arXiv:2301.09236v1 [quant-ph] 23 Jan 2023

基于抗碰撞哈希函数的公钥量子货币的（不）可信性

PrabhanjanAnanth* Zihan Hu† Henry Yuen‡UCSB 清华大学 哥伦比亚大学

摘要

公钥量子货币是一种密码学方案，旨在利用高度纠缠的量子态作为货币，使其既可公开验证，又能由于物理定律而抵抗伪造。尽管人们对此投入了大量关注，但基于标准密码学假设构造可证明安全的公钥量子货币方案，始终是一个难以实现的目标。即使是提出看似安全的候选方案，也颇具挑战。

这些困难促使人们更深入、系统地研究公钥量子货币方案的结构，以及其所能依赖的假设。受此启发，我们首次实现了量子货币与密码学原语之间的黑盒分离。具体而言，我们证明，抗碰撞哈希函数无法作为黑盒，用于构造这样的公钥量子货币方案：其中钞票验证会向哈希函数发起经典查询。我们的结果结合了量子复杂性理论中的新颖态合成技术与模拟技术，其中包括 Zhandry 压缩预言机技术。

1 引言

不可克隆密码学是量子密码学中的一个新兴领域，它利用量子力学的不可克隆原理 [WZ82；Die82] 以实现经典上不可能实现的密码学原语。多年来，人们提出并研究了许多有趣的不可克隆原语，包括量子复制保护 [Aar09]，一次性程序 [BGS13]，安全软件租赁 [AL21]，不可克隆加密 [BL20]，带认证删除的加密 [BI20],采用不可克隆解密密钥的加密 [GZ20；CLLZ21]，以及令牌化签名 [BS16]。

最古老且（可以说）最受欢迎的不可克隆原语之一是量子货币。Wiesner 在一篇开创性论文中首次提出了这一概念 [Wie83]。量子货币方案使银行能够发行以量子态表示的数字货币。简单来说，其安全性保证是：计算上无法生成伪造的数字货币态。也就是说，恶意用户得到一个货币态后，无法生成两个都能通过预先设定的验证程序的货币态。这里可以考虑两种概念。第一种是 私钥 量子货币，其验证程序是私有的。也就是说，要检查货币态是否有效，我们需要将该量子态提交给银行，由银行判定其有效性。更实用的一种概念是 公钥 量子货币，任何人都可以验证货币态的有效性

*prabhanjan@cs.ucsb.edu
†huzh19@mails.tsinghua.edu.cn
‡hyuen@cs.columbia.edu

。尽管私钥货币方案已得到广泛研究，并且已有许多构造被提出，其中包括信息论构造，但公钥量子货币方案却并非如此。

Aaronson 和 Christiano [AC13] 首次证明了在预言机模型中实现信息论安全的公钥量子货币是可行的；这意味着该方案中的所有算法（例如铸造和验证算法）在执行过程中都会查询一个黑盒预言机。在标准模型（即非预言机模型）中，已知有两类构造量子货币的方法：

- 第一类构造借鉴了数学不同领域中的复杂工具，例如纽结理论 [FGH+12]、四元数代数 [KSS21] 和格 [Zha21；KLS22]。近期的几项研究表明，这类构造容易受到密码分析攻击 [Rob21；BDG22]。我们对这些候选方案安全性的理解仍处于起步阶段。

- 第二类构造基于经过充分研究（或者说 研究得更为充分）的密码学原语。这一类中有基于不可区分混淆的构造 [Zha21；Shm22a；Shm22b]这类构造最初由 Zhandry 提出 [Zha21]。

我们关注第二类。基于现有原语构造，尤其是基于经过充分研究的假设的构造，将使公钥量子货币建立在更坚实的基础之上。遗憾的是，现有的不可区分混淆构造要么是后量子**不**安全的 [AJL+19；JLS21；JLS22] ，要么基于新提出的密码学假设 [GP21；BDGM20；WW21；DQV+21] ，而这些假设已遭受密码分析攻击 [HJL21]。

我们的工作旨在理解如何从基础且经过充分研究的密码学原语构造公钥量子货币。我们通过黑盒分离的视角来探索这一方向。黑盒分离已在经典密码学中得到广泛研究 [Rud91; Sim98; GKM+00; RTV04; BM09; DLMM11; GKLM12; BDV17]。如果存在一个由预言机定义的计算世界，其中 B 存在而 A 不存在，我们就说原语 A 无法以黑盒方式从另一原语 B 构造出来。换言之，这些分离排除了以黑盒方式使用原语 B 来构造原语 A 的可能性。在这种情况下，我们称 A 与 B之间存在黑盒分离。黑盒分离对于理解不同密码学原语之间的关系至关重要。或许令人惊讶的是，它们也为密码学构造的设计提供了指引。一个这样的例子是基于身份的加密（IBE）。有几项研究 [BPR+08；PRV12]证明，使用黑盒构造从判定性 Diffie-Hellman（DDH）假设构造 IBE 存在困难；这一结果促成了 [DG17]的工作，他们使用非黑盒技术从 DDH 构造了 IBE。

1.1 我们的工作

不可克隆密码学的黑盒分离。我们开启了对不可克隆密码学中黑盒分离的研究。在这项工作中，我们研究公钥量子货币与（后量子安全）抗碰撞哈希函数之间的黑盒分离。据我们所知，我们的工作迈出了第一步，排除从经过深入研究的假设出发构造公钥量子货币的某些方法。

模型。我们首先讨论用于证明黑盒分离的模型。我们考虑两个预言机：第一个是随机预言机 $\mathcal{R}$（即均匀随机函数），第二个是 PSPACE 预言机（即能够解决 PSPACE-完全问题的预言机）。我们研究在存在 $\mathcal{R}$和 PSPACE的情况下，量子货币方案和抗碰撞哈希函数是否可行。也就是说，量子货币方案中的所有算法以及对手实体都可以访问预言机 $\mathcal{R}$ 和 PSPACE。

我们可以通过两种方式对量子算法访问预言机进行建模。第一种是经典访问，在这种方式下，量子货币方案中的算法只能对预言机进行经典查询；也就是说，每次向预言机发出的查询在转交给预言机之前，都会在计算基下测量。如果算法 A 以经典访问方式访问预言机（例如 $\mathcal{U}$），我们将其记为 $\mathsf{A}^{\mathcal{U}}$。第二种是量子访问，在这种方式下，算法可以进行叠加查询。也就是说，算法可以向预言机 $\mathcal{O}$ 提交形式为 $\sum_{x,y}\alpha_{x,y}|x\rangle|y\rangle$ 的量子态，并收到$\sum_{x,y}\alpha_{x,y}|x\rangle|\mathcal{O}(x)\oplus y\rangle$。如果算法 A 以量子访问方式访问预言机 $\mathcal{U}$，我们将其记为$\mathsf{A}^{|\mathcal{U}\rangle}$。

我们的最终目标是在量子访问模型中获得黑盒分离，在该模型下，量子货币方案中的算法可以对预言机进行叠加查询。然而，要实现这一目标存在两个主要障碍。

首先，分析量子密码学中的量子访问模型一直极具挑战性。例如，目前尚不清楚如何将密钥协商协议与单向函数之间的黑盒分离推广到量子访问情形——前者是一种经典密码学原语 [IR90]。针对特殊情形的尝试已经遇到了重大障碍 [ACC+22]，并且与量子查询复杂度中由来已久的猜想（例如 Aaronson-Ambainis 猜想）有关 [AA09]）。

其次，我们必须应对量子货币本质上是一种量子密码学原语这一难题。要实现黑盒分离，就需要设计一个对手，使其能够仅凭一份单独的量子钞票副本就能有效地克隆它。此处会遇到一些独特的量子问题，例如不可克隆定理 [WZ82；Die82]以及测量钞票通常会扰动它这一事实。

我们通过简化问题并专注于上述第二个障碍，朝着最终目标迈出了初步的一步：我们证明了黑盒分离，其中量子货币方案中的钞票验证算法对随机预言机进行经典查询$\mathcal{R}$ （但仍可对 PSPACE 预言机进行量子查询），而铸造算法仍可对 $\mathcal{R}$ 和 PSPACE 预言机进行量子查询。正如我们将看到的，即便是量子货币方案这一特殊情形，分析起来也已颇具挑战，并非易事。我们相信，我们的技术最终或许能够推广到一般情形（如果一般情形中确实存在黑盒不可能性！），在该情形下，所有算法都可以对所有预言机进行量子查询；此外，这些技术还有望帮助证明其他量子密码学原语的黑盒分离。

主定理。 我们将更正式地陈述定理。量子货币方案由三个量子多项式时间（QPT）算法组成，即 (KeyGen, Mint, Ver)，其中 KeyGen生成一对公钥-私钥对， Mint 使用秘密密钥生成货币态及其对应的序列号，最后， Ver 使用公钥判定货币态的有效性。我们考虑可调用预言机的量子货币方案，其中这些算法可以访问上文定义的随机预言机 $\mathcal{R}$ 和 PSPACE 预言机。

定理 1（非正式）。 任何公钥量子货币方案（$\mathsf{KeyGen}^{|\mathcal{R}\rangle,|\mathsf{PSPACE}\rangle}, \mathsf{Mint}^{|\mathcal{R}\rangle,|\mathsf{PSPACE}\rangle}, \mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$）都是不安全的。

这里所说的不安全性是指：存在一个量子多项式时间（QPT）对手 $\mathcal{A}$，使得 $\mathcal{A}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$ 在给定一个货币态（pk，ρ_s，s）时，其中 pk 是公钥，s 是序列号，能够以不可忽略的概率生成两个（可能纠缠的）量子态，使它们都能通过针对同一序列号的验证检查 s。该概率是对 $\mathcal{R}$ 的随机性，以及 KeyGen、Mint 和 $\mathcal{A}$ 的随机性取平均得到的。需要指出的是，只有 KeyGen 和 Mint 能够量子访问 $\mathcal{R}$，而 Ver 只能经典访问。另一方面，我们证明，对手 $\mathcal{A}$ 只需经典访问 $\mathcal{R}$。

此外，我们注意到，随机预言机 $\mathcal{R}$ 构成了一个抗碰撞哈希函数，即使面对能够向 $(\mathcal{R}, |\mathsf{PSPACE}\rangle)$ [Zha15] 发起查询的 QPT 对手也是如此。我们注意到，$\mathcal{R}$ 即使对手能够向 量子 查询 $\mathcal{R}$，而不只是进行经典查询，它仍然是抗碰撞的。

推论。 我们的主要结果排除了一类公钥量子货币构造，这类构造 (a) 将安全性建立在抗碰撞哈希函数之上，(b) 以黑盒方式使用哈希函数，且 (c) 验证算法对哈希函数进行经典查询。显然，若能将结果推广到验证算法可以对哈希函数进行量子查询的情形，将会很有价值。然而，要超越经典验证查询还存在一些概念上的挑战（我们将在第 2.2.3 节中进一步讨论）。

混合经典-量子查询模型中的量子货币方案本身就很有趣，也是一个有充分动机的研究场景。例如，在 Zhandry 的公钥量子货币方案中 [Zha21], ，铸造过程只需经典访问底层密码学原语（将使用密码学原语的组件视为黑盒时），而验证程序则进行量子查询。在 Coladangelo 等人提出的复制保护构造中 [CLLZ21； CMP20], ，在 [CLLZ21] 的情形下，复制保护算法仅对密码学原语进行经典查询；在 [CMP20]的情形下，则仅对随机预言机进行经典查询；而这两种构造中的求值算法都会进行量子查询。最后，在 [AKL+22]中的不可克隆加密构造里，所有算法都只对随机预言机进行经典查询。鉴于这些构造，我们认为，理解混合经典-量子查询模型下不可克隆密码系统有哪些可行或不可能实现的方案十分重要。

其次，我们认为，混合经典-量子查询模型是一个有用的试验平台，可用于开发黑盒分离所需的技术，并深入了解不可克隆密码学原语的结构。即使在这个特殊情形下，要实现我们对定理 1的黑盒分离，仍有许多技术和概念上的挑战需要克服。我们认为，本文提出的技术将为未来研究不可克隆密码学中的黑盒分离提供有益的起点。

其他分离。 作为主要结果的一个推论，我们得到了公钥量子货币与许多其他研究成熟的密码学原语（例如单向函数、私钥加密和数字签名）之间的黑盒分离。

我们的结果还在与上述相同的设定下，给出了公钥量子货币与可坍缩哈希函数之间的分离；也就是说，当 Ver 对 $\mathcal{R}$进行经典查询时。这是由 Unruh 的一项结果 [Unr16] 推得的：他证明了随机预言机是可坍缩的。可坍缩

哈希函数是抗碰撞哈希函数的量子对应物。非正式地说，如果对手无法区分映射到某个随机输出$|\psi\rangle$的输入的均匀叠加态（例如 y），与通过在计算基下测量 $|\psi\rangle$而得到的计算基态，那么该哈希函数就是可坍缩的。Zhandry [Zha21]证明了，抗碰撞但不可坍缩的哈希函数意味着公钥量子货币的存在。因此，我们的结果排除了一类基于可坍缩函数构造量子货币的方法，增进了我们对二者关系的理解。

致谢。我们感谢匿名会议评审、Qipeng Liu、Yao Ching Hsieh 和 Xingjian Li 提供的宝贵意见。HY 得到了 AFOSR 项目资助 FA9550-21-1-0040 和美国国家科学基金会 CAREER 奖 CCF-2144219 的支持。

2 我们的技术概述

我们从高层次概述证明定理 1所涉及的技术。但首先，我们将简要讨论预言机辅助公钥量子货币方案的正确性保证。

可重复使用性。在量子货币方案 (KeyGen, Mint, Ver) 中，我们要求 Ver 以压倒性概率接受由 Mint 生成的量子态和序列号。然而，据我们所知，Ver在验证过程中可能会破坏该量子态。一般而言，一种更实用的正确性定义是可重复使用性，即货币态可以反复验证而不失去其有效性。通常可以利用温和测量引理 [Win99] 证明正确性意味着可重复使用性。然而，正如 [AK22], 所指出的，当 Ver只能经典访问预言机时，情况并非如此。具体而言， Ver 对 $\mathcal{R}$具有经典访问权限。因此，在这种情形下，我们需要明确地定义可重复使用性。粗略地说，我们要求满足以下条件：假设我们对由 Mint 生成的货币态 $\rho^{(0)}$ 执行 Ver ，且验证算法以概率 δ接受。剩余态可能是一个不同的量子态 $\rho^{(1)}$，而 Ver 对其执行验证时，也会以接近 δ的概率接受。事实上，即使我们将验证过程运行多项式次，最终得到的量子态仍应以接近 δ的概率被Ver 接受。

2.1 预热：当 $\mathcal{R}$ 缺席时的不安全性

为了发展证明定理 1的技术，我们先来解决一个更简单的命题。假设我们有一个安全的公钥量子货币方案(KeyGen， Mint， Ver)。这意味着任何 QPT 对手都无法破坏该方案的安全性。但对于带预言机辅助的对手呢？更具体地说，我们提出以下问题：是否存在一个量子多项式时间算法，在获得对 PSPACE 预言机的量子访问后，能够破坏(KeyGen， Mint， Ver)的安全性？也就是说，给定$(s,\ \rho_s)$，其中s 是一个序列号，而 ρ_s 是由 Mint生成的有效钞票，它应当能够针对同一个序列号 s生成两个都能通过验证器验证的量子态。

即使是这个看似简单的问题，似乎也颇具挑战性！让我们来理解其中的原因。经典密码学原语（甚至是后量子安全的经典密码学原语），例如加密方案或数字签名，可能会被能够访问甚至 NP 预言机的高效对手攻破。这是因为，我们可以将攻破该方案的问题高效归约为判定某种语言中的成员资格问题。例如，为了成功攻破一个加密

方案，对手必须判定实例（pk， ct， m） $\in L$，其中 pk 是公钥，ct 是密文， m 是消息，而 L 由形如（pk， ct， m）的实例组成，其中 ct 是相对于公钥 pk对 m 进行加密所得的结果。这里隐含地使用了这样一个事实： pk， ct， m都是二进制字符串。在量子货币的情况下采用类似的方法，会得到量子 实例；而目前尚不清楚如何利用 PSPACE，或更一般地说，如何利用任意语言的判定器来完成该归约。

合成见证态。 为了回答上述问题，我们将利用 PSPACE 破坏量子货币方案安全性的任务，归约为寻找被量子多项式空间中的验证器接受的量子态。该归约基于 Rosenthal 和 Yuen 的以下观察 [RY21]：量子多项式空间算法（通常可能包含中间测量）可计算的一组纯态，可以由具有量子访问 PSPACE 预言机的量子多项式时间算法合成。 [RY21] 的结果隐含了以下重要一点：为了利用 PSPACE预言机合成该量子态，我们必须知道生成这些纯态的量子多项式空间算法的完整描述。

更具体地说，我们证明以下结论：对于每个[1] 验证密钥 pk、序列号 s，都存在一个纯态[2] $\rho_{\mathsf{pk},s}$ 它以非可忽略的概率被 Ver(pk， s， ·)接受，而且可以由量子多项式空间算法生成。

第一种尝试是遵循经典暴力搜索算法。具体而言，我们重复以下步骤指数次：均匀随机地猜测一个量子态 ρ ，如果 ρ 被 Ver(pk， s， ·)以不可忽略的概率接受，就输出 ρ 并终止。（如果尝试次数用尽，则输出任意量子态。）然而，这种尝试存在两个问题。首先，一般来说，我们不清楚如何在多项式空间内计算 Ver(pk， s， ρ)的接受概率（ρ 需要指数位才能表示）。其次，在计算接受概率时， ρ 可能会被破坏。

为了解决第一个问题，我们注意到，对接受概率进行估算就已经足够了，而且可以使用 Marriott 和 Watrous 提出的方法来实现 [MW05] （称为 MW 技术）。MW 技术可以在仅有一份量子态副本的情况下，高效估算验证算法对该量子态的接受概率。此外，它对量子态的扰动很小：剩余态的期望接受概率不会显著下降，从而解决了第二个问题。

这引出了我们的第二次尝试。我们重复以下过程指数多次：对最大混合态应用 MW 技术，如果估计的接受概率恰好非可忽略，则输出剩余态并终止。（如果所有迭代都失败，则输出任意量子态。）

由于 MW 技术是高效的，该算法只使用多项式空间。此外，直观地说，鉴于这样的量子态确实存在，我们可以得到一个以非可忽略的接受概率被 Ver 接受的量子态。因为如果这样的量子态存在，根据一个简单的凸性论证，我们不失一般性地假设它是纯态。最大混合态可以视为包含该纯态的某个基的均匀混合。粗略地说，我们以指数倒数概率从该纯态出发，因此经过指数多次迭代后，我们以压倒性概率找到它。此方法几乎成功了，只是它输出的是一个混合量子态

[1]严格来说，我们证明了一个较弱的结论，该结论对（pk， s）中绝大多数情况都成立。[2]严格来说，我们要求 $\rho_{\mathsf{pk},s}$ 的约化密度矩阵被 Ver 接受。

一般而言，但 [RY21] 中已知的方法只能处理纯态。这有两个原因。首先，我们从最大混合态出发；其次，MW 技术涉及中间测量。

我们的最终尝试在第二次尝试的基础上做了以下细微改动。为解决第一个问题，它从一个最大纠缠态（而非最大混合态）开始，并且只对其中一半进行操作。为解决第二个问题，它通过延后所有中间测量，以相干的方式运行 MW 过程。这样，我们最终会得到一个纯态，其约化密度矩阵与第二次尝试的输出态相同。

2.2 存在 $\mathcal{R}$ 时的不安全性

到目前为止，我们考虑的是这样一个任务：在算法无法访问任何预言机的情况下，破坏量子货币方案的安全性。现在让我们回到预言机辅助量子货币方案，在这类方案中，所有算法（诚实算法和对手算法）都可以访问 $\mathcal{R}$ 和 $|\mathsf{PSPACE}\rangle$。我们的目标是构造一个能够破坏量子货币方案安全性的对手。但我们不是刚刚解决了这个问题吗？回想一下，调用 [RY21], 时，关键在于我们知道该多项式空间算法的完整描述，才能合成该量子态。然而，当我们考虑用 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$ 表示的预言机辅助验证算法时，我们并不知道[3] $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$的完整描述。因此，我们无法执行合成过程。

一种简单的做法是对我们自己的预言机 $\mathcal{R}'$ 进行采样，并相对于$\mathsf{Ver}^{\mathcal{R}',|\mathsf{PSPACE}\rangle}$合成该量子态。然而，这并没有帮助。首先，无法保证 $\mathsf{Ver}^{\mathcal{R}',|\mathsf{PSPACE}\rangle}(\mathsf{pk}, s, \cdot)$ 以足够高的概率接受任意量子态。没有这一保证，合成过程便无法奏效。暂且假设确实存在某个见证态 σ'，能够被 $\mathsf{Ver}^{\mathcal{R}',|\mathsf{PSPACE}\rangle}(\mathsf{pk}, s, \cdot)$ 以足够高的概率接受。然而，无法保证 σ' 会被 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(\mathsf{pk}, s, \cdot)$ 接受的概率高于可忽略概率。

为了解决这些难题，我们首先关注一个简单情形： KeyGen 和 Mint 向 $\mathcal{R}$ 进行经典查询；之后再关注量子查询情形。

2.2.1 KeyGen 和 Mint：向 $\mathcal{R}$进行经典查询

消除 Comp。假设我们能够神奇地找到一个数据库 D，只需对 $\mathcal{R}$ 进行多项式次查询，使得 Ver 对 $\mathcal{R}$ 进行的所有查询 - 答案对都包含在 D 中。在这种情况下，存在一个 QPT 对手 $\mathcal{A}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$，它在给定 (pk, s, ρ_s) 后，能够找到两个量子态 (s, σ_s') 和 (s, σ_s'')，使得 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$ 接受这两个量子态。 $\mathcal{A}$ 的操作如下：首先找到数据库 D，并构造另一个电路 $\mathsf{Ver}^{D,|\mathsf{PSPACE}\rangle}$，使得 $\mathsf{Ver}^{D,|\mathsf{PSPACE}\rangle}$ 运行 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$，并且当 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$ 向 $\mathcal{R}$ 发起查询时，由 D回答该查询。然后， $\mathcal{A}$ 使用 PSPACE 合成两个量子态 (s, σ_s') 和 (s, σ_s'')，使得 $\mathsf{Ver}^{D,|\mathsf{PSPACE}\rangle}$ 接受这两个量子态。根据数据库 D 的定义，这两个量子态也会被 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$ 接受。

当然，指望仅通过向 $\mathcal{R}$ 发起多项式次查询，就找到一个与 Ver 所发查询完全一致的数据库 D，是不切实际的。相反，我们希望恢复一个足够良好的数据库 D。具体来说，我们的目标是恢复一个数据库 D，使其能够涵盖 KeyGen 和 Mint 所发的所有相关查询。

令 D_{KeyGen} 和 D_{Mint} 分别为 KeyGen 和 Mint 所进行的查询-答案对集合。若 Ver 所进行的查询属于 $D_{\mathsf{KeyGen}} \cup D_{\mathsf{Mint}}$，且该查询未记录在 D中，则称该查询为错误查询。如果 Ver 进行了错误查询，那么返回的答案很可能

[3]问题在于，我们没有 $\mathcal{R}$ 的描述。

与 $\mathcal{R}$ 不一致，因此无法保证 Ver 能够正常工作。我们希望 Ver 进行错误查询的概率有一个逆多项式上界。

一旦我们有了这样的数据库 D，通过类似的论证，我们可以得出结论：使用 $\mathsf{Ver}^{D,|\mathsf{PSPACE}\rangle}$ 合成的量子态也会被 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$接受。

但是，我们该如何恢复这个数据库 D? 为了解答这个问题，我们先考察一个简单情形，然后再处理一般情形。

与状态无关的数据库模拟。请注意， Ver 发出的查询可能取决于其输入量子态。暂且假设 Ver 发出的查询的分布与输入量子态无关。稍后我们再处理查询分布依赖于量子态的情形。

生成 D 的初步尝试，是依赖 Canetti、Kalai 和 Paneth [CKP15]提出的技术。他们在另一种背景下——证明随机预言机模型中混淆的不可能性时——展示了如何生成一个足以模拟求值算法所发查询的数据库。假设 (s,ρ_s) 是由 Mint生成的量子态。然后，通过查询$\mathcal{R}$，固定地运行多项式次 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,s,\rho_s)$，这些运行称为测试执行。在 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$的每次执行中，记录 Ver 发出的所有查询及其答案。将 Ver 所有执行中发出的查询取并集，并将其指定为数据库 D。在经典电路混淆的背景下， [CKP15]论证指出，除以逆多项式概率发生的情况外，求值算法发出的查询都可以由 D成功模拟。该论证通过证明求值算法发出错误查询的概率上界来建立。

类似的分析也可以应用于我们的情形，以论证 D 足以成功模拟。也就是说，我们可以论证，在执行完所有 Ver 后得到的量子态（可能与最初的量子态大不相同），可以使用 D成功模拟。然而，对我们的分析而言，关键在于证明 D_{Ver} （Ver期间产生的查询-答案对）与输入给 Ver的量子态无关 。

与状态相关的数据库模拟。 据我们所知，D_{Ver} 确实可能取决于输入量子态。在这种情况下，我们就无法再援引 [CKP15]的论证。从较高层面来看，原因在于，每次执行 Ver之后，货币态都可能发生变化，进而影响后续执行 Ver 中 D_{Ver} 的分布，导致在最终量子态上执行 Ver （该量子态可能不同于第一次执行 Ver时的输入量子态）无法使用数据库 D进行模拟。

相反，我们将依赖 [AK22], 提出的一种技术。他们在复制保护的背景下研究了一个类似问题。他们证明，通过随机化执行次数，可以论证：在所有测试执行之后，使用 D可以成功模拟在所获得的量子态上执行 Ver 的过程，除非发生逆多项式概率的情况。也就是说，假设初始量子态为 $(s,\rho_s^{(0)})$，并且在运行 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$、 t 次（其中 $t\xleftarrow{\$}\{0,1,\cdots,T\}$）之后，得到的量子态为 $(s,\rho_s^{(t)})$。令 D 为之前定义的量。则我们可以保证， Ver 接受 $(s,\rho_s^{(t)})$，除非发生逆多项式概率的情况，即使 $\mathcal{R}$ 是使用 D模拟的。这是因为，在 $T+1$ 次验证过程中遇到的恶意查询总数至多为 $|D_{\mathsf{KeyGen}}\cup D_{\mathsf{Mint}}|$。因此，至多只有 $\epsilon^{-1}|D_{\mathsf{KeyGen}}\cup D_{\mathsf{Mint}}|$ 个点 $t'\in\{0,1,\cdots,T\}$，使得在第 $(t'+1)^{th}$ 次验证过程中发出恶意查询的概率至少为 ϵ。所以，当 T足够大时，我们很有可能选出一个 t，使得发出恶意

查询的概率在下一次验证过程中（即从头开始计数的第 $(t+1)^{th}$ 次验证）很小，此时 D 便能很好地模拟 $\mathcal{R}$。

这表明可以对量子货币方案实施如下攻击。输入一个货币态 $(s,\ \rho_s)$，执行以下操作：

- 运行 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$，$t$ 次，这也称为测试执行。我们需要运行 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$的次数，即 t，按照 [AK22] 随机确定。令 D 为测试执行期间 Ver 向 $\mathcal{R}$ 发出的查询-答案对集合。将经过 t 次执行 Ver 后得到的量子态记为 $\rho_{(t)s}$。
- 令 $\mathsf{Ver}^{D,|\mathsf{PSPACE}\rangle}$ 为前面定义的验证电路。
- 利用对 PSPACE 的量子访问，按照 第 2.1 节合成两个量子态 $(s,\ \sigma'_s)$ 和 $(s,\ \sigma''_s)$，使这两个量子态都被 $\mathsf{Ver}^{D,|\mathsf{PSPACE}\rangle}$ 接受。
- 输出 $(s,\ \sigma'_s)$ 和 $(s,\ \sigma''_s)$。

根据见证合成方法，我们可以保证 (s,σ'_s) 和 (s,σ''_s) 均被 $\mathsf{Ver}^{D,|\mathsf{PSPACE}\rangle}$ 接受。然而，这不足以证明上述攻击有效。请记住，对手应该输出两个均被 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$ 接受的量子态。遗憾的是，我们无法保证 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$ 会接受这两个量子态。事实上，σ'_s 和 σ''_s 都可能与 $\rho_s^{(t)}$ 大相径庭。因此，上述攻击无效。

我们犯的每一个错误都是进步。让我们了解上述攻击为何无效。注意，只要 Ver 没有进行任何恶意查询（即在 $D_{\mathsf{KeyGen}} \cup D_{\mathsf{Mint}}$ 中但不属于 D 的查询），它就无法区分自己的查询究竟是由 D 还是 $\mathcal{R}$ 模拟的。然而，当在 (s,σ'_s) 或 (s,σ''_s) 上执行 Ver 时，我们便无法再给出 Ver 不进行任何恶意查询的概率上界。

我们对上述方法作如下修改：每当 Ver 进行错误查询时，就可以更新数据库 D，将这些错误查询及其正确答案（即使用$\mathcal{R}$生成的答案）一并加入其中。更新 D 后，我们可以利用 $\mathsf{Ver}^{D,|\mathsf{PSPACE}\rangle}$合成两个新的量子态。重复这一过程，直到合成出两个被 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$接受的量子态。

能否保证这一过程会停止？我们的关键洞见是，每当我们犯错并合成出未被 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$ 接受的量子态时，我们必然会在 $D_{\mathsf{KeyGen}} \cup D_{\mathsf{Mint}}$ 中学到一个 D中未包含的新查询。因此，每次犯错都会让我们有所进展。由于 $D_{\mathsf{KeyGen}} \cup D_{\mathsf{Mint}}$中的查询数量只有多项式数量，我们最终会合成出两个被 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$接受的量子态。

我们的攻击。经过这项修改，我们得到如下攻击。输入一个货币态（$s,\ \rho_s$）后，执行以下操作：

- 测试阶段：运行 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$，$t$ 次，也称为测试执行。根据 [AK22]，我们需要运行 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$ 的次数（即 t）是随机确定的。令 D 为测试执行期间 Ver 向 $\mathcal{R}$ 发出的查询-回答对集合。记 $\rho_{(t)s}$ 为执行 Ver 次后获得的量子态。

- 更新阶段：重复以下操作多项式次。令 $\mathsf{Ver}^{D,|\mathsf{PSPACE}\rangle}$ 为前文定义的验证电路。利用对 PSPACE的量子访问，按照 $(s,\ \sigma_s)$合成一个量子态，第 2.1 节，使该量子态被 $\mathsf{Ver}^{D,|\mathsf{PSPACE}\rangle}$ 接受。在该量子态上运行 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$，并将 Ver 向 $\mathcal{R}$ 发出的任何新查询加入 D。

- 令 $D_1,\ldots,D_{\mathrm{poly}}$ 表示在更新阶段每次执行后得到的数据库。

- 利用对 PSPACE 的量子访问，合成两个量子态 (s,σ'_s) 和 (s,σ''_s)，使得对于某个随机选取的 j，这两个量子态都能被 $\mathsf{Ver}^{D_j,|\mathsf{PSPACE}\rangle}$ 接受。

- 输出 (s,σ'_s) 和 (s,σ''_s)。

在技术部分，我们分析上述攻击，并证明它有效 s.

