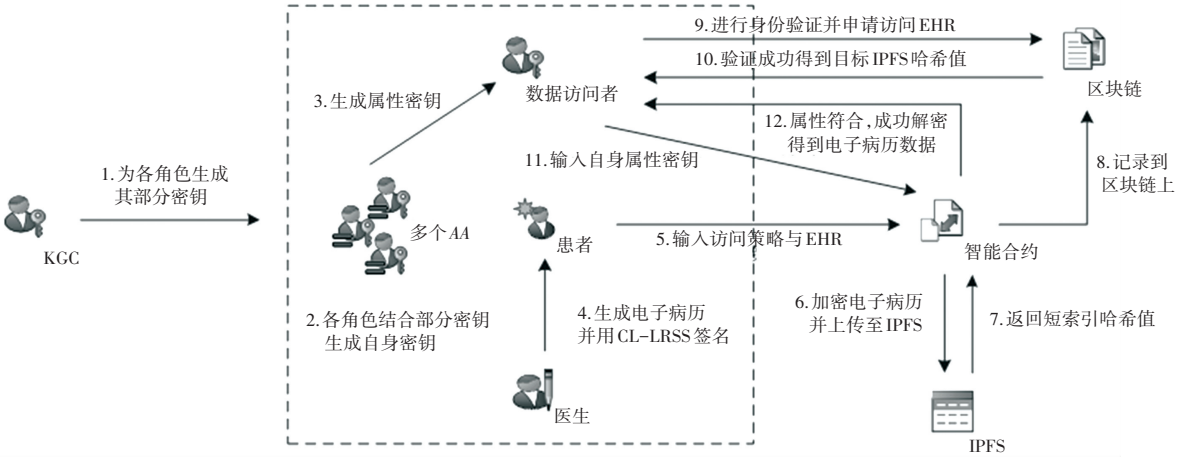

图 1 基于 CL-LRSS 的区块链电子病历方案框架图

Fig. 1 Frame diagram of block chain electronic medical record scheme based on CL-LRSS

3.1 系统初始化

首先,KGC 根据联盟链各个管理员(可由各个医疗机构、研究所等充当)共同决议的安全参数 λ,生成系统公共参数 pp 并将其公布。接着,所有 AA、医生和数据访问者从 KGC 处安全地获取各自的部分密钥并自主生成自己的公私钥,每个 $AA_i(i > 1)$ 发布自己的公钥 AA_{pki} 并组合成系统公共属性参数 AA_{pk},医生的公钥会被公布。每个用户拥有的身份

ID 唯一且各不相同，他人无法通过身份 *ID* 判断目标是何种角色。患者则是在医院注册挂号时，通过如病历卡等磁卡提交身份 *ID*，KGC 以此安全地将患者的部分私钥返回给患者，患者随后可自主生成自己的公私钥并选择合适的医生进行治疗。最后，数据访问者 j 为了访问电子病历需向多个 *AA* 申请属性私钥，通过输入公共参数 pp、用户身份 ID_j 和用户属性集 AS_j、接到申请的 AA_i，结合这些参数使用自身的私钥 AA_{ski} 输出该用户 j 的部分属性私钥 PA_{skj}，用户 j 结合获得到的所有部分属性私钥计算出自己的属性私钥 A_{skj}。

3.2　生成电子病历

诊治医生检查完患者后，先根据患者的公钥 pk_p 通过 CL-LRSS_SAGen 算法计算出患者的隐身地址 SA_p，然后对生成的 EHR 进行如下操作：

（1）负责签名的诊治医生 a，其公私钥为（pk_a，sk_a），区块链系统会随机为其分配 $n-1$ 个授权公钥，并组合成公钥集 $PK=\{pk_1,\cdots,pk_a,\cdots,pk_n\}$。

