

可撤销的可验证定时签名方案的设计及其拓展

张艳硕, 蔡润泽, 邹雨耀, 董有恒, 王彩冰

(北京电子科技学院密码科学与技术系, 北京市 100070)

摘要: 可验证定时签名通过引入时间延迟机制实现签名的延迟公开,并允许验证者在满足预定时间条件后验证或恢复签名.现有 VTS 方案主要围绕签名的延迟公开与强制打开展开,为需要时间约束的签名场景提供了有效的解决方案.然而,传统 VTS 缺乏针对已提交但尚未公开的特定签名进行动态状态管理的能力,难以满足部分应用场景下对签名撤销的需求.针对上述问题,本文形式化提出可撤销可验证定时签名,给出其形式化定义与安全模型,并构造了一个具体实例化方案.安全性分析表明,该实例化方案在所定义的安全模型下满足正确性,不可伪造性,可靠性,隐私性,策略可靠性与策略完备性等安全性质.在此基础上,本文将撤销行为抽象为可计算的策略函数,通过调整策略函数的定义实现不同的撤销策略,并针对不同撤销策略给出相应的应用场景,从而为定时签名提供更加灵活的签名状态管理机制.

关键词: 可撤销可验证定时签名; 可验证定时签名; 可验证延迟函数

中图分类号: TP309

文献标志码: A

doi: 10.11959/j.issn.1000

Design and Extension of Revocable Verifiable Timed Signature Scheme

Zhang Yanshuo, Cai Runze, Zou Yuyao, Dong Youheng, Wang Caibing

Department of Cryptology Science and Technology, Beijing Electronic Science and Technology Institute, Beijing 100070, China

Abstract: The Verifiable Timed Signature achieves the delayed disclosure of the signature by introducing a time-delay mechanism, and allows the verifier to verify or restore the signature after meeting the predetermined time conditions. The existing VTS schemes mainly focus on the delayed disclosure of the signature and the enforcement of it, providing effective solutions for signature scenarios with time constraints. However, the traditional VTS lacks the ability to dynamically manage the state of specific signatures that have been submitted but not yet disclosed, making it difficult to meet the requirements for signature revocation in some application scenarios. To address these issues, we formally propose the Revocable Verifiable Timed Signature, provides its formal definition and security model, and constructs a specific instantiation scheme. The security analysis shows that this instantiation scheme satisfies the security properties such as correctness, unforgeability, soundness, privacy, policy soundness, and policy completeness under the defined security model. Based on this, we abstract the revocation behavior as a computable policy function, adjusts the definition of the policy function to achieve different revocation strategies, and provides corresponding application scenarios for different revocation strategies, thereby providing a more flexible signature state management mechanism for timed signatures.

收稿日期: XXXX-XX-XX; 修回日期: XXXX-XX-XX

通信作者: 张艳硕 zhang_yanshuo@163.com

基金项目: 国家密码科学基金资助项目(No.2025NCSF02028);国家重点研发计划项目(No.2024YFB3108103);中央高校基本科研业务费资金资助(No.3282024043)

Foundation Items: Supported by the National Cryptography Science Foundation-funded Project (No.2025NCSF02028); National Key Research and Development Program of China (No.2024YFB3108103); Fundamental Research Funds for the Central Universities (No.3282024043)

Key words: Revocable Verifiable Timed Signature, Verifiable Timed Signature, Verifiable Delay Function

0 引言

数字签名是保障数字内容真实性、完整性和不可否认性的核心密码学工具,已广泛应用于电子合同、数字版权管理、电子商务以及金融交易等领域。随着应用场景的不断拓展,签名机制的功能需求也日趋多样化^[1-2]。在部分应用场景中,签名的发布不仅需要保证内容的真实性与完整性,还需要满足明确的时间约束——数字版权授权文件可能需要在特定时间后生效,电子合同可能需要在约定日期正式公开,而密封投标中的签名则需在开标时间到达后方可揭晓。在此类场景下,传统数字签名虽然能够证明内容来源,却无法实现签名信息的定时释放^[3]。

为解决上述问题,研究人员提出了时间锁机制及相关定时签名技术。Rivest 等^[4]提出的时间锁谜题(Time-Lock Puzzle, TLP)利用本质顺序计算实现信息的定时释放,为后续研究^[5]奠定了基础。随后, Boneh 等^[6]提出 Timed Commitment 方案,将时间锁机制与密码学承诺相结合,实现了承诺内容的延迟公开。Boneh 等人^[7]进一步提出可验证延迟函数(Verifiable Delay Function, VDF),使延迟计算结果能够被高效验证^[8-11]; Thyagarajan 等人^[12]提出可验证定时签名(Verifiable Timed Signature, VTS)框架,实现了签名的定时释放与公开验证。近年来,国内外学者围绕定时签名在属性基^[13]、格密码^[14-15]、代理签名^[16]、抗量子^[17-18]等方向开展了相关研究,提出了多种可验证定时签名方案,并将其应用于数字版权保护、电子拍卖、数据共享以及访问控制等场景^[19]。这些研究不断丰富完善了定时签名的理论体系,也推动了其在实际应用中的发展。

然而,现有 VTS 方案主要关注签名的延迟公开与强制打开,对于已经完成承诺但尚未公开的特定签名,缺乏动态的状态管理机制。在数字版权保护、电子合同以及风险交易等场景中,签名可能在进入定时释放过程后,由于密钥泄露、交易状态变化或其他原因而需要在正式公开前被撤销。传统 VTS 中签名的有效性主要由其自身的签名验证结果决定,一旦签名被公开,便难以针对该特定签名进一步施加撤销控制。因此,如何在保留 VTS 定

时释放、主动打开与强制打开能力的基础上,为特定定时签名引入可验证的撤销机制,并使撤销规则能够适应不同应用需求,是本文关注的主要问题。

针对上述问题,本文提出可撤销可验证定时签名(Revocable Verifiable Timed Signature, RVTS),旨在将定时签名的时间控制能力与针对特定签名实例的动态状态控制能力相结合。本文的核心贡献并非简单地在现有定时签名中增加撤销操作,而是将撤销过程形式化为由事件证明支撑、由策略函数 P 统一判定的可验证状态转换机制。在该框架下,签名从生成、承诺、撤销到主动打开或强制打开形成完整的生命周期,签名的最终有效性由时间条件、撤销状态及预定义策略共同决定。基于这一框架,本文进一步给出一个具体实例化方案,并从形式化安全模型、具体安全证明和理论效率分析三个层面对方案进行研究;同时,通过构造不同的撤销策略函数,将 RVTS 从固定撤销规则扩展为具有策略可配置性的定时签名机制,并针对不同策略讨论其适用的应用场景。

本文的主要工作如下:

(1) 建立可撤销可验证定时签名的形式化模型与安全框架。

本文在 VTS 的基础上引入撤销机制,形式化定义 RVTS 的系统模型及算法框架,并建立相应的安全模型。从正确性、不可伪造性、可靠性、隐私性、策略可靠性与策略完备性六个方面对方案的安全要求进行形式化描述,为后续实例化构造与安全性分析提供理论基础。其中,策略可靠性与策略完备性用于刻画撤销策略执行过程中的安全性与正确性。

(2) 提出一种基于 BLS 签名^[20]、zkVDF^[21]与 Merkle^[22-24]事件日志的 RVTS 实例化方案。

本文利用 BLS 签名生成待定时释放的数字签名,通过 zkVDF 构造时间锁承诺,并利用零知识证明使验证者能够验证承诺的有效性;同时利用 Merkle 事件日志记录撤销与打开事件,并生成相应的事件证明。基于“事件证明+策略评估”的方式,方案能够根据预先定义的策略函数判断签名的最终有效状态。

(3) 提出不同的撤销策略并分析其应用场景。

本文定义强撤销模型与时间窗口撤销模型两类典型撤销策略。强撤销模型规定撤销一旦发生即永久有效,而时间窗口撤销模型规定撤销仅在签名通过 Open 或达到 ForceOpen 所对应的时间 T 之前有效。针对不同的撤销语义,本文进一步给出相应的应用场景,从而说明 RVTS 通过调整策略函数 P 能够适应不同的签名状态管理需求。

本文内容安排如下:第一节介绍本文所需的预备知识,包括双线性配对与 BLS 签名、可验证定时签名、可验证延迟函数以及非交互式零知识证明;第二节给出 RVTS 的系统模型与安全模型,形式化定义方案涉及的算法及安全性质;第三节给出 RVTS 的具体实例化构造;第五节对方案进行安全性分析、效率分析,并与现有相关工作进行对比;第五节针对不同撤销策略讨论相应的应用场景;第六节对全文工作进行总结。

1 预备知识

1.1 BLS 签名

BLS (Boneh-Lynn-Shacham) [20] 签名方案由密钥生成、签名和验证三个算法组成。

(1) 密钥生成: 随机选择 $x \xleftarrow{\$} \mathbb{Z}_q$, 计算 $pk = g_2^x$ 。其中 x 为私钥, pk 为公钥。

