

一个非陷门动态伪累加器构造及其应用

关 诚^{1,2}, 薛 锐^{1,2}, 员凯立^{1,2}

¹中国科学院信息工程研究所信息安全国家重点实验室 北京 中国 100093

²中国科学院大学网络空间安全学院 北京 中国 100049

摘要 累加器是一个重要的密码学工具,在成员关系测试、证书管理等应用中发挥着重要作用。传统的基于RSA或双线性映射的累加器在成员关系验证和非成员关系验证时需要给出相应的证据,这对于不需要证据就能验证的应用来说是不必要的。另外,这两类累加器都具有相应的陷门信息,这使得累加器的安全性依赖于陷门信息的保密性,导致实际应用时需要假设拥有陷门信息的累加器管理者是诚实的。虽然基于哈希函数的累加器不需要任何计算假设,但能够做到成员关系验证时不需要给出相应的证据的构造仅支持元素的添加,而不允许元素的删除,而能够做到元素的动态添加和删除的构造其证据尺寸是被累计元素的对数。这些缺陷极大地限制了累加器在现实世界的应用范围。在本文中,我们首先介绍了“动态伪累加器”的概念并给出了相应的安全性定义,拥有动态添加和删除集合元素的功能。随后我们给出了一个具体的具有可证明安全性的动态伪累加器构造,它是非陷门的,不需要假设累加器管理者是诚实的,并且是动态的,允许添加新的元素和删除旧的元素。此外,相比于以往的工作,我们构造的累加器支持集合成员关系和非成员关系验证而不需要给出相应的证据。接着我们详细讨论了构造累加器需要的参数和可累计的集合尺寸上限之间的关系,并说明了在实际中如何选取这些参数。最后我们介绍了新构造的累加器如何用来构造分级访问控制系统。

关键词 动态伪累加器; 访问控制; 成员关系测试

中图法分类号 TP309.7 DOI号 10.19363/J.cnki.cn10-1380/tn.2025.09.14

A Non-Trapdoor Dynamic Pseudo-Accumulator Construction and its Application

GUAN Cheng^{1,2}, XUE Rui^{1,2}, YUN Kaili^{1,2}

¹State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

²School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract Accumulator is an important cryptographic tool, which plays an important role in membership test, certificate management and other applications. Traditional accumulators based on RSA or bilinear maps need to provide evidence for both membership validation and non-membership validation, which are not necessary for applications that verify validation without evidence. In addition, both types of accumulators have corresponded trapdoors, which makes the security of accumulators depend on the confidentiality of the trapdoors, this leads to the need to assume that the accumulator manager with trapdoor information is honest in practical applications. Although the accumulator based on the hash function does not require any computational assumptions, the construction that can realize membership verification without giving corresponding evidence only supports the addition of elements, but does not allow the deletion of elements, but can realize the construction of dynamic addition and deletion of elements, the size of the evidence is the logarithm of the accumulated term. These flaws greatly limit the real-world application of accumulators. In this article, we first introduce the concept of "dynamic pseudo-accumulator" and give the corresponding security definition, which has the functionality of adding and deleting elements of set dynamically. Then we give a concrete construction of a dynamic pseudo-accumulator with provable security which is no trapdoor and there is no need to assume that the accumulator manager is honest, and the accumulator is dynamic, allowing new elements to be added and old elements to be deleted. In addition, compared to previous work, the accumulator that we constructed supports the verification of set membership and non-membership without giving corresponding evidence. And we discuss in detail the relationship between the parameters required to construct the accumulator and the upper bound of the size of the accumulative set, and explain how to choose these parameters in practice. Finally, we introduce how the newly constructed accumulator can construct a hierarchical access control system.

Key words dynamic pseudo-accumulator; access control; membership test

通讯作者: 薛锐, 博士, 研究员, Email: xuerui@iie.ac.cn。

本课题得到国家自然科学基金项目(No. 61772514); 国家重点研发计划项目(No. 2017YFB1400700); 北京市科学技术委员会(No. Z191100007119006)资助。

收稿日期: 2020-11-09; 修改日期: 2021-02-23; 定稿日期: 2023-02-22

1 引言

累计器的概念最早由 Benaloh 和 Mare^[1]引进, 用以将一个含有有限个元素的集合 $S = \{x_1, \dots, x_n\}$ 累计成一个单个值(相当于对集合作承诺)。对于每一个属于集合的元素 x_i , 能够高效地计算相应的证据 v_i , 证明 x_i 确实属于集合 S 。对于一个不属于集合的元素 $x' \notin S$, 找到一个相应的证据 v' 证明 $x' \in S$ 在计算上是不可行的。

累计器在一些增强隐私的应用程序中非常有用, 例如: 时间戳^[1]、证书管理^[2-4]、群签名和匿名证书^[5, 6]、匿名电子现金^[7-8]、认证数据上的计算^[9]、可更新的签名^[10-11]、去中心化公告板^[12-13]、广播加密^[14]。最近, Boneh 等人^[15]介绍了累计器在区块链上的应用。以比特币区块链为例, 累计器的引入可以使得大部分节点不需要维护未花费交易输出 UTXOs, 在提高事物有效性验证效率的同时, 还极大地降低了节点的存储开销。

一般地, 有三种不同类型的累计器, 即基于 RSA^[1, 5-6, 16-19]、基于双线性映射^[4, 15, 20-21]、基于哈希函数^[22-23]。在基于 RSA 的累计器构造系列工作中, 继 Benaloh 和 Mare^[1]之后, Camenisch 和 Lysyanskaya^[5]引进了动态累计器的概念, 允许动态地添加和删除累计项, 并且添加或删除元素的成本与被累计项的数量无关。他们的构造支持成员关系证据的高效计算, 但却没有给出非成员关系证据^①计算的解决办法。对于这一不足, Li 等人^[16]构造了一个通用累计器, 不仅支持成员关系证据的高效计算, 而且支持非成员关系证据高效计算。尽管在前人研究下, 累计器的功能逐渐强大, 但每次添加或删除一个元素, 都需要重新计算证据, 这使得证据更新次数与添加或删除的元素个数成正比。为解决这一问题, Wang 等人^[24]提出了动态累计器的并行更新方法, 使得多个元素的添加或删除只需要一次操作便可完成, 同时, 证据的更新独立于添加或删除的元素个数。然而, 他们的构造被证明是不安全的^[25]。

基于双线性映射的累计器首次由 Nguyen^[20]提出并用于基于身份的环签名方案中, 其安全性基于强 Diffie-Hellman 假设。然而, 他们仅仅给出了如何利用他们的累计器构造成员关系证据, 没有给出非成员关系证据的构造。Damgaard 和 Triandopoulos^[21]

在 Nguyen^[20]的工作基础上, 扩展累计器以支持非成员证据构造以及相应的非成员证明, 这使得基于双线性映射的累计器的功能等效于基于 RSA 的累计器^[17]。Camenisch 等人^[4]针对基于证书的隐私保护认证系统的撤销问题, 提出了一个新的基于双线性映射的累计器构造。提供了第一个提供隐私保护的认证系统, 适用于电子令牌(如电子身份证或驾照)的实施。Thakur^[26]基于双线性映射构造的累计器能够将多个成员关系证据或非成员关系证据聚合成一个单一的证据, 以证明给定的成员集合在累计器表达的集合里, 或证明给定的非成员集合不在累计器表达的集合里。

基于哈希函数构造的累计器最早由 Nyberg^[24]给出。其构造仅仅使用哈希函数, 并且安全性仅依赖于哈希函数的随机性, 而不需要任何计算假设。不同于基于 RSA 和双线性映射的累计器, 这种构造产生的累计值不是一个群元素, 而是一个二进制向量。并且证明集合元素的成员关系不需要给出相应的证据。不足的是, 这种构造仅支持元素的添加, 而不允许元素的删除。Camacho 等人^[23]基于哈希函数构造了一个强通用累计器, 它在功能上类似于通用累计器, 不用假设累计器管理者是可信的。不足的是, 他们的成员关系或非成员关系证据的尺寸是被累计项的对数。