2.2.2 KeyGen 和 Mint：对 $\mathcal{R}$的量子查询

这里需要注意的关键点是前面提到的攻击者的形式。它只利用了这样一个事实： Ver 对 $\mathcal{R}$ 发起经典查询。当 KeyGen 和 Mint 对 $\mathcal{R}$ 发起量子查询，而 Ver 对 $\mathcal{R}$ 发起经典查询时，我们仍然可以运行该攻击者。接下来只需证明，即使 KeyGen 和 Mint 对 $\mathcal{R}$ 发起量子查询，同一个攻击者仍然有效。

将第 2.2.1 节中的直觉应用到 KeyGen 和Mint 对 $\mathcal{R}$ 进行量子查询的情形时，主要困难在于难以定义 D_{KeyGen} 和 D_{Mint}的类似概念。为了让大家对这一困难有所了解，我们先来看两种朴素的尝试。

第一种尝试是将 D_{KeyGen} 和 D_{Mint} 定义为在 KeyGen 和 Mint期间被查询（具有非零振幅）的那些查询-答案对。然而，这种尝试存在一个问题：按照这种定义， $D_{\mathsf{KeyGen}}\cup D_{\mathsf{Mint}}$ 可能包含指数多个元素。因此，即使我们每次都能取得进展，也就是说在 $D_{\mathsf{KeyGen}}\cup D_{\mathsf{Mint}}$中找回一些新元素，也无法保证更新阶段会在多项式时间内终止。

第二种尝试是仅纳入在 KeyGen 和 Mint期间被频繁查询的查询。具体而言，令 D_{KeyGen} 和 D_{Mint} 分别表示在 KeyGen 和 Mint期间以平方倒数多项式振幅被查询的查询-答案对。然而，采用这一看似合理的定义后，以下命题并不成立：只要 Ver^{D}的接受概率$^{|\mathsf{PSPACE}\rangle}$ 与 $\mathsf{Ver}^{\mathcal{R}}$的接受概率$^{|\mathsf{PSPACE}\rangle}$相差较大，我们就能在 $D_{\mathsf{KeyGen}}\cup D_{\mathsf{Mint}}-D$中找回一个查询。这是第 2.2.1 节中直觉背后的关键思想。让我们来理解一下，若采用 D_{KeyGen}和 D_{Mint}的这一种定义，为何该命题不成立。

考虑以下人为构造的反例（$\mathsf{KeyGen}^{|\mathcal{R}\rangle,|\mathsf{PSPACE}\rangle}$，$\mathsf{Mint}^{|\mathcal{R}\rangle,|\mathsf{PSPACE}\rangle}$，$\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$）。假设存在一个量子货币方案（$\mathsf{KeyGen}'^{|\mathcal{R}\rangle,|\mathsf{PSPACE}\rangle}$，$\mathsf{Mint}'^{|\mathcal{R}\rangle,|\mathsf{PSPACE}\rangle}$，$\mathsf{Ver}'^{\mathcal{R},|\mathsf{PSPACE}\rangle}$）。我们将该方案修改为（$\mathsf{KeyGen}^{|\mathcal{R}\rangle,|\mathsf{PSPACE}\rangle}$，$\mathsf{Mint}^{|\mathcal{R}\rangle,|\mathsf{PSPACE}\rangle}$，$\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$），如下所示：

- $\mathsf{KeyGen}^{|\mathcal{R}\rangle,|\mathsf{PSPACE}\rangle}(1^n)$：输出 $\mathsf{KeyGen}'^{|\mathcal{R}\rangle,|\mathsf{PSPACE}\rangle}$ 的密钥-公钥对。

- $\mathsf{Mint}^{|\mathcal{R}\rangle,|\mathsf{PSPACE}\rangle}(sk)$：以 sk 作为输入，对 $\mathcal{R}$ 进行量子查询，作用于量子态 $\frac{1}{\sqrt{2^n}}\sum_{s=0}^{2^n-1}|s\rangle|0\rangle$，得到量子态 $\frac{1}{\sqrt{2^n}}\sum_{s=0}^{2^n-1}|s\rangle|\mathcal{R}(s)\rangle$，随后测量第一个寄存器，得到值 s。它还运行 $\mathsf{Mint}'^{|\mathcal{R}\rangle,|\mathsf{PSPACE}\rangle}(sk)$，以获取序列号 s' 以及量子态 $\rho_{s'}$。它将 $(s,\ s')$ 作为序列号输出，并将 $(\mathcal{R}(s)\ \rho_{s'})$ 作为钞票输出。

- $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,\ ((s,\ s'),\ (h,\ \rho_{s'})))$: 将 pk 和一张声称的纸币 $((s,\ s'),\ (h,\ \rho_{s'}))$ 作为输入，向 $\mathcal{R}$ 发送关于输入 s 的经典查询，以获取 $R(s)$，并检查 $h=\mathcal{R}(s)$ 是否成立。它还会检查

$(s',\ \rho_{s'})$ 相对于 $\mathsf{Ver}'^{\mathcal{R},|\mathsf{PSPACE}\rangle}$ 是一个有效货币态。当且仅当两项检查均通过时，才接受。

在上述反例中，可能不存在以平方倒数多项式振幅提出的查询-答案对，因此 $D_{\mathsf{KeyGen}} \cup D_{\mathsf{Mint}} = \emptyset$。一开始 $D = \emptyset$，因为我们尚未开始记录查询-答案对。在这种情况下，Ver^{D}、$|\mathsf{PSPACE}\rangle$ $(pk,\ ((s,\ s'),\ (\mathcal{R}(s)\ \rho_{s'})))$ 的接受概率小于或等于 $\frac{1}{2^m}$，其中 m 是 $\mathcal{R}$ 在输入长度为 n 时的输出长度；而如果 $(\mathsf{KeyGen}'^{|\mathcal{R}\rangle},\ |\mathsf{PSPACE}\rangle,\ \mathsf{Mint}'^{|\mathcal{R}\rangle},\ |\mathsf{PSPACE}\rangle,\ \overline{\mathsf{Ver}}'^{\mathcal{R}},\ |\mathsf{PSPACE}\rangle)$ 是（完全）正确的，则 $\mathsf{Ver}^{\mathcal{R}}$、$|\mathsf{PSPACE}\rangle$ $(pk,\ ((s,\ s'),\ (\mathcal{R}(s)\ \rho_{s'})))$ 的接受概率为 1。然而，不可能在 $D_{\mathsf{KeyGen}} \cup D_{\mathsf{Mint}} - D$ 中恢复一个新的查询，因为它是空的。

纯化视图。我们的洞见是考虑一个称为纯化视图的替代世界。在这个替代世界中，我们以相干方式运行所有过程；更具体地说，我们考虑 $\mathcal{R}$的均匀叠加态，运行 Mint， KeyGen 甚至以相干方式运行攻击者（即不进行中间测量）。如果攻击者在这个替代世界中成功，那么在真实世界中也会成功；在真实世界中，$\mathcal{R}$ 以及 Ver 对 $\mathcal{R}$ 发出的查询都会被测量。接着，我们采用 Zhandry [Zha18] 的压缩预言机技术，以相干方式恢复在 KeyGen期间记录的查询-答案对数据库， Mint 并将其与在 Ver期间记录的数据库联系起来。通过复杂的分析，我们进而证明，当 KeyGen时， Mint 对 $\mathcal{R}$ 发出经典查询时所得出的许多洞见，都可以推广到量子查询设置中。

2.2.3 处理量子验证查询的挑战

很自然会想知道，我们是否也能类似地使用压缩预言机技术来处理 Ver 发出的量子查询。遗憾的是，这种做法存在固有的局限性。回想一下，在我们的攻击中，对手会将验证器的经典查询-答案对记录在数据库中，利用这些数据生成验证电路的经典描述（该电路不会向随机预言机发出任何查询），然后将电路描述提交给 a PSPACE 预言机，以合成一个货币态。如果验证器改为发出量子查询，那么一个自然的想法是使用 Zhandry 压缩预言机技术来记录这些量子查询。然而，要实现这一想法，存在两个概念上的挑战。

首先，在压缩预言机技术中，查询由预言机 自身记录在“数据库寄存器”中，而不是由密码系统中的对手记录。在我们的设定中，我们试图构造一个对手来记录这些查询，但它无法访问预言机的数据库寄存器。一般而言，对手若试图获取有关 Ver 查询位置的信息，可能会扰乱 Ver 算法的中间态；因此，如何利用 Ver 原有的保证值并不明确。换句话说，Zhandry 压缩预言机技术通常用于安全性分析，以证明对手破坏某个密码系统的能力受到限制。而在我们的情形中，我们希望在对手的攻击中使用某种量子记录技术。

其次，使用 PSPACE 预言机的自然方法是利用它来合成声称的钞票。然而，由于 PSPACE 预言机是一个经典函数（可以在叠加态中访问），因此它需要以多项式长度的经典字符串作为输入。在我们的方法中，对手提交一个验证电路的经典描述，并将查询/回答对硬编码在其中。另一方面，如果 Ver 进行量子查询，它可能会在

随机预言机 $\mathcal{R}$ 的叠加态中查询指数多个位置，而目前尚不清楚如何将有关这些查询的相关信息“压缩”成一个多项式大小的经典字符串，以供 PSPACE 预言机使用。

这表明，为了处理验证算法进行量子查询的情况，我们可能需要一种全新的方法来记录量子查询。

2.3 相关工作

量子货币。量子货币这一概念最早由 Wies- ner 在其论文中提出 [Wie83]。在同一项工作中，作者提出了一种私钥量子货币构造。Wies- ner 的构造已得到充分研究，其局限性 [Lut10] 和安全保证值 [MVW12]也已得到充分理解。其他私钥量子货币构造也得到了研究。Ji、Liu 和 Song [JLS18]基于伪随机量子态构造了私钥量子货币。Radian 和 Sattath [RS22] 基于带误差学习的量子困难性，构造了由经典银行管理的私钥量子货币。

关于公钥量子货币，Aaronson 和 Christiano [AC13] 在预言机模型中提出了一种公钥量子货币构造。Zhandry [Zha21]实例化了该预言机，并展示了如何基于后量子不可区分混淆（iO）的存在性构造公钥量子货币 [BGI+01]。最近，Shmueli [Shm22a] 展示了如何在假设后量子 iO 和带误差学习的量子困难性的条件下，实现由经典银行管理的公钥量子货币。基于较新假设的公钥量子货币构造 [FGH+12；KSS21；KLS22] 也已得到探索，尽管这些构造容易受到量子攻击 [Rob21；BDG22]。

量子密码学中的黑盒分离。迄今为止，现有的大多数量子密码学黑盒分离，都聚焦于将经典密码学原语的黑盒分离扩展到量子领域。Hosoyamada 和 Yamakawa [HY20] 将抗碰撞哈希函数与单向函数之间的黑盒分离 [Sim98] 扩展到了量子领域。Austrin、Chung、Chung、Fu、Lin 和 Mahmoody [ACC+22]证明了密钥协商与单向函数之间存在黑盒分离，其中诚实参与方可以进行量子计算，但只能进行经典通信。Cao 和 Xue [CX21] 将单向置换与单向函数之间的经典黑盒分离扩展到了量子领域。

3 预备知识

对于字符串 x，令 $|x|$ 表示其长度。对于任意正整数 n，令 $[n]$ 表示集合 $\{0,1,\cdots,n-1\}$。定义两个集合 X 和 Y 的对称差为仅包含在 X 和 Y 其中一个集合中的元素所构成的集合，即 $X\Delta Y=(X-Y)\cup(Y-X)$。

3.1 量子态、算法与预言机

一个寄存器 R 是一个有限维复希尔伯特空间。例如，如果 A，B，C 是寄存器，那么串接 ABC 表示相应希尔伯特空间的张量积。对于线性变换 L 和寄存器 R，我们有时写作 L_{R} 来表示 L 作用于 R；类似地，我们有时写作 ρ_{R} 来表示量子态 ρ 属于寄存器 R。我们用 $\mathrm{Tr}(\cdot)$ 表示迹，用 $\mathrm{Tr}_{\mathsf{R}}(\cdot)$ 表示对寄存器 R求偏迹。

对于纯态 $|\varphi\rangle$，我们写作 φ 表示其密度矩阵 $|\varphi\rangle\langle\varphi|$。令 I 表示单位矩阵。令 TD(ρ，σ) 表示两个密度矩阵 ρ，σ之间的迹距离。

对于以计算基表示的纯态 $|\varphi\rangle = \sum_i a_i |i\rangle$，我们用$|\varphi\rangle = \sum_i a_i |i\rangle$ 表示–$|\varphi\rangle$ 的共轭，其中 a_i 是复数 a_i 的复共轭。下面的观察说明了最大纠缠态在其他基下的形式。

引理 1. 对于两个维度相同的寄存器 A 和 B，其维度为 N，设 $(|i\rangle)_{i=1}^N$ 为计算基，$(|v_i\rangle)_{i=1}^N$ 为任意基。那么 $\frac{1}{\sqrt{N}} \sum_{i=1}^N |i\rangle_{\mathsf{A}} |i\rangle_{\mathsf{B}} = \frac{1}{\sqrt{N}} \sum_{i=1}^N |\overline{v_i}\rangle_{\mathsf{A}} |v_i\rangle_{\mathsf{B}}$.

证明。可以很容易证明，$(|v_i\rangle)_{i=1}^N$ 也构成一个基。设$|v_j\rangle = \sum_{i=1}^N a_{j,i} |i\rangle$。

$$
\begin{aligned}
\frac{1}{\sqrt{N}} \sum_{i=1}^N |i\rangle_{\mathsf{A}} |i\rangle_{\mathsf{B}} &= \sum_{j=1}^N |v_j\rangle\langle v_j|_{\mathsf{A}} \sum_{j'=1}^N |\overline{v_{j'}}\rangle\langle\overline{v_{j'}}|_{\mathsf{B}} \frac{1}{\sqrt{N}} \sum_{i=1}^N |i\rangle_{\mathsf{A}} |i\rangle_{\mathsf{B}} \\
&= \frac{1}{\sqrt{N}} \sum_{j=1}^N \sum_{j'=1}^N \sum_{i=1}^N \overline{a_{j,i}} a_{j',i} |v_j\rangle_{\mathsf{A}} |\overline{v_{j'}}\rangle_{\mathsf{B}} \\
&= \frac{1}{\sqrt{N}} \sum_{j=1}^N \sum_{j'=1}^N \langle v_j|v_{j'}\rangle |v_j\rangle_{\mathsf{A}} |\overline{v_{j'}}\rangle_{\mathsf{B}} \\
&= \frac{1}{\sqrt{N}} \sum_{j=1}^N |v_j\rangle_{\mathsf{A}} |\overline{v_j}\rangle_{\mathsf{B}}
\end{aligned}
$$

其中我们利用了 $\sum_{j=1}^N |v_j\rangle\langle v_j|_{\mathsf{A}}$ 和 $\sum_{j'=1}^N |v_{j'}\rangle\langle v_{j'}|_{\mathsf{B}}$ 是恒等算子的事实。 □

量子电路 我们将说明本文采用的量子电路模型。为方便起见，我们固定通用门集 $\{H, CNOT, T\}$ [NC00，第 4]（尽管我们的结果适用于由具有代数元素的门组成的任意通用门集）。量子电路可以包含来自上述通用门集的酉门，也可以包含以下非酉门：(a) 引入初始化为零态的新量子比特；(b) 将这些量子比特迹掉；或 (c) 在标准基下测量它们。我们称一个电路使用空间 s，如果在计算的任意时刻，涉及的量子比特总数至多为 s。电路的描述是一系列门（酉门或非酉门），并附有关于这些门作用于哪些量子比特的说明。

我们将量子电路序列 $C = (C_x)_{x\in\{0,1\}^*}$ 称为量子算法。如果存在一个多项式 p，使得 C_x 的规模至多为 $p(|x|)$，则称 C为多项式时间的。如果存在一个多项式 p，使得 C_x 使用的空间至多为 $p(|x|)$，则称C 为多项式空间的。

令 $C = (C_x)_{x\in\{0,1\}^*}$ 表示一个量子算法。给定一个字符串 $x \in \{0,1\}^*$ 和一个量子态 ρ，其量子比特数与电路 C_x的输入大小相匹配，我们用 $C(x, \rho)$ 表示电路 C_x 在输入 ρ上的输出。通常，输出将是一个混合态，因为电路 C_x 可以执行测量。

如果存在一台多项式时间图灵机，对于输入 x，能够输出 C_x 的描述，我们就称量子算法 $C = (C_x)_{x\in\{0,1\}^*}$ 是时间一致（或简称一致）的。类似地，如果存在一台多项式空间图灵机，对于输入 x，能够输出 C_x 的描述，我们就称 C 是空间一致的。

预言机算法预言机算法是指这样的量子算法：除了具有上述门之外，其电路还能够查询（可能处于叠加态的）函数O（称为预言机），该函数可以作用于多个量子比特。这本质上与标准量子查询模型相同 [NC00，第 6], 章，不同之处在于，电路可以执行测量、重置和偏迹等非酉算子。每次预言机调用都作为一个门计入电路的规模复杂度。预言机算法的时间一致性和空间一致性概念与非预言机算法相同：存在一台多项式时间／多项式空间图灵机——它不能访问预言机——能够输出这些电路的描述。

给定一个预言机 $\mathcal{O} = (\mathcal{O}_n)_{n\in\mathbb{N}}$，其中每个 $\mathcal{O}_n : \{0,1\}^n \to \{0,1\}$ 都是一个 n 位布尔函数，我们用 $C^{\mathcal{O}} = (C_x^{\mathcal{O}})^{x\in\{0,1\}^*}$ 表示一个预言机算法，其中每个电路 C_x 都可以查询任意一个函数 $(\mathcal{O}_n)_{n\in\mathbb{N}}$（前提是预言机作用的量子比特数不超过$C_x$的量子比特数）。

本文区分经典查询和量子查询。如果一个预言机算法 $C^{\mathcal{O}}$ 可以在叠加态下查询 $\mathcal{O}$，我们就称它进行量子查询；这类似于标准查询模型。如果 $C^{\mathcal{O}}$ 在每次调用预言机之前，都会在标准基下测量预言机的输入量子比特，我们就称它进行经典查询。在这种情况下，该算法查询的是一组输入的概率混合。为清楚起见，我们用 $C^{|\mathcal{O}\rangle}$ 表示进行量子查询的 C，用 $C^{\mathcal{O}}$ 表示进行经典查询的 C。

我们在全文中考虑的一种特定预言机是 PSPACE 预言机。具体来说，它是一列函数$(\mathsf{PSPACE}_n)_{n\in\mathbb{N}}$，其中对每个 n，函数 PSPACE_n 判定 n 位实例是否属于某个 PSPACE 完备语言（例如量化可满足性 [Pap94]）。

我们陈述以下观察。

引理 2。设 $C^{|\mathsf{PSPACE}\rangle} = (C_x^{|\mathsf{PSPACE}\rangle})^{x\in\{0,1\}^*}$ 表示一个以多项式时间运行的预言机算法（不一定是一致的），它对 PSPACE 进行量子查询，并且输出一个经典比特。那么，存在一个多项式空间算法 $D = (D_x)^{x\in\{0,1\}^*}$，使得对所有 $x \in \{0,1\}^*$，D_x 都是一个酉算子，并且 $C_x^{|\mathsf{PSPACE}\rangle}$的功能与以下过程完全相同：引入多项式数量的初始化为零态的量子比特，应用酉算子 D_x，然后在计算基下测量第一个量子比特以获得经典输出。此外，如果 C 是一致的，那么 D 也是空间一致的。

证明。这是因为对于多项式时间（预言机）算法，我们总可以只在开始时引入新的量子比特，并将测量和偏迹推迟到最后；而对 $C_x^{|\mathsf{PSPACE}\rangle}$中的 PSPACE 预言机的每次查询，都可以通过先引入若干初始化为零态的辅助量子比特，再应用一个酉算子来计算。该酉算子实现了针对 PSPACE 完全语言的经典多项式空间算法，并撤销所有中间结果。此外，多项式空间图灵机可以生成该酉算子的描述。

□

最后，我们将考虑由两个独立预言机 $\mathcal{R}$ 和$|\mathsf{PSPACE}\rangle$ 预言机构成的混合预言机 $\mathcal{O}$。在此模型中，预言机算法 $C^{\mathcal{O}}$ 向 $\mathcal{R}$发送经典查询，并向 PSPACE发送量子查询。我们稍稍滥用记法，将能够访问混合预言机的算法称为预言机算法。

态合成 我们定义如下“态复杂性类”。直观地说，它刻画了可由多项式空间量子算法合成的量子态集合。

定义 1 (statePSPACE)。 statePSPACE 是所有序列$(\rho_x)_{x\in S}$ 构成的类，其中存在某个集合 $S \subseteq\{0,1\}^*$（称为承诺）以及一个多项式 p，使得每个 ρ_x 都是作用于 $p(|x|)$ 个量子比特的密度矩阵；并且对于每个多项式 q，都存在一个空间一致的多项式空间量子算法 $C = (C_x)^{x\in\{0,1\}^*}$，使得对于所有 $x \in S$，电路 C_x 不接收任何输入，并输出一个密度矩阵 σ，满足$\mathrm{TD}(\sigma, \rho_x) \leq \exp(-q(|x|))$。

我们称量子态序列 $(\rho_x)_{x\in S}$ 为纯，如果其中每个 ρ_x 都是纯态 $|\psi_x\rangle\langle\psi_x|$；此时，我们通常将该序列记作 $(|\psi_x\rangle)_{x\in S}$。

以下定理表明，对于纯的 statePSPACE 序列，实际上存在一个多项式时间预言机算法，通过向 PSPACE 预言机进行量子查询来合成该状态序列。

定理 2 （[RY21]第 5 节）。 设 $(|\psi_x\rangle)_{x\in S}$ 是一个 statePSPACE 纯态族。那么存在一个多项式时间预言机算法 $A^{|\mathsf{PSPACE}\rangle}$ 使得在输入 $x \in S$时，该算法输出一个纯态，其与$\exp(-|x|)$的迹距离至多相差 $|\psi_x\rangle$。

3.2 公钥量子货币方案

定义 2 （预言机辅助公钥量子货币方案）。 一个 预言机辅助公钥量子货币方案 $\mathcal{S}^{\mathcal{O}}$ 由三个一致多项式时间预言机算法组成 $\left(\mathsf{KeyGen}^{\mathcal{O}}, \mathsf{Mint}^{\mathcal{O}}, \mathsf{Ver}^{\mathcal{O}}\right)$:

- $\mathsf{KeyGen}^{\mathcal{O}}(1^n)$：以一元表示的安全参数 n 为输入，并生成密钥-公钥对 (sk, pk)。
- $\mathsf{Mint}^{\mathcal{O}}(sk)$：以 sk 为输入，并铸造与序列号 s 关联的钞票 ρ_s 。
- $\mathsf{Ver}^{\mathcal{O}}(pk,(s, \rho_s))$：以 pk 和一张声称的纸币 (s, ρ_s) 为输入，并输出 $\rho'_s \otimes |\mathbf{x}\rangle\langle\mathbf{x}|$，其中 $\mathbf{x} \in \{\mathsf{Accept}, \mathsf{Reject}\}$。

为简单起见，当我们不关心 $\mathsf{Ver}^{\mathcal{O}}$ 中的输出 ρ'_s 时，有时将事件$\rho'_s \otimes |\mathsf{Accept}\rangle\langle\mathsf{Accept}| \leftarrow \mathsf{Ver}^{\mathcal{O}}(pk,(s, \rho_s))$ 接受记作 $\mathsf{Ver}^{\mathcal{O}}(pk,(s, \rho_s))$。

我们要求上述预言机辅助公钥量子货币方案满足正确性和安全性两项安全性质。

3.2.1 正确性

我们首先考虑以往研究采用的传统正确性定义。粗略地说，正确性意味着验证算法会接受铸造算法生成的货币态。随后，我们将考虑一种更强的概念，称为可重复使用性；它规定，对有效货币进行验证后，会输出另一个有效货币态（不一定与之前相同）。

定义 3 （正确性）。 预言机辅助公钥量子货币方案 $\left(\mathsf{KeyGen}^{\mathcal{O}}, \mathsf{Mint}^{\mathcal{O}}, \mathsf{Ver}^{\mathcal{O}}\right)$是 δ-正确的，如果对每个 $n \in \mathbb{N}$，以下条件都成立:

$$\Pr\left[\rho'_s \otimes |\mathsf{Accept}\rangle\langle\mathsf{Accept}| \leftarrow \mathsf{Ver}^{\mathcal{O}}(pk, (s, \rho_s)) \ : \ \begin{smallmatrix}(sk,pk)\leftarrow\mathsf{KeyGen}^{\mathcal{O}}(1^n)\\ (s,\rho_s)\leftarrow\mathsf{Mint}^{\mathcal{O}}(sk)\end{smallmatrix}\right] \geq \delta,$$

其中，概率也取自 $\mathcal{O}$ 的随机性。

我们省略 δ 当 $\delta \geq 1 - \mathsf{negl}(n)$。

可重复使用性。 在本文中，我们考虑满足更强可重复使用性概念的量子货币方案。

定义 4 (可重复使用性)。 预言机辅助公钥量子货币方案 $(\mathsf{KeyGen}^{\mathcal{O}}, \mathsf{Mint}^{\mathcal{O}}, \mathsf{Ver}^{\mathcal{O}})$是 δ-可重复使用的，如果对于每个 $n \in \mathbb{N}$ 以及每个多项式 $q(n)$，以下条件均成立：

$$\Pr\left[\rho_s^{(q(n))} \otimes |\mathsf{Accept}\rangle\langle\mathsf{Accept}| \leftarrow \mathsf{Ver}^{\mathcal{O}}(pk,(s,\rho_s^{(q(n)-1)})) \ : \ \begin{matrix}(sk,pk)\leftarrow\mathsf{KeyGen}^{\mathcal{O}}(1^n)\\ (s,\rho_s^{(0)})\leftarrow\mathsf{Mint}^{\mathcal{O}}(sk)\\ \forall i\in[q(n)-1],\ \rho_s^{(i+1)}\otimes|\mathbf{x}\rangle\langle\mathbf{x}|\leftarrow\mathsf{Ver}^{\mathcal{O}}(pk,(s,\rho_s^{(i)}))\end{matrix}\right] \geq \delta,$$

其中，概率还取自 $\mathcal{O}$ 的随机性。

当 $\delta \geq 1 - \mathsf{negl}(n)$ 时，我们省略 δ。

一般而言，可以调用温和测量引理 [Win99] 来证明正确性通常蕴含可重复使用性。然而，在我们的情境中并非如此。原因在于，验证算法每次对预言机进行经典查询时都会执行中间测量，而这些测量无法推迟到最后进行。

3.2.2 安全性

我们考虑以下安全性概念。基本上，它表示不存在高效对手能够从一张有效钞票生成两张序列号相同的声称的纸币。

定义 5 （安全性）。 预言机辅助公钥量子货币方案 $(\mathsf{KeyGen}^{\mathcal{O}}, \mathsf{Mint}^{\mathcal{O}}, \mathsf{Ver}^{\mathcal{O}})$是 δ安全的， 如果对每个 $n \in \mathbb{N}$ 以及每个一致多项式时间预言机算法 $\mathcal{A}^{\mathcal{O}}$，以下条件均成立：

$$\Pr\left[\mathsf{Ver}^{\mathcal{O}}(pk,(s,\phi_1)) \ accepts \ and \ \mathsf{Ver}^{\mathcal{O}}(pk,(s,\phi_2)) \ accepts \ : \ \begin{matrix}(sk,pk)\leftarrow\mathsf{KeyGen}^{\mathcal{O}}(1^n)\\ (s,\rho_s)\leftarrow\mathsf{Mint}^{\mathcal{O}}(sk)\\ \phi\leftarrow\mathcal{A}^{\mathcal{O}}(pk,(s,\rho_s))\end{matrix}\right] \leq \delta,$$

其中，概率也取自 $\mathcal{O}$的随机性。我们用 ϕ_i表示 ϕ 在 i^{th} 寄存器上的约化密度矩阵。

当 $\delta \leq \mathsf{negl}(n)$ 时，我们省略 δ。

3.3 Jordan 引理与交替投影

本节分析 交替投影算法。该工具用于估计验证算法在仅有一份量子态副本时对该量子态的接受概率，由 Marriott 和 Watrous [MW05]为见证保持的错误缩减而提出。本节主要遵循 [CMSZ22] 中的第 4.1 节。

对于两个二元结果投影测量 $M_1 = \{\Pi_1, I - \Pi_1\}$， $M_2 = \{\Pi_2, I - \Pi_2\}$，交替投影算法交替执行测量 M_1 和 M_2 （Π_1， Π_2 对应结果 1），直到满足停止条件。下面的引理可以通过将分布分解为若干个小子空间，帮助我们分析测量结果的分布。

引理 3 （Jordan 引理）。 对于任意两个投影算子 Π_1 和 Π_2，都存在希尔伯特空间的一个正交分解，将其分解为一维子空间和二维子空间 S_i，且这些子空间在 Π_1 和 Π_2 作用下均保持不变。此外，如果 S_i 是一个一维子空间，那么在 S_i 内， Π_1 和 Π_2 的作用分别为恒等算子或零秩投影算子。如果 S_i 是一个二维子空间，那么在 S_i 内， Π_1 和 Π_2 都是秩一投影算子。更具体地说，存在两个单位向量 $|v_i\rangle$ 和 $|w_i\rangle$，使得在 S_i 内， Π_1 投影到 $|v_i\rangle$ 上，而 Π_2 投影到 $|w_i\rangle$ 上。

令测量 $M_{\mathsf{Jor}} = \{P_i\}_i$，其中 P_i 是投影到上述定义的子空间 S_i 上的投影算子。那么 M_1 和 M_2 都与 M_{Jor} 对易。因此，如果在交替投影的任意位置插入 M_{Jor}，M_1 和 M_2 各自的结果分布都不会改变。我们可以先应用 M_{Jor}，然后交替应用 M_1、M_2，以此分析结果序列的分布。对于每个二维子空间 S_i，记 $p_i := |\langle v_i|w_i\rangle|^2$。（这可以看作衡量 S_i 内 Π_1 与 Π_2 之间角度的一个量。）

目前，先假设分解中只有二维子空间。存在一维子空间的一般情形本质上相同，可以用类似的方法处理。然后，$\Pi_1 = \sum_i |v_i\rangle\langle v_i|$，$\Pi_2 = \sum_i |w_i\rangle\langle w_i|$.[4]

现在陈述 一个最初由 [MW05]证明、并在许多后续研究中重述的结果（例如，[CMSZ22]）.

命题1. 如果初始量子态为 $|v_i\rangle$[5]，并交替应用 M_2、M_1 共 N 次，那么结果序列 b_1、b_2、b_3、$\cdots$、b_{2N} 将服从以下分布

1. 设定 $b_0 = 1$（因为对 $|v_i\rangle$ 应用 M_1 将得到结果*1*）。

2. 对于每个 j，我们以概率 p_i将 b_j 设为 b_{j-1}，否则设为 $1 - b_{j-1}$ $e.$

此外，每当我们测量 M_1 并得到结果 *1* 时，就会回到量子态 $|v_i\rangle$。

如果我们从 $|v_i\rangle$ 开始，那么结果序列中的比特翻转比例将是对 $1 - p_i$ 的良好估计。

3.4 压缩预言机技术

在本节中，我们 介绍 Zhandry 提出的压缩预言机技术的一些基础知识 [Zha1 8].

