How To Track Qubits Through Space and Time

(Or: Sailing in a Quantum Boat)

James Bartusek
Columbia

Zikuan Huang
University of Cambridge

Leo Orshansky
Columbia

Henry Yuen
Columbia

Abstract

While quantum position verification aims to certify a prover's location using quantum information, existing security definitions only guarantee that *part* of the successful adversarial party is in the claimed location. This leaves open the possibility that a distributed team of adversaries can jointly simulate a prover in a way that defeats the intended meaning of "being at a location" in position-based cryptography.

We introduce stronger notions of position verification that we call *quantum localization*, which requires that there is a specified, unclonable state at the verified spacetime point – and that this state can be found nowhere else. We show that quantum localization leads naturally to a meaningful notion of *trajectory* verification, in which quantum information is verifiably tracked through space and time. We construct quantum localization and trajectory verification protocols using *quantum anchor states*, which generalize coset states from unclonable cryptography. The security of our schemes is proven in the classical oracle (i.e. ideal obfuscation) model, which can be heuristically instantiated in the plain model using post-quantum indistinguishability obfuscation.

We also introduce and instantiate the concept of *functionality localization*, which guarantees that the adversary has the ability to compute a secret function at the verified spacetime point, and this function cannot be computed anywhere else. This raises the intriguing possibility of localizing computational capabilities in space and time.

More broadly, we believe our notions of quantum localization and subsequent feasibility results provide stronger foundations for position-based cryptography. At a conceptual level, our results also explore the tight link between two fundamental notions in physics – location and information – thus serving as a natural foundation for cryptographic capabilities tied to physical spacetime.

Contents

1 Introduction **3**
- 1.1 Entanglement Localization and Trajectory Verification . . . 4
- 1.2 State Localization . . . 7
- 1.3 Functionality Localization . . . 9
- 1.4 The Ideal Obfuscation Model . . . 10
- 1.5 Non-Localizability of f-BB84 . . . 11
- 1.6 Outlook and Future Directions . . . 12

2 Technical Overview **13**
- 2.1 Entanglement Localization and Trajectory Verification . . . 13
- 2.2 Functionality Localization . . . 19

3 Preliminaries **21**
- 3.1 General Notation . . . 21
- 3.2 Modeling Computations and Interactions in Spacetime . . . 21
- 3.3 Specifying Quantum Information in Spacetime . . . 23
- 3.4 Compressed Oracle . . . 24

4 The Quantum Anchor State **25**

5 Collapsing vs Non-collapsing Oracles **29**

6 Monogamy-of-Entanglement Games **32**
- 6.1 Asymmetric Monogamy-of-Entanglement Games . . . 33
- 6.2 Multi-Stage Monogamy-of-Entanglement Games . . . 43

7 Entanglement Extraction **49**
- 7.1 Extracting Incompatible Measurement Results via Collapsing Oracles . . . 49
- 7.2 Entanglement Rigidity in Multi-Stage Monogamy-of-Entanglement Games . . . 53

8 Warmup: Localizing Entanglement in the High-Success-Probability Regime **56**

9 Localizing Quantum Information **61**
- 9.1 Entanglement Localization . . . 61
- 9.2 Trajectory Verification . . . 65
- 9.3 State Localization . . . 68
- 9.4 Functionality Localization . . . 72

10 Discussion **82**
- 10.1 Publicly-Verifiable Localization . . . 82
  - 10.1.1 Applications to Black-Box Trajectory Verification . . . 84
- 10.2 Position-Based Cryptography . . . 84
  - 10.2.1 An explicit "attack" scenario . . . 85
  - 10.2.2 Towards stronger security notions . . . 86

1 Introduction

Position-based cryptography. Is it possible to base cryptographic identity on a party's *physical location*? This is in contrast with the typical cryptographic paradigm in which a party's identity, i.e. their *credential*, is tantamount to their knowledge of some secret information such as a password or private key. Instead, in *position-based cryptography*, it is a party's physical presence at an authorized location which serves as their credential [CGMO09]. The aim is to support tasks such as *position-based key exchange*, which seeks to establish secret communication with a party at an authorized location, and *position-based authentication*, wherein messages are signed with the guarantee of originating at an authorized location. A promising approach to establishing this type of positional credential is known as *quantum position verification* (QPV) [BCF+14, Unr14].

In a QPV protocol, verifiers attempt to verify the physical location of a party (called the *prover*) by sending challenges (which are, in general, quantum states) and receiving responses at precise times. Intuitively, the goal is to ensure that a successful prover, responding within some timing constraint, must be at a purported location L. As argued by [KMS11, CGMO09, BCF+14], secure position verification is impossible in the classical setting because a team of spoofers (all of whom are *not* in location L) can copy and forward messages to each other to simulate a fictitious prover in location L. On the other hand, the No-Cloning Theorem of quantum mechanics stymies such copy-and-forwarding attacks, and in fact secure QPV protocols have been established in the bounded-qubits model [TFKW13, BCS21, ACM25, ACCM25] and the random oracle model [Unr14]. Recently, QPV has been generalized to *privacy-preserving proofs of location* that only reveal part of a prover's location and nothing else [GGG+26]. For example, a user can prove to the police that they were somewhere far from the scene of a crime, without revealing any other information about their specific whereabouts. There have also recently been successful QPV experiments by multiple groups [KPB+25, KZG+26, FSZ+26], marking a new type of NISQ-era-implementable quantum advantage with many real-world applications.

Trajectory verification, and a conceptual gap. Despite the promise of QPV, closer examination reveals a gap between the intuitive goals of position-based cryptography and the formal notion of QPV security that has been considered thus far. In prior work, security of QPV was defined as follows: a team of spoofers $\{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$ can succeed in a QPV protocol with high probability only if *at least one* of the spoofers $\mathcal{P}_i$ is in the correct location L. This is a rather weak guarantee, however. For one thing, it should intuitively be possible to *attribute* a position-based credential to some party, who can then use it in downstream applications — but in QPV, a successful prover has only established that *somebody* exists at the given location, which lacks an attribution mechanism. To illustrate some of the issues stemming from this conceptual gap, we will introduce a natural generalization of QPV which we call *trajectory verification*.

In trajectory verification, we want to verify that a prover has traveled along a trajectory $L(\cdot)$ described as a function of time t. A natural approach is to run separate QPV protocols $\{\Pi_i\}_i$ for many spacetime points $\{(L(t_i), t_i)\}_i$, respectively, such that these points divide the trajectory into sufficiently small segments. An immediate problem is that the existing security guarantees of $\Pi_1, \Pi_2, \ldots$ do not distinguish between whether a *single* prover has traveled along the trajectory $L(\cdot)$ or whether an entire team $\{\mathcal{P}_1, \mathcal{P}_2, \ldots\}$ of provers participated in the protocol, where each prover $\mathcal{P}_i$ is stationed at location $L(t_i)$, only participating in protocol Π_i at time t_i. To rule out the latter scenario, one needs to explain why a line of stationary provers is fundamentally different from a single prover that is moving! QPV on its own is clearly ill-equipped to do this, since in both scenarios it is trivially true that *somebody* is at each location.

This work. The issues described above call for a notion of position verification in which a successful proof is attributable to some *persistent information* about the participating prover, rather than a simple binary stating that the location was occupied. Our work achieves this by formally introducing *localization*: pinpointing the behavior of an adversary who might *a priori* be distributed throughout space and time, to a persistent quantum state whose location is precisely known.

The main contributions of this paper are to formalize several notions of *quantum localization* – the aforementioned "pinpointing" – and establish their feasibility via constructions in the (classical) ideal obfuscation model,[1] which can be heuristically instantiated using post-quantum indistinguishability obfuscation of classical circuits. We then show that these notions enable meaningful notions of trajectory verification, yielding the first feasibility result for verifying the trajectory of an entity through spacetime.

We believe that our notions of localization come closer to capturing the concept of secure position verification. They give operational meaning to tracking the physical location of quantum states and computations, and lay the proper foundation for more complicated tasks such as trajectory verification.

1.1 Entanglement Localization and Trajectory Verification

We present stronger security notions for QPV that we call *quantum localization*, or *localization* for short. Informally, these security notions will guarantee that any successful prover strategy (which may involve a team of provers $\{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$ moving around) must contain a specific object at the correct point (L, t) in spacetime. By "contain," we mean that there exists a procedure called the *extractor* acting only on the prover's strategy at the point (L, t) that recovers the desired object. In this paper we identify three types of objects that can be localized: entanglement, (unclonable families of) quantum states, and (copy-protectable) functionalities.

We first describe *entanglement* localization and then describe how to build on entanglement localization in order to achieve a meaningful notion of *trajectory* verification.

Entanglement Localization. Let ψ be a bipartite entangled state on registers $\mathbf{A}$ and $\mathbf{B}$. Consider a QPV protocol where the verifiers generate ψ, keep register $\mathbf{A}$, and send register $\mathbf{B}$ to the prover who is purportedly at spacetime point (L, t).

We say that the protocol *localizes the entanglement in ψ at (L, t)* if for any (possibly nonlocal) prover strategy that is accepted with probability η, there exists an extractor $\mathcal{E}$ that acts on the quantum state at spacetime point (L, t) and achieves the following. With probability η, it outputs a register $\mathbf{B}$ that, together with the verifier's register $\mathbf{A}$, is close to being in the state ψ. In other words, it has recovered the entangled state ψ between register $\mathbf{A}$ and the spacetime point (L, t) with the same probability as the probability that the prover succeeds in the protocol. We formally define this in Definition 9.1.

Entanglement localization captures the idea that, in order to be successful, a team of provers *must* ensure that the quantum entanglement in the state ψ arrives at spacetime point (L, t). Suppose that ψ is a pure state that is maximally entangled across the $V : P$ cut. By monogamy of entanglement, the verifier's register $\mathbf{A}$ must then be unentangled with any other location $L' \neq L$ at time t.

[1] We provide more details about this model, which is sometimes referred to as the classical oracle model, in Section 1.4.

Trajectory Verification. The notion of entanglement localization gives rise to a natural definition of trajectory verification. Let $L(\cdot)$ denote a trajectory in spacetime. Intuitively, a trajectory verification protocol for $L(\cdot)$ is secure if there exists a maximally entangled state ψ such that entanglement localization can be performed with respect to ψ at spacetime points $(L(t), t)$ for all time t along the trajectory $L(\cdot)$. Importantly, the verifier's part of ψ remains at the same register V throughout the protocol, while the prover's part of ψ may travel around. This definition captures the idea of a *single* entity moving along the trajectory $L(\cdot)$ by tracking the movement of the entanglement ψ. If the prover is accepted by the protocol, monogamy of entanglement implies that ψ couldn't have strayed from the trajectory $L(\cdot)$.

We note that in order to repeatedly run an entanglement localization protocol on the same state, it must satisfy an additional completeness guarantee which we call *non-destructive*. That is, the prover's state must remain intact after interacting with the verifier. Therefore, our focus in this work is on building *non-destructive* versions of quantum localization.

Construction. We show how to construct secure trajectory verification in the ideal obfuscation model. In this model, the Setup samples the description of an efficient classical circuit, which all parties are given *black-box* access to throughout the protocol. Any protocol in this model can be heuristically instantiated in the plain model by applying a candidate post-quantum indistinguishability obfuscator to the classical circuit sampled by Setup.

Trajectory verification as (informally) defined above immediately yields a protocol for entanglement localization as a special case (where the trajectory is stationary at L). For the sake of exposition we first describe (a simplified version of) the entanglement localization protocol, and then describe how to extend it to obtain trajectory verification. Moreover, we focus here on the "high success probability" regime, where we only guarantee extraction success if the prover passes the protocol with probability close to 1.

Assume that space is one-dimensional and the location to be verified is the origin $L = 0$. Place verifiers $\mathcal{V}_L, \mathcal{V}_R$ at locations -1 and $+1$, respectively. We assume messages travel one unit of space per unit time.

The entangled state used in our protocol is what we call a *quantum anchor state*. Let $S \leq T \leq \mathbb{F}_2^{3n}$ be subspaces of dimension n and $2n$, respectively, and let $u, v \in \mathbb{F}_2^{3n}$ be some vectors. Let $\mathsf{CS} = \{x_1, x_2, \ldots\}$ denote some canonical set of 2^n coset representatives of S within T, indexed by $i \in [2^n]$. We define the state

$$|\Psi\rangle_{⚓🚢} = \frac{1}{\sqrt{2^n}} \sum_{i \in [2^n]} |i\rangle_{⚓} \otimes |\psi_i\rangle_{🚢}$$

where for each $i \in [2^n]$,

$$|\psi_i\rangle_{🚢} = \frac{1}{\sqrt{|S|}} \sum_{s \in S} (-1)^{\langle s,u \rangle} |s + x_i + v\rangle_{🚢} \ .$$

We use the icons ⚓ and 🚢 to denote the "anchor" and "vessel" registers, respectively. The reason for this naming will become apparent shortly.

Readers familiar with unclonable cryptography may recognize $|\psi_i\rangle$ as a *coset state*. That is, the state $|\Psi\rangle$ can be seen as a uniform superposition over a set of 2^n possible coset states, each defined with respect to subspace S, dual shift u, and primal shift $v + x_i$ for some choice of $x_i \in \mathsf{CS}$. Note further that $|\Psi\rangle$ is maximally entangled across the ⚓ : 🚢 cut with Schmidt rank 2^n.

In the Setup phase of the protocol (run any time before time $t = 0$), the verifiers generate the quantum anchor state $|\Psi\rangle$ corresponding to random subspaces $S \leq T \leq \mathbb{F}_2^{3n}$ and shifts $u, v \in \mathbb{F}_2^{3n}$. Furthermore, the verifiers prepare a classical oracle $\mathcal{O}$ (i.e., ideal obfuscation of some classical

functionality) which depends on S, T, u, v and whose behavior we describe shortly. The verifiers publish the oracle $\mathcal{O}$, which all parties (honest or adversarial) can access as a black box. They also release the vessel register 🚢 to the prover, while keeping hold of the entangled anchor register ⚓.

In the "online" phase of the protocol (at time $t = 0$), the verifiers sample random strings $a, b \in \{0,1\}^n$, the left verifier sends string a, and the right verifier sends string b. The honest prover with register 🚢 at location $L = 0$ gets a, b at time $t = 1$. Then,

1. The honest prover coherently queries the oracle $\mathcal{O}$ on input $|0, a \oplus b, z\rangle$ where $|z\rangle$ is the state of the vessel register 🚢 in the standard basis. The oracle, when the first bit of the input is 0, checks whether $z \in T + v$ (For the honest prover, this will be true.) and if so, outputs a string $r_0 = H(0, a \oplus b)$, where H is a random oracle.[2] Otherwise, the oracle outputs $\perp$.
2. The honest prover then queries the oracle $\mathcal{O}$ on input $|1, a \oplus b, w\rangle$ where $|w\rangle$ is the state of the vessel register 🚢 in the Hadamard basis. The oracle, when the first bit of the input is 1, checks whether $w \in S^{\perp} + u$ (which will be the case) and if so, outputs a string $r_1 = H(1, a \oplus b)$. Otherwise, the oracle outputs $\perp$.
3. The honest prover returns (r_0, r_1) back to the verifiers.

We note that after each query, the vessel register and the anchor state $|\Psi\rangle$ have not been disturbed.

The verifiers check that the responses came back by time $t = 2$, and that r_0, r_1 are indeed equal to $H(0, a \oplus b), H(1, a \oplus b)$.[3] The idea is that the only way for the prover to have "unlocked" r_0, r_1 is if it performed the honest strategy above in the correct location $L = 0$.

As mentioned above, this is the basic idea behind a protocol that allows us to extract from any prover that passes with probability close to 1. We formalize this in Theorem 8.4. Then, in Section 9.1, we apply an appropriate notion of sequential repetition in order to establish a more general result, informally stated as follows.

Theorem 1.1 (Entanglement localization (informal))**.** *Let (L, t) denote the spacetime point being verified. Sequentially repeating the protocol[4] above satisfies the following properties:*

- ***Completeness**: The honest prover strategy at spacetime point (L, t) is accepted by the verifiers with probability 1.*
- ***Extraction soundness**: Let $\mathcal{A}$ be any (possibly nonlocal) prover strategy that makes at most a polynomial number of queries to $\mathcal{O}$ and is accepted with probability at least $\eta = 1/\mathsf{poly}(\lambda)$. There exists an extractor $\mathcal{E}$ that takes as input the quantum state generated by the strategy at spacetime point (L, t) and, with probability roughly η, outputs a vessel register 🚢 such that the joint state of the anchor register ⚓ (held by the verifiers) and vessel register 🚢 has fidelity $1 - 1/\mathsf{poly}(\lambda)$ with the quantum anchor state $|\Psi\rangle$.*

We note that our extractor requires knowledge of S, T, u, v and CS. Indeed, we do not require any computational (or query) bound on the extractor for the above notion to be meaningful, since monogamy-of-entanglement is an information-theoretic property. However, we do show that the extractor is efficient if *given* a secret "extraction key" $\mathsf{ek} = (S, T, u, v, \mathsf{CS})$, which may be a useful feature, depending on the application.

[2] Technically, H will be implemented by a pseudorandom function so that the oracle $\mathcal{O}$ is efficiently computable.

[3] One way to check this is to have the verifiers remember a secret verification key $\mathsf{sk} = (u, v)$ consisting of the shifts and have them query $\mathcal{O}(a \oplus b, v)$ and $\mathcal{O}(a \oplus b, u)$.

[4] For technical reasons, our actual protocol is slightly more elaborate than this, where the standard- and Hadamard-basis queries are sequentially staggered (by a tiny time difference). We elaborate more on this in Section 2.

Given our entanglement localization protocol, the extension to trajectory verification is simple. Let $L(\cdot)$ be a physically realizable trajectory in spacetime that starts at time 0 and ends at τ, and the spatial points are all contained within the convex hull of the verifiers (i.e., strictly between $L = -1$ and $L = 1$). Discretize time $0 = t_1 < t_2 < \cdots < t_m = \tau$, and let $L(t_i)$ denote the spatial location of the trajectory at time t_i.

The trajectory verification protocol has the same Setup as the entanglement localization protocol. In the online phase of the protocol, the verifiers run m invocations of the entanglement localization protocol in sequence for spacetime points $(L(t_i), t_i)$, sending freshly sampled a_i, b_i each time. We can compose the entanglement localization guarantees of the QPV protocol above to obtain Theorem 9.6, informally stated as follows.

Theorem 1.2 (Trajectory verification (informal))**.** *The trajectory verification protocol above*[5] *satisfies the following properties:*

- ***Completeness****: An honest prover traversing the trajectory $L(\cdot)$ is accepted with probability 1.*
- ***Extraction soundness****: Let $\mathcal{A}$ be any (possibly nonlocal) prover strategy that makes at most a polynomial number of queries to $\mathcal{O}$ and is accepted with probability at least $\eta = 1/\mathsf{poly}(\lambda)$. There exists an extractor $\mathcal{E}$ that, for any $i \in [m]$, takes as input the quantum state generated by the strategy at spacetime point $(L(t_i), t_i)$ and, with probability roughly η, outputs a vessel register 🏴‍☠️ such that the joint state of the anchor register ⚓ (held by the verifiers) and vessel register 🏴‍☠️ has fidelity $1 - 1/\mathsf{poly}(\lambda)$ with the quantum anchor state $|\Psi\rangle$.*

At this point, we hope the "anchor" and "vessel" terminology should appear natural. The ⚓ part of the state $|\Psi\rangle$ is "anchored" at the verifier's location, while the 🏴‍☠️ part is launched into the spacetime region under the provers' control. The provers can *a priori* perform arbitrarily complex operations on the vessel, but in the end, if they are to pass the trajectory verification protocol, they must faithfully sail the quantum vessel 🏴‍☠️ along the trajectory $L(\cdot)$. The fact that the entanglement persists between the anchor and the points along the trajectory captures the fact that we are tracking the same quantum information through space and time.

1.2 State Localization

A curious aspect of the entanglement localization and trajectory verification protocols described above is that the verifier's register (the quantum anchor) is never acted upon after the entangled state is prepared. In fact, the verifiers don't even need this register to check the prover's responses. Thus, a slightly different way to describe the generation of the anchor state is as follows.

- Sample S, T, u, v, CS as above, and define an isometry $\mathsf{Enc}_{S,T,u,v,\mathsf{CS}}$ that maps

$$\mathsf{Enc}_{S,T,u,v,\mathsf{CS}} : |i\rangle \to |\psi_i\rangle = \frac{1}{\sqrt{|S|}} \sum_{s \in S} (-1)^{\langle s,u \rangle} |s + x_i + v\rangle .$$

- Prepare the maximally entangled state

$$\frac{1}{\sqrt{2^n}} \sum_{i \in [2^n]} |i\rangle_{⚓} |i\rangle_{🏴‍☠️}$$

  and apply $\mathsf{Enc}_{S,T,u,v,\mathsf{CS}}$ to the 🏴‍☠️ register.

[5] Again, the actual trajectory verification protocol we construct is slightly different, for technical reasons.

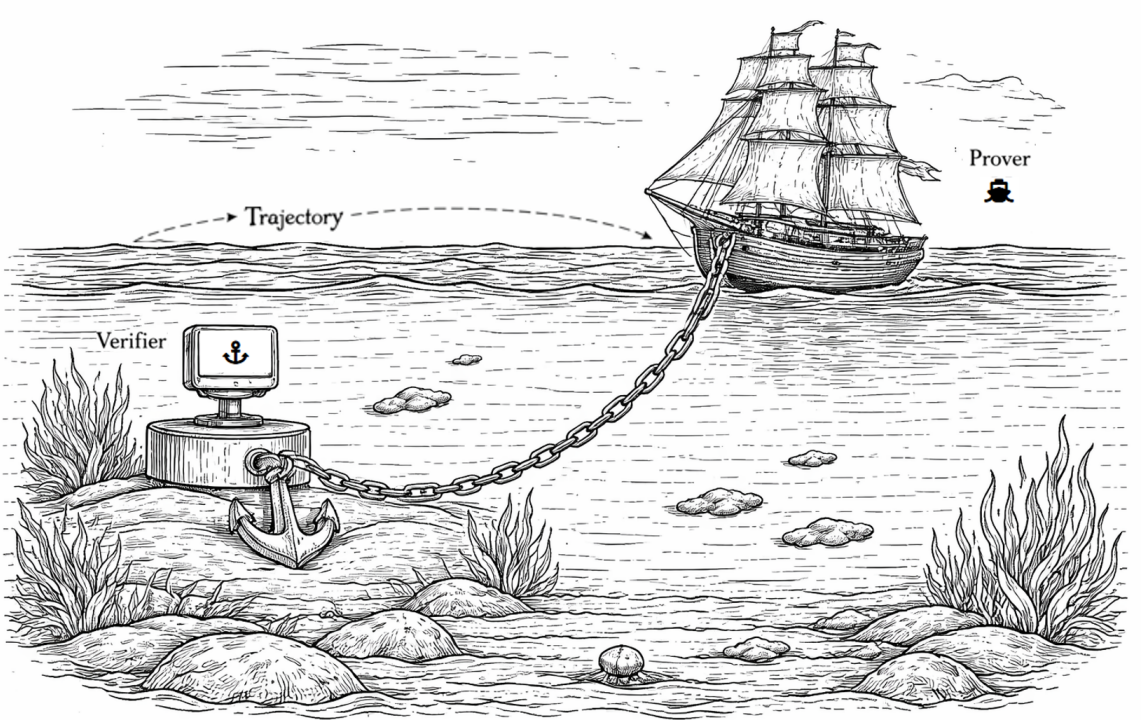

Figure 1: A quantum anchor. Generated by GPT.

- Discard ⚓.

Imagine, then, that instead of applying $\mathsf{Enc}_{S,T,u,v,\mathsf{CS}}$ to half of a maximally entangled state, the verifiers sample a pure state ψ_k from some unclonable family of states $\{\psi_k\}_{k\in[K]}$ (say, the BB84 states) and output $\mathsf{Enc}_{S,T,u,v,\mathsf{CS}}\,|\psi_k\rangle$. If the mixed state produced by this sampling procedure is the maximally mixed state, then this is completely identical from both the prover's *and* extractor's perspective, since neither touches the ⚓ register. It follows then that at each intermediate time t_i, the extractor acting on spacetime point $(L(t_i), t_i)$ will output a register in a state close to $\mathsf{Enc}_{S,T,u,v,\mathsf{CS}}\,|\psi_k\rangle$. In fact, since the extractor knows S, T, u, v, CS, it can undo the isometry to recover $|\psi_k\rangle$ itself.

Thus, even though there is no entanglement to keep track of anymore in this scenario, there is still a meaningful sense in which we have localized a quantum state ψ_k along the trajectory: At any given time t_i, the state ψ_k cannot be found at any other location in space since otherwise the adversary / extractor would have been able to clone it.

This motivates the following general definition of *state localization*. Let $\mathcal{S} = \{\psi_k\}_k$ denote a family of quantum states indexed by k (for example, BB84 states or coset states). Suppose the verifiers sample k, generate (and potentially encode) $|\psi_k\rangle$, and send the resulting state to the prover. We say that a protocol *localizes the state family* $\mathcal{S}$ if for any (possibly nonlocal) prover strategy that is accepted with high probability, there exists an extractor $\mathcal{E}$ that acts on the quantum state at spacetime point (L, t) and outputs $|\psi_k\rangle$ with high fidelity. Furthermore, the state extracted from (L, t) must be *unique*, in that, in expectation over $\psi_k \leftarrow \mathcal{S}$, it is impossible for any adversary to output *two* copies of ψ_k at any time t. We present a formal definition in Definition 9.7.

By applying the techniques used to prove Theorems 8.4 and 9.6, we prove Theorem 9.9, informally stated as follows.