(2) 签名: 设 $H: \{0,1\}^* \rightarrow G_1$ 为密码学哈希函数。对于消息 m , 签名者计算 $\sigma = H(m)^x$ 。

(3) 验证: 给定 m 、 σ 和 pk , 验证者检查 $e(\sigma, g_2) \stackrel{?}{=} e(H(m), pk)$, 其中 e 为一个双线性配对。若等式成立, 则接受签名。

本文利用 BLS 签名提供签名者身份认证及撤销授权的真实性。

1.2 可验证定时签名

可验证定时签名 (Verifiable Timed Signature, VTS) [12] 是一类结合时间锁机制与数字签名的密码学原语, 其核心目标是在签名生成后引入预定的时间延迟, 使签名在满足时间条件之前无法被公开获得, 同时允许验证者在签名公开后验证其真实性。

一般而言, VTS 包含签名生成、时间锁承诺、验证、打开以及强制打开等过程。签名者首先生成签名 σ , 随后对其进行时间锁承诺: $C \leftarrow \text{Commit}(\sigma, T)$, 其中 T 为时间参数。签名者可以在满足相应条件后主动打开承诺; 若签名者不主

动打开, 则其他参与方可以通过强制打开过程恢复签名。

与传统数字签名相比, VTS 的主要区别在于引入了签名公开时间的控制, 使签名的生成与公开在时间维度上相互分离。

本文在 VTS 的基础上进一步引入撤销机制, 使签名在完成承诺后、最终公开前仍能够发生状态变化。

1.3 可验证延迟函数

可验证延迟函数 (Verifiable Delay Function, VDF) [7] 是一类具有显式计算延迟和高效验证特性的函数。给定输入 x 和时间参数 T , VDF 的计算过程需要执行具有顺序性的计算, 而其输出可以通过较低的计算开销进行验证。

一个 VDF 通常包含两个算法, $(y, \pi) \leftarrow \text{Eval}(x, T)$ 以及 $\text{Verify}(x, y, \pi, T) \rightarrow \{0, 1\}$ 。

其中, Eval 根据输入 x 和时间参数 T 计算输出 y 及证明 π , 而 Verify 用于验证 y 是否为正确的 VDF 输出。

VDF 的核心性质是: 计算输出需要经过预定的顺序计算过程, 而验证过程不需要重复执行完整计算。因此, VDF 可以用于构造具有时间延迟特性的密码学机制。

本文利用 VDF 实现定时签名中的时间锁功能, 使承诺中的签名在达到预定时间条件前无法被高效恢复。

1.4 非交互式零知识证明

非交互式零知识证明 (Non-Interactive Zero-Knowledge Proof, NIZK) [25-26] 允许证明者在不泄露秘密信息本身的情况下, 使验证者相信某个公开声明成立。

设公开实例为 x , 秘密见证为 w , 且满足关系 $R(x, w) = 1$ 。证明者通过 $\pi \leftarrow \text{Prove}(x, w)$ 生成证明 π , 验证者执行 $\text{Verify}(x, \pi) \rightarrow \{0, 1\}$ 。

本文利用 NIZK 证明时间锁承诺及其相关计算的正确性, 使验证者能够验证承诺与隐藏签名之间的关系, 而无需直接获得隐藏信息。

2 可撤销可验证定时签名

本节对可撤销可验证定时签名 (Revocable Verifiable Timed Signature, RVTS) 进行形式化定义。首先给出 RVTS 的系统模型及完整算法框架,

明确各算法的输入、输出及执行关系；随后给出方案的安全模型，并对相关安全性质进行形式化描述，为后续具体实例化及安全性证明提供理论基础。

2.1 算法定义

本小节给出 RVTS 的系统模型，明确参与实体的角色划分、形式化算法定义及各算法的功能语义。

(1) 协议参与方

一个 RVTS 系统包含以下两类参与实体：

签名者 (Signer)： 持有长期私钥 sk ，是唯一能够执行签名生成算法 Sign 和撤销操作算法 Revoke 的实体。签名者负责对消息生成数字签名、构造时间锁承诺，并在承诺发布后的任意时刻根据需要发起撤销。在主动打开场景中，签名者还可执行 Open 算法提前释放签名。

其他参与者 (Any Other Party)： 包括接收者、验证者以及任何第三方。该类实体不持有签名者私钥，但拥有系统公开参数 pp 和公钥 pk 。任何其他参与者均可独立执行以下操作：

- 1) 通过 Vrfy 算法在不解开时间锁的情况下验证承诺的有效性；
- 2) 在时间参数 T 对应的顺序计算完成后，通过 ForceOpen 算法强制恢复签名；
- 3) 通过 Vf 算法验证恢复出的签名是否正确；
- 4) 收集公开的撤销事件证明 π_R 与打开事件证明 π_O ，运行 Eval 算法生成策略证明 π_P ，并最终通过 Accept 算法判定签名的最终有效性。

(2) 算法流程定义

定义 1 (可撤销可验证定时签名, RVTS) 设

$$\Pi_{\text{DS}} = (\text{Setup}, \text{Sign}, \text{Vf})$$

为一个数字签名方案，其中 Setup 生成系统公开参数、签名者公私钥对及撤销策略函数。一个可撤销可验证定时签名方案 RVTS 是 Π_{DS} 之上的算法元组：

$$\text{RVTS} = (\text{Commit}, \text{Vrfy}, \text{Open}, \text{ForceOpen}, \text{Revoke}, \text{Eval}, \text{Accept}),$$

各算法的具体定义如下：

- 1) $(C, \pi) \leftarrow \text{Commit}(pk, m, \sigma, T)$ ：承诺算法以公钥 pk 、消息 m 、签名 σ 和时间参数 T 为输入，输出时间锁承诺 C 及其有效性证明 π ；

2) $\{0,1\} \leftarrow \text{Vrfy}(pk, m, C, \pi)$ ：验证算法在不解开时间锁的情况下，验证 C 中是否包含消息 m 的有效签名；

3) $(\sigma, r, \pi_O) \leftarrow \text{Open}(C)$ ：打开算法由签名者执行，输入承诺 C ，输出被隐藏的签名 σ 及打开事件证明 π_O ；

4) $\sigma \leftarrow \text{ForceOpen}(C)$ ：强制打开算法由任意实体执行，经过 T 步顺序计算后从 C 中提取签名 σ ；

5) $(\rho, \pi_R) \leftarrow \text{Revoke}(sk, C)$ ：撤销算法由签名者执行，输入私钥 sk 和承诺 C ，输出撤销凭证 ρ (可选) 及撤销事件证明 π_R ；

6) $\pi_P \leftarrow \text{Eval}(\pi_R, \pi_O, T, P)$ ：策略评估算法输入撤销事件证明、打开事件证明、时间参数及策略函数，输出策略证明 π_P ；

7) $\{0,1\} \leftarrow \text{Accept}(pk, m, \sigma, \pi_P)$ ：最终验证算法输入公钥、消息、签名和策略证明，输出 1 表示签名最终有效。

运行流程。 一个完整的 RVTS 方案运行流程如下：

- 1) $(pp, sk, pk, P) \leftarrow \text{Setup}(1^\lambda)$ ：系统初始化；
- 2) $\sigma \leftarrow \text{Sign}(sk, m)$ ：生成签名；
- 3) $(C, \pi) \leftarrow \text{Commit}(pk, m, \sigma, T)$ ：构造时间锁承诺；
- 4) $\{0,1\} \leftarrow \text{Vrfy}(pk, m, C, \pi)$ ：验证承诺有效性；
- 5) $(\sigma, r, \pi_O) \leftarrow \text{Open}(C)$ 及 $\sigma \leftarrow \text{ForceOpen}(C)$ ：主动释放签名或强制打开；
- 6) $(\rho, \pi_R) \leftarrow \text{Revoke}(sk, C)$ ：发起撤销；
- 7) $\pi_P \leftarrow \text{Eval}(\pi_R, \pi_O, T, P)$ ：评估策略；
- 8) $\{0,1\} \leftarrow \text{Accept}(pk, m, \sigma, \pi_P)$ ：最终验证。

2.2 安全模型

本小节建立 RVTS 方案的安全模型。根据第 2.1 小节的形式化定义，RVTS 在标准 VTS 的定时释放与可验证性基础上引入了撤销机制。因此，其安全模型不仅需要涵盖底层签名与时间锁承诺的基本安全要求^[6,12]，还需额外刻画撤销相关的策略层面安全性质。

以下分别从正确性、不可伪造性、可靠性、隐私性、策略可靠性与策略完备性六个方面给出形式化定义。设 λ 为安全参数，所有算法均在多项式时

间内运行, $\text{negl}(\lambda)$ 表示可忽略函数。

(1) 正确性

正确性要求所有参与方按照协议诚实执行时, 系统功能完备: 承诺能够被正确打开, 强制打开能够在时间到达后恢复签名, 承诺验证能够正确判定承诺有效性, 且最终验证能够结合策略证明做出正确判定。

定义 2 (正确性, Correctness) 对于任意安全参数 λ 、消息 $m \in \{0,1\}^*$ 和时间参数 $T \in \mathbb{N}$, 设

$$(pp, sk, pk, P) \leftarrow \text{Setup}(1^\lambda), \sigma \leftarrow \text{Sign}(sk, m),$$