对于一个与随机预言机交互的量子查询算法 A，为简单起见，假设 A 仅以 n 位输入查询随机预言机，并获得 1 位输出。根据延迟测量原理，不失一般性，我们可以将 A 写成一系列酉算子的形式 U_0，U_f，U_1，U_f，$\cdots$，U_{k-1}，U_f，U_k，其中 U_i是一个酉算子，它将 A的第 $(i+1)^{th}$次查询准备为 R，而 U_f将 $|x\rangle|y\rangle$ 映射为 $|x\rangle|y \oplus f(x)\rangle$，其中 f是所有具有 n 位输入和 1 位输出的函数中选取的随机函数。

那么，可以通过以下纯化视图来分析 A 与随机预言机交互时的行为：

- 将寄存器 A 初始化为 A 的输入（并配备足够的辅助量子比特 $|0\rangle$），并将寄存器 F 初始化为从 $[2^n]$ 到 $\{0,1\}$ 的所有函数的真值表的均匀叠加（更具体地说，将 F初始化为 $\frac{1}{\sqrt{2^{2^n}}}\sum_{f\text{是一个 }2^n\text{位字符串}} |f\rangle$，其中 $|f\rangle = |f(0)\rangle|f(1)\rangle\cdots|f(2^n-1)\rangle$，且 $|f(i)\rangle$ 由一个量子比特组成）。

- 应用 U_0，U_F，U_1，U_F，$\cdots$，U_{k-1}，U_F，U_k其中 U_i作用于 A，而 U_F将 $|x\rangle|y\rangle|f\rangle_{\mathsf{F}}$ 映射到$|x\rangle|y \oplus f(x)\rangle|f\rangle_{\mathsf{F}}$。

[4]一般而言，对于每个满足 Π_1 作用为恒等算子的 一维子空间 S_i，我们可以将 $|v_i\rangle$ 设为张成 S_i的向量。令 A 为满足 Π_1 在 S_i内不是零秩投影算子的指标 i 的集合。那么 $\Pi_1 = \sum_{i\in A} |v_i\rangle\langle v_i|$。类似地，$\Pi_2 = \sum_{i\in B} |w_i\rangle\langle w_i|$，其中 B 和 $|w_i\rangle$ 以类似方式定义。[5]通常，对于每个 $|v_i\rangle(i \in A)$，情况也相同：若 Π_1 和 Π_2 在子空间 S_i 中均作用为恒等算子，则定义 $p_i = 1$；若 Π_1 作用为恒等算子，而 Π_2 在子空间 S_i中作用为零投影算子，则定义 $p_i = 0$。

事实上， A 的输出（混合）量子态（我们也将 f 的随机性纳入考虑）等于通过上述过程得到的量子态在输出寄存器上的约化密度矩阵，即U_i，U_F 与对 F进行的计算基测量可交换。更一般地，也可以用相同的方法分析可访问随机预言机的算法序列的输出（混合）量子态。

定义 6 （傅里叶基）。 $|\hat{0}\rangle := |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$。 $|\hat{1}\rangle := |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$。

容易验证， $\{|\hat{0}\rangle, |\hat{1}\rangle\}$ 是一个基，因为它只是将厄米矩阵 H 作用于 $|0\rangle, |1\rangle$所得的结果。我们将这个基称为傅里叶基。

以下事实简单易证，但在压缩预言机技术中至关重要。粗略地说，它表示，如果我们在傅里叶基中看到 $CNOT$，其控制位和目标位会互换。

事实 1. 对所有 $y, y' \in \{0,1\}$，由 $|y\rangle|y'\rangle \to |y \oplus y'\rangle|y'\rangle$ 定义的算子与对所有 $y, y' \in \{0,1\}$，由 $|\widehat{y}\rangle|y'\rangle \to |\hat{y}\rangle|y' \oplus y\rangle$ 定义的$\widehat{}$算子相同。

根据事实 1，当我们在傅里叶基下观察最后两个寄存器时， U_F 变为

$$|x\rangle|\widehat{y}\rangle|\widehat{y_0}\rangle|\widehat{y_1}\rangle\cdots|\widehat{y_{2^n-1}}\rangle \to |x\rangle|\widehat{y}\rangle|\widehat{y_0}\rangle|\widehat{y_1}\rangle\cdots|\widehat{y_{x-1}}\rangle|\widehat{y_x \oplus y}\rangle|\widehat{y_{x+1}}\rangle\cdots|\widehat{y_{2^n-1}}\rangle.$$

最初， F 是 $|\hat{0}\rangle|\hat{0}\rangle\cdots|\hat{0}\rangle$，并且每次调用 U_F时，如果我们在傅里叶基下观察最后两个寄存器，它只会改变一个位置。因此，调用 U_F k 次后，量子态可以写成

$$\sum_{\substack{a,y_0,y_1,\cdots,y_{2^n-1} \\ \text{such that there are at most } k \text{ non-zero} \\ \text{in } y_0, y_1, \cdots, y_{2^n-1}}} \alpha_{a,y_0,y_1,\cdots,y_{2^n-1}} |a\rangle_{\mathsf{A}} |\widehat{y_0}\rangle|\widehat{y_1}\rangle\cdots|\widehat{y_{2^n-1}}\rangle.$$

$\widehat{}$我们可以将那些非 0 项记录到数据库中。更具体地说，存在一个酉算子，可以将那些 $|\widehat{y_0}\rangle|\widehat{y_1}\rangle\cdots|\widehat{y_{2^n-1}}\rangle$（或许连同一些辅助量子比特）映射到数据库 $|x_1\rangle|\widehat{y_{x_1}}\rangle\cdots|x_l\rangle|\widehat{y_{x_l}}\rangle$（或许连同一些未使用空间），其中 $x_1 < x_2 < \cdots < x_l$，$\widehat{y_{x_i}} \neq \hat{0}$ 且 $l \leq k$。也就是说，存在一个能够压缩预言机的酉算子。此外，该酉算子的逆可以将数据库解压回预言机。

切尔诺夫界 最后，我们在此给出切尔诺夫界的一个变体，后文将会用到。

定理 3 （切尔诺夫界）。 假设 X_1、X_2、$\cdots$、X_n 是取值于 $\{0,1\}$ 的独立随机变量，且每个变量以概率 p 取值 $X_i = 1$。令 $\mu = pn$。那么对于任意 $\delta > 0$，

$$\mathsf{Pr}\left[\sum_{i=1}^{n} X_i \geq (1+\delta)\mu\right] \leq e^{-\delta^2\mu/(2+\delta)},$$

$$\mathsf{Pr}\left[\sum_{i=1}^{n} X_i \leq (1-\delta)\mu\right] \leq e^{-\delta^2\mu/2}.$$

4 在量子多项式空间中合成见证态

在经典情形下，很容易看出，给定一个（经典）验证器电路 V（它可能会对 PSPACE 进行预言机查询），可以在多项式空间内找到一个能被V接受的见证字符串 y：只需对所有字符串进行穷举搜索，并检查 V^{PSPACE} 是否接受x即可。

在本节中，我们证明其量子对应版本：此时，验证器电路是量子的，并且可以向 PSPACE 预言机发起量子查询。我们证明，给定此类验证器电路的描述后，借助量子 $|\mathsf{PSPACE}\rangle$ 预言机，我们可以高效地合成一个见证态 ρ，使其被 V 以大于所需保证值的概率接受（前提是存在一个接受概率大于阈值的见证态）。形式化地说：

定理 4。设 a（称为保证值）、b（称为阈值）是满足以下条件的函数：对于每个 n，其中 p 是一个多项式，均有 $b(n)-a(n)\geq \frac{1}{p(n)}$。设 $\mathsf{V}^{|\mathsf{PSPACE}\rangle}$ 表示一个一致预言机算法。那么存在一个一致预言机算法 Syn（称为合成器），使得对于每个 $x\in S$，

$$\Pr\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}(x,\mathsf{Syn}^{|\mathsf{PSPACE}\rangle}(x))\ accepts\right]\geq a(|x|)$$

其中 $S:=\left\{x:\text{最大值}_\rho \Pr\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}(x,\ \rho)\text{接受}\right]\geq b(|x|)\right\}$。

该定理可由定理 2 和以下引理直接推出。

引理 4. 设 a,b 是函数，使得对于每个 n，都有 $b(n)-a(n)\geq\frac{1}{p(n)}$，其中 p 是一个多项式。令 $\mathsf{V}^{|\mathsf{PSPACE}\rangle}$ 表示一个一致预言机算法，并令 S 表示定理 4 中对应的集合。则存在一个纯态族 $(|\psi_x\rangle)_{x\in S}$，其中每个量子态 $|\psi_x\rangle$ 都是定义在两个寄存器（标记为 M 和 E）上的双部分态，使得对于每个 $x\in S$，

$$\Pr\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}(x,\rho_{\mathsf{M}})\ accepts\right]\geq a(|x|)$$

其中 ρ_{M} 是 $|\psi_x\rangle$ 在寄存器 M上的约化密度矩阵，即 $\rho_{\mathsf{M}}=\text{迹}_{\mathsf{E}}(\psi_x)$。

定理 4 的证明。令 $a'(n)=a(n)+\exp(-n)$，且 $b'(n)=b(n)$，其中 $a(n)$ $b(n)$由定理 4 的条件给出。将函数 $a'(n)$ $b'(n)$代入引理 4，可得一个量子态序列 $(|\psi_x\rangle)_{x\in S}$，使得对于每个 $x\in S$，

$$\Pr\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}(x,\rho_{\mathsf{M}})\ \text{accepts}\right]\geq a(|x|)+\exp(-|x|)$$

其中 $S:=\left\{x:\max_\rho \Pr\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}(x,\ \rho)\text{接受}\right]\geq b(|x|)\right\}$。

定理 2 意味着存在一个多项式时间预言机算法 $A^{|\mathsf{PSPACE}\rangle}$，它在输入$x\in S$时，输出一个纯态 $|\varphi_x\rangle$，该纯态与$\exp(-|x|)$-接近 $|\psi_x\rangle$。这意味着 $A^{|\mathsf{PSPACE}\rangle}(x)$ 在寄存器 M上的约化密度矩阵（记作 σ_{M}）也与$\exp(-|x|)$-接近$\rho_{\mathsf{M}}=\mathrm{Tr}_{\mathsf{E}}(\psi_x)$。这是因为丢弃子系统时，迹距离不会增加。因此，对每个 $x\in S$，都有

$$\Pr\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}(x,\sigma_{\mathsf{M}})\ \text{accepts}\right]\geq a(|x|)$$

否则，$\mathsf{V}^{|\mathsf{PSPACE}\rangle}$ 就能够以超过 $\exp(-|x|)$ 的偏置区分 ρ_{M} 和 σ_{M}。

合成器 Syn 的工作方式如下：在输入 $x \in S$ 上，它运行预言机算法 $A^{|\mathsf{PSPACE}\rangle}(x)$，得到一个纯态 $|\varphi_x\rangle$，然后对 E 寄存器取迹，并将 M 寄存器上的剩余量子态作为输出返回。

□

第 4 节 余下的内容将用于证明 引理 4。我们将使用 [MW05] 中的技术和结果（第 3.3 节 也完整介绍了这些内容）。在第 4.1 节 中，我们将介绍状态族的描述，以及用于生成该状态族（近似态）的电路族的描述。在第 4.2 节 中，我们将证明该状态族满足这些要求。

4.1 状态族与电路族的描述

本节将以形式化的方式实现我们在第 2.1 节 中的想法。回顾一下，我们在第 2.1 节 中的算法会反复执行以下操作（我们称之为一次试验）：从一个最大纠缠态开始，使用 MW 技术相干地估计接受概率；如果估计的接受概率较高，就输出剩余的量子态。粗略地说，我们旨在生成的目标态，就是成功试验后剩余的量子态（当估计的接受概率较高时，试验即为成功）。展望后文，为了证明引理 4，我们只需证明两点。首先，我们的算法确实输出目标态的良好近似，因此目标态构成一个 statePSPACE 族；其次，我们的目标态确实会以高概率被接受。

现在，我们先对状态族进行正式描述。

状态族 设 $\mathsf{V}^{|\mathsf{PSPACE}\rangle}$ 为 ^ ^引理 4 条件中给出的一致预言机算法。根据 引理 2，存在一个空间一致多项式空间算法 $\mathsf{V} = (\mathsf{V}_x)^{x\in\{0,1\}^*}$，使得 $\widehat{\mathsf{V}}_x$ 是酉算子，并且 $\mathsf{V}_x^{|\mathsf{PSPACE}\rangle}$的功能与以下操作完全相同：在 $k(|x|)$ ^个新辅助量子比特上 $|0\rangle$施加酉算子 V_x，然后在计算基下测量第一个量子比特，其中 k 是一个多项式。设 $m(|x|)$ 为 V_x 接受的输入量子比特数，这也是一个多项式。

固定 $x \in S$，$n = |x|$。在上下文明确时，我们有时省略下标 x。为方便起见，从现在起，我们分别将 $m(n)$ $k(n)$ 写作 m，k。

令 M 表示包含 m 输入量子比特的寄存器。令 K 表示包含k 辅助量子比特的寄存器。令 Ans 表示第一个量子比特（即将在计算基下测量，以判断 $\mathsf{V}_x^{|\mathsf{PSPACE}\rangle}$是接受还是拒绝；测量结果为 1 表示接受，结果为 0 表示拒绝）。令 Aux 表示一个包含另外 m 个全新量子比特的寄存器。

这里我们定义作用于 MK的两个二元结果投影测量。定义 $P^1 := |0^k\rangle\langle 0^k|_{\mathsf{K}}, P^0 := I_{\mathsf{MK}} - P^1$ 和 $P := \{P^0, P^1\}$。直观地说，P^1 对应于“有效输入子空间”（即辅助量子比特已正确初始化）。定义 $Q^1 := \widehat{\mathsf{V}}_x^\dagger |1\rangle\langle 1|_{\mathsf{Ans}} \widehat{\mathsf{V}}_x, Q^0 := \widehat{\mathsf{V}}_x^\dagger |0\rangle\langle 0|_{\mathsf{Ans}} \widehat{\mathsf{V}}_x$ 和$Q := \{Q^0, Q^1\}$。直观地说，Q^1 对应于这样一种量子态：对其施加 $\widehat{\mathsf{V}}_x$ 后，再在计算基下测量 Ans，该量子态将被接受。因此，只要寄存器 K 已正确初始化，Q 就会检查 $\mathsf{V}_x^{|\mathsf{PSPACE}\rangle}$是否接受。以下简单观察在 [MW05]中已有隐含说明。

观察 1。 $\mathsf{V}^{|\mathsf{PSPACE}\rangle}(x, \cdot)$ 的最大接受概率恰好是 $P^1Q^1P^1$ 的最大特征值。

证明。 首先，我们证明 $P^1Q^1P^1$ 的最大特征值不超过 $\mathsf{V}^{|\mathsf{PSPACE}\rangle}(x,\cdot)$ 的最大接受概率。

对于寄存器 MK 上的任意纯态 $|\phi\rangle$，令 $\rho_{\mathsf{M}}=\frac{1}{\|P^1|\phi\rangle\|^2}\mathrm{Tr}_{\mathsf{K}}(P^1|\phi\rangle\langle\phi|P^1)$。那么

$$\begin{aligned}\langle\phi|P^1Q^1P^1|\phi\rangle=&\mathrm{Tr}(Q^1P^1|\phi\rangle\langle\phi|P^1)=\|P^1|\phi\rangle\|^2\mathrm{Tr}(Q^1\frac{1}{\|P^1|\phi\rangle\|^2}P^1|\phi\rangle\langle\phi|P^1)\\=&\|P^1|\phi\rangle\|^2\mathrm{Tr}(Q^1(\rho_{\mathsf{M}}\otimes|0^k\rangle\langle 0^k|_{\mathsf{K}}))\\=&\Pr\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}(x,\rho_{\mathsf{M}})\text{ accepts}\right]\\\le&\max_{\rho}\Pr\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}(x,\rho)\text{ accepts}\right]\end{aligned}$$

其次，我们证明，$P^1Q^1P^1$ 的最大特征值下界为 $\mathsf{V}^{|\mathsf{PSPACE}\rangle}(x,\cdot)$ 的最大接受概率。根据一个简单的凸性论证，不失一般性，我们可以假设 $\mathsf{V}^{|\mathsf{PSPACE}\rangle}(x,\cdot)$ 的接受概率在纯态 $|\phi_0\rangle$ 上达到最大值。令 $|\phi\rangle=|\phi_0\rangle|0^k\rangle$。那么

$$\langle\phi|P^1Q^1P^1|\phi\rangle=\Pr\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}(x,\phi_0)\text{ accepts}\right]=\max_{\rho}\Pr\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}(x,\rho)\text{ accepts}\right]$$

因此，这一观察成立。

我们首先定义一个子程序 Trial，其中 $N:=\max\left(\binom{3a+b}{b-a}^2(m+2-\log(b-a)),\ 16b(b-a)^2\right)$ 在 n 上是多项式（回顾一下，$b(n)-a(n)\ge\frac{1}{p(n)}$，其中 p 是一个多项式）。

1：初始化寄存器 MAux，使其为 $\frac{1}{\sqrt{2^m}}\sum_{i\in\{0,1\}^m}|i\rangle_{\mathsf{M}}|i\rangle_{\mathsf{Aux}}$ 2：初始化寄存器 K，使其为 $|0^k\rangle_{\mathsf{K}}$ 3：引入一个新的 $2N+1$ 量子比特寄存器 $\mathsf{Y}:=\mathsf{Y}_0\cdots\mathsf{Y}_{2N}$，初始化为 $|1\rangle|0^{2N}\rangle$ 4：引入一个新的寄存器 Cnt，并将其初始化为 $|0\rangle$ 5：对于 $i=1,2,\cdots,N$，执行 6：使用 Q 对 MK 进行相干测量，并将测量结果存储在 Y_{2i-1} 中 7：使用 P 对 MK 进行相干测量，并将测量结果存储在 Y_{2i} 中 8：结束循环 9：计算 $y_j=y_{j-1}$ 处于叠加态的次数，并将结果存储在 Cnt 中 10：执行投影测量 $\mathsf{Test}:=\{\mathsf{Yes}:=\sum_{j\ge N(a+b)}|j\rangle\langle j|_{\mathsf{Cnt}}\otimes|1\rangle\langle 1|_{\mathsf{Y}_{2N}},\mathsf{No}:=I-\mathsf{Yes}\}$ 定义 $\mathsf{E}:=\mathsf{K}\otimes\mathsf{Aux}\otimes\mathsf{Y}\otimes\mathsf{Cnt}$（即除 M 外的所有寄存器）。

定义 7（状态族 $(|\psi_x\rangle)_{x\in S}$）。令 S 为 引理 4 中定义的集合。当 $x\in S$ 时，令 $|\psi_x\rangle$ 表示成功实现 Trial 后寄存器 $\mathsf{M}\otimes\mathsf{E}$ 中的量子态（即 Test 的结果为 Yes）。当 x 可由上下文明确得知时，我们也将其记作 $|\psi\rangle$。

注意，在 Trial 中，我们先在寄存器 $\mathsf{M}\otimes\mathsf{E}$ 中初始化一个纯态（第 1 行至第 4 行），然后对其施加一个酉算子（第 5 行至第 9 行），因为所有测量都是相干地进行的，最后执行一次投影测量（第 10 行）。因此，上述定义确实给出了一个态族 $(|\psi_x\rangle)_{x\in S}$，其中每个 $|\psi_x\rangle$ 都是 $l(n)$ 个量子比特上的纯态，而 l 是一个多项式。

电路族 现在，让我们构造一个电路族（或算法），以高效生成状态族 $(|\psi_x\rangle)_{x\in S}$ 的近似。对于任意多项式 q 和近似因子 $\exp(-q(n))$，电路 C_x 的运作方式如下，其中 $T := 2^{m+2}q(n)$ 在 n 中是指数级的。

1: **对于** $t = 1, \cdots, T$ **执行** 2: 运行 Trial
3: **若**成功**则** 4: **返回**寄存器 ME 中的量子态 5:
结束若 6: **结束循环** 7: **返回**一个具有 $l(n)$ 个量子比特的任意量子态

4.2 引理 4 的证明

本节通过应用第 3.3 节中的已知结果，证明纯态族$(|\psi_x\rangle)_{x\in S}$ 满足引理 4中的要求。

固定 $x \in S$。我们将 Π_1 与 P^1、Π_2 与 Q^1、M_1 与 P 以及 M_2 与 Q对应，并采用第 3.3 节中的记号。于是 $P^1Q^1P^1 = \sum_i |v_i\rangle\langle v_i| \sum_i |w_i\rangle\langle w_i| \sum_i |v_i\rangle\langle v_i| = \sum_i p_i |v_i\rangle\langle v_i|$。根据$x \in S$ 以及观察 1，我们可以假设 $p_1 = \max_i p_i \geq b$。 [6]

首先，我们证明量子态族$(|\psi_x\rangle)_{x\in S}$ 满足引理 4中的第一个要求。也就是说，它是一个 statePSPACE 族，可以由该电路族近似生成。注意，C_x 会在任意一个指数多个 $|\psi_x\rangle$ 成功时输出 Trial。因此，我们先分析单个 Trial的成功概率。

引理 5. $\Pr[\text{Trial 成功}] \geq \frac{1}{2^{m+2}}$. ——

证明。如果在每个量子比特被写入结果后对 Y 中的每个量子比特测量一次，Trial 的成功概率不会改变，因为对 Y 进行的计算基测量与 Trial 第 9 行和第 10 行中的操作可交换。因此，我们可以将其视为直接用 P 和 Q 交替测量 MK，得到经典结果序列 $y := y_1y_2\cdots y_{2N}$；若 $y_{2N} = 1$ 且 $y_j = y_{j-1}(1 \leq j \leq 2N)$出现的次数至少为 $N(a + b)$（其中 $y_0 = 1$），则返回 Yes。这是一种交替投影算法。为简单起见，我们将与结果 Yes对应的 y 集合记作 Good 。现在来分析 $y \in$ Good的概率。

一个重要的观察是，初始量子态也可以写成以下形式：$|v_i\rangle$。由于$P^1 = \sum_i |v_i\rangle\langle v_i|$，$|v_i\rangle$ 构成希尔伯特空间 $\mathcal{H}_{\mathsf{M}} \otimes |0^k\rangle\langle 0^k|_{\mathsf{K}}$ 的一组基。令 $|u_i\rangle$ 为 M 上 $|v_i\rangle$的截断，即 $|v_i\rangle_{\mathsf{MK}} = |u_i\rangle_{\mathsf{M}} |0^k\rangle_{\mathsf{K}}$。那么 $|u_i\rangle$ 构成 $\mathcal{H}_{\mathsf{M}}$ 的一组基。因此，根据 引理 1，量子态

$$\frac{1}{\sqrt{2^m}} \sum_{i=0}^{2^m-1} |i\rangle_{\mathsf{M}} |i\rangle_{\mathsf{Aux}} = \frac{1}{\sqrt{2^m}} \sum_i |u_i\rangle_{\mathsf{M}} |\overline{u_i}\rangle_{\mathsf{Aux}}$$

因此，$\frac{1}{\sqrt{2^m}} \sum_{i=0}^{2^m-1} |i\rangle_{\mathsf{M}} |0^k\rangle_{\mathsf{K}} |i\rangle_{\mathsf{Aux}} = \frac{1}{\sqrt{2^m}} \sum_i |v_i\rangle_{\mathsf{MK}} |u_i\rangle_{\mathsf{Aux}}$. [7]

注意，在进行交替投影之前，我们可以对寄存器 MK 应用 M_{Jor}，而不会改变 y 的分布。对上述量子态应用 M_{Jor} 后，测量后状态将

[6] 一般而言 $P^1Q^1P^1 = \sum_{i\in A\cap B} p_i |v_i\rangle\langle v_i|$。我们可以假设 $p_1 = \max_{i\in A\cap B} p_i \geq b$。 [7] 在一般情形下，求和是对 $i \in A$ 进行的。也就是说，$\frac{1}{\sqrt{2^m}} \sum_{i=0}^{2^m-1} |i\rangle_{\mathsf{M}} |0^k\rangle_{\mathsf{K}} |i\rangle_{\mathsf{Aux}} = \frac{1}{\sqrt{2^m}} \sum_{i\in A} |v_i\rangle_{\mathsf{MK}} |u_i\rangle_{\mathsf{Aux}}$. 因此，对于每个 $i \in A$，如果我们应用 M_{Jor}，就会以概率 $\frac{1}{2^m}$ 处于子空间 S_i 中。 —

为 $|v_i\rangle_{\mathsf{MK}}|\overline{u_i}\rangle_{\mathsf{Aux}}$，其概率为 $\frac{1}{2^m}$。根据 命题1，当我们从 $|v_i\rangle_{\mathsf{MK}}|u_i\rangle_{\mathsf{Aux}}$ 开始时，$y_j = y_{j-1}$ 对每个 j，其概率为 p_i，且彼此独立。

特别地，我们以概率 $\frac{1}{2^m}$ 从 $|v_1\rangle_{\mathsf{MK}}|u_1\rangle_{\mathsf{Aux}}$ 开始。当我们从 $|v_1\rangle_{\mathsf{MK}}|u_1\rangle_{\mathsf{Aux}}$ 开始时，对每个 j，独立地以概率 p_1 取 $y_j = y_{j-1}$。这可以看作以偏置 $p_1 \geq b$ 独立抛硬币 $2N$ 次。而且，若正面数（记为 cnt）大于或等于 $N(a+b)$，则 $y \in \mathsf{Good}$。

根据切尔诺夫界，

$$\Pr[cnt < (a+b)N] \leq \exp(-Np_1(1-\frac{a+b}{2p_1})^2) \leq \exp(-N\frac{(b-a)^2}{4b}) \leq \frac{1}{4}$$

$$\begin{aligned}\Pr[cnt \text{ is an odd}] =& \sum_{j=0}^{N-1}\binom{2N}{2j+1}p_1^{2j+1}(1-p_1)^{2N-2j-1}\\ =&\frac{1}{2}((p_1+1-p_1)^{2N}-(p_1-(1-p_1))^{2N})\\ \leq&\frac{1}{2}\end{aligned}$$

因此，根据并集界，当 M_{Jor} 之后的测量后状态为 $|v_1\rangle_{\mathsf{MK}}|u_1\rangle_{\mathsf{Aux}}$ 时，$y \in \mathsf{Good}$ 的概率至少为 $\frac{1}{4}$。

因此 $\Pr[\text{Trial 成功}] = \Pr[y \in \mathsf{Good}] \geq \frac{1}{2^m}\frac{1}{4} = \frac{1}{2^{m+2}}$.

□

断言1. $(|\psi_x\rangle)_{x\in S}$ 是一个 $\mathsf{statePSPACE}[S]$ 族。

证明。我们只需 证明我们的构造 C_x 满足定义 1 .

根据构造，C_x 可由多项式空间图灵机在输入 x时生成，并且 C_x 在任何时刻使用的空间至多为多项式。因此，$C = (C_x)_{x\in\{0,1\}^*}$ 是一个空间一致的多项式空间量子算法。显然，根据构造，C_x 不接受任何输入。剩下唯一要证明的是，当 $x \in S$时，C_x 会输出 $|\psi_x\rangle$ 的良好近似。

固定 $x \in S$。每当 Trial成功实现时，C_x 将输出 $|\psi_x\rangle$。此外，根据引理 5，回顾一下，$T = 2^{m+2}q(n)$，

$$\Pr[T\text{次独立重复都失败}] = (1-\Pr[\text{Trial 成功}])^T \leq (1-\frac{1}{2^{m+2}})^T \leq e^{-q(n)}$$

也就是说，除去概率为 $e^{-q(n)}$的情况，C_x 会输出 $|\psi_x\rangle$。

因此，C_x 输出的量子态与 $|\psi_x\rangle$ 之间的迹距离为 $e^{-q(n)}$。

□

引理 4中的第二个$(|\psi_x\rangle)_{x\in S}$ 需要满足的条件是，$|\psi_x\rangle$ 的约化密度矩阵将以 $\mathsf{V}^{|\mathsf{PSPACE}\rangle}(x,\cdot)$ 较高概率接受。直观而言，这是合理的，因为实际接受概率与估计接受概率不应相差太远。现在让我们正式证明这一点。

断言 2. 对于每个 $x \in S$, $\Pr\left[\mathsf{V}_x^{|\mathsf{PSPACE}\rangle}(\rho_{\mathsf{M}}) \text{ 均接受}\right] \geq a$，其中 ρ_{M} 是 ψ_x 在寄存器 M上的约化密度矩阵。

证明。 固定 $x \in S$。当上下文明确时，我们将省略下标。

与我们在 引理 5 中所做的类似，如果我们改为直接测量 ψ_x （即 Trial），那么这个概率不会改变，因为在生成 Y 时，我们只关心 M中的部分。令 ψ'_x 为改为直接测量时，从 Trial 的成功实现中获得的量子态。那么， ψ_x 和 ψ'_x 在寄存器 M上的约化密度矩阵相同。

注意， $\mathsf{V}_x^{|\mathsf{PSPACE}\rangle}(\rho_{\mathsf{M}})$只是在 $\rho_{\mathsf{M}} \otimes |0^k\rangle\langle 0^k|_{\mathsf{K}}$ 上应用 $\widehat{\mathsf{V}}_x$ ，然后在计算基下测量第一个量子比特；等价地，它就是用 Q测量 $\rho_{\mathsf{M}} \otimes |0^k\rangle\langle 0^k|_{\mathsf{K}}$ 。根据定义， ψ'_x 在寄存器 K 上为 $|0^k\rangle$ （因为测量结果 y_{2N} 应为 1）。因此， ψ'_x 在寄存器 MK 上的约化密度矩阵恰好是 $\rho_{\mathsf{M}} \otimes |0^k\rangle\langle 0^k|_{\mathsf{K}}$。

考虑以下交替投影算法：

我们从 $\frac{1}{\sqrt{2^m}} \sum_{i \in \{0,1\}^m} |i\rangle_{\mathsf{M}} |0^k\rangle_{\mathsf{K}} |i\rangle_{\mathsf{Aux}}$ 开始，将 Q、P交替作用于量子态，共作用 N次，得到经典结果序列 $y := y_1 y_2 \cdots y_{2N}$如果 y满足要求（更准确地说， $y \in \mathsf{Good}$ 其中 Good 定义见 引理 5），我们还将额外作用 Q ，得到结果 z ，并在 $z = 1$时接受。

在上述算法中，如果 $y \in \mathsf{Good}$，剩余的（混合）量子态恰好为 ψ'_x，其在 MK 上的约化密度矩阵为 $\rho_{\mathsf{M}} \otimes |0^k\rangle\langle 0^k|_{\mathsf{K}}$。回想一下， $\mathsf{V}_x^{|\mathsf{PSPACE}\rangle}(\rho_{\mathsf{M}})$ 就是用 Q 测量 $\rho_{\mathsf{M}} \otimes |0^k\rangle\langle 0^k|_{\mathsf{K}}$。因此，

$$\Pr\left[\mathsf{V}_x^{|\mathsf{PSPACE}\rangle}(\rho_{\mathsf{M}}) \text{ accepts}\right] = \Pr\left[z = 1 \mid y \in \mathsf{Good}\right]$$

从 第 3.3 节， $\Pr[z = 1 \mid y \in \mathsf{Good}]$ 若在交替投影前插入 M_{Jor}，则不会改变。因此，我们可以通过将初始量子态 $\frac{1}{\sqrt{2^m}} \sum_{i=0}^{2^m-1} |i\rangle_{\mathsf{M}} |0^k\rangle_{\mathsf{K}} |i\rangle_{\mathsf{Aux}}$ 投影到其中一个子空间 S_i，得到测量后状态 $|v_i\rangle_{\mathsf{MK}} |u_i\rangle_{\mathsf{Aux}}$ ，然后对 y 和 z 进行采样，就如同我们从该量子态开始一样（这里我们还利用了 $\frac{1}{\sqrt{2^m}} \sum_{i=0}^{2^m-1} |i\rangle_{\mathsf{M}} |0^k\rangle_{\mathsf{K}} |i\rangle_{\mathsf{Aux}}$可以写成 $\frac{1}{\sqrt{2^m}} \sum_i |v_i\rangle_{\mathsf{MK}} |u_i\rangle_{\mathsf{Aux}}$这种形式这一事实）。记 E_i 为获得测量后状态$|v_i\rangle_{\mathsf{MK}} |u_i\rangle_{\mathsf{Aux}}$这一事件。那么 $\Pr[E_i] = \frac{1}{2^m}$[8]。因此，

$$\begin{aligned}
\Pr\left[z = 1 \mid y \in \mathsf{Good}\right] &= \frac{\Pr\left[z = 1 \wedge y \in \mathsf{Good}\right]}{\Pr\left[y \in \mathsf{Good}\right]} \\
&= \frac{\sum_i \Pr[E_i] \Pr\left[y \in \mathsf{Good} \mid E_i\right] \Pr\left[z = 1 \mid E_i \wedge y \in \mathsf{Good}\right]}{\sum_i \Pr[E_i] \Pr\left[y \in \mathsf{Good} \mid E_i\right]} \\
&= \frac{\sum_i p_i \Pr\left[y \in \mathsf{Good} \mid E_i\right]}{\sum_i \Pr\left[y \in \mathsf{Good} \mid E_i\right]}
\end{aligned}$$

与引理 5相同，当我们从 $|v_i\rangle_{\mathsf{MK}} |u_i\rangle_{\mathsf{Aux}}$开始时，$y \in \mathsf{Good}$ 的概率等于在 $2N$ 次偏置为 p_i的独立抛硬币中，正面朝上的次数（记为 cnt）为偶数且大于或等于 $N(a+b)$的概率。

