软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2023,34(5):2465−2481 [doi: 10.13328/j.cnki.jos.006396]

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

可验证的属性基定时签名方案及其应用*

侯慧莹[1], 宁建廷[2,3,4], 黄欣沂[2,3], 赵运磊[1,5]

[1](复旦大学 计算机科学技术学院, 上海 200433)

[2](福建省网络安全与密码技术重点实验室 (福建师范大学), 福建 福州 350007)

[3](福建师范大学 计算机与网络空间安全学院, 福建 福州 350007)

[4](信息安全国家重点实验室 (中国科学院 信息工程研究所), 北京 100093)

[5](综合业务网络国家重点实验室 (西安电子科技大学), 陕西 西安 710071)

通信作者: 赵运磊, Email: ylzhao@fudan.edu.cn

摘　要: 可验证定时签名 (VTS) 方案允许在给定的时间内对已知消息上的签名进行锁定, 在执行时间为 T 的顺序计算后, 任何人都可从时间锁 (time-lock) 中提取出该签名. 可验证性保证了在无需解开时间锁的情况下, 任何人都可以公开地验证时间锁中是否包含已知消息上的合理签名, 且可以在执行时间 T 的顺序计算后获得该签名. 提出了可验证的属性基定时签名 (verifiable attribute-based timed signatures, VABTS) 概念, 并给出了一个可撤销和可追溯的 VABTS 方案 (RT-VABTS) 的具体构造. RT-VABTS 方案可同时支持签名者身份隐私保护、动态的用户撤销、可追溯性和定时性, 并能解决属性基密码中的密钥托管问题. VABTS 具有非常广阔的应用前景, 特别列举了 VABTS 的两种应用场景: 构建准入区块链中隐私保护的支付通道网络和实现公平的隐私多方计算. 最后, 通过形式化的安全性分析和性能评估证明实例化的 RT-VABTS 方案是安全且高效的.

关键词: 定时签名; 属性基签名; 隐私保护; 可追溯性; 支付通道网络

中图法分类号: TP309

中文引用格式: 侯慧莹, 宁建廷, 黄欣沂, 赵运磊. 可验证的属性基定时签名方案及其应用. 软件学报, 2023, 34(5): 2465–2481. http://www.jos.org.cn/1000-9825/6396.htm

英文引用格式: Hou HY, Ning JT, Huang XY, Zhao YL. Verifiable Attribute-based Timed Signatures and Its Applications. Ruan Jian Xue Bao/Journal of Software, 2023, 34(5): 2465–2481 (in Chinese). http://www.jos.org.cn/1000-9825/6396.htm

Verifiable Attribute-based Timed Signatures and Its Applications

HOU Hui-Ying[1], NING Jian-Ting[2,3,4], HUANG Xin-Yi[2,3], ZHAO Yun-Lei[1,5]

[1](School of Computer Science and Technology, Fudan University, Shanghai 200433, China)

[2](Fujian Provincial Key Laboratory of Network Security and Cryptology (Fujian Normal University), Fujian 350007, China)

[3](College of Computer and Cyberspace Security, Fujian Normal University, Fujian 350007, China)

[4](State Key Laboratory of Information Security (Institute of Information Engineering, Chinese Academy of Sciences), Beijing 100093, China)

[5](State Key Laboratory of Integrated Services Networks (Xidian University), Xi'an 710071, China)

Abstract: A verifiable timed signature (VTS) scheme allows one to time-lock a signature on a known message for a given amount of time T such that after performing a sequential computation for time T anyone can extract the signature from the time-lock. Verifiability ensures that anyone can publicly check if a time-lock contains a valid signature on the message without solving it first, and that the

* 基金项目: 国家自然科学基金 (U1536205, 61472084, 61972094, 62032005); 国家重点研发计划 (2017YFB0802000); 上海市创新行动计划 (16DZ1100200); 上海市科学技术发展基金 (16JC1400801); 山东省重点研发计划 (2017CXGC0701, 2018CXGC0701); 福建省科协第二届青年人才托举工程
收稿时间: 2021-04-14; 修改时间: 2021-05-22; 采用时间: 2021-06-08; jos 在线出版时间: 2022-09-16
CNKI 网络首发时间: 2022-11-15

signature can be obtained by solving the same for time T. This study first proposes the notion of verifiable attribute-based timed signatures (VABTS) and gives an instantiation VABTS further. The instantiation VABTS scheme can not only simultaneously support identity privacy-preserving, dynamic user revocation, traceability, timing, but also solve the problem of key escrow in attribute-based scheme. In addition, VABTS has many applications. This study lists two application scenarios of VABTS: building a privacy-preserving payment channel network for the permissioned blockchain and realizing a fair privacy-preserving multi-party computing. Finally, it is proved that the instantiation VABTS scheme is secure and efficient via formal security analysis and performance evaluation.

Key words: timed signatures; attribute-based signatures; privacy-preserving; traceability; payment channel networks (PCNs)

Rivest 等人[1]于 1996 年首次提出了时间锁定问题 (time-lock puzzles) 的概念. 时间锁定问题[2−4]是一种需要通过一系列顺序计算才能解决的计算问题, 它可用于构建其他的定时密码学原语. 定时密码学是一类允许发送者将加密信息发送到未来的原语. 在定时密码学原语中, 发送者预先定义一段时间 T, 在预定的时间 T 或经过时间 T 内的一系列计算后, 任何人都可获得该加密的消息. 定时密码学主要由时间锁定问题、定时承诺[5]和定时释放签名[6]这 3 类原语组成, 具有广阔的应用前景[7−9].

在定时密码学应用中, 接收方希望在花费大量的时间和资源来解决相应的时间锁定问题之前, 能够确保发送方发送的消息是可信的. 例如, 在定时释放签名算法中, 接收方为了避免浪费时间和计算资源, 希望在解决时间锁定问题之前, 需要验证发送的消息中是否含有对已知消息的合理签名. 因此, 用可验证性的概念来增强定时密码学原语的实用性是必要的. Thyagarajan 等人[10]于 2020 年提出了可验证定时签名 (verifiable timed signatures, VTS) 的概念. VTS 允许发送方以可验证和可抽取的方式对已知消息的签名生成承诺. 其中, 公开可验证性指的是任何人都可以检查发送方发送的承诺中是否包含有一个对已知消息的合理签名, 可抽取性保证了签名可以在至多时间 T 后从该承诺中恢复出来. 他们还给出了基于 BLS[11]、Schnorr[12]和 ECDSA[13]签名的可验证定时签名方案的具体构造, 并设计了一个基于同态时间锁定问题 (homomorphic time-lock puzzles, *HTLP*) 的高效剪切选择协议 (cut-and-choose protocol), 证明了封装在时间锁定问题中的签名的合理性.

匿名交易和匿名认证等应用场景对用户身份的隐私性要求较高. 但上述基于 BLS、Schnorr 和 ECDSA 的可验证定时签名方案均不能实现签名者身份的隐私保护. 为此, 本文提出了可验证属性基定时签名的概念并给出了一个实例化的构造. 可验证属性基定时签名 (VABTS) 支持签名者的身份隐私保护. 在 VABTS 方案中, 用户的身份由一组属性集合表示, 该属性集合可以表示多个用户. 在这种属性集合与用户的一对多的关系下, 用户身份可以得到隐藏.

可验证属性基定时签名有非常广阔的应用前景, 它不仅可以应用在传统的场景中, 如合同的公平签署[5,6], 还能进一步用于解决以下两个问题.

(1) 构建准入区块链中隐私保护的支付通道网络

比特币[14]以及很多非许可区块链的可扩展性受到低吞吐量以及高额交易费的限制. 为了解决上述问题, 许多非许可区块链采用支付通道[15,16]来提高吞吐量和减少交易费. 支付通道允许一对用户在一段时间内执行多个交易, 只需要将该时间段内的最终交易结果提交到区块链, 而无需提交所有的中间交易. 虽然提交到区块链的交易隐式包含了多笔交易, 但只需支付单笔交易的交易费, 矿工也只需打包需要上链的最终交易. 支付通道不仅可用于非许可区块链, 也可用于准入区块链中来变相的减小交易费和提高交易吞吐量. 如图 1 所示, 支付通道由以下 3 步组成: 首先, 两个用户 A 和 B 通过创建一个区块链交易来打开一个支付通道. 假设该支付通道的目的是 A 向 B 支付一定数量的货币. 那么需要创建一个承诺在时间 T 内 A 向 B 支付确定数量的货币的区块链交易, 且需要 A 和 B 两者的双重签名. 交易创建成功意味着支付通道成功开启. 之后, A 和 B 可以向彼此通过创建相应的交易从联合地址中互相支付一定数量的货币. 最后, 当最后的交易完成后, 或者时间 T 后, 支付通道关闭, 并将最新状态的交易提交到区块链.

支付通道网络 (payment channel networks, PCNs)[16]是支付通道的扩展. 如图 2 所示, 发送者通过中间 5 个用户向接受者支付 100 个货币. 每个中间用户收取一个货币的交易费, 每两个用户之间耗费的时间为 Δ. 在支付通道网络中, 用户可以实现多跳支付. 货币可以通过网络中的一组中间用户路由转移到同处于该网络的接收者, 而不需要一个

共同的支付通道. 支付通道网络中安全的多跳支付是通过使用多跳锁来实现的, 例如比特币区块链中使用哈希时间锁定智能合约 (HTLC) 来实现多跳支付, 只有在一定时间内计算出哈希函数 $H(x)=y$ 的前像 x 的用户, 才能收到货币.

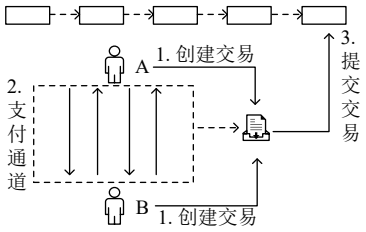


图 1 支付通道实例

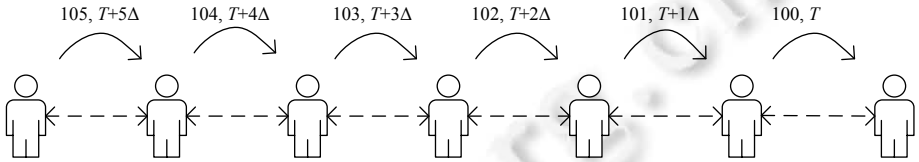


图 2 支付通道网络实例

由于支付通道网络能够增强区块链 (非许可区块链和准入区块链) 的可扩展性, 它得到了来自学术界和工业界的广泛关注[17−20]. 但支付通道网络同时也带来了一系列的交易隐私泄露问题. 使用匿名多跳锁 (anonymous multi-hop locks, AMHL)[21,22]可以在一定程度上将支付路由隐藏起来. 为了防止竞争, 使得多跳支付能够有序地进行, 如图 2 所示, 第 i 跳节点的时间锁比第 i+1 跳的时间锁大 Δ. 链上的攻击者可以根据这些时间锁信息来判断两个用户的支付交易是否属于同一多跳支付路径. 进一步地, 能够恢复出该多跳支付路径.

为了解决上述隐私问题[10], 可使用可验证定时签名 (VTS) 将时间 T 完全地从支付交易中移除. 这样即使时间锁仍然具有相关性, 但不会被发布到链上, 攻击者从区块链上得不到任何关于该多跳支付路径的信息. 此外, 在现实应用中, 用户可能希望在交易过程中隐藏自己的真实身份, 但 VTS 并不能实现对交易双方身份的隐私保护.

众所周知, 属性基签名可以保护签名者的身份隐私[23,24]. 使用可验证的属性基定时签名既可以像 VTS 一样, 将时间 T 完全地从支付交易中移除, 又可以实现对签名者身份的隐私保护. 也就是说, 可验证的属性基定时签名既可以实现对支付通道网络中的多跳支付路径的隐藏, 又可以实现对交易双方身份的隐私保护. 因此, 可验证的属性基定时签名 (VABTS) 可以用于构建准入区块链中隐私保护的支付通道网络.