$(C, \pi) \leftarrow \text{Commit}(pk, m, \sigma, T)$. 则以下条件同时成立:

1) 签名验证正确性: $\Pr[\text{Vf}(pk, m, \sigma) = 1] = 1$;

2) 承诺验证正确性: $\Pr[\text{Vrfy}(pk, m, C, \pi) = 1] = 1$;

3) 主动打开正确性: 若 $(\sigma', r, \pi_O) \leftarrow \text{Open}(C)$, 则 $\Pr[\sigma' = \sigma] = 1$;

4) 强制打开正确性: 若 $\sigma' \leftarrow \text{ForceOpen}(C)$, 则 $\Pr[\sigma' = \sigma] \geq 1 - \text{negl}(\lambda)$;

5) 最终验证正确性: 对于任意由合法撤销生成的 $(\rho, \pi_R) \leftarrow \text{Revoke}(sk, C)$ 与合法打开生成的 $(\sigma, r, \pi_O) \leftarrow \text{Open}(C)$, 以及评估后的策略证明 $\pi_P \leftarrow \text{Eval}(\pi_R, \pi_O, T, P)$:

若撤销成立 $P(\pi_R, \pi_O, T) = 1$, 则 $\Pr[\text{Accept}(pk, m, \sigma, \pi_P) = 0] = 1$;

若撤销不成立或未进行撤销 $P(\pi_R, \pi_O, T) = 0$, 则 $\Pr[\text{Accept}(pk, m, \sigma, \pi_P) = 1] = 1$ 。

(2) 不可伪造性

不可伪造性要求任何概率多项式时间敌手即使能够获得多个合法消息-签名对及其对应承诺, 也无法伪造一个能够通过签名验证的新签名。

定义 3 (不可伪造性, Unforgeability) 定义如下敌手游戏 $\text{Exp}_{\mathcal{A}}^{\text{UF}}(\lambda)$:

1) 初始化。挑战者运行 $(pp, sk, pk, P) \leftarrow \text{Setup}(1^\lambda)$, 将 (pp, pk, P) 发送给敌手 $\mathcal{A}$;

2) 查询阶段。敌手 $\mathcal{A}$ 自适应地向签名预言机 $\mathcal{O}_{\text{Sign}}$ 查询任意消息 m , 获得对应的签名

$\sigma \leftarrow \text{Sign}(sk, m)$ 。设查询消息集合为 Q ;

3) 伪造阶段。敌手 $\mathcal{A}$ 输出 $(m^*|\sigma^*)$, 其中 $m^* \notin Q$;

4) 判定。若 $\text{Vf}(pk, m^*, \sigma^*) = 1$, 则实验输出 1 (敌手获胜); 否则输出 0。

定义敌手优势为

$\text{Adv}_{\mathcal{A}}^{\text{UF}}(\lambda) = \Pr[\text{Exp}_{\mathcal{A}}^{\text{UF}}(\lambda) = 1]$. 若对于任意概率多项式时间敌手 $\mathcal{A}$, 均有 $\text{Adv}_{\mathcal{A}}^{\text{UF}}(\lambda) \leq \text{negl}(\lambda)$, 则称 RVTS 方案满足不可伪造性。

(3) 可靠性

可靠性要求: 若承诺验证算法 Vrfy 接受一个承诺, 则强制打开该承诺所恢复的签名必须能通过底层签名验证算法 Vf 。换言之, 敌手无法构造一个“承诺验证通过但强制打开后签名无效”的承诺。

定义 4 (可靠性, Soundness) 称 RVTS 方案满足可靠性, 若存在可忽略函数 negl , 使得对于任意概率多项式时间敌手 $\mathcal{A}$, 均有

$$\Pr \left[\begin{array}{l} (pk, m, C, \pi, T) \leftarrow \mathcal{A}(1^\lambda) \\ (\sigma, r) \leftarrow \text{ForceOpen}(C) \\ b_1 := \text{Vrfy}(pk, m, C, \pi) \\ b_2 := \text{Vf}(pk, m, \sigma) \end{array} : b_1 = 1 \wedge b_2 = 0 \right] \leq \text{negl}(\lambda).$$

该定义刻画了 Vrfy 与 ForceOpen 之间的语义一致性: 承诺验证算法不能接受一个“打开后无效”的承诺。若上式成立, 则称 RVTS 方案满足可靠性。

(4) 隐私性

隐私性要求: 在时间参数 T 对应的顺序计算完成之前, 任何计算能力有限的敌手无法从承诺中区分被隐藏的签名内容。形式化地, 要求存在一个模拟器能够在不知道签名的情况下生成与真实承诺计算不可区分的视图。

定义 5 (隐私性, Privacy) 称 RVTS 方案满足隐私性, 若存在概率多项式时间模拟器 $\mathcal{S}$ 、可忽略函数 negl 以及多项式 $\tilde{T}$, 使得对于所有 $T > \tilde{T}$ 、所有运行时间至多 $t < T$ 的概率 RAM 算法 $\mathcal{A}$ 、所有消息 $m \in \{0,1\}^*$ 以及所有 $\lambda \in \mathbb{N}$, 均有

$$\left| \begin{array}{l} \Pr \left[\begin{array}{l} \mathcal{A}(pk, m, C, \pi) = 1 : \quad (pp, sk, pk, P) \leftarrow \text{Setup}(1^\lambda) \\ \sigma \leftarrow \text{Sign}(sk, m) \\ (C, \pi) \leftarrow \text{Commit}(pk, m, \sigma, T) \end{array} \right] \\ - \Pr \left[\begin{array}{l} \mathcal{A}(pk, m, C, \pi) = 1 : \quad (pp, sk, pk, P) \leftarrow \text{Setup}(1^\lambda) \\ (C, \pi) \leftarrow \mathcal{S}(pk, T) \end{array} \right] \end{array} \right| \leq \text{negl}(\lambda)$$

其中模拟器 $\mathcal{S}$ 输入公钥 pk 和时间参数 T ，输出模拟承诺 C 及其证明 π ，该模拟承诺中不包含任何有效签名。敌手 $\mathcal{A}$ 的运行时间 t 严格小于 T ，反映其在时间参数到达之前计算能力不足以完成强制打开。若上式成立，则称 RVTS 方案满足隐私性。

(5) 策略可靠性

策略可靠性要求任何敌手无法生成一个使非法撤销（即不满足策略函数 P 的撤销）通过最终验证的策略证明。该性质是 RVTS 撤销机制的核心安全保证，防止敌手绕过策略约束使合法签名被错误撤销。

定义 6（策略可靠性，Policy Soundness） 定义如下敌手游戏 $\text{Exp}_{\mathcal{A}}^{\text{PS}}(\lambda)$ ：

1) **初始化**。挑战者运行 $(pp, sk, pk, P) \leftarrow \text{Setup}(1^\lambda)$ ，将 (pp, pk, P) 发送给敌手 $\mathcal{A}$ 。挑战者进一步生成消息 m 的签名 $\sigma \leftarrow \text{Sign}(sk, m)$ 及承诺 $(C, \pi) \leftarrow \text{Commit}(pk, m, \sigma, T)$ ，并将 (m, C) 发送给敌手；

2) **敌手输出**。敌手 $\mathcal{A}$ 输出 $(\tilde{\sigma}, \tilde{\pi}_p)$ ；

3) **判定**。若同时满足以下条件，则实验输出 1（敌手获胜），否则输出 0：

(a) $\text{Vf}(pk, m, \tilde{\sigma}) = 1$ （签名有效）；

(b) $\text{Accept}(pk, m, \tilde{\sigma}, \tilde{\pi}_p) = 1$ （最终验证通过）；

(c) 撤销条件不满足策略函数 P ，即不存在合法的 π_R, π_O 使得 $P(\pi_R, \pi_O, T) = 1$ 能证明本次撤销成立。

定义敌手优势为

$\text{Adv}_{\mathcal{A}}^{\text{PS}}(\lambda) = \Pr[\text{Exp}_{\mathcal{A}}^{\text{PS}}(\lambda) = 1]$ 。若对于任意概率多项式时间敌手 $\mathcal{A}$ ，均有 $\text{Adv}_{\mathcal{A}}^{\text{PS}}(\lambda) \leq \text{negl}(\lambda)$ ，则称 RVTS 方案满足策略可靠性。

(6) 策略完备性

策略完备性要求：当撤销事件满足策略函数 P 时，诚实生成的策略证明 π_p 能够通过最终验证，

即合法的撤销不会被 Accept 算法错误拒绝。

定义 7（策略完备性，Policy Completeness）。对于任意安全参数 λ 、消息 m 、时间参数 T ，设

$(pp, sk, pk, P) \leftarrow \text{Setup}(1^\lambda), \sigma \leftarrow \text{Sign}(sk, m), (C, \pi) \leftarrow \text{Commit}(pk, m, \sigma, T)$ 。设签名者执行撤销 $(\rho, \pi_R) \leftarrow \text{Revoke}(sk, C)$ 与打开 $(\sigma, \pi_O) \leftarrow \text{Open}(C)$ ，且 $P(\pi_R, \pi_O, T) = 1$ 。令 $\pi_p \leftarrow \text{Eval}(\pi_R, \pi_O, T, P)$ ，则