值得一提的是, 基于 RSA 和基于双线性映射的累计器构造都存在一个陷门, 基于 RSA 的累计器的陷门信息是 RSA 模数 N 的分解, 基于双线性映射的累计器的陷门信息是一个秘密值 s 。陷门信息的存在使得累计器的管理者能高效地计算成员关系证据或非成员关系证据, 但这也使得管理者能够作弊。正如 Sander^[16]指出: 在基于 RSA 的累计器中, 对于累计项 $y_1, \dots, y_m$, 累计值

$$z = x^{\prod_{i=1}^m y_i} \bmod N$$

其中, $N=pq$ 。知道 p 和 q , 使得对于一个合理的 $y \neq y_i, i=1, 2, \dots, m$, 管理者能够计算 $t \cdot y = 1 \bmod \phi(N)$, 从而 $z^t \bmod N$ 是 y 在累计项中的一个证据。因此, 对于这种由陷门信息造成的缺陷, 这两类累计器的构造要么假设管理者是诚实的, 要么假设秘密信息来自可信设置。第一个非陷门累计器由 Nyberg^[22]给出, 随后 Sander^[16]在 Camenisch 和 Lysyanskaya^[5]的累计器基础上, 构造未知分解的

①非成员关系证据: 对于一个不属于集合 S 的元素 x' , 计算一个相应的证据 v' , 使得验证者确信 x' 不属于 S 。

RSA 模数, 使得新构造的累计器不存在陷门。最近, Boneh 等人^[15]也构造了一个非陷门通用累计器, 其基本想法是构造一个未知阶数的群并利用群的一个生成元来累计元素。

在本文中, 我们用一个矩阵来累计集合元素, 相比于传统的累计器是一个单个的值, 我们的累计器是一个矩阵, 为此, 我们称之为“伪累计器”。我们介绍了动态伪累计器的概念及其安全性定义, 并给出了动态伪累计器的一个构造方法。构造的伪累计器没有陷门, 不用假设伪累计器管理者是诚实可信的。伪累计器支持动态地添加和删除元素, 且成员关系和非成员关系的验证不需要相应的证据, 这使得我们构造的伪累计器适用于如下的分级访问控制场景。

住宅小区访问控制: 现代住宅小区一般都配备小区大门门禁系统和单元楼门禁系统。大门门禁系统的功能是防止非小区用户随意进入, 持有小区门禁系统卡的用户可刷卡进入。单元楼门禁系统控制着非此单元楼的用户不能随意进入。一个一般的方法实现这两个访问控制功能是给用户配备两张门禁卡, 一张卡用于小区大门门禁, 另一张卡用于单元楼门禁。但这样做使得门禁卡过于浪费, 不方便用户携带。应用我们的累计器构造, 可实现单个门禁卡的分级访问功能。

我们在第 4 节描述了如何应用我们构造的伪累计器来实现这个分级访问控制系统。

2 基本概念

本节给出本文用到的相关符号和概念的定义。

记计算安全参数为 λ . 用 $[m]$ 表示集合 $\{1, 2, \dots, m\}$. 用 $[m]^w$ 表示取值在 $[m]$ 中长度为 w 的向量。 $\{0, 1\}^\ell$ 表示长度为 ℓ 的比特串。哈希函数 $H: \{0, 1\}^* \rightarrow \{0, 1\}^\ell$, 伪随机函数 $F: \{0, 1\}^\lambda \times \{0, 1\}^\ell \rightarrow [m]^w$ 。

对于函数 $\mu: \mathbb{Z} \rightarrow \mathbb{R}$. 如果对于任何一个多项式 $\text{poly}: \mathbb{Z}^+ \rightarrow \mathbb{R}^+$, 存在正整数 $N_0 \in \mathbb{Z}$ 使得所有 $n > N_0$ 满足 $\mu(n) < 1/\text{poly}(n)$, 则说 μ 是一个可忽略函数。

我们在如下的章节中使用 $\text{negl}(\cdot)$ 来表示一个可忽略函数。

定义 1. 伪随机函数: 对于任意整数 $m, w \in \mathbb{Z}$, 记 $t = \lceil \log m \rceil$. 函数 $F: \{0, 1\}^\lambda \times \{0, 1\}^\ell \rightarrow \{0, 1\}^{wt}$ 是一个高效可抽样的键值函数, 如果 F 满足下述条件则

说 F 是一个伪随机函数: 对所有的概率多项式时间的分辨器 D , 存在一个可忽略函数 $\text{negl}(\cdot)$ 使得

$$|\Pr[D^{F_k(\cdot)}(1^\lambda)] - \Pr[D^{f(\cdot)}(1^\lambda)]| < \text{negl}(\lambda),$$

这里 $k \leftarrow \{0, 1\}^\lambda$ 是均匀随机选取的, $f: \{0, 1\}^\ell \rightarrow \{0, 1\}^{wt}$ 是随机函数。

定义 2. 抗碰撞哈希函数: 哈希函数 $H: \{0, 1\}^* \rightarrow \{0, 1\}^\ell$ 是抗碰撞的, 如果对所有概率多项式时间敌手 $\mathcal{A}$, 存在一个可忽略函数 $\text{negl}(\cdot)$ 使得

$$\Pr[x, x' \leftarrow \mathcal{A}^H(\lambda): H(x) = H(x')] \leq \text{negl}(\lambda).$$

定义 3. 动态伪累计器: H 是一个公开的抗碰撞哈希函数。一个安全的动态伪累计器 Π 由 5 个算法 (Gen, Acc, Add, Del, Vrfy) 组成:

-**函数生成算法 Gen** 是一个随机算法, 输入安全参数 λ 以及累计器的上界 n , 输出系统的参数 para 。

-**累计值算法 Acc** 是一个随机算法, 输入累计项集合 $S \subseteq \{0, 1\}^*$, 输出累计值 D 和辅助信息 A 。

-**添加算法 Add** 是一个随机算法, 输入一个新的元素 $x \notin S$ 以及当前累计值 D 和辅助信息 A , 输出更新后的累计值 D' 和辅助信息 A' 。

-**删除算法 Del** 是一个随机算法, 输入一个元素 $x \in S$, 当前累计值 D 和辅助信息 A , 输出更新后的累计值 D' 和辅助信息 A' 。

-**验证算法 Vrfy**: 输入某个元素 x 和当前累计值 D , 输出 1 表示 $x \in S$; 或者 0 表示 $x \notin S$ 。

考虑到累计器的主要应用, 对于任何累计器 Π , 其安全威胁是敌手能够伪造一个非累计元素, 并且通过验证。注意到敌手在实际攻击中可以得到初始的累计值, 并且能够得到添加和删除的值。我们使用下述随机试验 $\text{Expt}_{\mathcal{A}, \Pi}^{\text{forge}}$ 描述敌手对于 $\Pi = (\text{Gen}, \text{Acc}, \text{Add}, \text{Del}, \text{Vrfy})$ 的伪造攻击。为了刻画敌手得到添加和删除任意消息的能力, 我们给敌手赋以添加应答器 $\text{Add}_{D, A}(\cdot)$ 和删除应答器 $\text{Del}_{D, A}(\cdot)$:

$\text{Expt}_{\mathcal{A}, \Pi}^{\text{forge}}(\cdot)$: 输入 λ

1. $S \leftarrow \mathcal{A}(1^\lambda)$

2. $\text{para} \leftarrow \text{Gen}(1^\lambda, n)$,

$(D, A) \leftarrow \text{Acc}(1^\lambda, S)$

3. $(x, D', S') \leftarrow \mathcal{A}^{\text{Add}_{D, A}(\cdot), \text{Del}_{D, A}(\cdot)}(D, 1^\lambda)$, 记询问过的元素集合为 Q , $x \notin Q$