(2) 实现公平的隐私保护的多方计算

如果多方计算 (multi-party computation, MPC) 的所有参与方都接收到或都没有接收到输出, 则该多方计算是公平的. 最近一些学者提出利用区块链来实现公平的多方计算[25−27]. 大多数方案是鼓励用户完成各自的计算, 如果用户没有计算, 就对其强制执行一些经济惩罚. 具体而言, 所有多方计算的参与者都支付一定数量的货币并签署同一个交易. 如果用户完成了计算, 其可以通过提供完成计算的证据来赎回自己的货币. 没有完成计算的恶意用户不能赎回自己的货币, 他们的货币将作为奖励分发给诚实的参与者. 上述参与者签署的支付交易在链上是时间锁定的, 只在时间 T 之后有效. 这确保了其他用户不能在多方计算终止之前拿走并分发货币.

如同支付通道网络一样, 上述做法需要时间锁, 这使得它不能兼容于不提供时间锁的区块链 Zcash[28]和 Monero[29]. 此外, 该方法完全暴露了多方计算参与者的身份信息. 可验证的属性基定时签名 (VABTS) 可以通过让所有参与者都使用 VABTS 来签署他们的付款交易, 而不是以明文发送签名来解决上述问题. VABTS 的安全性确保安全计算的所有参与者的身份隐私得到了保护, 也使得在时间 T 之前没有任何一方能够获得其他参与者的签名, 这实现了多方计算的公平.

本文贡献可以归纳为以下 3 点.

(1) 本文提出了可验证属性基定时签名的概念. 可验证属性基定时签名允许对已知消息上的签名进行锁定, 在执行时间为 T 的顺序计算后, 任何人都可以从时间锁 (time-lock) 中提取出该签名. 可验证性保证了在无需解开时

间锁的情况下, 任何人都可以公开地验证时间锁中是否包含已知消息上的合理签名, 且可以在执行时间 T 的顺序计算后获得该签名. 签名者的身份由一组由多个用户共享的属性集合表示, 借助于这种一对多的对应关系, 实现了签名者的身份隐私保护.

(2) 本文给出了一个实例化可验证属性基定时签名方案. 基于 RT-ABS[30]方案, 本文构造了一个可撤销和可追溯的可验证属性基定时签名方案 RT-VABTS. 该方案不仅可以同时支持签名者身份隐私保护、动态的用户撤销、可追溯性以及定时性, 还解决了属性基密码中的密钥托管问题.

(3) 本文给出了 VABTS 的两种应用场景: 构建准入区块链中隐私保护的支付通道网络和实现公平的隐私保护的多方计算. 最后, 通过形式化的安全性分析和性能评估证明实例化的 RT-VABTS 方案是安全且高效的.

1 相关工作

(1) 定时签名

定时密码学主要由时间锁难题、定时承诺和定时签名 3 类原语组成. 时间锁难题可用于构建其他两个原语. 时间锁定问题的概念是由 Rivest 等人[1]提出的. 随后, 大量的学者致力于时间锁定问题的相关研究. 其中, Mahmoody 等人给出了一个随机预言机模型中的时间锁定问题方案[31], Bitansky 等人提出了一个基于随机编码的时间锁定问题方案[32], Malavolta 等人提出了一个同态的时间锁定问题[3]. 在 2000 年的 CRYPTO 会议上, Boneh 等人提出了定时承诺的概念[5]. 之后, 研究学者也陆续提出了多种定时承诺方案. 其中 Baum 等人[33]在通用可组合性框架下形式化了时间锁定问题和时间承诺, 并给出了一个 (可编程的) 随机预言机模型中的定时承诺的方案. Ephraim 等人[34]给出了一个在随机预言机模型中由 VDF (verifiable delay function) 构建的时间锁定问题.

Garay 等人提出了定时释放数字签名的概念[6], 允许对给定消息上的签名在时间 T 内进行锁定. 在定时释放签名算法的很多应用中, 接收方为了避免浪费时间和计算资源, 希望在解决时间锁定问题之前, 验证发送方发送的消息中是否含有已知消息的合理签名. 因此, 用可验证性的概念来增强定时密码学原语的实用性是必要的. 在 2020 年的 CCS 中, Thyagarajan 等人提出了可验证定时签名 (VTS) 的概念[10]. VTS 允许发送方以可验证和可抽取的方式对已知消息的签名生成承诺. 其中, 公开可验证性指的是任何人都可以检查发送方发送的承诺中是否包含有一个对已知消息的合理签名, 可抽取性保证了签名可以在至多时间 T 后从该承诺中恢复出来. 他们还给出了基于 BLS[11]、Schnorr[12]和 ECDSA[13]签名的可验证定时签名方案的具体构造, 并设计了一个基于同态时间锁定问题 HTLP 的高效剪切选择协议 (cut-and-choose protocol), 证明了封装在时间锁定问题中的签名的合理性.

现在的一些应用可能会要求实现用户身份隐私保护, 例如匿名交易和匿名认证. 但上述基于 BLS、Schnorr 和 ECDSA 的可验证定时签名方案均不能实现签名者身份的隐私保护.

(2) 属性基签名

Maji 等人[35]对身份基签名进行了扩展, 首次提出了属性基签名的概念. 在属性基签名方案中, 签名者的身份由一组多个用户共享的属性集合表示. 如果用户拥有满足访问控制结构的属性集合, 则可以对消息生成合理的属性基签名. 属性基签名支持隐私保护的认证和公开验证, 任何人都可以验证该签名的合理性. Li 等人[36]以及 Shahandashti 等人[37]分别提出了一个 (t, n) 门限的属性基签名方案. Li 等人[38]提出了另一种性能优于文献 [37] 的属性基签名方案. 随后, Tang 等人提出了一个多授权中心的门限属性基签名[39], Zhang 等人提出了一个可验证的门限属性基签名[40]. Okamoto 等人[41]提出了完全安全的支持非单调谓词的属性基签名方案. 然而, 该方案在实际应用中是低效的. 除了文献 [41], 上述的属性基签名方案的签名的大小都与相关联的属性的数量呈线性关系. 为此, Herranz 等人[42]提出了两个拥有常数签名大小的门限属性基签名方案.

Lian 等人[43]提出了第一个可撤销的属性基签名方案. 在该方案中, 属性颁发机构定期为未撤销的用户颁发密钥更新消息. 之后, Seo 等人[44]发现文献 [43] 不能抵抗签名密钥泄露攻击. 攻击者可以从过期的签名密钥中提取签名者的秘密密钥, 然后通过将其与后续的更新密钥结合来伪造一个合理的签名. 为了解决这个问题, Wei 等人[45]提出了一个抗签名密钥泄露的可撤销的属性基签名方案. 但在该方案中, 需要建立一个属性颁发机构与用户之间的安全通道且存在密钥托管问题. Cui 等人[30]提出了一个可撤销和可追溯的属性基签名方案, 该方案还解决了一般

属性基密码方案中的密钥托管问题.

一些应用场景需要实现签名者身份的隐私保护, 如匿名交易和匿名凭证. 然而, 现存的可验证的定时签名方案不能实现签名者的身份隐私保护. 本文提出了可验证属性基定时签名的概念, 并基于文献 [30] 给出了一个实例化的构造. 可验证属性基定时签名 (VABTS) 不仅实现了可验证定时签名的所有功能, 还支持签名者的身份隐私保护.

2 背景知识

(1) 双线性映射

设 G,G_1 为两个大素数 p 阶的乘法循环群, 其中群 G 的生成元为 g. 双线性映射是一个具有以下特点的映射 $e:G\times G_1\to G_T$.

① 双线性: 对任意的 $g\in G$ 以及 $a,b\in Z_p^*$, $e(g^a,g^b)=e(g,g)^{ab}$.

② 非退化性: $e(g,g)\neq 1$.

③ 可计算性: 对所有的 $g\in G$, $e(g,g)$ 都是可以高效计算的.

(2) Diffie-Hellman 指数问题

l-Diffie-Hellman 指数问题的定义是, 给定一个元组 $(g,g^s,g^a,\ldots,g^{a^l},g^{a^{l+2}},\ldots,g^{a^{2l}})$, 计算 $g^{a^{l+1}s}$ 是困难的, 其中 $a,s\in Z_p$.

(3) 非交互式零知识证明

设 $L=\{x|\exists w:R(x,w)=1\}$ 其中 L 是具有相关证据关系 R 的 NP 语言. 语言 L 的非交互式证明系统 Ω 由以下 3 种算法组成.

① $ZKSetup_{\Omega}(1^{\lambda})$: 该算法输入安全参数 λ, 输出 CRS (common reference string) crs_{Ω}.

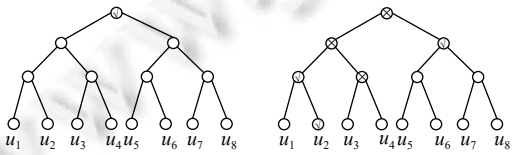
② $ZKProve_{\Omega}(crs_{\Omega},x,w)$: 该算法以 CRS crs_{Ω}, 声明 x 以及相应的证据 w 作为输入, 输出证明 π.

③ $ZKVerify_{\Omega}(crs_{\Omega},x,\pi)$: 该算法以 CRS crs_{Ω}, 声明 x 以及相应的证明 π 作为输入, 如果证明 π 是合理的, 输出 1. 否则, 该算法输出 0.

非交互式的零知识证明 (non-interactive zero-knowledge proof, NIZK) 应该满足以下条件: 1) 零知识, 即验证者除了声明 x 的合理性之外不能获得关于它的任何其他信息; 2) 合理模拟, 对于任何一个证明者来说, 即使可以多项式次自适应的访问预言机生成证明, 也很难向验证者证明其拥有一个不合理的声明.

(4) KUNodes 算法

令 BT 表示为包含有 N 个叶子节点的二叉树[46], 其中每个叶子节点分别对应一个用户. $root$ 为二叉树的根节点, 二叉树中的其他根节点用 θ 表示. 如果 θ 为叶子节点, 那么用 $Path(\theta)$ 表示从该叶子节点到二叉树根节点的路径上所有节点的集合. 需要注意的是, 该集合包括该叶子节点和根节点. 如果 θ 是非叶子节点, θ_l 和 θ_r 分别表示该节点的左孩子节点和右孩子节点. 本文使用到的 KUNodes 算法[30]是用来计算需要发布密钥更新消息的最小节点集和, 以确保只有在 t 时间周期内未被撤销的用户才能够解密密文. KUNodes 算法以二叉树 BT, 撤销列表 rl 以及时间周期 t 作为输入, 它输出一个满足以下条件的 BT 中最小的节点集合: 列表 rl 中节点及其祖先均不在该集合中, 而后选取其他叶子节点的最浅的祖先节点. 如图 3 所示, KUNodes 算法首先标记所有已经撤销的叶子节点及其祖先节点 (假设用户 u_4 撤销), 而后输出所有被标记为撤销的节点的未撤销的孩子节点. 算法 1 是 KUNodes 算法的形式化表述.



⊙ KUNodes 输出的节点
⊗ KUNodes 标记为已经撤销的节点

图 3 KUNodes 算法实例

算法 1. $KUNodes(BT,rl,t)$.