因此，如果 $p_i < a$，根据切尔诺夫界，

$$\Pr\left[y \in \mathsf{Good} \mid E_i\right] \leq \Pr\left[cnt \geq N(a+b)\right] \leq \exp(-2Np_i(\frac{a+b}{2p_i}-1)^2/(1+\frac{a+b}{2p_i})) \leq \exp(-N\frac{(b-a)^2}{3a+b})$$

[8]一般来说，$\Pr[E_i] = \frac{1}{2^m}$对每个 $i \in A$。因此，下面所有的求和都只针对 $i \in A$。

根据引理 5，$\Pr[y \in \mathsf{Good} \mid E_1] \geq \frac{1}{4}$。因此，

$$\sum_{p_i \geq a}(p_i - a)\Pr[y \in \mathsf{Good} \mid E_i] - \sum_{p_i < a}(a - p_i)\Pr[y \in \mathsf{Good} \mid E_i] \geq (b-a)\frac{1}{4} - 2^m a \exp(-N\frac{(b-a)^2}{3a+b}) > 0$$

这里我们利用了这样一个事实：只有 2^m $|v_i\rangle$s，因为 P^1 的秩为 2^m。[9]

上述不等式可重写为 $\sum_i p_i \Pr[y \in \mathsf{Good} \mid E_i] > a \sum_i \Pr[y \in \mathsf{Good} \mid E_i]$.因此，$\Pr\left[\mathsf{V}_x^{|\mathsf{PSPACE}\rangle}(\rho_{\mathsf{M}})\ \text{接受}\right] \geq a$，至此命题的证明完成。 □

引理 4可由上述两个命题直接推出。

5 预言机辅助公钥量子货币的不安全性

在本节中，我们将使用第 4 节中的合成器作为构件，来攻击这样的预言机辅助公钥量子货币方案：其中 $\mathcal{O}$ 是由随机预言机 $\mathcal{R}$和 $|\mathsf{PSPACE}\rangle$组成的混合预言机。形式化地说：

定理 5. 不存在可重复使用且安全的预言机辅助公钥量子货币方案$(\mathsf{KeyGen}^{\mathcal{R},\,|\mathsf{PSPACE}\rangle}, \mathsf{Mint}^{\mathcal{R},\,|\mathsf{PSPACE}\rangle}, \mathsf{Ver}^{\mathcal{R},\,|\mathsf{PSPACE}\rangle})$，其中 $\mathcal{R}$ 是一个随机预言机。

非正式地说，我们在定理 4 中的合成器适用于一致预言机算法 $\mathsf{V}^{|\mathsf{PSPACE}\rangle}$。然而，在我们旨在攻击的预言机辅助公钥量子货币方案中，验证算法除了访问 $\mathcal{R}$ 之外，还可以访问 $|\mathsf{PSPACE}\rangle$。受 [CKP15；AK22], 的启发，我们尝试移除 $\mathcal{R}$ ，并用一个良好数据库对其进行模拟。基于第 2.2 节中的思路，我们给出以下攻击者。

令 $\mathcal{O}$ 为由随机预言机 $\mathcal{R}$ 和 $|\mathsf{PSPACE}\rangle$组成的混合预言机。对于一个 δ_r-可重用的 δ_s-安全预言机辅助量子货币方案 $(\mathsf{KeyGen}^{\mathcal{O}}, \mathsf{Mint}^{\mathcal{O}}, \mathsf{Ver}^{\mathcal{O}})$ 其中 $\delta_r = 0.99$，$\delta_s = \mathsf{negl}(n)$，记 $l(n)$ 为一次执行 $\mathsf{KeyGen}^{\mathcal{O}}$ 和一次执行$\mathsf{Mint}^{\mathcal{O}}$所进行的对 $\mathcal{R}$ 的查询次数。由 $\mathsf{Ver}^{\mathcal{O}}$的效率可知，存在一个一致预言机算法 $\mathsf{V}^{|\mathsf{PSPACE}\rangle} = (\mathsf{V}_x^{|\mathsf{PSPACE}\rangle})_{x\in\{0,1\}^*}$，使得运行 $\mathsf{V}^{|\mathsf{PSPACE}\rangle}{}_{(pk,\ D,\ s)}(\rho)$ 与运行 $\mathsf{Ver}^{\mathcal{O}}(pk, (s, \rho))$ 相同，其中 $\mathcal{R}$ 由数据库 D模拟。

设 $\epsilon = 0.01$, $b = 1 - \sqrt{1 - \delta_r + \epsilon}$, $a = 0.99b$。根据定理 4，存在一个多项式时间一致预言机算法 $\mathsf{Syn}^{|\mathsf{PSPACE}\rangle}$ ，能够生成一个具有保证值 a 和阈值 b的$\mathsf{V}^{|\mathsf{PSPACE}\rangle}{}_{(pk,\ D,\ s)}$ “几乎最优”的见证态。现在构造对手 $\mathcal{A}^{\mathcal{O}}$。

对手 $\mathcal{A}^{\mathcal{O}}$。它以一个有效钞票（$s$，$\rho_s$）和公钥 pk作为输入，并按如下方式运行。

1. 设 $t \xleftarrow{\$} \{0, \ldots, \lceil \frac{l}{\epsilon} \rceil - 1\}$。设 $D = \emptyset$, $\rho_s^{(0)} = \rho_s$。重复执行以下步骤 t 次。在第 i^{th} 次迭代中，
   (a) $\rho_s^{(i)} \otimes |\mathbf{x}\rangle\langle\mathbf{x}| \leftarrow \mathsf{Ver}^{\mathcal{O}}(pk, s, \rho_s^{(i-1)})$。
   (b) 将项目 (a) 中的查询-回答对添加到 $\mathcal{R}$ D。

[9] 对于一般情形，我们仅对满足 $i \in A$ 和 $|A|$ 等于 P^1 的秩（即 2^m）的项求和。

2. 记作 $D_0 = D$。

3. 对于 $k = 0,\ 1,\ \cdots,\ N(n) - 1$ 其中 $N(n) = \frac{100l(n)}{(1-\sqrt{1-\delta_r+\epsilon})^2}$ 关于 n 是多项式，

(a) $\sigma_{D_k} \leftarrow \mathsf{Syn}^{|\mathsf{PSPACE}\rangle}(pk,\ D_k,\ s)$。(b) 运行 $\mathsf{Ver}^{\mathcal{O}}(pk,\ (s,\ \sigma_{D_k}))$。(c) 令 D_{k+1} 由 (b) 项中对 $\mathcal{R}$ 的所有查询-回答对以及 D_k 中的对组成。

4. $j \xleftarrow{\$} \{0, 1, \ldots, N(n) - 1\}$。

5. 输出 $\phi = \phi_1 \otimes \phi_2$，其中 $\phi_i \leftarrow \mathsf{Syn}^{|\mathsf{PSPACE}\rangle}(pk,\ D_j,\ s)$ $(i = 1,\ 2)$。

对 $\mathcal{A}^{\mathcal{O}}$ 现在证明 $\mathcal{A}^{\mathcal{O}}$ 会输出我们想要的结果。我们将使用在 $\mathcal{A}^{\mathcal{O}}$ 的构造中定义的记号。

定理 6. 给定由 $\mathsf{KeyGen}^{\mathcal{O}}$ 和 $\mathsf{Mint}^{\mathcal{O}}$ 生成的输入 $(pk, (s, \rho_s))$，$\mathcal{A}^{\mathcal{O}}$ 输出两张与序列号 s 关联的声称的钞票，它们将以高概率被接受。形式化地说：

$$\Pr\left[\mathsf{Ver}^{\mathcal{O}}(pk, (s, \phi_1))\ accepts\ and\ \mathsf{Ver}^{\mathcal{O}}(pk, (s, \phi_2))\ accepts\right] \geq 1.8\left(1 - \sqrt{1 - \delta_r + \epsilon}\right)^2 - 1,$$

其中，概率是关于 $\mathcal{R}$ 的随机性、为 $\mathcal{A}^{\mathcal{O}}$ 生成输入时所用的随机性（即 $\mathsf{KeyGen}^{\mathcal{O}}$ 和 $\mathsf{Mint}^{\mathcal{O}}$ 的随机性），以及我们的对手 $\mathcal{A}^{\mathcal{O}}$ 的随机性。

定理 6 的证明定理 6。证明分为两部分。粗略地说，在第一部分中，我们将证明，对于每个 k，σ_{D_k} 在模拟中表现良好，即 $\mathsf{V}^{\mathsf{PSPACE}}_{(pk,D_k,s)}(\ \sigma_{D_k})$ 以高概率接受；在第二部分中，我们将证明，对于每个 k，若 $\mathsf{Ver}^{\mathcal{O}}(pk, (s, \cdot))$ 在输入 σ_{D_k} 上表现得与 $\mathsf{V}^{\mathsf{PSPACE}}_{(pk,D_k,s)}$ 相差甚远，那么我们就能取得进展。随后，我们将结合这些结果来证明 定理 6。

第一部分只要存在适用于 $\mathsf{V}^{|\mathsf{PSPACE}\rangle}_{(pk,D_k,s)}$ 的良好见证态，定理 4 中的合成器 Syn 就能良好工作。我们选取 $\rho_s^{(t)}$ 作为良好见证态，因为根据可重复使用性的定义，它以高概率被 $\mathsf{Ver}^{\mathcal{O}}$ 接受。我们首先说明，以高概率，我们的数据库包含对 $\rho_s^{(t)}$ 运行验证所需的信息，因此 Ver 无法区分自己是在与随机预言机 $\mathcal{R}$ 交互，还是与模拟的预言机交互。具体来说：

断言 3。 设 $D_{\mathsf{KeyGen}}, D_{\mathsf{Mint}}$ 是在生成输入 pk 以及 (s, ρ_s)（即执行 $\mathsf{KeyGen}^{\mathcal{O}}$ 和 $\mathsf{Mint}^{\mathcal{O}}$）期间发出的查询-答案对。那么

$$\Pr\left[\mathsf{Ver}^{\mathcal{O}}(pk, (s, \rho_s^{(t)}))\ queries\ in\ D_{\mathsf{Mint}} \cup D_{\mathsf{KeyGen}} - D\right] \leq \epsilon,$$

其中，概率取自 $\mathcal{R}$ 的随机性、为 $\mathcal{A}^{\mathcal{O}}$ 生成输入时的随机性，以及我们的对手 $\mathcal{A}^{\mathcal{O}}$ 的随机性。

证明。我们只关心 $l(n)$ 个查询位置（即 $D_{\mathsf{Mint}} \cup D_{\mathsf{KeyGen}}$ 内的位置），并反复采样 $t \xleftarrow{\$} \{0, \ldots, \lceil \frac{l}{\epsilon} \rceil - 1\}$ 次。因此，直观而言，D 应该能揭示我们关心的所有位置。形式化地说，

$$\Pr\left[\mathsf{Ver}^{\mathcal{O}}(pk,(s,\rho_s^{(t)}))\text{ queries in }D_{\mathsf{Mint}}\cup D_{\mathsf{KeyGen}}-D\right]$$

$$\leq \sum_{q\in D_{\mathsf{Mint}}\cup D_{\mathsf{KeyGen}}}\Pr\left[t=\min_j\left[\mathsf{Ver}^{\mathcal{O}}(pk,(s,\rho_s^{(j)}))\text{ queries }q\right]\right]$$

$$\leq l\cdot\frac{1}{\lceil\frac{l}{\epsilon}\rceil}$$

$$\leq\epsilon$$

其中，概率仅针对对手 $\mathcal{A}^{\mathcal{O}}$ 的随机性；我们利用了这样一个事实： t 是从 $\{0,1,\ldots,\lceil\frac{l}{\epsilon}\rceil-1\}$ 中均匀随机选取的，因此它与 $\min_j\left[\mathsf{Ver}^{\mathcal{O}}(pk,(s,\rho_s^{(j)}))\text{ 个查询 }q\right]$（这些查询可能服从某种分布，但无论如何都独立于 t ）匹配的概率小于或等于$1\lceil\frac{l}{\epsilon}\rceil$。

将 $\mathcal{R}$ 的随机性以及 $\mathcal{A}^{\mathcal{O}}$输入的生成随机性考虑在内后，我们可以得到该命题。

□

随机预言机 $\mathcal{R}$ 可以通过即时模拟来实现。因此，可以通过使用数据库 $D_{\mathsf{Mint}}\cup D_{\mathsf{KeyGen}}\cup D$模拟 $\mathcal{R}$，来实现 $\mathsf{Ver}^{\mathcal{O}}(pk,(s,\rho_s^{(t)}))$。如果 Ver 没有在 $(D_{\mathsf{Mint}}\cup D_{\mathsf{KeyGen}}\cup D)$ $\Delta D=D_{\mathsf{Mint}}\cup D_{\mathsf{KeyGen}}-D$ 中进行查询，那么它就无法区分 $\mathcal{R}$是通过 $D_{\mathsf{Mint}}\cup D_{\mathsf{KeyGen}}\cup D$还是通过 D模拟的。也就是说，如果 Ver 没有在 $D_{\mathsf{Mint}}\cup D_{\mathsf{KeyGen}}-D$ 内进行查询，那么 D就是一个良好的数据库，可用于模拟对输入 $\rho_s^{(t)}$的验证过程。因此， $\mathsf{V}^{|\mathsf{PSPACE}\rangle}{}_{(pk,D,s)}(\rho_s^{(t)})$的接受概率应接近 $\mathsf{Ver}^{\mathcal{O}}(pk,(s,\rho_s^{(t)}))$ 的接受概率；根据可重复使用性的定义，后者很高。平均而言，只有在向数据库中加入更多查询时，针对输入 $\rho_s^{(t)}$的模拟表现才会提升。因此，对于每个 k， $\rho_s^{(t)}$都应是 $\mathsf{V}^{|\mathsf{PSPACE}\rangle}{}_{(pk,D_k,s)}$的良好见证态。上述直觉由以下命题加以阐述。

命题 4. 我们使用与 D_{KeyGen} 和 D_{Mint} 相同的定义，如 断言 3所示。 $\forall k\in[N(n)\quad]$，

$$\Pr\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}_{(pk,D_k,s)}(\rho_s^{(t)})\ accepts\right]\geq\delta_r-\epsilon$$

其中，概率是针对 $\mathcal{R}$的随机性、为 $\mathcal{A}^{\mathcal{O}}$ 生成输入时所用的随机性，以及我们的对手 $\mathcal{A}^{\mathcal{O}}$的随机性。

证明。 该命题可由 定义 4 和 断言 3推出。以下概率与上述命题中的概率基于相同的随机性。

$$\Pr\left[\mathsf{Ver}^{\mathcal{O}}(pk,(s,\rho_s^{(t)}))\text{ accepts}\right]$$

$$\begin{aligned}
=&\Pr\left[\mathsf{Ver}^{\mathcal{O}}(pk,(s,\rho_s^{(t)}))\text{ accepts and queries in }D_{\mathsf{Mint}}\cup D_{\mathsf{KeyGen}}-D_k\right]\\
&+\Pr\left[\mathsf{Ver}^{\mathcal{O}}(pk,(s,\rho_s^{(t)}))\text{ accepts and never queries in }D_{\mathsf{Mint}}\cup D_{\mathsf{KeyGen}}-D_k\right]\\
\le&\Pr\left[\mathsf{Ver}^{\mathcal{O}}(pk,(s,\rho_s^{(t)}))\text{ queries in }D_{\mathsf{Mint}}\cup D_{\mathsf{KeyGen}}-D\right]\\
&+\Pr\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}_{(pk,D_k,s)}(\rho_s^{(t)})\text{ accepts and never queries in }D_{\mathsf{Mint}}\cup D_{\mathsf{KeyGen}}-D_k\right]\\
\le&\epsilon+\Pr\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}_{(pk,D_k,s)}(\rho_s^{(t)})\text{ accepts}\right]
\end{aligned}$$

其中我们利用了 $D\subseteq D_k$ 这一事实，以及可以使用即时模拟来实现$\mathcal{R}$这一事实。因此， $\mathsf{Ver}^{\mathcal{O}}(pk,(s,\rho_s^{(t)}))$ 也可以看作是使用 $D_{\mathsf{KeyGen}}\cup D_{\mathsf{Mint}}\cup D_k$模拟 $\mathcal{R}$，两者仅在 $(D_{\mathsf{KeyGen}}\cup D_{\mathsf{Mint}}\cup D_k)\Delta D_k=D_{\mathsf{KeyGen}}\cup D_{\mathsf{Mint}}-D_k$ 上有所不同。因此，如果 $\mathcal{R}$ 从未查询$D_{\mathsf{KeyGen}}\cup D_{\mathsf{Mint}}-D_k$，那么 Ver 无法区分 $\mathcal{R}$ 是通过 D_k 还是 $D_{\mathsf{Mint}}\cup D_{\mathsf{KeyGen}}\cup D_k$ 进行模拟的。上述不等式可以重新整理为

$$\Pr\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}_{(pk,D_k,s)}(\rho_s^{(t)})\text{ accepts}\right]\ge\Pr\left[\mathsf{Ver}^{\mathcal{O}}(pk,(s,\rho_s^{(t)}))\text{ accepts}\right]-\epsilon\ge\delta_r-\epsilon.$$

□

直观地，根据 主张 4，对于很大比例的 $\mathsf{V}^{|\mathsf{PSPACE}\rangle}{}_{(pk,D_k,s)}$，存在良好的见证态。因此，我们的合成器可以找到一个“几乎最优”的见证态。形式化地说：

命题 5。 对于每个 $k\in[N(n)]$,

$$\Pr\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}_{(pk,D_k,s)}(\sigma_{D_k})\ accepts\right]\ge 0.99\left(1-\sqrt{1-\delta_r+\epsilon}\right)^2$$

其中，概率是针对 $\mathcal{R}$的随机性、针对为 $\mathcal{A}^{\mathcal{O}}$ 生成输入时所用的随机性，以及我们的对手 $\mathcal{A}^{\mathcal{O}}$所用的随机性而言的。

证明。 除非另有说明，以下概率均针对与上述命题中的概率相同的随机性。

定义 $S:=\{(pk,D_k,s):\max_w\Pr\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}_{pk\ D_k\ s}(,,)(w)\text{ 接受}\right]\ge 1-\sqrt{1-\delta_r+\epsilon}\}$，其中概率仅针对 V的随机性。于是，根据 主张 4 和平均论证，

$$\Pr\left[(pk,D_k,s)\in S\right]\ge 1-\sqrt{1-\delta_r+\epsilon}$$

根据定理 4， $\forall(pk,D_k,s)\in S$, $\Pr\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}_{pk\ D_k\ s}(,,)(\sigma_{D_k})\text{ 接受}\right]\ge 0.99(1-\sqrt{1-\delta_r+\epsilon})$，其中概率仅取决于 V 的随机性。因此，

$$\Pr\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}_{(pk,D_k,s)}(\sigma_{D_k})\text{ accepts}\right]\ge 0.99\left(1-\sqrt{1-\delta_r+\epsilon}\right)^2.$$

□

第二部分 我们已经知道，σ_{D_k} 以很高的概率被 $\mathsf{V}^{|\mathsf{PSPACE}\rangle}{}_{(pk,D_k,s)}$ 接受。下一步是在 σ_{D_k} 上，将 $\mathsf{V}^{|\mathsf{PSPACE}\rangle}{}_{(pk,D_k,s)}$ 的接受概率与 $\mathsf{Ver}^{\mathcal{O}}(pk,(s,\cdot))$ 的接受概率联系起来。如果这两个量之差很大，那么使用 D_k 的模拟就不够好。也就是说，$\mathsf{Ver}^{\mathcal{O}}(pk,(s,\sigma_{D_k}))$ 会在 D_k 之外提出一些重要查询。因此在这种情况下，D_{k+1} 将包含更多重要查询，我们也就取得了进展。形式化地说：

命题 6. 我们使用与上文相同的记号。对于每个 $k\in[N(n)]$,

$$\begin{aligned}&\mathsf{Pr}\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}_{(pk,D_k,s)}(\sigma_{D_k})\ accepts\right]-\mathsf{Pr}\left[\mathsf{Ver}^{\mathcal{O}}(pk,(s,\sigma_{D_k}))\ accepts\right]\\ \leq&\mathbf{E}\left[|D_{k+1}\cap(D_{\mathsf{KeyGen}}\cup D_{\mathsf{Mint}})|\right]-\mathbf{E}\left[|D_k\cap(D_{\mathsf{KeyGen}}\cup D_{\mathsf{Mint}})|\right]\end{aligned}$$

其中，概率和期望是针对 $\mathcal{R}$ 的随机性、为 $\mathcal{A}^{\mathcal{O}}$ 生成输入时的随机性，以及我们的对手 $\mathcal{A}^{\mathcal{O}}$ 的随机性取值的。

证明。 除非另有说明，以下概率和期望均针对与上述命题中相同的随机性取值。

与 主张 4 中的论证类似，$\mathsf{V}^{|\mathsf{PSPACE}\rangle}{}_{(pk,s,D_k)}(\sigma_{D_k})$ 和 $\mathsf{Ver}^{\mathcal{O}}(pk,(s,\sigma_{D_k}))$ 仅在进行 $(D_{\mathsf{Mint}}\cup D_{\mathsf{KeyGen}}\cup D_k)\Delta D_k=D_{\mathsf{Mint}}\cup D_{\mathsf{KeyGen}}-D_k$ 中的查询时才会表现不同。因此，

$$\begin{aligned}&\mathsf{Pr}\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}_{(pk,D_k,s)}(\sigma_{D_k})\ \text{accepts}\right]\\ =&\mathsf{Pr}\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}_{(pk,D_k,s)}(\sigma_{D_k})\ \text{accepts and queries in}\ D_{\mathsf{Mint}}\cup D_{\mathsf{KeyGen}}-D_k\right]\\ &+\mathsf{Pr}\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}_{(pk,D_k,s)}(\sigma_{D_k})\ \text{accepts and never queries in}\ D_{\mathsf{Mint}}\cup D_{\mathsf{KeyGen}}-D_k\right]\\ \leq&\mathsf{Pr}\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}_{(pk,D_k,s)}(\sigma_{D_k})\ \text{queries in}\ D_{\mathsf{Mint}}\cup D_{\mathsf{KeyGen}}-D_k\right]\\ &+\mathsf{Pr}\left[\mathsf{Ver}^{\mathcal{O}}(pk,(s,\sigma_{D_k}))\ \text{accepts and never queries in}\ D_{\mathsf{Mint}}\cup D_{\mathsf{KeyGen}}-D_k\right]\\ =&\mathsf{Pr}\left[\mathsf{Ver}^{\mathcal{O}}(pk,(s,\sigma_{D_k}))\ \text{queries in}\ D_{\mathsf{Mint}}\cup D_{\mathsf{KeyGen}}-D_k\right]\\ &+\mathsf{Pr}\left[\mathsf{Ver}^{\mathcal{O}}(pk,(s,\sigma_{D_k}))\ \text{accepts and never queries in}\ D_{\mathsf{Mint}}\cup D_{\mathsf{KeyGen}}-D_k\right]\\ \leq&\mathsf{Pr}\left[(D_{k+1}-D_k)\cap(D_{\mathsf{KeyGen}}\cup D_{\mathsf{Mint}})\neq\emptyset\right]+\mathsf{Pr}\left[\mathsf{Ver}^{\mathcal{O}}(pk,(s,\sigma_{D_k}))\ \text{accepts}\right]\\ \leq&\mathbf{E}\left[|(D_{k+1}-D_k)\cap(D_{\mathsf{KeyGen}}\cup D_{\mathsf{Mint}})|\right]+\mathsf{Pr}\left[\mathsf{Ver}^{\mathcal{O}}(pk,(s,\sigma_{D_k}))\ \text{accepts}\right]\end{aligned}$$

Note that $D_k\subseteq D_{k+1}$,

$$\mathbf{E}\left[|(D_{k+1}-D_k)\cap(D_{\mathsf{KeyGen}}\cup D_{\mathsf{Mint}})|\right]=\mathbf{E}\left[|D_{k+1}\cap(D_{\mathsf{KeyGen}}\cup D_{\mathsf{Mint}})|\right]-\mathbf{E}\left[|D_k\cap(D_{\mathsf{KeyGen}}\cup D_{\mathsf{Mint}})|\right]$$

因此，该命题成立。 □

现在结合上述结果来证明 定理 6。以下概率和期望均是关于 $\mathcal{R}$ 的随机性、为 $\mathcal{A}^{\mathcal{O}}$ 生成输入时的随机性，以及我们的对手 $\mathcal{A}^{\mathcal{O}}$ 的随机性（因此也关于 t 和 j 的随机性），除非另有说明。

根据我们的构造和并集界，

$$\begin{aligned}
&\mathsf{Pr}\left[\mathsf{Ver}^{\mathcal{O}}(pk,(s,\phi_1)) \text{ accepts and } \mathsf{Ver}^{\mathcal{O}}(pk,(s,\phi_2)) \text{ accepts}\right]\\
\geq&2\mathsf{Pr}\left[\mathsf{Ver}^{\mathcal{O}}(pk,(s,\sigma_{D_j})) \text{ accepts}\right]-1\\
=&\frac{2}{N(n)}\sum_{k=0}^{N(n)-1}\mathsf{Pr}\left[\mathsf{Ver}^{\mathcal{O}}(pk,(s,\sigma_{D_k})) \text{ accepts}\right]-1
\end{aligned}$$

由命题 5 和命题 6，

$$\begin{aligned}
&\frac{1}{N(n)}\sum_{k=0}^{N(n)-1}\mathsf{Pr}\left[\mathsf{Ver}^{\mathcal{O}}(pk,(s,\sigma_{D_k})) \text{ accepts}\right]\\
\geq&\frac{1}{N(n)}\sum_{k=0}^{N(n)-1}\left(\mathsf{Pr}\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}_{(pk,D_k,s)}(\sigma_{D_k}) \text{ accepts}\right]-\mathbf{E}\left[|D_{k+1}\cap(D_{\mathsf{KeyGen}}\cup D_{\mathsf{Mint}})|\right]+\mathbf{E}\left[|D_k\cap(D_{\mathsf{KeyGen}}\cup D_{\mathsf{Mint}})|\right]\right)\\
\geq&\frac{1}{N(n)}\sum_{k=0}^{N(n)-1}\mathsf{Pr}\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}_{(pk,D_k,s)}(\sigma_{D_k}) \text{ accepts}\right]-\frac{1}{N(n)}\mathbf{E}\left[|D_{N(n)}\cap(D_{\mathsf{KeyGen}}\cup D_{\mathsf{Mint}})|\right]\\
\geq&\frac{1}{N(n)}\sum_{k=0}^{N(n)-1}\mathsf{Pr}\left[\mathsf{V}^{|\mathsf{PSPACE}\rangle}_{(pk,D_k,s)}(\sigma_{D_k}) \text{ accepts}\right]-\frac{l(n)}{N(n)}\\
\geq&0.99\left(1-\sqrt{1-\delta_r+\epsilon}\right)^2-0.01\left(1-\sqrt{1-\delta_r+\epsilon}\right)^2\\
\geq&0.9\left(1-\sqrt{1-\delta_r+\epsilon}\right)^2
\end{aligned}$$

因此，

$$\mathsf{Pr}\left[\mathsf{Ver}^{\mathcal{O}}(pk,(s,\phi_1)) \text{ accepts and } \mathsf{Ver}^{\mathcal{O}}(pk,(s,\phi_2)) \text{ accepts}\right]\geq 1.8\left(1-\sqrt{1-\delta_r+\epsilon}\right)^2-1,$$

由此结束了定理 6的证明。 □

定理 5. 所提出的对手 $\mathcal{A}^{\mathcal{O}}$ 是一个有效攻击，因为当 $\epsilon=0.01,\ \delta_r=0.99$ ，

$$1.8\left(1-\sqrt{1-\delta_r+\epsilon}\right)^2-1\geq 1.8(1-0.2)^2-1\geq 0.1,$$

这是不可忽略的。 □

6 量子访问的扩展

本节将探讨一种特殊情形：某些算法可以量子访问随机预言机。我们考虑可重用安全预言机辅助公钥量子货币方案（$\mathsf{KeyGen}^{|\mathcal{R}\rangle,|\mathsf{PSPACE}\rangle}$，$\mathsf{Mint}^{|\mathcal{R}\rangle,|\mathsf{PSPACE}\rangle}$，$\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$）。形式化地说：

定理 7。 不存在可重复使用且安全的预言机辅助公钥量子货币方案（$\mathsf{KeyGen}^{|\mathcal{R}\rangle,\,|\mathsf{PSPACE}\rangle}$,$\mathsf{Mint}^{|\mathcal{R}\rangle,\,|\mathsf{PSPACE}\rangle}$, $\mathsf{Ver}^{\mathcal{R},\,|\mathsf{PSPACE}\rangle}$），其中 $\mathcal{R}$ 是随机预言机。

不失一般性，我们可以假设这些算法仅针对输入长度为 $l(n)$ 的输入向随机预言机发起查询，并接收 1 比特输出，其中 l 是一个多项式。（如果它们针对不同输入长度向 $\mathcal{R}$ 发起查询，则假设最大输入长度为 $l'(n)$。令 $l(n) = l'(n)+\log l'(n)$。我们可以修改这些算法，使其针对输入长度为 $k(n)$的查询改为针对输入长度为 $l(n)$的输入发起查询，其中前 $k(n)$ 比特存储真实查询位置，中间的 $l'(n)-k(n)$ 比特为 0，最后 $\log l'(n)$ 比特表示 $k(n)$。）

令 Ver 向 $\mathcal{R}$发起 $q(n)$ 次经典查询。令 KeyGen 和 Mint 总共向$\mathcal{R}$ 发起 $q'(n)$ 次量子查询。分别用 δ_r 和 δ_s 表示该方案的可重复使用性和安全性，其中$\delta_r = 1-\mathsf{negl}(n)$ $\delta_s = \mathsf{negl}(n)$。在上下文清楚时，我们有时会为简洁起见省略 n。

值得注意的是，第 5 节中的攻击者并未利用以下事实： KeyGen和 Mint 只能向 $\mathcal{R}$进行经典查询。事实上，即使KeyGen 和 Mint 可以向 $\mathcal{R}$ 进行量子查询（只需对迭代次数作一些修改），同一攻击者仍然有效。具体而言，我们的攻击者构造如下，其中 $T(n)$、 $N(n)$、保证值 a 以及 Syn 的阈值将在稍后确定。

$\mathcal{A}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$ 其耗将有效钞票（s, ρ_s）和公钥 pk 作为输入，并按如下方式运行。

1. **测试阶段**：令 $t \xleftarrow{\$} \{0,1,\ldots,T(n)-1\}$。令 $D=\emptyset$, $\rho_s^{(0)}=\rho_s$。重复执行以下步骤 t 次。在第 i^{th} 次迭代中，

   (a) $\rho_s^{(i)} \otimes |\mathbf{x}\rangle\langle\mathbf{x}| \leftarrow \mathsf{Ver}^{\mathcal{R},\,|\mathsf{PSPACE}\rangle}(pk,\ s,\ \rho_s^{(i-1)})$。
   (b) 将项目 (a) 中的查询-答案对添加到 $\mathcal{R}$，放入 D。

2. **更新阶段**：令 $j \xleftarrow{\$} \{0,1,\ldots,N(n)-1\}$。令 $D_0=D$。运行以下过程 j 次。在k^{th} 次迭代中，

   (a) $\sigma_{D_{k-1}} \leftarrow \mathsf{Syn}^{|\mathsf{PSPACE}\rangle}(pk,\ D_{k-1},\ s)$。(b) 运行 $\mathsf{Ver}^{\mathcal{R},\,|\mathsf{PSPACE}\rangle}(pk,\ (s,\ \sigma_{D_{k-1}}))$。
   (c) 令 D_k 由条目 (b) 中针对 $\mathcal{R}$ 的所有查询-答案对以及 D_{k-1}中的对组成。

3. **合成阶段**：输出 $\phi=\phi_1\otimes\phi_2$，其中 $\phi_i \leftarrow \mathsf{Syn}^{|\mathsf{PSPACE}\rangle}(pk,\ D_j,\ s)$ $(i=1,\ 2)$。