4. 如果 $\text{Vrfy}(x, D') = 1$ 则输出 1; 否则输出 0

当实验 $\text{Expt}_{A, \Pi}^{\text{forge}}(\cdot)$ 输出 1 时敌手伪造成功。敌手成功的概率是 $\Pr[\text{Expt}_{A, \Pi}^{\text{forge}}(\lambda) = 1]$ 。

注意到上述实验中, 敌手可以访问应答器 $\text{Add}_{D, A}(\cdot)$ 和 $\text{Del}_{D, A}(\cdot)$ 。这时的两个应答器是联合状态: 添加应答器和删除应答器的最初状态具有累计值 D 和辅助信息 A 。当敌手要求添加 (或删除) 一个元素 x 时, 访问应答器 $\text{Add}_{D, A}(\cdot)$ (或 $\text{Del}_{D, A}(\cdot)$)。该应答器计算得到新的累计值 D' 以及新的辅助值 A' 。将累计值返回给敌手, 并将两个应答器 $\text{Add}_{D, A}(\cdot)$ 和 $\text{Del}_{D, A}(\cdot)$ 同时更新为 $\text{Add}_{D', A'}(\cdot)$ 和 $\text{Del}_{D', A'}(\cdot)$ 。如此敌手 $\mathcal{A}$ 可以适应性访问两个应答器, 之后选择自己的伪造 $x \notin Q$ 。伪累计器的安全性定义如下。

定义 4. 如果任意概率多项式时间的算法 $\mathcal{A}$ 对于伪累计器 $\Pi = (\text{Gen}, \text{Acc}, \text{Add}, \text{Del}, \text{Vrfy})$ 的伪造攻击实验成功的概率是可忽略的, 则伪累计器 Π 是安全的。即如果任意概率多项式时间的算法 $\mathcal{A}$ 存在可忽略函数 $\text{negl}(\cdot)$ 使得

$$\Pr[\text{Expt}_{A, \Pi}^{\text{forge}}(\lambda) = 1] < \text{negl}(\lambda),$$

则说 Π 是安全的伪累计器。

安全性确保了对任何概率多项式时间的敌手 $\mathcal{A}$, 在经过一系列元素的添加和删除操作后, 对于当前的集合 S' , 找到一个伪造项 $x \notin S'$ 通过验证算法的概率是可忽略的。

3 动态伪累计器构造

我们在本节给出一个非陷门动态伪累计器的具体构造, 及其安全性证明, 并讨论分析构造累计器使用的参数和可累计的集合上界之间的关系。

Chase 和 Miao^[27]构造了两方 P_1, P_2 参与的私有集合交集计算协议, 其中, 参与者 P_2 构造一个初始值全为 1 的 $m \times w$ 矩阵 D , 利用本地私有集合 Y 将 D 中第 j 列第 $v(j)$ 行位置 $D_j(v(j))$, $j = 1, 2, \dots, w$ 处的值替换成 0, 其中 $v = F_k(H(y))$, $y \in Y$ 。然后 P_2 构造一个随机的 $m \times w$ 二元矩阵 (元素为 0 或 1) A , 利用 D 和 A 计算矩阵 $B = A \oplus D$ 。之后, P_1 和 P_2 执行一个随机不经意传输扩展协议。其中 P_2 充当发送方, 输入为 $\{A_i, B_i\}_{i \in [w]}$, 这里 A_i 和 B_i 分别为 A 和 B 的第 i 个列向量; 参与者 P_1 充当接受方, 输入为一个随机串 $s \in \{0, 1\}^w$, 最后 P_1 得到一个 $m \times w$ 矩阵 C 。他们的

协议的一个关键点在于对所有的 $j \in [w]$, 有 $A_j(v(j)) = B_j(v(j)) = C_j(v(j))$, 这表明, 如果 P_1 一个元素 $x \in X$ 于交集 $X \cap Y$, 那么存在一个 $y \in Y$, $c = C_1(v(1)) \parallel \dots \parallel C_w(v(w))$ 与 $a = A_1(v(1)) \parallel \dots \parallel A_w(v(w))$ 相等, 这里 $v = F_k(H(x)) = F_k(H(y))$, 从而对 c 和 a 做相同的映射 (原文中采用一个哈希函数 H_2) 可得到相同的值。基于这个观察, P_2 可通过将 P_1 发送过来的所有 H_2 映射后的值与本地计算的 H_2 映射后的值对比求出交集。

受上述协议构造启发, 我们摒弃上述协议中不经意传输阶段, 利用矩阵 D 来累计集合的每个元素得到累计值 (一个矩阵)。矩阵 D 的初始元素全部属于某个域 $\mathbb{Z}_p$ 。添加一个元素 x 到 D 中时, 首先计算 $v = F_k(H(x))$, 然后随机选取一个 $j \in w$, 修改 $D_j(v(j))$ 处的值为 $\mathbb{Z}_p$ 中的某个值, 使得 $D_1(v(1)) + \dots + D_w(v(w)) = 0 \pmod p$ 。这也就是说, x 映射到 D 中的 w 个位置处的值的和模 p 为 0。我们以此作为判别条件来判定某个元素 x' 是否属于被累计过的集合。为叙述方便, 我们将元素 x 被累计到矩阵 D 中的位置称为 x 在 D 中的路径。要删除元素 x , 只需要修改它在 D 中的某个位置的值, 使得判定条件不成立即可。为保证添加或删除元素能够正确执行, 我们还构造了一个辅助矩阵 A , 其初始化为一个全 0 的 $m \times w$ 矩阵。添加元素 x 时, 需要将 A 中位置 $A_j(v(j))$, $j = 1, 2, \dots, w$ 处 (x 在 A 中的路径) 的值加 1。如果 A 的某个位置的值大于 1, 这说明存在两个或多个元素在 D 中的路径都经过了此位置。删除元素 x 时, 首先在 A 中寻找 x 的路径上为 1 的位置, 然后在 D 中将相同位置处的值进行修改使得判定条件不成立。我们的动态伪累计器 Π 的具体构造如下。

1. 参数生成算法 $\text{Gen}(\cdot)$: 输入 λ , 输出一组正整数 (k, ℓ, m, n, w, p) 和两个 $m \times w$ 的矩阵 D_0, A_0 。其中这组正整数的长度均是 λ 的多项式并且 $m > 2n$ 。矩阵 D_0 中元素是域 $\mathbb{Z}_p$ 中的随机元素。矩阵 A_0 是元素全为 0 的矩阵。这些参数确定了伪随机函数 $F_k: \{0, 1\}^\ell \rightarrow \{0, 1\}^t$ 和哈希函数 $H: \{0, 1\}^* \rightarrow \{0, 1\}^\ell$ 。其中 $t = w \lceil \log m \rceil$ 。

2. 累计函数 Acc : 输入参数 para , 以及要累计的元素集合 $S = \{x_1, \dots, x_r\}$ 。以 D_0 和 A_0 作为初始的 (对于空集的累计) 累计矩阵和辅助矩阵, 利用下述添

加函数 **Add** 将 S 中元素逐一动态添加到累计值中, 最终得到累计值 D 和辅助矩阵 A 。

3. **添加函数 Add**: 输入 (D, A) 作为累计值和辅助矩阵, 以及新的元素 x 。计算 $v := F_k(H(x))$ 。这时 v 是长 $w \lceil \log m \rceil$ 的比特串, 将其中每 $\log m$ 比特转化为一个自然数, 从而 v 视作 $[m]^w$ 中的一个向量。记 A_j 为矩阵 A 的第 j 个列向量, $A_j(u)$ 是该向量的第 u 个元素。如下进行:

-选择 $j \in [w]$ 使得 $A_j(v(j)) = 0$ 。(后面的讨论中将证明不存在这样 j 的概率是可忽略的。)

-修改矩阵 D 的第 j 列向量 D_j 的第 $v(j)$ 元素值

$$D_j(v(j)) \in \mathbb{Z}_p \text{ 使得 } \sum_{i=1}^w D_i(v(i)) = 0 \pmod{p}.$$

-将所有 $A_i(v(i))$ 的值加 1, $i = 1, 2, \dots, w$ 。

-输出更新后的累计值矩阵 D' 和辅助矩阵 A' 。

4. **删除算法 Del**: 输入累计值 D 和辅助矩阵 A 以及累计的元素 x 。

-计算 $v = F_k(H(x))$, 同上将 v 视为 $[m]^w$ 中的向量。

-在 A 中寻找 A_j 满足 $A_j(v(j)) = 1$ 。

-修改 $D_j(v(j))$ 的值为 $\mathbb{Z}_p$ 中随机值满足

$$\sum_{i=1}^w D_i(v(i)) \neq 0 \pmod{p}.$$

-将所有 $A_i(v(i))$ 的值减去 1。

-输出更新后的累计值 D' 和辅助矩阵 A' 。

5. **验证算法 Vrfy**: 输入当前累计值 D 和元素

x 。计算 $v = F_k(H(x)) \in [m]^w$ 。计算 $s := \sum_{i=1}^w D_i(v(i))$ 。如果 $s = 0 \pmod{p}$ 则输出 1; 否则输出 0。

如图 1 和图 2 分别展示了我们构造的累计器添加两个元素和删除一个元素的处理过程, 这里选取了 $p = 79$ 。在图 1 中, 添加元素 x_1 时, 我们给定了 $v_1 = F_k(H(x_1))$ 的值为 $(4, 5, 1, 2)$ 。元素 x 在 D 中的路径如绿色部分所示。为使得其路径上的和模 p 为 0, 选择将 $D_4(2)$ 处的值修改为 58, 如图 1 中 D' 红色位置所示。同理, 添加元素 x_2 时, 给定了 $v_2 = F_k(H(x_2))$ 的值为 $(2, 5, 3, 6)$, 其在 D 中的路径如蓝色部分所示(第 2 个位置与 x_1 的第 2 个位置重叠, 即 $D_2(5)$ 处绿色部分)。选取位置 $D_3(3)$, 将此处值修改为 64 得到累计器 D' 。辅助信息 A' 标识了累计器 D' 中已被元素占有的位置, 其中位置 $A'_2(5)$ 处的值

为 2 表明了两个元素都占有此位置。

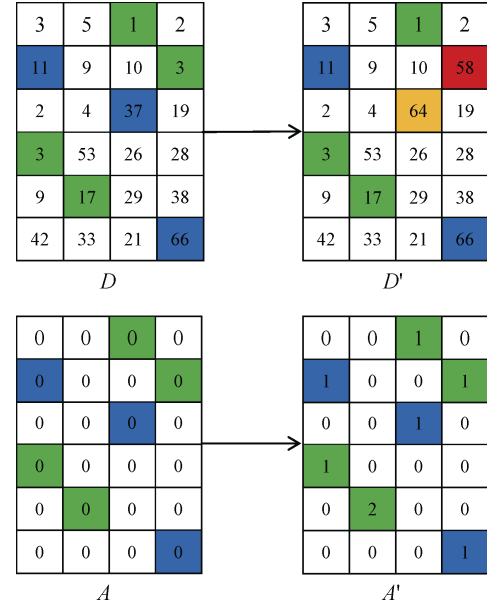


图 1 累计器添加两个元素示例

Figure 1 Example of adding two elements to the accumulator

图 2 中, 将元素 x_1 删除, 首先在 A' 中选取被其唯一占有的位置, 这里选取 $A'_3(1)$, 即红色部分所示。然后将 D' 中相同位置 $D'_3(1)$ 的值修改为 19 (使 x_1 在 D' 中路径上的值的和模 p 不为 0 即可) 得到更新后的累计器 D'' , 将 A' 中 x_1 所在路径上的值全部减 1 得到更新后的辅助信息 A'' 。

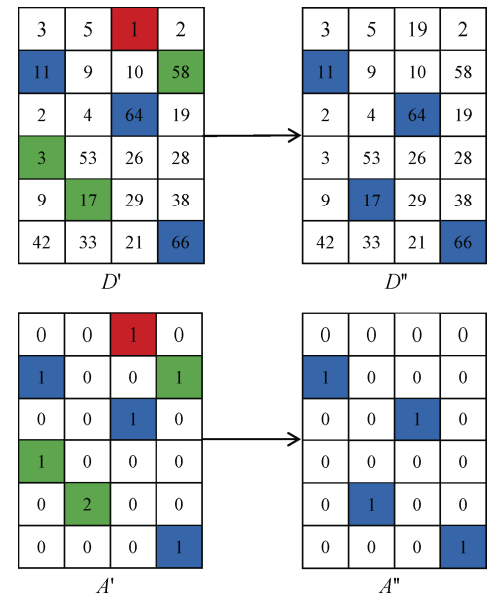


图 2 累计器删除一个元素示例

Figure 2 Example of deleting an element from the accumulator

3.1 安全性

在证明上述方案的安全性之前,我们先给出如下引理。

引理 1. 概率差引理: 设 E_1, E_2, E_3 是某个概率空间上的三个事件, 满足 $E_1 \wedge \neg E_3 \equiv E_2 \wedge \neg E_3$, 则 $|\Pr[E_1] - \Pr[E_2]| \leq \Pr[E_3]$ 。(证明见文献[19], P22)

接着我们证明上述方案构造的累计器的安全性取决于哈希函数的抗碰撞性和伪随机函数的随机性。具体地, 有如下理论。

定理 1. 如果函数 H 是一个抗碰撞的哈希函数, 函数 F 是一个伪随机函数, 那么上述方案构造的累计器是安全的。

证. 挑战者 $\mathcal{C}$ 和概率多项式时间敌手 $\mathcal{A}$ 进行如下几个实验:

Ex₀(·): 输入 λ

1. $S \leftarrow A(1^\lambda)$

2. $\mathcal{C}$: $\text{para} \leftarrow \text{Gen}(1^\lambda, n)$, $(D, A) \leftarrow \text{Acc}(1^\lambda, S)$

3. $(x, D', S') \leftarrow A^{\text{Add}_{D,A}(\cdot), \text{Del}_{D,A}(\cdot)}(D, 1^\lambda)$, 记询问过的元素集合为 Q , $x \notin Q$

4. 如果 $\text{Vrfy}(x, D') = 1$ 则输出 1; 否则输出 0

Ex₁(·): 输入 λ

1. $S \leftarrow A(1^\lambda)$

2. $\mathcal{C}$: $\text{para} \leftarrow \text{Gen}(1^\lambda, n)$, 如果存在 $x_i \neq x_j \in S$ 使得 $H(x_i) = H(x_j)$, 则停止实验。否则, $(D, A) \leftarrow \text{Acc}(1^\lambda, S)$, 将 D 发送给 $\mathcal{A}$ 。记录 x_i 与其哈希值 y_i 的对应关系 (x_i, y_i) 。

3. $(x, D', S') \leftarrow A^{\text{Add}_{D,A}(\cdot), \text{Del}_{D,A}(\cdot)}(D, 1^\lambda)$, 记询问过的元素集合为 Q , $x \notin Q$ 。