$X,Y \leftarrow \varnothing$

$\forall(\theta_i,t_i)\in rl$, 如果 $t_i \leqslant t$, 将 $Path(\theta_i)$添加到 X 中

$\forall x\in X$, 如果 $x_l \notin X$, 将 x_l 添加到 Y 中

　　如果 $x_r \notin X$, 将 x_r添加到 Y 中

如果 $Y=\varnothing$, 将 $root$ 添加到 Y 中

返回 Y

(5) 门限秘密分享

秘密分享是一种可以先将一个秘密分成多个份额, 而后使用阈值数量的秘密份额就可以恢复出原始秘密的算法. Shamir[47]提出了门限秘密分享方案由以下两种算法组成.

① $SS.share(s,n)$: 该算法以秘密 s 以及用户总数量 n 作为输入, 输出 n 个秘密份额 $(s_1,s_2,\ldots,s_n)$.

② $SS.reconstruct(t,$至少 t 个秘密份额$)$: 该算法以阈值 t 以及至少 t 个秘密份额作为输入, 输出秘密 s.

秘密共享方案的安全性要求只知道一组小于阈值大小的秘密份额的集合并不能了解到关于秘密选择 s 的任何信息.

(6) 时间锁定问题

时间锁定问题允许在一定时间内隐藏某个数值[1], 且确保了该问题能够在多项式时间内解决. 时间锁定问题通常由以下两种算法组成.

① $PGen(T,s\in\{0,1\}^*,r)$: 该算法以难度参数 T, 解决方法 s 以及随机数 r 作为输入, 输出一个问题 Z.

② $PSolve(Z)$: 该算法以问题 Z 作为输入, 输出解决方法 s.

(7) 同态时间锁定问题

同态时间锁定问题[3]允许对不同的时间锁定问题执行同态操作. 具体地, 同态时间锁定问题由以下 4 种算法组成.

① $HTLP.PSetup(1^\lambda,T)$: 该算法以安全参数 1^λ 和时间难度 T 作为输入, 输出公开参数 pp_{HTLP}.

② $HTLP.PGen(T,s,r,pp_{HTLP})$: 该算法以难度参数 T, 解决方法 $s\in\{0,1\}^*$, 随机数 r 以及公开参数 pp_{HTLP} 作为输入, 输出一个问题 Z.

③ $HTLP.PSolve(Z,pp_{HTLP})$: 该算法以问题 Z 以及公开参数 pp_{HTLP} 作为输入, 输出解决方法 s.

④ $HTLP.PEval(C,\{Z_1,\ldots,Z_n\},pp_{HTLP})$: 该算法以电路 $C:\{0,1\}^n\rightarrow\{0,1\}$, 公开参数 pp_{HTLP}以及问题集合 $\{Z_1,\ldots,Z_n\}$ 作为输入, 输出一个问题 Z'.

3 可验证的属性基定时签名

3.1 算法定义

可验证的属性基定时签名 (VABTS) 允许签名者 (其身份用一组属性集合表示) 对给定消息生成签名, 并对该签名生成承诺, 最后将该承诺分享给其他用户. 在时间 T 之后, 签名可以从承诺中释放出来. 如果在时间 T 之后, 签名并未得到释放, 用户可以通过执行多项式时间 (大于时间 T) 的计算来强制释放该签名. 为了避免造成计算资源的浪费, 用户在执行上述多项式时间的计算之前, 可以验证该承诺中是否包含了对给定消息的签名者的合理签名.

可验证的属性基定时签名方案比传统的属性基签名方案增加了 3 个算法: $Commit$, $Open$ 和 $ForceOp$. 其中 $Commit$ 算法使用承诺方案将签名算法生成的签名 σ 进行锁定, 只有经过时间 T 或者解决了同态时间锁定生成的难题才能解开承诺获得签名. 此外, 在 $Commit$ 算法中也使用零知识证明系统来确保承诺中包含有合理的签名. $Open$ 算法是经过时间 T 后, 承诺打开, 获得签名 σ. $ForceOp$ 算法描述的是接收者欲通过解决同态时间锁定难题来获得签名 σ. 具体地, 可验证的属性基定时签名主要由以下 7 种算法组成.

(1) $Setup(1^{\lambda})$: 该算法以安全参数 λ 为输入, 输出公开参数 par 和主私钥 msk.

(2) $KeyGen(par,msk,A)$: 该算法以公开参数 par, 主私钥 msk 和用户拥有的属性集合 A 作为输入, 输出用户 id 的属性私钥 $sk_{id,A}$.

(3) $Sign(par,m,sk_{id,A},\Gamma_{k,S})$: 该算法以公开参数 par, 用户的属性私钥 $sk_{id,A}$, 消息 m 和声称谓词 $\Gamma_{k,S}$ 作为输入, 输出签名 σ.

(4) $Commit(par,\sigma,T)$: 该算法以公开参数 par, 签名 σ 和时间难度 T 为输入, 输出承诺 C 和证明 π. 其中 π 是用于向用户证明该承诺 C 中包含有对消息 m 的合理签名.

(5) $Vrfy(par,m,C,\pi)$: 该算法以公开参数 par, 消息 m, 承诺 C 和证明 π 为输入, 如果承诺 C 中包含一个消息 m 的合理签名 σ, 输出 1. 否则, 该算法输出 0.

(6) $Open(C)$: 该算法以承诺 C 为输入, 输出承诺中包含的签名 σ 和生成承诺 C 过程中使用到的随机数 r.

(7) $ForceOp(C)$: 该算法以承诺 C 为输入, 输出签名 σ.

对于满足 $\Gamma_{k,S}$ 的属性集合 A, 如果 $(par,msk) \leftarrow Setup(1^{\lambda})$, $sk_{id,A} \leftarrow KeyGen(par,msk,A)$, $\sigma \leftarrow Sign(par,m,sk_{id,A},\Gamma_{k,S})$, 对于 $(C,\pi) \leftarrow Commit(\sigma,T)$ 使得 $1 \leftarrow Vrfy(par,m,C,\pi)$, $(C,r) \leftarrow Open(C)$, $\sigma \leftarrow ForceOp(C)$, 则称该可验证的属性基定时签名是正确的.

3.2 安全性定义

可验证的属性基定时签名应该满足以下安全特性: 不可伪造性、匿名性、合理性和隐私性.

定义 1. VABTS 不可伪造性. 不可伪造性保证了只有拥有满足签名策略 $\Gamma_{k,S}$ 的属性集合的用户才能生成合理的签名. 为了形式化的表述不可伪造性, 我们定义了挑战者 C 与敌手 $\mathcal{A}$ 之间的安全性游戏. 这里的敌手 $\mathcal{A}$ 不知道主私钥.

(1) $Setup$. 挑战者 C 运行 $Setup$ 算法, 生成公开参数 par 和主私钥 msk. 而后, 将公开参数 par 发送敌手 $\mathcal{A}$, 并将主私钥 msk 秘密保存在本地.

(2) 询问阶段. 敌手 $\mathcal{A}$ 可以向挑战者 C 发送多项式次询问.

① $KeyGen$ 询问: 在这个阶段, 敌手 $\mathcal{A}$ 可以向挑战者 C 发送多项式次私钥问询. 假设敌手 $\mathcal{A}$ 向挑战者 C 问询属性集合 A 的属性密钥, 挑战者 C 运行 $KeyGen$ 算法, 并将 $sk_{id,A}$ 发送给敌手 $\mathcal{A}$.

② $Sign$ 询问: 敌手 $\mathcal{A}$ 向挑战者 C 问询消息 m 的签名, $\mathcal{A}$ 向 C 提交一个满足签名策略 $\Gamma_{k,S}$ 的属性集合, 挑战者 C 首先运行 $KeyGen$ 算法为其生成属性密钥, 而后运行 $Sign$ 算法生成签名 σ 并将该签名发送给 $\mathcal{A}$.

③ $Commit$ 询问: 敌手 $\mathcal{A}$ 向挑战者 C 问询签名 σ 在时间难度 T 下的承诺和证明. C 运行 $Commit$ 算法为签名 σ 和时间难度 T 生成承诺 C 和证明 π, 并将其返回给 $\mathcal{A}$.

④ $Vrfy$ 询问: 敌手 $\mathcal{A}$ 向挑战者 C 问询关于签名 σ 的承诺 C 和证明 π 的合理性. 如果承诺 C 中包含一个消息 m 的合理签名 σ, C 返回 1 给 $\mathcal{A}$. 否则, 返回 0 给 $\mathcal{A}$.

⑤ $Open$ 询问: 敌手 $\mathcal{A}$ 向挑战者 C 问询承诺 C 中包含的签名 σ. C 运行 $Open$ 算法并将承诺 C 中的签名 σ 以及生成承诺过程中使用到的随机数 r 返回给 $\mathcal{A}$.

(3) 伪造阶段. 敌手 $\mathcal{A}$ 输出一个关于属性密钥 sk_{id,A^*}, 消息 m^*, 签名策略 $\Gamma^*_{k,S}$ 的签名 σ^*. 并运行 $Commit(par,\sigma^*,T)$ 算法得到 (C^*,π^*), 如果签名 (C^*,π^*) 通过了 $Vrfy$ 算法, 则称敌手 $\mathcal{A}$ 赢得了该游戏. 这里需要注意的是, 在 $KeyGen$ 问询时, 不能问询关于属性集合 A^* 的密钥 sk_{id,A^*}. 在 $Sign$ 问询时, 不能问询关于消息 m^*, 签名策略 $\Gamma^*_{k,S}$ 的签名 σ^*.

如果上述游戏中, 敌手 $\mathcal{A}$ 获胜的概率 $Adv^{\mathrm{UNF}}_{\mathcal{A}}(\lambda) = \Pr[\mathcal{A}\ wins]$ 是可忽略的, 则称该可验证属性基定时签名是不可伪造的.

定义 2. 匿名性. 匿名性保证了通过合理的签名, 不能得到关于满足该签名策略 $\Gamma_{k,S}$ 的属性集合的任何其他消息. 换句话说, 敌手不能分辨出满足同一签名策略 $\Gamma_{k,S}$ 的两个属性集合. 为了形式化的表述匿名性, 我们定义了挑战者 C 与敌手 $\mathcal{A}$ 之间的安全性游戏.

① $Setup$. 与不可伪造性的 $Setup$ 相同.

② 询问阶段. 与不可伪造性的询问阶段相同.

③ 挑战阶段. $\mathcal{A}$ 将两个满足签名策略 $\Gamma_{k,S}^{*}$ 的两个属性集合 A_0^{*}, A_1^{*}, 消息 m^{*} 发送给挑战者 C. 挑战者 C 选择随机数 $b\in\{0,1\}$, 为签名策略 $\Gamma_{k,S}^{*}$ 下的消息 m^{*}, 签名密钥 $sk_{id,A_b^{*}}$ 生成签名 σ^{*}. 并将该签名发送给 $\mathcal{A}$.

④ 猜测阶段. 首先, $\mathcal{A}$ 可以进行与询问阶段相同的操作. 而后敌手 $\mathcal{A}$ 输出猜测结果 b'. 如果 $b'=b$, 则称 $\mathcal{A}$ 赢得了该游戏.

如果上述游戏中, 敌手 $\mathcal{A}$ 获胜的概率 $Adv_{\mathcal{A}}^{\mathrm{ANON}}(\lambda)=|\Pr[b'=b]-1/2|$ 是可忽略的, 则称该可验证属性基定时签名是匿名的.

定义 3. 合理性. 如果一个可验证的属性基定时签名是合理的, 那么对于任意的多项式时间敌手 $\mathcal{A}$, 以下概率是可忽略的:

$$\Pr\left[\begin{array}{l} b_1=1\wedge b_2=0: \end{array}\begin{array}{c}(par,m,C,\pi,T)\leftarrow\mathcal{A}\left(1^{\lambda}\right)\\ \sigma\leftarrow ForceOp(C)\\ b_1=Vrfy(par,m,C,\pi)\\ b_2=VerifyS\left(par,m,\sigma,\Gamma_{k,S}\right)\end{array}\right]\leqslant negl(\lambda)$$

, 其中 $VerifyS(par,m,\sigma,\Gamma_{k,S})$ 为传统属性基签名中的签名验证算法.