对 $\mathcal{A}^{\mathcal{R},\,|\mathsf{PSPACE}\rangle}$ 的这种描述实际上等价于我们在第 5 节中的对手。我们将行 $j \xleftarrow{\$} \{0,1,\ldots,N(n)-1\}$ 移到前面，因为这样更容易分析。

剩下要做的就是证明一个类似于定理 6的结果。也就是说， $\mathcal{A}$ 的输出量子态将以高概率被接受。

定理 8. 给定由 $\mathsf{KeyGen}^{|\mathcal{R}\rangle,|\mathsf{PSPACE}\rangle}$ 和 $\mathsf{Mint}^{|\mathcal{R}\rangle,|\mathsf{PSPACE}\rangle}$, $\mathcal{A}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$生成的输入 $(pk, (s, \rho_s))$，输出两张与序列号 s 相关联的声称的钞票，它们以高概率会被接受。形式化地：

$$\Pr\left[\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,(s,\phi_1))\ accepts\ and\ \mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,(s,\phi_2))\ accepts\right] \geq 1.8\left(1-\sqrt{1-\delta_r+\epsilon}\right)^2-1,$$

其中，概率取自 $\mathcal{R}$ 的随机性、为 $\mathcal{A}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$ 生成输入时所用的随机性（即 $\mathsf{KeyGen}^{|\mathcal{R}\rangle,|\mathsf{PSPACE}\rangle}$ 和 $\mathsf{Mint}^{|\mathcal{R}\rangle,|\mathsf{PSPACE}\rangle}$的随机性），以及我们的对手 $\mathcal{A}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$的随机性。

与定理 6类似，我们将证明对 $\sigma_{D_j} \leftarrow \mathsf{Syn}^{|\mathsf{PSPACE}\rangle}(pk,\ D_j,\ s)$ 的验证以高概率接受，然后通过并集界证明该定理。

在第 5 节中，我们关键地依赖这样一个事实：每当我们犯错时，都会取得进展，也就是说，我们会在 $D_{\mathsf{KeyGen}} \cup D_{\mathsf{Mint}} - D_k$ 中恢复一个查询。然而，现在 KeyGen、Mint 可以进行量子查询。因此，KeyGen 和 Mint 可能会“触及”指数多个位置。幸运的是，Zhandry [Zha18] 提出的压缩预言机技术可以看作是将查询记录到数据库中的量子对应物。基本上，如果我们在纯化视图中运行所有算法，并在傅里叶基下查看包含预言机的寄存器（标记为 F），那么经过多项式次量子查询后，除多项式多个位置外，其余位置都将是 $|\hat{0}\rangle$，因此可以使用酉算子压缩该寄存器。在本文中，为了更好地模拟第 5 节中的 D_k 和 $D_{\mathsf{KeyGen}} \cup D_{\mathsf{Mint}} - D_k$，我们利用 Ver 只进行经典查询这一事实。具体来说，我们将维护一个寄存器来存储所有经典查询的数据库，并且只将数据库之外那些非 $|\hat{0}\rangle$ 的位置记录到 F 中。这两个寄存器将作为 D_k 和 $D_{\mathsf{KeyGen}} \cup D_{\mathsf{Mint}} - D_k$ 的对应物。我们将在第 6.2 节中详细阐述这一思路。

6.1 算法的纯化视图

根据 第 3.4 节，对于任何仅在输入长度为 $l(n)$ 时向随机预言机查询并接收 1 位输出的算法序列，我们可以改为运行所有算法的纯化视图，并利用由此得到的纯态来分析输出。这里所说的纯化视图，是指我们将以下列方式纯化算法的执行过程：

- 我们将引入一个寄存器 F，其中存储预言机的真值表。在第一个算法执行之前，将其初始化为预言机所有可能真值表的均匀叠加，即 $|\hat{0}\rangle^{\otimes 2^l}$。

- 我们不再对 $\mathcal{R}$进行量子查询，而是应用一个酉算子 U_Q：$|x\rangle_{\mathsf{Q}}|y\rangle_{\mathsf{A}}|f\rangle_{\mathsf{F}} \to |x\rangle_{\mathsf{Q}}|y \oplus f(x)\rangle_{\mathsf{A}}|f\rangle_{\mathsf{F}}$，其中 Q 存储查询位置，而 A 用于存储答案位。（U_Q 中的下标 Q 表示量子查询。）

- 我们不进行计算基测量，而是应用 $CNOT$ 将其复制到一个新的辅助寄存器中。

不失一般性，我们可以假设，对于对 $\mathcal{R}$的任何经典查询，查询答案寄存器在查询前总是被设为 $|0\rangle$。注意，对 $\mathcal{R}$ 进行经典查询，等价于先对查询位置进行计算基测量，然后对 $\mathcal{R}$进行量子查询。对查询答案额外进行一次计算基测量不会改变视图。因此，在纯化视图中，经典查询可以视为应用酉算子

$$U_C : |x\rangle_{\mathsf{Q}}|0\rangle_{\mathsf{A}}|f\rangle_{\mathsf{F}}|D_{\mathcal{R}}\rangle_{\mathsf{D}_{\mathcal{R}}} \to |x\rangle_{\mathsf{Q}}|f(x)\rangle_{\mathsf{A}}|f\rangle_{\mathsf{F}}|D_{\mathcal{R}},(x,f(x))\rangle_{\mathsf{D}_{\mathcal{R}}}$$

其中，$\mathsf{D}_{\mathcal{R}}$ 是一个寄存器，我们将用它来纯化经典查询中的计算基测量。$|D_{\mathcal{R}}\rangle$表示一系列对 $|(x_1,z_1)$、$(x_2,z_2)\cdots(x_k,z_k)\rangle$，其中 x_1、x_2、$\cdots$、x_k不一定互不相同，但如果$x_i = x_j$，那么我们就能保证$z_i = z_j$。这里 $\mathsf{D}_{\mathcal{R}}$ 有足够的空间。也就是说，$|(x_1,z_1)\cdots(x_k,z_k)\rangle$实际上表示 $|(x_1,z_1)\cdots(x_k,z_k)\ \perp,\ \cdots,\ \perp\rangle$

其中，$\perp$ 是一个表示空的特殊符号。虽然这并非标准用法，我们有时会将$D_{\mathcal{R}}$ 称为数据库。（ U_C 中的下标 C 表示经典查询。）

理解纯化视图的另一种便捷方式，是将算法的执行视为两个参与方——算法和预言机——之间的交互。预言机将维护两个私有寄存器 F 和 $\mathsf{D}_{\mathcal{R}}$ （以及一些初始化为 $|0\rangle$的辅助量子比特）。如果允许算法对 $\mathcal{R}$进行量子查询，算法就会向预言机提交 QA ，然后预言机应用 U_Q 并将 QA 发送回算法。如果只允许算法进行经典查询，算法就会向预言机提交 Q ，然后预言机在 A上放置一个新的辅助量子比特，应用 U_C ，并将 QA 发送给算法。

我们用$U_{\mathsf{KeyGen},\,n}$，$U_{\mathsf{Mint},\,n}$，$U_{\mathsf{Ver},\,n}$ 和$U_{\mathsf{Syn},\,n}$ 分别表示安全参数为 n 时， KeyGen、 Mint、 Ver 和 Syn 的纯化版本所对应的酉算子。于是，$U_{\mathsf{KeyGen},\,n}$，$U_{\mathsf{Mint},\,n}$和$U_{\mathsf{Ver},\,n}$ 都具有这样的形式：先准备第一个查询，然后通过应用 U_Q 或 U_C 来重复回答查询，并准备下一个查询（如果没有后续查询，则准备最终输出）。特别地，我们将$U_{\mathsf{Ver},\,n}$记作 $U_{q(n)}U_CU_{q(n)-1}\cdots U_CU_0$。在上下文清楚时，我们会省略下标 n。

令 $U'_{\mathsf{Ver}}:=U_qU_RU_{q-1}\cdots U_RU_0$，其中 U_R（下标 R 表示记录）是一个酉算子；除了进行经典查询 U_C 外，它还会将查询-答案对记录到由 $\mathcal{A}$ 维护的数据库中。也就是说，

$$U_R:|x\rangle_{\mathsf{Q}}|0\rangle_{\mathsf{A}}|D_{\mathcal{A}}\rangle_{\mathsf{D}_{\mathcal{A}}}|f\rangle_{\mathsf{F}}|D_{\mathcal{R}}\rangle_{\mathsf{D}_{\mathcal{R}}}\to|x\rangle_{\mathsf{Q}}|f(x)\rangle_{\mathsf{A}}|D_{\mathcal{A}},(x,f(x))\rangle_{\mathsf{D}_{\mathcal{A}}}|f\rangle_{\mathsf{F}}|D_{\mathcal{R}},(x,f(x))\rangle_{\mathsf{D}_{\mathcal{R}}}$$

其中 $\mathsf{D}_{\mathcal{A}}$ 是存储由 $\mathcal{A}$ 维护的数据库的寄存器。同样， $|D_{\mathcal{A}}\rangle$ 和 $|D_{\mathcal{R}}\rangle$ 表示由查询-答案对组成的序列，其中查询位置不必各不相同，但这些对必须保持一致。 $\mathsf{D}_{\mathcal{A}}$ 有足够的空间。

很容易看出， U'_{Ver} 对应于运行 $\mathsf{Ver}^{\mathcal{R},\,|\mathsf{PSPACE}\rangle}$ ，同时对手记录由 $\mathsf{Ver}^{\mathcal{R},\,|\mathsf{PSPACE}\rangle}$ 发出的查询-答案对。

那么，在纯化视图中， $\mathcal{A}^{\mathcal{R},\,|\mathsf{PSPACE}\rangle}$ 如下（我们将其记为 $U_{\mathcal{A}}$）：

1. 给定输入公钥、序列号、声称的纸币，以及包含预言机真值表的寄存器，引入一个寄存器 T ，并将其初始化为 $\frac{1}{\sqrt{T(n)}}\sum_{t=0}^{T(n)-1}|t\rangle_{\mathsf{T}}$ ；再引入一个寄存器 J ，并将其初始化为 $\frac{1}{\sqrt{N(n)}}\sum_{j=0}^{N(n)-1}|j\rangle_{\mathsf{J}}$。

2. **测试阶段**：在给定 T 中的内容为 t 的条件下，对钞票依次应用 U'_{Ver} 共 t 次。（或者等价地，应用酉算子 $U_{\mathsf{Test}}:=\sum_{t=0}^{T(n)-1}U'^{t}_{\mathsf{Ver}}\otimes|t\rangle\langle t|_{\mathsf{T}}$，其中 U'^{t}_{Ver} 表示应用 U'_{Ver} 共 t 次。）

3. **更新 阶段**：在给定 J 中的内容为 j 的条件下，应用以下操作 j 次 s:

   (a) 对迄今为止我们学到的所有查询-答案对（即 $\mathsf{D}_{\mathcal{A}}$ 中的内容）应用 U_{Syn}。(b) 对在项目 (a) 中合成的量子态应用 U'_{Ver}。

   （或者等价地，应用酉算子 $U_{\mathsf{Upd}}:=\sum_{j=0}^{N(n)-1}(U'_{\mathsf{Ver}}U_{\mathsf{Syn}})^j\otimes|j\rangle\langle j|_{\mathsf{J}}$，其中 $(U'_{\mathsf{Ver}}U_{\mathsf{Syn}})^j$ 表示交替应用 U_{Syn} 和 U'_{Ver} 共 j 次。）

4. **合成阶段**：在 $\mathsf{D}_{\mathcal{A}}$ 中的查询-答案对上应用U_{Syn_1} 和 U_{Syn_2} ，以获得两张声称的钞票，其中 U_{Syn_1} 和 U_{Syn_2} 是作用于不同寄存器的 U_{Syn}。

然后，可以通过在纯化视图中运行以下算法序列，分析以下算法在相应量子态上的接受概率（将 $\mathcal{R}$ 的随机性考虑在内）。

$\mathbf{Ver}^{\mathcal{R},\,|\mathsf{PSPACE}\rangle}(\boldsymbol{pk},\ (\boldsymbol{s},\ \cdot))$ 上 $\boldsymbol{\rho_s^{(t)}}$接受概率是指.....的概率

1. 按顺序以纯化视图运行算法 $\mathsf{KeyGen}^{|\mathcal{R}\rangle,\,|\mathsf{PSPACE}\rangle}$，$\mathsf{Mint}^{|\mathcal{R}\rangle,\,|\mathsf{PSPACE}\rangle}$ 和 $\mathcal{A}^{\mathcal{R},\,|\mathsf{PSPACE}\rangle}$，其中 KeyGen 的输入是以一元形式表示的安全参数，Mint 的输入是与 KeyGen的密钥相对应的输出寄存器，而 $\mathcal{A}$ 的输入是 Mint 的输出寄存器以及公钥寄存器。

2. 此外，以纯化视图运行 $\mathsf{Ver}^{\mathcal{R},\,|\mathsf{PSPACE}\rangle}$，其中 Ver 的输入是公钥寄存器、序列号寄存器以及 $\rho_s^{(t)}$的寄存器（它位于 $\mathcal{A}$的工作空间寄存器内）。

3。测量上述 $\mathsf{Ver}^{\mathcal{R},\,|\mathsf{PSPACE}\rangle}$ 的结果寄存器，并得到 Accept。

$\mathbf{Ver}^{D_j,\,|\mathsf{PSPACE}\rangle}(\boldsymbol{pk},\ (\boldsymbol{s},\ \cdot))$ 上 $\boldsymbol{\rho_s^{(t)}}$这里 $\mathsf{Ver}^{D_j,\,|\mathsf{PSPACE}\rangle}$ 表示运行 $\mathsf{Ver}^{\mathcal{R},\,|\mathsf{PSPACE}\rangle}$ 其中，对 $\mathcal{R}$ 的查询由数据库 D_j（$\mathsf{D}_{\mathcal{A}}$中的所有查询-答案对）作答。

定义一个酉算子

$$U_D : |x\rangle_{\mathsf{Q}} |0\rangle_{\mathsf{A}} |D_j\rangle_{\mathsf{D}_{\mathcal{A}}} \to \begin{cases} |x\rangle_{\mathsf{Q}} |D_j(x)\rangle_{\mathsf{A}} |D_j, (x, D_j(x))\rangle_{\mathsf{D}_{\mathcal{A}}}, & x \in D_j \\ |x\rangle_{\mathsf{Q}} \sum_{z=0}^{1} \frac{1}{\sqrt{2}} |z\rangle_{\mathsf{A}} |D_j, (x, z)\rangle_{\mathsf{D}_{\mathcal{A}}}, & x \notin D_j \end{cases}$$

其中，$|D_j\rangle$表示查询-答案对序列 $|(x_1,\ z_1),\ (x_2,\ z_2)\ \cdots (x_k,\ z_k)\rangle$，其中查询位置 $x_1,\ x_2,\ \cdots,\ x_k$ 不必互不相同，但这些对必须一致。$x \in D_j$表示存在 z，使得 $(x,\ z)$ 是 D_j 中的一对，我们将此 z 记为 $D_j(x)$。$x \notin D_j$表示对于所有 z，$(x,\ z)$ 都不是 D_j中的一对。（U_D 中的下标 D 用于通过数据库进行模拟。）

然后，应用 U_D 就是使用 D_j 回答查询 x（如果 x 在数据库中，就使用 D_j 回答查询；否则，随机给出一个答案，同时将该查询-答案对记录到数据库中，以供后续使用）。因此，$\mathsf{Ver}^{D_j,|\mathsf{PSPACE}\rangle}$ 的纯化版本为

$$U_{\mathsf{Sim}} := U_q U_D U_{q-1} \cdots U_D U_0.$$

因此，$\mathsf{Ver}^{D_j,|\mathsf{PSPACE}\rangle}(pk, (s, \cdot))$ 在 $\rho_s^{(t)}$ 上的接受概率，是以下事件发生的概率：

1. 在 $\rho_s^{(t)}$ 上运行 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,(s, \cdot))$ 的第一步。

2. 此外，以纯化视图运行 $\mathsf{Ver}^{D_j,|\mathsf{PSPACE}\rangle}$，其中 Ver 的输入是公钥寄存器、序列号寄存器以及 $\rho_s^{(t)}$ 的寄存器（输入与在 $\rho_s^{(t)}$ 上运行 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,(s, \cdot))$ 的情形相同）。

3. 测量上述内容的结果寄存器 $\mathsf{Ver}^{D_j,|\mathsf{PSPACE}\rangle}$，并得到 Accept .

$\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,(s,\cdot))$ **对** σ_{D_j} 接受概率是指.....的概率

1. 在 $\rho_s^{(t)}$ 上运行情况 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,(s,\cdot))$ 的第一步。

2. 此外，以纯化视图运行 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$，其中 Ver 的输入是公钥寄存器、序列号寄存器以及 σ_{D_j} 的寄存器（它是 $\mathcal{A}$ 输出量子态的第一个寄存器）。

3. 测量上述 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$ 的结果寄存器，并得到 Accept。

$\mathsf{Ver}^{D_j,|\mathsf{PSPACE}\rangle}(pk,(s,\cdot))$ **在** σ_{D_j} 上的接受概率是以下事件发生的概率

1. 在 $\rho_s^{(t)}$ 上运行情况 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,(s,\cdot))$ 的第一步。

2. 此外，以纯化视图运行 $\mathsf{Ver}^{D_j,|\mathsf{PSPACE}\rangle}$，其中 Ver 的输入是公钥寄存器、序列号寄存器以及 σ_{D_j} 的寄存器（输入与 σ_{D_j} 上的情况 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,(s,\cdot))$ 相同）。

3. 对上述 $\mathsf{Ver}^{D_j,\ |\mathsf{PSPACE}\rangle}$ 的结果寄存器进行测量，并得到 Accept。

6.2 压缩与解压缩

直观地说， F 中的 $|\hat{0}\rangle$ 位置是值域上的均匀叠加态，且不与其他任何内容纠缠，因此可以将其视为独立地从值域中均匀随机选取一个值，这正是模拟所做的事情。在纯经典查询的情形下，它相当于算法序列中从未被查询过的位置。

在本小节中，我们将展示如何从纯态中提取 D_k 和 $D_{\mathsf{KeyGen}} \cup D_{\mathsf{Mint}} - D_k$的类似物。粗略地说，记录下来的经典查询是 D_k 的类似物，而我们将压缩寄存器 F ，以提取 D_k 之外那些非$|\hat{0}\rangle$位置，从而构成$D_{\mathsf{KeyGen}} \cup D_{\mathsf{Mint}} - D_k$的类似物。

由于写出并分析压缩的逆操作更为容易，我们先给出解压缩酉算子 Decomp的形式化描述。回想一下， $\mathsf{D}_{\mathcal{R}}$ 存储了对 $\mathcal{R}$的所有经典查询。 F是一个用于随机函数的寄存器。定义

$$\mathsf{Decomp} : |D_F\rangle_{\mathsf{F}} |D_{\mathcal{R}}\rangle_{\mathsf{D}_{\mathcal{R}}} \to |f_0, f_1, \cdots, f_{2^l-1}\rangle_{\mathsf{F}} |D_{\mathcal{R}}\rangle_{\mathsf{D}_{\mathcal{R}}}$$

其中， $|D_F\rangle_{\mathsf{F}}$可写成由以下各对组成的序列： $|(x_1,\ \widehat{y_1}),\ (x_2,\ \widehat{y_2})\ \cdots,\ (x_{k'},\ \widehat{y_{k'}})\rangle_{\mathsf{F}}$； $|D_{\mathcal{R}}\rangle_{\mathsf{D}_{\mathcal{R}}}$可写成由以下各对组成的序列： $|(x'_1,\ z_1),\ (x'_2,\ z_2)\ \cdots,\ (x'_k,\ z_k)\rangle$，且输入 $|D_F\rangle_{\mathsf{F}} |D_{\mathcal{R}}\rangle_{\mathsf{D}_{\mathcal{R}}}$ 满足

- if $x'_i = x'_j$, then $z_i = z_j$;
- $x_1 < x_2 < \cdots < x_{k'}, \widehat{y_i} \neq \widehat{0}$;
- $\forall i, j, x_j \neq x'_i$;

且输出满足

- 若 $x'_j = i$，则 $f_i = z_j$;
- 如果 $x_j = i$，那么 $f_i = \widehat{y_j}$;

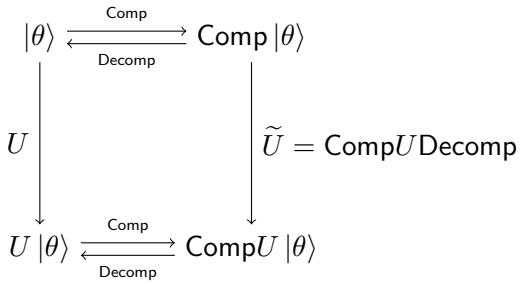
- 如果 $\forall j,\ x_j \neq i,\ x'_j \neq i$，那么 $f_i = \widehat{0}$。

粗略地说，我们通过考察这些对 $(x'_1,\ z_1)$、$(x'_2,\ z_2)\ \cdots$、$(x'_k,\ z_k)$ 和 $(x_1,\ \widehat{y_1})$、$(x_2,\ \widehat{y_2})\ \cdots$、$(x_{k'},\ \widehat{y_{k'}})$，来填充 $f_0,\ f_1,\ \cdots,\ f_{2^{l'}-1}$。其余位置则全部填入 $\hat{0}$。

这里，我们的寄存器 F 也有足够的空间。由于我们的随机函数只有单比特输出，$z_1,\ z_2,\ \cdots,\ z_k,\ y_1,\ y_2,\ \cdots,\ y_{k'} \in \{0,1\}$。回想一下，$|\hat{0}\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$ $|\hat{1}\rangle = |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)$。可以验证，上述形式中的任意两个输入都正交，并且它们映射到正交的输出。因此，我们可以定义其他不属于上述形式的输入的输出，使得 Decomp 是一个酉算子。

更多记号 为简洁起见，默认情况下，当我们写作 $D_{\mathcal{R}}$ 时，指的是一列一致对 $(x'_1, z_1), (x'_2, z_2)\ \cdots,$ (x'_k, z_k)。$x \in D_{\mathcal{R}}$ 表示 $\exists i, x'_i = x$。当 $x \in D_{\mathcal{R}}$ 时，$D_{\mathcal{R}}(x)$ 表示 z_i where $x'_i = x$。当我们写作 D_F 时，指的是满足上述输入要求第二项的一列 $(x_1, \widehat{y_1}), (x_2, \widehat{y_2})\ \cdots, (x_{k'}, \widehat{y_{k'}})$。$x \in D_F$ 表示 $\exists i,$ $x_i = x$.$\frown$ $D_F(x)$ 表示当 $x \in D_F$ 时的 $\widehat{y_i}$ where $x_i = x$。$D_F(x)$ 表示当 $x \in D_F$ 时的 y_i where$x_i = x$。$D_F - x$ 表示从序列 $D_F$where $x_i = x$ 中删除 $x_i, \widehat{y_i}$ 后得到的序列。定义 $D_{\mathcal{R}} \cap D_F = \{x :\ x \in D_{\mathcal{R}}$ and $x \in D_F\}$。

上述酉算子的逆操作是压缩 Decomp，它可以将我们的数据库 $D_{\mathcal{R}}$ 和寄存器 F 中的真值表作为输入，并将它们压缩成两个数据库 $D_{\mathcal{R}}$ 和D_F。将其定义为 Comp := Decomp†。这两个酉算子使我们能够在未解压视图（一个用于经典查询的数据库和一个真值表）与压缩视图（一个用于经典查询的数据库和另一个数据库）之间切换。下面用一幅图来说明这一点。$|\theta\rangle$ 是一个未经压缩的任意量子态。U 是未解压视图中的酉算子（它以未经压缩的量子态~为输入，并输出一个未经压缩的量子态）。然后，$U :=$ Comp$U$Decomp是 U 在压缩视图中的版本（它以压缩后的量子态为输入，并输出一个压缩后的量子态）。从现在起，我们写酉算子~时，指的都是压缩视图中的酉算子。



读者可以将 $D_{\mathcal{R}}$ 视为数据库 D_k 在 第 5 节 中的一个类比，并将 D_F 视为数据库 $D_{\mathsf{KeyGen}} \cup D_{\mathsf{Mint}} - D_k$ 在 第 5 节中的一个类比。粗略地说，$D_{\mathcal{R}}$ 存储着我们的经典查询。D_F 中的查询位置是由 KeyGen 和 Mint提出的，但未记录在 $D_{\mathcal{R}}$中。看过下面压缩视图中的查询酉算子后，我们就能更好地理解这一类比。

回顾 U_C、U_R 和 U_D，它们来自第 6.1 节。它们分别用于经典查询、记录过程中的经典查询和模拟经典查询。它们的压缩版本是~ ~ ~酉算子 $U_C :$ $=$ CompU_CDecomp，$U_R :=$ CompU_RDecomp 和 $U_D :=$ CompU_DDecomp。

根据 Decomp 的描述，我们可以得到 $D_F \cap D_{\mathcal{R}} = \emptyset$,

$$\widetilde{U_C}(|x\rangle_{\mathsf{Q}} |0\rangle_{\mathsf{A}} |D_F\rangle_{\mathsf{F}} |D_{\mathcal{R}}\rangle_{\mathsf{D}_{\mathcal{R}}})$$
$$= \begin{cases} |x\rangle_{\mathsf{Q}} |D_{\mathcal{R}}(x)\rangle_{\mathsf{A}} |D_F\rangle_{\mathsf{F}} |D_{\mathcal{R}}, (x, D_{\mathcal{R}}(x))\rangle_{\mathsf{D}_{\mathcal{R}}} & x \in D_{\mathcal{R}} \\ |x\rangle_{\mathsf{Q}} \frac{1}{\sqrt{2}} \sum_{z=0}^{1} |z\rangle_{\mathsf{A}} |D_F\rangle_{\mathsf{F}} |D_{\mathcal{R}}, (x, z)\rangle_{\mathsf{D}_{\mathcal{R}}} & x \notin D_{\mathcal{R}}, x \notin D_F \\ |x\rangle_{\mathsf{Q}} \frac{1}{\sqrt{2}} \sum_{z=0}^{1} (-1)^{z\widehat{D_F(x)}} |z\rangle_{\mathsf{A}} |D_F - x\rangle_{\mathsf{F}} |D_{\mathcal{R}}, (x, z)\rangle_{\mathsf{D}_{\mathcal{R}}} & x \notin D_{\mathcal{R}}, x \in D_F \end{cases}$$

$$\widetilde{U_R}(|x\rangle_{\mathsf{Q}} |0\rangle_{\mathsf{A}} |D_{\mathcal{A}}\rangle_{\mathsf{D}_{\mathcal{A}}} |D_F\rangle_{\mathsf{F}} |D_{\mathcal{R}}\rangle_{\mathsf{D}_{\mathcal{R}}})$$
$$= \begin{cases} |x\rangle_{\mathsf{Q}} |D_{\mathcal{R}}(x)\rangle_{\mathsf{A}} |D_{\mathcal{A}}, (x, D_{\mathcal{R}}(x))\rangle_{\mathsf{D}_{\mathcal{A}}} |D_F\rangle_{\mathsf{F}} |D_{\mathcal{R}}, (x, D_{\mathcal{R}}(x))\rangle_{\mathsf{D}_{\mathcal{R}}} & x \in D_{\mathcal{R}} \\ |x\rangle_{\mathsf{Q}} \frac{1}{\sqrt{2}} \sum_{z=0}^{1} |z\rangle_{\mathsf{A}} |D_{\mathcal{A}}, (x, z)\rangle_{\mathsf{D}_{\mathcal{A}}} |D_F\rangle_{\mathsf{F}} |D_{\mathcal{R}}, (x, z)\rangle_{\mathsf{D}_{\mathcal{R}}} & x \notin D_{\mathcal{R}}, x \notin D_F \\ |x\rangle_{\mathsf{Q}} \frac{1}{\sqrt{2}} \sum_{z=0}^{1} (-1)^{z\widehat{D_F(x)}} |z\rangle_{\mathsf{A}} |D_{\mathcal{A}}, (x, z)\rangle_{\mathsf{D}_{\mathcal{A}}} |D_F - x\rangle_{\mathsf{F}} |D_{\mathcal{R}}, (x, z)\rangle_{\mathsf{D}_{\mathcal{R}}} & x \notin D_{\mathcal{R}}, x \in D_F \end{cases}$$

$\widetilde{\ }$由于 U_D 不会作用于 F 和 $\mathsf{D}_{\mathcal{R}}$, $U_D = U_{D^\circ}$。因此

$$\widetilde{U_D}(|x\rangle_{\mathsf{Q}} |0\rangle_{\mathsf{A}} |D_{\mathcal{A}}\rangle_{\mathsf{D}_{\mathcal{A}}})$$
$$= \begin{cases} |x\rangle_{\mathsf{Q}} |D_{\mathcal{A}}(x)\rangle_{\mathsf{A}} |D_{\mathcal{A}}, (x, D_{\mathcal{A}}(x))\rangle_{\mathsf{D}_{\mathcal{A}}}, & x \in D_{\mathcal{A}} \\ |x\rangle_{\mathsf{Q}} \frac{1}{\sqrt{2}} \sum_{z=0}^{1} |z\rangle_{\mathsf{A}} |D_{\mathcal{A}}, (x, z)\rangle_{\mathsf{D}_{\mathcal{A}}}, & x \notin D_{\mathcal{A}} \end{cases}$$

$\widetilde{\ }$ $\widetilde{\ }$根据 U_C 和 U_R的描述，每当我们针对输入 $x \in D_{\mathcal{R}}$发起经典查询时，就用我们的数据库 $D_{\mathcal{R}}$ 回答，并记录 $(x,\ D_{\mathcal{R}}(x))$ 以备后用；每当我们针对输入 $x \notin D_{\mathcal{R}} \cup D_F$发起经典查询时，就用一个随机的 z回答，并将 $(x,\ z)$ 记录在数据库中以备后用；每当我们针对输入 $x \in D_F$发起经典查询时，我们实际上会从 D_F复制答案，记录下来，并从 D_F中移除 x。上述三种情况类似于经典的即时模拟：其中，对 D_k内部 $\mathcal{R}$的查询可以由 D_k回答；对 $D_{\mathsf{KeyGen}} \cup D_{\mathsf{Mint}} - D_k$内部 $\mathcal{R}$的查询应由 $D_{\mathsf{KeyGen}} \cup D_{\mathsf{Mint}} - D_k$ 保持一致地回答；对于对外部（$D_{\mathsf{KeyGen}} \cup D_{\mathsf{Mint}} - D_k$）$\cup D_k$的$\widetilde{\ }$ $\widetilde{\ }$查询，我们可以给出随机答案。值得指出的是，U_C 和 U_R 保持着 $D_{\mathcal{R}} \cap D_F$为空这一性质（类似于（$D_{\mathsf{KeyGen}} \cup D_{\mathsf{Mint}} - D_k$）$\cap D_k = \emptyset$）。

此外，回想一下，$U_{\mathsf{Ver}} = U_q U_C U_{q-1} \cdots U_1 U_C U_0$和 U_i不会作用于 F或 $\mathsf{D}_{\mathcal{R}}$。因此 $\widetilde{U_{\mathsf{Ver}}} = U_q U_C U_{q-1} \cdots U_1 U_C U_0 = U_q U_C U_{q-1} \cdots U_1 U_C U_0 = \widetilde{U_i} = U_i$, $\widetilde{U_{\mathsf{Ver}}} = U_q U_C U_{q-1} \cdots U_1 U_C U_0 = U_q U_C U_{q-1} \cdots U_1 U_C U_0$.类似地，回想一下，$\mathsf{Ver}^{D_j, |\mathsf{PSPACE}\rangle}$的纯化版本是 $U_{\mathsf{Sim}} = U_q U_D U_{q-1} \cdots U_D U_0$。因此 $\widetilde{U_{\mathsf{Sim}}} = U_q \widetilde{U_D} U_{q-1} \cdots \widetilde{U_D} U_0 = \widetilde{U_{\mathsf{Sim}}}$（因为 $U_D = U_D$）。