Theorem 1.3 (State localization (informal))**.** *Let $\mathcal{S}$ be any set of n-qubit states that consist of a union of orthonormal bases. Then if $\mathcal{S}$ is unclonable, there exists a protocol that (non-destructively) localizes $\mathcal{S}$ in the ideal obfuscation model.*

While state localization is closely related to entanglement localization, there are some meaningful differences. The extractor produces a quantum state (close to) ψ_k that is classically correlated with the verifiers' choice of k. That is, unlike in the case of entanglement localization, the verifier has an actual classical description k of the state being tracked, as opposed to some entangled register. Thus, there may exist some other side information in the protocol that also depends on k, which may be a useful feature depending on the application. Furthermore, in the setting of purely classical verifiers (such as that of [LLQ21]) we cannot obtain entanglement localization, but state localization is still potentially possible (e.g., the verifiers can delegate the preparation of ψ_k to the prover).

Trajectory verification with localized states. Similarly to how trajectory verification follows naturally from our (non-destructive) entanglement localization protocol, it is straightforward to see that one could build trajectory verification from our (non-destructive) state localization protocol as well. We do not make this explicit, and only present trajectory verification based on entanglement localization in the body.

1.3 Functionality Localization

Entanglement and state localization capture the idea that the prover must carry a specific quantum state to some spacetime point (L, t). However, can we say anything about the prover's *computational capabilities* at (L, t)? This could be desirable in a scenario like export control of secret, proprietary programs (like a foundation language model), where a company or a government would like to guarantee that the ability to run the secret program is localized to a specific location (say within an authorized datacenter).

Let $\mathcal{F} = \{f_k\}_k$ denote a family of functions indexed by some key k (for example, a pseudorandom function family). Suppose that at the beginning of the protocol, the verifiers sample k privately. We say that a protocol *localizes functionality* $\mathcal{F}$ if for any (possibly nonlocal) prover strategy that is accepted with high probability, there exists an extractor $\mathcal{E}$ that acts on the quantum state at verified spacetime point (L, t) and is able to compute $f_k(x)$ for a random input x. On the other hand, there must be no (efficient) adversary that can, for random inputs x, y, compute $f_k(x)$ and $f_k(y)$ at two separate locations without communicating. In other words, the ability to compute $f_k(x)$ has been localized to (L, t), and no other location (at time t) has the ability to compute f_k. This is formally defined in Definition 9.14.

The notion of functionality localization is closely related to (and in fact strengthens) the notion of *quantum copy-protection*, which is the concept of encoding a function f in a quantum state ψ_f that can be used to evaluate f, but cannot be used to compute f in two locations that cannot communicate [Aar09]. There are broad classes of function families that are known to be copy-protectable under various cryptographic assumptions and in various models (see e.g., [ALL+21, CLLZ21, CMP24, LLQZ22, AB24, ABH+26, ÇG26]). Here, we show that any functionality $\mathcal{F}$ that can *in principle* be copy-protected can also be localized. In fact, we don't even need to know an explicit copy-protection scheme for $\mathcal{F}$. Our construction relies on a combination of techniques described above as well as the "best-possible" copy-protection guarantees given by quantum state obfuscation [CG24, BBV24]. We defer further details to Section 2, and establish the following theorem, stated informally.

Theorem 1.4 (Functionality localization (informal))**.** *Let $\mathcal{F} = \{f_k\}_k$ denote any copy-protectable functionality. Then there exists a protocol that (non-destructively) localizes the functionality $\mathcal{F}$ in the ideal obfuscation model.*

1.4 The Ideal Obfuscation Model

We now provide some more context on the ideal obfuscation model (also referred to as the classical oracle model) that we use to establish the security of our protocols. In this model, one can prepare and then "obfuscate" any polynomial-time computable classical functionality f, and thereafter all entities are granted black-box (superposition) access to f. That is, anyone can apply the unitary $U_f : |x\rangle |z\rangle \to |x\rangle |z \oplus f(x)\rangle$ without learning anything else about f.

The ideal obfuscation model has a long history of study in quantum cryptography. In particular, it has been used to establish the feasibility of primitives for which no prior construction existed, e.g. publicly-verifiable quantum money [AC12], signature tokens [BS23], copy-protection for unlearnable programs [ALL+21], witness encryption for QMA [BM22], and obfuscation for quantum circuits [BKNY23, BBV24, HT25, HT26].

While ideal obfuscation of classical circuits has long been known to be impossible to achieve for certain contrived classes of functionalities [BGI+12], the ideal obfuscation model remains a useful model in which to obtain feasibility results, for the following reasons.

- First, any construction in the ideal obfuscation model (that does not require obfuscating the contrived functions from [BGI+12]) yields a plausibly secure construction in the plain model by using *indistinguishability obfuscation* (iO) to obfuscate the function f. While we may not have a formal reduction to the security of iO, it is a reasonable heuristic to expect that the plain model construction is secure. This is analogous to how the community proves security in the *random oracle* model and then replaces the random oracle with a concrete hash function in practice, without necessarily having a formal reduction to any security property of the hash function.
- Second, a recent work [BJRT25], building on [JLLW23], has shown that the (quantum-accessible) ideal obfuscation model can be instantiated from indistinguishability obfuscation plus the heuristic use of a *hash function* (as opposed to an obfuscator) modeled as a "pseudorandom oracle." This brings the ideal obfuscation model even closer to the widely-used random oracle model in that the only cryptographic object we need to treat heuristically is a hash function, which can be instantiated with a cryptographic hash such as SHA3.

We also mention that our use of oracles is to be expected, as our protocols are strengthenings of plain position-verification, which (in the unbounded entanglement setting) is only known in the random oracle model [Unr14]. It would be considered a major breakthrough to prove the security of any of our protocols, or indeed position-verification itself, in the plain model.

Finally, we remark on a slight technical gap in the informal theorem statements above. While the ideal obfuscation model requires the circuit f to be obfuscated to be polynomial-time computable, many of our constructions are most simply described as using a random oracle (plus other manipulations) to define f. Thus, in order to make f efficient, we technically have to replace the random oracle with a (post-quantum) pseudorandom function, which is known from any post-quantum one-way function [Zha21]. Thus, all of our results additionally assume one-way functions.

1.5 Non-Localizability of f-BB84

Now that we have stronger notions of position verification and have shown how to achieve them, it is natural to wonder whether these notions can be achieved with simpler protocols. We show that the well-studied f-BB84 protocol does *not* satisfy entanglement localization. Let $f(x, y)$ be a random boolean function. In this protocol, the honest prover at the proper location L receives classical challenges x, y as well as half of an EPR pair. It is supposed to compute a bit $\theta = f(x, y)$ and measure its qubit in the standard or Hadamard basis according to θ to obtain a bit b, which it sends back in response. The verifier performs the same measurement on its half of the EPR pair and checks if it got the same outcome.

Unruh showed that if f is modeled as a random oracle, then unless the spoofers $\{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$ make $\exp(\Omega(n))$ queries to f (where we think of x, y as n-bit strings), then any successful spoofing strategy requires at least one of the provers $\mathcal{P}_i$ to be in the correct location L [Unr14]. Now, is it possible to strengthen Unruh's result to show that the EPR entanglement can be localized to where the honest prover was supposed to be? We show that, perhaps surprisingly, this is not possible. We describe one attack here and will actually describe a slightly different one in Section 2.

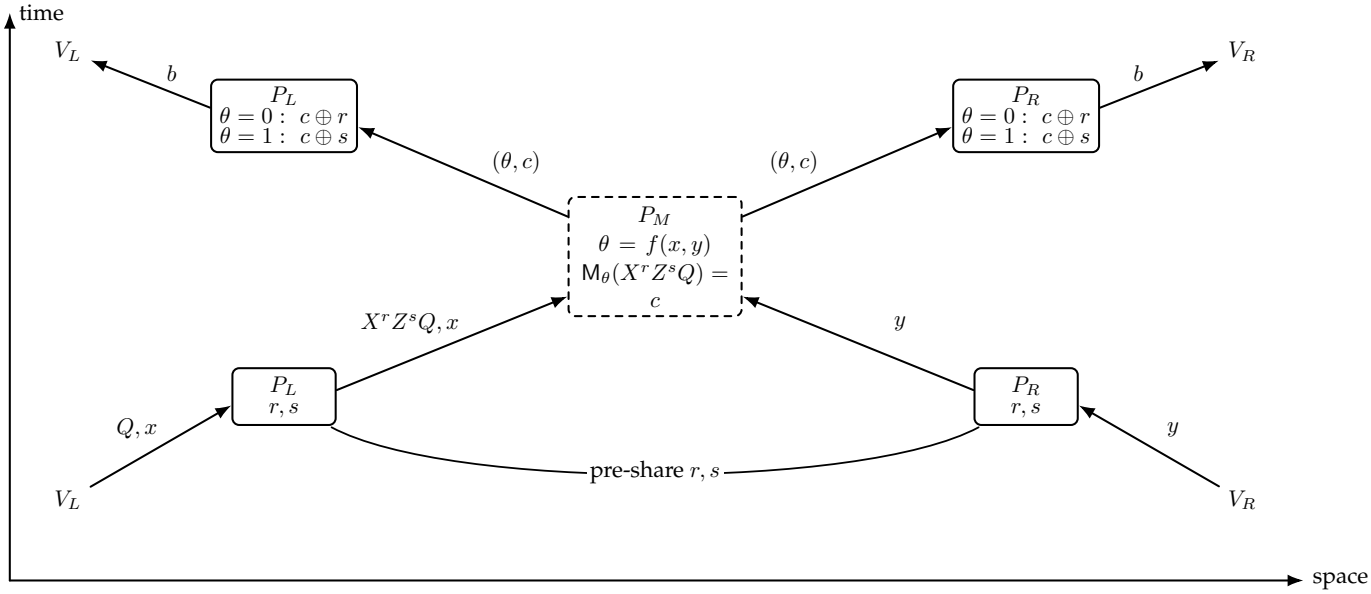

Figure 2: Non-localizability attack on f-BB84

Consider the following 3 prover strategy, pictured in Figure 2. There are provers $\mathcal{P}_L, \mathcal{P}_M, \mathcal{P}_R$ (for "left", "middle", and "right"). Prover $\mathcal{P}_M$ is in the correct location but $\mathcal{P}_L, \mathcal{P}_R$ are on either side of $\mathcal{P}_M$. Suppose that $\mathcal{P}_L$ and $\mathcal{P}_R$ share a random quantum one-time pad key $r, s \in \{0, 1\}$ in superposition. Suppose that the leftmost verifier sends half of the EPR pair; call this qubit Q. It gets intercepted by $\mathcal{P}_L$, who applies a quantum one-time pad $X^r Z^s$ to Q, and forwards it to $\mathcal{P}_M$. The middle prover $\mathcal{P}_M$ gets the one-time padded qubit, along with x, y. It computes $\theta = f(x, y)$, and measures the qubit in the corresponding basis to obtain a bit c. The middle prover $\mathcal{P}_M$ then sends θ, c to both $\mathcal{P}_L$ and $\mathcal{P}_R$. Both $\mathcal{P}_L$ and $\mathcal{P}_R$ compute b as follows: if $\theta = 0$, then $b = c \oplus r$, and otherwise $b = c \oplus s$. All operations are controlled by the superposition of r, s. It is easy to check that this strategy satisfies the timing constraints and will be accepted with probability 1.

Although the middle prover is performing the correct measurement, the quantum entanglement in the EPR pair cannot be localized to the middle prover. At all times in this strategy, the

marginal state of the middle prover is completely uncorrelated with θ and the verifier's EPR qubit. Thus there is no way for the middle prover to locally extract the other end of the EPR pair; it has been nonlocally distributed between the three provers $\mathcal{P}_L, \mathcal{P}_M, \mathcal{P}_R$.

Thus, (entanglement) localization is a rather strong security condition that doesn't hold for all QPV protocols studied in the literature.

1.6 Outlook and Future Directions

Position-based cryptography. In light of our new notions of localization and trajectory verification, it will be interesting to revisit implications for position-based cryptography. As mentioned, there has been a subtle but important gap between the goals of position-based cryptography and the previous formalizations of position security. In Section 10.2, we illustrate this gap by presenting an "attack" on the position-based signature protocol of [BCF+14] that does not violate their security definition but intuitively should not be allowed in any reasonable real-world application of position-based authentication. We then further discuss why we believe our concepts and techniques should provide a stronger foundation for position-based cryptography. We leave further exploration of new definitions and constructions of position-based cryptography to future work.

Future work. Our work raises several other directions for future exploration, which we list here.

- In this work, we restrict our attention to one spatial dimension. While some subtleties typically arise when generalizing to more dimensions (see e.g. [Unr14]), we expect that our techniques should apply in higher dimensions, and we leave a formalization of this to future work.
- A recent work [GGG+26] has shown how to build position *commitments* and *zero-knowledge* position-verification. Can we build on their techniques to obtain *trajectory commitments* and *zero-knowledge localization* protocols?
- [LLQ21, KLC+25] have shown how to achieve position verification with purely classical communication. Can we obtain quantum localization and trajectory verification with only classical communication?
- Can we further generalize the notion of quantum localization? For example, can we localize any (unclonable) QMA witness, perhaps by integrating our techniques with recent work [BJRT25, KKR26] that shows how to non-destructively verify QMA witnesses?
- Our localization and trajectory verification protocols all rely on highly entangled and difficult-to-implement states. Might there exist simpler protocols that are more amenable to experimental realization in the short term? We remark that simpler localization protocols are likely easier to achieve if we drop the non-destructive requirement, but such protocols would not serve as building blocks to trajectory verification.
- Our protocols assume noiseless transmission of quantum states, and that the prover is in exactly the right position at the right time. Can we achieve more *robust* versions of localization and trajectory verification, which resist environmental noise and allow the prover some leeway in their declared position / trajectory?
- In this work, we discuss the way to achieve publicly verifiable trajectory verification and publicly verifiable localization schemes except publicly verifiable functionality localization (See Footnote 20 for more details). Can we construct a publicly verifiable functionality localization scheme?

Acknowledgments. We thank Tal Malkin, Alex May, and Saachi Mutreja for helpful discussions. HY is supported by the Sloan Foundation. LO is supported by an NSF Graduate Fellowship.

2 Technical Overview

In this overview, we first introduce the main techniques used to construct trajectory verification. Along the way, we'll see how to perform entanglement and state localization. Then, we'll cover some additional ideas required to obtain functionality localization.

2.1 Entanglement Localization and Trajectory Verification

To develop our trajectory verification scheme, we first develop a *non-destructive test of entanglement*, next upgrade it to *entanglement localization*, and finally derive trajectory verification. A non-destructive test of entanglement distributes an entangled state between two parties, $\mathcal{V}$ (the verifier) and $\mathcal{P}$ (the prover), and then specifies an interactive protocol that can be repeatedly used to test whether the prover still holds the state entangled with the verifier. An entanglement localization scheme is a proof system where the prover can convince the verifier that the entanglement exists across the verifier's state and a state within a specific defined region, such as an interval $[L - \Delta, L + \Delta]$. We first consider how to verify entanglement in a non-destructive manner.

Is verifying entanglement always destructive? We begin with the standard approach for verifying one bit of entanglement. Suppose two parties $\mathcal{V}$ (an honest verifier) and $\mathcal{P}$ (a potentially adversarial prover) initially share an EPR pair. The prover $\mathcal{P}$ may perform some operations on its subsystem, resulting in a joint state $\rho_{\mathbf{AB}}$, where $\mathbf{A}$ is held by $\mathcal{V}$ and $\mathbf{B}$ is held by $\mathcal{P}$. As discussed for example in [VZ21], the entanglement can be tested via the following protocol:

1. $\mathcal{V}$ samples a random bit $b \xleftarrow{\$} \{0, 1\}$. If $b = 0$, it measures $\mathbf{A}$ in the standard basis; otherwise, it measures $\mathbf{A}$ in the Hadamard basis. Let the measurement outcome be x. The verifier then sends b to $\mathcal{P}$.
2. Upon receiving b, the prover $\mathcal{P}$ performs the corresponding measurement operation and outputs a value x'.

If $\mathcal{P}$ can perfectly predict the standard or Hadamard basis measurement outcome on $\mathcal{V}$'s side, i.e., $x = x'$ with probability 1 over the choice of b, then $\mathcal{V}$ is convinced that its qubit was maximally entangled with $\mathcal{P}$'s system. However, this procedure necessarily collapses the entanglement, and at first glance this appears unavoidable.

Our main observation here is the following:

*To verify entanglement, it suffices to argue the **existence** of a procedure that would pass the measurement test, without actually executing this procedure. In other words, there is a **non-collapsing test** such that, if the prover passes the non-collapsing test, it implies that the prover could be used to pass the collapsing test, which is only used in the analysis.*

More concretely, we design two tests Π_{ncol} and Π_{col} for $\mathcal{P}$. The test Π_{ncol} is the non-collapsing test that we actually execute. We then argue, via an indistinguishability argument, that any prover passing Π_{ncol} must also perform well in a corresponding collapsing test Π_{col}, which certifies the presence of entanglement.

Deploying decoy qubits. We now present a first toy example. In addition to the original EPR pair shared between $\mathcal{V}$ and $\mathcal{P}$, we sample n random standard-basis qubits from $\{|0\rangle, |1\rangle\}$ and n random Hadamard-basis qubits from $\{|+\rangle, |-\rangle\}$. These $2n$ qubits, together with the prover's half of the EPR pair, are randomly shuffled into the register $\mathbf{B}$, which now contains $2n+1$ qubits.

To test for entanglement, instead of querying the unique (hidden) entangled qubit directly, $\mathcal{V}$ proceeds as follows:

1. Sample a random basis (standard or Hadamard).
2. Send $\mathcal{P}$ the indices of the n qubits prepared in that basis and request their measurement outcomes in the corresponding basis.
3. Check whether the answers are correct.

From the prover's perspective, all $2n+1$ qubits appear as random BB84 states and are thus maximally mixed. The $2n$ additional qubits serve as decoys that hide the location of the entangled qubit. To succeed, $\mathcal{P}$ must effectively retain all qubits, and hence preserve the entanglement. Discarding any qubit risks failing the test, since that qubit may be queried.

To formalize this intuition, consider the single-shot setting with a prover $\mathcal{P}$ that passes the protocol with probability 1. Let S denote the indices of the standard-basis qubits, and consider some arbitrary position $x \in S$. We claim that $\mathcal{P}$ would also answer correctly if, instead of querying S, we queried $(S \setminus \{x\}) \cup \{e\}$, where e is the index of the entangled qubit. Here, correctness for e means consistency with the standard-basis measurement outcome on $\mathbf{A}$. This follows because S and $(S \setminus \{x\}) \cup \{e\}$ are indistinguishable from the prover's perspective.

This illustrates our earlier principle: the test on S is non-collapsing, while the test on $(S \setminus \{x\}) \cup \{e\}$ is collapsing, as it measures the entangled qubit.

However, this approach has a clear limitation. If the test is repeated multiple times, $\mathcal{P}$ may learn which qubit is never queried, thereby identifying the entangled qubit. Once identified, the prover can discard it and still answer future queries correctly.

Reusable security via coset states. To address this issue, we introduce a stronger method for hiding the decoys, based on *coset states*. A coset state $|A_{s,s'}\rangle$, for a subspace $A \leq \mathbb{F}_2^n$ of dimension $n/2$ and $s, s' \in \mathbb{F}_2^n$, is defined as

$$|A_{s,s'}\rangle = \frac{1}{\sqrt{|A|}} \sum_{a \in A} (-1)^{\langle a, s' \rangle} |a + s\rangle .$$

This state admits the following alternative interpretation:

1. Prepare an n-qubit state in which the first $n/2$ qubits are random Hadamard-basis states, and the remaining $n/2$ qubits are random standard-basis states.
2. Sample a random change-of-basis (invertible) matrix $U_{\mathsf{shift}} \in \mathbb{F}_2^{n \times n}$ and apply the unitary $\mathcal{U}_{\mathsf{shift}}$ defined as
$$\mathcal{U}_{\mathsf{shift}} |x\rangle := |U_{\mathsf{shift}} x\rangle .$$

The second step plays a crucial role: without knowledge of U_{shift}, the subspace A and the cosets $A + s$ and $A^\perp + s'$ are computationally hidden. In particular, given $|A_{s,s'}\rangle$, one cannot distinguish between:

- Access to membership oracles $\mathcal{O}_{A+s}$ and $\mathcal{O}_{A^\perp + s'}$, and
- Access to membership oracles $\mathcal{O}_{B+s}$ and $\mathcal{O}_{C+s'}$, where B and C are random superspaces of dimensions $n - \omega(\log n)$ containing A and $A^\perp$, respectively.

Here, $\mathcal{O}_S$ denotes the membership oracle for a set S, outputting $\top$ on inputs in S and $\bot$ otherwise.

Intuitively, the random change of basis hides the "positions" of the standard- and Hadamard-basis qubits, in a strictly stronger manner than the random permutation from before. We now apply this idea to also hide the positions of *entangled* qubits. Consider the following construction:

1. Let ⚓ be an n-qubit register and 🏴‍☠️ a $3n$-qubit register. Initialize the first n qubits of 🏴‍☠️ as random Hadamard-basis states, the next n qubits as halves of EPR pairs with ⚓, and the final n qubits as random standard-basis states.
2. Sample a random change-of-basis matrix $U_{\mathsf{shift}} \in \mathbb{F}_2^{3n \times 3n}$ and apply the unitary $\mathcal{U}_{\mathsf{shift}}$ defined as
$$\mathcal{U}_{\mathsf{shift}} |x\rangle := |U_{\mathsf{shift}} x\rangle$$
to the register 🏴‍☠️.

We refer to the resulting state as the *anchor state* $|\Psi^{\mathsf{sk}}\rangle_{⚓🏴‍☠️}$, where $\mathsf{sk} = (U_{\mathsf{shift}}, u, v)$.

1. S is the subspace spanned by the first n columns of U_{shift}.
2. T is the subspace spanned by the first $2n$ columns of U_{shift}.
3. $u := U_{\mathsf{shift}}^{-t} u'$ where u' is the vector obtained by appending $2n$ zeros to the vector of Hadamard-basis values (where $|+\rangle$ is mapped to 0 and $|-\rangle$ is mapped to 1), and U_{shift}^{-t} is the inverse transpose of U_{shift}.
4. $v := U_{\mathsf{shift}} v'$ where v' is the vector obtained by appending $2n$ zeros before the vector of the standard-basis values.

We clarify some facts here. First, if we measure 🏴‍☠️ in the standard basis, we will get a vector in $T + v$. If we measure 🏴‍☠️ in the Hadamard basis, we will get a vector in $S^\perp + u$. Combining our previous non-destructive test with the hiding property achieved by oracles, we define the following test:

1. In the setup phase, ⚓ is given to $\mathcal{A}$ and 🏴‍☠️ is given to $\mathcal{B}$. Sample $\theta \xleftarrow{\$} \{0,1\}$ and $s \xleftarrow{\$} \{0,1\}^\lambda$. θ is given to $\mathcal{B}$.
2. If $\theta = 0$, give the oracle $\mathcal{O}^{\mathsf{ncol}}_{T+v}$ to $\mathcal{B}$, where the oracle is defined as follows:
   - $\mathcal{O}^{\mathsf{ncol}}_{T+v}$: On input z, check whether $z \in T + v$. Output s if it is, and output $\bot$ otherwise.

   If $\theta = 1$, give the oracle $\mathcal{O}^{\mathsf{ncol}}_{S^\perp+u}$ to $\mathcal{B}$, where the oracle is defined as follows:
   - $\mathcal{O}^{\mathsf{ncol}}_{S^\perp+u}$: On input z, check whether $z \in S^\perp + u$. Output s if it is, and output $\bot$ otherwise.

   $\mathcal{B}$ is supposed to return s.

An honest $\mathcal{B}$ can simply perform the corresponding basis evaluation coherently. That is, for $\theta = 0$, it computes $\mathcal{O}^{\mathsf{ncol}}_{T+v}$ coherently on register 🏴‍☠️ and outputs the measurement result. For example, Figure 3 shows the non-collapsing test when $U_{\mathsf{shift}} = I$ and $\theta = 0$. Only the correctness of the standard-basis qubits is tested. One may notice that this is a non-collapsing test, and the state is unchanged after the test is completed.

Our next step will be to define a *collapsing* test that is designed to be indistinguishable from the non-collapsing test to $\mathcal{B}$. To show indistinguishability, we invoke an argument based on *subspace bloating*, which says that a query-efficient algorithm cannot distinguish between membership oracles for a random subspace S, and a randomly sampled superspace $T \geq S$ – more intuition is provided in Section 5.

1. In the setup phase, ⚓ is given to $\mathcal{A}$ and 🏴‍☠️ is given to $\mathcal{B}$. Sample $\theta \xleftarrow{\$} \{0,1\}$ and a random oracle $\mathcal{O}_{\mathsf{random}}$. θ is given to $\mathcal{B}$.

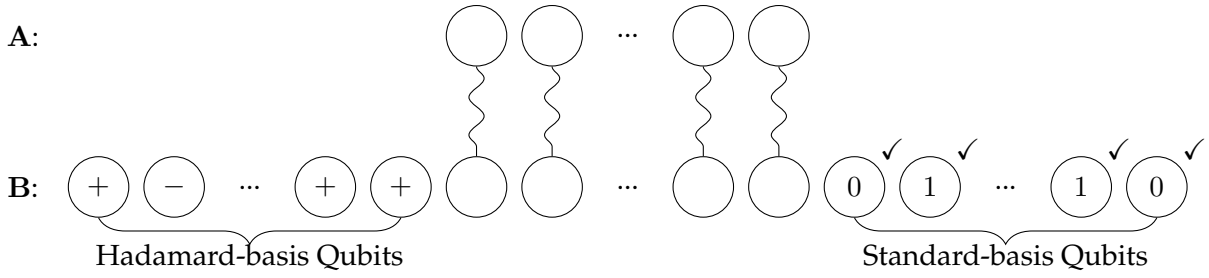

Figure 3: The Non-collapsing Test. In an actual test, all qubits in **B** are shuffled.