（2）诊治医生 a 随机生成随机数，并根据 CL-LRSS_Sign 算法计算相应参数，最终生成环签名 σ。

（3）医生 a 使用患者的公钥 pk_p 对电子病历和签名组合成的消息 $M=(m,\sigma)$ 进行加密，并将其发送到患者的隐身地址。

3.3　上传电子病历

患者通过 CL-LRSS_SAVer 算法验证隐身地址是否属于自己，以此判断该隐身地址所属的交易信息是否是自己的电子病历。如果验证通过，患者会用自己的私钥解密信息得到消息 $M=(m,\sigma)$，通过运行 CL-LRSS_SVer 算法验证该环签名的合法性。如果签名合法且本次是首次进行检查治疗，则患者会使用其隐身地址执行智能合约 EHR 上传（如图 2 所示）。通过输入系统公共属性参数 AA_{pk}，待加密电子病历 m 以及访问策略 *policy* 对其电子病历信息进行加密，并自动上传至 IPFS 服务器，IPFS 会返回对应信息的唯一短哈希值 *cid*。最后通过智能合约中的事件功能，当定义的事件被触发时，事件数据就会被记录在区块链中，即将返回的唯一短哈希值以及相关的隐身地址通过链上共识验证存储至区块链中。

算法 1　EHR 上传

```
1.  contract ERR{
2.    //定义事件,用于记录电子病历的上传事件
3.    event EHRupload(address indexed patientAddr,string indexed cid);
4.  function uploadEHR(AAPK,string memory m, string memory policy)public view returns(bytes memory){
5.      //将待加密数据以及访问策略编码为字节数组
6.      bytes memory dataBytes=bytes(m)
7.      bytes memory policyBytes-bytes(policy);
8.      //使用 OpenABE 库进行加密
9.      bytes memory ciphertext=openabe.encrypt(dataBytes,policyBytes,AAPK);
10.     //将密文上传 IPFS
11.     string memory cid=ipfs.upload(eiphertext);
12.     //触发事件,将发起方地址和相应返回的 IPFS 哈希值记录到区块链上
13.     emit EHRupload(msg.sender,cid);
14.   }
15. }
```

图 2　EHR 上传算法

Fig. 2　EHR upload algorithm

如果签名合法但本次治疗会对之前相应的电子病历进行更新，那么患者接收到的消息会包含之前发送的电子病历信息及其签名以及本次发送的新电子病历信息和新签名，患者可以通过 CL-LRSS_LVer 算法验证两者的签名标签是否一致，从而判断电子病历的真实性与完整性，之后执行相同的操作将唯一短哈希值以及相关的隐身地址通过链上共识验证存储至区块链中。

3.4　访问电子病历

数据访问者想要对目标电子病历进行访问时，需要向区块链提交访问请求验证其是否有权访问他人电子病历，并执行智能合约访问 EHR（如图 3 所示）。区块链只需将发起方地址与存储的授权地址进行比对即可判断访问者是否拥有权限。验证通过后，将自己的属性私钥 A_{sk} 与待访问的电子病历的 IPFS 哈希值作为输入参数，智能合约会自动运行相

应操作;如果访问者的属性私钥与密文中的访问策略相匹配,那么智能合约会成功返回解密后的电子病历信息并将此次访问 EHR 事件记录在区块链中。

算法 2 访问 EHR

```
1.  contract EHR{
2.    //定义事件,用于记录电子病历的访问事件
3.     event EHRupaccess(address indexed visitAddr,string indexed cid);
4.    function visitEHR(ASK,string memory cid) public view returns (bytes memory){
5.      //身份验证
6.          require (authorizedRole [msg.sender]," Permission denied");
7.      //将 IPFS 哈希值编码为字节数组
8.      bytes memory ipfshash=bytes(cid);
9.      //从 IPFS 上获取加密的电子病历
10.       bytes memory ciphertext=ipfs.download(ipfshash);
11.       //使用 OpenABE 库进行解密
12.       bytes memory decrypted=openabe.decrypt(cipertext,ASK);
13.       //将解密后的字节数解码为字符串
14.       string memory plaintext=string(decrypted);
15.       return plaintext;
16.       //触发事件,将发起方地址和访问的 IPFS 哈希值记录到区块链上
17.       emit EHRaccess(msg.sender,cid);
18.     }
19.  }
```

图 3 EHR 访问算法

Fig. 3 EHR access algorithm

4 实验分析

为了判断 CL-LRSS 算法性能,又进行了相关实验,实验环境为 64 位 Windows 10 以及搭载了 Ubuntu 20.04(64 位)虚拟机的笔记本。

首先在 CL-LRSS 算法中的签名生成与验证时间在环成员个数变化情况下进行了实验。接着对隐身地址生成与验证所需要的时间进行了测试,最后将 CL-LRSS 算法与同类型算法进行理论对比分析。