$\Pr[\text{Accept}(pk, m, \sigma, \pi_p) = 0] \leq \text{negl}(\lambda)$ 。即合法的

撤销以压倒性概率被 Accept 正确接受（使签名失效）。若上式成立，则称 RVTS 方案满足策略完备性。

3 方案设计

本节基于前文定义的 RVTS 框架，给出一个具体的实例化构造。该构造采用 BLS 短签名方案生成数字签名，以 MinRoot^[27] 为底层 VDF，并结合零知识证明机制^[25,28]构造 zkVDF，实现签名的定时承诺及其可验证性；同时采用基于 Merkle 树的事件日志结构记录撤销与打开事件，并通过策略函数实现不同的撤销语义。

3.1 签名与时间锁承诺生成

(1) 系统初始化(Setup)

1) 系统初始化算法用于生成 RVTS 的公共参数、签名者密钥对、事件日志初始状态及撤销策略。设 λ 为安全参数，则 $(pp, sk, pk, P, R_0) \leftarrow \text{Setup}(1^\lambda)$ 。其中，首先生成 BLS 签名方案所需的双线性配对参数 $pp_{\text{BLS}} = (G_1, G_2, G_T, e, g_1, g_2, H)$ 其中 G_1, G_2, G_T 为同阶循环群， $e: G_1 \times G_2 \rightarrow G_T$ 为双线性配对， $g_1 \in G_1, g_2 \in G_2$ 为生成元， $H: \{0,1\}^* \rightarrow G_1$ 为密码学哈希函数。随机选择签名者私钥 $sk \xleftarrow{\$} \mathbb{Z}_q$ ，并计算公钥 $pk = g_2^{sk}$ 。

2) 随后生成 MinRoot VDF 所需的公共参数

$pp_{\text{vdf}} \leftarrow \text{MinRoot.Setup}(1^\lambda)$, 并初始化 Merkle 事件日志, 其初始根记为 R_0 。根据应用需求确定公开可计算的撤销策略函数 $P(\pi_R, \pi_O, T) \rightarrow \{0, 1\}$ 其中 π_R 和 π_O 分别表示撤销事件证明和打开事件证明, T 为时间参数, $P = 1$ 表示撤销条件成立。

3) 最终公共参数记为

$$pp = (pp_{\text{BLS}}, pp_{\text{vdf}}, R_0, P).$$

(2) 签名生成 (Sign)

签名生成算法由签名者执行, 输入私钥 sk 和消息 m , 生成消息的 BLS 签名: $H(m)^{sk}$. 其中, $H(m)$ 为消息 m 的哈希值, 签名 σ 由签名者私钥 sk 计算得到。对应的签名验证算法 Vf 检查 $e(\sigma, g) \stackrel{?}{=} e(H(m), pk)$, 若等式成立, 则输出 1, 否则输出 0。

(3) 时间锁承诺 (Commit)

时间锁承诺算法由签名者执行, 输入公钥 pk 、消息 m 、签名 σ 和时间参数 T , 生成隐藏签名的时间锁承诺及其零知识证明: $(C, \pi_{zk}) \leftarrow \text{Commit}(pk, m, \sigma, T)$ 。

1) 首先随机选择 $r \leftarrow \{0, 1\}^\lambda$, 并将签名 σ 与随机数 r 编码为 MinRoot VDF 的输入 $x = \text{Encode}(\sigma, r)$ 。随后执行 T 轮 MinRoot 顺序计算, 得到输出 y 及其 VDF 证明 π_{VDF} :

$(y, \pi_{\text{VDF}}) \leftarrow \text{MinRoot.Eval}(pp_{\text{vdf}}, x, T)$. 将 $C = (y, \pi_{\text{VDF}})$ 作为公开的时间锁承诺。

2) 同时, 利用零知识证明生成算法构造证明 π_{zk} , 证明存在签名 σ 、随机数 r 及 VDF 输入 x , 使得该签名对消息 m 有效, 且 x 由 (σ, r) 正确编码, 并满足 y 为 x 经过 T 轮 MinRoot 计算所得。即 π_{zk} 所证明的关系为

$$\exists \sigma, r, x: \quad \text{Vf}(pk, m, \sigma) = 1 \wedge x = \text{Encode}(\sigma, r) \wedge y = \text{MinRoot}^T(x).$$

3) 最终输出时间锁承诺 C 及其零知识证明 π_{zk} 。其中, 随机数 r 用于增强签名隐藏过程的随机性, 避免攻击者通过对签名空间进行离线枚举直接判断承诺内容; π_{VDF} 用于验证 MinRoot 计算结果的正确性, 而 π_{zk} 则在不公开 σ 、 r 和 x 的情况下, 证明承诺中确实对应一个关于消息 m 的有效签名。

3.2 时间锁承诺验证与打开

(1) 承诺验证 (Vrfy)

承诺验证算法由任意参与者执行, 输入公钥 pk 、消息 m 、时间锁承诺 C 及其零知识证明 π_{zk} , 用于验证承诺中是否包含消息 m 对应的有效签名:

$\text{Vrfy}(pk, m, C, \pi_{zk}) \rightarrow \{0, 1\}$. 设时间锁承诺 $C = (y, \pi_{\text{VDF}})$, 其中 y 为 MinRoot VDF 的计算结果, π_{VDF} 为对应的 VDF 正确性证明。验证算法通过验证零知识证明 π_{zk} , 检查其是否证明存在签名 σ 、随机数 r 及 VDF 输入 x , 满足

$$\exists \sigma, r, x: \quad \text{Vf}(pk, m, \sigma) = 1 \wedge x = \text{Encode}(\sigma, r)$$

$\wedge \text{VDF.Verify}(pp_{\text{vdf}}, x, y, \pi_{\text{VDF}}) = 1$. 若零知识证明验证通过, 则输出 1, 表示承诺 C 中绑定了消息 m 的有效签名; 否则输出 0。

由于 x 、 σ 和 r 均未公开, 验证者无需恢复隐藏签名或执行 MinRoot 的时间锁求解过程, 即可确认承诺的有效性。

(2) 打开操作 (Open)

主动打开算法由签名者执行, 输入时间锁承诺 C , 直接释放其中隐藏的签名及承诺随机数, 并生成对应的打开事件证明: $(\sigma, r, \pi_O) \leftarrow \text{Open}(C)$ 。由于签名者在生成承诺时已知签名 σ 及随机数 r , 因此无需执行 MinRoot 的时间锁求解过程即可恢复承诺内容。算法公开 (σ, r) , 并将对应的打开事件写入 Merkle 事件日志, 生成打开事件证明 π_O 。其中, (σ, r) 用于验证承诺的正确打开, π_O 用于证明该打开事件已被记录。

因此, Open 的主要计算开销不依赖时间参数 T , 无需重新执行 VDF 求解。

(3) 强制打开 (ForceOpen)

强制打开算法由其他参与者执行, 输入时间锁承诺 $C = (y, \pi_{\text{VDF}})$, 通过求解 MinRoot 时间锁恢复承诺中隐藏的签名 $\sigma \leftarrow \text{ForceOpen}(C)$ 。首先从承诺中获取 VDF 输出 y , 并执行 MinRoot 逆向迭代计算, 经过 T 轮计算恢复隐藏输入 $x = F_{\text{MinRoot}}^{-T}(y)$ 。随后对 x 进行解析, 得到 $(\sigma, r) = \text{Decode}(x)$, 并通过 $\text{Vf}(pk, m, \sigma)$ 验证恢复出的签名有效。若验证通过, 则输出签名 σ 。

与 Open 不同, ForceOpen 无需签名者参与, 也不要求签名者提供随机数 r ; r 仅作为恢复隐藏

输入的中间信息使用。因此, ForceOpen 的主要计算开销来自 MinRoot 的 T 轮逆向计算。

3.3 撤销与最终状态验证

(1) 撤销操作(Revoke)

撤销算法由签名者执行, 输入私钥 sk 和待撤销的时间锁承诺 C , 生成撤销事件及其公开可验证的事件证明: $(\rho, \pi_R) \leftarrow \text{Revoke}(sk, C)$ 。

具体而言, 签名者首先生成撤销授权 $\rho = \text{Sign}_{sk}(\text{Revoke} \parallel C \parallel t_R)$, 其中 t_R 为撤销事件发生时间。随后构造撤销事件记录 $e_R = (\text{Revoke}, C, t_R, \rho)$, 并将其写入 Merkle 事件日志, 得到更新后的日志根 R 。最后生成事件 e_R 相对于 R 的 Merkle 包含证明 $\pi_R = \text{Merkle.Prove}(e_R, R)$ 。算法最终输出撤销凭证 ρ 和撤销事件证明 π_R 。

(2) 策略评估(Eval)