6.3 对 $\mathcal{A}^{\mathcal{R}, |\mathsf{PSPACE}\rangle}$的分析

在此，我们分析 $\mathsf{Ver}^{\mathcal{R}, |\mathsf{PSPACE}\rangle}$ 在我们的 $\mathcal{A}^{\mathcal{R}, |\mathsf{PSPACE}\rangle}$输出上的接受概率。我们沿用第 5 节中的思路。如果对符号有疑问，建议读者参阅附录 C中的符号表。

下面的命题可视为命题 6 的一个类比，只不过我们研究的是一般量子态，而不只是分析 σ_{D_k}。其基本含义是：当 $\mathsf{Ver}^{\mathcal{R}, |\mathsf{PSPACE}\rangle}$（对应于 $\widetilde{U_{\mathsf{Ver}}} = U_q \widetilde{U_C} \cdots \widetilde{U_C} U_0$）的行为与模拟 $\mathsf{Ver}^{D, |\mathsf{PSPACE}\rangle}$（对应于 $\widetilde{U_{\mathsf{Sim}}} = U_q U_D \cdots U_D U_0$）相差甚远时，验证后，$\mathsf{F}$中的对数（类似于 $|D_{\mathsf{KeyGen}} \cup D_{\mathsf{Mint}} - D_k|$）将$\widetilde{\ }$大幅减少。直观地说，大致而言，$U_C$ 和 U_D 只有在收到查询位置 时才会表现

不同；在这种情况下，应用 U_C 后， x 将从 D_F 中排除。因此， F中的对数会减少。形式化地说：

命题 2. 记 O 为对应于 F 中对数的可观测量。更具体地说， $O=\sum_{D_F}|D_F|\,|D_F\rangle\langle D_F|_{\mathsf{F}}$，其中 $|D_F|$ 是 D_F 中对的数量。

对于如下形式的量子态 $|\phi\rangle$ （即它处于压缩视图中，且 $\mathsf{D}_{\mathcal{R}}$和 $\mathsf{D}_{\mathcal{A}}$ 中的内容相同），

$$|\phi\rangle=\sum_{\substack{pk,s,m,D,D_F,g\\ s.t.\ D\cap D_F=\emptyset}}\alpha_{pk,s,m,D,D_F,g}\,|pk\rangle_{\mathsf{Pk}}\,|s\rangle_{\mathsf{S}}\,|m\rangle_{\mathsf{M}}\,|D\rangle_{\mathsf{D}_{\mathcal{A}}}\,|D_F\rangle_{\mathsf{F}}\,|D\rangle_{\mathsf{D}_{\mathcal{R}}}\,|g\rangle_{\mathsf{G}}\,.$$

设在 $|\phi\rangle\Big]$ 上运行时 $\mathsf{Pr}\Big[\widetilde{U_{\mathsf{Ver}}}$ 接受的概率为：当公钥、序列号和声称的货币态分别为 Pk、 S 和 M 时， $\widetilde{U_{\mathsf{Ver}}}$ 的接受概率（它也等于 U_{Ver} 在 $\mathsf{Decomp}\,|\phi\rangle$ 上的接受概率 $\widetilde{\ }$ ，因为 $U_{\mathsf{Ver}}\,|\phi\rangle=\mathsf{Comp}U_{\mathsf{Ver}}$（$\mathsf{Decomp}\,|\phi\rangle$）和Comp 不会作用于输出位）。

设 $\mathsf{Pr}\left[\widetilde{U_{\mathsf{Sim}}}\right.$ 在运行于 $\left.|\phi\rangle\right]$ 时接受的概率为 $\widetilde{U_{\mathsf{Sim}}}$ 的接受概率，其中公钥、序列号、声称的货币态以及用于模拟随机预言机的数据库分别处于 Pk、 S、 M 和 $\mathsf{D}_{\mathcal{A}}$ 中（这也等于 U_{Sim} 在 $\mathsf{Decomp}\,|\phi\rangle$ 上的接受概率）。

$$\begin{aligned}&\left|\mathsf{Pr}\left[\widetilde{U_{\mathsf{Ver}}}\ accepts\ when\ running\ on\ |\phi\rangle\right]-\mathsf{Pr}\left[\widetilde{U_{\mathsf{Sim}}}\ accepts\ when\ running\ on\ |\phi\rangle\right]\right|\\ \leq&\mathrm{TD}\left(\mathrm{Tr}_{\mathsf{FD}_{\mathcal{A}}\mathsf{D}_{\mathcal{R}}}(\widetilde{U_{\mathsf{Ver}}}\,|\phi\rangle\langle\phi|\,\widetilde{U_{\mathsf{Ver}}}^{\dagger}),\mathrm{Tr}_{\mathsf{FD}_{\mathcal{A}}\mathsf{D}_{\mathcal{R}}}(\widetilde{U_{\mathsf{Sim}}}\,|\phi\rangle\langle\phi|\,\widetilde{U_{\mathsf{Sim}}}^{\dagger})\right)\\ \leq&6\sqrt{q\left(\mathrm{Tr}(O\,|\phi\rangle\langle\phi|)-\mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}\,|\phi\rangle\langle\phi|\,\widetilde{U_{\mathsf{Ver}}}^{\dagger})\right)}\end{aligned}$$

证明。 第一个不等式直接源于以下事实：我们可以测量 $U_{\mathsf{Ver}}\,|\phi\rangle$ 和 $U_{\mathsf{Sim}}\,|\phi\rangle$ 中的一个量子比特（不是 $\widetilde{\ }$ $\widetilde{\ }$ 在 $\mathsf{FD}_{\mathcal{A}}\mathsf{D}_{\mathcal{R}}$ 中的量子比特），以判断它们是否接受；并且我们无法以大于两个量子态之间迹距离的概率区分它们。

至于第二个不等式，回顾一下： $\widetilde{U_{\mathsf{Ver}}}=U_q\widetilde{U_C}\cdots\widetilde{U_C}U_0$ 和 $\widetilde{\ }$ $U_{\mathsf{Sim}}=U_qU_D\cdots U_DU_0$。令 U'_D 与 U_D 相同，不同之处在于它使用 $D_{\mathcal{R}}$ 中的内容进行模拟，而不是使用 $D_{\mathcal{A}}$ 中的内容。更具体地说，

$$U'_D(|x\rangle_{\mathsf{Q}}\,|0\rangle_{\mathsf{A}}\,|D_{\mathcal{R}}\rangle_{\mathsf{D}_{\mathcal{R}}})=\begin{cases}|x\rangle_{\mathsf{Q}}\,|D_{\mathcal{R}}(x)\rangle_{\mathsf{A}}\,|D_{\mathcal{R}},(x,D_{\mathcal{R}}(x))\rangle_{\mathsf{D}_{\mathcal{R}}}\,, & x\in D_{\mathcal{R}}\\ |x\rangle_{\mathsf{Q}}\,\frac{1}{\sqrt{2}}\sum_{z=0}^{1}|z\rangle_{\mathsf{A}}\,|D_{\mathcal{R}},(x,z)\rangle_{\mathsf{D}_{\mathcal{R}}}\,, & x\notin D_{\mathcal{R}}\end{cases}$$

$\widetilde{\ }$ $\widetilde{\ }$ $\widetilde{\ }$定义 $|\phi_j\rangle=U_CU_{j-1}\cdots U_CU_0\,|\phi\rangle$，其中 $0\leq j\leq q$。为了分析 $|\phi\rangle$上 $U_{\mathsf{Ver}}\widetilde{\ }$ 与 U_{Sim} 的差异，只需分析一次真实查询与一次模拟查询之间的差异。形式化地，

$$\begin{aligned}&\mathrm{TD}\left(\mathrm{Tr}_{\mathsf{FD}_{\mathcal{A}}\mathsf{D}_{\mathcal{R}}}(\widetilde{U_{\mathsf{Ver}}}\,|\phi\rangle\langle\phi|\,\widetilde{U_{\mathsf{Ver}}}^{\dagger}),\mathrm{Tr}_{\mathsf{FD}_{\mathcal{A}}\mathsf{D}_{\mathcal{R}}}(\widetilde{U_{\mathsf{Sim}}}\,|\phi\rangle\langle\phi|\,\widetilde{U_{\mathsf{Sim}}}^{\dagger})\right)\\ =&\mathrm{TD}\left(\mathrm{Tr}_{\mathsf{FD}_{\mathcal{A}}\mathsf{D}_{\mathcal{R}}}(U_q\,|\phi_q\rangle\langle\phi_q|\,U_q^{\dagger}),\mathrm{Tr}_{\mathsf{FD}_{\mathcal{A}}\mathsf{D}_{\mathcal{R}}}(U_qU'_D\cdots U'_DU_0\,|\phi_0\rangle\langle\phi_0|\,U_0^{\dagger}U_D'^{\dagger}\cdots U_D'^{\dagger}U_q^{\dagger})\right)\\ \leq&\sum_{j=0}^{q-1}\mathrm{TD}\left(\mathrm{Tr}_{\mathsf{FD}_{\mathcal{A}}\mathsf{D}_{\mathcal{R}}}(U_qU'_D\cdots U_{j+1}\,|\phi_{j+1}\rangle\langle\phi_{j+1}|\,U_{j+1}^{\dagger}U_D'^{\dagger}\cdots U_q^{\dagger}),\mathrm{Tr}_{\mathsf{FD}_{\mathcal{A}}\mathsf{D}_{\mathcal{R}}}(U_qU'_D\cdots U_j\,|\phi_j\rangle\langle\phi_j|\,U_j^{\dagger}U_D'^{\dagger}\cdots U_q^{\dagger})\right)\\ \leq&\sum_{j=0}^{q-1}\mathrm{TD}\left(\widetilde{U_C}U_j\,|\phi_j\rangle\langle\phi_j|\,U_j^{\dagger}\widetilde{U_C}^{\dagger},U'_DU_j\,|\phi_j\rangle\langle\phi_j|\,U_j^{\dagger}U_D'^{\dagger}\right)\end{aligned}$$

其中我们利用了这样一个事实：$|\phi\rangle$ 在 $\mathsf{D}_{\mathcal{A}}$ 和 $\mathsf{D}_{\mathcal{R}}$ 上具有相同的内容，因此 $\mathbf{Tr}^{\mathsf{FD}_{\mathcal{A}}\mathsf{D}_{\mathcal{R}}}(\widetilde{U_{\mathsf{Sim}}}|\phi\rangle\langle\phi|\widetilde{U_{\mathsf{Sim}}}^{\dagger})$ 等于 $\mathbf{Tr}^{\mathsf{FD}_{\mathcal{A}}\mathsf{D}_{\mathcal{R}}}(U_qU'_D\cdots U'_DU_0|\phi_0\rangle\langle\phi_0|U_0^{\dagger}U_D'^{\dagger}\cdots U_D'^{\dagger}U_q^{\dagger})$。$\sim$

U_C和U'_D仅当查询位置位于 D_F内部时才会有不同。因此，一次真实查询与一次模拟查询之间的差异，可以由位于 D_F内部的查询权重来界定；该权重等于查询后 F 中对数的减少量。形式化地，我们给出以下引理，其证明推迟至 附录 A。

Lemma 6. $\mathrm{TD}\left(\widetilde{U_C}U_j|\phi_j\rangle\langle\phi_j|U_j^{\dagger}\widetilde{U_C}^{\dagger},U'_DU_j|\phi_j\rangle\langle\phi_j|U_j^{\dagger}U_D'^{\dagger}\right)\leq 6\sqrt{\mathrm{Tr}(O|\phi_j\rangle\langle\phi_j|)-\mathrm{Tr}(O|\phi_{j+1}\rangle\langle\phi_{j+1}|)}.$

将 引理 6 代入上述不等式，可得

$$
\begin{aligned}
&\mathrm{TD}\left(\mathrm{Tr}_{\mathsf{FD}_{\mathcal{A}}\mathsf{D}_{\mathcal{R}}}(\widetilde{U_{\mathsf{Ver}}}|\phi\rangle\langle\phi|\widetilde{U_{\mathsf{Ver}}}^{\dagger}),\mathrm{Tr}_{\mathsf{FD}_{\mathcal{A}}\mathsf{D}_{\mathcal{R}}}(\widetilde{U_{\mathsf{Sim}}}|\phi\rangle\langle\phi|\widetilde{U_{\mathsf{Sim}}}^{\dagger})\right)\\
\leq&\sum_{j=0}^{q-1}6\sqrt{\mathrm{Tr}(O|\phi_j\rangle\langle\phi_j|)-\mathrm{Tr}(O|\phi_{j+1}\rangle\langle\phi_{j+1}|)}\\
\leq&6\sqrt{q\sum_{j=0}^{q-1}\left(\mathrm{Tr}(O|\phi_j\rangle\langle\phi_j|)-\mathrm{Tr}(O|\phi_{j+1}\rangle\langle\phi_{j+1}|)\right)}\ \text{(Cauchy–Schwarz inequality)}\\
=&6\sqrt{q\left(\mathrm{Tr}(O|\phi\rangle\langle\phi|)-\mathrm{Tr}(O\widetilde{U_C}U_{q-1}\cdots\widetilde{U_C}U_0|\phi\rangle\langle\phi|U_0^{\dagger}\widetilde{U_C}^{\dagger}\cdots U_{q-1}^{\dagger}\widetilde{U_C}^{\dagger})\right)}\\
=&6\sqrt{q\left(\mathrm{Tr}(O|\phi\rangle\langle\phi|)-\mathrm{Tr}(OU_q\widetilde{U_C}\cdots\widetilde{U_C}U_0|\phi\rangle\langle\phi|U_0^{\dagger}\widetilde{U_C}^{\dagger}\cdots\widetilde{U_C}^{\dagger}U_q^{\dagger})\right)}\\
=&6\sqrt{q\left(\mathrm{Tr}(O|\phi\rangle\langle\phi|)-\mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}|\phi\rangle\langle\phi|\widetilde{U_{\mathsf{Ver}}}^{\dagger})\right)}
\end{aligned}
$$

其中 我们再次利用如下事实：U_q 不作用于 F，因此 O 与 U_q可交换 .

□

下一个主张是 主张 4的一个类似版本。基本上，它说明，即使 $\rho_s^{(t)}$和 D_j 均可对 KeyGen 和 Mint 进行量子查询，$\mathcal{R}$在使用数据库 D_j 进行模拟时，平均而言仍是一个良好的见证态。

证明的直觉如下：根据第 6.1 节，在 $\rho_s^{(t)}$ 上，$\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,(s,\cdot))$ 与 $\mathsf{Ver}^{D_j,|\mathsf{PSPACE}\rangle}(pk,(s,\cdot))$ 的接受概率之差，等于在同一个量子态上于纯化视图中运行 $\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}$ 和 $\mathsf{Ver}^{D_j,|\mathsf{PSPACE}\rangle}$ 的差异（即在同一个量子态上分别施加 U_{Ver} 和 U_{Sim} 的差异）。这可以转换为压缩视图；根据命题 2，该差异可由验证后 F 中对数的减少量来界定。粗略地说，这一减少量等于验证期间在 D_F 中查询的对数。但是，我们对 t 进行随机化，因此，在 $\rho_s^{(t)}$ 上再运行一次验证，平均而言不应使 F 中的对数减少太多。形式化地说：

命题 7。

$$
\left|\Pr\left[\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,(s,\rho_s^{(t)}))\ accepts\right]-\Pr\left[\mathsf{Ver}^{D_j,|\mathsf{PSPACE}\rangle}(pk,(s,\rho_s^{(t)}))\ accepts\right]\right|\leq 6\sqrt{\frac{q(n)q'(n)}{T(n)}}
$$

其中的概率是针对 $\mathcal{R}$ 的随机性、对手输入的随机性以及对手自身的随机性取值的（因此 t 和 j 都是随机化的）。

证明。 根据 第 6.1 节，可以通过在纯化视图中运行算法序列来分析这两个概率。由于 $\mathsf{CompDecomp} = I$，我们可以在这些算法之间插入 Comp 和 Decomp 。因此，也可以通过在压缩视图中运行算法序列来分析这两个概率。

令 $|\phi\rangle$ 为通过应用酉算子所得到的整个纯态 $U_{\mathsf{KeyGen}} = \mathsf{Comp}U_{\mathsf{KeyGen}}\mathsf{Decomp}$，$\widetilde{U_{\mathsf{Mint}}} = \mathsf{Comp}U_{\mathsf{Mint}}\mathsf{Decomp}$ 以及 $U_{\mathcal{A}} = \mathsf{Comp}U_{\mathcal{A}}\mathsf{Decomp}$ 作用于量子态 $|1^n\rangle|\emptyset\rangle_{\mathsf{D}_{\mathcal{A}}}|\emptyset\rangle_{\mathsf{F}}|\emptyset\rangle_{\mathsf{D}_{\mathcal{R}}}$ 并配上足够多的辅助量子比特（即“$\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,\ (s,\ \cdot))$”中第一步的压缩版本，作用于 $\rho_s^{(t)}$”中第 6.1 节）。令 M 为对应于 $\rho_s^{(t)}$的寄存器。

不难看出，对手记录了每个经典查询。也就是说，$|\phi\rangle$ 在 $\mathsf{D}_{\mathcal{A}}$ 和 $\mathsf{D}_{\mathcal{R}}$中具有相同的内容。根据 命题 2，

$$\begin{aligned}&\left|\Pr\left[\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,(s,\rho_s^{(t)}))\ \text{accepts}\right]-\Pr\left[\mathsf{Ver}^{D_j,|\mathsf{PSPACE}\rangle}(pk,(s,\rho_s^{(t)}))\ \text{accepts}\right]\right|\\ \le&6\sqrt{q\left(\mathrm{Tr}(O|\phi\rangle\langle\phi|)-\mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}|\phi\rangle\langle\phi|\widetilde{U_{\mathsf{Ver}}}^\dagger)\right)}\end{aligned}$$

$\widetilde{U_{\mathsf{Upd}}}$ 将 U_{Upd} 记为描述我们在压缩视图中更新阶段的酉算子。形式化地，$\widetilde{U_{\mathsf{Upd}}} = \mathsf{Comp}U_{\mathsf{Upd}}\mathsf{Decomp} = \sum_{j=0}^{N(n)-1}(\widetilde{U'_{\mathsf{Ver}}}\widetilde{U_{\mathsf{Syn}}})^j \otimes |j\rangle\langle j|$，其中 $\widetilde{U'_{\mathsf{Ver}}}$ 作用于由 U_{Syn}合成的量子态。

令 $|\psi\rangle$ 表示通过对量子态 $|1^n\rangle|\emptyset\rangle_{\mathsf{D}_{\mathcal{A}}}|\emptyset\rangle_{\mathsf{F}}|\emptyset\rangle_{\mathsf{D}_{\mathcal{R}}}\frac{1}{\sqrt{T(n)}}\sum_{t=0}^{T(n)-1}|t\rangle_{\mathsf{T}}\frac{1}{\sqrt{N(n)}}\sum_{j=0}^{N(n)-1}|j\rangle_{\mathsf{J}}$ 以及足够多的辅助量子比特应用酉算子 U_{KeyGen}、U_{Mint} 和 $U_{\mathsf{Test}} := \mathsf{Comp}U_{\mathsf{Test}}\mathsf{Decomp}$ 所得到的整个纯态（即运行“$\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,(s,\cdot))$”的第一步的压缩版本，直到 $\mathcal{A}$的**测试**阶段结束）。也就是说，$|\phi\rangle = U_{\mathsf{Syn}_2}U_{\mathsf{Syn}_1}\widetilde{U_{\mathsf{Upd}}}|\psi\rangle$ 其中 U_{Syn_i} 合成声称的纸币。注意，U_{Syn_i} 只读取公钥和序列号$\widetilde{\ }$，并作用于 $\mathsf{D}_{\mathcal{A}}$ 及一些新的辅助量子比特。因此，U_{Syn_i} 与 O 和 U_{Ver}可交换。（回想一下，U_{Ver} 不会为 $\mathcal{A}$记录查询。也就是说，它不作用于 $\mathsf{D}_{\mathcal{A}}$。）因此

$$\begin{aligned}&\mathrm{Tr}(O|\phi\rangle\langle\phi|)-\mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}|\phi\rangle\langle\phi|\widetilde{U_{\mathsf{Ver}}}^\dagger)\\ =&\mathrm{Tr}(O\widetilde{U_{\mathsf{Syn}_2}}\widetilde{U_{\mathsf{Syn}_1}}\widetilde{U_{\mathsf{Upd}}}|\psi\rangle\langle\psi|\widetilde{U_{\mathsf{Upd}}}^\dagger\widetilde{U_{\mathsf{Syn}_1}}^\dagger\widetilde{U_{\mathsf{Syn}_2}}^\dagger)-\mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}\widetilde{U_{\mathsf{Syn}_2}}\widetilde{U_{\mathsf{Syn}_1}}\widetilde{U_{\mathsf{Upd}}}|\psi\rangle\langle\psi|\widetilde{U_{\mathsf{Upd}}}^\dagger\widetilde{U_{\mathsf{Syn}_1}}^\dagger\widetilde{U_{\mathsf{Syn}_2}}^\dagger\widetilde{U_{\mathsf{Ver}}}^\dagger)\\ =&\mathrm{Tr}(O\widetilde{U_{\mathsf{Upd}}}|\psi\rangle\langle\psi|\widetilde{U_{\mathsf{Upd}}}^\dagger)-\mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}\widetilde{U_{\mathsf{Upd}}}|\psi\rangle\langle\psi|\widetilde{U_{\mathsf{Upd}}}^\dagger\widetilde{U_{\mathsf{Ver}}}^\dagger)\end{aligned}$$

接下来，我们要界定在 $\widetilde{U_{\mathsf{Upd}}}|\psi\rangle$ 上进行验证时，F 中对的减少量。然而，U_{Ver} 与测试阶段中随机数量的验证之间的更新阶段可能会带来一些麻烦。因此，我们给出以下引理，它对应于经典情形下的事实：在一次验证期间，$D_{\mathsf{Mint}} \cup D_{\mathsf{KeyGen}} - D_k$ 中的查询数量不超过 $D_{\mathsf{Mint}} \cup D_{\mathsf{KeyGen}} - D$ 中的查询数量。

引理 7. 我们使用与上文相同的记号。那么

$$\begin{aligned}&\mathrm{Tr}(O\widetilde{U_{\mathsf{Upd}}}|\psi\rangle\langle\psi|\widetilde{U_{\mathsf{Upd}}}^\dagger)-\mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}\widetilde{U_{\mathsf{Upd}}}|\psi\rangle\langle\psi|\widetilde{U_{\mathsf{Upd}}}^\dagger\widetilde{U_{\mathsf{Ver}}}^\dagger)\\ \le&\mathrm{Tr}(O|\psi\rangle\langle\psi|)-\mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}|\psi\rangle\langle\psi|\widetilde{U_{\mathsf{Ver}}}^\dagger)\end{aligned}$$

该引理的证明见附录 B。

现在我们来界定验证过程中 F 中的对数减少量。根据引理 7，

$$\begin{aligned}
&\left|\Pr\left[\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,(s,\rho_s^{(t)}))\text{ accepts}\right]-\Pr\left[\mathsf{Ver}^{D_j,|\mathsf{PSPACE}\rangle}(pk,(s,\rho_s^{(t)}))\text{ accepts}\right]\right| \\
\leq& 6\sqrt{q\left(\mathrm{Tr}(O\widetilde{U_{\mathsf{Upd}}}\left|\psi\right\rangle\left\langle\psi\right|\widetilde{U_{\mathsf{Upd}}}^{\dagger})-\mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}\widetilde{U_{\mathsf{Upd}}}\left|\psi\right\rangle\left\langle\psi\right|\widetilde{U_{\mathsf{Upd}}}^{\dagger}\widetilde{U_{\mathsf{Ver}}}^{\dagger})\right)} \\
\leq& 6\sqrt{q\left(\mathrm{Tr}(O\left|\psi\right\rangle\left\langle\psi\right|)-\mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}\left|\psi\right\rangle\left\langle\psi\right|\widetilde{U_{\mathsf{Ver}}}^{\dagger})\right)} \\
=& 6\sqrt{q\left(\mathrm{Tr}(O\left|\psi\right\rangle\left\langle\psi\right|)-\mathrm{Tr}(O\widetilde{U'_{\mathsf{Ver}}}\left|\psi\right\rangle\left\langle\psi\right|\widetilde{U'_{\mathsf{Ver}}}^{\dagger})\right)}
\end{aligned}$$

这里我们利用了这样一个事实：$\widetilde{U'_{\mathsf{Ver}}}$ 与 $\widetilde{U_{\mathsf{Ver}}}$ 相同，只不过它会记录 $\mathcal{A}$的查询-答案对。

注意，我们也可以将 $\left|\psi\right\rangle$ 写作 $\frac{1}{\sqrt{T(n)}}\sum_{t=0}^{T(n)-1}\left|\psi^{(t)}\right\rangle\left|t\right\rangle_{\mathsf{T}}$，其中 $\left|\psi^{(t)}\right\rangle$ 是我们在测试阶段运行 t 次迭代后得到的量子态。于是 $\left|\psi^{(t+1)}\right\rangle=\widetilde{U'_{\mathsf{Ver}}}\left|\psi^{(t)}\right\rangle$。由于 O 和 $\widetilde{U'_{\mathsf{Ver}}}$ 不会作用于 T，对 T 取偏迹不会改变该量。因此，

$$\begin{aligned}
&\left|\Pr\left[\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,(s,\rho_s^{(t)}))\text{ accepts}\right]-\Pr\left[\mathsf{Ver}^{D_j,|\mathsf{PSPACE}\rangle}(pk,(s,\rho_s^{(t)}))\text{ accepts}\right]\right| \\
\leq& 6\sqrt{q\left(\mathrm{Tr}(O\left|\psi\right\rangle\left\langle\psi\right|)-\mathrm{Tr}(O\widetilde{U'_{\mathsf{Ver}}}\left|\psi\right\rangle\left\langle\psi\right|\widetilde{U'_{\mathsf{Ver}}}^{\dagger})\right)} \\
=& 6\sqrt{q\left(\mathrm{Tr}(O\mathrm{Tr}_{\mathsf{T}}(\left|\psi\right\rangle\left\langle\psi\right|))-\mathrm{Tr}(O\widetilde{U'_{\mathsf{Ver}}}\mathrm{Tr}_{\mathsf{T}}(\left|\psi\right\rangle\left\langle\psi\right|)\widetilde{U'_{\mathsf{Ver}}}^{\dagger})\right)} \\
=& 6\sqrt{q\left(\frac{1}{T(n)}\sum_{t=0}^{T(n)-1}\mathrm{Tr}\left(O\left|\psi^{(t)}\right\rangle\left\langle\psi^{(t)}\right|\right)-\frac{1}{T(n)}\sum_{t=0}^{T(n)-1}\mathrm{Tr}\left(O\widetilde{U'_{\mathsf{Ver}}}\left|\psi^{(t)}\right\rangle\left\langle\psi^{(t)}\right|\widetilde{U'_{\mathsf{Ver}}}^{\dagger}\right)\right)} \\
=& 6\sqrt{q\left(\frac{1}{T(n)}\sum_{t=0}^{T(n)-1}\mathrm{Tr}\left(O\left|\psi^{(t)}\right\rangle\left\langle\psi^{(t)}\right|\right)-\frac{1}{T(n)}\sum_{t=0}^{T(n)-1}\mathrm{Tr}\left(O\left|\psi^{(t+1)}\right\rangle\left\langle\psi^{(t+1)}\right|\right)\right)} \\
\leq& 6\sqrt{\frac{q}{T(n)}\mathrm{Tr}\left(O\left|\psi^{(0)}\right\rangle\left\langle\psi^{(0)}\right|\right)} \\
\leq& 6\sqrt{\frac{qq'}{T(n)}}
\end{aligned}$$

这里我们利用了压缩预言机技术的性质：经过 $q'(n)$ 次量子查询后，F中至多有 $q'(n)$ 个非$\hat{0}$ 元素。$\left|\psi^{(0)}\right\rangle$ 正是我们运行 KeyGen 和Mint （因此至多进行 $q'(n)$ 次量子查询）后，再应用酉算子 Comp所得到的量子态。因此，$\left|\psi^{(0)}\right\rangle$的 F 中至多有 $q'(n)$ 对。于是 $\mathrm{Tr}\left(O\left|\psi^{(0)}\right\rangle\left\langle\psi^{(0)}\right|\right)\leq q'(n)$。

□

下一个命题类似于对 k 的命题 6求和，从而得到$\mathsf{Ver}^{D_j,\ |\mathsf{PSPACE}\rangle}$ 在σ_{D_j}上的表现如何的界。证明的直觉与命题 7类似。

命题 8。

$$\left|\Pr\left[\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,(s,\sigma_{D_j}))\ accepts\right]-\Pr\left[\mathsf{Ver}^{D_j,|\mathsf{PSPACE}\rangle}(pk,(s,\sigma_{D_j}))\ accepts\right]\right|\leq 6\sqrt{\frac{q(n)q'(n)}{N(n)}}$$

其中，概率是对 $\mathcal{R}$ 的随机性、对手输入的随机性以及对手本身的随机性取的（因此 t 和 j 都是随机化的）。

证明。与命题 7 的证明类似，可以通过在压缩视图中运行第 6.1 节中指定的一系列算法来分析这两个概率。令 $|\phi\rangle$ 为与命题 7 中相同的量子态，只是现在 M 是 $\mathcal{A}$ 的第一个输出寄存器（对应于第一个合成态 σ_{D_j}）。令 $|\psi\rangle$ 为我们通过在 $\rho_s^{(t)}$ 上运行第 6.1 节中"$\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,(s,\cdot))$"第一步的压缩版本，直至 $\mathcal{A}$ 的更新阶段结束而得到的量子态。也就是说，$|\phi\rangle=U_{\mathsf{Syn}_2}U_{\mathsf{Syn}_1}|\psi\rangle$。那么，$|\phi\rangle$ 在 $\mathsf{D}_{\mathcal{A}}$ 和 $\mathsf{D}_{\mathcal{R}}$ 中的内容相同。

回想一下，$\widetilde{U_{\mathsf{Ver}}}$ 和 $\widetilde{U'_{\mathsf{Ver}}}$ 都对 M 中的量子态执行验证，即第一个合成态 σ_{D_j}。回想一下，U'_{Ver} 与 U_{Ver} 相同，只不过它还会记录 $\mathcal{A}$ 的查询-答案对，并且 O 只作用于 F。根据 命题 2，

$$\begin{aligned}
&\left|\Pr\left[\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,(s,\sigma_{D_j}))\ \text{accepts}\right]-\Pr\left[\mathsf{Ver}^{D_j,|\mathsf{PSPACE}\rangle}(pk,(s,\sigma_{D_j}))\ \text{accepts}\right]\right|\\
\leq&6\sqrt{q\left(\mathrm{Tr}(O|\phi\rangle\langle\phi|)-\mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}|\phi\rangle\langle\phi|\widetilde{U_{\mathsf{Ver}}}^{\dagger})\right)}\\
=&6\sqrt{q\left(\mathrm{Tr}(OU_{\mathsf{Syn}_2}U_{\mathsf{Syn}_1}|\psi\rangle\langle\psi|U_{\mathsf{Syn}_1}^{\dagger}U_{\mathsf{Syn}_2}^{\dagger})-\mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}U_{\mathsf{Syn}_2}U_{\mathsf{Syn}_1}|\psi\rangle\langle\psi|U_{\mathsf{Syn}_1}U_{\mathsf{Syn}_2}^{\dagger}\widetilde{U_{\mathsf{Ver}}}^{\dagger})\right)}\\
=&6\sqrt{q\left(\mathrm{Tr}(O|\psi\rangle\langle\psi|)-\mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}U_{\mathsf{Syn}_1}|\psi\rangle\langle\psi|U_{\mathsf{Syn}_1}^{\dagger}\widetilde{U_{\mathsf{Ver}}}^{\dagger})\right)}\\
=&6\sqrt{q\left(\mathrm{Tr}(O|\psi\rangle\langle\psi|)-\mathrm{Tr}(O\widetilde{U'_{\mathsf{Ver}}}U_{\mathsf{Syn}_1}|\psi\rangle\langle\psi|U_{\mathsf{Syn}_1}^{\dagger}\widetilde{U'_{\mathsf{Ver}}}^{\dagger})\right)}
\end{aligned}$$