4.1 签名生成与验证时间测试

根据 CL-LRSS 算法定义进行了代码实验与测试。签名生成与验证时间如图 4 所示。由图 4 可知,环成员个数从 5 增加到 25 的过程中,CL-LRSS 签名生成与验证的时间也随之增加,即生成时间与验证时间会随着环成员个数的增加呈线性增长。在实际的电子病历系统中,环成员个数基本不会太多,即使环成员个数达到了 25 个,签名生成时间与验证时间仍然能够接受,环签名生成时间与环签名验证时间约为 150 ms。

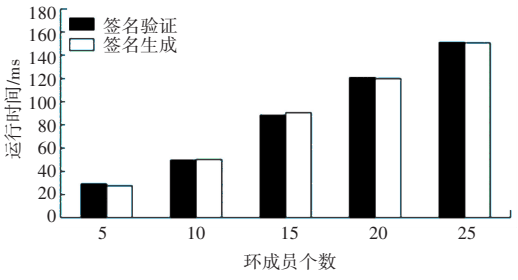

图 4 签名生成与验证时间

Fig. 4 Signature generation and verification time

4.2 隐身地址生成与验证时间测试

方案需要医生生成患者的隐身地址,患者需要从隐身地址中获取需要的 EHR,因此对隐身地址生成与验证所需要的时间进行测试,判断其是否会对整体方案有所影响。针对隐身地址的生成与验证,执行 100 次后所得平均时间如图 5 所示。运行结果表明,隐身地址的生成时间是毫秒级别、且接近 1.2 ms,验证时间更是达到了微秒级别、近似 674 μs,因此在实际应用中对于隐身地址的生成与验证可忽略不计。

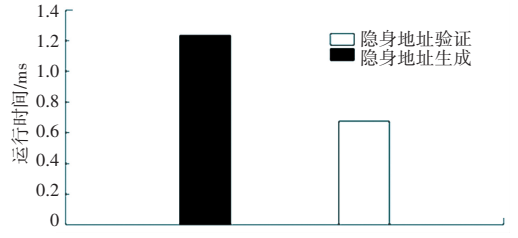

图 5 隐身地址生成与验证时间

Fig. 5 Stealth address generation and verification time

4.3 理论对比分析

将 CL-LRSS 算法与 Lai 等学者[19]所提的环签名电子病历方案进行比较,对比结果见表 1。其中,n 表示环签名的环成员个数,T_{bp} 表示双线性对运算,T_M 表示在阶为 160 字节的加法群 G 上的标量乘法运算,T_E 表示在 G_1 或 G_T 上的指数运算(G_1 为定义在有限域 F_p 上阶为 512 字节的加法群,p 为 160 字节,G_T 为乘法群,双线性映射为:$G_1 \times G_1 \rightarrow G_T$),$T_{M1}$ 表示在 G_1 上的标量乘法运算。

根据算法中的关键运算进行分析,已知双线性对运算比在椭圆曲线上的标量乘法运算的时间要多 10 倍[20],即 T_{bp} 所需要的时间要远大于 T_M,T_{M1} 所需要的时间约为 T_M 的 4 倍。因此,CL-LRSS 算法的性能优于文献[19],同时 CL-LRSS 算法的运算时间均在毫秒级,完全能够应用于电子病历场景中。

表 1　CL-LRSS 算法与其他环签名方案对比

Table 1　Comparison of CL-LRSS scheme with other ring signature schemes

方案	生成签名	验证签名
文献[19]	$(4n+6)T_E+(4n-1)T_{M1}$	$(2n-1)T_{M1}+nT_E+2T_{bp}$
CL-LRSS	$(2n+3)T_M$	$(2n+3)T_M$

实验证明，本文的算法与方案是可行的，运行存在的时间开销都是可以接受的，用户负责计算的部分所需要的时间更是可以忽略不计。同时 CL-LRSS 算法对于其他环签名方案来说，所需要的理论计算时间也更少，效率更高。

5　结束语

本文提出了一个结合隐身地址的无证书可链接环签名算法 CL-LRSS，该算法通过结合隐身地址切断了链上聚类发现链上与现实之间联系的可能。使用无证书签名体系，弱化密钥分发中心的同时解决密钥托管问题，也更便于部署在资源受限的设备上。避免了双线性对映射的使用，使运算效率更高。同时，基于此算法设计了区块链电子病历方案，结合智能合约与 IPFS 实现对电子病历的加密与存储，并将最终的哈希值记录在区块链上使其不可篡改以及可追溯。最后，通过安全性分析论证，该方案具有正确性、匿名性、不可伪造性等安全特性，并通过模拟实验验证 CL-LRSS 算法整体运行的效率有显著改进。

在未来的研究中，将会针对加密信息与访问控制进行进一步的优化，减少其在加密过程中冗杂的运算，并且尝试引入可搜索概念进行更加细粒度的访问控制，从而提升整体方案的可扩展性与准确性。

参考文献

[1] BLUMENTHAL D, TAVENNER M. The "mean-ingful use" regulation for electronic health records[J]. New England Journal of Medicine, 2010, 363(6): 501-504.