4. 如果 $\text{Vrfy}(x, D') = 1$ 则输出 1; 否则输出 0

Ex₂(·): 输入 λ

1. $S \leftarrow A(1^\lambda)$

2. $\mathcal{C}$: $\text{para} \leftarrow \text{Gen}(1^\lambda, n)$, 随机选取 n 个不同的 $y_i \in \{0, 1\}^\ell$, 令 $y_i = H(x_i)$, 记录 x_i 与其哈希值 y_i 的对应关系 (x_i, y_i) $(D, A) \leftarrow \text{Acc}(1^\lambda, S)$, 将 D 发送给 $\mathcal{A}$

3. $(x, D', S') \leftarrow A^{\text{Add}_{D,A}(\cdot), \text{Del}_{D,A}(\cdot)}(D, 1^\lambda)$, 记询问

过的元素集合为 Q , $x \notin Q$

4. 如果 $\text{Vrfy}(x, D') = 1$ 则输出 1; 否则输出

0 Ex₃(·): 输入 λ

1. $S \leftarrow A(1^\lambda)$

2. $\mathcal{C}$: $\text{para} \leftarrow \text{Gen}(1^\lambda, n)$, 随机选取 n 个不同的 $y_i \in \{0, 1\}^\ell$, 令 $y_i = H(x_i)$ 。记录 x_i 与其哈希值 y_i 的对应关系 (x_i, y_i) $(D, A) \leftarrow \text{Acc}(1^\lambda, S)$, 将 D 发送给 $\mathcal{A}$

3. $(x, D', S') \leftarrow A^{\text{Add}_{D,A}(\cdot), \text{Del}_{D,A}(\cdot)}(D, 1^\lambda)$: $\mathcal{A}$ 询问添加应答器 $\text{Add}_{D,A}(\cdot)$ 和删除应答器 $\text{Del}_{D,A}(\cdot)$ 。记 L 为包含当前所有哈希值的集合(动态变化)。在询问元素 $\tilde{x}$ (不在当前集合中) 的添加操作时, 如果存在 $y \in L$ 使得 $H(\tilde{x}) = y$, 则 $\text{Add}_{D,A}(\cdot)$ 输出 $\perp$ 。否则

$\text{Add}_{D,A}(\cdot)$ 执行添加操作并记录 $(\tilde{x}, H(\tilde{x}))$ 。在第 i 次询问后, $\mathcal{C}$ 将更新后的累计器 D'_i 发送给 $\mathcal{A}$ 。记最后更新后的累计值为 D' , 集合为 S' , $\mathcal{A}$ 输出 (x, D', S') , 记询问过的元素集合为 Q , $x \notin Q$

4. 如果 $\text{Vrfy}(x, D') = 1$ 则输出 1; 否则输出

0 Ex₄(·): 输入 λ

1. $S \leftarrow A(1^\lambda)$

2. $\mathcal{C}$: $\text{para} \leftarrow \text{Gen}(1^\lambda, n)$, 随机选取 n 个不同的 $y_i \in \{0, 1\}^\ell$, 令 $y_i = H(x_i)$ 。记录 x_i 与其哈希值 y_i 的对应关系 (x_i, y_i) $(D, A) \leftarrow \text{Acc}(1^\lambda, S)$, 将 D 发送给 $\mathcal{A}$

3. $(x, D', S') \leftarrow A^{\text{Add}_{D,A}(\cdot), \text{Del}_{D,A}(\cdot)}(D, 1^\lambda) \sum_{i=1}^n X_i$:

$\mathcal{A}$ 询问添加应答器 $\text{Add}_{D,A}(\cdot)$ 和删除应答器 $\text{Del}_{D,A}(\cdot)$ 。记 L 为包含当前所有哈希值的集合(动态变化)。在询问元素 $\tilde{x}$ (不在当前集合中) 的添加操作时,

$\text{Add}_{D,A}(\cdot)$ 随机选择一个值 $\tilde{y} \in \{0, 1\}^\ell \setminus L$, 令 $H(\tilde{x}) = \tilde{y}$ 并记录 $(\tilde{x}, \tilde{y})$ 。在第 i 次询问后, $\mathcal{C}$ 将更新后的累计器 D'_i 发送给 $\mathcal{A}$ 。记最后更新后的累计值为

D' , 集合为 S' , $\mathcal{A}$ 输出 (x, D', S') , 记询问过的元素集合为 Q , $x \notin Q$

4. 如果 $\text{Vrfy}(x, D') = 1$ 则输出 1; 否则输出

0 Ex₅(·): 输入 λ

1. $S \leftarrow A(1^\lambda)$

2. $\mathcal{C}: \text{para} \leftarrow \text{Gen}(1^\lambda, n)$, 从映射 ℓ 比特串到

$w \lceil \log m \rceil$ 比特串的函数集合中均匀随机选取一个函数 f , 将 para 中 F_k 用 f 替换。随机选取 n 个不同的

$y_i \in \{0, 1\}^\ell$, 令 $y_i = H(x_i)$ 。记录 x_i 与其哈希值 y_i 的对应关系 (x_i, y_i) 。 $(D, A) \leftarrow \text{Acc}(1^\lambda, S)$, 将 D 发送给 $\mathcal{A}$

3. $(x, D', S') \leftarrow A^{\text{Add}_{D,A}(\cdot), \text{Del}_{D,A}(\cdot)}(D, 1^\lambda)$: $\mathcal{A}$ 询问添加应答器 $\text{Add}_{D,A}(\cdot)$ 和删除应答器 $\text{Del}_{D,A}(\cdot)$ 。记 L 为包含当前所有哈希值的集合(动态变化)。在询问元素 $\tilde{x}$ (不在当前集合中)的添加操作时, $\text{Add}_{D,A}(\cdot)$ 随机选择一个值 $\tilde{y} \in \{0, 1\}^\ell \setminus L$, 令 $H(\tilde{x}) = \tilde{y}$ 并记录 $(\tilde{x}, \tilde{y})$ 。在第 i 次询问后, $\mathcal{C}$ 将更新后的累计器 D'_i 发送给 $\mathcal{A}$ 。记最后更新后的累计值为 D' , 集合为 S' , $\mathcal{A}$ 输出 (x, D', S') , 记询问过的元素集合为 Q , $x \notin Q$

4. 如果 $\text{Vrfy}(x, D') = 1$ 则输出 1; 否则输出 0

记事件 Ev_1 : 构造累计器时发生哈希值碰撞, 即存在 $x_i \neq x_j$, 满足 $H(x_i) = H(x_j)$ 。记事件 Ev_2 : 在实验询问阶段添加元素时发生哈希值碰撞。

在上述实验中, 实验 Ex_0 与原始敌手伪造实验 $\text{Expt}_{\mathcal{A}, \Pi}^{\text{forge}}(\cdot)$ 相同, 其功能是刻画敌手对于累计器构造方案 Π 的伪造攻击。实验 Ex_1 与实验 Ex_0 的功能相同, 实验过程唯一不同之处在于, 实验 Ex_1 的步骤 2 中, 如果存在 $x_i \neq x_j \in S$ 使得 $H(x_i) = H(x_j)$, 则停止实验。如果事件 Ev_1 不发生, 那么实验 Ex_1 与实验 Ex_0 是等价的, 即 $(\text{Ex}_0 = 1) \cap \neg \text{Ev}_1 \equiv (\text{Ex}_1 = 1) \cap \neg \text{Ev}_1$ 。由引理 1 可知

$$|\Pr[\text{Ex}_1 = 1] - \Pr[\text{Ex}_0 = 1]| \leq \Pr[\text{Ev}_1] \quad (1)$$

实验 Ex_2 与实验 Ex_1 的功能相同, 实验过程唯一不同之处在于, 实验 Ex_1 的步骤 2 中, 从哈希函数的值域中随机选取了 n 个不同的值 $y_i \in \{0, 1\}^\ell$ 。如果事件 Ev_1 不发生, 那么实验 Ex_2 与实验 Ex_1 是等价的, 即

$$(\text{Ex}_2 = 1) \wedge \neg \text{Ev}_1 \equiv (\text{Ex}_2 = 1) \wedge \neg \text{Ev}_1$$