2. If $\theta = 0$, give the oracle $\mathcal{O}^{\mathsf{col}}_{T+v}$ to $\mathcal{B}$, where the oracle is defined as follows:
   - $\mathcal{O}^{\mathsf{col}}_{T+v}$: On input z, check whether $z \in T + v$. Output $\mathcal{O}_{\mathsf{random}}(\mathsf{Can}_S(z))$ where $\mathsf{Can}_S(z)$ is the canonical representation of coset $S + z$ if it is, and output $\perp$ otherwise.

   If $\theta = 1$, give the oracle $\mathcal{O}^{\mathsf{col}}_{S^\perp+u}$ to $\mathcal{B}$, where the oracle is defined as follows:
   - $\mathcal{O}^{\mathsf{col}}_{S^\perp+u}$: On input z, check whether $z \in S^\perp + u$. Output $\mathcal{O}_{\mathsf{random}}(\mathsf{Can}_{T^\perp}(z))$ where $\mathsf{Can}_{T^\perp}(z)$ is the canonical representation of coset $T^\perp + z$ if it is, and output $\perp$ otherwise.
3. If $\theta = 0$, the verifier measures ⚓ in the standard basis to obtain x. Otherwise, if $\theta = 1$, the verifier measures ⚓ in the Hadamard basis to obtain x.

One can think of the canonical representation of a coset as a unique identifier for the coset. Now suppose that $\theta = 0$ and the prover coherently computes $\mathcal{O}^{\mathsf{col}}_{T+v}$ on register 🏴‍☠️. A measurement on the oracle output will collapse the state according to which coset of S the vector z belongs to, where z is the standard-basis value on 🏴‍☠️. This is equivalent to measuring all entangled qubits in the standard basis. For example, Figure 4 shows the collapsing test when $U_{\mathsf{shift}} = I$ and $\theta = 0$.

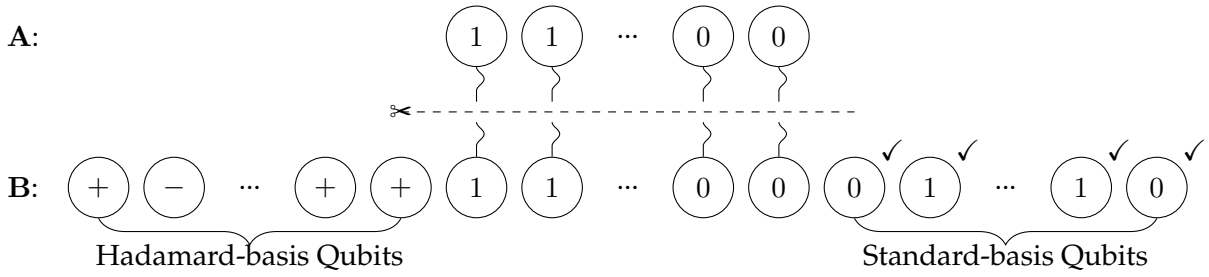

Figure 4: The Collapsing Test. In an actual test, all qubits in **B** are shuffled.

In the collapsing test, each potential measurement result x obtained by the verifier corresponds to a different input on which the honest prover queried $\mathcal{O}_{\mathsf{random}}$. In the main proof, we show that *any* (potentially adversarial) prover that passes with probability close to 1 in the non-collapsing test will yield the following behavior in the collapsing test with overwhelming probability:

- If $\theta = 0$, the only input on which $\mathcal{O}_{\mathsf{random}}$ is accessed is $\mathsf{Can}_S(U_{\mathsf{shift}}(0^n \times x \times 0^n) + v)$, where x is the standard-basis measurement result obtained by the verifier.
- If $\theta = 1$, the only input on which $\mathcal{O}_{\mathsf{random}}$ is accessed is $\mathsf{Can}_{T^\perp}(U^{-t}_{\mathsf{shift}}(0^n \times x \times 0^n) + u)$, where x is the Hadamard-basis measurement result obtained by the verifier.

Hence, we can test entanglement by repeatedly asking the prover to answer non-collapsing tests. If the prover is able to answer all of them, they would also succeed in the collapsing test, and we will be convinced that the prover is holding the entanglement. Indeed, a prover that passes the collapsing test will predict the verifier's standard / Hadamard basis measurement results, allowing the extraction of EPR pairs following the strategy of [VZ21].

Remark 2.1. *In [VZ21], it was shown that classically-verifiable proofs of knowledge of an unclonable state cannot be non-destructive, otherwise an unbounded cloner would exist. Our discussion above can be used to obtain a non-destructive classically-verifiable* argument *of knowledge for quantum money states. This does not contradict the result in [VZ21] since soundness holds only against efficient malicious provers.*

Non-localizability of f-BB84. We now move to the second component mentioned above: Localization of entanglement. As discussed in the introduction, the famous f-BB84 protocol does not satisfy entanglement localization. Here we give an alternative attack on this protocol that better conveys the motivation for our own construction.

Consider the following strategy involving three provers. There are provers $\mathcal{P}_L$, $\mathcal{P}_M$, and $\mathcal{P}_R$ (for "left", "middle", and "right"). Prover $\mathcal{P}_M$ is in the correct location, while $\mathcal{P}_L$ and $\mathcal{P}_R$ are positioned on either side of $\mathcal{P}_M$. Upon receiving the EPR pair $|\Phi\rangle_{\mathbf{AB}}$, the prover applies the following unitary:

$$|x\rangle_{\mathbf{B}} \to \frac{1}{\sqrt{2}} \sum_{b\in\{0,1\}} |b\rangle_{\mathbf{L}} |b\rangle_{\mathbf{M}_0} |b\oplus x\rangle_{\mathbf{M}_1} |b\oplus x\rangle_{\mathbf{R}}$$

and sends $\mathbf{L}$, $\mathbf{M} = \mathbf{M}_0 \otimes \mathbf{M}_1$, and $\mathbf{R}$ to $\mathcal{P}_L$, $\mathcal{P}_M$, and $\mathcal{P}_R$, respectively. It can be seen that there is no entanglement between $\mathbf{A}$ and $\mathbf{M}$. To pass the challenge, the prover does the following.

If $\theta = 0$:

- The middle prover $\mathcal{P}_M$ sends $\mathbf{M}_1$ to $\mathcal{P}_L$ and $\mathbf{M}_0$ to $\mathcal{P}_R$.
- The left prover $\mathcal{P}_L$ measures $\mathbf{L}$ and $\mathbf{M}_1$. If the bits are the same, it outputs 0; otherwise, it outputs 1.
- The right prover $\mathcal{P}_R$ measures $\mathbf{R}$ and $\mathbf{M}_0$. If the bits are the same, it outputs 0; otherwise, it outputs 1.

If $\theta = 1$:

- The middle prover $\mathcal{P}_M$ sends $\mathbf{M}_0$ to $\mathcal{P}_L$ and $\mathbf{M}_1$ to $\mathcal{P}_R$.
- The left prover $\mathcal{P}_L$ outputs 0 if its state is $\frac{1}{\sqrt{2}}|00\rangle_{\mathbf{LM}_0} + \frac{1}{\sqrt{2}}|11\rangle_{\mathbf{LM}_0}$; otherwise, it outputs 1.
- The right prover $\mathcal{P}_R$ outputs 0 if its state is $\frac{1}{\sqrt{2}}|00\rangle_{\mathbf{RM}_1} + \frac{1}{\sqrt{2}}|11\rangle_{\mathbf{RM}_1}$; otherwise, it outputs 1.

Intuitively, this attack works because the left prover and the right prover "encrypt" the answer, such that the answer is stored in the interference between $\mathbf{M}$ and $\mathbf{LR}$, while the middle prover has no clue about the answer when receiving the challenge. The middle prover does not measure anything, and the measurement is delayed until the global interference becomes local at the left and right provers.

Forcing a measurement via monogamy-of-entanglement. To bypass this attack, the key idea is to *force* the prover to perform a particular "measurement" (in quotes because in reality they will be performing a non-destructive oracle query) at the position of the middle prover. More precisely,

we want to ensure that the prover demonstrates its ability, in principle, to measure the state in a specific basis. We begin by replacing the random basis test on the EPR pair with a non-destructive test for entanglement:

1. In the setup phase, ⚓ is given to $\mathcal{A}$ and 🏴‍☠️ is given to $\mathcal{B}$. Sample $s \xleftarrow{\$} \{0,1\}^\lambda$.
2. The left verifier $\mathcal{V}_L$ samples x_L, and the right verifier $\mathcal{V}_R$ samples x_R. Both x_L and x_R are broadcast at the appropriate time to ensure they meet at the correct position.
3. The middle prover computes $\theta = f(x_L, x_R)$ and performs the following:
   - If $\theta = 0$, it is granted access to the oracle $\mathcal{O}^{\mathsf{ncol}}_{T+v}$ and performs the coherent standard-basis measurement described above.
   - If $\theta = 1$, it is granted access to the oracle $\mathcal{O}^{\mathsf{ncol}}_{S^\perp+u}$ and performs the coherent Hadamard-basis measurement described above.
   - It broadcasts the result.
4. Both the left and right verifiers receive the result broadcast by the prover. The verifier returns $\top$ if both messages are on time and correct (equal to s); otherwise, it returns $\bot$.

One may ask how the prover can be given access to an oracle during the middle of the computation. It can be assumed that there is a class of oracles indexed by x_L, x_R, and a particular oracle is selected for the parties once they know both x_L and x_R.

Now, actually proving the extractability of this simple protocol appears challenging. Our solution is to repeat this protocol twice: one standard round followed by one Hadamard round, or vice versa. For example, let's consider the case where the first round is a standard-basis round and the second round is a Hadamard-basis round.

1. In the setup phase, ⚓ is given to $\mathcal{A}$ and 🏴‍☠️ is given to $\mathcal{B}$. Sample $s_0, s_1 \xleftarrow{\$} \{0,1\}^\lambda$.
2. The first round has $\theta_0 = 0$. The prover is granted access to the oracle $\mathcal{O}^{\mathsf{ncol}}_{T+v}$ with s_0 encoded as the secret and performs the coherent standard-basis measurement described above.
3. After a brief interval, the second round arrives with $\theta_1 = 1$. The prover is granted access to the oracle $\mathcal{O}^{\mathsf{ncol}}_{S^\perp+u}$ with s_1 encoded as the secret and performs the coherent Hadamard-basis measurement described above.
4. Both the left and right verifiers receive the results broadcast by the prover. The verifier returns $\top$ if all messages are on time and correct; otherwise, it returns $\bot$.

Now, let's examine how this two-round protocol prevents the attack described above. Suppose there are three provers $\mathcal{P}_L, \mathcal{P}_M, \mathcal{P}_R$ that succeed at passing the protocol, but where only $\mathcal{P}_M$ is at the correct spacetime location. Furthermore, suppose that $\mathcal{P}_M$ fails to "measure" and determine the answer for the first challenge at the correct location and time. In this case, the strategy can be recast as a successful adversary $\mathcal{A} = (\mathcal{A}_L, \mathcal{A}_M, \mathcal{A}_R)$ in the following monogamy-of-entanglement type game.[6]

1. The challenger generates $\left|\Psi^{\mathsf{sk}}\right\rangle$ and gives the register 🏴‍☠️ to $\mathcal{A}_M$.
2. $\mathcal{A}_M$ generates a bipartite state on $\mathbf{LR}$. It sends $\mathbf{L}$ to $\mathcal{A}_L$ and sends $\mathbf{R}$ to $\mathcal{A}_R$.
3. The challenger samples $s_0, s_1 \xleftarrow{\$} \{0,1\}^\lambda$. It gives $\mathcal{O}^{\mathsf{ncol}}_{T+v}$ to $\mathcal{A}_L$ that encodes s_0 as its underlying secret, and it gives $\mathcal{O}^{\mathsf{ncol}}_{S^\perp+u}$ to $\mathcal{A}_R$ that encodes s_1 as the underlying secret. In addition, it gives s_0 to $\mathcal{A}_R$.

[6] We note that the description of this game is slightly inaccurate, and we refer the reader to Section 6, and in particular Lemma 6.7, for formal details.

4. $\mathcal{A}_L$ returns s'_0 and $\mathcal{A}_R$ returns s'_1.
5. The challenge is passed if $s'_0 = s_0$ and $s'_1 = s_1$.

Intuitively, the reduction works as follows: $\mathcal{P}_M$ corresponds to $\mathcal{A}_M$, the left prover $\mathcal{P}_L$'s process for solving the $\mathcal{O}^{\mathsf{ncol}}_{T+v}$ challenge is used to construct $\mathcal{A}_L$, and the right prover $\mathcal{P}_R$'s process for solving the $\mathcal{O}^{\mathsf{ncol}}_{S^\perp+u}$ challenge is used to construct $\mathcal{A}_R$. Due to the structure of the game, we also have to include the correct "left" answer s_0 in the view of the right prover, which is a non-standard variant of monogamy-of-entanglement.

In Section 6, we show, via a sequence of reductions, that this game is hard for any adversary to win with probability better than $\frac{1}{2} + \mathsf{negl}(\lambda)$. Thus, we can eliminate this type of prover, meaning that any successful prover must indeed "measure" (meaning query the oracle on its state) at the correct location and time. Finally, we argue that this implies the existence of an *extractor* that, using quantum recording techniques [Zha19], can obtain either the result of measuring 🐞 in the standard or the Hadamard basis at the correct location and time. By the previous entanglement verification arguments, we can thus localize the entanglement to the desired spacetime location.

Trajectory verification. As mentioned earlier, trajectory verification follows fairly naturally from our entanglement localization protocol. At a high level, the verifier first sends the vessel register 🐞 to the prover, and then runs the entanglement localization protocol at a series of steps along the prover's claimed trajectory. In slightly more detail, at each step, the verifier sends a random standard- or Hadamard- basis challenge.[7] Therefore, each *pair* of steps gives rise to a slightly modified version of the two-round protocol described above, where the cadence is not fixed standard followed by Hadamard, but rather both are randomly chosen. We show that in this modified game, one can still extract entanglement from any successful prover at the time of the first challenge. This means that one can extract entanglement from the prover at any step on their trajectory, assuming they have a high probability of passing all challenges. This is formalized in Section 9.2.

Optimizations and applications. We conclude by highlighting two additional features of our constructions. First, while the informal guarantees stated above only yield meaningful extraction when the adversary succeeds with probability $1 - o(1)$, this limitation can be overcome via sequential repetition. By executing sufficiently many copies of the localization protocol in rapid succession, we can amplify soundness so that any prover accepted with non-negligible probability must still contain the localized object at the verified spacetime point. In this way, our localization guarantees can be strengthened to hold against adversaries whose success probability is merely inverse polynomial.

Second, we discuss how to upgrade our protocols to achieve *public verifiability*. Intuitively, because the verification procedure is implemented using public classical oracles, one can arrange the protocol so that anyone with oracle access can verify the prover's responses, without requiring access to any secret verification key. We defer the precise formulation and construction of publicly-verifiable localization protocols to Section 10.1. After that, in Section 10.2, we discuss broader implications for position-based cryptography and related applications.

2.2 Functionality Localization

Functionality localization can be seen as a strengthening of quantum *copy-protection*, which enables encoding a function f (sampled from some family $\mathcal{F}$) into a quantum state ρ_f such that (i) ρ_f

[7] We note that one could alternatively specify that the verifier sent strictly alternating challenges, but this requires it to keep some state in between each challenge, which may be undesirable.

can be used to compute $f(x)$ for any input x, and (ii) no adversary can create two disjoint registers that can simultaneously be used to compute f (over some distribution $\mathcal{D}$ on inputs x). That is, copy-protection establishes that certain functionalities can be rendered unclonable, meaning they cannot exist in two places at once. This then raises the possibility that functionalities can be *localized*! In this work, we show that any functionality $\mathcal{F}$ that can in principle be copy-protected, can also be localized, in the classical ideal obfuscation model.[8]

Our definition of functionality localization extends copy-protection by adding a new security property, *extraction soundness*, in addition to the usual unclonability guarantee. Extraction soundness, in a similar fashion to the notion of extractability for our other primitives, ensures that a successful prover can be converted into an evaluator for the copy-protected functionality by means of a physical procedure called the extractor. Here, the extractor is given access to the registers of a prover in the functionality localization protocol at a particular time, and should be able to evaluate the functionality on random inputs with high probability. Combined with the typical copy-protection security (which we term "uniqueness"), this guarantees that a successful prover must possess the *only* working implementation of the functionality at a specific point in spacetime. As discussed later on, this can have interesting applications to quantum cryptographic protocols, such as position-based authentication.

Our construction of functionality localization combines the localization techniques introduced in the previous sections with "best-possible" copy-protection. A best-possible copy-protector is a scheme that takes as input a function $f \leftarrow \mathcal{F}$ and outputs a state ρ_f such that, if there *exists* any method of copy-protecting f, then ρ_f is itself a copy-protected version of f. The idea of best-possible copy-protection was introduced by [CG24], who showed that any "quantum state obfuscator" serves as a best-possible copy protection scheme. Quantum state obfuscation was then shown to exist in the classical oracle model [BBV24].

Let QSO be a quantum state obfuscator, let $(\mathsf{Auth}, \mathsf{Ver})$ be a message authentication code, and consider the following (simplified) proposal for functionality localization.

- In the setup phase, the verifiers sample a functionality $f \leftarrow \mathcal{F}$, an authentication key sk, and a random coset state $|S_{v,u}\rangle$. Let $\widehat{f}_{\mathsf{sk}}$ be the functionality that, on input x, outputs $f(x)$ along with a signature σ on the pair $(x, f(x))$. Output $\rho_f \leftarrow \mathsf{QSO}(\widehat{f}_{\mathsf{sk}})$ and $|S_{v,u}\rangle$.
- In the online phase, the verifiers sample $\theta \leftarrow \{0,1\}, x \leftarrow \mathcal{D}$, and a random oracle $\mathcal{O}_{\mathsf{random}}$, and do the following.
  - If $\theta = 0$, they send a secret-sharing of θ, x to the prover along with an oracle $\mathcal{O}_{\mathsf{sk},S,v}$ that takes as input (z, x, y, σ) and if (i) $z \in S + v$ and (ii) $\mathsf{Ver}(\mathsf{sk}, (x, y), \sigma) = \top$, outputs $\mathcal{O}_{\mathsf{random}}(x, y)$.
  - If $\theta = 1$, they send a secret-sharing of θ, x to the prover along with an oracle $\mathcal{O}_{\mathsf{sk},S^\perp,u}$ that takes as input (z, x, y, σ) and if (i) $z \in S^\perp + u$ and (ii) $\mathsf{Ver}(\mathsf{sk}, (x, y), \sigma) = \top$, outputs $\mathcal{O}_{\mathsf{random}}(x, y)$.
- The honest prover, sitting at the correct spacetime point, uses ρ_f to compute $f(x), \sigma_{x,f(x)}$, and then queries its oracle using either $|S_{v,u}\rangle$ in the standard or Hadamard basis to obtain the value $\mathcal{O}_{\mathsf{random}}(x, f(x))$, which it sends back to the verifier.

We note that there are several ways in which our final protocol differs from this simplified description, and we refer the reader to Section 9.4 for more details.

[8] Technically, we say that a functionality is "copy-protectable" if there exists some distribution $\mathcal{D}$ over inputs such that $\mathcal{F}$ is copy-protectable with respect to independent challenges x_0, x_1 sampled from $\mathcal{D}$. See Section 9.4 for more details.

The intuition here is that, in order to succeed in the protocol, the prover *must* query their oracle at the correct location on $z, x, f(x), \sigma_{x,f(x)}$, where z is either in $S + v$ or $S^{\perp} + u$. On the other hand, due to the security of the message authentication code and the hiding of QSO, they will not be able to learn a signature $\sigma_{x,y}$ on *any* $y \neq f(x)$. Hence, the oracle $\mathcal{O}_{\mathsf{random}}$ will be queried on $(x, f(x))$ but not on (x, y) for any $y \neq f(x)$. This yields the existence of an *extractor* that, using quantum recording techniques, can extract the value of $f(x)$ from any successful adversarial prover at the location to be verified. We note that we crucially rely on the techniques developed above in the context of entanglement localization in order to establish that the adversary must indeed query their oracle at the correct spacetime location. In particular, even though there is no entanglement in this setting, we can view a random coset state as a "degenerate" version of the anchor state described above with the number of EPR pairs set to 0, and our techniques carry over naturally to this setting.

3 Preliminaries

3.1 General Notation

In many contexts, objects are parameterized by a security parameter $\lambda \in \mathbb{N}$, which is often omitted to reduce clutter. We write PPT and QPT to denote probabilistic polynomial-time and quantum polynomial-time, respectively. Algorithms are denoted with calligraphic letters, such as $\mathcal{A}, \mathcal{B}, \mathcal{P}, \mathcal{V}$. Quantum registers are denoted with bold letters, such as $\mathbf{A}, \mathbf{B}, \mathbf{X}$. We write $\mathsf{negl}(\lambda)$ to denote a negligible function in λ. For a protocol Prot and two parties $\mathcal{P}$ and $\mathcal{V}$ (prover and verifier), we write $Y_{\mathcal{P}}, Y_{\mathcal{V}} \leftarrow \mathsf{Prot}(\mathcal{P}(\mathbf{A}) \rightleftharpoons \mathcal{V}(\mathbf{B}))(x)$ to denote the experiment where $\mathcal{P}$ and $\mathcal{V}$ interact by running Prot with public input x, private prover input $\mathbf{A}$, and private verifier input $\mathbf{B}$. Here, $Y_{\mathcal{P}}$ is some set of prover outputs and $Y_{\mathcal{V}}$ is some set of verifier outputs. $Y_{\mathcal{V}}$ typically includes a bit $b \in \{\top, \bot\}$ known as the acceptance decision, with $\top$ meaning the verifiers accept and $\bot$ meaning that they reject.

3.2 Modeling Computations and Interactions in Spacetime

In this section, we describe our model for how algorithms and interactive protocols take place in physical spacetime, which will be important for the interpretation of our main results. This model is somewhat new to our work, but is inspired by similar sections in [CGMO14],[Unr14], and [GGG+26]. Note that later sections of this paper, as well as our main results, do not rely closely on the wording of this section — instead, they tend to use terminology of a broader and more standard flavor. We strongly believe that our results are general enough that there is a wide variety of acceptable modeling choices under which they can retain correctness and meaningful interpretability.

Spacetime. All of our protocols are set in a bounded one-dimensional space[9], represented as $[-1, 1]$. This is the line segment between two statically placed verifiers, $\mathcal{V}_L$ at position -1 and $\mathcal{V}_R$ at position 1. We model time as $\mathbb{R}^{\geq 0}$, and we assume that all parties have access to a synchronized clock that reports the current time $t \in \mathbb{R}^{\geq 0}$. Units of space and time are taken such that light travels

[9] It is natural to ask whether our protocols can be extended to work in standard three-dimensional space. We expect that, similarly to [Unr14], there is a natural generalization to three dimensions which can be then compiled to a more abstracted causal circuit, and analyzed without dealing with three-dimensional geometry. However, we leave this generalization as an open problem.

1 unit of space in 1 unit of time. A *point in spacetime* is a pair $(L, t) \in [-1, 1] \times \mathbb{R}^{\geq 0}$, combining a spatial point with a time.

Messages. We always consider messages to be directional — i.e. they travel either only to the left, or only to the right — and to travel with speed exactly 1 (the speed of light). We treat this process as the movement of a physical quantum register through space at the speed of light. In general, a message is a quantum state, although we often distinguish between the classical and quantum parts of a given message (for example, a party may send a string x and a qubit $|\psi\rangle$ in the same message).

Algorithms. We define the following model of computation in spacetime. An algorithm consists of a finite number of movable "cells", which are quantum registers labeled $\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_M$. Corresponding to each cell is a predetermined, continuous trajectory through space, which we'll label $T_1, \ldots, T_M : \mathbb{R}^{\geq 0} \to [-1, 1]$. $T_i(t)$ denotes the spatial location of cell $\mathbf{R}_i$ at time t. Cells can move at any speed up to the speed of light. Inputs are given to an algorithm by setting the initial value of some cells designated as input cells. Additionally, for each message that could possibly be sent or received by the algorithm, we add a cell which travels along the same path as this message, in perpetuity or until the message goes "out of bounds", in which case the cell lingers at -1 or 1. Prior to the departure time of a message, its corresponding cell will simply be dormant in the message's starting location. The reason for this addition will become clear in the next paragraph.

The computational units of an algorithm are contained at the intersection points between its cells[10] — let X be the set containing all of these points. At each intersection point $(L, t) \in X$, the algorithm contains a quantum circuit $C_{L,t}$ acting on all of the cells which intersect there. $C_{L,t}$ can implement any unitary on these cells, and the output registers can be arbitrarily subdivided among the participating cells. The size of a circuit, denoted $|C_{L,t}|$, refers to the number of gates used to implement it, taken from some fixed universal quantum gate set. No matter the size of a circuit, it is always assumed to be an instantaneous operation. An algorithm is said to eventually *terminate* if X is finite, and it is said to have terminated whenever the last intersection point is reached. At termination, some of its cells are designated as outputs, and the rest discarded. As mentioned above, all messages are also treated as cells — to send a message, the algorithm transfers some information into the cell having that message's trajectory, which occurs at the intersection point corresponding to the message departing. To receive a message, an algorithm might place a SWAP circuit between a message cell and another ancilla cell, for example.

We sometimes specify that a family of algorithms $\{\mathcal{A}_\lambda\}_\lambda$ is QPT, which is defined with respect to a security parameter, and abbreviated as "$\mathcal{A}$ is QPT" if dependence on λ is clear from context. Let X_λ and $\{C_\lambda^{L,t}\}_{(L,t)\in X_\lambda}$ be the sets of intersection points and circuits, respectively, for $\mathcal{A}_\lambda$. QPT is then shorthand for the following requirement: there exists some polynomial $p(\lambda)$ such that for all $\lambda \in \mathbb{N}$,

1. $\mathcal{A}_\lambda$ terminates, and
2. $\sum_{(L,t)\in X_\lambda} |C_\lambda^{L,t}| \leq p(\lambda)$.