定义 4. 隐私性. 如果存在一个多项式时间模拟器 S, 一个可忽略函数 $negl$, 所有 PRAM 敌手 $\mathcal{A}$ 的最大运行时间 $t<T$, 且下面不等式成立.

$$\left|\Pr\left[\mathcal{A}(par,m,C,\pi)=1:\begin{array}{c}sk_{id,A}\leftarrow KeyGen(par,msk,A)\\ \sigma\leftarrow Sign(par,m,sk_{id,A},\Gamma_{k,S})\\ (C,\pi)\leftarrow Commit(\sigma,T)\end{array}\right]-\Pr\left[\mathcal{A}(par,m,C,\pi)=1:\begin{array}{c}sk_{id,A}\leftarrow KeyGen(par,msk,A)\\ (C,\pi,m)\leftarrow S(par,T)\end{array}\right]\right|\leqslant negl(\lambda),$$

则称该可验证的属性基定时签名是隐私的.

4 一个实例化的可验证的属性基定时签名方案

本节给出了一个属性基定时签名的实例化构造. 基于属性基签名[30], 我们构造出一个可撤销的, 可追溯的可验证属性基定时签名方案 (RT-VABTS), 并给出了相应的安全性定义. 此外, RT-VABTS 方案也解决了密钥托管问题.

4.1 算法定义

相较于前面定义的可验证属性基定时签名, RT-VABTS 方案的组成算法中增加了 *UserKG*, *KeyUp*, *SignKG*, *Trace* 和 *Revoke* 算法. 具体而言, RT-VABTS 方案包括如下算法.

(1) $Setup(1^{\lambda})$: 该算法以安全参数 λ 作为输入, 输出公开参数 par, 主私钥 msk, 初始为空的撤销列表 rl 和状态 st.

(2) $UserKG(par,id)$: 该算法以公开参数 par, 用户身份 id 作为输入, 输出用户的公私密钥对 (sk_{id},pk_{id}).

(3) $KeyGen(par,msk,pk_{id},A,st)$: 该算法以公开参数 par, 主私钥 msk, 用户公钥 pk_{id}, 属性集合 A 以及状态 st 作为输入, 输出属性私钥 $sk_{id,A}$, 并更新状态 st. 同时, 属性密钥颁发机构定义用户列表 L_U, 存储有 (id,pk_{id}).

(4) $KeyUp(par,msk,t,rl,st)$: 该算法以公开参数 par, 主私钥 msk, 时间段 t, 撤销列表 rl 以及状态 st 作为输入, 输出密钥更新消息 ku_t 并更新状态 st.

(5) $SignKG(par,id,sk_{id,A},ku_t)$: 该算法以公开参数 par, 用户身份 id, 属性私钥 $sk_{id,A}$ 以及密钥更新消息 ku_t 作为输入, 输出用户在时间段 t 的签名密钥 $sk_{id,A}^{t}$.

(6) $Sign(par,sk_{id},sk_{id,A}^{t},t,\Gamma_{k,S},m)$: 该算法以公开参数 par, 用户私钥 sk_{id}, 签名密钥 $sk_{id,A}^{t}$, 时间段 t, 签名策略 $\Gamma_{k,S}$ 以及消息 m 作为输入, 输出签名 σ.

(7) $Commit(par,\sigma,T)$: 该算法以公开参数 par, 签名 σ 和时间难度 T 为输入, 输出承诺 C 和证明 π. 其中 π 是用于向用户证明该承诺 C 中包含有对消息 m 的合理签名.

(8) $Vrfy(par,m,C,\pi)$: 该算法以公开参数 par, 消息 m, 承诺 C 和证明 π 为输入, 如果承诺 C 中包含一个消息 m 的合理签名 σ, 输出 1. 否则, 该算法输出 0.

(9) $Open(C)$: 该算法以承诺 C 为输入, 输出承诺中包含的签名 σ 和生成承诺 C 过程中使用到的随机数 r.

(10) $ForceOp(C)$: 该算法以承诺 C 为输入, 输出签名 σ.

(11) $Trace(par,msk,(t,\Gamma_{k,S},m,\sigma),L_U)$: 该算法以公开参数 par, 主私钥 msk, 消息 m 在签名策略 $\Gamma_{k,S}$ 的时间段 t 内的签名 σ 以及用户列表 L_U 作为输入, 输出用户身份 id.

(12) $Revoke(id,t,rl,st)$: 该算法以需要撤销的用户身份 id, 时间段 t, 撤销列表 rl 和状态 st 作为输入, 输出更新后的撤销列表 rl.

对于满足 $\Gamma_{k,S}$ 的属性集合 A, 对于所有按照上述定义的算法执行得到的签名, 都能通过验证算法, 则称该 RT-VABTS 方案是正确的.

4.2 安全性定义

除了需要满足一般的可验证属性基定时签名方案的所有安全特性, RT-VABTS 方案需要额外满足可追溯性. 此外, RT-VABTS 解决了密钥托管的问题, 在不可伪造性方面有了更强的安全假设. 下面我们只介绍 RT-VABTS 与一般的可验证属性基定时签名方案的安全特性不同的地方.

定义 5. RT-VABTS 不可伪造性. 不可伪造性保证了只有拥有满足签名策略 $\Gamma_{k,S}$ 的属性集合的用户才能生成合理的签名. 为了形式化地表述不可伪造性, 我们定义了挑战者 C 与敌手 $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$ 之间的安全性游戏. 敌手 $\mathcal{A}_1$ 不知道主私钥, 敌手 $\mathcal{A}_2$ 知道主私钥. 由于挑战者 C 与敌手 $\mathcal{A}_1$ 之间的游戏与第 3.2 节定义的相同, 这里我们只介绍挑战者 C 与敌手 $\mathcal{A}_2$ 之间的游戏.

① *Setup*. 挑战者 C 运行 *Setup* 算法, 生成公开参数 *par* 和主私钥 *msk*. 而后, 挑战者 C 为每个用户 *id* 生成公私密钥对 (sk_{id},pk_{id}), 并将 (id,pk_{id}) 存储到用户列表 L_U 中. 最后, C 将公开参数 *par*, 主私钥 *msk* 以及用户列表 L_U 发送敌手 $\mathcal{A}_2$.

② 询问阶段. 敌手 $\mathcal{A}_2$ 可以向挑战者 C 发送多项式次用户私钥问询, 挑战者 C 返回用户 *id* 的私钥 sk_{id} 给 $\mathcal{A}_2$.

③ 伪造阶段. 敌手 $\mathcal{A}_2$ 输出一个时间段 t^*, 签名策略 $\Gamma_{k,S}^*$, 消息 m^* 以及签名 σ^*. 并运行 $Commit(par,\sigma^*,T)$ 算法得到 (C^*,π^*), 如果签名 (C^*,π^*) 通过了 *Vrfy* 算法, 且 $Trace(par,msk,(t^*,\Gamma_{k,S}^*,m^*,\sigma^*),L_U)\to id$, 则称敌手 $\mathcal{A}_2$ 赢得了该游戏. 这里需要注意的是, 在 *KeyGen* 问询时, 不能问询关于 id^* 的私钥 sk_{id^*}.

如果上述游戏中, 敌手 $\mathcal{A}_2$ 获胜的概率 $Adv_{\mathcal{A}_2}^{\mathrm{UNF}}(\lambda)=\Pr[\mathcal{A}_2\ wins]$ 是可忽略的, 则称该可验证属性基定时签名是不可伪造的.

定义 6. 可追溯性. RT-VABTS 方案的可追溯性是它的正确性来保证的. 也就是, 对于任何诚实生成的签名可以以不可忽略的概率追溯到签名者的身份.

4.3 具体构造

定义拉格朗日系数为 $\Delta_i^{\gamma}(x)=\prod_{j\in\gamma}^{j\neq i}(x-j/i-j)$, 其中 $i,\gamma\in\mathbb{Z}_p$. $\mathbb{Z}_p$ 上的 $d-1$ 阶多项式 $q(x)$ 可用拉格朗日插值计算得到 $q(x)=\sum_{i\in\gamma}q(i)\Delta_i^{\gamma}(x)$, 其中 $|\gamma|=d$.

(1) $Setup(1^{\lambda})$: 输入安全参数 λ, 密钥颁发机构进行以下操作. 令 d 为签名策略的阈值上限, N 为可服务的用户的最大数量. A 为属性空间, $D(|D|=d)$ 为默认属性, 并假设 $A\cup D$ 中的每个元素均来自 $\mathbb{Z}_p$.

① 运行 $ZKSetup_{\Omega}(1^{\lambda})$ 算法得到 crs_{Ω}, 运行 $HTLP.PSetup(1^{\lambda},T)$ 得到 pp_{HTLP}.

② 令 $e:G\times G\to G_1$ 为一个双线性映射, 其中 G 为一个素数 p 阶的乘法循环群, 它的生成元是 g. 将撤销列表 rl 初始化为空, BT 为一颗有 N 个叶子节点的二叉树. 设置 $st=BT$. 为二叉树中的每个节点 x 选取一个随机数 $r_x\in\mathbb{Z}_p$, 并将随机数存储在对应节点中.

③ 随机选取 $\alpha\in\mathbb{Z}_p$, 并计算 $V=e(g,g)$, $Z=V^{\alpha}$. 随机选取一个向量 $\vec{v}=(v_0,\ldots,v_l)\in\mathbb{Z}_p^{l+1}$, 并计算 $h_i=g^{v_i}$, 其中 $l=2d+1$, $i\in[0,l]$. 此外, 定义两个函数 $F_1(t)=f_0\prod_{j=1}^{n_t}f_j^{t_j}$, $F_2(m)=w_0\prod_{i=1}^{n_m}w_i^{m_i}$, 其中 $f_0,\ldots,f_{n_t}$, $w_0,\ldots,w_{n_m}\in G$ 为随机数, t_j 表示时间段 t 的第 j 个比特, m_i 表示为消息 m 的第 i 个比特.

④ 密钥颁发机构将主私钥 $msk=\alpha$ 保存在本地, 将公开参数 $par=(g,G,G_1,\hat{e},p,H,V,Z,crs_{\Omega},pp_{HTLP},\{h_0,\ldots,h_l\},\{f_0,\ldots,f_{n_t}\},\{w_0,\ldots,w_{n_m}\})$ 公开, 其中 H 是一个将 G_1 和 $\mathbb{Z}_p$ 中的元素映射到 $\mathbb{Z}_p$ 的哈希函数.

(2) $UserKG(par,id)$: 输入公开参数 par, 用户身份 id, 用户做如下操作. 首先, 选择随机数 $\beta \leftarrow \mathbb{Z}_p^*$ 作为用户的私钥 sk_{id}, 计算用户公钥 $pk_{id}=g^{\beta}$. 用户将它的公钥 pk_{id} 以及公钥的零知识证明 $PK\{(\beta): pk_{id}=g^{\beta}\}$ 发送给密钥颁发中心, 作为注册信息.

(3) $KeyGen(par,msk,pk_{id},A,st)$: 输入公开参数 par, 主私钥 msk, 用户公钥 pk_{id}, 属性集合 A 以及状态 st, 密钥颁发中心首先将 (id,pk_{id}) 添加到用户列表 L_U. 然后, 将用户的身份 id 存储在二叉树 BT 中未定义的叶子节点 θ 中. 对每个节点 $x \in Path(\theta)$ 做如下操作.