由引理 1 可知

$$|\Pr[\text{Ex}_1 = 1] - \Pr[\text{Ex}_2 = 1]| \leq \Pr[\text{Ev}_1] \quad (2)$$

实验 Ex_3 与实验 Ex_2 的功能相同, 实验过程唯一不同之处在于, 实验 Ex_3 的步骤 3 中, 在询问元素的添加时, 如果发生哈希值碰撞, 则应答器回答 $\perp$ 。可以看到, 如果事件 Ev_2 不发生, 那么实验 Ex_3 与实验 Ex_2 是等价的, 即 $(\text{Ex}_2 = 1) \cap \neg \text{Ev}_2 \equiv (\text{Ex}_3 = 1)$

$\cap \neg \text{Ev}_2$ 。由引理 1 可知

$$|\Pr[\text{Ex}_2 = 1] - \Pr[\text{Ex}_3 = 1]| \leq \Pr[\text{Ev}_2] \quad (3)$$

实验 Ex_4 与实验 Ex_3 的功能相同, 实验过程唯一不同之处在于, 实验 Ex_4 的步骤 3 中, 只要是进行某个元素 $\tilde{x}$ (不在当前集合中)的添加询问, 挑战者不计算 $\tilde{x}$ 的真实哈希值, 而是从哈希函数的值域去掉当前已用哈希值集合 L 之后的集合 $\{0, 1\}^\ell \setminus L$ 中随机选择一个值 $\tilde{y}$, 并令 $H(\tilde{x}) = \tilde{y}$ 。可以看到, 如果事件 Ev_2 不发生, 那么实验 Ex_4 与实验 Ex_3 是等价的, 即 $(\text{Ex}_3 = 1) \wedge \neg \text{Ev}_2 \equiv (\text{Ex}_4 = 1) \wedge \neg \text{Ev}_2$ 。由引理 1 可知

$$|\Pr[\text{Ex}_3 = 1] - \Pr[\text{Ex}_2 = 1]| \leq \Pr[\text{Ev}_2] \quad (4)$$

由哈希函数的抗碰撞性, 存在两个可忽略函数 $\text{negl}_1(\cdot)$ 和 $\text{negl}_2(\cdot)$ 使得

$$\Pr[\text{Ev}_1] \leq \text{negl}_1(\lambda), \Pr[\text{Ev}_2] \leq \text{negl}_2(\lambda) \quad (5)$$

实验 Ex_5 与实验 Ex_4 的功能相同, 实验过程唯一不同之处在于, 实验 Ex_5 的步骤 2 中构造累计器时, 没有使用伪随机函数 F_k , 而是从映射 ℓ 比特串到 $w \lceil \log m \rceil$ 比特串的函数集合中均匀随机选取一个函数 f 。由函数 F_k 的伪随机性, 实验 Ex_5 与实验 Ex_4 是统计等价的。即

$$\Pr[\text{Ex}_4 = 1] = \Pr[\text{Ex}_5 = 1] \quad (6)$$

考虑在实验 Ex_5 中, 敌手 $\mathcal{A}$ 成功伪造了一个 x^* , 即 $\text{Ex}_5 = 1$ 。那么有如下的 3 种情形。

Ev_3 : 在当前哈希值集合 L 中, 存在 $y \in L$, 使得 $y = H(x^*)$, 从而 $v_1 = f(y) = v_2 = f(H(x^*))$ 。因此, $\sum_{j=1}^w \text{val}(D'_j(v_1(j))) = \sum_{j=1}^w \text{val}(D'_j(v_2(j))) = 0 \pmod{p}$ 。由于哈希函数 H 的抗碰撞性, 存在一个可忽略函数 $\text{negl}_3(\cdot)$, 使得

$$\Pr[\text{Ev}_3] \leq \text{negl}_3(\lambda)$$

Ev_4 : 在当前哈希值集合 L 中, 不存在 $y \in L$, 使得 $y = H(x^*)$, 但存在 $y \in L$ 使得

$$v_1 = f(y) = v_2 = f(H(x^*))$$

因此, $\sum_{j=1}^w \text{val}(D'_j(v_1(j))) = \sum_{j=1}^w \text{val}(D'_j(v_2(j))) = 0 \pmod{p}$ 。记 $v_1 = (\alpha_1, \dots, \alpha_w), v_2 = (\beta_1, \dots, \beta_w)$, 由 f 的随机性, 有 $\Pr[\alpha_i = \beta_i] = \frac{1}{m}$, 从而

$$\Pr[\text{Ev}_4] = \Pr[v_1 = v_2] = \frac{1}{m^w} \leq \text{negl}_4(\lambda)$$

Ev_5 : 在当前哈希值集合 L 中, 不存在 $y \in L$, 使得 $y = H(x^*)$, 且

$$\forall y \in L, v_1 = f(y) \neq v_2 = f(H(x^*)),$$

但有

$$\sum_j^w \text{val}(D'_j(v_2(j))) = 0 \pmod{p}.$$

这意味着在 D' 中随机地找一条路径使得路径上的值的和模 p 为 0。由于 D' 中的每一个位置上存放的都是 $\{1, \dots, p-1\}$ 中的一个随机值, 因此, 这个事件可转化为: 从 $\{1, \dots, p-1\}$ 中任意随机选取 w 个值

$p_i, i = 1, \dots, w$, 使得 $\sum_{i=1}^w p_i = 0 \pmod{p}$ 。分析可知,

$$\Pr[Ev_5] = \frac{1}{p-1} \leq \text{negl}_5(\lambda).$$

因此,

$$\begin{aligned} \Pr[Ex_5 = 1] &= \Pr[Ev_3 \vee Ev_4 \vee Ev_5] \\ &\leq \Pr[Ev_3] + \Pr[Ev_4] + \Pr[Ev_5] \\ &\leq \text{negl}_3(\lambda) + \frac{1}{m^w} + \frac{1}{p-1} \\ &\leq \text{negl}_3(\lambda) + \text{negl}_4(\lambda) + \text{negl}_5(\lambda) \\ &= \text{negl}_6(\lambda) \end{aligned} \quad (7)$$

利用式(7), 我们有

$$\begin{aligned} \Pr[Ex_0 = 1] &= \{|\Pr[Ex_0 = 1] - \Pr[Ex_1 = 1]| \\ &+ |\Pr[Ex_1 = 1] - \Pr[Ex_2 = 1]| \\ &+ |\Pr[Ex_2 = 1] - \Pr[Ex_3 = 1]| \\ &+ |\Pr[Ex_3 = 1] - \Pr[Ex_4 = 1]| \\ &+ |\Pr[Ex_4 = 1] - \Pr[Ex_5 = 1] + \Pr[Ex_5 = 1]|\} \\ &\leq \{|\Pr[Ex_0 = 1] - \Pr[Ex_1 = 1]| \\ &+ |\Pr[Ex_1 = 1] - \Pr[Ex_2 = 1]| \\ &+ |\Pr[Ex_2 = 1] - \Pr[Ex_3 = 1]| \\ &+ |\Pr[Ex_3 = 1] - \Pr[Ex_4 = 1]| \\ &+ |\Pr[Ex_4 = 1] - \Pr[Ex_5 = 1]| \\ &+ |\Pr[Ex_5 = 1]|\} \\ &\leq 2\text{negl}_1(\lambda) + 2\text{negl}_2(\lambda) + \text{negl}_6(\lambda) \end{aligned}$$

对于任何概率多项式时间的敌手 $\mathcal{A}$, 其攻击动

态伪累计器成功的概率就是实验 Ex_0 输出 1 的概率。由于 $2\text{negl}_1(\lambda) + 2\text{negl}_2(\lambda) + \text{negl}_6(\lambda) = \text{negl}_7(\lambda)$ 是一个可忽略函数。因此, 由动态伪累计器的安全性定义, 我们构造的累计器是安全的。