[2] BHUYAN S S, KABIR U, ESCARENO J M, et al. Transforminghealthcare cybersecurity from reactive to proactive: Current status and future recommendations[J]. Journal of Medical Systems, 2020, 44: 1-9.

[3] 白枫. 基于区块链技术的医院电子病历系统的设计与实现[J]. 无线互联科技, 2022, 19(14): 53-55.

[4] 袁勇, 王飞跃. 区块链技术发展现状与展望[J]. 自动化学报, 2016, 42(4): 481-494.

[5] 程泽川. 基于区块链的电子病历可信管理技术的研究[D]. 南京:南京邮电大学, 2022.

[6] GAIHRE A, LUO Yan, LIU Hang. Do bitcoin users really care about anonymity? an analysis of the bitcoin transaction graph[C]//2018 IEEE International Conference on Big Data (big data). Seattle, WA, USA: IEEE, 2018: 1198-1207.

[7] 薛腾飞, 傅群超, 王枞, 等. 基于区块链的医疗数据共享模型研究[J]. 自动化学报, 2017, 43(9): 1555-1562.

[8] DAGHER G G, MOHLER J, MILOJKOVIC M, et al. Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology[J]. Sustainable Cities and Society, 2018, 39: 283-297.

[9] LI Fengqi, LIU Kemeng, ZHANG Lupeng, et al. EHRChain: A blockchain-based EHR system using attribute-based and homomorphic cryptosystem[J]. IEEE Transactions on Services Computing, 2021, 5: 2755-2765.

[10] WATER B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization[C]// International Workshop on Public Key Cryptography. Berlin/Heidelberg: Springer, 2011: 53-70.

[11] WANG Baocheng, LI Zetao. Healthchain: A privacy protection system for medical data based on blockchain[J]. Future Internet, 2021, 13(10): 247.

[12] CHAUM D, VAN H E. Group signatures[C]//Advances in Cryptology—EUROCRYPT′91: Workshop on the Theory and Application of Cryptographic Technique. Berlin/Heidelberg: Springer, 1991,547: 257-265.

[13] RIVEST R L, SHAMIR A, TAUMAN Y. How to leak a secret[C]//Advances in Cryptology—ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security. Berlin/Heidelberg: Springer, 2001: 552-565.

[14] CHASE M. Multi-authority attribute based encryption[C]// Theory of Cryptography: 4th Theory of Cryptography Conference. Berlin/Heidelberg: Springer, 2007: 515-534.

[15] STEICHEN M, FIZ B, NORVILL R, et al. Blockchain-based, decentralized access control for IPFS[C]//2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). Halifax, NS, Canada: IEEE, 2018: 1499-1506.

[16] AL-RIYAMI SS, PATERSON K G. Certificateless public key cryptography[C]//Advances in Cryptology (ASIACRYPT). Berlin: Springer-Verlag, 2003, 2894: 452-473.

[17] 刘帅, 陈建华. 无双线性对的无证书签名方案及其在配电网中的应用[J]. 计算机科学, 2020, 47(9): 304-310.

[18] HERRANZ J, SÁEZ G. Forking lemmas for ring signature schemes[C]//Progress in Cryptology-INDOCRYPT 2003: 4th International Conference on Cryptology. Berlin/Heidelberg: Springer, 2003: 266-279.

[19] LAI Chengzhe, MA Zhe, GUO Rui, et al. Secure medical data sharing scheme based on traceable ring signature and blockchain[J]. Peer-to-Peer Networking and Applications, 2022, 15(3): 1562-1576.

[20] HE Debiao, WANG Huaqun, WANG Lina, et al. Efficient certificateless anonymous multi-receiver encryption scheme for mobile devices[J]. Soft Computing, 2017, 21: 6801-6810.