策略评估算法由任意参与者执行, 用于验证撤销事件和打开事件的有效性, 并依据预设策略函数 P 判断撤销条件是否成立。

1) 给定撤销事件证明 π_R 、打开事件证明 π_O 、时间参数 T 及策略函数 P , 执行 $(\pi_P, b) \leftarrow \text{Eval}(\pi_R, \pi_O, T, P)$, 其中 $b \in \{0, 1\}$ 表示策略判定结果, $b = 1$ 表示撤销条件成立。

2) 若提供撤销事件证明 π_R , 首先验证其对应事件 $e_R = (\text{Revoke}, C, t_R, \rho)$ 的 Merkle 包含性, 并验证撤销授权满足 $\text{Vf}(pk, \text{Revoke} \parallel C \parallel t_R, \rho) = 1$ 。若提供打开事件证明 π_O , 则验证对应打开事件 $e_O = (\text{Open}, C, t_O, \sigma)$ 的 Merkle 包含性, 并提取其时间戳 t_O 。随后根据事件时间戳及时间参数 T 计算策略结果 $b = P(\pi_R, \pi_O, T)$ 。

3) 最终生成策略证明 $\pi_P = (\pi_R, \pi_O, \rho, b, R)$, 其中 R 为对应的 Merkle 日志根。若上述事件验证或策略计算失败, 则输出无效结果。当 $b = 1$ 时, 表示撤销条件成立, 签名应被视为已撤销; 当 $b = 0$ 时, 表示撤销条件不成立, 签名保持有效。

(3) 最终验证(Accept)

最终验证算法由任意参与者执行, 输入公钥 pk 、消息 m 、签名 σ 、时间参数 T 及策略证明 π_P , 用于判断签名是否满足 RVTS 的最终有效性条件:

$$b_{\text{acc}} \leftarrow \text{Accept}(pk, m, \sigma, T, \pi_P).$$

1) 首先验证签名的正确性 $\text{Vf}(pk, m, \sigma) = 1$ 。

2) 随后解析策略证明 $\pi_P = (\pi_R, \pi_O, \rho, b, R)$, 并对其中的事件证明进行验证。若存在撤销事件, 则验证其 Merkle 包含性以及撤销授权:

$$\text{MerkleVerify}(e_R, \pi_R, R) = 1,$$

$\text{Vf}(pk, \text{Revoke} \parallel C \parallel t_R, \rho) = 1$ 。若存在打开事件, 则验证其 Merkle 包含性。

3) 最后重新计算策略函数并检查 $b = P(\pi_R, \pi_O, T)$ 。仅当签名验证通过、所有事件证明有效且 $b = 0$ 时, 输出 1, 表示签名最终有效; 否则输出 0。

上述各算法共同保证了 RVTS 实例化方案的正确运行: 合法生成的签名能够通过签名验证, 合法构造的承诺能够通过承诺验证, 签名者能够通过 Open 正确释放签名, 其他参与者能够通过 Force-Open 在满足时间条件后恢复签名, 并能够依据撤销事件及打开事件的真实状态通过 Eval 和 Accept 得到一致的最终验证结果。

4 方案相关分析

本节对前文提出的 RVTS 具体构造进行全面的分析与评估。首先从正确性角度证明方案中各算法的功能完备性, 随后基于 2.2 节定义的安全模型, 逐一证明方案满足不可伪造性、可靠性、隐私性、策略可靠性与策略完备性。最后从计算开销、存储开销与通信开销三个方面评估方案的效率, 并与现有的可验证定时签名方案进行比较。

4.1 正确性分析

本小节对 RVTS 方案的正确性进行分析, 主要验证各参与方按照方案规定执行时, 合法生成的签名能够被正确验证, 并在相应条件下被正确打开或判定其状态。

定理 1 (正确性) 若底层 BLS 签名方案、zkVDF 和 Merkle 事件日志均满足正确性, 则第 3 节构造的 RVTS 方案满足第 2.2 节定义的正确性。

证明: 考虑一个诚实执行的 RVTS 实例。首先, 由 BLS 签名方案的正确性, 对于任意由 $\sigma \leftarrow \text{Sign}(sk, m)$ 生成的签名, 均有 $\text{Vf}(pk, m, \sigma) = 1$ 。

其次, 执行 $(C, \pi_C) \leftarrow \text{Commit}(pk, m, \sigma, T)$ 。由 zkVDF 的正确性, 承诺 C 与签名 σ 满足 $C =$

$\text{VDF.Eval}(\sigma, T)$, 且对应证明 π_c 能够通过验证。因此 $\text{Vrfy}(pk, m, C, \pi_c) = 1$.

对于主动打开, 诚实签名者执行 $(\sigma, r, \pi_o) \leftarrow \text{Open}(C)$, 并由时间锁求解过程恢复原签名, 因此得到的 σ 满足 $\text{Vf}(pk, m, \sigma) = 1$.

对于强制打开, 任意参与者在完成 T 步顺序计算后执行 $\sigma \leftarrow \text{ForceOpen}(C, T)$. 由 VDF 的正确性和确定性, 恢复出的输入即为 Commit 阶段所隐藏的签名, 因此同样满足 $\text{Vf}(pk, m, \sigma) = 1$.

最后考虑最终验证。对于诚实生成的撤销事件和打开事件, Merkle 事件日志的正确性保证相应事件证明能够通过验证; 策略函数 P 为确定性函数, 因此 Eval 得到的策略结果与真实事件状态一致。当 $P(\pi_R, \pi_o, T) = 1$ 时, Eval 正确判定撤销成立, Accept 输出 0; 当撤销未发生或策略条件不成立, 即 $P(\pi_R, \pi_o, T) = 0$, 则 Accept 输出 1。

因此, 签名验证、承诺验证、主动打开、强制打开及最终验证均能够在诚实执行下得到正确结果, 故 RVTS 满足正确性。□

4.2 安全性分析

本节证明第 2 节所构造的 RVTS 方案满足第 2.2 节定义的六项安全性质。方案的安全性分别建立在底层密码学组件的安全性以及撤销策略函数的公开可计算性之上。具体而言, BLS 签名的正确性与不可伪造性保证签名部分的安全性; zkVDF 的正确性、可靠性、零知识性及时间顺序性保证时间锁承诺的正确性、可靠性与隐私性; Merkle 事件日志的完整性保证撤销与打开事件证明不可被伪造; 撤销策略函数的确定性则保证策略评估结果的一致性。

基于本方案所使用的底层组件^[12,20-21,23,27-28], 以下分析均假设:

- (1) BLS 签名方案满足 EUF-CMA 不可伪造性;
- (2) zkVDF 满足正确性、证明可靠性和零知识性, 并满足其底层 VDF 的顺序性;
- (3) Merkle 哈希函数具有抗碰撞性;
- (4) 撤销策略函数 P 为公开可计算的确定性函数;
- (5) 签名者私钥 sk 未被敌手获得。

定理 2 (不可伪造性) 若底层 BLS 签名方案满足 EUF-CMA 不可伪造性, 则第 3 节构造的

RVTS 方案满足不可伪造性。

证明: 假设存在概率多项式时间敌手 $\mathcal{A}$, 能够以非可忽略概率生成一个新的消息—签名对 (m^*, σ^*) 使得 $\text{Vf}(pk, m^*, \sigma^*) = 1$, 且 m^* 未被查询过, 则可以构造一个 BLS 签名方案的攻击者 $\mathcal{B}$ 。 $\mathcal{B}$ 将 BLS 公钥 pk 提供给 $\mathcal{A}$, 并将 $\mathcal{A}$ 的签名查询转发给 BLS 签名预言机, 将所得签名返回给 $\mathcal{A}$ 。对于承诺、打开和事件记录, $\mathcal{B}$ 可按照第 3 节算法进行计算。

最终, 若 $\mathcal{A}$ 输出上述新消息—签名对, 则 $\mathcal{B}$ 直接输出 (m^*, σ^*) 作为 BLS 签名伪造。由于 m^* 未被查询且 $\text{Vf}(pk, m^*, \sigma^*) = 1$, $\mathcal{B}$ 以与 $\mathcal{A}$ 相同的非可忽略概率攻破 BLS 的 EUF-CMA 安全性, 与假设矛盾。因此不存在这样的概率多项式时间敌手, RVTS 满足不可伪造性。□

定理 3 (可靠性) 若 zkVDF 的证明系统满足可靠性, 则第 3 节构造的 RVTS 方案满足可靠性, 即任何能够通过 Vrfy 验证的承诺, 在 ForceOpen 后恢复的签名均能够通过底层 BLS 签名验证。

证明: 假设存在敌手 $\mathcal{A}$ 构造承诺 C^* 及证明 π_c^* , 使得 $\text{Vrfy}(pk, m, C^*, \pi_c^*) = 1$, 但经过强制打开得到的签名 $\sigma^* = \text{ForceOpen}(C^*, T)$ 满足 $\text{Vf}(pk, m, \sigma^*) = 0$ 。

根据 Vrfy 的定义, 验证通过意味着 π_c^* 能够证明存在某个 σ , 使得 $\text{Vf}(pk, m, \sigma) = 1$ 且 $C^* = \text{VDF.Eval}(\sigma, T)$ 。