① 随机选择 $a_1,\ldots,a_{d-1},r_w \in \mathbb{Z}_p$, 定义多项式 $q_x(w)\sum_{i=1}^{d-1} a_i w^i+\alpha$. 对每个属性 $w \in A \cup D$, $\forall i \in [i,l-1]$, 计算 $P_{x,w}=(pk_{id}^{q_x(w)}/g^{r_x})\cdot h_0^{r_w}$, $P_{x,w,0}=g^{r_w}$, $P_{x,w,i}=(h_1^{-w^i}\cdot h_{i+1})^{r_w}$.

② 为用户 id 输出属性私钥 $sk_{id,A}=\{x,\{P_{x,w},P_{x,w,0},\{P_{x,w,i}\}_{i\in[1,l-1]}\}_{w\in A\cup D}\}_{x\in Path(\theta)}$.

(4) $KeyUp(par,msk,t,rl,st)$: 输入公开参数 par, 主私钥 msk, 时间段 t, 撤销列表 rl 以及状态 st, 密钥颁发机构对每个 $x \in KUNodes(BT,rl,t)$, 随机选择 $s_x \in \mathbb{Z}_p$, 并计算 $Q_{x,1}=g^{r_x}\cdot F_1(t)^{s_x}$, $Q_{x,2}=g^{s_x}$. 密钥更新消息为 $ku_t=\{x,Q_{x,1},Q_{x,2}\}_{x\in KUNodes(BT,rl,t)}$.

(5) $SignKG(par,id,sk_{id,A},ku_t)$: 输入公开参数 par, 用户身份 id, 属性私钥 $sk_{id,A}$ 以及密钥更新消息 ku_t, 用户做如下操作. 令 $I=Path(\theta)$, $J=KUNodes(BT,rl,t)$, 如果 $I\cap J\neq\varnothing$, 该算法返回 $\perp$. 否则, 对任意 $x\in I\cap J$, 选择随机数 $r'_w \in \mathbb{Z}_p$. 并对每个属性 $w\in A\cup D$, 计算 $k_w=P_{x,w}\cdot Q_{x,1}\cdot F_1(t)^{r'_w}=pk_{id}^{q_x(w)}\cdot h_0^{r_w}\cdot F_1(t)^{S_x+r'_w}$, $k_{w,0}=P_{x,w,0}=g^{r_w}$, $k_{w,t}=Q_{x,2}\cdot g^{r'_w}=g^{S_x+r'_w}$, $k_{w,i}=P_{x,w,i}=(h_1^{-w^i}\cdot h_{i+1})^{r_w}$, $\forall i\in[i,l-1]$. 最后, 设置用户的签名密钥为 $sk_{id,A}^t=\{k_w,k_{w,0},\{k_{w,i}\}_{i\in[i,l-1]},k_{w,t}\}_{w\in A\cup D}$.

(6) $Sign(par,sk_{id},sk_{id,A}^t,t,\Gamma_{k,S},m)$: 输入公开参数 par, 用户私钥 sk_{id}, 时间段 t 的签名密钥 $sk_{id,A}^t$, 签名策略 $\Gamma_{k,S}$ 以及消息 m, 用户的做法如下.

① 随机选择一个 $\{A\cap S\}$ 的子集 S' 以及默认属性子集 $\Omega'\in\Omega$, 使得 $|S'|=k$, $|\Omega'|=d-k$. 然后, 从多项式 $\varphi(y)=\prod_{w\in S'\cup\Omega'}(y-w)=\sum_{i=1}^{l} b_i y^{i-1}$ 中定义一个系数向量 $\vec{b}=(b_1,\ldots,b_l)\in\sum_{i=1}^{l} b_i y^{i-1}$, 其中 $b_i\in[0,|S\cup\Omega'|+2\leqslant i\leqslant l]$.

② 令 $\gamma=S'\cup\Omega'$, 并对每个属性 $w\in\gamma$ 计算 $k'_w=k_w\cdot\prod_{i=1}^{l-1}K_{w,i}^{b_{i+1}}=pk_{id}^{q_x(w)}\cdot\left(h_0\prod_{i=1}^{l}h_i^{b_i}\right)^{r_w}\cdot F_1(t)^{S_x+r'_w}$. 之后, 计算得到 $K_0=\prod_{w\in\gamma}K_w^{\Delta_w^{\gamma}(0)}=pk_{id}^{\alpha}\cdot\left(h_0\prod_{i=1}^{l}h_i^{b_i}\right)^{r}\cdot F_1(t)^{r}$, $K'_1=\prod_{w\in\gamma}K_{w,0}^{\Delta_w^{\gamma}(0)}=g^{r}$, $K'_t=\prod_{w\in\gamma}K_{w,t}^{\Delta_w^{\gamma}(0)}=g^{r'}$, 其中 $r=\sum_{w\in\gamma}\Delta_w^{\gamma}(0)\cdot r_w$, $r'=\sum_{w\in\gamma}\Delta_w^{\gamma}(0)\cdot S_x+r_w$.

③ 随机选择 $s,s_0,s_1,s_2\in\mathbb{Z}_p$, 并计算 $B=Z^{\beta}\cdot V^{s}$, $\sigma_0=g^{s}\cdot K'_0\cdot\left(h_0\prod_{i=1}^{l}h_i^{b_i}\right)^{s_0}\cdot F_1(t)^{s_1}\cdot F_2(m)^{s_2}$, $\sigma_1=K'_1\cdot g^{s_0}$, $\sigma_t=K'_t\cdot g^{s_1}$, $\sigma_2=g^{s_2}$.

④ 随机选择 $u_0,u_1\in\mathbb{Z}_p$, 计算 $R_1=Z^{u_0}$, $R_2=Z^{u_1}\cdot V^{u_0}$, $c=H(R_1\|R_2\|Y\|B\|m)$, $\theta_0=u_0-cs$, $\theta_1=u_1-c\beta$.

⑤ 输出签名 $\sigma=(B,\sigma_0,\sigma_1,\sigma_t,\sigma_2,R_1,R_2,c,\theta_0,\theta_1,Y=Z^{s})$.

(7) $Commit(par,\sigma,T)$: 输入公开参数 par, 签名 σ 以及时间难度 T, 用户操作如下:

① 对所有的 $i\in[k-1]$, 随机选择 $\beta_i\in\mathbb{Z}_p$, 计算 $B'_i=Z^{\beta_i}$, $B_i=B'_i\cdot V^{s}$, $K'_{0,i}=g^{\alpha\beta_i}$, $\sigma_{0,i}=K'_{0,i}\cdot g^{s}\cdot\left(h_0\prod_{i=1}^{l}h_i^{b_i}\right)^{r}\cdot F_1(t)^{r'}\cdot\left(h_0\prod_{i=1}^{l}h_i^{b_i}\right)^{s_0}\cdot F_1(t)^{s_1}\cdot F_2(m)^{s_2}$.

② 对所有的 $i\in[k,n]$, 计算 $B_i=\left(Z^{\beta}/\prod_{j\in[k-1]}B_j^{l_j(0)}\right)^{l_{i(0)}^{-1}}$, $B_i=B'_i\cdot V^{s}$, $K_{0,i}=\left(g^{\alpha\beta}/\prod_{j\in[k-1]}K_{0,i}^{l_j(0)}\right)^{l_{i(0)}^{-1}}$, $\sigma_{0,i}=K'_{0,i}\cdot g^{s}\cdot\left(h_0\prod_{i=1}^{l}h_i^{b_i}\right)^{r}\cdot F_1(t)^{r'}\cdot\left(h_0\prod_{i=1}^{l}h_i^{b_i}\right)^{s_0}\cdot F_1(t)^{s_1}\cdot F_2(m)^{s_2}$, 其中 $l_j(\cdot)$ 表示为第 i 个拉格朗日多项式基.

③ 对每个 $i\in[n]$, 输出签名 $\sigma_i=(B_i,\sigma_{0,i},\sigma_1,R_1,R_2,c_i,\theta_0,\theta_{1,i},Y=Z^{s})$, 其中 $c_i=H(R_1\|R_2\|Y\|B_i\|m)$, $\theta_{1,i}=u_1-c\beta_i$.

④ 对每个 $i\in[n]$, 随机选择 $r_i\leftarrow\{0,1\}^{\lambda}$, 计算下列数值 $Z_i=HTLP.PGen(pp_{HTLP},\sigma_i,r_i)$, $\pi_i=ZKProve_{\Omega}(crs_{\Omega},(Z_i,0,2^{\lambda},T),(\sigma_i,r_i))$. 计算 $I'=H'(B,(B'_1,B_1,Z_1,\pi_1),\ldots,(B'_n,B_n,Z_n,\pi_n))$.

⑤ 输出承诺 $C=(Z_1,\ldots,Z_n,T)$, 证明 $\pi=(\{(B_i,\pi_i)\}_{i\in[n]},I',\{\sigma_i,r_i\}_{i\in I'})$.

(8) $Vrfy(par,m,C,\pi)$: 输入 (par,m,C,π), 做如下操作:

① 是否存在 $j \notin I'$, 使得 $\prod_{i\in I} B_i^{l_i(0)} \cdot B_j^{l_j(0)} \neq Z^{\beta}$.

② 是否存在 $i \in [n]$, 使得 $ZKProve_{\Omega}(crs_{\Omega},(Z_i,0,2^{\lambda},T),\pi_i) \neq 1$.

③ 计算 $R'_{1,i} = Y^{c_i} \cdot Z^{\theta_0}$, $R'_{2,i} = B_i^{c_i} \cdot Z^{\theta_{1,i}} \cdot V^{\theta_0}$, 检查是否 $c_i \neq H(R'_{1,i} \| R'_{2,i} \| Y \| B_i \| m)$.

④ 检测是否存在一个 $i \in I'$, 使得下列不等式 $Z_i \neq HTLP.PGen(pp_{HTLP},\sigma_i,r_i)$ 或 $(e(g,\sigma_{0,i})/(e(h_0\prod_{i=1}^{l} h_i^{b_i},\sigma_1)\cdot e(F_1(t),\sigma_t)\cdot e(F_2(m),\sigma_2)) \neq B_i$ 成立.

⑤ $I' \neq H'(B,(B'_1,B_1,Z_1,\pi_1),...,(B'_n,B_n,Z_n,\pi_n))$.

如果上述均不成立, 该算法输出 1. 否则, 输出 0.

(9) $Open(C)$: 该算法输出 $(\sigma,\{r_i\}_{i\in[n]})$.

(10) $ForceOp(C)$: 对所有 $i \in [n]$, 运行 $HTLP.PSolve(pp_{HTLP},Z_i)$ 得到 σ_i. 注意到, 承诺者已经打开了 $t-1$ 个问题, 这意味着 $ForeOp$ 只需要解决 $(n-t+1)$ 个问题. 输出 $\sigma = \left(\prod_{j\in[k]} B_j^{l_j(0)} \cdot V^{s},\sigma_0,\sigma_1,R_1,R_2,c,\theta_0,Y,\theta_{1,i}\right)$, 其中 $Y = Z^{s}$, $\sigma_0 = \prod_{j\in[k]} K_{0,j}^{l_j(0)} \cdot g^{s} \cdot \left(h_0\prod_{i=1}^{l} h_i^{b_i}\right)^{r} \cdot F_1(t)^{r} \cdot \left(h_0\prod_{i=1}^{l} h_i^{b_i}\right)^{s_0} \cdot F_1(t)^{s_1} \cdot F_2(m)^{s_2}$, $c = H\left(R_1 \| R_2 \| Y \| \prod_{j\in[k]} B_j^{l_j(0)} \| m\right)$, $\theta_{1,i} = u_1 - c\prod_{j\in[k]} \beta_j^{l_j(0)}$.

(11) $Trace(par,msk,(t,\Gamma_{k,s},m,\sigma),L_U)$: 输入公开参数 par, 主私钥 msk, 签名 $(t,\Gamma_{k,s},m,\sigma)$ 以及用户列表 L_U, 检查列表中是否存在一个元组 (id,pk_{id}), 使得 $B = e(pk_{id},g^{\alpha})\cdot Y^{1/\alpha}$. 如果存在, 输出该用户身份 id.