$\widetilde{\ }$ 这是因为 U_{Syn_2} 作用于第二个输出寄存器，因此它与 U_{Ver} 对易（后者验证第一个输出量子态）。此外，U_{Syn_1} 和 U_{Syn_2} 都不作用于 F，因此与 O 对易。

注意，我们可以写作 $|\psi\rangle=\frac{1}{\sqrt{N(n)}}\sum_{j=0}^{N(n)-1}|\psi^{(j)}\rangle|j\rangle_{\mathsf{J}}$，其中 $|\psi^{(j)}\rangle$ 是我们在测试阶段执行随机次数的迭代、并在更新阶段执行 j 次迭代后得到的量子态。于是 $|\psi^{(j+1)}\rangle=\widetilde{U'_{\mathsf{Ver}}}U_{\mathsf{Syn}}|\psi^{(j)}\rangle$。注意，$O$、$U_{\mathsf{Syn}_1}$ 和 $\widetilde{U'_{\mathsf{Ver}}}$ 不会作用于 J。因此，对 J 取偏迹

不会改变其值。因此，

$$
\begin{aligned}
&\left|\mathsf{Pr}\left[\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,(s,\sigma_{D_j}))\text{ accepts}\right]-\mathsf{Pr}\left[\mathsf{Ver}^{D_j,|\mathsf{PSPACE}\rangle}(pk,(s,\sigma_{D_j}))\text{ accepts}\right]\right| \\
\leq & 6\sqrt{q\left(\mathrm{Tr}(O|\psi\rangle\langle\psi|)-\mathrm{Tr}(O\widetilde{U'_{\mathsf{Ver}}}U_{\mathsf{Syn}_1}|\psi\rangle\langle\psi|U^\dagger_{\mathsf{Syn}_1}\widetilde{U'_{\mathsf{Ver}}}^\dagger)\right)} \\
= & 6\sqrt{q\left(\mathrm{Tr}(O\mathrm{Tr}_{\mathsf{J}}(|\psi\rangle\langle\psi|))-\mathrm{Tr}(O\widetilde{U'_{\mathsf{Ver}}}U_{\mathsf{Syn}_1}\mathrm{Tr}_{\mathsf{J}}(|\psi\rangle\langle\psi|)U^\dagger_{\mathsf{Syn}_1}\widetilde{U'_{\mathsf{Ver}}}^\dagger)\right)} \\
= & 6\sqrt{q\left(\frac{1}{N(n)}\sum_{j=0}^{N(n)-1}\mathrm{Tr}\left(O|\psi^{(j)}\rangle\langle\psi^{(j)}|\right)-\frac{1}{N(n)}\sum_{j=0}^{N(n)-1}\mathrm{Tr}\left(O\widetilde{U'_{\mathsf{Ver}}}U_{\mathsf{Syn}_1}|\psi^{(j)}\rangle\langle\psi^{(j)}|U^\dagger_{\mathsf{Syn}_1}\widetilde{U'_{\mathsf{Ver}}}^\dagger\right)\right)} \\
= & 6\sqrt{q\left(\frac{1}{N(n)}\sum_{j=0}^{N(n)-1}\mathrm{Tr}\left(O|\psi^{(j)}\rangle\langle\psi^{(j)}|\right)-\frac{1}{N(n)}\sum_{j=0}^{N(n)-1}\mathrm{Tr}\left(O|\psi^{(j+1)}\rangle\langle\psi^{(j+1)}|\right)\right)} \\
\leq & 6\sqrt{\frac{q}{N(n)}\mathrm{Tr}\left(O|\psi^{(0)}\rangle\langle\psi^{(0)}|\right)} \\
\leq & 6\sqrt{\frac{qq'}{N(n)}}
\end{aligned}
$$

其中我们利用了压缩预言机技术的性质：经过 $q'(n)$ 次量子查询后， F 中至多有 $q'(n)$ 个非 $\hat{0}$ 元素。 $|\psi^{(0)}\rangle$ 正是我们运行 KeyGen 和 Mint （因此至多有 $q'(n)$ 次量子查询），并在压缩视图中运行 $\mathcal{A}$ 的测试阶段（仅进行经典查询 $\widetilde{\ }$ 并记录）后得到的量子态。根据 U_R 的描述，在进行经典查询并记录时， F 中的对数不会增加。因此，$|\psi^{(0)}\rangle$ 的 F 中至多有 $q'(n)$ 对。故 $\mathrm{Tr}\left(O|\psi^{(0)}\rangle\langle\psi^{(0)}|\right)\leq q'(n)$。 □

现在我们结合上述结果来证明定理 8。

证明。令 $\epsilon=0.01$, $b=1-\sqrt{1-\delta_r+\epsilon}$, $a=0.99b$, $T(n)=36q(n)q'(n)\epsilon^2$，且 $N(n)=q(n)q'(n)\epsilon^2(1-\sqrt{1-\delta_r+\epsilon})^4$。

由命题 7， $\mathsf{Pr}\left[\mathsf{Ver}^{D_j,|\mathsf{PSPACE}\rangle}(pk,(s,\rho_s^{(t)}))\text{ 接受}\right]\geq\delta_r-\epsilon$。因此，使用命题 5 中相同的论证，$\mathsf{Pr}\left[\mathsf{Ver}^{D_j,|\mathsf{PSPACE}\rangle}(pk,(s,\sigma_{D_j}))\text{ 接受}\right]\geq 0.99(1-\sqrt{1-\delta_r+\epsilon})^2$。由命题 8， $\mathsf{Pr}\left[\mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle}(pk,(s,\sigma_{D_j}))\text{ 接受}\right]\geq 0.99(1-\sqrt{1-\delta_r+\epsilon})^2-6\sqrt{\frac{q(n)q'(n)}{N}}\geq 0.99(1-\sqrt{1-\delta_r+\epsilon})^2$。因此，根据并集界，我们的对手的输出同时通过两次验证的概率至少为 $1.8(1-\sqrt{1-\delta_r+\epsilon})^2-1$；当 $\delta_r\geq 0.99$ 时，该概率不可忽略。 □

也就是说，我们构造的对手对 $(\mathsf{KeyGen}^{|\mathcal{R}\rangle,|\mathsf{PSPACE}\rangle},\ \mathsf{Mint}^{|\mathcal{R}\rangle,|\mathsf{PSPACE}\rangle},\ \mathsf{Ver}^{\mathcal{R},|\mathsf{PSPACE}\rangle})$ 给出了一个有效的攻击，其中 $\delta_r\geq 0.99$ 且 $\delta_s=\mathsf{negl}(n)$，由此证明了 定理 7。

参考文献

[AA09] Scott Aaronson 和 Andris Ambainis。“量子加速中对结构的需求”。载于：*arXiv* 预印本 *arXiv:0911.0996* (2009)（引文见第 3页）。

[Aar09] Scott Aaronson。“量子复制保护与量子货币”。载于：第24届 IEEE 计算复杂性年度会议。IEEE Computer Soc.，加利福尼亚州洛斯阿拉米托斯，2009年，第229–242页。doi：10.1109/CCC.2009.42。网址：https://doi.org/10. 1109/CCC.2009.42（引文见第1页）。[AC13] Scott Aaronson 和 Paul Christiano。“来自隐藏子空间的量子货币”。载于：Theory Comput. 9（2013），第349–401页。doi：10. 4086 / toc . 2013 . v009a009。网址：https://doi.org/10.4086/toc.2013.v009a009（引文见第2、12页）。[ACC+22] Per Austrin、Hao Chung、Kai-Min Chung、Shiuan Fu、Yao-Ting Lin 和 Moham-mad Mahmoody。“关于基于量子随机预言机的密钥协商不可能性”。载于：密码学电子印本档案（2022）（引文见第3、12页）。[AJL+19] Prabhanjan Ananth、Aayush Jain、Huijia Lin、Christian Matt 和 Amit Sahai。“无需多线性映射的不可区分混淆：通过低次弱伪随机性和安全性增强实现的新范式”。载于：国际密码学年会。Springer。2019年，第284–332页（引文见第2页）。[AK22] Prabhanjan Ananth 和 Fatih Kaleoglu。《关于来自随机预言机的复制保护的注记》。密码学电子印本档案，论文2022/1109。https://eprint.iacr.org/ 2022/1109。2022年。网址：https://eprint.iacr.org/2022/1109（引文见第5、8、9、25页）。[AKL+22] Prabhanjan Ananth、Fatih Kaleoglu、Xingjian Li、Qipeng Liu 和 Mark Zhandry。“关于不可克隆加密的可行性及其他问题”。载于：国际密码学年会。Springer。2022年，第212–241页（引文见第4页）。[AL21] Prabhanjan Ananth 和 Rolando L La Placa。“安全软件租赁”。载于：Eurocrypt（2021）（引文见第1页）。[BDG22] Andriyan Bilyk、Javad Doliskani 和 Zhiyong Gong。“三种量子货币方案的密码分析”。载于：arXiv 预印本 arXiv:2205.10488（2022）（引文见第2、12页）。[BDGM20] Zvika Brakerski、Nico Döttling、Sanjam Garg 和 Giulio Malavolta。“因数分解和配对并非 iO 所必需：循环安全学习带误差即已足够”。载于：密码学电子印本档案（2020）（引文见第2页）。[BDV17] Nir Bitansky、Akshay Degwekar 和 Vinod Vaikuntanathan。“通过混淆视角看结构与困难性”。载于：国际密码学年会。Springer。2017年，第696–723页（引文见第2页）。[BGI+01] Boaz Barak、Oded Goldreich、Rusell Impagliazzo、Steven Rudich、Amit Sahai、Salil Vadhan 和 Ke Yang。“关于程序混淆的（不）可能性”。载于：国际密码学年会。Springer。2001年，第1–18页（引文见第12页）。[BGS13] Anne Broadbent、Gus Gutoski 和 Douglas Stebila。“量子一次性程序”。载于：密码学年会。Springer。2013年，第344–360页（引文见第1页）。[BI20] Anne Broadbent 和 Rabib Islam。“带认证删除的量子加密”。载于：密码学理论会议。Springer。2020年，第92–122页（引文见第1页）。

[BL20] Anne Broadbent 和 Sébastien Lord。《通过预言机实现不可克隆量子加密》。载于：第 15 届量子计算、通信与密码学理论会议（TQC 2020）。Steven T. Flammia 编。第 158 卷。莱布尼茨国际信息学会议论文集（LIPIcs）。德国达格斯图尔：Schloss Dagstuhl–Leibniz-Zentrum für Informatik，2020，4:1–4:22。doi：10.4230/LIPIcs.TQC.2020.4（引文见第 1 页）。[BM09] Boaz Barak 和 Mohammad Mahmoody-Ghidary。《Merkle 谜题已达最优——针对任何基于随机预言机的密钥交换方案的 o (n 2) 次查询攻击》。载于：国际密码学年会。Springer。2009，第 374–390 页（引文见第 2 页）。[BPR+08] Dan Boneh、Periklis Papakonstantinou、Charles Rackoff、Yevgeniy Vahlis 和 Brent Waters。《基于陷门置换构造基于身份的加密方案的不可能性》。载于：2008 年第 49 届 IEEE计算机科学基础研讨会。IEEE。2008，第 283–292 页（引文见第 2 页）。[BS16] Shalev Ben-David 和 Or Sattath。《用于数字签名的量子令牌》。载于：arXiv 预印本 arXiv:1609.09047（2016）（引文见第 1 页）。[CKP15] Ran Canetti、Yael Tauman Kalai 和 Omer Paneth。《关于使用随机预言机进行混淆》。载于：密码学理论会议。Springer。2015，第 456–467 页（引文见第 8、25 页）。[CLLZ21] Andrea Coladangelo、Jiahui Liu、Qipeng Liu 和 Mark Zhandry。《隐藏陪集及其在不可克隆密码学中的应用》。载于：国际密码学年会。Springer。2021，第 556–584 页（引文见第 1、4 页）。[CMP20] Andrea Coladangelo、Christian Majenz 和 Alexander Poremba。《量子随机预言机模型中计算并比较程序的量子复制保护》。载于：arXiv 预印本 arXiv:2009.13865（2020）（引文见第 4 页）。[CMSZ22] Alessandro Chiesa、Fermi Ma、Nicholas Spooner 和 Mark Zhandry。《后量子简洁论证：突破量子回绕障碍》。载于：2021 年 IEEE 第 62 届计算机科学基础年会——FOCS 2021。IEEE Computer Soc.，加利福尼亚州洛斯阿拉米托斯， [2022] ©2022，第 49–58 页。doi：10.1109/FOCS52979.2021. 00014。网址：https://doi.org/10.1109/FOCS52979.2021.00014（引文见第 16、17 页）。[CX21] Shujiao Cao 和 Rui Xue。《在量子世界中，置换性也与单向性正交：无法从单向性原语构造量子单向置换》。载于：《理论计算机科学》855（2021），第 16–42 页（引文见第 12 页）。[DG17] Nico Döttling 和 Sanjam Garg。《基于 Diffie-Hellman 假设的基于身份的加密》。载于：国际密码学年会。Springer。2017，第 537–569 页（引文见第 2 页）。[Die82] DGBJ Dieks。《通过 EPR 设备进行通信》。载于：《物理快报A》92.6（1982），第 271–272 页（引文见第 1、3 页）。[DLMM11] Dana Dachman-Soled、Yehuda Lindell、Mohammad Mahmoody 和 Tal Malkin。《关于最优公平抛硬币的黑盒复杂度》。载于：密码学理论会议。Springer。2011，第 450–467 页（引文见第 2 页）。

[DQV+21] Lalita Devadas、Willy Quach、Vinod Vaikuntanathan、Hoeteck Wee 和 Daniel Wichs。《简洁 LWE 采样、随机多项式与混淆》。载于：密码学理论会议。Springer。2021，第 256–287 页（引文见第 2 页）。[FGH+12] Edward Farhi、David Gosset、Avinatan Hassidim、Andrew Lutomirski 和 Peter Shor。《由纽结构造量子货币》。载于：第 3 届理论计算机科学创新会议论文集。2012，第 276–289 页（引文见第 2、12 页）。[GKLM12] Vipul Goyal、Virendra Kumar、Satya Lokam 和 Mohammad Mahmoody。《关于谓词加密方案之间的黑盒归约》。载于：密码学理论会议。Springer。2012，第 440–457 页（引文见第 2 页）。[GKM+00] Yael Gertner、Sampath Kannan、Tal Malkin、Omer Reingold 和 Mahesh Viswanathan。《公钥加密与不经意传输之间的关系》。载于：第 41 届计算机科学基础年会论文集。IEEE。2000，第 325–335 页（引文见第 2 页）。[GP21] Romain Gay 和 Rafael Pass。《基于循环安全性的不可区分混淆》。载于：第 53 届 ACM SIGACT计算理论研讨会论文集。2021，第 736–749 页（引文见第 2 页）。[GZ20] Marios Georgiou 和 Mark Zhandry。《不可克隆解密密钥》。载于：密码学电子印本档案（2020）（引文见第 1 页）。[HJL21] Sam Hopkins、Aayush Jain 和 Huijia Lin。《对 iO 所依赖的新循环安全性假设的反例》。载于：国际密码学年会。Springer。2021，第 673–700 页（引文见第 2 页）。[HY20] Akinori Hosoyamada 和 Takashi Yamakawa。《在量子世界中寻找碰撞：抗碰撞性与单向性的量子黑盒分离》。载于：国际密码技术理论与应用会议。Springer。2020，第 3–32 页（引文见第 12 页）。[IR90] Russell Impagliazzo 和 Steven Rudich。《单向置换可证明推论的局限》。载于：密码学进展——CRYPTO ’88（加利福尼亚州圣巴巴拉，1988）。第 403 卷。《计算机科学讲义》。Springer，柏林，1990，第 8–26 页。doi：10.1007/0-387-34799-2_2。网址：https://doi.org/10.1007/0-387-34799-2_2（引文见第 3 页）。[JLS18] Zhengfeng Ji、Yi-Kai Liu 和 Fang Song。《伪随机量子态》。载于：国际密码学年会。Springer。2018，第 126–152 页（引文见第 12 页）。[JLS21] Aayush Jain、Huijia Lin 和 Amit Sahai。《基于良基假设的不可区分混淆》。载于：第 53 届 ACM SIGACT计算理论研讨会论文集。2021，第 60–73 页（引文见第 2 页）。[JLS22] Aayush Jain、Huijia Lin 和 Amit Sahai。《基于 LPN、DLIN 和 NC 中伪随机生成器的不可区分混淆》。载于：国际密码技术理论与应用年会。Springer。2022，第 670–699 页（引文见第 2 页）。[KLS22] Andrey Boris Khesin、Jonathan Z Lu 和 Peter W Shor。《基于随机格的可公开验证量子货币》。载于：arXiv 预印本 arXiv:2207.13135（2022）（引文见第 2、12 页）。

[KSS21] Daniel M Kane、Shahed Sharif 和 Alice Silverberg。“四元数代数中的量子货币”。载于：arXiv预印本 arXiv:2109.12643（2021）（见第 2、12 页）。[Lut10] Andrew Lutomirski。“针对 Wiesner量子货币的在线攻击”。载于：arXiv预印本 arXiv:1010.0256（2010）（见第 12 页）。[MVW12] Abel Molina、Thomas Vidick 和 John Watrous。“Wiesner量子货币的最优伪造攻击及其推广”。载于：量子计算、通信与密码学会议。Springer，2012 年，第 45–64 页（见第 12 页）。[MW05] Chris Marriott 和 John Watrous。“量子 Arthur–Merlin博弈”。载于：计算复杂性 14.2（2005），第 122–152 页（见第 6、16、17、20 页）。[NC00] Michael A. Nielsen 和 Isaac L. Chuang。《量子计算与量子信息》。剑桥大学出版社，剑桥，2000 年，第 xxvi+676 页。isbn：0-521-63235-8；0-521-63503-9（见第 13、14 页）。[Pap94] Christos H. Papadimitriou。《计算复杂性》。Addison-Wesley，1994 年（见第 14 页）。[PRV12] Periklis A Papakonstantinou、Charles W Rackoff 和 Yevgeniy Vahlis。“DDH困难群有多强大？”载于：密码学电子印本档案（2012）（见第 2 页）。[Rob21] Bhaskar Roberts。“量子闪电的安全性分析”。载于：国际密码技术理论与应用年会。Springer，2021 年，第 562–567 页（见第 2、12 页）。[RS22] Roy Radian 和 Or Sattath。“半量子货币”。载于：《密码学杂志》35.2（2022），第 1–70 页（见第 12 页）。[RTV04] Omer Reingold、Luca Trevisan 和 Salil Vadhan。“密码学原语之间的可归约性概念”。载于：密码学理论会议。Springer，2004 年，第 1–20 页（见第 2 页）。[Rud91] Steven Rudich。“交互在公钥密码系统中的应用”。载于：国际密码学会议。Springer，1991 年，第 242–251 页（见第 2 页）。[RY21] Gregory Rosenthal 和 Henry Yuen。“用于合成量子态和酉算子的交互式证明”。载于：arXiv预印本 arXiv:2108.07192（2021）（见第 6、7、15 页）。[Shm22a] Omri Shmueli。“具有经典银行的公钥量子货币”。载于：第 54 届 ACM SIGACT计算理论研讨会论文集。2022 年，第 790–803 页（见第 2、12 页）。[Shm22b] Omri Shmueli。“半量子令牌化签名”。载于：密码学电子印本档案（2022）（见第 2 页）。[Sim98] Daniel R Simon。“在单行道上寻找碰撞：安全哈希函数能否建立在一般性假设之上？”载于：国际密码技术理论与应用会议。Springer，1998 年，第 334–345 页（见第 2、12 页）。[Unr16] Dominique Unruh。“计算绑定的量子承诺”。载于：国际密码技术理论与应用年会。Springer，2016 年，第 497–527 页（见第 4 页）。

[Wie83] Stephen Wiesner。“共轭编码”。载于：ACM Sigact News 15.1（1983），第 78–88 页（见第 1、12 页）。[Win99] Andreas Winter。“量子信道的编码定理与强逆定理”。载于：IEEE 信息论汇刊 45.7（1999），第 2481–2485 页（见第 5、16 页）。[WW21] Hoeteck Wee 和 Daniel Wichs。“通过盲学习带误差采样实现候选混淆”。载于：国际密码技术理论与应用年会。Springer，2021 年，第 127–156 页（见第 2 页）。[WZ82] William K Wootters 和 Wojciech H Zurek。“单个量子不可克隆”。载于：《自然》299.5886（1982），第 802–803 页（见第 1、3 页）。[Zha15] Mark Zhandry。“关于量子碰撞问题和集合相等问题的注记”。载于：量子信息与计算 15.7-8（2015），第 557–567 页（见第 4 页）。[Zha18] Mark Zhandry。《如何记录量子查询及其在量子不可区分性中的应用》。密码学电子印本档案，论文 2018/276。https://eprint.iacr.org/2018/276。2018 年。url：https://eprint.iacr.org/2018/276（见第 11、17、32 页）。[Zha21] Mark Zhandry。“量子闪电绝不会两次击中同一量子态。或：来自密码学假设的量子货币”。载于：《密码学杂志》34.1（2021），论文编号 6，共 56 页。issn：0933-2790。doi：10.1007/s00145-020-09372-x。url：https://doi.org/10.1007/s00145-020-09372-x（见第 2、4、5、12 页）。

缺失的证明：引理 6

证明分为两部分。第一部分，我们将 TD $\left(\widetilde{U_C}U_j\left|\phi_j\right\rangle\!\left\langle\phi_j\right|U_j^\dagger\widetilde{U_C}^\dagger,\ U'_DU_j\left|\phi_j\right\rangle\!\left\langle\phi_j\right|U_j^\dagger U_D'^\dagger\right)$界定为 D_F 内查询的权重。第二部分，我们将证明该权重等于查询后 F 中对数的减少量。

证明。 先从一些记号开始第一部分。

对于 $0 \le j \le q-1$，U_j 准备了 $(j+1)^{th}$ 查询，因此我们可以将 $U_j\left|\phi_j\right\rangle$ 写成如下形式，

$$U_j\left|\phi_j\right\rangle = \sum_{\substack{x,D_F,D_{\mathcal{R}},h \\ \text{s.t. } D_F\cap D_{\mathcal{R}}=\emptyset}} \alpha_{x,D_F,D_{\mathcal{R}},h}\left|x\right\rangle_{\mathsf{Q}}\left|0\right\rangle_{\mathsf{A}}\left|D_F\right\rangle_{\mathsf{F}}\left|D_{\mathcal{R}}\right\rangle_{\mathsf{D}_{\mathcal{R}}}\left|h\right\rangle_{\mathsf{H}}$$

其中 Q 是下一个查询位置，A 表示查询答案，H 包含所有其他寄存器，包括公钥、序列号、$\mathsf{D}_{\mathcal{A}}$等。我们可以将这些项分为两类，一类是 $x \notin D_F$，另一类是 $x \in D_F$。也就是说，$U_j\left|\phi_j\right\rangle = \sqrt{1-\alpha}\left|u\right\rangle + \sqrt{\alpha}\left|v\right\rangle$ 其中 $\alpha := \sum_{x,\ D_F,\ D_{\mathcal{R}},\ h \text{使得}\ D_F\cap D_{\mathcal{R}}=\emptyset,\ x\in D_F} \left|\alpha_{x,\ D_F,\ D_{\mathcal{R}},\ h}\right|^2$ 是 D_F 内查询的权重，

$$\sqrt{1-\alpha}\left|u\right\rangle = \sum_{\substack{x,D_F,D_{\mathcal{R}},h \\ \text{s.t. } D_F\cap D_{\mathcal{R}}=\emptyset, x\notin D_F}} \alpha_{x,D_F,D_{\mathcal{R}},h}\left|x\right\rangle_{\mathsf{Q}}\left|0\right\rangle_{\mathsf{A}}\left|D_F\right\rangle_{\mathsf{F}}\left|D_{\mathcal{R}}\right\rangle_{\mathsf{D}_{\mathcal{R}}}\left|h\right\rangle_{\mathsf{H}}$$

$$\sqrt{\alpha}\left|v\right\rangle = \sum_{\substack{x,D_F,D_{\mathcal{R}},h \\ \text{s.t. } D_F\cap D_{\mathcal{R}}=\emptyset, x\in D_F}} \alpha_{x,D_F,D_{\mathcal{R}},h}\left|x\right\rangle_{\mathsf{Q}}\left|0\right\rangle_{\mathsf{A}}\left|D_F\right\rangle_{\mathsf{F}}\left|D_{\mathcal{R}}\right\rangle_{\mathsf{D}_{\mathcal{R}}}\left|h\right\rangle_{\mathsf{H}}$$

回顾一下，

$$\begin{aligned}&\widetilde{U_C}(|x\rangle_{\mathsf{Q}}|0\rangle_{\mathsf{A}}|D_F\rangle_{\mathsf{F}}|D_{\mathcal{R}}\rangle_{\mathsf{D}_{\mathcal{R}}})\\&=\begin{cases}|x\rangle_{\mathsf{Q}}|D_{\mathcal{R}}(x)\rangle_{\mathsf{A}}|D_F\rangle_{\mathsf{F}}|D_{\mathcal{R}},(x,D_{\mathcal{R}}(x))\rangle_{\mathsf{D}_{\mathcal{R}}} & x\in D_{\mathcal{R}}\\|x\rangle_{\mathsf{Q}}\frac{1}{\sqrt{2}}\sum_{z=0}^{1}|z\rangle_{\mathsf{A}}|D_F\rangle_{\mathsf{F}}|D_{\mathcal{R}},(x,z)\rangle_{\mathsf{D}_{\mathcal{R}}} & x\notin D_{\mathcal{R}},x\notin D_F\\|x\rangle_{\mathsf{Q}}\frac{1}{\sqrt{2}}\sum_{z=0}^{1}(-1)^{z\widehat{D_F(x)}}|z\rangle_{\mathsf{A}}|D_F-x\rangle_{\mathsf{F}}|D_{\mathcal{R}},(x,z)\rangle_{\mathsf{D}_{\mathcal{R}}} & x\notin D_{\mathcal{R}},x\in D_F\end{cases}\end{aligned}$$

~ ~因此，当 $x\notin D_F$、U_C和U'_D的作用完全相同时，结果是$U_C|u\rangle=U'_D|u\rangle$。因此，

$$\begin{aligned}&\mathrm{TD}\left(\widetilde{U_C}U_j|\phi_j\rangle\langle\phi_j|U_j^\dagger\widetilde{U_C}^\dagger,U'_DU_j|\phi_j\rangle\langle\phi_j|U_j^\dagger U_D'^\dagger\right)\\=&\left\|\widetilde{U_C}U_j|\phi_j\rangle\langle\phi_j|U_j^\dagger\widetilde{U_C}^\dagger-U'_DU_j|\phi_j\rangle\langle\phi_j|U_j^\dagger U_D'^\dagger\right\|_1\\=&\|\widetilde{U_C}\left((1-\alpha)|u\rangle\langle u|+\alpha|v\rangle\langle v|+\sqrt{\alpha(1-\alpha)}(|u\rangle\langle v|+|v\rangle\langle u|)\right)\widetilde{U_C}^\dagger\\&-U'_D\left((1-\alpha)|u\rangle\langle u|+\alpha|v\rangle\langle v|+\sqrt{\alpha(1-\alpha)}(|u\rangle\langle v|+|v\rangle\langle u|)\right)U_D'^\dagger\|_1\\=&\|\widetilde{U_C}\left(\alpha|v\rangle\langle v|+\sqrt{\alpha(1-\alpha)}(|u\rangle\langle v|+|v\rangle\langle u|)\right)\widetilde{U_C}^\dagger-U'_D\left(\alpha|v\rangle\langle v|+\sqrt{\alpha(1-\alpha)}(|u\rangle\langle v|+|v\rangle\langle u|)\right)U_D'^\dagger\|_1\\\le&2\left\|\alpha|v\rangle\langle v|+\sqrt{\alpha(1-\alpha)}(|u\rangle\langle v|+|v\rangle\langle u|)\right\|_1\\\le&2(\alpha+2\sqrt{\alpha(1-\alpha)})\\\le&6\sqrt{\alpha}\end{aligned}$$

其中我们使用了矩阵迹范数的性质 $\|A+B\|_1\le\|A\|_1+\|B\|_1$，$\left\|UAU^\dagger\right\|_1=\|A\|_1$和$\|A\|_1=\sum_i|\lambda_i|$如果 $A=\sum_i\lambda_i|s_i\rangle\langle v_i|$，其中 $|s_i\rangle$，$|v_i\rangle$是两个基。

现在我们来讨论第二部分。我们将证明，α 等于 F 在 $|\phi_j\rangle$ 和 $|\phi_{j+1}\rangle=U_CU_j|\phi_j\rangle$上成对的数量之差 ~。

首先，我们来计算 F on 状态 $|\phi_j\rangle$ 中的对数。

注意，U_j 不作用于 F，因此它与 O 对易。接着，$\mathrm{Tr}(O|\phi_j\rangle\langle\phi_j|)=\mathrm{Tr}(OU_j|\phi_j\rangle\langle\phi_j|U_j^\dagger)$。回想一下，

$$U_j|\phi_j\rangle=\sum_{\substack{x,D_F,D_{\mathcal{R}},h\\\text{s.t. }D_F\cap D_{\mathcal{R}}=\emptyset}}\alpha_{x,D_F,D_{\mathcal{R}},h}|x\rangle_{\mathsf{Q}}|0\rangle_{\mathsf{A}}|D_F\rangle_{\mathsf{F}}|D_{\mathcal{R}}\rangle_{\mathsf{D}_{\mathcal{R}}}|h\rangle_{\mathsf{H}}$$

在求和中，各项彼此正交。因此，$\mathsf{F}$of $U_j|\phi_j\rangle$ 中对的期望数量为$\sum_{\substack{x,\ D_F\ D_{\mathcal{R}}\ h\\D_F\cap D_{\mathcal{R}}=\emptyset}}|D_F|\,|\alpha_{x,\text{,s.t.},D_F,\ D_{\mathcal{R}},\ h}|^2$。所以

$$\mathrm{Tr}(O|\phi_j\rangle\langle\phi_j|)=\sum_{\substack{x,D_F,D_{\mathcal{R}},h\\\text{s.t. }D_F\cap D_{\mathcal{R}}=\emptyset}}|D_F|\,|\alpha_{x,D_F,D_{\mathcal{R}},h}|^2$$

接下来，让我们计算 F on 量子态 $|\phi_{j+1}\rangle$中的对数。

$\widetilde{}$根据 $|\phi_{j+1}\rangle$ 和 U_C的定义

$$
\begin{aligned}
&|\phi_{j+1}\rangle \\
=&\widetilde{U_C}U_j\,|\phi_j\rangle \\
=&\sum_{\substack{x,D_F,D_{\mathcal{R}},h \\ \text{s.t. } D_F\cap D_{\mathcal{R}}=\emptyset, x\in D_{\mathcal{R}}}} \alpha_{x,D_F,D_{\mathcal{R}},h}\,|x\rangle_{\mathsf{Q}}\,|D_{\mathcal{R}}(x)\rangle_{\mathsf{A}}\,|D_F\rangle_{\mathsf{F}}\,|D_{\mathcal{R}},(x,D_{\mathcal{R}}(x))\rangle_{\mathsf{D}_{\mathcal{R}}}\,|h\rangle_{\mathsf{H}} \\
&+\sum_{\substack{x,D_F,D_{\mathcal{R}},h \\ \text{s.t. } D_F\cap D_{\mathcal{R}}=\emptyset, x\notin D_{\mathcal{R}}, x\notin D_F}} \alpha_{x,D_F,D_{\mathcal{R}},h}\frac{1}{\sqrt{2}}\sum_{z=0}^{1}|x\rangle_{\mathsf{Q}}\,|z\rangle_{\mathsf{A}}\,|D_F\rangle_{\mathsf{F}}\,|D_{\mathcal{R}},(x,z)\rangle_{\mathsf{D}_{\mathcal{R}}}\,|h\rangle_{\mathsf{H}} \\
&+\sum_{\substack{x,D_F,D_{\mathcal{R}},h \\ \text{s.t. } D_F\cap D_{\mathcal{R}}=\emptyset, x\in D_F}} \alpha_{x,D_F,D_{\mathcal{R}},h}\frac{1}{\sqrt{2}}\sum_{z=0}^{1}(-1)^{z\widehat{D_F(x)}}\,|x\rangle_{\mathsf{Q}}\,|z\rangle_{\mathsf{A}}\,|D_F-x\rangle_{\mathsf{F}}\,|D_{\mathcal{R}},(x,z)\rangle_{\mathsf{D}_{\mathcal{R}}}\,|h\rangle_{\mathsf{H}}
\end{aligned}
$$