3.2 复杂性分析

添加一个元素 x , 需计算一次哈希运算、一次伪随机函数运算、一次赋值运算、 $w-1$ 次加法运算以及一次模运算, 即单个元素的添加需要 $O(w)$ 次算术计算。对于含有 n 个元素的集合, 计算其累计值需要 $O(nw)$ 次算术计算。删除操作执行的计算与添加操作类似, 删除 n 个元素需要 $O(nw)$ 次算术计算。

3.3 参数分析

当确定了矩阵 D 和矩阵 A 的尺寸参数, 即 m 和 w , 能够使得上述构造的伪累计器正常工作的集合尺寸 $|S|$ 不能超过某个上界。即 $|S|$ 不可能无限增大, 否则累计器在添加元素时会出错。例如, 添加一个元素 x' 时, $v' = F_k(H(x'))$, 可能会出现对任意的一个 $j \in \{1, 2, \dots, w\}$, 都有 $A_j(v'(j)) > 1$ 。这表明 D' 中 x' 被累计的所有位置上已经全部被其他的累计项所占有, 从而没有空位置来赋值 $\mathbb{Z}_p$ 中的一个随机元素使得它的这些位置的值的和满足模 p 为 0。但是这并没有排除 D' 中 x' 被累计的所有位置上的值(其他被累计元素在此位置的值的和满足模 p 为 0, 这使得我们需要对 D' 中除被累计元素所在的路径上所有值的和满足模 p 为 0 以外, 随机选取一条路径使得其上所有值的和模 p 为 0 的概率是可忽略的。由定理 1 中证明可知, 这个概率是可忽略的。

累计器所允许的集合尺寸的上界 $|S|$ 除了保证了累计器能够正常添加一个元素, 还需要保证, 当累计器累计的元素个数接近所允许的上界时, 可以正常删除一个元素。正常删除一个元素意味着这个元素在累计器中的路径中至少存在一个位置是这个元素唯一占有, 否则这个元素在添加进来时就已经出错。实际上, 如果能够保证正常添加一个元素, 就能够保证正常地删除一个元素。这是因为, 我们对正常添加一个元素的限制是, 累计器中, 新添加的元素的路径中至少存在一个空位置。这就使得我们在删除这个元素时可以在这个空位置上放入一个新的随机值, 使得之后这个元素在累计器中的路径上的值的和模 p 不为 0。

现在我们正式讨论当给定 m 和 w 时, 累计器所能容忍的集合尺寸上界 $n = |S|$ 需要满足的条件。正如

上文所述,要使得累计器的功能正常,需要保证每个元素被累计进累计器之后,其在累计器中的路径上至少存在一个位置是被这个元素唯一占有的。我们考虑事件 E: 一个新的元素被累计到累计器中时,其在累计器中的路径上不存在一个位置是被其唯一占有的。由于一个元素 x 在累计器中路径上的位置实际上对应于 $F_k(H(x))$ 的分量值。因此,这个事件在形式上可描述为: 考虑累计器已经累计了 $n-1$ 个元素 $\{x_1, x_2, \dots, x_{n-1}\}$, 有

$v_i = F_k(H(x_i)) = (a_{i1}, a_{i2}, \dots, a_{iw}), i = 1, \dots, n-1$ 。添加第 n 个元素 x_n 时, 有 $F_k(H(x_n)) = (a_{n1}, a_{n2}, \dots, a_{nw})$, 对所有的 $j = 1, \dots, w$, 都存在 $i \in \{1, \dots, n-1\}$ 使得 $a_{nj} = a_{ij}$ 。考虑如下事件:

事件 $E_j, j = 1, \dots, w$: 存在 $i \in \{1, \dots, n-1\}$, 满足 $a_{ij} = a_{nj}$ 。由 F 的随机性, $E_1, \dots, E_w$ 是相互独立的。且有

$$E = \bigcap_{j=1}^w E_j.$$

固定 $j \in \{1, \dots, w\}$, 事件 $E_{ij}, i = 1, \dots, n-1$: 第 i 个向量 v_i 的第 j 个值满足 $a_{ij} = a_{nj}$ 。则有

$$E_j = \bigcup_{i=1}^{n-1} E_{ij}.$$

由 F 的随机性, 有

$$\Pr[E_{ij}] = \frac{1}{m}$$

以及

$$\Pr[E_j] = \Pr\left[\bigcup_{i=1}^{n-1} E_{ij}\right] \leq \bigcup_{i=1}^{n-1} \Pr[E_{ij}] = \frac{n-1}{m},$$

从而

$$\Pr[E] = \Pr\left[\bigcap_{j=1}^w E_j\right] = \prod_{j=1}^w \Pr[E_j] \leq \left(\frac{n-1}{m}\right)^w.$$

置 $m \geq 2n$, w 是关于安全参数 λ 的一个多项式, 那么

$$\Pr[E] \leq \left(\frac{n-1}{m}\right)^w < \left(\frac{1}{2}\right)^w$$

是可忽略的。这说明, 只要在累计器所允许累计的集合尺寸上界 ($n \leq m/2$) 之内, 一个元素被累计到累计器中时, 其在累计器的路径上至少存在一个位置是被这个元素唯一占有的。

正如前文所述, Chase 和 Miao^[27]在他们的集合交集方案中, 参与方 P_2 实际上给 P_1 传输了一个 $m \times w$ 的矩阵。在他们的实验中, 选取 $m = n = 2^{16}$,

$w = 609$ 时, P_2 的存储开销为 4.76MB, 在 LAN 网络中两方求交集仅需 0.63s, 在 30Mbps 的网络中也仅需 2.66s。选取 $m = n = 2^{24}, w = 633$ 时, P_2 的存储开销为 1226MB, 在 LAN 网络中两方求交集需 190s, 在 30Mbps 的网络中需 431s。我们的伪累计器构造不涉及数据交互式传输, 且与他们相同参数设置下, 存储的是的两个矩阵, 因此存储开销大大小于他们同参数设置下的开销。对于仅需要成员关系和非成员关系验证的应用来说, 我们的参数设置是合理的。在实际应用当中是可行的。

3.4 性能对比

如表 1, 我们将伪累计器构造方案同前人的累计器构造方案在性能方面做了详细的对比。表中“动态”属性表示文中构造的累计器是否同时支持元素的添加和删除; “陷门”属性表示构造的累计器是否有陷门信息; “证据-1”表示构造的累计器在验证成员关系时是否需要相应的证据; “证据-2”表示构造的累计器在验证非成员关系时是否需要相应的证据, 其中“-”表示该方案不支持非成员关系验证功能。

对比结果表明, 在“动态”“陷门”“证据-1”“证据-2”这四个属性上, 我们的累计器构造方案比前人的累计器构造方案具有相同或更好的性能, 总体性能上更优于前人的累计器构造方案。

表 1 相关工作性能对比

Table 1 The performance comparison of related work

方案	动态	陷门	证据-1	证据-2
[1]	否	有	需要	-
[4]	是	有	需要	-
[5]	是	有	需要	-
[15]	是	无	需要	需要
[16]	否	无	需要	-
[17]	是	有	需要	需要
[20]	是	有	需要	-
[21]	是	有	需要	需要
[22]	否	无	不需要	-
[23]	是	无	需要	需要
[24]	是	有	需要	-
[26]	是	有	需要	需要
Ours	是	无	不需要	不需要