(12) $Revoke(id,t,rl,st)$: 输入用户身份 id, 时间段 t, 撤销列表 rl 以及状态 st, 对所有用户 id 相关的节点 x, 将 (x,t) 添加到撤销列表 rl, 并输出更新后的撤销列表 rl.

5 安全性分析

以下我们从不可伪造性、匿名性、可追溯性、合理性、隐私性 5 个方面分析 RT-VABTS 方案的安全性.

定理 1. RT-VABTS 不可伪造性. 假设 NIZK 是一个安全的零知识证明, l-DHE 问题在群 G 中是困难的, 那么 RT-VABTS 方案是不可伪造的.

证明: 如果一个敌手攻破了 RT-VABTS 方案, 可以伪造出一个合理的签名, 则可以构造出一个模拟器 $\mathcal{B}$ 来解决 l-DHE 问题或攻破 NIZK 的安全性.

在这里我们考虑两种类型的敌手 $(\mathcal{A}_1,\mathcal{A}_2)$. 敌手 $\mathcal{A}_1$ 代表了没有注册过的、已经撤销的、不满足签名策略的用户. 敌手 $\mathcal{A}_2$ 代表拥有主私钥的密钥颁发机构. 首先, 我们介绍敌手 $\mathcal{A}_1$ 是如何尝试着攻破 RT-VABTS 方案.

挑战者 C 和敌手 $\mathcal{A}_1$ 的做法与第 4.2 节安全定义中一样. 也就是挑战者 C 运行 $Setup$ 算法, 生成公开参数 par 和主私钥 msk. 而后, 将公开参 par 发送敌手 $\mathcal{A}_1$, 并将主私 msk 秘密保存在本地. 敌手 $\mathcal{A}_1$ 可以向挑战者 C 发送多项式次问询, 包括属性密钥询问、签名密钥询问、密钥更新消息询问、签名询问、$Commit$ 询问、$Vrfy$ 询问等.

之后, 敌手 $\mathcal{A}_1$ 输出一个关于属性密钥 sk_{id,A^*}, 消息 m^*, 签名策略 $\Gamma^*_{k,s}$ 的签名 σ^*. 并运行 $Commit(par,\sigma^*,T)$ 算法得到 (C^*,π^*), 如果签名 (C^*,π^*) 通过了 $Vrfy$ 算法, 则称敌手 $\mathcal{A}_1$ 赢得了该游戏. 这里需要注意的是, 在 $KeyGen$ 问询时, 不能问询关于属性集合 A^* 的密钥 sk_{id,A^*}. 在 $Sign$ 问询时, 不能问询关于消息 m^*, 签名策略 $\Gamma^*_{k,s}$ 的签名 σ^*.

由签名 σ^*, 可计算得到:

$$\sigma_0^* = g^{\alpha\beta}\left(h_0\prod_{i=1}^{l} h_i^{b_i^*}\right)^{s_0} F_1(t^*)^{s_1}F_2(m^*)^{s_2} = g^{\alpha\beta}g^{s_0\gamma_0}g^{s_1\left(\delta_0+\sum_{j=1}^{l_1}\delta_j t^*[j]\right)}g^{s_2\left(y_0+\sum_{j=1}^{l_2} y_j m^*[j]\right)} = g^{\alpha\beta}(\sigma'_0)^{\gamma_0}\sigma_1^{*s1\left(\delta_0+\sum_{j=1}^{1}\delta_j t^*[j]\right)}\sigma_2^{*s_2\left(y_0+\sum_{j=1}^{l_2} y_j m^*[j]\right)}.$$

$\mathcal{B}$ 可以得 $g_{l+1} = g^{a^{l+1}} = \left(\sigma_0^*/\left(\sigma_0'^{*}\right)^{y_0}\cdot\sigma_1^{*s1\left(\delta_0+\sum_{j=1}^{l_1}\delta_j t^*[j]\right)}\cdot\sigma_2^{*\left(y_0+\sum_{j=1}^{l_2} y_j m^*[j]\right)}\right)^{1/\alpha'\beta}$.

因 $(\sigma_0,\sigma_1,\sigma_t,\sigma_2)$ 的生成过程与文献 [46] 相同, 详细的安全性分析及概率分析请参考文献 [46].

下面, 我们介绍敌手 $\mathcal{A}_2$ 是如何尝试攻破 RT-VABTS 方案的.

挑战者 C 和敌手 $\mathcal{A}_2$ 的做法与第 4.2 节安全定义中一样. 挑战者 C 运行 $Setup$ 算法, 生成公开参数 par 和主私钥 msk. 而后, 挑战者 C 为每个用户 id 生成公私密钥对 (sk_{id},pk_{id}), 并将 (id,pk_{id}) 存储到用户列表 L_U 中. 最后, C 将公开参数 par, 主私钥 msk 以及用户列表 L_U 发送敌手 $\mathcal{A}_2$. 敌手 $\mathcal{A}_2$ 可以向挑战者 C 发送多项式次用户私钥问询, 挑

战者 C 返回用户 id 的私钥 sk_{id} 给 $\mathcal{A}_2$. 敌手 $\mathcal{A}_2$ 输出一个时间段 t^*, 签名策略 $\Gamma_{k,s}^*$, 消息 m^* 以及签名 σ^*. 并运行 $Commit(par,\sigma^*,T)$ 算法得到 (C^*,π^*), 如果签名 (C^*,π^*) 通过了 $Vrfy$ 算法, 且 $Trace(par,msk,(t^*,\Gamma_{k,s}^*,m^*,\sigma^*),L_U)\to id$, 则称敌手 $\mathcal{A}_2$ 赢得了该游戏. 这里需要注意的是, 在 $KeyGen$ 问询时, 不能问询关于 id^* 的私钥 sk_{id^*}. 因为在计算 B 以及 c,θ_0,θ_1 的过程中需要知晓 id^* 的私钥 sk_{id^*}. 由 NIZK 的安全性可知, 敌手 $\mathcal{A}_2$ 获得 sk_{id^*} 的概率是可忽略的. 具体地, 请参考文献 [30].

综上所述, 没有敌手可以以不可忽略的概率伪造出一个合理的签名. 故 RT-VABTS 方案是不可伪造的.

证毕.

定理 2. 匿名性. 匿名性保证了通过合理的签名, 不能得到关于满足该签名策略 $\Gamma_{k,s}$ 的属性集合的任何其他消息. 换句话说, 敌手不能分辨出满足同一签名策略 $\Gamma_{k,s}$ 的两个属性集合.

证明: 挑战者 C 与敌手 $\mathcal{A}$ 的操作与前面的安全性定义相同. 也就是挑战者 C 运行 $Setup$ 算法, 生成公开参数 par 和主私钥 msk. 而后, 将公开参数 par 发送敌手 $\mathcal{A}$, 并将主私钥 msk 秘密保存在本地. 敌手 $\mathcal{A}$ 可以向挑战者 C 发送多项式次问询, 包括属性密钥询问、签名密钥询问、密钥更新消息询问、签名询问、$Commit$ 询问、$VerifyS$ 询问等. 之后, $\mathcal{A}$ 将两个满足签名策略 $\Gamma_{k,s}^*$ 的两个属性集合 A_0^*, A_1^*, 消息 m^* 发送给挑战者 C. 挑战者 C 选择随机数 $b\leftarrow\{0,1\}$, 为签名策略 $\Gamma_{k,s}^*$ 下的消息 m^*, 签名密钥 sk_{id,A_b^*} 生成签名 σ^*. 并将该签名发送给 $\mathcal{A}$. 最后, 敌手 $\mathcal{A}$ 输出猜测结果 b'. 如果 $b'=b$, 则称 $\mathcal{A}$ 赢得了该游戏.

注意到, 在签名中, $(\sigma_0,\sigma_1,\sigma_t,\sigma_2)$ 的值与属性集合 A 无关. 同样地, $\{b_i\}_{1\leqslant i\leqslant l}$ 的取值也独立于属性集合 A. 签名的其他组成部分 $(R_1,R_2,c,\theta_0,\theta_1,Y=Z^s)$ 同样与 A 无关. 敌手 $\mathcal{A}$ 获胜的概率 $Adv_{\mathcal{A}}^{ANON}(\lambda)=|\Pr[b'=b]-1/2|$ 是可忽略的. 因此, RT-VABTS 方案是匿名的.

证毕.

定理 3. 可追溯性. 这在 $Trace$ 算法中很容易得到证明. 给定一个签名 σ, 检查列表中是否存在一个元组 (id,pk_{id}), 使得 $B=e(pk_{id},g^{\alpha})\cdot Y^{1/\alpha}$. 如果存在, 输出该用户身份 id.

定理 4. 合理性. 如果一个 RT-VABTS 方案是合理的, 那么对于任意的多项式时间敌手 $\mathcal{A}$, 以下概率是可忽略的

$$\Pr\left[\begin{array}{l} b_1=1\wedge b_2=0: \\ \end{array}\begin{array}{c}(par,m,C,\pi,T)\leftarrow\mathcal{A}(1^{\lambda})\\ \sigma\leftarrow ForeOp(C)\\ b_1=Vrfy(par,m,C,\pi)\\ b_2=VerifyS(par,m,\sigma,\Gamma_{k,S})\end{array}\right]\leqslant negl(\lambda).$$

证明: 假设敌手攻破了 RT-VABTS 方案的合理性, 这意味着对所有的 $Z_i\notin I'$, $(Z_1,\ldots,Z_n)$ 有 $HTLP.PSolve(par,Z_i)=\sigma_i$ 满足 $(e(g,\sigma_{0,i})\Big/(e(h_0\prod_{i=1}^{l}h_i^{b_i},\sigma_1)\cdot e(F_1(t),\sigma_t)\cdot e(F_2(m),\sigma_2)))\neq B_i$. 如果相反, 可以恢复出消息 m 的一个合理签名 σ. 进一步, 所有问题 $(Z_1,\ldots,Z_n)$ 都是正确的, 即, 解决算法总是以不可忽略的概率输出一些定义明确的值. 因此, 在给定 $(Z_1,\ldots,Z_n)$ 的条件下, 我们可以在多项式时间内通过解决问题和检查哪个签名满足上述关系来恢复集合 I'. 为了让验证者接受, $I=I'$ 必须成立, 这意味着证明者需要从 $n/2$ 位字符串集合中除去全 0 的字符串中, 猜测出一个 n 位字符串. 这个成功地概率为 $(n/2!)^2\Big/n!$.

证毕.

定理 5. 隐私性. 如果存在一个多项式时间模拟器 $\mathcal{S}$, 一个可忽略函数 $negl$, 所有 PRAM 敌手 $\mathcal{A}$ 的最大运行时间 $t<T$, 且下面不等式成立:

$$\left|\Pr\left[\mathcal{A}(par,m,C,\pi)=1:\begin{array}{c}sk_{id,A}\leftarrow KeyGen(par,msk,A)\\ \sigma\leftarrow Sign(par,m,sk_{id,A},\Gamma_{k,S})\\ (C,\pi)\leftarrow Commit(\sigma,T)\end{array}\right]-\Pr\left[\mathcal{A}(par,m,C,\pi)=1:\begin{array}{c}sk_{id,A}\leftarrow KeyGen(par,msk,A)\\ (C,\pi,m)\leftarrow S(par,T)\end{array}\right]\right|\leqslant negl(\lambda).$$

证明: 下面我们通过一系列的混合实验来证明 RT-VABTS 是隐私的, 且可以抵抗深度为 T^{ε} 的敌手, 其中 $\varepsilon<1$ 为非负数. 然后我们说明邻近实验是不可区分的.