Positional Protocols. Our protocols take place between two kinds of parties: provers and verifiers. As stated above, the verifiers always consist of two parties $\mathcal{V}_L$ and $\mathcal{V}_R$, whose locations are

[10] This restriction is without loss of generality. Since we assume – as is standard in QPV literature – that all computation is instantaneous, any computation which is being done locally at one cell, whether the cell is stationary or moving, can always be freely pushed forward or backward to the nearest intersection points with other cells.

publicly known to be -1 and 1, respectively. Provers are, in general, allowed to have circuits anywhere in $[-1, 1]$. The protocol typically includes a Setup operation, whose outputs are distributed between these parties unequally. We clarify this distinction, as well as other distinctions between provers and verifiers, below:

- All verifiers can:
  - Take as input some secret information sp from Setup, in addition to the public information pp. In our protocols, these are both classical.
  - Send classical or quantum messages to any position (L, t).
  - Send classical or quantum messages to other verifiers through private, trusted channels.
- A prover can:
  - Take as input some prover-specific information from Setup. In our protocols, this will always be a quantum state ρ.
  - Send classical or quantum messages to any position (L, t).
  - Send classical or quantum messages between its various distributed components through private, trusted channels.
- We assume that the prover's trajectory is a continuous function $L(\cdot)$ with $L(t) \in [-1, 1]$ and $\left|\frac{L(t_1)-L(t_0)}{t_1-t_0}\right| \leq 1$ for all $t, t_0, t_1 \in \mathbb{R}^{\geq 0}$. The second condition is just the speed of light constraint, as measured by the secant line of the trajectory.

The following lemma is imported from [Unr14]. It states that no adversary can distinguish an oracle reprogramming on the position decided by the xor of random domain element sampled by $\mathcal{V}_L, \mathcal{V}_R$ if one of the domain elements hasn't reach the adversary.

Lemma 3.1 ([Unr14])**.** *Let $\mathcal{O}_{\mathsf{random}}$ be a random oracle with exponential domain and arbitrary image size and the size is a power of 2. Any QPT prover cannot distinguish between the following two cases with noticeable probability:*

- *$\mathcal{V}_L$ samples a uniform random vector x_l and broadcasts it at time t_l. $\mathcal{V}_R$ samples a uniform random vector x_r and broadcasts it at time t_r.*
- *$\mathcal{V}_L$ samples a uniform random vector x_l and broadcasts it at time t_l. $\mathcal{V}_R$ samples a uniform random vector x_r and broadcasts it at time t_r. Then the oracle $\mathcal{O}_{\mathsf{random}}$ is reprogrammed to a uniform random output y on input $x_l \oplus x_r$ for all space-time coordinates (p, t) within the region:*
  - *$t - t_l \geq p + 1$.*
  - *$t - t_r \geq 1 - p$.*

3.3 Specifying Quantum Information in Spacetime

We introduce notation, $\mathsf{register}[\cdot](\cdot)$, for convenience in formalizing many of our extraction-based security definitions in this paper. A definition is given below.

Definition 3.2. *Let $\mathsf{Exp}_{\mathcal{P},V}$ refer to some experiment, taking place between a prover $\mathcal{P}$ and verifier $\mathcal{V}$, and let $R_1, \ldots, R_k$ be disjoint regions of spacetime. Then, we use the following notation,*

$$\mathbf{A}_1, \ldots, \mathbf{A}_k \leftarrow \mathsf{register}[R_1; \ldots; R_k](\mathsf{Exp}_{\mathcal{P},V}) \ ,$$

to mean that for $i \in [k]$, $\mathbf{A}_i$ is the tensor product of the following:

1. *All of $\mathcal{P}$'s registers within region R_i.*

2. *All message registers currently inflight within region R_i.*
3. *The transcript of all classical messages sent by $\mathcal{V}$ which causally precede any points in R_i. In other words, any verifier messages which would be "heard" within R_i.*

Note that this may not correspond to a physical operation – in particular, if R contains points which are at the same spatial location at different times, these might have states which cannot physically coexist – but we still allow this to be defined for notational convenience. Also, when convenient, we will use the notation "$S \circledcirc t$", where S is a spatial region and t a time, to refer to the spacetime region $S \times \{t\}$.

3.4 Compressed Oracle

In this subsection, we recall the technique introduced by Zhandry [Zha19]. For more details please refer to [Zha19]. The following part is adapted from [HHL26]. We will show two equivalent oracle forms: the standard oracle and the compressed oracle. Note that Fourier basis is considered in other papers that use this technique but here we use Hadamard basis instead for simplicity.

We first model an oracle quantum algorithm. It consists of the following registers:

- **X** is the register that stores either an oracle query or an answer waiting to be written.
- **U** is the register that stores the oracle's response or is used to store phase for the output process (will explain later).
- **W** is the register that stores ancilla qubits in the computation.

Standard oracle. Let $\mathcal{O} : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a random oracle. We can view an algorithm that runs in the random oracle model with respect to $\mathcal{O}$ as the algorithm itself concatenated with a random oracle register **D** that is initialized to $\sum_{\mathcal{O}} |\mathcal{O}\rangle \langle\mathcal{O}|_{\mathbf{D}}$ (ignoring the normalizing factor). The register **D** stores the random function $|\mathcal{O}\rangle_{\mathbf{D}} = |\mathcal{O}(0^n)\rangle \left|\mathcal{O}(0^{n-1}1)\right\rangle \cdots |\mathcal{O}(1^n)\rangle$. The oracle unitary StO can be written as follows:

$$\mathsf{StO} \, |x\rangle_{\mathbf{X}} |u\rangle_{\mathbf{U}} |w\rangle_{\mathbf{W}} \otimes |\mathcal{O}\rangle_{\mathbf{D}} = |x\rangle_{\mathbf{X}} |u + \mathcal{O}(x)\rangle_{\mathbf{U}} |w\rangle_{\mathbf{W}} \otimes |\mathcal{O}\rangle_{\mathbf{D}} ,$$

The following lemma shows that the output distribution using a standard oracle is exactly the same as using a random oracle.

Lemma 3.3 ([Zha19, Lemma 2])**.** *Let $\mathcal{A}$ be an (unbounded) quantum algorithm making oracle queries. The output of $\mathcal{A}$ given a random function $\mathcal{O}$ is exactly identical to the output of $\mathcal{A}$ given access to a standard oracle. Therefore, a random oracle with quantum query access can be perfectly simulated as a standard oracle.*

Compressed oracle. The compressed oracle can be viewed a type of lazy sampling technique. Instead of initializing $\mathcal{O}$ at the very beginning, the compressed oracle creates a database $|D\rangle_{\mathbf{D}} = |D(0^n)\rangle_{\mathbf{D}_{0^n}} \left|D(0^{n-1}1)\right\rangle_{\mathbf{D}_{0^{n-1}1}} \cdots |D(1^n)\rangle_{\mathbf{D}_{1^n}}$ where $D(x) \in \mathbb{F}_2^m \cup \{\perp\}$ and $|D\rangle$ is initialized to $|\emptyset\rangle_{\mathbf{D}} = |\perp, \perp, \cdots, \perp\rangle$ where $\perp$ is a symbol that indicates the lack of information of the algorithm on certain function value. Let $|D|$ denote the number of entries in D that are not $\perp$. The database is initialized as an empty list D_0 of length N, in other words, it is initialized as the pure state $|\emptyset\rangle := |\perp, \perp, \cdots, \perp\rangle$.

For any D and x such that $D(x) = \perp$, we define $D' = D \cup (x, u)$ to be the database D', such that for every $x' \neq x$, $D'(x') = D(x)$ and at the input x, $D'(x) = u$.

The compressed oracle is the unitary $\mathsf{CStO} := \mathsf{StdDecomp} \cdot \mathsf{CStO}' \cdot \mathsf{StdDecomp}$, where

- CStO' writes $D(x)$ to the answer register $\mathbf{U}$ by writing $u + D(x)$ into it when $D(x) \neq \perp$ as usual but does nothing when $D(x) = \perp$. Or to say that we can define addition for $\perp$: $u + \perp = u, \forall u \in \mathbb{F}_2^m$. Formally,

$$\mathsf{CStO}' |x\rangle_{\mathbf{X}} |u\rangle_{\mathbf{U}} |w\rangle_{\mathbf{W}} \otimes |D\rangle_{\mathbf{D}} = |x\rangle_{\mathbf{X}} |u + D(x)\rangle_{\mathbf{U}} |w\rangle_{\mathbf{W}} \otimes |D\rangle_{\mathbf{D}} .$$

- When the algorithm queries, the database calls $\mathsf{StdDecomp}$ which unfolds the database and samples a value y for positions that the algorithm does not know what the value is. More specifically, $\mathsf{StdDecomp} |x\rangle_{\mathbf{X}} |u\rangle_{\mathbf{U}} |w\rangle_{\mathbf{W}} \otimes |D\rangle_{\mathbf{D}} := |x\rangle_{\mathbf{X}} |u\rangle_{\mathbf{U}} |w\rangle_{\mathbf{W}} \otimes \mathsf{StdDecomp}_x |D\rangle_{\mathbf{D}}$, where $\mathsf{StdDecomp}_x$ works on $\mathbf{D}_x$.
  - If $D(x) = \perp$, $\mathsf{StdDecomp}_x$ maps $|\perp\rangle$ to

$$\frac{1}{\sqrt{2^m}} \sum_{y \in \mathbb{F}_2^m} |y\rangle .$$

  - If $D(x) \neq \perp$, $\mathsf{StdDecomp}_x$ works on the x-th register, and it is an identity on

$$\frac{1}{\sqrt{2^m}} \sum_{y \in \mathbb{F}_2^m} (-1)^{\langle u,y \rangle} |y\rangle$$

    for all $u \neq 0$; it maps the uniform superposition $\frac{1}{\sqrt{2^m}} \sum_{y \in \mathbb{F}_2^m} |y\rangle$ to $|\perp\rangle$.
    More formally, for a D' such that $D'(x) = \perp$,

$$\mathsf{StdDecomp}_x \frac{1}{\sqrt{2^m}} \sum_{y \in \mathbb{F}_2^m} (-1)^{\langle u,y \rangle} \left|D' \cup (x,y)\right\rangle_{\mathbf{D}} = \frac{1}{\sqrt{2^m}} \sum_{y \in \mathbb{F}_2^m} (-1)^{\langle u,y \rangle} \left|D' \cup (x,y)\right\rangle_{\mathbf{D}}$$

    for any $u \neq 0$ and,

$$\mathsf{StdDecomp}_x \frac{1}{\sqrt{2^m}} \sum_{y \in \mathbb{F}_2^m} \left|D' \cup (x,y)\right\rangle_{\mathbf{D}} = \left|D'\right\rangle_{\mathbf{D}} .$$

  Intuitively, it swaps a uniform superposition $\frac{1}{\sqrt{2^m}} \sum_{y \in \mathbb{F}_2^m} |y\rangle$ with $|\perp\rangle$ on $\mathbf{D}_x$ and does nothing on other orthogonal basis. So it is a well defined unitary.

Note that the compressed oracle can be implemented in an efficient way where we don't create $\mathbf{D}_x$ for all x and we only store all non-$\perp$ entries. For more details, please refer to [Zha19]. Zhandry proves that, StO and CStO are perfectly indistinguishable to any *unbounded* quantum algorithm.

Lemma 3.4 ([Zha19, Lemma 4])**.** *Let $\mathcal{A}$ be an (unbounded) quantum algorithm making oracle queries. The output of $\mathcal{A}$ given access to the standard oracle is exactly identical to the output of $\mathcal{A}$ given access to a compressed oracle.*

4 The Quantum Anchor State

In this section we introduce the quantum anchor state, a structured bipartite quantum state that will serve as a basic primitive throughout the paper. Intuitively, the generation process starts from several EPR pairs shared between two registers ⚓ and 🏴‍☠️. Some 'decoy' qubits are then tensored to the 🏴‍☠️ register. A random unitary U_{shift} is applied to 🏴‍☠️ to 'scramble' and hide the entanglement

among all qubits in 🏴‍☠️. This ensures that once 🏴‍☠️ is given to another party who does not know U_{shift}, it cannot separate the part of 🏴‍☠️ that is entangled with ⚓ from the other 'decoy' qubits.

If not specified, in remaining paragraphs we consider vectors and matrices with elements in $\mathbb{F}_2$. Normally, we use $\mathcal{O}_S^b$, where S is a set of elements (subsets, cosets, subspaces) and b is any object (such as symbols, bit strings, and field elements), to denote the oracle that outputs b on input $x \in S$ and outputs $\perp$ otherwise. When $b = \top$, it is called a membership oracle and we write $\mathcal{O}_S$ for simplicity. Also, if not specified, the domain of $\mathcal{O}_S^b$ is the natural domain depending on S. For example, when S is a subset/coset/subspace of $\mathbb{F}_2^n$, then the domain is $\mathbb{F}_2^n$. We abuse notations like $\{0,1\}^n$ here so they can interact with field element such as $x \in \mathbb{F}_2^n$. For example, we use $0^n \times x$ to denote a vector in $\mathbb{F}_2^{2n}$ with its first n bits all zeros and its last n bits being x. Another example is that we use $U_{\mathsf{shift}}(0^n \times S)$, where S is a subset/coset/subspace of $\mathbb{F}_2^n$ and U_{shift} is a change of basis matrix, to denote the subset/coset/subspace of $\mathbb{F}_2^{2n}$ consisting of all vectors of the form $U_{\mathsf{shift}}(0^n \times x)$ for $x \in S$.

Remark 4.1. *When we define subspaces in this paper, we use the columns of a matrix to describe the subspace so that we can define the canonical representation of the subspace and its dual. For example, let $A \leq \mathbb{F}_2^n$ be a subspace of dimension m defined by m columns of a $n \times n$ change of basis (invertible) matrix U. Let the set of indices of these columns be C. This means that A is the span of the set of m columns of U with indices in C. Now we define $\mathsf{Can}_A(s)$ and $\mathsf{Can}_{A^\perp}(s')$ to be the canonical representations of the cosets $A + s$ and $A^\perp + s'$:*

- *Let w be the vector obtained by replacing m bits of $U^{-1}s$ with indices in C by zeros. Define $\mathsf{Can}_A(s) := Uw$.*
- *Let w' be the vector obtained by replacing $n - m$ bits of $U^t s'$ with indices not in C by zeros. Define $\mathsf{Can}_{A^\perp}(s') := U^{-t}w'$. Here U^{-t} is defined as $\left(U^t\right)^{-1}$.*

To understand what this definition means, we use the following example. Let A be a $n/2$ dimensional subspace of $\mathbb{F}_2^n$ spanned by the first $n/2$ columns of U_A. Let $x = U_A\left(x_A \times 0^{n/2} + 0^{n/2} \times x_{A^\perp}\right)$ be a vector in $\mathbb{F}_2^n$ where x_A and $x_{A^\perp}$ are vectors in $\mathbb{F}_2^{n/2}$. Then we have that $\mathsf{Can}_A(x) = U_A\left(0^{n/2} \times x_{A^\perp}\right)$ and $\mathsf{Can}_{A^\perp}(x) = U_A^{-t}\left(x_A \times 0^{n/2}\right)$. Define $\mathsf{CS}(A) := \{\mathsf{Can}_A(s) : s \in \mathbb{F}_2^n\}$ and $\mathsf{CS}(A^\perp) := \{\mathsf{Can}_{A^\perp}(s') : s' \in \mathbb{F}_2^n\}$.

Definition 4.2 (Quantum Anchor State)**.** *Let λ be the security parameter and let $n_e(\lambda), n_s(\lambda), n_h(\lambda)$ be polynomials. Take a uniformly random $(n_e + n_h + n_s) \times (n_e + n_h + n_s)$ change of basis (invertible) matrix U_{shift}. The subspaces S, T are described by the first n_h columns and the first $n_h + n_e$ columns of U_{shift}, respectively. The anchor state $\left|\Psi^{\mathsf{sk}}_{n_e,n_h,n_s}\right\rangle_{⚓🏴‍☠️}$ is a bipartite state on a n_e-qubit anchor register ⚓ and a $(n_e + n_h + n_s)$-qubit vessel register 🏴‍☠️. This state is also parameterized by a secret key sk, which we specify later. The state $\left|\Psi^{\mathsf{sk}}_{n_e,n_h,n_s}\right\rangle_{⚓🏴‍☠️}$ is generated in the following way:*

1. *First generate a $2(n_e + n_h + n_s)$-qubit EPR state between $\mathbf{U} \otimes ⚓ \otimes \mathbf{V}$ and 🏴‍☠️. In other words, after this step, the registers $\mathbf{U}$ of size n_h, ⚓ of size n_e, and $\mathbf{V}$ of size n_s are fully entangled (in the form of EPR pairs) with the first n_h qubits of 🏴‍☠️, the middle n_e qubits of 🏴‍☠️, and the last n_s qubits of 🏴‍☠️, respectively.*
2. *Measure $\mathbf{U}$ in the Hadamard basis to obtain a vector $u' \in \mathbb{F}_2^{n_h}$ ($|+\rangle$ is mapped to 0 and $|-\rangle$ is mapped to 1 on each coordinate). Then measure $\mathbf{V}$ in the standard basis to obtain a vector $v' \in \mathbb{F}_2^{n_s}$. Compute $u = U_{\mathsf{shift}}^{-t}\left(u' \times 0^{n_e+n_s}\right)$ and $v = U_{\mathsf{shift}}\left(0^{n_h+n_e} \times v'\right)$.*
3. *Apply the following change of basis unitary $\mathcal{U}_{\mathsf{shift}}$ on 🏴‍☠️:*

$$\mathcal{U}_{\mathsf{shift}} |x\rangle_{🏴‍☠️} \to |U_{\mathsf{shift}}x\rangle_{🏴‍☠️}.$$

This procedure is shown in Figure 5. Define the functionality $\mathsf{GenAnchorState}_{n_e,n_h,n_s}(1^\lambda)$ *as the function that samples random* S, T*, generates* $\left|\Psi^{\mathsf{sk}}_{n_e,n_h,n_s}\right\rangle_{\text{⚓🏴‍☠️}}$*, and outputs* $\left(\mathsf{sk} := (U_{\mathsf{shift}}, u, v), \left|\Psi^{\mathsf{sk}}_{n_e,n_h,n_s}\right\rangle_{\text{⚓🏴‍☠️}}\right)$*. One can see that by storing* U_{shift} *in* sk*, we also store* S, T *in it.*

The generation process creates the following state on ⚓🏴‍☠️:

$$\propto \sum_{x\in\mathbb{F}_2^{n_e}} |x\rangle_{\text{⚓}} \sum_{s\in S} (-1)^{\langle s,u\rangle} \left|s + U_{\mathsf{shift}}\left(0^{n_h} \times x \times 0^{n_s}\right) + v\right\rangle_{\text{🏴‍☠️}}.$$

Remark 4.3. *We may omit the subscript when* $n_e = n_h = n_s = n$ *and* n *is clear from the context. For example,* $\left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}\right\rangle_{\text{⚓🏴‍☠️}}\right) \leftarrow \mathsf{GenAnchorState}(1^\lambda)$ *means* $\left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}_{n,n,n}\right\rangle_{\text{⚓🏴‍☠️}}\right) \leftarrow \mathsf{GenAnchorState}_{n,n,n}(1^\lambda)$ *when* n *is clear from the context.*

Definition 4.4 (Domain Extension and Oracle Simulation)**.** *Define* $\mathsf{DomainExtension}_{n'_h,n'_s}$ *as follows:*

1. *It takes* 1^λ *and works as an isometry on the* 🏴‍☠️ *register of an anchor state* $\left|\Psi^{\mathsf{sk}}_{n_e,n_h,n_s}\right\rangle$*.*
2. *Sample a uniformly random* $(n_e + n_h + n_s + n'_h + n'_s) \times (n_e + n_h + n_s + n'_h + n'_s)$ *change of basis matrix* U^*_{shift}*. Define* S^* *to be the subspace described by the first* n'_h *columns of* U^*_{shift} *and define* T^* *to be the subspace described by the first* $n'_h + n_h + n_e + n_s$ *columns of* U^*_{shift}*.*
3. *Create a* n'_h*-qubit register* $\mathbf{U}^*$ *and a* n'_s*-qubit register* $\mathbf{V}^*$*. Then extend* 🏴‍☠️ *by tensoring it with* n'_h/n'_s *qubits at the front/end that are fully entangled with* $\mathbf{U}^*/\mathbf{V}^*$ *in the form of* EPR *pairs, respectively. The extended register is called* 🏴‍☠️**.*
4. *Measure* $\mathbf{U}^*$ *in the Hadamard basis to obtain a vector* $u'^* \in \mathbb{F}_2^{n'_h}$ *(*$|+\rangle$ *is mapped to* 0 *and* $|-\rangle$ *is mapped to* 1 *on each coordinate). Then measure* $\mathbf{V}^*$ *in the standard basis to obtain a vector* $v'^* \in \mathbb{F}_2^{n'_s}$*. Compute* $u^* = {U^*_{\mathsf{shift}}}^{-t}(u'^* \times 0^{n_h+n_e+n_s+n'_s})$ *and* $v^* = U^*_{\mathsf{shift}}(0^{n'_h+n_h+n_e+n_s} \times v'^*)$*.*
5. *Apply the following change of basis unitary* $\mathcal{U}^*_{\mathsf{shift}}$ *on* 🏴‍☠️**:*

$$\mathcal{U}^*_{\mathsf{shift}} |x\rangle_{\text{🏴‍☠️}^*} \to |U^*_{\mathsf{shift}} x\rangle_{\text{🏴‍☠️}^*}.$$

6. *Let* $\mathsf{sk}^* = (U^*_{\mathsf{shift}}, u^*, v^*)$*. Return* sk^* *and* 🏴‍☠️**.*

Also we define

- $S_{\mathsf{ext}} := U^*_{\mathsf{shift}}\left(\{0,1\}^{n'_h} \times S \times 0^{n'_s}\right)$*, described by the first* $n_h + n'_h$ *columns of* U_{extshift}*. Where* $U_{\mathsf{extshift}} := U^*_{\mathsf{shift}} \begin{pmatrix} I_{n'_h} & & \\ & U_{\mathsf{shift}} & \\ & & I_{n'_s} \end{pmatrix}$*.*
- $T_{\mathsf{ext}} := U^*_{\mathsf{shift}}\left(\{0,1\}^{n'_h} \times T \times 0^{n'_s}\right)$*, described by the first* $n'_h + n_h + n_e$ *columns of* U_{extshift}*.*
- $u_{\mathsf{ext}} := {U^*_{\mathsf{shift}}}^{-t}\left(0^{n'_h} \times u \times 0^{n'_s}\right) + u^*$*.*
- $v_{\mathsf{ext}} := U^*_{\mathsf{shift}}\left(0^{n'_h} \times v \times 0^{n'_s}\right) + v^*$*.*

$\mathsf{GenAnchorState}$ and $\mathsf{DomainExtension}$ are shown in Figure 5. Here are some useful facts that we will use later in the proof.