而 ForceOpen 在完成 T 步顺序计算后得到的结果正是该 VDF 输入。因此, 由 VDF 的确定性, 有 $\sigma^* = \sigma$ 。于是应有 $\text{Vf}(pk, m, \sigma^*) = 1$, 这与假设 $\text{Vf}(pk, m, \sigma^*) = 0$ 矛盾。

因此, 敌手无法构造“承诺验证通过但强制打开后签名无效”的承诺。故 RVTS 满足可靠性。□

定理 4 (隐私性) 若底层时间锁承诺构造满足第 2 节所定义的模拟性, 且 zkVDF 证明系统满足零知识性, VDF 满足顺序性, 则第 3 节构造的 RVTS 方案满足隐私性。

证明: 考虑时间参数 T 对应的顺序计算尚未完成的阶段。真实承诺由 $C = \text{VDF.Eval}(\sigma, T)$ 生成, 并同时附带零知识证明 π_c , 其中 π_c 证明

$$\exists \sigma: \text{Vf}(pk, m, \sigma) = 1 \wedge C = \text{VDF.Eval}(\sigma, T).$$

构造模拟器 $\mathcal{S}$ 。模拟器不获得真实签名 σ ，而仅根据公开参数及时间参数生成与真实承诺对应的模拟视图，并利用 zkVDF 的零知识模拟算法生成模拟证明 π_C^{sim} 。由 zkVDF 的零知识性，真实证明 π_C 与模拟证明 π_C^{sim} 在计算上不可区分。因此，敌手无法通过证明部分获得隐藏签名的信息。

另一方面，在时间参数 T 对应的顺序计算完成之前，敌手若能够从承诺 C 中恢复或有效区分隐藏签名，则意味着其能够在少于 T 步的计算时间内获得 VDF 的隐藏输入，从而违反底层 VDF 的顺序性。

因此，在敌手计算能力受限于 T 的条件下，真实承诺及其证明与模拟器生成的视图计算不可区分。故 RVTS 满足隐私性。□

定理 5 (策略可靠性) 若 BLS 签名满足 EUF-CMA 不可伪造性，Merkle 事件日志满足抗篡改性，且撤销策略函数 P 为确定性函数，则第 3 节构造的 RVTS 方案满足策略可靠性。

证明： 假设存在概率多项式时间敌手 $\mathcal{A}$ ，能够构造一个策略证明 π_p ，使得：

- (1). 对应签名 σ 有效；
- (2) $\text{Accept}(pk, m, \sigma, T, \pi_p) = 1$ ；

(3) 实际撤销事件并不满足策略函数，即撤销条件实际上不成立。

根据 Accept 的定义，要使其输出 1，策略证明中的撤销事件必须通过 Merkle 包含性验证，撤销授权 ρ 必须满足 $\text{Vf}(pk, \text{Revoke} \parallel C \parallel t_R, \rho) = 1$ ，且策略结果必须满足 $P(\pi_R, \pi_O, T) = 0$ 。

若敌手试图伪造一个不存在的撤销事件，则必须伪造有效的 Merkle 包含证明或构造不同事件但具有相同 Merkle 根的记录，这将导致 Merkle 哈希函数抗碰撞性的破坏。

若敌手试图伪造撤销授权 ρ ，使其满足 $\text{Vf}(pk, \text{Revoke} \parallel C \parallel t_R, \rho) = 1$ ，则在未知 sk 的情况下相当于伪造签名者对撤销事件的 BLS 签名，从而破坏 BLS 的 EUF-CMA 安全性。

最后，由于 P 是公开可计算的确定性函数，只要事件及其时间戳均未被篡改， Eval 与 Accept 对同一组事件得到的策略结果必然一致。因此敌手无法在实际策略条件不成立时构造一个能够通过最

终验证的非法撤销证明。故敌手成功违反策略可靠性的概率至多为底层 BLS 不可伪造性和 Merkle 抗碰撞性的可忽略函数之和，因此为可忽略函数。故 RVTS 满足策略可靠性。□

定理 6 (策略完备性) 若 BLS、zkVDF 和 Merkle 事件日志均满足正确性，且撤销策略函数 P 为确定性函数，则第 3 节构造的 RVTS 方案满足策略完备性，即当合法撤销事件满足预设策略条件时，诚实生成的策略证明能够被最终验证算法正确识别，并使签名失效。

证明： 设签名者对承诺 C 执行合法撤销操作，得到 $(\rho, \pi_R) \leftarrow \text{Revoke}(sk, C)$ 。由 BLS 签名正确性，有 $\text{Vf}(pk, \text{Revoke} \parallel C \parallel t_R, \rho) = 1$ 。同时，由 Merkle 事件日志的正确性，撤销事件 $e_R = (\text{Revoke}, C, t_R, \rho)$ 及其证明 π_R 能够通过 Merkle 包含性验证。

若存在合法打开事件，则同理，打开事件及其证明 π_O 能够通过相应验证。于是 Eval 根据真实事件执行策略函数 $b = P(\pi_R, \pi_O, T)$ 。由于假设撤销事件满足策略条件，因此 $b = 1$ 。 Eval 由此生成合法策略证明 π_p 。

在最终验证阶段， Accept 能够正确验证撤销授权、事件证明以及策略结果，并得到 $b = 1$ 。按照最终判定规则， Accept 输出 0，表示签名已被撤销。

因此，当合法撤销满足预设策略条件时，诚实生成的策略证明不会被 Accept 错误拒绝，签名能够被正确判定为失效。故 RVTS 满足策略完备性。□

4.3 效率分析

RVTS 在传统定时签名机制的基础上引入了撤销机制、Merkle 事件日志以及策略验证，其效率主要取决于 BLS 签名、VDF、零知识证明和 Merkle 事件日志等密码学组件。本节从算法结构和渐近复杂度的角度，对方案的计算开销进行分析。记时间锁参数为 T ，事件日志中的事件数量为 N ，Merkle 树高度为 $h = \lceil \log_2 N \rceil$ ；记 C_{BLS} 为一次 BLS 签名或验证的计算开销， $C_{\text{VDF}}(T)$ 为时间参数为 T 时的 VDF 顺序计算开销， C_{ZKP}^G 和 C_{ZKP}^V 分别表示零知识证明的生成与验证开销， C_H 表示一次底层哈希计算的开销。

各算法的计算开销主要由其所调用的密码学组

件决定。Sign 仅执行一次 BLS 签名, 与时间参数 T 无关; Commit 需要执行 VDF 顺序计算并生成零知识证明, 因此主要开销为 $C_{\text{VDF}}(T) + C_{\text{ZKP}}^G$ 。Vrfy 无需重新执行 VDF, 仅需验证零知识证明, 因此开销主要来自 C_{ZKP}^V 。Revoke 需要生成撤销授权并将事件写入 Merkle 日志, 其证明生成开销随事件数量呈对数增长。Open 直接释放签名并生成打开事件证明, 无需重新执行 VDF; ForceOpen 则需要完成时间锁对应的顺序计算, 因此其主要开销为 $C_{\text{VDF}}(T)$ 。Eval 和 Accept 主要涉及事件证明、签名及策略结果的验证, 其开销主要与 BLS 验证、Merkle 证明验证及策略函数计算有关。

综上, 各算法的主要理论计算开销如表 1 所示。

表 1 算法理论计算开销

算法	主要计算开销
Setup	一次性参数初始化开销
Sign	$O(C_{\text{BLS}})$
Commit	$O(C_{\text{VDF}}(T) + C_{\text{zk}}^{\text{gen}})$
Vrfy	$O(C_{\text{zk}}^{\text{ver}})$
Revoke	$O(C_{\text{BLS}} + hC_H)$
Open	$O(hC_H)$
ForceOpen	$O(C_{\text{VDF}}(T))$
Eval	$O(C_{\text{BLS}} + hC_H + C_P)$
Accept	$O(C_{\text{BLS}} + hC_H + C_P)$

由表 1 可见, RVTS 的主要计算开销集中在时间锁承诺与强制打开阶段, 即 Commit 和 ForceOpen, 主要承担 VDF 顺序计算。相比之下, Vrfy、Eval 和 Accept 均不需要重新执行完整的时间锁计算, 其开销主要来自证明验证、签名验证以及 Merkle 事件验证。

4.4 对比分析

现有工作中虽然也存在将“撤销”引入定时密码机制的研究, 但不同方案中撤销的对象和含义并不完全相同。例如, 侯慧莹等的可验证的属性基定时签名方案及其应用 (Verifiable Attribute-Based Timed Signatures, VABTS)^[13]中的撤销主要针对用户或属性资格; Almashaqbeh, G. 等的 Anonymous, Timed and Revocable Proxy Signatures^[29]中的撤销则针对代理签名场景中的委托关系。此外, 部分时间相关签名^[30-31]通过预先设置有效期使签名在特定时间后失效, 其本质上属于时间到期, 而非针对已生成签名的主动撤销。

表 2 对上述不同机制中的撤销对象进行了对比。

由表 2 可见, 不同方案虽然均使用了“撤销”这一概念, 但其作用对象存在明显区别。现有方案中的撤销主要针对参与者资格、委托关系或时间有效性, 而本文 RVTS 的撤销对象是一个已经完成承诺但尚未最终打开的特定定时签名。

具体而言, 签名者可以针对特定承诺执行 Revoke 操作, 并通过撤销事件及预先定义的策略 P 决定该签名最终是否有效。因此, 本文方案属于面向签名实例的撤销, 其关注的问题不是“谁仍具有签名权限”, 也不是“签名何时自然过期”, 而是“一个已经进入定时释放过程的具体签名, 在最终打开之前是否仍然有效?”