① Hybrid H_0: 同算法定义的一样.

② Hybrid H_1: 除了随机预言机通过惰性抽样模拟之外, 同 H_0 相同. 随机预言机提前取样, 设置并输出一个随机集合 $I^*(|I^*|=t-1)$. 因为只是语法上的改变, 并没有改变分布, 故 H_1 与 H_0 是不可区分的.

③ Hybrid H_2: H_2 模拟了一个公共参数 crs_Ω. 根据 NIZK 的特性, 这个改变是计算不可区分的.

④ Hybrid $H_3 \ldots H_{3+n}$: 对所有的 $i \in [n]$, 在 H_{3+i} 中, π_i 是通过模拟器提供的 NIZK 得到的. 根据 NIZK 的性质, 两个相邻 H_{3+i} 之间的 π_i 是不可区分的.

⑤ Hybrid $H_{3+n} \ldots H_{3+2n-t+1}$: 对所有的 $i \in [n-(t-1)]$, 在 H_{3+i} 中, 集合 I^* 中第 i 个元素对应的问题是由 $HTLP.PGen(pp_{HTLP}, 0^\lambda, r_i)$ 计算得到的. 由于敌手是深度受限的, 由 HTLP 的安全性知, 敌手无法区分相邻 H_{3+i} 的问题.

⑥ Hybrid $H_{3+2n-t+2}$: 对所有的 $i \in I^*$, 选择随机数 $\beta_i \leftarrow \mathbb{Z}_p$, 计算 $B'_i = Z^{\beta_i}$, $B_i = B'_i \cdot V^s$. 对所有 $i \notin I^*$, 计算 $B'_i = \left(Z^\beta / \prod_{j \in [k-1]} B_j^{l_j(0)}\right)^{l_i(0)^{-1}}$, $B_i = B'_i \cdot V^s$. 对所有 $i \notin I^*$, 有 $\prod_{j \in I} B_j^{l_j(0)} \cdot B_i^{l_i(0)} \neq Z^\beta$. 由此可见, 该混合实验的变化只是句法上的, 其分布与前一混合实验相同.

⑦ *Simulator* $\mathcal{S}$: 模拟器的定义与 $H_{3+2n-t+2}$ 相同. 注意到, 没有关于证据的信息被用来计算证明.

证毕.

6 性能评估

在本节中, 我们首先对 RT-VABTS 方案进行了定量分析. 之后, 我们又对 RT-VABTS 方案与相关方案[10,30,42−45]进行了功能性的对比. 最后, 通过一系列仿真实验评估了 RT-VABTS 方案的效率.

(1) 性能分析

令 $|A|$ 表示用户的属性集合中的元素个数, d 表示默认属性集合的大小. Exp 和 $Pair$ 分别表示模幂和配对运算. R 和 N 分别表示撤销用户的数量和用户总数量. RT-VABTS 方案的属性密钥大小为 $O((d+|A|)\cdot\log(N+1))$, 签名密钥大小为 $4(d+|A|)$. 当 $1 \leqslant R \leqslant N/2$, 密钥更新消息的大小为 $O(R\cdot\log(N/R))$; 当 $N/2 \leqslant R \leqslant N$ 时, 密钥更新消息的大小为 $O(N-R)$. 签名的大小为 11 个群 G 中的元素, 约为 364 字节. 签名生成过程中需要执行 $7d+14$ 次模幂操作, 签名验证过程需要执行 $2d+6$ 次模幂操作和 4 次配对操作. 我们首先评估了 *HTLP* 中每个子算法需要的运行时间, 而后评估了 RT-VABTS 方案中承诺和证明的产生需要的时间以及证明验证需要的时间. RT-VABTS 的总体性能分析如表 1 所示.

表 1 RT-VABTS 方案的性能分析

性能类别			数值
存储负担	密钥大小	属性	$O((d+\|A\|)\cdot\log(N+1))$
		签名	$4(d+\|A\|)$
	密钥更新消息大小		$O(R\cdot\log(N/R))$ 或 $O(N-R)$
	签名大小		约364 B
计算负担	签名		$(7d+14)Exp$
	验证		$(2d+6)Exp+4Pair$
计算时间 (s)	*HTLP.PSetup*		5.63
	HTLP.PGen		9.98
	HTLP.PSlove		0.704
	Commit		22.34
	Verify		34.32

(2) 功能对比

如表 2 所示, 文献 [10] 不支持匿名性. 文献 [10] 和文献 [42] 不支持可撤销. 只有 RT-VABTS 和文献 [30] 能够实现可追溯性. 定时性允许签名者在给定的时间内对已知消息上的签名进行锁定, 在执行时间为 T 的顺序计算后, 任何人都可从时间锁中提取出该签名. 定时性是合约的公平签署的重要保证之一. 从表 2 可知, 只有 RT-VABTS 和文献 [10] 能够实现定时性. 总的来说只有 RT-VABTS 能够实现上述所有功能, 即匿名性、不可伪造性、可撤销、可追溯性和定时性.

(3) 实验结果

在本节中, 我们通过几个实验来评估本文方案的性能. 仿真实验代码是使用 C++编写的, 运行在 Intel 处理器为 2.30 GHz 和 8 GB 内存的 Linux 服务器上. 这些实验是在基于配对的密码学库[48] (版本为 PBC 0.5.14) 和 GNU 多精度算法[49]的帮助下完成的. 在实验中, 我们使用 PBC 中的参数 *a.param* 来设置基础字段大小为 320 KB, 分为 16384 块, $\mathbb{Z}_p$ 中一个元素的大小在是 20 B.

① 密钥更新算法的计算负担. 在该部分的实验中, 设定用户总数量 $N=1024$, 默认属性集合的大小为 $d=4$, 阈值 $k=3$. 如图 4 所示, 我们评估撤销用户数量从 1 到 1024 变化时, $KeyUp$ 的计算时间. 发现当撤销用户的数量小于 $(N/2)=512$ 时, 计算时间与撤销用户的数量呈对数关系. 当撤销用户的数量大于 $N/2$ 时, 计算时间与 $N-R$ 呈线性关系. 具体地, 随着撤销用户的数量 1–1024, $KeyUp$ 的计算时间从 0.042–4.4034 s.

表 2 本文方案与相关方案的功能对比

功能	RT-VABTS	文献 [10]	文献 [30]	文献 [42]	文献 [43]	文献 [44]	文献 [45]
匿名性	√	×	√	√	√	√	√
不可伪造性	√	√	√	√	√	√	√
可撤销	√	×	√	×	√	√	√
可追溯性	√	×	√	×	×	×	×
定时性	√	√	×	×	×	×	×

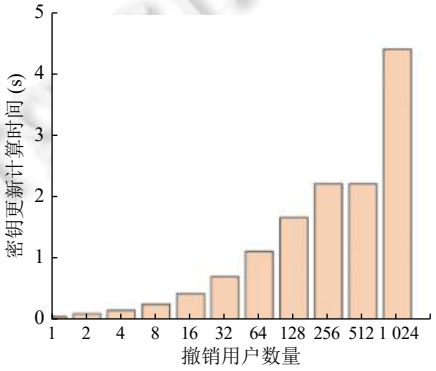


图 4 *KeyUp* 算法的计算开销

② 签名算法的开销. 如图 5 所示, 签名数量从 0 增加到 100 时, RT-VABTS 和文献 [30] 的签名算法的计算时间均呈线性增长. 具体来说, 当签名数量为 0–100 时, RT-VABTS 方案的签名计算时间为 0–3.162 s. 在文献 [30] 中, 签名计算时间为 0–3.157 s. 因此, RT-VABTS 方案在额外支持定时性的条件下, 并未损失签名效率.

③ 签名验证算法的开销. 如图 6 所示, 签名数量从 0 增加到 100 时, RT-VABTS 和文献 [30] 的签名验证算法的计算时间均呈线性增长. 具体来说, 当签名数量为 0–100 时, RT-VABTS 方案的签名验证计算时间为 0–5.02 s. 在文献 [30] 中, 签名计算时间为 0–5.04 s. 因此, RT-VABTS 与文献 [30] 的签名验证所需的时间几乎相同.

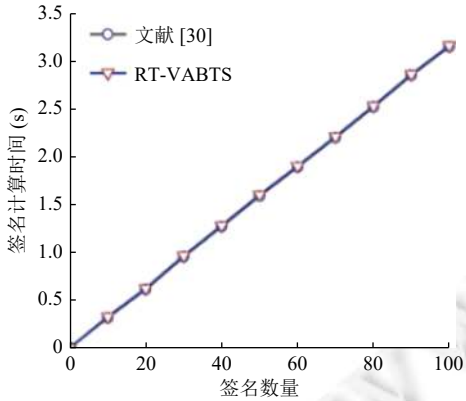


图 5 签名算法的计算开销

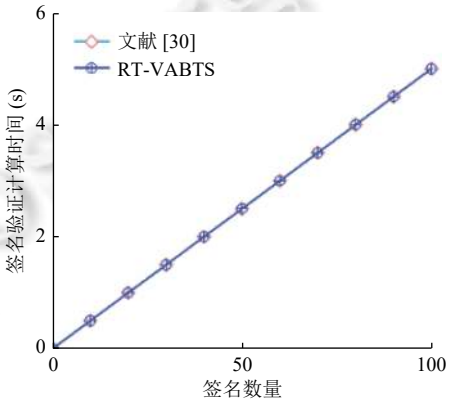


图 6 签名验证算法的计算开销

7 总 结

本文提出了可验证的属性基定时签名 (VABTS) 的概念, 并给出了一个可撤销和追踪的 VABTS 方案 RT-VABTS

的具体构造. 在本文中, RT-VABTS 方案不仅可以同时支持签名者身份隐私保护、动态的用户撤销、可追溯性以及定时性, 还能解决属性基密码中的密钥托管问题. 此外, 本文列举了 VABTS 的两种应用场景: 构建准入区块链中隐私保护的支付通道网络和实现公平的隐私保护的多方计算. 最后, 通过形式化的安全性分析和性能评估证明实例化的 RT-VABTS 方案是安全且高效的.

References:

[1] Rivest RL, Shamir A, Wagner DA. Time-lock puzzles and timed-release crypto. Technical Report, MIT/LCS/TR-684, MIT Laboratory for Computer Science, 1996.