4 分级访问控制系统

本节介绍上述方案构造的累计器如何用来构造住宅小区分级访问控制系统。

住宅小区分级访问控制系统: 首先, 为实现小

区大门的门禁系统, 我们将整个小区所有业主门禁卡信息构成一个集合, 在累计器构造的第二步中, 记伪随机函数的密钥为 k_0 , 接着构造累计器 D'_0 和辅助信息 A'_0 , 这便实现了大门的访问控制系统。为实现单元楼 $i, i=1, 2, \dots, t$ 的门禁系统, 将单元楼 i 的所有业主的门禁卡信息构成一个集合, 在累计器构造的第二步中, 记伪随机函数的密钥为 k_i , 接着构造累计器 D'_i 和辅助信息 A'_i , 这便实现了单元楼 i 的访问控制系统。需要注意的是, 这里的 $k_i, i=0, 1, \dots, t$ 必须两两不同。由于累计器是动态的, 支持元素的添加和删除, 因此这里构造的访问控制系统也支持权限的添加和删除。例如, 可将分发给新来的业主的门禁卡信息添加到大门门禁系统 D'_0 中, 同时也添加到业主所在的单元楼 j 的门禁系统 D'_j 中。业主的搬离, 可简单地将门禁卡收回, 或者将门禁卡信息从大门门禁系统和单元楼门禁系统删除(如果门禁卡丢失)。

5 总结

我们引入了“动态伪累计器”的概念, 并给出了一个具体的构造, 使得成员关系和非成员关系的验证不需要相应的证据, 且支持元素的动态添加和删除。累计器是非陷门的, 其安全性不依赖于陷门信息的保密性。展示了我们构造的动态伪累计器在现实生活中的具体应用。如何构造一个累计器尺寸不依赖于被累计项数量, 且不需要给出成员关系和非成员关系证据的非陷门动态累计器是我们将来的一个主要研究问题。

参考文献

- [1] Benaloh J, de Mare M. One-Way Accumulators: A Decentralized Alternative to Digital Signatures[C]. *EUROCRYPT '93: Workshop on the theory and application of cryptographic techniques on Advances in cryptography*, 1994: 274-285.
- [2] Naor M, Nissim K. Certificate Revocation and Certificate Update[C]. *IEEE Journal on Selected Areas in Communications*, 2002: 561-570.
- [3] Buldas A, Laud P, Lipmaa H. Accountable Certificate Management Using Undeniable Attestations[C]. *The 7th ACM conference on Computer and Communications Security*, 2000: 9-17.
- [4] Camenisch J, Kohlweiss M, Soriente C. An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials[C]. *The 12th International Conference on Practice and Theory in Public Key Cryptography: PKC '09*, 2009: 481-500.
- [5] Camenisch J, Lysyanskaya A. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials[C]. *The 22nd Annual International Cryptology Conference on Advances in Cryptology*, 2002: 61-76.
- [6] Baldimtsi F, Camenisch J, Dubovitskaya M, et al. Accumulators with Applications to Anonymity-Preserving Revocation[C]. *2017 IEEE European Symposium on Security and Privacy*, 2017: 301-315.
- [7] Sander T, Ta-Shma A. Flow Control: A New Approach for Anonymity Control in Electronic Cash Systems[C]. *The Third International Conference on Financial Cryptography*, 1999: 46-61.
- [8] Miers I, Garman C, Green M, et al. ZeroCoin: Anonymous Distributed E-Cash from Bitcoin[C]. *2013 IEEE Symposium on Security and Privacy*, 2013: 397-411.
- [9] Ahn J H, Boneh D, Camenisch J, et al. Computing on Authenticated Data[M]. *Theory of Cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012: 1-20.
- [10] Canard S, Jambert A. On Extended Sanitizable Signature Schemes[C]. *Cryptographers' Track at the RSA Conference*. Berlin, Heidelberg: Springer, 2010: 179-194.
- [11] Pöhls H C, Samelin K. On Updatable Redactable Signatures[M]. *Applied Cryptography and Network Security*. Cham: Springer International Publishing, 2014: 457-475.
- [12] Garman C, Green M, Miers I. Decentralized Anonymous Credentials[C]. *Proceedings 2014 Network and Distributed System Security Symposium*, 2014.
- [13] Fromknecht C, Velicanu D, Yakoubov S. A Decentralized Public Key Infrastructure with Identity Retention[J]. *IACR Cryptology EPrint Archive*, 2014, 2014: 803.
- [14] Genty C, Ramzan Z. RSA Accumulator Based Broadcast Encryption[C]. *In Information Security, 7th International Conference, ISC 2004, Palo Alto, CA, USA, September 27-29, 2004, Proceedings*, 2004: 73-86.
- [15] Boneh D, Bünz B, Fisch B. Batching Techniques for Accumulators with Applications to IOPs and Stateless Blockchains[C]. *Advances in Cryptology - CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I*, 2019: 561-586.
- [16] Sander T. Efficient Accumulators without Trapdoor Extended Abstract[C]. *International Conference on Information and Communications Security*. Berlin, Heidelberg: Springer, 1999: 252-262.
- [17] Li J T, Li N H, Xue R. Universal Accumulators with Efficient Nonmembership Proofs[M]. *Applied Cryptography and Network Security*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007: 253-269.
- [18] Goodrich M T, Tamassia R, Hasić J. An Efficient Dynamic and Distributed Cryptographic Accumulator[M]. *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002: 372-388.
- [19] Xue R. Public key encryption theory[M]. Beijing: Science Press, 2016.
(薛锐. 公钥加密理论[M]. 北京: 科学出版社, 2016.)
- [20] Nguyen L. Accumulators from Bilinear Pairings and Applications[C]. *In Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18, 2005, Proceedings*, 2005: 275-292.
- [21] Damgård, Triandopoulos N. Supporting Non-membership Proofs with Bilinear-map Accumulators[EB/OL]. *IACR Cryptology ePrint Archive*, 2008:538.

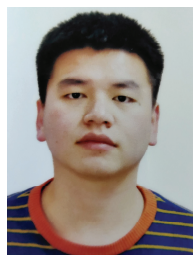
- [22] Nyberg K. Fast Accumulated Hashing[C]. *In Fast Software Encryption, Third International Workshop, Cambridge, UK, February 21-23, 1996, Proceedings, 1996*, 83-87.
- [23] Camacho P, Hevia A, Kiwi M, et al. Strong Accumulators from Collision-Resistant Hashing[J]. *International Journal of Information Security*, 2012, 11(5): 349-363.
- [24] Wang P S, Wang H X, Pieprzyk J. A New Dynamic Accumulator for Batch Updates[M]. *Information and Communications Security*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007: 98-112.
- [25] Camacho P, Hevia A. On the Impossibility of Batch Update for Cryptographic Accumulators[C]. *The First international conference on Progress in cryptology: cryptology and information security in Latin America*, 2010: 178-188.
- [26] Thakur S. Batching Non-Membership Proofs with Bilinear Accumulators[J]. *IACR Cryptology EPrint Archive*, 2019, 2019: 1147.
- [27] Chase M, Miao P H. Private Set Intersection in the Internet Setting from Lightweight Oblivious PRF[C]. *Advances in Cryptology - CRYPTO 2020: 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III*, 2020: 34-63.



关诚 于 2018 年在红河学院信息与计算科学专业获得理学学士学位。现在中国科学院信息工程研究所信息安全国家重点实验室攻读硕士学位。研究领域为公钥密码学。研究兴趣包括: 安全多方计算、密码协议。Email: guancheng@iie.ac.cn



薛锐 于 1999 年在北京师范大学数学专业获得理学博士学位。现任中国科学院信息工程研究所信息安全国家重点实验室研究员。研究领域为公钥密码学、安全协议。研究兴趣包括: 公钥密码学、安全协议的形式化分析。Email: xuerui@iie.ac.cn.



员凯立 于 2015 年在湖南大学计算机科学与技术专业获得工学学士学位。现在中国科学院信息工程研究所攻读博士学位。研究领域为公钥密码学。研究兴趣包括: 格密码、函数加密。Email: yunkaili15@mails.ucas.ac.cn