由此, RVTS 在传统定时签名提供的时间控制基础上进一步引入了针对具体签名的状态控制。

5 方案拓展

RVTS 的一个重要特点是, 撤销行为并不由固定的时间规则决定, 而是通过公开可计算的撤销策略函数 P 对撤销事件、打开事件及相关时间信息进行统一判定。由此, 同一 RVTS 基础构造可以通过选择不同的策略函数, 适配具有不同撤销语义的

表 2 不同方案中撤销含义

方案/机制	撤销对象	撤销含义
VABTS ^[13]	用户或属性资格	撤销用户继续参与签名的资格
Timed and Revocable Proxy Signatures ^[29]	代理签名授权	终止原签名者授予代理签名者的委托权限
基于有效期的时间相关签名 ^[30-31]	签名的时间有效性	达到预设时间后签名自然失效
本文 RVTS	特定定时签名	在签名最终打开前主动撤销该签名, 使其在满足策略条件时失效

应用场景。第 3 节所构造的事件日志与策略评估机制为这种策略化应用提供了统一基础：事件由 Merkle 事件日志进行记录并提供可验证证明，策略函数则根据公开的事件状态计算签名最终是否失效。

本节选取具有代表性的撤销策略进行分析，以说明 RVTS 如何通过改变策略函数实现不同的签名生命周期管理机制。首先介绍撤销一旦发生便永久有效的强撤销模型，随后进一步讨论具有时间窗口限制以及延迟生效特征的撤销模型。强撤销模型

5.1 强撤销策略下的可撤销可验证定时签名方案

(1) 强撤销的策略定义

强撤销模型要求：一旦合法撤销事件发生，该撤销状态永久有效，不受后续时间推进或 Open 行为影响。

设 π_R 为撤销事件证明， π_O 为打开事件证明。对于强撤销策略，仅需判断合法撤销事件是否存在，而不考虑撤销事件与打开事件之间的先后关系。因此，可以定义策略函数为

$$P_{SR}(\pi_R, \pi_O, T) = \begin{cases} 1, & \pi_R \neq \perp, \\ 0, & \pi_R = \perp. \end{cases} \text{ 其中, } P_{SR} = 1 \text{ 表示撤销条件成立, 即签名被永久撤销; 反之, } P_{SR} = 0 \text{ 表示不存在有效撤销事件, 签名保持有效。}$$

需要指出的是，在该策略下，时间参数 T 以及打开事件 π_O 不参与撤销条件的判定。只要撤销事件已经按照方案规定被合法记录并通过事件证明验证，该签名即进入永久撤销状态。因此，即使撤销事件发生后签名随后通过 Open 主动公开，或者达到时间参数后由其他参与者通过 ForceOpen 恢复，均不会改变已经成立的撤销结果。

因此，强撤销模型的核心并非限制撤销发生的时间，而是将合法撤销事件视为签名状态的一次不可逆转换，一旦完成该状态转换，后续签名公开行为仅负责恢复签名内容，而不具有恢复其有效状态的作用。

因此，强撤销模型的核心并非限制撤销发生的时间，而是将合法撤销事件视为签名状态的一次不可逆转换，一旦完成该状态转换，后续签名公开行为仅负责恢复签名内容，而不具有恢复其有效状态的作用。

(2) 强撤销的策略实例

1) 数字安全吊销系统

在数字证书、授权凭证等场景中，签名者可能在定时签名发布后、正式公开前发现对应私钥已经泄露，或者发现该签名对应的安全凭证存在风险。此时，签名者可以执行 Revoke 生成撤销事件证明 π_R ，并将该事件记录于事件日志中。

若撤销事件有效，则有

$\pi_R \neq \perp \Rightarrow P_{SR} = 1 \Rightarrow \text{Accept} = 0$ 。此后即使达到时间参数 T ，其他参与者通过 ForceOpen 恢复出原始签名，撤销状态仍然成立。因此，该模型能够使撤销结果与具体签名实例绑定，并保证已撤销签名不会因后续时间推进而重新获得有效性。

2) 风险控制签名系统

在金融交易、审计等风险敏感场景中，签名可能需要经过一定时间后才能公开。在等待期间，如果签名者发现对应交易或审计数据存在异常，可以针对该签名执行撤销。

此时，若撤销事件发生在签名尚未公开的阶段，则系统记录相应的撤销事件并得到 $P_{SR} = 1$ 。

即使随后执行 Open，或者其他参与者通过 ForceOpen 恢复签名，Accept 仍根据已经记录的撤销事件输出 $\text{Accept} = 0$ 。

因此，在该场景下，签名的有效状态由撤销事件决定，而不随后续的签名打开操作恢复。

5.2 时间窗口撤销策略下的可撤销可验证定时签名方案

(1) 时间窗口撤销的策略定义

时间窗口撤销模型规定，签名者仅能够在签名尚未公开的时间窗口内执行有效撤销。由于签名可以通过主动 Open 或 ForceOpen 两种方式公开，因此撤销必须发生在主动 Open 以及 ForceOpen 时刻之前。

设 t_R 为撤销事件发生时间， t_O 为主动 Open 导致签名公开的时间， T 为 ForceOpen 导致签名公开的时间，则时间窗口撤销策略定义为

$$P_{TW}(\pi_R, \pi_O, T) = \begin{cases} 1, & \pi_O = \perp \wedge t_R < T, \\ 1, & \pi_O \neq \perp \wedge t_R < t_O < T, \\ 0, & \text{otherwise.} \end{cases} \text{ 其}$$

中， $\pi_O = \perp$ 表示尚未发生主动 Open。由于 ForceOpen 由验证者独立执行，方案不要求验证者向签名者提供 ForceOpen 后的事件记录，因此直接使用时间参数 T 作为 ForceOpen 的时间边界。

在该模型中，主动 Open 与 ForceOpen 分别构成撤销窗口的两类关闭条件。若签名者在 t_O 时刻执行 Open，则签名已经正式公开，此后发生的撤销不满足 $t_R < t_O$ ，因而不能使签名失效。另一方面，若签名者始终未执行 Open，则当时间参数达到 T 后，其他参与者即可通过 ForceOpen 恢复签

名, 此时撤销窗口同样关闭, 因此后续撤销不再满足 $t_R < T$ 。

因此, 时间窗口撤销模型的核心是根据撤销事件与签名公开事件之间的时序关系确定撤销是否有效。一旦签名已经通过 Open 公开, 或者已经达到 T 所对应的 ForceOpen 时间边界, 后续撤销均不再改变签名的最终有效状态。

(2) 时间窗口撤销的策略实例

1) 延迟公开的交易签名

在需要延迟公开交易结果的场景中, 签名者首先通过 Commit 将交易签名置于等待状态。在等待期间, 如果交易状态发生变化或交易需要被取消, 签名者可以在撤销窗口内执行 Revoke, 生成相应的撤销事件证明 π_R 。

当撤销事件发生于签名公开之前且尚未超过时间参数 T 时, 有

$t_R < t_O < T \Rightarrow P_{TW} = 1 \Rightarrow \text{Accept} = 0$. 此时, 即使后续执行 Open 或 ForceOpen, 已经记录的撤销事件仍满足时间窗口撤销策略, 签名最终被判定为无效。

若签名已经通过 Open 公开, 即 $t_O < t_R$, 则撤销事件不再满足 $t_R < t_O$, 因而有 $P_{TW} = 0$ 。

同样, 若撤销事件发生在 T 之后, 即 $t_R \geq T$, 则不再满足 $t_R < T$, 后续撤销不会改变签名的最终状态。因此, 该模型可以将交易签名的可撤销性限制在正式公开之前的有限时间窗口内。

2) 限时审批与授权

在限时审批或授权场景中, 可以将 Commit 作为审批结果的预提交, 并在满足相应条件后通过 Open 正式公开。在签名公开之前, 审批者仍可以通过 Revoke 取消该审批结果。

设签名尚未主动 Open, 则在 $t_R < T$ 的条件下, 审批者仍处于有效撤销窗口内。若此时执行撤销并满足 $t_R < t_O < T$, 则有 $P_{TW} = 1 \Rightarrow \text{Accept} = 0$. 若审批结果已经通过 Open 正式公开, 则 $t_O < t_R$, 后续撤销不再满足策略条件, 签名保持有效。另一方面, 即使审批者始终未主动 Open, ForceOpen 也会在时间参数 T 后使签名进入公开状态, 因此 T 同时构成撤销窗口的时间上界。