Fact 4.5. *Let* n_e, n_h, n_s *be polynomials of* λ *and let* $\left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}_{n_e,n_h,n_s}\right\rangle_{\text{⚓🏴‍☠️}}\right) \leftarrow \mathsf{GenAnchorState}_{n_e,n_h,n_s}(1^\lambda)$*. The standard-basis measurement on* 🏴‍☠️ *always outputs a vector in* $T + v$*, and the Hadamard-basis measurement on* 🏴‍☠️ *always outputs a vector in* $S^\perp + u$*. Furthermore, the anchor state has the following symmetric*

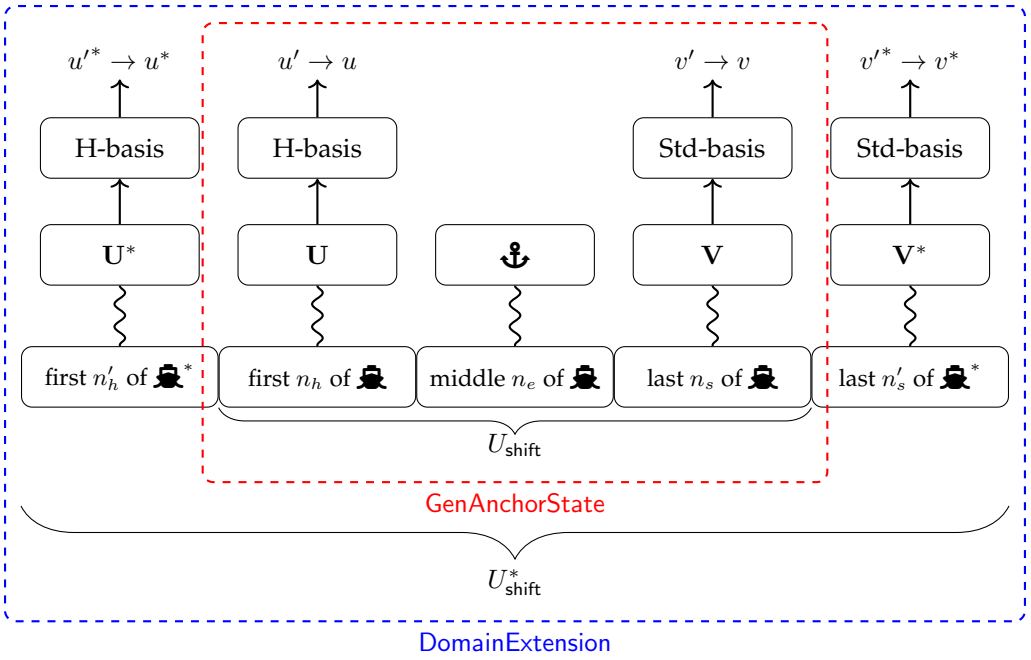

Figure 5: Compact state-generation diagram. GenAnchorState and DomainExtension.

property: the following two distributions are equal for fixed n_e, n_h, n_s.

$$\left\{ S, T, u, v, \left|\Psi^{\mathsf{sk}}_{n_e,n_h,n_s}\right\rangle_{\text{⚓🏴‍☠️}} \middle| \left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}_{n_e,n_h,n_s}\right\rangle_{\text{⚓🏴‍☠️}}\right) \leftarrow \mathsf{GenAnchorState}_{n_e,n_h,n_s}(1^\lambda) \right\}$$
$$= \left\{ T^\perp, S^\perp, v, u, \left(H^{\otimes n_e}_{\text{⚓}} \otimes H^{\otimes(n_e+n_h+n_s)}_{\text{🏴‍☠️}}\right) \left|\Psi^{\mathsf{sk}}_{n_e,n_s,n_h}\right\rangle_{\text{⚓🏴‍☠️}} \middle| \left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}_{n_e,n_s,n_h}\right\rangle_{\text{⚓🏴‍☠️}}\right) \leftarrow \mathsf{GenAnchorState}_{n_e,n_s,n_h}(1^\lambda) \right\}.$$

Fact 4.6. *Let* $n_e, n_h, n_s, n'_h, n'_s$ *be polynomials of* λ. *The following two distributions*

$$\left\{ \begin{matrix} U_{\mathsf{extshift}}, \\ S_{\mathsf{ext}}, T_{\mathsf{ext}}, \\ u_{\mathsf{ext}}, v_{\mathsf{ext}}, \\ \textit{state on } \text{⚓🏴‍☠️}^* \end{matrix} \middle| \begin{matrix} \left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}_{n_e,n_h,n_s}\right\rangle_{\text{⚓🏴‍☠️}}\right) \leftarrow \mathsf{GenAnchorState}_{n_e,n_h,n_s}(1^\lambda) \\ \mathsf{sk}^* \leftarrow \mathsf{DomainExtension}_{n'_h,n'_s}(1^\lambda) \textit{ on } \text{🏴‍☠️} \textit{ to obtain } \text{🏴‍☠️}^* \end{matrix} \right\}$$

and

$$\left\{ \begin{matrix} U_{\mathsf{shift}}, \\ S, T, \\ u, v, \\ \textit{state on } \text{⚓🏴‍☠️} \end{matrix} \middle| \left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}_{n_e,n_h+n'_h,n_s+n'_s}\right\rangle_{\text{⚓🏴‍☠️}}\right) \leftarrow \mathsf{GenAnchorState}_{n_e,n_h+n'_h,n_s+n'_s}(1^\lambda) \right\}$$

are equal.

Fact 4.7. *Let* $\left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}_{n_e,n_h,n_s}\right\rangle_{\text{⚓🏴‍☠️}}\right) \leftarrow \mathsf{GenAnchorState}_{n_e,n_h,n_s}(1^\lambda)$ *and let* sk^* *be generated by* $\mathsf{DomainExtension}_{n'_h,n'_s}(1^\lambda)$ *working on* 🏴‍☠️. *Any party with* sk^* *can do the following:*

- *It can simulate oracle access to* $\mathcal{O}^b_{T^*+v^*}$ *with* b: *on input* z, *check whether* ${U^*_{\mathsf{shift}}}^{-1}(z - v^*)$ *has its last* n'_s *bits equal to zero. Output* b *if so and output* $\perp$ *otherwise.*
- *It can simulate oracle access to* $\mathcal{O}^b_{S^{*\perp}+u^*}$ *with* b: *on input* z, *check whether* ${U^*_{\mathsf{shift}}}^{t}(z - u^*)$ *has its first* n'_h *bits equal to zero. Output* b *if so and output* $\perp$ *otherwise.*
- *It can simulate oracle access to* $\mathcal{O}^b_{T_{\mathsf{ext}}+v_{\mathsf{ext}}}$ *given oracle access to* $\mathcal{O}^b_{T+v}$: *on input* z, *let* z' *be the result of discarding the first* n'_h *bits and the last* n'_s *bits of* ${U^*_{\mathsf{shift}}}^{-1}(z - v^*)$. *If any of the last* n'_s *discarded bits is non-zero, output* $\perp$; *otherwise forward the output of* $\mathcal{O}^b_{T+v}(z')$.

- *It can simulate oracle access to $\mathcal{O}^b_{S^\perp_{\mathsf{ext}}+u_{\mathsf{ext}}}$ given oracle access to $\mathcal{O}^b_{S^\perp+u}$: on input z, let z' be the result of discarding the first n'_h bits and the last n'_s bits of ${U^*_{\mathsf{shift}}}^t(z-u^*)$. If any of the first n'_h discarded bits is non-zero, output $\perp$; otherwise forward the output of $\mathcal{O}^b_{S^\perp+u}(z')$.*

5 Collapsing vs Non-collapsing Oracles

Let $a(\lambda) \le b(\lambda) < c(\lambda) \le d(\lambda)$ be polynomials of λ. Let $W \le \mathbb{F}_2^d$ be a subspace of dimension c. Let $U^{(\cdot)}$ be any efficient isometry with superposition access to a classical oracle. We introduce a subspace oracle indistinguishability lemma below. In the remaining part of this section S, T are sampled so that $T \le W$ is a uniform random subspace of dimension b and $S \le T$ be a uniform random subspace of dimension a.

Lemma 5.1. *Let $\{\mathcal{O}_S\}_S$ be any family of oracles such that $\mathcal{O}_S$ outputs $\perp$ on $z \notin S$. Let $\{\mathcal{O}_{S,T}\}_{S,T}$ be any family of oracles such that $\mathcal{O}_{S,T}$ outputs $\mathcal{O}_S(z)$ on $z \in S$ and outputs $\perp$ on $z \notin T$. There exists a polynomial $q(\lambda)$ such that the following holds. Define $|\psi_S\rangle := U^{\mathcal{O}_S}|S\rangle$, $|\psi_T\rangle := U^{\mathcal{O}_{S,T}}|S\rangle$, $|\phi_S\rangle = \frac{1}{\sqrt{\binom{c}{b}_2\binom{b}{a}_2}}\sum_{S,T}|S,T\rangle|\psi_S\rangle$ and $|\phi_T\rangle = \frac{1}{\sqrt{\binom{c}{b}_2\binom{b}{a}_2}}\sum_{S,T}|S,T\rangle|\psi_T\rangle$ we have*

$$\mathrm{TD}\left(|\phi_S\rangle\langle\phi_S|, |\phi_T\rangle\langle\phi_T|\right) \le \frac{q}{2^{(c-b)/2}}.$$

Proof. We prove it by hybrid argument. Let the number of oracle queries of $U^{\mathcal{O}}$ be $q'(\lambda)$. Define $\left|\psi^i\right\rangle$ to be the state after applying $U^{\mathcal{O}}$ on $|S\rangle$, where the first i queries use oracle $\mathcal{O}_S$ and other queries use oracle $\mathcal{O}_{S,T}$. We have $\left|\psi^0\right\rangle = |\psi_T\rangle$ and $\left|\psi^{q'}\right\rangle = |\psi_S\rangle$. Similarly we define $\left|\phi^i\right\rangle = \frac{1}{\sqrt{\binom{c}{b}_2\binom{b}{a}_2}}\sum_{S,T}|S,T\rangle\left|\psi^i\right\rangle$. To use hybrid argument, we need the following claim.

Claim 5.2. *For all $0 \le i < q'$,*

$$\mathrm{TD}\left(\left|\phi^i\right\rangle\left\langle\phi^i\right|, \left|\phi^{i+1}\right\rangle\left\langle\phi^{i+1}\right|\right) \le \frac{2}{2^{(c-b)/2}}.$$

Proof. Let $\left|\psi^{\le i}\right\rangle$ be the state of applying $U^{\mathcal{O}_S}$ on $|S\rangle$ but stop before the $i+1$-th oracle query. Define $\Pi_{T\setminus S}$ to be the projector onto states where the value z on the oracle query register is in T but is not in S. We have

$$\begin{aligned}
&\mathrm{TD}\left(\left|\phi^i\right\rangle\left\langle\phi^i\right|, \left|\phi^{i+1}\right\rangle\left\langle\phi^{i+1}\right|\right)\\
=&\frac{1}{\binom{c}{b}_2\binom{b}{a}_2}\mathrm{TD}\left(\left(\sum_{S,T}|S,T\rangle\mathcal{O}_{S,T}\left|\psi^{\le i}\right\rangle\right)\left(\sum_{S,T}\langle S,T|\left\langle\psi^{\le i}\right|\mathcal{O}^\dagger_{S,T}\right), \left(\sum_{S,T}|S,T\rangle\mathcal{O}_S\left|\psi^{\le i}\right\rangle\right)\left(\sum_{S,T}\langle S,T|\left\langle\psi^{\le i}\right|\mathcal{O}^\dagger_S\right)\right)\\
=&\sqrt{1-\left|\mathop{\mathbb{E}}_{S,T}\left[\left\langle\psi^{\le i}\right|\mathcal{O}^\dagger_{S,T}\mathcal{O}_S\left|\psi^{\le i}\right\rangle\right]\right|^2}\\
=&\sqrt{1-\left|\mathop{\mathbb{E}}_{S,T}\left[\left\langle\psi^{\le i}\right|\Pi_{T\setminus S}\mathcal{O}^\dagger_{S,T}\mathcal{O}_S\Pi_{T\setminus S}\left|\psi^{\le i}\right\rangle + \left\langle\psi^{\le i}\right|(I-\Pi_{T\setminus S})\mathcal{O}^\dagger_{S,T}\mathcal{O}_S(I-\Pi_{T\setminus S})\left|\psi^{\le i}\right\rangle\right]\right|^2}\\
\le&\sqrt{1-\left(\left|\mathop{\mathbb{E}}_{S,T}\left[\left\langle\psi^{\le i}\right|(I-\Pi_{T\setminus S})\mathcal{O}^\dagger_{S,T}\mathcal{O}_S(I-\Pi_{T\setminus S})\left|\psi^{\le i}\right\rangle\right]\right| - \left|\mathop{\mathbb{E}}_{S,T}\left[\left\langle\psi^{\le i}\right|\Pi_{T\setminus S}\mathcal{O}^\dagger_{S,T}\mathcal{O}_S\Pi_{T\setminus S}\left|\psi^{\le i}\right\rangle\right]\right|\right)^2}
\end{aligned}$$

$$\leq \sqrt{1-\left(1-\frac{2}{2^{c-b}}\right)^2}$$
$$\leq \frac{2}{2^{(c-b)/2}}.$$

Now we explain these (in)equations:

- The equation between line 2 and line 3 is by the definition of trace distance between pure state
$$\mathrm{TD}\left(|\phi\rangle\langle\phi|, |\psi\rangle\langle\psi|\right) = \sqrt{1-|\langle\phi|\psi\rangle|^2}.$$
And also notice that all cross terms for different S, T in the inner product is zero because $\langle S, T|S', T'\rangle = 0$ for $(S, T) \neq (S', T')$.
- The equation between line 3 and line 4 is because that the cross terms, for example,
$$\left\langle\psi^{\leq i}\right| (I - \Pi_{T\setminus S})\mathcal{O}_{S,T}^\dagger \mathcal{O}_S \Pi_{T\setminus S} \left|\psi^{\leq i}\right\rangle$$
is zero because $\Pi_{T\setminus S}$ and $I - \Pi_{T\setminus S}$ maps the state to disjoint basis on the query register.
- The inequality between line 4 and line 5 is because $|a+b|^2 \geq (|b|-|a|)^2$.
- The inequality between line 5 and line 6 is because that $\mathcal{O}_{S,T}$ and $\mathcal{O}_S$ is the same oracle if the input state is in $I - \Pi_{T\setminus S}$. We have
$$\left|\mathop{\mathbb{E}}_{S,T}\left[\left\langle\psi^{\leq i}\right| (I - \Pi_{T\setminus S})\mathcal{O}_{S,T}^\dagger \mathcal{O}_S (I - \Pi_{T\setminus S}) \left|\psi^{\leq i}\right\rangle\right]\right|$$
$$= \mathop{\mathbb{E}}_{S,T}\left[\left|(I - \Pi_{T\setminus S})\left|\psi^{\leq i}\right\rangle\right|^2\right] \geq 1 - \frac{1}{2^{c-b}}$$
and
$$\left|\mathop{\mathbb{E}}_{S,T}\left[\left\langle\psi^{\leq i}\right| \Pi_{T\setminus S}\mathcal{O}_{S,T}^\dagger \mathcal{O}_S \Pi_{T\setminus S} \left|\psi^{\leq i}\right\rangle\right]\right|$$
$$\leq \mathop{\mathbb{E}}_{S,T}\left[\left|\Pi_{T\setminus S}\left|\psi^{\leq i}\right\rangle\right|^2\right] \leq \frac{1}{2^{c-b}}.$$
The last steps of both calculation are because that $\left|\psi^{\leq i}\right\rangle$ is independent of T thus the average weight on $\Pi_{T\setminus S}$ is at most $\frac{1}{2^{c-b}}$.
- The inequality between line 6 and line 7 is because $\sqrt{1-(1-x)^2} \leq \sqrt{2x}$.

□

Setting $q(\lambda) = 2q'(\lambda)$, we see that the lemma follows. □

Let $m(\lambda)$ be a polynomial. Now we consider the case that $c = d$ and let w be a uniform random vector in $\mathbb{F}_2^c$. We further define some oracles:

- $\mathcal{O}_{\mathsf{any}} : \mathbb{F}_2^c \to \mathbb{F}_2^m$ be an oracle that is sampled from a distribution depending on S, T, w such that every output alone is independent of T given S, w. (But maybe with two outputs of inputs, one can recover S, T, w)
- $\mathcal{O}_{T+w}^{\mathsf{ncol}} : \mathbb{F}_2^c \to \mathbb{F}_2^m \cup \{\perp\}$ on input z checks if it is in $T + w$. It returns $\perp$ if it is not. Otherwise, it returns the fixed vector $\mathcal{O}_{\mathsf{any}}(\mathsf{Can}_S(w))$, disregarding the input z.

- $\mathcal{O}^{\mathsf{col}}_{T+w} : \mathbb{F}_2^c \to \mathbb{F}_2^m \cup \{\bot\}$ on input z checks if it is in $T+w$. It returns $\bot$ if it is not. Otherwise, it returns $\mathcal{O}_{\mathsf{any}}(\mathsf{Can}_S(z))$.

Intuitively, $\mathcal{O}_{\mathsf{any}}$ can just be thought of a random oracle mapping cosets of S to length-m labels, while the other oracles are two types of membership oracles for $T+w$. The non-collapsing oracle returns the single fixed label associated with $\mathsf{Can}_S(w)$ for every accepted query, whereas the collapsing oracle returns the particular label associated with z through $\mathcal{O}_{\mathsf{any}}(\mathsf{Can}_S(z))$.

Lemma 5.3. *For any efficient isometry*

$$\mathcal{U}^{\mathcal{O}} : \mathcal{H}_{\mathbf{S}} \otimes \mathcal{H}_{\mathbf{W}} \to \mathcal{H}_{\mathbf{Z}}$$

with oracle access to $\mathcal{O}$ where $\mathcal{O}$ is either $\mathcal{O}^{\mathsf{ncol}}_{T+w}$ or $\mathcal{O}^{\mathsf{col}}_{T+w}$. There exists a polynomial $q(\lambda)$ such that let $\left|\psi^{\mathsf{ncol}}_{T+w}\right\rangle = \mathcal{U}^{\mathcal{O}^{\mathsf{ncol}}_{T+w}} \left|S\right\rangle_{\mathbf{S}} \left|w\right\rangle_{\mathbf{W}}$ and $\left|\psi^{\mathsf{col}}_{T+w}\right\rangle = \mathcal{U}^{\mathcal{O}^{\mathsf{col}}_{T+w}} \left|S\right\rangle_{\mathbf{S}} \left|w\right\rangle_{\mathbf{W}}$. We have

$$\mathop{\mathbb{E}}_{S,T,w,\mathcal{O}_{\mathsf{any}}} \left[\mathrm{TD}\left(\left|\psi^{\mathsf{ncol}}_{T+w}\right\rangle\left\langle\psi^{\mathsf{ncol}}_{T+w}\right|, \left|\psi^{\mathsf{col}}_{T+w}\right\rangle\left\langle\psi^{\mathsf{col}}_{T+w}\right|\right)\right] \leq \frac{q}{2^{(c-b)/2}}.$$

Proof. We prove it by hybrid argument.

Hybrid 0: This is the hybrid corresponding to $\left|\psi^{\mathsf{ncol}}_{T+w}\right\rangle = \mathcal{U}^{\mathcal{O}^{\mathsf{ncol}}_{T+w}} \left|S\right\rangle_{\mathbf{S}} \left|w\right\rangle_{\mathbf{W}}$.

Hybrid 1: In this hybrid, the final state is $\left|\psi^{\mathsf{ncol}}_{S+w}\right\rangle = \mathcal{U}^{\mathcal{O}^{\mathsf{ncol}}_{S+w}} \left|S\right\rangle_{\mathbf{S}} \left|w\right\rangle_{\mathbf{W}}$, where the new oracle is defined as:

- $\mathcal{O}^{\mathsf{ncol}}_{S+w} : \mathbb{F}_2^c \to \mathbb{F}_2^m \cup \{\bot\}$ on input z checks if it is in $S+w$. It returns $\bot$ if it is not. Otherwise, it returns $\mathcal{O}_{\mathsf{any}}(\mathsf{Can}_S(w))$.

Claim 5.4. *There exists polynomial $q_0(\lambda)$ such that*

$$\mathop{\mathbb{E}}_{S,T,w,\mathcal{O}_{\mathsf{any}}} \left[\mathrm{TD}\left(\left|\psi^{\mathsf{ncol}}_{T+w}\right\rangle\left\langle\psi^{\mathsf{ncol}}_{T+w}\right|, \left|\psi^{\mathsf{ncol}}_{S+w}\right\rangle\left\langle\psi^{\mathsf{ncol}}_{S+w}\right|\right)\right] \leq \frac{q_0}{2^{(c-b)/2}}.$$

Proof. T is a uniform random subspace containing S given S, w. Also the output of $\mathcal{O}^{\mathsf{ncol}}_{S+w}$ is independent of T. Thus by Lemma 5.1 there exists such polynomial $q_0(\lambda)$. The reason that we can call this lemma is because w is given at the beginning so $\mathcal{O}^{\mathsf{ncol}}_{T+w}$ is equivalent to $\mathcal{O}^{\mathsf{ncol}}_{T}(\cdot - w)$ and $\mathcal{O}^{\mathsf{ncol}}_{S+w}$ is equivalent to $\mathcal{O}^{\mathsf{ncol}}_{S}(\cdot - w)$ where for $A = S, T$ we define

- $\mathcal{O}^{\mathsf{ncol}}_{A} : \mathbb{F}_2^c \to \mathbb{F}_2^m \cup \{\bot\}$ on input z checks if it is in A. It returns $\bot$ if it is not. Otherwise, it returns $\mathcal{O}_{\mathsf{any}}(\mathsf{Can}_S(w))$.

While $\left\{\mathcal{O}^{\mathsf{ncol}}_{S}\right\}_S$ and $\left\{\mathcal{O}^{\mathsf{ncol}}_{T}\right\}_T$ satisfy the requirement in Lemma 5.1. □

Hybrid 2: In this hybrid, the final state is $\left|\psi^{\mathsf{col}}_{S+w}\right\rangle = \mathcal{U}^{\mathcal{O}^{\mathsf{col}}_{S+w}} \left|S\right\rangle_{\mathbf{S}} \left|w\right\rangle_{\mathbf{W}}$, where the new oracle is defined as:

- $\mathcal{O}^{\mathsf{col}}_{S+w} : \mathbb{F}_2^c \to \mathbb{F}_2^m \cup \{\bot\}$ on input z checks if it is in $S+w$. It returns $\bot$ if it is not. Otherwise, it returns $\mathcal{O}_{\mathsf{any}}(\mathsf{Can}_S(z))$.

Claim 5.5. *For all $S, T, w, \mathcal{O}_{\mathsf{any}}$,*

$$\mathrm{TD}\left(\left|\psi^{\mathsf{ncol}}_{S+w}\right\rangle\left\langle\psi^{\mathsf{ncol}}_{S+w}\right|, \left|\psi^{\mathsf{col}}_{S+w}\right\rangle\left\langle\psi^{\mathsf{col}}_{S+w}\right|\right) = 0.$$

Proof. For any vector $z \in S+w$, $S+w$ and $S+z$ are the same coset. These two hybrids are identical because $\mathcal{O}_{\mathsf{any}}(\mathsf{Can}_S(S+w)) = \mathcal{O}_{\mathsf{any}}(\mathsf{Can}_S(S+z))$. □

Hybrid 3: In this hybrid, the final state is $\left|\psi^{\mathsf{col}}_{T+w}\right\rangle = \mathcal{U}^{\mathcal{O}^{\mathsf{col}}_{T+w}} |S\rangle_{\mathbf{S}} |w\rangle_{\mathbf{W}}$.

Claim 5.6. *There exists polynomial $q_1(\lambda)$ such that*

$$\mathop{\mathbb{E}}_{S,T,w,\mathcal{O}_{\mathsf{any}}} \left[\mathrm{TD}\left(\left|\psi^{\mathsf{col}}_{S+w}\right\rangle\left\langle\psi^{\mathsf{col}}_{S+w}\right|, \left|\psi^{\mathsf{col}}_{T+w}\right\rangle\left\langle\psi^{\mathsf{col}}_{T+w}\right|\right)\right] \leq \frac{q_1}{2^{(c-b)/2}}.$$

Proof. T is a uniform random subspace containing S given S, w. Also the output of $\mathcal{O}^{\mathsf{col}}_{S+w}$ is independent of T. Thus by Lemma 5.1 there exists such polynomial $q_1(\lambda)$. The reason is similar to the one mentioned in the proof of Claim 5.4. □

Combine all above, let $q = q_0 + q_1$, we proved the lemma. □

6 Monogamy-of-Entanglement Games

The purpose of this section is to prove properties of the following game.