[2] Liu J, Jager T, Kakvi SA, Warinschi B. How to build time-lock encryption. Designs, Codes and Cryptography, 2018, 86(11): 2549–2586. [doi: 10.1007/s10623-018-0461-x]

[3] Malavolta G, Thyagarajan SAK. Homomorphic time-lock puzzles and applications. In: Proc. of the 39th Annual Int'l Cryptology Conf. Santa Barbara: Springer, 2019. 620–649. [doi: 10.1007/978-3-030-26948-7_22]

[4] Katz J, Loss J, Xu JY. On the security of time-lock puzzles and timed commitments. In: Proc. of the 18th Theory of Cryptography Conf. Durham: Springer, 2020. 390–413. [doi: 10.1007/978-3-030-64381-2_14]

[5] Boneh D, Naor M. Timed commitments. In: Proc. of the 20th Annual Int'l Cryptology Conf. Santa Barbara: Springer, 2000. 236–254. [doi: 10.1007/3-540-44598-6_15]

[6] Garay JA, Jakobsson M. Timed release of standard digital signatures. In: Proc. of the 6th Int'l Conf. on Financial Cryptography. Southampton: Springer, 2002. 168–182. [doi: 10.1007/3-540-36504-4_13]

[7] Self decrypting files. 2021. https://www.gwern.net/Self-decrypting-files

[8] Katz J, Miller A, Shi E. Pseudonymous secure computation from time-lock puzzles. 2014. https://www.semanticscholar.org/paper/Pseudonymous-Secure-Computation-from-Time-Lock-Katz-Miller

[9] Lin HJ, Pass R, Soni P. Two-round and non-interactive concurrent non-malleable commitments from time-lock puzzles. In: Proc. of the 58th Annual Symp. on Foundations of Computer Science. Berkeley: IEEE, 2017. 576–587. [doi: 10.1109/FOCS.2017.59]

[10] Thyagarajan SAK, Bhat A, Malavolta G, Döttling N, Kate A, Schröder D. Verifiable timed signatures made practical. In: Proc. of the 2020 ACM SIGSAC Conf. on Computer and Communications Security. Virtual Event: ACM, 2020. 1733–1750. [doi: 10.1145/3372297.3417263]

[11] Boneh D, Lynn B, Shacham H. Short signatures from the Weil pairing. In: Proc. of the 7th Int'l Conf. on the Theory and Application of Cryptology and Information Security. Security Gold Coast: Springer, 2001. 514–532. [doi: 10.1007/3-540-45682-1_30]

[12] Schnorr CP. Efficient identification and signatures for smart cards. In: Proc. of the 1989 Conf. on the Theory and Application of Cryptology. New York: Springer, 1989. 239–252. [doi: 10.1007/0-387-34805-0_22]

[13] Johnson D, Menezes A, Vanstone S. The elliptic curve digital signature algorithm (ECDSA). Int'l Journal of Information Security, 2001, 1(1): 36–63. [doi: 10.1007/s102070100002]

[14] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. 2008. https://bitcoin.org/bitcoin.pdf

[15] Bitcoin wiki: Payment channels. 2021. https://en.bitcoin.it/wiki/Payment_channels

[16] Poon J, Dryja T. The Bitcoin lightning network: Scalable off-chain instant payments. 2016. https://scholar.archive.org/work/on4k2pmakbgenbn7loxrmubroi

[17] Bagaria V, Neu J, Tse D. Boomerang: Redundancy improves latency and throughput in payment-channel networks. In: Proc. of the 24th Int'l Conf. on Financial Cryptography and Data Security. Kota Kinabalu: Springer, 2020. 304–324. [doi: 10.1007/978-3-030-51280-4_17]

[18] Eckey L, Faust S, Hostáková K, Roos S. Splitting payments locally while routing inter dimensionally. IACR Cryptology ePrint Archive 2020, 2020: 555.

[19] Egger C, Moreno-Sanchez P, Maffei M. Atomic multi-channel updates with constant collateral in bitcoin-compatible payment-channel networks. In: Proc. of the 2019 ACM SIGSAC Conf. on Computer and Communications Security. London: ACM, 2019. 801–815. [doi: 10.1145/3319535.3345666]

[20] Sivaraman V, Venkatakrishnan SB, Ruan K, Negi P, Yang L, Mittal R, Fanti G, Alizadeh M. High throughput cryptocurrency routing in payment channel networks. In: Proc. of the 17th USENIX Symp. on Networked Systems Design and Implementation. Santa Clara: USENIX Association, 2020. 777–796.

[21] Malavolta G, Moreno-Sanchez P, Kate A, Maffei M, Ravi S. Concurrency and privacy with payment-channel networks. In: Proc. of the 2017 ACM SIGSAC Conf. on Computer and Communications Security. Dallas: ACM, 2017. 455–471. [doi: 10.1145/3133956.3134096]

[22] Malavolta G, Moreno-Sanchez P, Schneidewind C, Kate A, Maffei M. Anonymous multi-hop locks for blockchain scalability and

interoperability. In: Proc. of the 26th Annual Network and Distributed System Security Symp. San Diego: The Internet Society, 2019. [doi: 10.14722/ndss.2019.23330]

[23] Liu XD, Zhang WF, Wang XM. Multi-authority attribute-based alterable threshold ring signature without central authority. Ruan Jian Xue Bao/Journal of Software, 2018, 29(11): 3528–3543 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5293.htm [doi: 10.13328/j.cnki.jos.005293]

[24] Zhang YH, Hu YP, Chen JS. Hidden attribute-based signatures without anonymity revocation from lattices. Chinese Journal of Computers, 2018, 41(2): 481–492 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2018.00481]

[25] Bentov I, Kumaresan R. How to use Bitcoin to design fair protocols. In: Proc. of the 34th Annual Cryptology Conf. Santa Barbara: Springer, 2014. 421–439. [doi: 10.1007/978-3-662-44381-1_24]

[26] Kumaresan R, Bentov I. How to use Bitcoin to incentivize correct computations. In: Proc. of the 2014 ACM SIGSAC Conf. on Computer and Communications Security. Scottsdale: ACM, 2014. 30–41. [doi: 10.1145/2660267.2660380]

[27] Kumaresan R, Moran T, Bentov I. How to use Bitcoin to play decentralized poker. In: Proc. of the 22nd ACM SIGSAC Conf. on Computer and Communications Security. Denver: ACM, 2015. 195–206. [doi: 10.1145/2810103.2813712]

[28] Sasson EB, Chiesa A, Garman C, Green M, Miers I, Tromer E, Virza M. Zerocash: Decentralized anonymous payments from Bitcoin. In: Proc. of the 2014 IEEE Symp. on Security and Privacy. Berkeley: IEEE, 2014. 459–474. [doi: 10.1109/SP.2014.36]

[29] Lai RWF, Ronge V, Ruffing T, Schröder D, Thyagarajan SAK, Wang JF. Omniring: Scaling private payments without trusted setup. In: Proc. of the 2019 ACM SIGSAC Conf. on Computer and Communications Security. London: ACM, 2019. 31–48. [doi: 10.1145/3319535.3345655]

[30] Cui H, Deng RH, Wang GL. An attribute-based framework for secure communications in vehicular ad hoc networks. IEEE/ACM Trans. on Networking, 2019, 27(2): 721–733. [doi: 10.1109/tnet.2019.2894625]

[31] Mahmoody M, Moran T, Vadhan S. Time-lock puzzles in the random oracle model. In: Proc. of the 31st Annual Cryptology Conf. Santa Barbara: Springer, 2011. 39–50. [doi: 10.1007/978-3-642-22792-9_3]

[32] Bitansky N, Goldwasser S, Jain A, Paneth O, Vaikuntanathan V, Waters B. Time-lock puzzles from randomized encodings. In: Proc. of the 2016 ACM Conf. on Innovations in Theoretical Computer Science. Cambridge: ACM, 2016. 345–356. [doi: 10.1145/2840728.2840745]

[33] Baum C, David B, Dowsley R, Nielsen JB, Oechsner S. TARDIS: Time and relative delays in simulation. IACR Cryptology ePrint Archive 2020, 2020: 537.

[34] Ephraim N, Freitag C, Komargodski I, Pass R. Non-malleable time-lock puzzles and applications. IACR Cryptology ePrint Archive 2020, 2020: 779.

[35] Maji HK, Prabhakaran M, Rosulek M. Attribute-based signatures: Achieving attribute-privacy and collusion-resistance. IACR Cryptology ePrint Archive 2008, 2008: 328.

[36] Li J, Kim K. Attribute-based ring signatures. IACR Cryptology ePrint Archive 2008, 2008: 394.

[37] Shahandashti SF, Safavi-Naini R. Threshold attribute-based signatures and their application to anonymous credential systems. In: Proc. of the 2nd Int'l Conf. on Cryptology in Africa. Gammarth: Springer, 2009. 198–216. [doi: 10.1007/978-3-642-02384-2_13]

[38] Li J, Au MH, Susilo W, Xie DQ, Ren K. Attribute-based signature and its applications. In: Proc. of the 5th ACM Symp. on Information, Computer and Communications Security. Beijing: ACM, 2010. 60–69. [doi: 10.1145/1755688.1755697]

[39] Tang F, Bao JL, Huang YH, Huang D, Wang HL. Multi-authority attribute-based identification scheme. Journal on Communications, 2021, 42(3): 220–228 (in Chinese with English abstract). [doi: 10.11959/j.issn.1000-436x.2021047]

[40] Zhang YH, He JY, Guo R, Zheng D. Server-Aided and verifiable attribute-based signature for industrial Internet of Things. Journal of Computer Research and Development, 2020, 57(10): 2177–2187 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2020.20200421]

[41] Okamoto T, Takashima K. Efficient attribute-based signatures for non-monotone predicates in the standard model. In: Proc. of the 14th Int'l Workshop on Public Key Cryptography. Taormina: Springer, 2011. 35–52. [doi: 10.1007/978-3-642-19379-8_3]

[42] Herranz J, Laguillaumie F, Libert B, Ràfols C. Short attribute-based signatures for threshold predicates. In: Proc. of the 2012 Cryptographers' Track at the RSA Conf. San Francisco: Springer, 2012. 51–67. [doi: 10.1007/978-3-642-27954-6_4]

[43] Lian YL, Xu L, Huang XY. Attribute-based signatures with efficient revocation. In: Proc. of the 5th Int'l Conf. on Intelligent Networking and Collaborative Systems. Xi'an: IEEE, 2013. 573–577. [doi: 10.1109/INCoS.2013.106]

[44] Seo JH, Emura K. Revocable identity-based encryption revisited: Security model and construction. In: Proc. of the 16th Int'l Conf. on Practice and Theory in Public-key Cryptography. Nara: Springer, 2013. 216–234. [doi: 10.1007/978-3-642-36362-7_14]

[45] Wei JH, Huang XY, Hu XX, Liu WF. Revocable threshold attribute-based signature against signing key exposure. In: Proc. of the 11th

Int'l Conf. on Information Security Practice and Experience. Beijing: Springer, 2015. 316–330. [doi: 10.1007/978-3-319-17533-1_22]
[46] Boldyreva A, Goyal V, Kumar V. Identity-based encryption with efficient revocation. In: Proc. of the 15th ACM Conf. on Computer and Communications Security. Alexandria: ACM, 2008. 417–426. [doi: 10.1145/1455770.1455823]
[47] Shamir A. How to share a secret. Communications of the ACM, 1979, 22(11): 612–613. [doi: 10.1145/359168.359176]
[48] Pairing-Based Cryptography (PBC) Library. 2020. https://crypto.stanford.edu/pbc/howto.html
[49] GMP. The GNU multiple precision arithmetic library. 2020. http://gmplib.org

附中文参考文献:

[23] 刘旭东, 张文芳, 王小敏. 分布式无中心授权的属性基可变门限环签名. 软件学报, 2018, 29(11): 3528–3543. http://www.jos.org.cn/1000-9825/5293.htm [doi: 10.13328/j.cnki.jos.005293]
[24] 张彦华, 胡予濮, 陈江山. 格上无匿名性撤销的隐藏的属性签名. 计算机学报, 2018, 41(2): 481–492. [doi: 10.11897/SP.J.1016.2018.00481]
[39] 唐飞, 包佳立, 黄永洪, 黄东, 王惠莅. 基于属性的多授权中心身份认证方案. 通信学报, 2021, 42(3): 220–228. [doi: 10.11959/j.issn.1000-436x.2021047]
[40] 张应辉, 贺江勇, 郭瑞, 郑东. 工业物联网中服务器辅助且可验证的属性基签名方案. 计算机研究与发展, 2020, 57(10): 2177–2187. [doi: 10.7544/issn1000-1239.2020.20200421]

侯慧莹 (1992－), 女, 博士生, CCF 学生会员, 主要研究领域为应用密码学, 信息安全, 车联网安全, 属性基密码.

黄欣沂 (1981－) 男, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为密码学, 网络安全.

宁建廷 (1988－), 男, 博士, 研究员, 主要研究领域为密码学, 数据安全.

赵运磊 (1974－) 男, 博士, 教授, 博士生导师, 主要研究领域为后量子密码, 密码协议, 计算理论.