因此, 该模型适用于具有明确“公开前可撤销, 公开后不可撤销”要求的应用场景。与强撤销模型相比, 其撤销状态不再具有永久追溯性, 而是

受到签名公开时间和时间参数 T 的共同约束。

6 总结

本文针对传统定时签名在签名提交后缺乏动态状态管理能力的问题, 提出了一种可撤销可验证定时签名方案, 通过时间锁承诺实现签名延迟公开, 实现针对特定定时签名的状态控制。在此基础上, 本文从正确性、不可伪造性、可靠性、隐私性、策略可靠性与策略完备性六个方面对方案进行了安全性分析, 并从计算、存储和通信开销等方面进行了理论效率分析。进一步通过强撤销模型和时间窗口撤销模型说明了不同撤销策略的适用性, 表明 RVTS 能够通过定义不同的策略函数实现灵活的撤销语义, 从而将定时签名由单一的时间控制扩展为时间控制与状态控制相结合的签名机制。

参考文献:

- [1] BONEH D, SHOU P V. A Graduate Course in Applied Cryptography[J].
- [2] ANDERSON R. Security Engineering: A Guide to Building Dependable Distributed Systems[M]. John Wiley & Sons, 2010.
- [3] GARAY J A, JAKOBSSON M. Timed Release of Standard Digital Signatures[C/OL]//BLAZE M. Financial Cryptography. Berlin, Heidelberg: Springer, 2003: 168-182. DOI:10.1007/3-540-36504-4_13.
- [4] RIVEST R L, SHAMIR A, WAGNER D A. Time-lock Puzzles and Timed-release Crypto[J/OL]. 1996. <https://dspace.mit.edu/handle/1721.1/149822>.
- [5] MAO W. Timed-Release Cryptography[C/OL]//VAUDENAY S, YOUSSEF A M. Selected Areas in Cryptography. Berlin, Heidelberg: Springer, 2001: 342-357. DOI:10.1007/3-540-45537-X_27.
- [6] BONEH D, NAOR M. Timed Commitments[C/OL]//BELLARE M. Advances in Cryptology — CRYPTO 2000. Berlin, Heidelberg: Springer, 2000: 236-254. DOI:10.1007/3-540-44598-6_15.
- [7] BONEH D, BONNEAU J, BÜNZ B, et al. Verifiable Delay Functions [C/OL]//SHACHAM H, BOLDYREVA A. Advances in Cryptology — CRYPTO 2018. Cham: Springer International Publishing, 2018: 757-788. DOI:10.1007/978-3-319-96884-1_25.
- [8] WESOŁOWSKI B. Efficient Verifiable Delay Functions[J/OL]. Journal of Cryptology, 2020, 33 (4): 2113-2147. DOI: 10.1007/s00145-020-09364-x.
- [9] PIETRZAK K. Simple Verifiable Delay Functions[C/OL]//BLUM A. 10th Innovations in Theoretical Computer Science Conference (ITCS 2019): Vol. 124. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019: 60:1-60:15. <https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2019.60>. DOI: 10.4230/LIPIcs.ITCS.2019.60.
- [10] BONEH D, BÜNZ B, FISCH B. A Survey of Two Verifiable Delay Functions[EB/OL]. Cryptology ePrint Archive, 2018. <https://eprint.iacr.org/2018/712>.
- [11] COHEN B, PIETRZAK K. Simple Proofs of Sequential Work[C/OL]//

- NIELSEN J B, RIJMEN V. Advances in Cryptology – EUROCRYPT 2018. Cham: Springer International Publishing, 2018: 451-467. DOI: 10.1007/978-3-319-78375-8_15.
- [12] Thyagarajan Sri Aravinda Krishnan, BHAT A, MALAVOLTA G, et al. Verifiable Timed Signatures Made Practical[C/OL]//Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. New York, NY, USA: Association for Computing Machinery, 2020: 1733-1750. <https://doi.org/10.1145/3372297.3417263>. DOI: 10.1145/3372297.3417263.
- [13] 侯慧莹, 宁建廷, 黄欣沂, 等. 可验证的属性基定时签名方案及其应用[J/OL]. 软件学报, 2023, 34(5): 2465-2481. DOI: 10.13328/j.cnki.jos.006396.
- [14] 陈辉焱, 王庆楠, 谭舜聪, 等. 基于身份的可验证定时格签名方案[J]. 信息网络安全, 2024, 24(9): 1317-1327.
- [15] 陈辉焱, 王庆楠, 王克, 等. 基于格的可验证定时签名与应用[J]. 通信学报, 2024, 45(10): 142-152.
- [16] OZDEN D, YAYLA O. Verifiable Timed Proxy Signatures and Multi-signatures[EB/OL]. Cryptology ePrint Archive, 2023. <https://eprint.iacr.org/2023/1107>.
- [17] USLU E. Quantum-Resistant Verifiable Timed Signature Schemes[J/OL]. 2025. <https://open.metu.edu.tr/handle/11511/115491>.
- [18] USLU E, YAYLA O. UOV-Based Verifiable Timed Signature Scheme[EB/OL]. Cryptology ePrint Archive, 2025. <https://eprint.iacr.org/2025/1181>.
- [19] THYAGARAJAN S A, MALAVOLTA G, SCHMID F, et al. Verifiable Timed Linkable Ring Signatures for Scalable Payments for Monero[C/OL]//ATLURI V, DI PIETRO R, JENSEN C D, et al. Computer Security – ESORICS 2022. Cham: Springer Nature Switzerland, 2022: 467-486. DOI: 10.1007/978-3-031-17146-8_23.
- [20] BONEH D, LYNN B, SHACHAM H. Short Signatures from the Weil Pairing[C/OL]//BOYD C. Advances in Cryptology – ASIACRYPT 2001. Berlin, Heidelberg: Springer, 2001: 514-532. DOI: 10.1007/3-540-45682-1_30.
- [21] HOFFMANN C, PIETRZAK K. Watermarkable and Zero-Knowledge Verifiable Delay Functions from any Proof of Exponentiation[EB/OL]. Cryptology ePrint Archive, 2024. <https://eprint.iacr.org/2024/481>.
- [22] MERKLE R C. Secrecy, authentication, and public key systems. [D]. Stanford, CA, USA: Stanford University, 1979.
- [23] MERKLE R C. A Certified Digital Signature[C/OL]//BRASSARD G. Advances in Cryptology – CRYPTO' 89 Proceedings. New York, NY: Springer, 1990: 218-238. DOI: 10.1007/0-387-34805-0_21.
- [24] CROSBY S A, WALLACH D S. Efficient data structures for tamper-evident logging[C]//Proceedings of the 18th conference on USENIX security symposium. USA: USENIX Association, 2009: 317-334.
- [25] BLUM M, FELDMAN P, MICALI S. Non-interactive zero-knowledge and its applications[C/OL]//Proceedings of the twentieth annual ACM symposium on Theory of computing. New York, NY, USA: Association for Computing Machinery, 1988: 103-112. <https://dl.acm.org/doi/10.1145/62212.62222>. DOI: 10.1145/62212.62222.
- [26] BEN-SASSON E, CHIESA A, TROMER E, et al. Succinct non-interactive zero knowledge for a von Neumann architecture[C]//Proceedings of the 23rd USENIX conference on Security Symposium. USA: USENIX Association, 2014: 781-796.
- [27] KHOVRATOVICH D, MALLER M, TIWARI P R. MinRoot: Candidate Sequential Function for Ethereum VDF[EB/OL]. Cryptology ePrint Archive, 2022. <https://eprint.iacr.org/2022/1626>.
- [28] GABIZON A, WILLIAMSON Z J, CIOBOTARU O. PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge[EB/OL]. Cryptology ePrint Archive, 2019. <https://eprint.iacr.org/2019/953>.
- [29] ALMASHAQBEH G, NITULESCU A. Anonymous, Timed and Revocable Proxy Signatures[C/OL]//MOUHA N, NIKIFORAKIS N. Information Security. Cham: Springer Nature Switzerland, 2025: 23-43. DOI: 10.1007/978-3-031-75757-0_2.
- [30] COLBURN M. Short-lived signatures[D/OL]. Concordia University, 2018. <https://spectrum.library.concordia.ca/id/eprint/984260/>.
- [31] WANG W, DUAN J, ZUO C, et al. A TimeBound NFT Rights Protocol From Time Interval Signatures[J/OL]. IEEE Transactions on Dependable and Secure Computing, 2026, 23(1): 209-220. DOI: 10.1109/TDSC.2025.3604511.



张艳硕 (1979—), 男, 博士, 北京电子科技学院副教授、博士生导师, 主要研究方向为密码理论及其应用。



蔡润泽 (2002—), 男, 北京电子科技学院硕士生, 主要研究方向为密码学。



邹雨耀 (2004—), 男, 北京电子科技学院硕士生, 主要研究方向为密码学。



董有恒 (1995—), 男, 博士, 北京电子科技学院密码科学与技术系讲师, 主要研究方向为序列密码设计及分析、混沌密码理论及其应用。



王影冰 (1996—), 女, 博士, 北京电子科技学院密码科学与技术系讲师, 主要研究方向为密码设计与分析。