Definition 6.1 (Multi-Stage Decision/Search Monogamy-of-Entanglement Game)**.** *Let $\lambda \in \mathbb{N}$ be the security parameter and let $n(\lambda) \geq \lambda$, and $n_e(\lambda), m_0(\lambda), m_1(\lambda)$ be polynomials. Consider the following game between the challenger and an adversary $\mathcal{A} = (\mathcal{A}^0_M, \mathcal{A}^1_M, \mathcal{A}^2_M, \mathcal{A}^0_L, \mathcal{A}^1_L, \mathcal{A}^0_R, \mathcal{A}^1_R)$:*

1. *The challenger generates $\left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}_{n_e,n,n}\right\rangle_{⚓🏴‍☠️}\right) \leftarrow \mathsf{GenAnchorState}_{n_e,n,n}(1^\lambda)$ and gives the register 🏴‍☠️ to $\mathcal{A}^0_M$. All parties of the adversary are given oracle access to $\mathcal{O}_{T+v}$ and $\mathcal{O}_{S^\perp+u}$.*
2. *$\mathcal{A}^0_M$ can choose to abort in this step. If it aborts, the output of the game is $\perp$. Otherwise, $\mathcal{A}^0_M$ generates a tripartite state on $\mathbf{LMR}$. It sends $\mathbf{L}$ to $\mathcal{A}^0_L$, sends $\mathbf{M}$ to $\mathcal{A}^1_M$ and sends $\mathbf{R}$ to $\mathcal{A}^0_R$.*
3. *$\mathcal{A}^1_M$ is given the access to $\mathcal{O}^{b_0}_{T+v}$ for random $b_0 \xleftarrow{\$} \mathbb{F}_2^{m_0}$. We can imagine implementing this oracle in a purified manner, where we create a register that is initialized to $|+_{\mathbf{B}}\rangle_{\mathbf{B}}$, the uniform superposition over all b_0, and $\mathcal{A}^1_M$ is able to perform the following operation in order to access $\mathbf{B}$:*

$$\sum_{b_0 \in \mathbb{F}_2^{m_0}} |b_0\rangle\langle b_0|_{\mathbf{B}} \otimes \mathcal{O}^{b_0}_{T+v}.$$

   *$\mathcal{A}^1_M$ with access to this oracle and $\mathbf{M}$ generates a tripartite state on $\mathbf{M}_L\mathbf{M}'\mathbf{M}_R$. It sends $\mathbf{M}_L$ to $\mathcal{A}^0_L$, sends $\mathbf{M}'$ to $\mathcal{A}^2_M$ and sends $\mathbf{M}_R$ to $\mathcal{A}^0_R$. We will name the time just after all operations of $\mathcal{A}^1_M$ are complete and before renaming the registers and splitting them as **BeforeSplit**.*
4. *$\mathcal{A}^0_L$ and $\mathcal{A}^0_R$ are given access to $\mathcal{O}^{b_0}_{T+v}$. $\mathcal{A}^0_L$ on $\mathbf{LM}_L$ produces answer b^l_0 and a state on register $\mathbf{L}'$ that is sent to $\mathcal{A}^1_L$. $\mathcal{A}^0_R$ on $\mathbf{RM}_R$ produces answer b^r_0 and a state on register $\mathbf{R}'$ that is sent to $\mathcal{A}^1_R$.*
5. *$\mathcal{A}^2_M$ is given access to $\mathcal{O}^{b_0}_{T+v}$ and $\mathcal{O}^{b_1}_{S^\perp+u}$ for random $b_1 \xleftarrow{\$} \mathbb{F}_2^{m_1}$. It generates a bipartite state on $\mathbf{M}'_L\mathbf{M}'_R$. $\mathbf{M}'_L$ is given to $\mathcal{A}^1_L$ and $\mathbf{M}'_R$ is given to $\mathcal{A}^1_R$.*
6. *$\mathcal{A}^1_L$ and $\mathcal{A}^1_R$ are given access to $\mathcal{O}^{b_0}_{T+v}$ and $\mathcal{O}^{b_1}_{S^\perp+u}$. $\mathcal{A}^1_L$ on $\mathbf{L}'\mathbf{M}'_L$ generates the answer b^l_1. $\mathcal{A}^1_R$ on $\mathbf{R}'\mathbf{M}'_R$ generates the answer b^r_1. The adversary wins iff $b^l_0 = b^r_0 = b_0$ and $b^l_1 = b^r_1 = b_1$, where b_0 is obtained by measuring the $\mathbf{B}$ register.*

This game is displayed in Figure 6. Let $\mathsf{MultiStageSearchMonogamy}(\mathcal{A}, 1^\lambda)$ be the random variable that takes value $1/0/\perp$ if the adversary $\mathcal{A}$ wins/loses/aborts in the above game, respectively. Specifically, we use

$\mathsf{MultiStageDecisionMonogamy}(\mathcal{A}, 1^\lambda)$ *to denote the same random variable* ***when*** $m_0(\lambda) = 1$ *(but* m_1 *can still be arbitrary).*

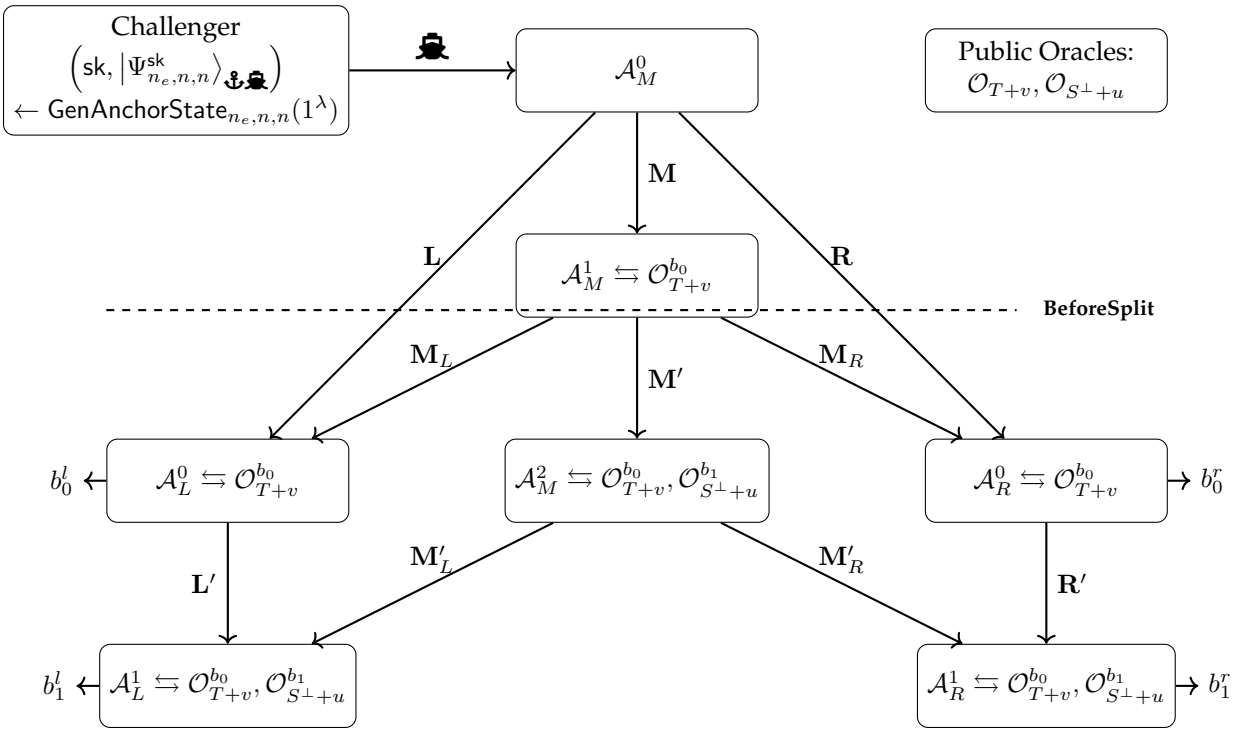

Figure 6: The multi-stage monogamy-of-entanglement game MultiStageSearch-Monogamy/MultiStageDecisionMonogamy.

The main result of this section is Lemma 6.20. Intuitively, it states that any adversary passing the above game with good probability has to recover the value of b_0 by querying its oracle already at the time **BeforeSplit**. Formally, we show that for each value of b_0, the states corresponding to b_0 at **BeforeSplit** are almost orthogonal to each other. Our proof establishing this fact will involve several reductions between various monogamy-of-entanglement games.

6.1 Asymmetric Monogamy-of-Entanglement Games

Theorem 6.2 ([KY25] Theorem 5.4)**.** *Let* $A \leq \mathbb{F}_2^n$ *be a uniformly random subspace of dimension* $n/2$ *described by the first* $n/2$ *columns of a uniformly random change of basis matrix* U*. Let* s *be a vector in* $\mathsf{CS}(A)$ *and* t *be a vector in* $\mathsf{CS}(A^\perp)$*. Define the coset state as*

$$|A_{s,t}\rangle = \frac{1}{\sqrt{|A|}} \sum_{a \in A} (-1)^{\langle a,t\rangle} |a + s\rangle ,$$

Consider the following coset monogamy game for adversary $\mathcal{A} = (\mathcal{A}_M, \mathcal{A}_L, \mathcal{A}_R)$*.*

1. *The challenger picks a random change of basis matrix* U *that describes a uniformly random subspace* $A \subseteq \mathbb{F}_2^n$ *of dimension* $n/2$*, samples* $s \xleftarrow{\$} \mathsf{CS}(A)$ *and* $t \xleftarrow{\$} \mathsf{CS}(A^\perp)$*. The challenger sends* $|A_{s,t}\rangle_{\mathbf{M}}$ *to the adversary* $\mathcal{A}_M$*.*
2. $\mathcal{A}_M$ *generates a bipartite state on* $\mathbf{LR}$*, sends* $\mathbf{L}$ *to* $\mathcal{A}_L$ *and sends* $\mathbf{R}$ *to* $\mathcal{A}_R$*.*

3. *The challenger samples* $r_L, r_R \xleftarrow{\$} \mathbb{F}_2^n$. *$\mathcal{A}_L$ is given* (U, r_L) *and $\mathcal{A}_R$ is given* (U, r_R). *$\mathcal{A}_L$ returns* b_0^l *and $\mathcal{A}_R$ returns* b_1^r.

The adversary wins if and only if $b_0^l \oplus b_1^r = \langle r_L, s\rangle \oplus \langle r_R, t\rangle$. *Let* $\mathsf{CosetMonogamy}(\mathcal{A}, 1^\lambda)$ *be the random variable that takes value* $1/0/\perp$ *if the adversary $\mathcal{A}$ wins/loses/aborts in the above game, respectively. For any adversary $\mathcal{A}$,*

$$\Pr\left[\mathsf{CosetMonogamy}(\mathcal{A}, 1^\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Remark 6.3. *The canonical representation set* CS *defined in [KY25] is a bit different from ours. In [KY25] they don't have U_A, instead they define* $\mathsf{Can}_A(z)$ *as the lexicographically smallest element in $A+z$. However, it is easy to see that the proof in [KY25] holds even if we define the canonical representation our way.*

Now we present a modified version of the above game where the prover can choose to abort before the test.

Lemma 6.4 (Coset Monogamy-of-Entanglement Game With Abort)**.** *Let* $A \leq \mathbb{F}_2^n$ *be a uniformly random subspace of dimension* $n/2$ *described by the first* $n/2$ *columns of a uniformly random change of basis matrix* U. *Let* s *be a vector in* $\mathsf{CS}(A)$ *and* t *be a vector in* $\mathsf{CS}(A^\perp)$. *Define the coset state as*

$$|A_{s,t}\rangle = \frac{1}{\sqrt{|A|}} \sum_{a \in A} (-1)^{\langle a,t\rangle} |a + s\rangle,$$

Consider the following coset monogamy game with abort for adversary $\mathcal{A} = (\mathcal{A}_M, \mathcal{A}_L, \mathcal{A}_R)$. *Modifications are highlighted.*

1. *The challenger picks a random change of basis matrix U that describes a uniformly random subspace $A \subseteq \mathbb{F}_2^n$ of dimension $n/2$, samples $s \xleftarrow{\$} \mathsf{CS}(A)$ and $t \xleftarrow{\$} \mathsf{CS}(A^\perp)$. The challenger sends $|A_{s,t}\rangle_{\mathbf{M}}$ to the adversary $\mathcal{A}_M$.*
2. *$\mathcal{A}_M$ can choose to abort in this step. If it aborts, the output of the game is $\perp$. Otherwise, $\mathcal{A}_M$ generates a bipartite state on $\mathbf{LR}$, sends $\mathbf{L}$ to $\mathcal{A}_L$ and sends $\mathbf{R}$ to $\mathcal{A}_R$.*
3. *The challenger samples* $r_L, r_R \xleftarrow{\$} \mathbb{F}_2^n$. *$\mathcal{A}_L$ is given* (U, r_L) *and $\mathcal{A}_R$ is given* (U, r_R). *$\mathcal{A}_L$ returns* b_0^l *and $\mathcal{A}_R$ returns* b_1^r.

The adversary wins if and only if $b_0^l \oplus b_1^r = \langle r_L, s\rangle \oplus \langle r_R, t\rangle$. *Let* $\mathsf{CosetMonogamy}(\mathcal{A}, 1^\lambda)$ *be the random variable that takes value* $1/0/\perp$ *if the adversary $\mathcal{A}$ wins/loses/aborts in the above game, respectively. For any adversary $\mathcal{A}$ such that*

$$\Pr\left[\mathsf{CosetMonogamy}(\mathcal{A}, 1^\lambda) \neq \perp\right] \geq \frac{1}{\mathsf{poly}(\lambda)},$$

we have

$$\frac{\Pr\left[\mathsf{CosetMonogamy}(\mathcal{A}, 1^\lambda) = 1\right]}{\Pr\left[\mathsf{CosetMonogamy}(\mathcal{A}, 1^\lambda) \neq \perp\right]} \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Proof. Suppose that there exists an adversary $\mathcal{A}$ and an inverse polynomial $\delta(\lambda)$ such that

$$\Pr\left[\mathsf{CosetMonogamy}(\mathcal{A}, 1^\lambda) \neq \perp\right] \geq \delta(\lambda),$$

and

$$\frac{\Pr\left[\mathsf{CosetMonogamy}(\mathcal{A}, 1^\lambda) = 1\right]}{\Pr\left[\mathsf{CosetMonogamy}(\mathcal{A}, 1^\lambda) \neq \perp\right]} \geq \frac{1}{2} + \delta(\lambda).$$

Then we construct an adversary $\mathcal{B}$ that breaks Theorem 6.2. $\mathcal{B}$ works as follows:

1. $\mathcal{B}_M$ runs $\mathcal{A}_M$. If $\mathcal{A}_M$ aborts then $\mathcal{B}_M$ sends an abort symbol to $\mathcal{B}_L$ and $\mathcal{B}_R$. Otherwise, it forwards $\mathbf{LR}$, $\mathcal{A}_M$'s output to $\mathcal{B}_L$ and $\mathcal{B}_R$, respectively.
2. $\mathcal{B}_L$ outputs a random bit if an abort symbol is received. Otherwise, it runs $\mathcal{A}_L$ and forwards the answer.
3. $\mathcal{B}_R$ outputs a random bit if an abort symbol is received. Otherwise, it runs $\mathcal{A}_R$ and forwards the answer.

Condition on not aborting, the winning probability of $\mathcal{B}$ is at least $\frac{1}{2}+\delta(\lambda)$. On the other hand, if $\mathcal{A}_M$ aborts, the winning probability of $\mathcal{B}$ is exactly $\frac{1}{2}$ because it outputs two random bits. Thus, the overall winning probability of $\mathcal{B}$ is

$$\delta(\lambda)\cdot\left(\frac{1}{2}+\delta(\lambda)\right)+(1-\delta(\lambda))\cdot\frac{1}{2}=\frac{1}{2}+(\delta(\lambda))^2.$$

By contradiction, we proved the lemma. □

Lemma 6.5. *Consider the following asymmetric coset monogamy game for adversary $\mathcal{A}=(\mathcal{A}_M,\mathcal{A}_L,\mathcal{A}_R)$.*

1. *The challenger picks a random change of basis matrix U that describes a uniformly random subspace $A\subseteq\mathbb{F}_2^n$ of dimension $n/2$ using its first $n/2$ columns, samples $s\xleftarrow{\$}\mathsf{CS}(A)$ and $t\xleftarrow{\$}\mathsf{CS}(A^\perp)$. The challenger sends $|A_{s,t}\rangle_{\mathbf{M}}$ to the adversary $\mathcal{A}_M$.*
2. *$\mathcal{A}_M$ can choose to abort in this step. If it aborts, the output of the game is $\perp$. Otherwise, $\mathcal{A}_M$ generates a bipartite state on $\mathbf{LR}$, sends $\mathbf{L}$ to $\mathcal{A}_L$ and sends $\mathbf{R}$ to $\mathcal{A}_R$.*
3. *The challenger samples $r_L,r_R\xleftarrow{\$}\mathbb{F}_2^n$. $\mathcal{A}_L$ is given (U,r_L) and $\mathcal{A}_R$ is given $(\langle r_L,s\rangle,U,r_R)$. $\mathcal{A}_L$ returns b_0^l and $\mathcal{A}_R$ returns b_1^r.*

The adversary wins if and only if $b_0^l=\langle r_L,s\rangle$ and $b_1^r=\langle r_R,t\rangle$. Let $\mathsf{AsymmetricCosetMonogamy}(\mathcal{A},1^\lambda)$ be the random variable that takes value $1/0/\perp$ if the adversary $\mathcal{A}$ wins/loses/aborts in the above game, respectively. For any efficient adversary $\mathcal{A}$ such that

$$\Pr\left[\mathsf{AsymmetricCosetMonogamy}(\mathcal{A},1^\lambda)\neq\perp\right]\geq\frac{1}{\mathsf{poly}(\lambda)},$$

we have

$$\frac{\Pr\left[\mathsf{AsymmetricCosetMonogamy}(\mathcal{A},1^\lambda)=1\right]}{\Pr\left[\mathsf{AsymmetricCosetMonogamy}(\mathcal{A},1^\lambda)\neq\perp\right]}\leq\frac{1}{2}+\mathsf{negl}(\lambda).$$

Proof. The difference between this lemma and Lemma 6.4 is that $\mathcal{A}_R$ is given the correct answer of $\mathcal{A}_L$. For any adversary $\mathcal{A}$ that wins with probability at least $\frac{1}{2}+\delta(\lambda)$ in this game condition on not aborting and its non-aborting probability is at least $\delta(\lambda)$ for some inverse polynomial $\delta(\lambda)$, we construct adversary $\mathcal{B}$ that violates Lemma 6.4. $\mathcal{B}$ is constructed as follows:

1. $\mathcal{B}_M$ runs $\mathcal{A}_M$ on $\mathbf{M}$ to obtain $\mathbf{LR}$. It aborts if $\mathcal{A}_M$ aborts.
2. $\mathcal{B}_L$ receives (U,r_L) and runs $\mathcal{A}_L$ on $\mathbf{L}$. It outputs b_0^l, the output of $\mathcal{A}_L$.
3. $\mathcal{B}_R$ receives (U,r_R) and samples $b_0\xleftarrow{\$}\mathbb{F}_2$ as a guess to the answer. It runs $\mathcal{A}_R$ with input (b_0,U,r_R) on $\mathbf{R}$. It outputs b_1^r, the output of $\mathcal{A}_R$.

The aborting probability of $\mathcal{B}$ is exactly the same as the aborting probability of $\mathcal{A}$, now let us investigate the winning probability of $\mathcal{B}$ condition on not aborting. Define

$$p_{l,r}^b=\Pr_{U,s,t,r_L,r_R,b_0}\left[\begin{matrix}\mathbb{1}[b_0^l=\langle r_L,s\rangle]=l\\ \mathbb{1}[b_1^r=\langle r_R,t\rangle]=r\end{matrix}\middle|\begin{matrix}\mathbb{1}[b_0=\langle r_L,s\rangle]=b\\ \mathcal{A}\text{ does not abort}\end{matrix}\right].$$

where $\mathbb{1}[a=b]$ is 1 when $a=b$ and 0 otherwise. From our assumption on $\mathcal{A}$ we have

$$p^1_{1,1} = \Pr_{U,s,t,r_L,r_R,b_0}\left[\begin{matrix}b_0^l=\langle r_L,s\rangle\\ b_1^r=\langle r_R,t\rangle\end{matrix}\middle|\begin{matrix}b_0=\langle r_L,s\rangle\\ \mathcal{A}\text{ does not abort}\end{matrix}\right] \geq \frac{1}{2}+\delta(\lambda).$$

Notice that whether the event $b_0=\langle r_L,s\rangle$ happens or not cannot be detected by both parties with noticeable probability. For $\mathcal{B}_L$, it does not know whether b_0 is sampled correctly, so the correctness of b_0 will not affect its winning probability on outputting the correct $b_0^l=\langle r_L,s\rangle$. For $\mathcal{B}_R$, as long as $s\neq 0$ which happens with $1-\mathsf{negl}$ probability, it has no information about whether r_L satisfies $\langle r_L,s\rangle=b_0$. So whether $\langle r_L,s\rangle=b_0$ or not will not affect the winning probability of $\mathcal{B}_R$. Thus we have for $l=0,1$,

$$p^1_{l,0}+p^1_{l,1}=p^0_{l,0}+p^0_{l,1}.$$

And we have for $r=0,1$,

$$\left|(p^1_{0,r}+p^1_{1,r})-(p^0_{0,r}+p^0_{1,r})\right|=\mathsf{negl}(\lambda)$$

Thus there exists a constant c such that

$$\left\|\begin{pmatrix}p^0_{0,0}&p^0_{0,1}\\ p^0_{1,0}&p^0_{1,1}\end{pmatrix}-\begin{pmatrix}p^1_{0,0}&p^1_{0,1}\\ p^1_{1,0}&p^1_{1,1}\end{pmatrix}+c\begin{pmatrix}1&-1\\ -1&1\end{pmatrix}\right\|_\infty=\mathsf{negl}(\lambda).$$

From this we know that the total winning probability of $\mathcal{B}$ is

$$\begin{aligned}&\frac{1}{2}(p^0_{0,0}+p^0_{1,1}+p^1_{0,0}+p^1_{1,1})-\mathsf{negl}(\lambda)\\ \geq&\frac{1}{2}(-p^0_{0,0}+p^0_{1,1}+p^1_{0,0}+p^1_{1,1})-\mathsf{negl}(\lambda)\\ \geq&\frac{1}{2}(-p^1_{0,0}+p^1_{1,1}+p^1_{0,0}+p^1_{1,1})-\mathsf{negl}(\lambda)\\ \geq&p^1_{1,1}-\mathsf{negl}(\lambda)\\ \geq&\frac{1}{2}+\delta(\lambda)-\mathsf{negl}(\lambda).\end{aligned}$$

This yields a contradiction, proving the lemma. □

Intuitively, $|A_{s,t}\rangle$ and $\left|\Psi^{\mathsf{sk}}_{0,n/2,n/2}\right\rangle$ have the same distribution. The coset state is exactly the anchor state with $n_e=0, n_h=n_s=n/2$. We can use this theorem to prove a similar statement in our case.

Lemma 6.6. *Let $n(\lambda)\geq\lambda, n_e(\lambda)$ be polynomials. Define the following asymmetric anchor monogamy game* AsymmetricAnchorMonogamy *for adversary $\mathcal{A}=(\mathcal{A}_M,\mathcal{A}_L,\mathcal{A}_R)$.*

1. *The challenger samples $\left(\mathsf{sk},\left|\Psi^{\mathsf{sk}}_{n_e,n/2,n/2}\right\rangle_{\text{⚓🏴‍☠️}}\right)\leftarrow\mathsf{GenAnchorState}_{n_e,n/2,n/2}(1^\lambda)$ and sends 🏴‍☠️ to the adversary $\mathcal{A}_M$.*
2. *$\mathcal{A}_M$ can choose to abort in this step. If it aborts, the output of the game is $\perp$. Otherwise, $\mathcal{A}_M$ generates a bipartite state on $\mathbf{LR}$, sends $\mathbf{L}$ to $\mathcal{A}_L$ and sends $\mathbf{R}$ to $\mathcal{A}_R$.*
3. *The challenger samples $r_L,r_R\xleftarrow{\$}\mathbb{F}_2^{n_e+n}$. $\mathcal{A}_L$ is given (U_{shift},r_L) and $\mathcal{A}_R$ is given $(\langle r_L,v\rangle,U_{\mathsf{shift}},r_R)$. $\mathcal{A}_L$ returns b_0^l and $\mathcal{A}_R$ returns b_1^r.*

The adversary wins if and only if $b_0^l = \langle r_L, v\rangle$ *and* $b_1^r = \langle r_R, u\rangle$. *Let* $\mathsf{AsymmetricAnchorMonogamy}(\mathcal{A}, 1^\lambda)$ *be the random variable that takes value* $1/0/\bot$ *if the adversary* $\mathcal{A}$ *wins/loses/aborts in the above game, respectively. For any efficient adversary* $\mathcal{A}$ *such that*

$$\Pr\left[\mathsf{AsymmetricAnchorMonogamy}(\mathcal{A}, 1^\lambda) \neq \bot\right] \geq \frac{1}{\mathsf{poly}(\lambda)},$$

we have

$$\frac{\Pr\left[\mathsf{AsymmetricAnchorMonogamy}(\mathcal{A}, 1^\lambda) = 1\right]}{\Pr\left[\mathsf{AsymmetricAnchorMonogamy}(\mathcal{A}, 1^\lambda) \neq \bot\right]} \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Proof. The proof is similar to the domain extension technique, the only difference is that we are extending the EPR part. For any adversary $\mathcal{A}$ that wins with probability at least $\frac{1}{2} + \delta(\lambda)$ in this game condition on not aborting and its non-aborting probability is at least $\delta(\lambda)$ for some inverse polynomial $\delta(\lambda)$, we construct adversary $\mathcal{B}$ that violates Lemma 6.5:

1. $\mathcal{B}_M$ receives $|A_{s,t}\rangle$ on n-qubit register $\mathbf{A}$, and tensors it with a $2n_e$-qubit EPR state $|\mathsf{EPR}\rangle^{\otimes n_e}_{\mathbf{A}'}$ which is fully entangled between n_e-qubit register ⚓ and n_e-qubit register $\mathbf{A}'$. Let 🏴‍☠️ $= \mathbf{A} \otimes \mathbf{A}'$, it samples a uniform random change of basis matrix U_{shift} in $\mathbb{F}_2^{n_e+n}$ and applies

$$\mathcal{U}_{\mathsf{shift}} |z\rangle_{🏴‍☠️} := |U_{\mathsf{shift}} z\rangle_{🏴‍☠️}$$

   on $\mathbf{A} \otimes \mathbf{A}'$. Finally, it runs $\mathcal{A}_M$ on 🏴‍☠️ to obtain $\mathbf{LR}$. It aborts if $\mathcal{A}_M$ aborts. It sends $\mathbf{L}, U_{\mathsf{shift}}$ to $\mathcal{B}_L$ and sends $\mathbf{R}, U_{\mathsf{shift}}$ to $\mathcal{B}_R$.
2. $\mathcal{B}_L$ receives $\mathbf{L}$ and (U, r_L). It samples $r'_L \xleftarrow{\$} \mathbb{F}_2^{n_e}$, computes

$$U'_{\mathsf{shift}} = U_{\mathsf{shift}} \begin{pmatrix} U & \\ & I_{n_e} \end{pmatrix} \begin{pmatrix} I_{n/2} & & \\ & & I_{n/2} \\ & I_{n_e} & \end{pmatrix}.$$

   It runs $\mathcal{A}_L$ on $\mathbf{L}$ with input

$$\left(U'_{\mathsf{shift}}, U^{-t}_{\mathsf{shift}}\left(r_L \times r'_L\right)\right).$$

   It outputs b_0^l, the output of $\mathcal{A}_L$.
3. $\mathcal{B}_R$ receives $\mathbf{R}$ and (b_0, U, r_R). It samples $r'_R \xleftarrow{\$} \mathbb{F}_2^{n_e}$, computes U'_{shift} in the same way and runs $\mathcal{A}_R$ on $\mathbf{R}$ with input

$$\left(b_0, U'_{\mathsf{shift}}, U_{\mathsf{shift}}\left(r_R \times r'_R\right)\right).$$

   It outputs b_1^r, the output of $\mathcal{A}_R$.

First note that the distribution of the state on ⚓🏴‍☠️ after step 1 together with U'_{shift} has the same distribution as $\left(U_{\mathsf{shift}}, \left|\Psi^{\mathsf{sk}}_{n_e,n/2,n/2}\right\rangle\right)$ sampled by $\mathsf{GenAnchorState}_{n_e,n/2,n/2}(1^\lambda)$. Let v' be the last $n/2$ bits of $U^{-1}v$. If $\mathcal{A}_L$ returns correctly, its output is

$$\begin{aligned}&\left\langle U^{-t}_{\mathsf{shift}}(r_L \times r'_L), U_{\mathsf{shift}}(s \times 0^{n_e})\right\rangle \\ =&\left\langle r_L \times r'_L, s \times 0^{n_e}\right\rangle \\ =&\langle r_L, s\rangle.\end{aligned}$$

Similarly, if $\mathcal{A}_R$ returns correctly, its output is

$$\left\langle U_{\mathsf{shift}}(r_R \times r'_R), U^{-t}_{\mathsf{shift}}(t \times 0^{n_e})\right\rangle$$

$$= \langle r_R \times r'_R, t \times 0^{n_e} \rangle$$
$$= \langle r_R, t \rangle .$$

□

Now we switch from the Goldreich-Levin style query to the oracle style query.