$\sim U_C$是一个酉算子，因此各项 $|x\rangle_{\mathsf{Q}}\,|D_{\mathcal{R}}(x)\rangle_{\mathsf{A}}\,|D_F\rangle_{\mathsf{F}}\,|D_{\mathcal{R}},(x,\ D_{\mathcal{R}}(x))\rangle_{\mathsf{D}_{\mathcal{R}}}\,|h\rangle_{\mathsf{H}}$其中 $D_F\cap D_{\mathcal{R}}=\emptyset$ 和 $x\in D_{\mathcal{R}}$，$\frac{1}{\sqrt{2}}\sum_{z=0}^{1}|x\rangle_{\mathsf{Q}}\,|z\rangle_{\mathsf{A}}\,|D_F\rangle_{\mathsf{F}}\,|D_{\mathcal{R}},(x,\ z)\rangle_{\mathsf{D}_{\mathcal{R}}}\,|h\rangle_{\mathsf{H}}$其中 $D_F\cap D_{\mathcal{R}}=\emptyset$和 $x\notin D_{\mathcal{R}}\cup D_F$，$\widehat{\ }$以及$\frac{1}{\sqrt{2}}\sum_{z=0}^{1}(-1)^{zD_F(x)}|x\rangle_{\mathsf{Q}}\,|z\rangle_{\mathsf{A}}\,|D_F-x\rangle_{\mathsf{F}}\,|D_{\mathcal{R}},(x,\ z)\rangle_{\mathsf{D}_{\mathcal{R}}}\,|h\rangle_{\mathsf{H}}$其中 $D_F\cap D_{\mathcal{R}}=\emptyset$和 $x\in D_F$在上述求和中彼此正交。

因此， F 中的 $|\phi_{j+1}\rangle$ 对的期望数量为

$$
\begin{aligned}
&\mathrm{Tr}(O\,|\phi_{j+1}\rangle\langle\phi_{j+1}|) \\
=&\sum_{\substack{x,D_F,D_{\mathcal{R}},h \\ \text{s.t. } D_F\cap D_{\mathcal{R}}=\emptyset, x\notin D_F}} |D_F|\,|\alpha_{x,D_F,D_{\mathcal{R}},h}|^2+\sum_{\substack{x,D_F,D_{\mathcal{R}},h \\ \text{s.t. } D_F\cap D_{\mathcal{R}}=\emptyset, x\in D_F}} |D_F-x|\,|\alpha_{x,D_F,D_{\mathcal{R}},h}|^2 \\
=&\sum_{\substack{x,D_F,D_{\mathcal{R}},h \\ \text{s.t. } D_F\cap D_{\mathcal{R}}=\emptyset}} |D_F|\,|\alpha_{x,D_F,D_{\mathcal{R}},h}|^2-\sum_{\substack{x,D_F,D_{\mathcal{R}},h \\ \text{s.t. } D_F\cap D_{\mathcal{R}}=\emptyset, x\in D_F}} |\alpha_{x,D_F,D_{\mathcal{R}},h}|^2 \\
=&\mathrm{Tr}(O\,|\phi_j\rangle\langle\phi_j|)-\alpha
\end{aligned}
$$

所以 $\alpha=\mathrm{Tr}(O\,|\phi_j\rangle\langle\phi_j|)-\mathrm{Tr}(O\,|\phi_{j+1}\rangle\langle\phi_{j+1}|)$。也就是说， D_F之外的查询权重等于查询后 F 中对的数量减少值。

结合以上两部分，我们可以得到

$$
\mathrm{TD}\left(\widetilde{U_C}U_j\,|\phi_j\rangle\langle\phi_j|\,U_j^\dagger\widetilde{U_C}^\dagger, U'_DU_j\,|\phi_j\rangle\langle\phi_j|\,U_j^\dagger U'^\dagger_D\right)\le 6\sqrt{\mathrm{Tr}(O\,|\phi_j\rangle\langle\phi_j|)-\mathrm{Tr}(O\,|\phi_{j+1}\rangle\langle\phi_{j+1}|)}.
$$

□

B 缺失证明引理 7

我们证明 引理7 首先证明，对于两个并行查询，我们可以移除其中一个而不降低其值，然后将结果推广到 $\widetilde{U_{\mathsf{Ver}}}$ 关于 $\rho_s^{(t)}$和 $\widetilde{U_{\mathsf{Upd}}}$ （每个都有多项式次查询）。

证明。设 U_R 作用于第一个查询位置寄存器 Q_1、第一个查询答案寄存器 A_1，$\mathsf{D}_{\mathcal{A}}$，F 和 $\mathsf{D}_{\mathcal{R}}$，而 U_C 作用于第二个查询位置寄存器 Q_2、第二个查询位置寄存器 A_2，F 和 $\mathsf{D}_{\mathcal{R}}$。

我们首先证明，对于任意量子态

$$|\varphi\rangle = \sum_{\substack{x_1,x_2,D_{\mathcal{A}},D_F,D_{\mathcal{R}},h \\ \text{s.t. } D_F\cap D_{\mathcal{R}}=\emptyset}} \alpha_{x_1,x_2,D_{\mathcal{A}},D_F,D_{\mathcal{R}},h} |x_1\rangle_{\mathsf{Q}_1} |0\rangle_{\mathsf{A}_1} |x_2\rangle_{\mathsf{Q}_2} |0\rangle_{\mathsf{A}_2} |D_{\mathcal{A}}\rangle_{\mathsf{D}_{\mathcal{A}}} |D_F\rangle_{\mathsf{F}} |D_{\mathcal{R}}\rangle_{\mathsf{D}_{\mathcal{R}}} |h\rangle_{\mathsf{H}},$$

我们都有如下不等式

$$\mathrm{Tr}(O\widetilde{U_R}|\varphi\rangle\langle\varphi|\widetilde{U_R}^\dagger) - \mathrm{Tr}(O\widetilde{U_C}\widetilde{U_R}|\varphi\rangle\langle\varphi|\widetilde{U_R}^\dagger\widetilde{U_C}^\dagger) \le \mathrm{Tr}(O|\varphi\rangle\langle\varphi|) - \mathrm{Tr}(O\widetilde{U_C}|\varphi\rangle\langle\varphi|\widetilde{U_C}^\dagger).$$

事实上，根据引理 6中的相同论证，$\mathrm{Tr}(O|\varphi\rangle\langle\varphi|) - \mathrm{Tr}(O\widetilde{U_C}|\varphi\rangle\langle\varphi|\widetilde{U_C}^\dagger)$ 恰好是我们获得结果 $(x_2,\ D_F)$ 的概率，其中 $x_2 \in D_F$ 当我们在量子态 $|\varphi\rangle$ 上测量寄存器 Q_2 和 D_F 时满足。也就是说

$$\mathrm{Tr}(O|\varphi\rangle\langle\varphi|) - \mathrm{Tr}(O\widetilde{U_C}|\varphi\rangle\langle\varphi|\widetilde{U_C}^\dagger) = \sum_{\substack{x_1,x_2,D_{\mathcal{A}},D_F,D_{\mathcal{R}},h \\ \text{s.t. } D_F\cap D_{\mathcal{R}}=\emptyset, x_2\in D_F}} |\alpha_{x_1,x_2,D_{\mathcal{A}},D_F,D_{\mathcal{R}},h}|^2.$$

类似地，$\mathrm{Tr}(O\widetilde{U_R}|\varphi\rangle\langle\varphi|\widetilde{U_R}^\dagger) - \mathrm{Tr}(O\widetilde{U_C}\widetilde{U_R}|\varphi\rangle\langle\varphi|\widetilde{U_R}^\dagger\widetilde{U_C}^\dagger)$ 恰好是我们获得结果 $(x_2,\ D_F)$ 的概率，其中 $x_2 \in D_F$ 当我们在量子态 $U_R|\varphi\rangle$ 上测量寄存器 Q_2 和 D_F 时满足。注意，我们可以将 $U_R|\varphi\rangle$ 写成

$$\sum_{\substack{x_1,x_2,D_{\mathcal{A}},D_F,D_{\mathcal{R}},h \\ \text{s.t. } D_F\cap D_{\mathcal{R}}=\emptyset}} \alpha_{x_1,x_2,D_{\mathcal{A}},D_F,D_{\mathcal{R}},h} |x_2\rangle_{\mathsf{Q}_2} |0\rangle_{\mathsf{A}_2} |x_1\rangle_{\mathsf{Q}_1} U_{x_1,D_F}(|0\rangle_{\mathsf{A}_1} |D_{\mathcal{A}}\rangle_{\mathsf{D}_{\mathcal{A}}} |D_{\mathcal{R}}\rangle_{\mathsf{D}_{\mathcal{R}}}) |D_F - x_1\rangle_{\mathsf{F}} |h\rangle_{\mathsf{H}}$$

其中 $U_{x_1,\ D_F}$ 是一个仅依赖于 x_1，D_F 的酉算子。U_R 是酉算子，因此上面的求和中的各项彼此正交。因此

$$\mathrm{Tr}(O\widetilde{U_R}|\varphi\rangle\langle\varphi|\widetilde{U_R}^\dagger) - \mathrm{Tr}(O\widetilde{U_C}\widetilde{U_R}|\varphi\rangle\langle\varphi|\widetilde{U_R}^\dagger\widetilde{U_C}^\dagger) = \sum_{\substack{x_1,x_2,D_{\mathcal{A}},D_F,D_{\mathcal{R}},h \\ \text{s.t. } D_F\cap D_{\mathcal{R}}=\emptyset, x_2\in D_F - x_1}} |\alpha_{x_1,x_2,D_{\mathcal{A}},D_F,D_{\mathcal{R}},h}|^2.$$

因此，$\mathrm{Tr}(O\widetilde{U_R}|\varphi\rangle\langle\varphi|\widetilde{U_R}^\dagger) - \mathrm{Tr}(O\widetilde{U_C}\widetilde{U_R}|\varphi\rangle\langle\varphi|\widetilde{U_R}^\dagger\widetilde{U_C}^\dagger) \le \mathrm{Tr}(O|\varphi\rangle\langle\varphi|) - \mathrm{Tr}(O\widetilde{U_C}|\varphi\rangle\langle\varphi|\widetilde{U_C}^\dagger)$。也就是说，在量子态的另一部分上进行额外查询 U_R，只会降低在 U_C 中进行错误查询的概率，因为该额外查询只会使错误查询集合变小。

U_{Ver} 且 U_{Upd} 由.....组成 U_C 以及 U_R。事实上，上述论证也可以扩展到 U_{Ver} 和 U_{Upd}，以体现我们的直觉：U_{Upd} 只能减少在 U_{Ver} 期间发出的错误查询数量，因为 U_{Upd} 只会使错误查询集合变小。

我们首先证明，固定次数的更新阶段只能减少在 U_{Ver} 期间发出的错误查询数量，然后证明这一结论对 U_{Upd} 也成立。

设 $|\psi\rangle$ 为量子态 $|\psi\rangle$，如 命题 7所述。我们可以将其写为 $|\psi_0\rangle \otimes \frac{1}{\sqrt{N(n)}} \sum_{j=0}^{N(n)-1} |j\rangle_{\mathsf{J}}$。

对于任意 $j \ge 1$，对于酉算子 U'_{Ver} 作用于合成态并记录针对 $\mathcal{A}$ 的查询，酉算子 U_{Ver} 作用于 $\rho_s^{(t)}$，记作 $|\mu\rangle = U_{\mathsf{Syn}}(U'_{\mathsf{Ver}}U_{\mathsf{Syn}})^{j-1}|\psi_0\rangle$。令 $U_{(1)i}$ 准备

下一个针对 $\rho_s^{(t)}$的查询，而 $U_i^{(2)}$ 为合成态准备下一个查询。对于每个 i，记作$U_{(1)i,\ C} = U_{(}$
$_i{}^1\ \widetilde{U_C}\cdots U_0^{(1))}$以及 $U_{(2)i,\ R} = U_{(2}$ $_i\ \widetilde{U_R}\cdots U^{)(2)0}$。应用上述不等式，我们可以得到

$$
\begin{aligned}
&\mathrm{Tr}(O(\widetilde{U'_{\mathsf{Ver}}}\widetilde{U_{\mathsf{Syn}}})^j\,|\psi_0\rangle\langle\psi_0|\,(\widetilde{U_{\mathsf{Syn}}}^\dagger\widetilde{U'_{\mathsf{Ver}}}^\dagger)^j) - \mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}(\widetilde{U'_{\mathsf{Ver}}}\widetilde{U_{\mathsf{Syn}}})^j\,|\psi_0\rangle\langle\psi_0|\,(\widetilde{U_{\mathsf{Syn}}}^\dagger\widetilde{U'_{\mathsf{Ver}}}^\dagger)^j\widetilde{U_{\mathsf{Ver}}}^\dagger)\\
=&\mathrm{Tr}(OU_{q,R}^{(2)}\,|\mu\rangle\langle\mu|\,U_{q,R}^{(2)\dagger}) - \mathrm{Tr}(OU_{q,C}^{(1)}U_{q,R}^{(2)}\,|\mu\rangle\langle\mu|\,U_{q,R}^{(2)\dagger}U_{q,C}^{(1)\dagger})\\
=&\mathrm{Tr}(OU_q^{(2)}\widetilde{U_R}U_{q-1,R}^{(2)}\,|\mu\rangle\langle\mu|\,U_{q-1,R}^{(2)}{}^\dagger\widetilde{U_R}^\dagger U_q^{(2)\dagger})\\
&-\mathrm{Tr}(OU_q^{(1)}U_q^{(2)}\widetilde{U_C}\widetilde{U_R}U_{q-1,C}^{(1)}U_{q-1,R}^{(2)}\,|\mu\rangle\langle\mu|\,U_{q-1,R}^{(2)}{}^\dagger U_{q-1,C}^{(1)}{}^\dagger\widetilde{U_R}^\dagger\widetilde{U_C}^\dagger U_q^{(2)\dagger}U_q^{(1)\dagger})\\
=&\mathrm{Tr}(O\widetilde{U_R}U_{q-1,R}^{(2)}\,|\mu\rangle\langle\mu|\,U_{q-1,R}^{(2)}{}^\dagger\widetilde{U_R}^\dagger) - \mathrm{Tr}(O\widetilde{U_C}\widetilde{U_R}U_{q-1,C}^{(1)}U_{q-1,R}^{(2)}\,|\mu\rangle\langle\mu|\,U_{q-1,R}^{(2)}{}^\dagger U_{q-1,C}^{(1)}{}^\dagger\widetilde{U_R}^\dagger\widetilde{U_C}^\dagger)\\
\le&\mathrm{Tr}(OU_{q-1,R}^{(2)}\,|\mu\rangle\langle\mu|\,U_{q-1,R}^{(2)}{}^\dagger) - \mathrm{Tr}(O\widetilde{U_C}U_{q-1,C}^{(1)}U_{q-1,R}^{(2)}\,|\mu\rangle\langle\mu|\,U_{q-1,R}^{(2)}{}^\dagger U_{q-1,C}^{(1)}{}^\dagger\widetilde{U_C}^\dagger)\\
=&\mathrm{Tr}(OU_{q-1,R}^{(2)}\,|\mu\rangle\langle\mu|\,U_{q-1,R}^{(2)}{}^\dagger) - \mathrm{Tr}(OU_{q,C}^{(1)}U_{q-1,R}^{(2)}\,|\mu\rangle\langle\mu|\,U_{q-1,R}^{(2)}{}^\dagger U_{q,C}^{(1)\dagger})
\end{aligned}
$$

也就是说，我们可以删除一个针对合成态的查询，而不减少在 $\widetilde{U_{\mathsf{Ver}}}$ on $\rho_s^{(t)}$期间发出的错误查询数量。反复删除针对合成态的查询，我们可以得到

$$
\begin{aligned}
&\mathrm{Tr}(O(\widetilde{U'_{\mathsf{Ver}}}\widetilde{U_{\mathsf{Syn}}})^j\,|\psi_0\rangle\langle\psi_0|\,(\widetilde{U_{\mathsf{Syn}}}^\dagger\widetilde{U'_{\mathsf{Ver}}}^\dagger)^j) - \mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}(\widetilde{U'_{\mathsf{Ver}}}\widetilde{U_{\mathsf{Syn}}})^j\,|\psi_0\rangle\langle\psi_0|\,(\widetilde{U_{\mathsf{Syn}}}^\dagger\widetilde{U'_{\mathsf{Ver}}}^\dagger)^j\widetilde{U_{\mathsf{Ver}}}^\dagger)\\
\le&\mathrm{Tr}(OU_{0,R}^{(2)}\,|\mu\rangle\langle\mu|\,U_{0,R}^{(2)\dagger}) - \mathrm{Tr}(OU_{q,C}^{(1)}U_{0,R}^{(2)}\,|\mu\rangle\langle\mu|\,U_{0,R}^{(2)}{}^\dagger U_{q,C}^{(1)\dagger})\\
=&\mathrm{Tr}(O\,|\mu\rangle\langle\mu|) - \mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}\,|\mu\rangle\langle\mu|\,\widetilde{U_{\mathsf{Ver}}}^\dagger)
\end{aligned}
$$

其中我们利用了以下事实： $U_{(1)i}$与 $U_{(2)j}$和 $\widetilde{U_C}$对易， $U_{(2)i}$与 $\widetilde{U_R}$对易，并且$U_{(1)i}$和 $U_j^{(2)}$都与 O对易。

⁀我们成功删除了一个针对合成态的 U'_{Ver}。我们可以持续这样做，直到删除针对合成态的所有⁀U'_{Ver}。也就是说，

$$
\begin{aligned}
&\mathrm{Tr}(O(\widetilde{U'_{\mathsf{Ver}}}\widetilde{U_{\mathsf{Syn}}})^j\,|\psi_0\rangle\langle\psi_0|\,(\widetilde{U_{\mathsf{Syn}}}^\dagger\widetilde{U'_{\mathsf{Ver}}}^\dagger)^j) - \mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}(\widetilde{U'_{\mathsf{Ver}}}\widetilde{U_{\mathsf{Syn}}})^j\,|\psi_0\rangle\langle\psi_0|\,(\widetilde{U_{\mathsf{Syn}}}^\dagger\widetilde{U'_{\mathsf{Ver}}}^\dagger)^j\widetilde{U_{\mathsf{Ver}}}^\dagger)\\
\le&\mathrm{Tr}(O\widetilde{U_{\mathsf{Syn}}}(\widetilde{U'_{\mathsf{Ver}}}\widetilde{U_{\mathsf{Syn}}})^{j-1}\,|\psi_0\rangle\langle\psi_0|\,(\widetilde{U_{\mathsf{Syn}}}^\dagger\widetilde{U'_{\mathsf{Ver}}}^\dagger)^{j-1}\widetilde{U_{\mathsf{Syn}}}^\dagger)\\
&-\mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}\widetilde{U_{\mathsf{Syn}}}(\widetilde{U'_{\mathsf{Ver}}}\widetilde{U_{\mathsf{Syn}}})^{j-1}\,|\psi_0\rangle\langle\psi_0|\,(\widetilde{U_{\mathsf{Syn}}}^\dagger\widetilde{U'_{\mathsf{Ver}}}^\dagger)^{j-1}\widetilde{U_{\mathsf{Syn}}}^\dagger\widetilde{U_{\mathsf{Ver}}}^\dagger)\\
=&\mathrm{Tr}(O(\widetilde{U'_{\mathsf{Ver}}}\widetilde{U_{\mathsf{Syn}}})^{j-1}\,|\psi_0\rangle\langle\psi_0|\,(\widetilde{U_{\mathsf{Syn}}}^\dagger\widetilde{U'_{\mathsf{Ver}}}^\dagger)^{j-1}) - \mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}(\widetilde{U'_{\mathsf{Ver}}}\widetilde{U_{\mathsf{Syn}}})^{j-1}\,|\psi_0\rangle\langle\psi_0|\,(\widetilde{U_{\mathsf{Syn}}}^\dagger\widetilde{U'_{\mathsf{Ver}}}^\dagger)^{j-1}\widetilde{U_{\mathsf{Ver}}}^\dagger)\\
\le&\cdots\\
\le&\mathrm{Tr}(O\,|\psi_0\rangle\langle\psi_0|) - \mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}\,|\psi_0\rangle\langle\psi_0|\,\widetilde{U_{\mathsf{Ver}}}^\dagger)
\end{aligned}
$$

$\widetilde{\ }$ $\widetilde{\ }$其中我们利用了以下事实： U_{Syn} 在 $D_{\mathcal{A}}$ 中使用数据库，而 U_{Ver} 不会记录针对 $\mathcal{A}$ 的查询。$\widetilde{\ }$ $\widetilde{\ }$因此， U_{Syn} 与 U_{Ver} 对易。 U_{Syn} 不作用于 F 和 $\mathsf{D}_{\mathcal{R}}$，所以它与 O对易。

最后，我们将证明，随机确定更新阶段的迭代次数，无法增加在 U_{Ver}期间产生的错误查询数量。由于 U_{Ver} 和 O 都不作用于寄存器 J，对 J做偏迹

之后，不会改变该量。回想一下，$\widetilde{U_{\mathsf{Upd}}}\left|\psi\right\rangle = \frac{1}{\sqrt{N(n)}}\sum_{j=0}^{N(n)-1}(\widetilde{U'_{\mathsf{Ver}}}\widetilde{U_{\mathsf{Syn}}})^j(\left|\psi_0\right\rangle)\ \left|j\right\rangle_{\mathsf{J}}$。因此

$$
\begin{aligned}
&\mathrm{Tr}(O\widetilde{U_{\mathsf{Upd}}}\left|\psi\right\rangle\left\langle\psi\right|\widetilde{U_{\mathsf{Upd}}}^\dagger) - \mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}\widetilde{U_{\mathsf{Upd}}}\left|\psi\right\rangle\left\langle\psi\right|\widetilde{U_{\mathsf{Upd}}}^\dagger\widetilde{U_{\mathsf{Ver}}}^\dagger)\\
=&\mathrm{Tr}(O\mathrm{Tr}_{\mathsf{J}}(\widetilde{U_{\mathsf{Upd}}}\left|\psi\right\rangle\left\langle\psi\right|\widetilde{U_{\mathsf{Upd}}}^\dagger)) - \mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}\mathrm{Tr}_{\mathsf{J}}(\widetilde{U_{\mathsf{Upd}}}\left|\psi\right\rangle\left\langle\psi\right|\widetilde{U_{\mathsf{Upd}}}^\dagger)\widetilde{U_{\mathsf{Ver}}}^\dagger)\\
=&\frac{1}{N(n)}\sum_{j=0}^{N(n)-1}\left(\mathrm{Tr}(O(\widetilde{U'_{\mathsf{Ver}}}\widetilde{U_{\mathsf{Syn}}})^j\left|\psi_0\right\rangle\left\langle\psi_0\right|(\widetilde{U_{\mathsf{Syn}}}^\dagger\widetilde{U'_{\mathsf{Ver}}}^\dagger)^j) - \mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}(\widetilde{U'_{\mathsf{Ver}}}\widetilde{U_{\mathsf{Syn}}})^j\left|\psi_0\right\rangle\left\langle\psi_0\right|(\widetilde{U_{\mathsf{Syn}}}^\dagger\widetilde{U'_{\mathsf{Ver}}}^\dagger)^j\widetilde{U_{\mathsf{Ver}}}^\dagger)\right)\\
\leq&\mathrm{Tr}(O\left|\psi_0\right\rangle\left\langle\psi_0\right|) - \mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}\left|\psi_0\right\rangle\left\langle\psi_0\right|\widetilde{U_{\mathsf{Ver}}}^\dagger)\\
=&\mathrm{Tr}(O\left|\psi\right\rangle\left\langle\psi\right|) - \mathrm{Tr}(O\widetilde{U_{\mathsf{Ver}}}\left|\psi\right\rangle\left\langle\psi\right|\widetilde{U_{\mathsf{Ver}}}^\dagger)
\end{aligned}
$$

□

C 符号表

表 1： 第 第 6.3 节

$q(n)$ $q'(n)$	n 是安全参数。 Ver 进行 $q(n)$ 次经典查询。 KeyGen 和 Mint 进行 $q'(n)$ 次量子查询。我们有时会省略 n。
$T(n)$ $N(n)$	两个多项式，用于确定我们执行的最大迭代次数在测试阶段和更新阶段。

表 2： 第 第 6.3 节

Pk	存储公钥的寄存器
S	存储序列号的寄存器
M	存储声称的货币状态的寄存器
$\mathsf{D}_{\mathcal{A}}$	由 $\mathcal{A}$ 维护的经典查询数据库寄存器
$\mathsf{D}_{\mathcal{R}}$	用于存储迄今为止我们发出的经典查询及其答案的寄存器（由预言机维护）
F	在解压视图中存储预言机的寄存器，或存储 D_F（用于存储非 $\left\|0\right\rangle$ 元素）的数据库（如果处于压缩视图中）
G, H	存储分析中无关内容的寄存器。例如，它可能包括秘密密钥、 KeyGen 和 Mint的工作空间，以及一些未使用的新的辅助量子比特。
T	存储测试阶段迭代次数的寄存器。
J	存储更新阶段迭代次数的寄存器。
Q, Q_1, Q_2	存储下一个查询位置的寄存器。
A, A_1, A_2	用于存储下一个查询答案的寄存器。

表 3： 第 第 6.3 节

$\|\phi\rangle$	以下形式的量子态（即它处于压缩视图中，且其中的内容 $\mathsf{D}_{\mathcal{R}}$ 和 $\mathsf{D}_{\mathcal{A}}$ 相同）， $\|\phi\rangle = \sum_{\substack{pk,s,m,D,D_F,g \\ \text{使得 } D\cap D_F=\emptyset}} \alpha_{pk,s,m,D,D_F,g} \|pk\rangle_{\mathsf{Pk}} \|s\rangle_{\mathsf{S}} \|m\rangle_{\mathsf{M}} \|D\rangle_{\mathsf{D}_{\mathcal{A}}} \|D_F\rangle_{\mathsf{F}} \|D\rangle_{\mathsf{D}_{\mathcal{R}}} \|g\rangle_{\mathsf{G}}$. 在命题 7和命题 8中，我们将用通过以下操作得到的纯态来实例化它 通过应用酉算子 $\widetilde{U_{\mathsf{KeyGen}}}$, $\widetilde{U_{\mathsf{Mint}}}$ 和 $\widetilde{U_{\mathcal{A}}}$ 到量子态 $\|1^n\rangle \|\emptyset\rangle \ _{\mathsf{D}_{\mathcal{A}}} \|\emptyset\rangle_{\mathsf{F}} \|\emptyset\rangle_{\mathsf{D}_{\mathcal{R}}}$ 以及足够多的辅助量子比特。
$\|\phi_j\rangle$	$\|\phi_j\rangle = \widetilde{U_C} U_{j-1} \cdots \widetilde{U_C} U_0 \|\phi\rangle$。这是在我们于 $\|\phi\rangle$ 上运行 $\mathsf{Ver}^{\mathcal{R},\|\mathsf{PSPACE}\rangle}$ 时的量子态，处于压缩视图中，直到我们回答了 j^{th} 查询。
$\|\psi\rangle$	我们滥用一下记号。 $\|\psi\rangle$ 是通过执行第一步得到的纯态 在压缩视图中，针对 $\mathsf{Ver}^{\mathcal{R},\|\mathsf{PSPACE}\rangle}(pk,(s,\cdot))$ 作用于 $\rho_s^{(t)}$ 的情形 直至结束 of 测试**阶段（见**命题 7）或更新阶段（见**命题 8）**，其对应于 $\mathcal{A}$。
$\|\varphi\rangle$	一个任意的量子态，已准备好在不同寄存器上进行两次“并行”经典查询。
$\|\psi_0\rangle$	在这种情况下，通过在压缩视图中执行第一步所得到的纯态 $\mathsf{Ver}^{\mathcal{R},\ \|\mathsf{PSPACE}\rangle}(pk,\ (s,\ \cdot))$ 在 $\rho_s^{(t)}$ 上 直到我们完成**测试**阶段，然后截断 J.
$\|\psi^{(t)}\rangle$, $\|\psi^{(j)}\rangle$	我们滥用一下记号。在命题 7中， $\|\psi^{(t)}\rangle$ 是我们运行 t 次迭代后得到的量子态，位于在压缩视图中执行测试阶段，但不执行更新阶段。在 命题 8， $\|\psi^{(j)}\rangle$ 是我们在中执行随机次数的迭代后获得的量子态 在压缩视图中执行测试阶段以及在更新阶段执行 j 次迭代后。

表 4： 第 6.3 节中的可观测量和酉算子

O	对应于 F 中对数的可观测量（即非空长度的一半）。形式上，$O = \sum_{D_F} \lvert D_F\rvert \lvert D_F\rangle\langle D_F\rvert_{\mathsf{F}}$ 其中 $\lvert D_F\rvert$ 是 D_F 中的对数。它只会作用于压缩视图中的那些量子态。
Decomp	第 6.2 节中定义的酉算子。它作用于 $\mathsf{D}_{\mathcal{R}}$ 和 F，并解压两个将数据库合并为一个数据库和预言机。
Comp	酉算子 Decomp 的逆。
$\widetilde{U}$	$\widetilde{U} = \mathsf{Comp}U\mathsf{Decomp}$。这是一般酉算子 U 的压缩视图版本 U。更多详情请参见 第 6.2 节 中的图。
U_Q	对应于使用真实预言机回答**量子**查询的酉算子。
U_C	对应于使用真实预言机回答**经典**查询的酉算子。
U_R	与 U_C 相同，但同时记录 $\mathcal{A}$ 的查询-答案。
U_D	对应于使用寄存器 $\mathsf{D}_{\mathcal{A}}$ 中的数据库回答经典查询，同时记录 $\mathsf{D}_{\mathcal{A}}$ 的查询-答案以供后续使用的酉算子。
U'_D	对应于使用寄存器 $\mathsf{D}_{\mathcal{R}}$ 中的数据库回答经典查询，同时记录 $\mathsf{D}_{\mathcal{R}}$ 的查询-答案以供后续使用的酉算子。
U_i	当 $0 \le i \le q-1$ 时，它是对应于制备以下对象的酉算子 $(i+1)^{th}$ 对 Ver 的查询。当 $i = q$ 时，这是对 Ver 的最终查询之后的酉算子。
U_{KeyGen}，U_{Mint}	酉算子 $U_{\mathsf{KeyGen},\,n}$，$U_{\mathsf{Mint},\,n}$ 定义于 第 6.1 节。
U_{Ver}，U_{Syn}	酉算子 $U_{\mathsf{Ver},\,n}$，$U_{\mathsf{Syn},\,n}$ 定义于 第 6.1 节，$U_{\mathsf{Ver}} = U_q U_C U_{q-1} \cdots U_C U_0$。
U'_{Ver}	执行验证并记录查询-答案对 $\mathcal{A}$、$U'_{\mathsf{Ver}} = U_q U_R U_{q-1} \cdots U_R U_0$。
U_{Sim}	运行 $\mathsf{Ver}^{D,\,\lvert\mathsf{PSPACE}\rangle}$ 其中 D 是 $\mathsf{D}_{\mathcal{A}}$ 中的内容，$U_{\mathsf{Sim}} = U_q U_D U_{q-1} \cdots U_D U_0$。
U_{Upd}	定义于 第 6.1 节、描述我们更新阶段的酉算子。形式上，它是酉算子 $\sum_{j=0}^{N(n)-1} (U_{\mathsf{Ver}} U_{\mathsf{Syn}})^j \lvert j\rangle\langle j\rvert_{\mathsf{J}}$。