Lemma 6.7. *Let $n(\lambda) \geq \lambda$, $n_e(\lambda)$, and $m_1(\lambda)$ be polynomials. Define the following asymmetric oracle monogamy game* AsymmetricOracleMonogamy *for adversary $\mathcal{A} = (\mathcal{A}_M, \mathcal{A}_L, \mathcal{A}_R)$.*

1. *The challenger generates $\left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}_{n_e,3n/4,3n/4}\right\rangle_{\text{⚓🏴‍☠️}}\right) \leftarrow \mathsf{GenAnchorState}_{n_e,3n/4,3n/4}(1^\lambda)$ and gives the register 🏴‍☠️ to $\mathcal{A}_M$.*
2. *$\mathcal{A}_M$ can choose to abort in this step. If it aborts, the output of the game is $\perp$. Otherwise, $\mathcal{A}_M$ generates a bipartite state on $\mathbf{LR}$. It sends $\mathbf{L}$ to $\mathcal{A}_L$ and sends $\mathbf{R}$ to $\mathcal{A}_R$.*
3. *The challenger samples $b_0 \xleftarrow{\$} \mathbb{F}_2$ and $b_1 \xleftarrow{\$} \mathbb{F}_2^{m_1}$. It gives $\mathcal{O}^{b_0}_{T+v}$ to $\mathcal{A}_L$. Then it gives b_0 and $\mathcal{O}^{b_1}_{S^\perp+u}$ to $\mathcal{A}_R$.*
4. *$\mathcal{A}_L$ returns b_0^l and $\mathcal{A}_R$ returns b_1^r.*

The adversary wins if $b_0^l = b_0$ and $b_1^r = b_1$. Note that the oracle given to $\mathcal{A}_L$ has one-bit output, but the oracle given to $\mathcal{A}_R$ has m_1 bits output. Let AsymmetricOracleMonogamy$(\mathcal{A}, 1^\lambda)$ *be the random variable that takes value $1/0/\perp$ if the adversary $\mathcal{A}$ wins/loses/aborts in the above game, respectively. For any efficient adversary $\mathcal{A}$ such that*

$$\Pr\left[\mathsf{AsymmetricOracleMonogamy}(\mathcal{A}, 1^\lambda) \neq \perp\right] \geq \frac{1}{\mathsf{poly}(\lambda)},$$

we have

$$\frac{\Pr\left[\mathsf{AsymmetricOracleMonogamy}(\mathcal{A}, 1^\lambda) = 1\right]}{\Pr\left[\mathsf{AsymmetricOracleMonogamy}(\mathcal{A}, 1^\lambda) \neq \perp\right]} \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Proof. For any adversary $\mathcal{A}$ that wins with probability at least $\frac{1}{2} + \delta(\lambda)$ in this game condition on not aborting and its non-aborting probability is at least $\delta(\lambda)$ for some inverse polynomial $\delta(\lambda)$, we consider a series of hybrids.

Hybrid 0: This hybrid corresponds to the adversary $\mathcal{A}$ in the AsymmetricOracleMonogamy game. The non-aborting probability for $\mathcal{A}$ is at least $\delta(\lambda)$ and conditional on that, the winning probability is at least $\frac{1}{2} + \delta(\lambda)$.

Hybrid 1: This hybrid corresponds to the adversary $\mathcal{A}$ in the following game:

1. The challenger generates $\left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}_{n_e,n/2,n/2}\right\rangle_{\text{⚓🏴‍☠️}}\right) \leftarrow \mathsf{GenAnchorState}_{n_e,n/2,n/2}(1^\lambda)$. It then runs $\mathsf{sk}^* \leftarrow \mathsf{DomainExtension}_{n/4,n/4}(1^\lambda)$ on 🏴‍☠️ and gives the register 🏴‍☠️* to $\mathcal{A}_M$.
2. $\mathcal{A}_M$ can choose to abort in this step. If it aborts, the output of the game is $\perp$. Otherwise, $\mathcal{A}_M$ generates a bipartite state on $\mathbf{LR}$. It sends $\mathbf{L}$ to $\mathcal{A}_L$ and sends $\mathbf{R}$ to $\mathcal{A}_R$.
3. The challenger samples $b_0 \xleftarrow{\$} \mathbb{F}_2$ and $b_1 \xleftarrow{\$} \mathbb{F}_2^{m_1}$. It gives $\mathcal{O}^{b_0}_{T_{\mathsf{ext}}+v_{\mathsf{ext}}}$ to $\mathcal{A}_L$. Then it gives b_0 and $\mathcal{O}^{b_1}_{S^\perp_{\mathsf{ext}}+u_{\mathsf{ext}}}$ to $\mathcal{A}_R$.
4. $\mathcal{A}_L$ returns b_0^l and $\mathcal{A}_R$ returns b_1^r. The adversary wins if $b_0^l = b_0$ and $b_1^r = b_1$.

Claim 6.8. *The non-aborting probability for $\mathcal{A}$ in **Hybrid 1** is at least $\delta(\lambda)$ and conditional on that, the winning probability is at least $\frac{1}{2} + \delta(\lambda)$.*

Proof. By Fact 4.6 and Fact 4.7, the input distribution of $\mathcal{A}$ doesn't change. □

Hybrid 2: This hybrid corresponds to the adversary $\mathcal{A}$ in the following game:

1. The challenger generates $\left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}_{n_e,n/2,n/2}\right\rangle_{\text{🏴‍☠️}}\right) \leftarrow \mathsf{GenAnchorState}_{n_e,n/2,n/2}(1^\lambda)$. It then runs $\mathsf{sk}^* \leftarrow \mathsf{DomainExtension}_{n/4,n/4}(1^\lambda)$ on 🏴‍☠️ and gives the register 🏴‍☠️* to $\mathcal{A}_M$.
2. $\mathcal{A}_M$ can choose to abort in this step. If it aborts, the output of the game is $\perp$. Otherwise, $\mathcal{A}_M$ generates a bipartite state on $\mathbf{LR}$. It sends $\mathbf{L}$ to $\mathcal{A}_L$ and sends $\mathbf{R}$ to $\mathcal{A}_R$.
3. The challenger samples $b_0 \xleftarrow{\$} \mathbb{F}_2$ and $b_1 \xleftarrow{\$} \mathbb{F}_2^{m_1}$. It gives $\mathcal{O}^{b_0}_{T^*+v^*}$ to $\mathcal{A}_L$. Then it gives b_0 and $\mathcal{O}^{b_1}_{S^{*\perp}+u^*}$ to $\mathcal{A}_R$.
4. $\mathcal{A}_L$ returns b_0^l and $\mathcal{A}_R$ returns b_1^r. The adversary wins if $b_0^l = b_0$ and $b_1^r = b_1$.

Claim 6.9. *The non-aborting probability for $\mathcal{A}$ in **Hybrid 2** is at least $\delta(\lambda)$ and conditional on that, the winning probability is at least $\frac{1}{2} + \delta(\lambda) - \mathsf{negl}(\lambda)$.*

Proof. T^* is a uniformly random subspace of dimension $n_e + 5n/4$ satisfying $T_{\mathsf{ext}} \leq T^* \leq \mathbb{F}_2^{n_e+3n/2}$ and it is also independent of v_{ext}. S^* is a uniformly random subspace of S_{ext} of dimension $n/4$ and it is also independent of u_{ext}. The claim follows from Lemma 5.1. □

Hybrid 3: This hybrid corresponds to the adversary $\mathcal{A}$ in the following game:

1. The challenger generates $\left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}_{n_e,n/2,n/2}\right\rangle_{\text{🏴‍☠️}}\right) \leftarrow \mathsf{GenAnchorState}_{n_e,n/2,n/2}(1^\lambda)$. It then runs $\mathsf{sk}^* \leftarrow \mathsf{DomainExtension}_{n/4,n/4}(1^\lambda)$ on 🏴‍☠️ and gives the register 🏴‍☠️* to $\mathcal{A}_M$.
2. $\mathcal{A}_M$ can choose to abort in this step. If it aborts, the output of the game is $\perp$. Otherwise, $\mathcal{A}_M$ generates a bipartite state on $\mathbf{LR}$. It sends $\mathbf{L}$ to $\mathcal{A}_L$ and sends $\mathbf{R}$ to $\mathcal{A}_R$.
3. The challenger samples $r_L \xleftarrow{\$} \mathbb{F}_2^{n_e+3n/2}$ and $b_1 \xleftarrow{\$} \mathbb{F}_2^{m_1}$. It also samples $w_v \xleftarrow{\$} T_{\mathsf{ext}} + v_{\mathsf{ext}}$ and gives the following oracle to $\mathcal{A}_L$:
   - $\mathcal{O}^{\mathsf{ncol},r_L}_{T^*+w_v} : \mathbb{F}_2^{n_e+3n/2} \to \mathbb{F}_2 \cup \{\perp\}$ on input z checks if it is in $T^* + w_v$. It returns $\perp$ if it is not. Otherwise, it returns $\langle r_L, \mathsf{Can}_{T_{\mathsf{ext}}}(w_v)\rangle$.

   Then it gives $\langle r_L, \mathsf{Can}_{T_{\mathsf{ext}}}(w_v)\rangle$ and $\mathcal{O}^{b_1}_{S^{*\perp}+u^*}$ to $\mathcal{A}_R$.
4. $\mathcal{A}_L$ returns b_0^l and $\mathcal{A}_R$ returns b_1^r. The adversary wins if $b_0^l = \langle r_L, \mathsf{Can}_{T_{\mathsf{ext}}}(w_v)\rangle$ and $b_1^r = b_1$.

Claim 6.10. *The non-aborting probability for $\mathcal{A}$ in **Hybrid 3** is at least $\delta(\lambda)$ and conditional on that, the winning probability is at least $\frac{1}{2} + \delta(\lambda) - \mathsf{negl}(\lambda)$.*

Proof. In the adversary's view, $\langle r_L, v_{\mathsf{ext}}\rangle$ is a uniform random bit and is equivalent to b_0. Also $T_{\mathsf{ext}} + v_{\mathsf{ext}}$ and $T_{\mathsf{ext}} + w_v$ are the same coset. □

Hybrid 4: This hybrid corresponds to the adversary $\mathcal{A}$ in the following game:

1. The challenger generates $\left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}_{n_e,n/2,n/2}\right\rangle_{\text{🏴‍☠️}}\right) \leftarrow \mathsf{GenAnchorState}_{n_e,n/2,n/2}(1^\lambda)$. It then runs $\mathsf{sk}^* \leftarrow \mathsf{DomainExtension}_{n/4,n/4}(1^\lambda)$ on 🏴‍☠️ and gives the register 🏴‍☠️* to $\mathcal{A}_M$.
2. $\mathcal{A}_M$ can choose to abort in this step. If it aborts, the output of the game is $\perp$. Otherwise, $\mathcal{A}_M$ generates a bipartite state on $\mathbf{LR}$. It sends $\mathbf{L}$ to $\mathcal{A}_L$ and sends $\mathbf{R}$ to $\mathcal{A}_R$.
3. The challenger samples $r_L \xleftarrow{\$} \mathbb{F}_2^{n_e+3n/2}$ and $b_1 \xleftarrow{\$} \mathbb{F}_2^{m_1}$. It also samples $w_v \xleftarrow{\$} T_{\mathsf{ext}} + v_{\mathsf{ext}}$ and gives the following oracle to $\mathcal{A}_L$:

- $\mathcal{O}^{\mathsf{col},r_L}_{T^*+w_v}: \mathbb{F}_2^{n_e+3n/2} \to \mathbb{F}_2 \cup \{\perp\}$ on input z checks if it is in $T^* + w_v$. It returns $\perp$ if it is not. Otherwise, it returns $\langle r_L, \mathsf{Can}_{T_{\mathsf{ext}}}(z)\rangle$.

Then it gives $\langle r_L, v_{\mathsf{ext}}\rangle$ and $\mathcal{O}^{b_1}_{S^{*\perp}+u^*}$ to $\mathcal{A}_R$.

4. $\mathcal{A}_L$ returns b_0^l and $\mathcal{A}_R$ returns b_1^r. The adversary wins if $b_0^l = \langle r_L, v_{\mathsf{ext}}\rangle$ and $b_1^r = b_1$.

Claim 6.11. *The non-aborting probability for $\mathcal{A}$ in **Hybrid 4** is at least $\delta(\lambda)$ and conditional on that, the winning probability is at least $\frac{1}{2} + \delta(\lambda) - \mathsf{negl}(\lambda)$.*

Proof. Note that w_v is a uniform random vector independent of T_{ext} and T^*, with $\mathsf{Can}_{T_{\mathsf{ext}}}(w_v) = v_{\mathsf{ext}}$. For any coset of T_{ext} contained in $T^* + v^*$, $\langle r_L, \mathsf{Can}_{T_{\mathsf{ext}}}(z)\rangle$ alone is a bit independent of T^* given T_{ext}. To use Lemma 5.3 we only need to show that we can generate the anchor state and simulate the whole game using T_{ext} and w_v. To be more specific, we will rewrite **Hybrid 3** and **Hybrid 4** as two games with input $(T_{\mathsf{ext}}, T^*, w_v)$ where one of them is the collapsing version and the other one is the non-collapsing version in order to use Lemma 5.3. Now we rewrite **Hybrid 3** and **Hybrid 4** for fixed T_{ext}, T^* and w_v.

1. Sample random subspaces $S^* \le S_{\mathsf{ext}} \le T_{\mathsf{ext}}$ where S^* is a $n/4$ dimensional subspace and S_{ext} is a $3n/4$ dimensional subspace. Sample a $(n_e+3n/2)\times(n_e+3n/2)$ uniform random change of basis matrix U_{shift} condition on the span of its first $n/4, 3n/4, n_e+3n/4, n_e+5n/4$ columns are $S^*, S_{\mathsf{ext}}, T_{\mathsf{ext}}, T^*$, respectively. Sample $u_{\mathsf{ext}} \xleftarrow{\$} \mathsf{CS}(S_{\mathsf{ext}}^{\perp})$ and compute u^* accordingly.
2. Decompose $w_v = w_s + w_x + v_{\mathsf{ext}}$ where $w_s \in S_{\mathsf{ext}}$ and $w_x = U_{\mathsf{shift}}\left(0^{3n/4} \times x' \times 0^{3n/4}\right)$ for a unique x'. Generate the following state:

$$\begin{aligned}
&\propto \sum_x |x\rangle_{\text{⚓}} \sum_{s\in S_{\mathsf{ext}}} (-1)^{\langle u_{\mathsf{ext}}, s\rangle} \left| s + U_{\mathsf{shift}}\left(0^{3n/4}\times x\times 0^{3n/4}\right) + w_v\right\rangle_{\text{🏴‍☠️}^*}\\
&\propto \sum_x |x\rangle_{\text{⚓}} \sum_{s\in S_{\mathsf{ext}}} (-1)^{\langle u_{\mathsf{ext}}, s\rangle} \left| (s+w_s) + U_{\mathsf{shift}}\left(0^{3n/4}\times (x+x')\times 0^{3n/4}\right) + v_{\mathsf{ext}}\right\rangle_{\text{🏴‍☠️}^*}\\
&\propto \sum_x (-1)^{\langle u_{\mathsf{ext}}, w_s\rangle} |x-x'\rangle_{\text{⚓}} \sum_{s\in S_{\mathsf{ext}}} (-1)^{\langle u_{\mathsf{ext}}, s\rangle} \left| s + U_{\mathsf{shift}}\left(0^{3n/4}\times x\times 0^{3n/4}\right) + v_{\mathsf{ext}}\right\rangle_{\text{🏴‍☠️}^*}.
\end{aligned}$$

Note that the adversary $\mathcal{A}$ can only access 🏴‍☠️, thus it cannot detect the phase shift $(-1)^{\langle u_{\mathsf{ext}}, w_s\rangle}$ and the control shift $|x-x'\rangle_{\text{⚓}}$. The result of the game is the same if we give the following state that is exactly the anchor state, instead of the above state.

$$\propto \sum_x |x\rangle_{\text{⚓}} \sum_{s\in S_{\mathsf{ext}}} (-1)^{\langle u_{\mathsf{ext}}, s\rangle} \left| s + U_{\mathsf{shift}}\left(0^{3n/4}\times x\times 0^{3n/4}\right) + v_{\mathsf{ext}}\right\rangle_{\text{🏴‍☠️}^*}.$$

3. Run $\mathcal{A}_M, \mathcal{A}_L, \mathcal{A}_R$, provide the corresponding oracles and check the outcome.

From the above description, we can use Lemma 5.3 to show the indistinguishability. □

Hybrid 5: This hybrid corresponds to the adversary $\mathcal{A}$ in the following game:

1. The challenger generates $\left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}_{n_e,n/2,n/2}\right\rangle_{\text{⚓🏴‍☠️}}\right) \leftarrow \mathsf{GenAnchorState}_{n_e,n/2,n/2}(1^\lambda)$. It then runs $\mathsf{sk}^* \leftarrow \mathsf{DomainExtension}_{n/4,n/4}(1^\lambda)$ on 🏴‍☠️ and gives the register 🏴‍☠️* to $\mathcal{A}_M$.
2. $\mathcal{A}_M$ can choose to abort in this step. If it aborts, the output of the game is $\perp$. Otherwise, $\mathcal{A}_M$ generates a bipartite state on $\mathbf{LR}$. It sends $\mathbf{L}$ to $\mathcal{A}_L$ and sends $\mathbf{R}$ to $\mathcal{A}_R$.

3. The challenger samples $r_L \xleftarrow{\$} \mathbb{F}_2^{n_e+3n/2}$, $r_R \xleftarrow{\$} \mathbb{F}_2^{n_e+3n/2}$ and $b'_1 \xleftarrow{\$} \mathbb{F}_2^{m_1-1}$. It gives $\mathcal{O}^{\mathsf{col},r_L}_{T^*+v^*}$ to $\mathcal{A}_L$. Then it samples $w_u \xleftarrow{\$} S^{\perp}_{\mathsf{ext}} + u_{\mathsf{ext}}$, gives $\langle r_L, v_{\mathsf{ext}}\rangle$ and this oracle to $\mathcal{A}_R$:
   - $\mathcal{O}^{\mathsf{ncol},r_R,b'_1}_{S^{*\perp}+w_u} : \mathbb{F}_2^{n_e+3n/2} \to \mathbb{F}_2^{m_1} \cup \{\perp\}$ on input z checks if it is in $S^{*\perp} + w_u$. It returns $\perp$ if it is not. Otherwise, it returns $\left\langle r_R, \mathsf{Can}_{S^{\perp}_{\mathsf{ext}}}(w_u)\right\rangle \times b'_1$.
4. $\mathcal{A}_L$ returns b^l_0 and $\mathcal{A}_R$ returns b^r_1. The adversary wins if $b^l_0 = \langle r_L, v_{\mathsf{ext}}\rangle$ and $b^r_1 = \left\langle r_R, \mathsf{Can}_{S^{\perp}_{\mathsf{ext}}}(w_u)\right\rangle \times b'_1$.

Claim 6.12. *The non-aborting probability for $\mathcal{A}$ in **Hybrid 5** is at least $\delta(\lambda)$ and conditional on that, the winning probability is at least $\frac{1}{2} + \delta(\lambda) - \mathsf{negl}(\lambda)$.*

Proof. In the adversary's view, $\langle r_R, u_{\mathsf{ext}}\rangle \times b'_1$ is a uniform random string and is equivalent to b_1. Also $T^* + v^*$ and $T^* + w_v$ are the same coset, and similarly $S^{*\perp} + u^*$ and $S^{*\perp} + w_u$ are the same coset. □

Hybrid 6: This hybrid corresponds to the adversary $\mathcal{A}$ in the following game:

1. The challenger generates $\left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}_{n_e,n/2,n/2}\right\rangle_{\text{⚓🏴‍☠️}}\right) \leftarrow \mathsf{GenAnchorState}_{n_e,n/2,n/2}(1^\lambda)$. It then runs $\mathsf{sk}^* \leftarrow \mathsf{DomainExtension}_{n/4,n/4}(1^\lambda)$ on 🏴‍☠️ and gives the register 🏴‍☠️* to $\mathcal{A}_M$.
2. $\mathcal{A}_M$ can choose to abort in this step. If it aborts, the output of the game is $\perp$. Otherwise, $\mathcal{A}_M$ generates a bipartite state on $\mathbf{LR}$. It sends $\mathbf{L}$ to $\mathcal{A}_L$ and sends $\mathbf{R}$ to $\mathcal{A}_R$.
3. The challenger samples $r_L \xleftarrow{\$} \mathbb{F}_2^{n_e+3n/2}$, $r_R \xleftarrow{\$} \mathbb{F}_2^{n_e+3n/2}$ and $b'_1 \xleftarrow{\$} \mathbb{F}_2^{m_1-1}$. It gives $\mathcal{O}^{\mathsf{col},r_L}_{T^*+v^*}$ to $\mathcal{A}_L$. Then it samples $w_u \xleftarrow{\$} S^{\perp}_{\mathsf{ext}} + u_{\mathsf{ext}}$, gives $\langle r_L, v_{\mathsf{ext}}\rangle$ and this oracle to $\mathcal{A}_R$:
   - $\mathcal{O}^{\mathsf{col},r_R,b'_1}_{S^{*\perp}+w_u} : \mathbb{F}_2^{n_e+3n/2} \to \mathbb{F}_2^{m_1} \cup \{\perp\}$ on input z checks if it is in $S^{*\perp} + w_u$. It returns $\perp$ if it is not. Otherwise, it returns $\left\langle r_R, \mathsf{Can}_{S^{\perp}_{\mathsf{ext}}}(z)\right\rangle \times b'_1$.
4. $\mathcal{A}_L$ returns b^l_0 and $\mathcal{A}_R$ returns b^r_1. The adversary wins if $b^l_0 = \langle r_L, v_{\mathsf{ext}}\rangle$ and $b^r_1 = \langle r_R, u_{\mathsf{ext}}\rangle \times b'_1$.

Claim 6.13. *The non-aborting probability for $\mathcal{A}$ in **Hybrid 6** is at least $\delta(\lambda)$ and conditional on that, the winning probability is at least $\frac{1}{2} + \delta(\lambda) - \mathsf{negl}(\lambda)$.*

Proof. Similar to the proof of Claim 6.11 except this time in the Hadamard basis. □

Hybrid 7: This hybrid corresponds to the adversary $\mathcal{A}$ in the following game:

1. The challenger generates $\left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}_{n_e,n/2,n/2}\right\rangle_{\text{⚓🏴‍☠️}}\right) \leftarrow \mathsf{GenAnchorState}_{n_e,n/2,n/2}(1^\lambda)$. It then runs $\mathsf{sk}^* \leftarrow \mathsf{DomainExtension}_{n/4,n/4}(1^\lambda)$ on 🏴‍☠️ and gives the register 🏴‍☠️* to $\mathcal{A}_M$.
2. $\mathcal{A}_M$ can choose to abort in this step. If it aborts, the output of the game is $\perp$. Otherwise, $\mathcal{A}_M$ generates a bipartite state on $\mathbf{LR}$. It sends $\mathbf{L}$ to $\mathcal{A}_L$ and sends $\mathbf{R}$ to $\mathcal{A}_R$.
3. The challenger samples $r_L \xleftarrow{\$} \mathbb{F}_2^{n_e+3n/2}$, $r_R \xleftarrow{\$} \mathbb{F}_2^{n_e+3n/2}$ and $b'_1 \xleftarrow{\$} \mathbb{F}_2^{m_1-1}$. It gives $\mathcal{O}^{\mathsf{col},r_L}_{T^*+v^*}$ to $\mathcal{A}_L$. Then it gives $\langle r_L, v_{\mathsf{ext}}\rangle$ and $\mathcal{O}^{\mathsf{col},r_R,b'_1}_{S^{*\perp}+u^*}$ to $\mathcal{A}_R$.
4. $\mathcal{A}_L$ returns b^l_0 and $\mathcal{A}_R$ returns b^r_1. The adversary wins if $b^l_0 = \langle r_L, v_{\mathsf{ext}}\rangle$ and $b^r_1 = \langle r_R, u_{\mathsf{ext}}\rangle \times b'_1$.

Claim 6.14. *The non-aborting probability for $\mathcal{A}$ in **Hybrid 7** is at least $\delta(\lambda)$ and conditional on that, the winning probability is at least $\frac{1}{2} + \delta(\lambda) - \mathsf{negl}(\lambda)$.*

Proof. $S^{*\perp} + u^*$ and $S^{*\perp} + w_u$ are the same coset. □

Using **Hybrid 7** as a tool, we construct an adversary $\mathcal{B}$ that violates Lemma 6.6. $\mathcal{B}$ simulates part of the challenger in **Hybrid 7**.

1. $\mathcal{B}_M$ receives the 🥷 register. It then runs $\mathsf{sk}^* \leftarrow \mathsf{DomainExtension}_{n/4,n/4}(1^\lambda)$ on 🥷 and gives the register 🥷* to $\mathcal{A}_M$ to obtain $\mathbf{LR}$. $\mathcal{B}_M$ aborts if $\mathcal{A}_M$ aborts. It samples $r_L^s \xleftarrow{\$} \mathbb{F}_2^{n/4}$. It sends $\mathsf{sk}^*, \mathbf{L}, r_L^s$ to $\mathcal{B}_L$ and sends $\mathsf{sk}^*, \mathbf{R}, r_L^s$ to $\mathcal{B}_R$.
2. $\mathcal{B}_L$ receives $\mathsf{sk}^*, \mathbf{L}, r_L^s$ from $\mathcal{B}_M$ and $(U_{\mathsf{shift}}, r_L)$ from the challenger. It samples $r_L^h \xleftarrow{\$} \mathbb{F}_2^{n/4}$ and computes $r'_L = {U^*_{\mathsf{shift}}}^{-t}\left(r_L^h \times r_L \times r_L^s\right)$. It runs $\mathcal{A}_L$ on $\mathbf{L}$ with oracle access to $\mathcal{O}^{\mathsf{col},r'_L}_{T^*+v^*}$. Let the output of $\mathcal{A}_L$ be b_0^l. It outputs $b_0^l \oplus \left\langle {U^*_{\mathsf{shift}}}^{-t}\left(0^{n_e+5n/4} \times r_L^s\right), v^*\right\rangle$.
3. $\mathcal{B}_R$ receives $\mathsf{sk}^*, \mathbf{R}, r_L^s$ from $\mathcal{B}_M$ and $(U_{\mathsf{shift}}, r_R, b_0)$ from the challenger. It computes $b'_0 = b_0 \oplus \left\langle {U^*_{\mathsf{shift}}}^{-t}\left(0^{n_e+5n/4} \times r_L^s\right), v^*\right\rangle$. It samples $b'_1 \xleftarrow{\$} \mathbb{F}_2^{m_1-1}$, $r_R^h, r_R^s \xleftarrow{\$} \mathbb{F}_2^{n/4}$ and computes $r'_R = U^*_{\mathsf{shift}}\left(r_R^h \times r_R \times r_R^s\right)$. It runs $\mathcal{A}_R$ on $\mathbf{R}$ with b'_0 and oracle access to $\mathcal{O}^{\mathsf{col},r'_R,b'_1}_{S^{*\perp}+u^*}$. Let the first bit of the output of $\mathcal{A}_R$ be b_1^r. It outputs $b_1^r \oplus \left\langle U^*_{\mathsf{shift}}\left(r_R^h \times 0^{n_e+5n/4}\right), u^*\right\rangle$.

Claim 6.15. *We have*

$$\Pr\left[\mathsf{AsymmetricAnchorMonogamy}(\mathcal{B}, 1^\lambda) \neq \perp\right] \geq \delta(\lambda)$$

and

$$\frac{\Pr\left[\mathsf{AsymmetricAnchorMonogamy}(\mathcal{B}, 1^\lambda) = 1\right]}{\Pr\left[\mathsf{AsymmetricAnchorMonogamy}(\mathcal{B}, 1^\lambda) \neq \perp\right]} \geq \frac{1}{2} + \delta(\lambda) - \mathsf{negl}(\lambda).$$

Proof. Clearly the aborting probability of $\mathcal{B}$ is the same as the aborting probability of $\mathcal{A}$ in **Hybrid 7**. For the winning probability of $\mathcal{B}$, first we show that if $\mathcal{A}$ outputs correctly (as in **Hybrid 7**) then the output of $\mathcal{B}$ is correct. Suppose the output of $\mathcal{A}_L$ is correct: $b_0^l = \langle r'_L, v_{\mathsf{ext}}\rangle$, then the output of $\mathcal{B}_L$ is correct:

$$\begin{aligned}
&b_0^l \oplus \left\langle {U^*_{\mathsf{shift}}}^{-t}\left(0^{n_e+5n/4} \times r_L^s\right), v^*\right\rangle \\
&= \left\langle r'_L, v_{\mathsf{ext}}\right\rangle \oplus \left\langle {U^*_{\mathsf{shift}}}^{-t}\left(0^{n_e+5n/4} \times r_L^s\right), v^*\right\rangle \\
&= \left\langle {U^*_{\mathsf{shift}}}^{-t}\left(r_L^h \times r_L \times r_L^s\right), U^*_{\mathsf{shift}}\left(0^{n/4} \times v \times 0^{n/4}\right) + v^*\right\rangle \oplus \left\langle {U^*_{\mathsf{shift}}}^{-t}\left(0^{n_e+5n/4} \times r_L^s\right), v^*\right\rangle \\
&= \left\langle {U^*_{\mathsf{shift}}}^{-t}\left(r_L^h \times r_L \times r_L^s\right), U^*_{\mathsf{shift}}\left(0^{n/4} \times v \times 0^{n/4}\right)\right\rangle \oplus \left\langle {U^*_{\mathsf{shift}}}^{-t}\left(r_L^h \times r_L \times 0^{n/4}\right), v^*\right\rangle \\
&= \langle r_L, v\rangle \oplus \left\langle {U^*_{\mathsf{shift}}}^{-t}\left(r_L^h \times r_L \times 0^{n/4}\right), U^*_{\mathsf{shift}}\left(0^{n_e+5n/4} \times v'^*\right)\right\rangle \\
&= \langle r_L, v\rangle .
\end{aligned}$$

Suppose the output of $\mathcal{A}_R$ is correct for its first bit: $b_1^r = \langle r'_R, u_{\mathsf{ext}}\rangle$, then the output of $\mathcal{B}_R$ is correct:

$$\begin{aligned}
&b_1^r \oplus \left\langle U^*_{\mathsf{shift}}\left(r_R^h \times 0^{n_e+5n/4}\right), u^*\right\rangle \\
&= \left\langle r'_R, u_{\mathsf{ext}}\right\rangle \oplus \left\langle U^*_{\mathsf{shift}}\left(r_R^h \times 0^{n_e+5n/4}\right), u^*\right\rangle \\
&= \left\langle U^*_{\mathsf{shift}}\left(r_R^h \times r_R \times r_R^s\right), {U^*_{\mathsf{shift}}}^{-t}\left(0^{n/4} \times u \times 0^{n/4}\right) + u^*\right\rangle \oplus \left\langle U^*_{\mathsf{shift}}\left(r_R^h \times 0^{n_e+5n/4}\right), u^*\right\rangle \\
&= \left\langle U^*_{\mathsf{shift}}\left(r_R^h \times r_R \times r_R^s\right), {U^*_{\mathsf{shift}}}^{-t}\left(0^{n/4} \times u \times 0^{n/4}\right)\right\rangle \oplus \left\langle U^*_{\mathsf{shift}}\left(0^{n/4} \times r_R \times r_R^s\right), u^*\right\rangle
\end{aligned}$$

$$= \langle r_R, u \rangle \oplus \left\langle U^*_{\mathsf{shift}} \left(0^{n/4} \times r_R \times r_R^s\right), {U^*_{\mathsf{shift}}}^{-t} \left(u'^* \times 0^{n_e+5n/4}\right) \right\rangle$$
$$= \langle r_R, u \rangle .$$

Now we show that the input distribution of $\mathcal{A}$ in $\mathcal{B}$ is the same as the input distribution it receives from the challenger in **Hybrid 7**. By definition, r'_L and r'_R are two uniform random vectors in $\mathbb{F}_2^{n_e+3n/2}$ and b'_1 is a uniform random vector in $\mathbb{F}_2^{m_1-1}$. And b'_0, given to $\mathcal{A}_R$, is the correct answer for $\mathcal{A}_L$:

$$\begin{aligned} b'_0 &= b_0 \oplus \left\langle {U^*_{\mathsf{shift}}}^{-t} \left(0^{n_e+5n/4} \times r_L^s\right), v^* \right\rangle \\ &= \langle r_L, v \rangle \oplus \left\langle {U^*_{\mathsf{shift}}}^{-t} \left(0^{n_e+5n/4} \times r_L^s\right), v^* \right\rangle \\ &= \left\langle r'_L, v_{\mathsf{ext}} \right\rangle . \end{aligned}$$

The last equation comes from previous calculations on the correctness of $\mathcal{B}_L$'s output. □

□

6.2 Multi-Stage Monogamy-of-Entanglement Games

Lemma 6.16. *Take any efficient adversary $\mathcal{A}$ in the multi-stage monogamy-of-entanglement game described in Definition 6.1 such that*

$$\Pr\left[\mathsf{MultiStageDecisionMonogamy}(\mathcal{A}, 1^\lambda) \neq \perp\right] = \varepsilon'(\lambda)$$

and

$$\frac{\Pr\left[\mathsf{MultiStageDecisionMonogamy}(\mathcal{A}, 1^\lambda) = 1\right]}{\Pr\left[\mathsf{MultiStageDecisionMonogamy}(\mathcal{A}, 1^\lambda) \neq \perp\right]} = 1 - \varepsilon(\lambda)$$

*for any function $\varepsilon(\lambda)$ and any inverse polynomial $\varepsilon'(\lambda)$.. Consider the state $🔑⚓\mathbf{LMRB}$ at **BeforeSplit** conditioned on $\mathcal{A}^0_M$ not aborting, which we can write as (we also purify sk in register $🔑$ here):*

$$\sum_{\mathsf{sk}} \sum_x \alpha_{\mathsf{sk},x} |\mathsf{sk}\rangle_{🔑} |x\rangle_{⚓} \left(\frac{1}{\sqrt{2}} \left|\psi^{\mathsf{sk}}_{x,0}\right\rangle_{\mathbf{LMR}} |0\rangle_{\mathbf{B}} + \frac{1}{\sqrt{2}} \left|\psi^{\mathsf{sk}}_{x,1}\right\rangle_{\mathbf{LMR}} |1\rangle_{\mathbf{B}} \right).$$

where the state $\left|\psi^{\mathsf{sk}}_{x,0}\right\rangle$ and $\left|\psi^{\mathsf{sk}}_{x,1}\right\rangle$ are normalized states. We have,

$$\mathop{\mathbb{E}}_{\mathsf{sk},x} \left[\left| \left\langle \psi^{\mathsf{sk}}_{x,0} | \psi^{\mathsf{sk}}_{x,1} \right\rangle \right|^2 \right] \leq 4\varepsilon(\lambda) + \mathsf{negl}(\lambda)$$

*where the expectation is over x, the standard basis value on $⚓$ and over sk, generated by the process of $\left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}_{n_e,n,n}\right\rangle_{⚓🏴‍☠️}\right) \leftarrow \mathsf{GenAnchorState}_{n_e,n,n}(1^\lambda)$. The expectation is weighted where the weight on sk, x is $|\alpha_{\mathsf{sk},x}|^2$. **When we say expectation over this quantity throughout this section, we mean the above weighted definition.***

Remark 6.17. *Here we assume WLOG that $\mathcal{A}$'s state is always purified. The purification can be taken to live within registers $\mathbf{L}$ or $\mathbf{R}$, and ignored by later computations. The reason we do not allow it to go into $\mathbf{M}$ is because we later extract from the middle prover, as will be made clear in later theorems. In later of this section, expectations are weighted in the same manner where these uneven weights are due to the fact that we are conditioning on $\mathcal{A}^0_M$ not aborting that may depend on sk, x.*

Proof. Consider this modified version of the multi-stage monogamy-of-entanglement game, where the challenge oracles given to $\mathcal{A}_M^1$ and $(\mathcal{A}_L^0, \mathcal{A}_R^0)$ return independently sampled random bits (b_0' and b_0, respectively).

1. The challenger generates $\left(\mathsf{sk}, \left|\Psi_{n_e,n,n}^{\mathsf{sk}}\right\rangle_{\mathbf{M}}\right) \leftarrow \mathsf{GenAnchorState}_{n_e,n,n}(1^\lambda)$ and gives the register $\mathbf{M}$ to $\mathcal{A}_M^0$. All parties of the adversary are given oracle access to $\mathcal{O}_{T+v}$ and $\mathcal{O}_{S^\perp+u}$.
2. $\mathcal{A}_M^0$ can choose to abort in this step. If it aborts, the output of the game is $\perp$. Otherwise, it generates a tripartite state on $\mathbf{LMR}$. It sends $\mathbf{L}$ to $\mathcal{A}_L^0$, sends $\mathbf{M}$ to $\mathcal{A}_M^1$ and sends $\mathbf{R}$ to $\mathcal{A}_R^0$.
3. $\mathcal{A}_M^1$ is given the access to $\mathcal{O}_{T+v}^{b_0'}$ for random $b_0' \xleftarrow{\$} \mathbb{F}_2$. We can consider purified version where we create a register that stores b_0', initialized to $|+\rangle_{\mathbf{B}'}$, a uniform superposition over all b_0'. And $\mathcal{A}_M^1$ is able to perform the following operation to access $\mathbf{B}'$:

$$\sum_{b_0' \in \mathbb{F}_2} |b_0'\rangle\langle b_0'|_{\mathbf{B}'} \otimes \mathcal{O}_{T+v}^{b_0'}.$$

   $\mathcal{A}_M^1$ with access to this oracle and $\mathbf{M}$ generates a tripartite state on $\mathbf{M}_L\mathbf{M}'\mathbf{M}_R$. It sends $\mathbf{M}_L$ to $\mathcal{A}_L^0$, sends $\mathbf{M}'$ to $\mathcal{A}_M^2$ and sends $\mathbf{M}_R$ to $\mathcal{A}_R^0$.
4. $\mathcal{A}_L^0$ and $\mathcal{A}_R^0$ are given the access to $\mathcal{O}_{T+v}^{b_0}$ for fresh generated $b_0 \xleftarrow{\$} \mathbb{F}_2$. Similarly, we do this in the purified way and introduce a $|+_{\mathbf{B}}\rangle_{\mathbf{B}}$ as the control register. $\mathcal{A}_L^0$ on $\mathbf{LM}_L$ produces answer b_0^l and a state on register $\mathbf{L}'$ that is sent to $\mathcal{A}_L^1$. $\mathcal{A}_R^0$ on $\mathbf{RM}_R$ produces answer b_0^r and a state on register $\mathbf{R}'$ that is sent to $\mathcal{A}_R^1$.
5. $\mathcal{A}_M^2$ is given the access to $\mathcal{O}_{T+v}^{b_0}$ and $\mathcal{O}_{S^\perp+u}^{b_1}$ for random $b_1 \xleftarrow{\$} \mathbb{F}_2^{m_1}$. It generates a bipartite state on $\mathbf{M}_L'\mathbf{M}_R'$. $\mathbf{M}_L'$ is given to $\mathcal{A}_L^1$ and $\mathbf{M}_R'$ is given to $\mathcal{A}_R^1$.
6. $\mathcal{A}_L^1$ and $\mathcal{A}_R^1$ are given the access to $\mathcal{O}_{T+v}^{b_0}$ and $\mathcal{O}_{S^\perp+u}^{b_1}$. $\mathcal{A}_L^1$ on $\mathbf{L}'\mathbf{M}_L'$ generates the answer b_1^l. $\mathcal{A}_R^1$ on $\mathbf{R}'\mathbf{M}_R'$ generates the answer b_1^r. The adversary wins iff $b_0^l = b_0^r = b_0$ and $b_1^l = b_1^r = b_1$.

Let $\mathsf{ModifiedMultiStageDecisionMonogamy}(\mathcal{A}, 1^\lambda)$ be the random variable that takes value $1/0/\perp$ if the adversary $\mathcal{A}$ wins/loses/aborts in the above game, respectively. This game is displayed in Figure 7.

Claim 6.18. *For all $\lambda \in \mathbb{N}$ and any adversary $\mathcal{A}$, the aborting probabilities are the same in both games,*

$$\begin{aligned}&\Pr\left[\mathsf{ModifiedMultiStageDecisionMonogamy}(\mathcal{A}, 1^\lambda) = \perp\right]\\ =&\Pr\left[\mathsf{MultiStageDecisionMonogamy}(\mathcal{A}, 1^\lambda) = \perp\right]\end{aligned}$$

and the winning probabilities conditioned on not aborting are related

$$\begin{aligned}&\frac{\Pr\left[\mathsf{ModifiedMultiStageDecisionMonogamy}(\mathcal{A}, 1^\lambda) = 1\right]}{\Pr\left[\mathsf{ModifiedMultiStageDecisionMonogamy}(\mathcal{A}, 1^\lambda) \neq \perp\right]}\\ \geq&\frac{\Pr\left[\mathsf{MultiStageDecisionMonogamy}(\mathcal{A}, 1^\lambda) = 1\right]}{\Pr\left[\mathsf{MultiStageDecisionMonogamy}(\mathcal{A}, 1^\lambda) \neq \perp\right]} - \frac{1}{2} + \frac{1}{4}\mathop{\mathbb{E}}_{\mathsf{sk},x}\left[\left|\left\langle\psi_{x,0}^{\mathsf{sk}}|\psi_{x,1}^{\mathsf{sk}}\right\rangle\right|^2\right].\end{aligned}$$

Proof. Given an adversary $\mathcal{A}$ we are able to construct two corresponding projector $\Pi_0^{\mathcal{A}}, \Pi_1^{\mathcal{A}}$ which simulate the game from **BeforeSplit** and project on the adversary winning in the case that $b_0 = 0$, $b_0 = 1$, respectively. It is crucial to notice that this projector does not act on $\mathbf{B}'$, meaning that 🔑, ⚓, $\mathbf{B}'$ and $\mathbf{B}$ all serve as control registers when calculating the maximum distinguishing probability

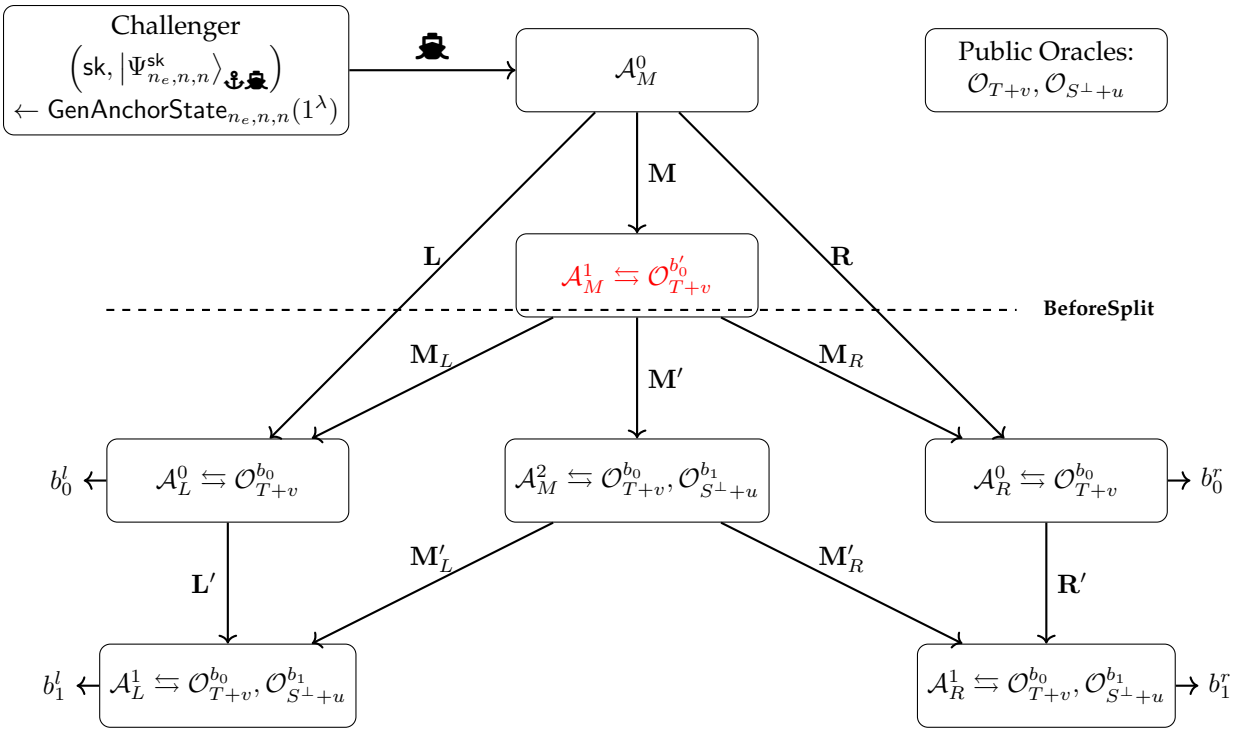

Figure 7: The modified multi-stage decision monogamy-of-entanglement game ModifiedMultiStageDecisionMonogamy.

between ModifiedMultiStageDecisionMonogamy and MultiStageDecisionMonogamy. In the game ModifiedMultiStageDecisionMonogamy, the state at **BeforeSplit** is

$$\propto \sum_{\mathsf{sk}} \sum_{x} \sum_{b'_0, b_0 \in \mathbb{F}_2} \alpha_{\mathsf{sk},x} \left|\mathsf{sk}\right\rangle_{\text{🔑}} \left|x\right\rangle_{\text{⚓}} \left|b'_0\right\rangle_{\mathbf{B}'} \left|b_0\right\rangle_{\mathbf{B}} \left|\psi^{\mathsf{sk}}_{x,b'_0}\right\rangle_{\mathbf{LMR}}.$$

In the game MultiStageDecisionMonogamy, the state at **BeforeSplit** is

$$\propto \sum_{\mathsf{sk}} \sum_{x} \sum_{b'_0, b_0 \in \mathbb{F}_2} \alpha_{\mathsf{sk},x} \left|\mathsf{sk}\right\rangle_{\text{🔑}} \left|x\right\rangle_{\text{⚓}} \left|b'_0\right\rangle_{\mathbf{B}'} \left|b_0\right\rangle_{\mathbf{B}} \left|\psi^{\mathsf{sk}}_{x,b_0}\right\rangle_{\mathbf{LMR}}.$$

if we include an untouched $\mathbf{B}'$ register. Thus the maximum distinguishing probability for any $\Pi^{\mathcal{A}}_0, \Pi^{\mathcal{A}}_1$ is

$$\begin{aligned}
&\frac{\Pr\left[\mathsf{MultiStageDecisionMonogamy}(\mathcal{A}, 1^\lambda) = 1\right]}{\Pr\left[\mathsf{MultiStageDecisionMonogamy}(\mathcal{A}, 1^\lambda) \neq \perp\right]} \\
&\quad - \frac{\Pr\left[\mathsf{ModifiedMultiStageDecisionMonogamy}(\mathcal{A}, 1^\lambda) = 1\right]}{\Pr\left[\mathsf{ModifiedMultiStageDecisionMonogamy}(\mathcal{A}, 1^\lambda) \neq \perp\right]} \\
=&\frac{1}{2} \mathop{\mathbb{E}}_{\mathsf{sk},x} \left[\mathrm{TD}\left(\left|\psi^{\mathsf{sk}}_{x,0}\right\rangle\left\langle\psi^{\mathsf{sk}}_{x,0}\right|, \left|\psi^{\mathsf{sk}}_{x,1}\right\rangle\left\langle\psi^{\mathsf{sk}}_{x,1}\right|\right)\right] \\
=&\frac{1}{2} \mathop{\mathbb{E}}_{\mathsf{sk},x} \left[\sqrt{1 - \left|\left\langle\psi^{\mathsf{sk}}_{x,0}|\psi^{\mathsf{sk}}_{x,1}\right\rangle\right|^2}\right]
\end{aligned}$$

$$\leq \frac{1}{2} \mathop{\mathbb{E}}_{\mathsf{sk},x} \left[1 - \frac{1}{2} \left| \left\langle \psi^{\mathsf{sk}}_{x,0} | \psi^{\mathsf{sk}}_{x,1} \right\rangle \right|^2 \right]$$
$$= \frac{1}{2} - \frac{1}{4} \mathop{\mathbb{E}}_{\mathsf{sk},x} \left[\left| \left\langle \psi^{\mathsf{sk}}_{x,0} | \psi^{\mathsf{sk}}_{x,1} \right\rangle \right|^2 \right].$$

□

Now we prove another claim showing that there is not noticeable advantage for any adversary to win ModifiedMultiStageDecisionMonogamy.

Claim 6.19. *The winning probability of any adversary $\mathcal{A}$ in* ModifiedMultiStageDecisionMonogamy *is at most $\frac{1}{2} + \mathsf{negl}$. For any adversary $\mathcal{A}$ such that*

$$\Pr\left[\mathsf{ModifiedMultiStageDecisionMonogamy}(\mathcal{A}, 1^\lambda) \neq \bot\right] \geq \frac{1}{\mathsf{poly}(\lambda)},$$

then we have

$$\frac{\Pr\left[\mathsf{ModifiedMultiStageDecisionMonogamy}(\mathcal{A}, 1^\lambda) = 1\right]}{\Pr\left[\mathsf{ModifiedMultiStageDecisionMonogamy}(\mathcal{A}, 1^\lambda) \neq \bot\right]} \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Proof. For any adversary $\mathcal{A}$ that does not abort with at least $\delta(\lambda)$ probability in the game ModifiedMultiStageDecisionMonogamy, and conditioned on that it wins with probability at least $\frac{1}{2} + \delta(\lambda)$ for some inverse polynomial $\delta(\lambda)$, we construct an adversary $\mathcal{B} = (\mathcal{B}_M, \mathcal{B}_L, \mathcal{B}_R)$ that wins the following game with probability at least $\frac{1}{2} + \delta(\lambda)$ conditioned on not aborting, and the non-aborting probability is at least $\delta(\lambda)$.

1. The challenger generates $\left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}_{n_e,n,n}\right\rangle_{\text{⚓🏴‍☠️}}\right) \leftarrow \mathsf{GenAnchorState}_{n_e,n,n}(1^\lambda)$ and gives the register 🏴‍☠️ to $\mathcal{B}_M$. Membership oracles $\mathcal{O}_{T+v}$ and $\mathcal{O}_{S^\perp+u}$ are made to public.
2. $\mathcal{B}_M$ can choose to abort in this step. If it aborts, the output of the game is $\bot$. Otherwise, it generates a bipartite state on $\mathbf{LR}$. It sends $\mathbf{L}$ to $\mathcal{B}_L$ and sends $\mathbf{R}$ to $\mathcal{B}_R$.
3. The challenger samples $b_0 \xleftarrow{\$} \mathbb{F}_2$ and $b_1 \xleftarrow{\$} \mathbb{F}_2^{m_1}$. It gives $\mathcal{O}^{b_0}_{T+v}$ to $\mathcal{B}_L$. Then it gives both $\mathcal{O}^{b_0}_{T+v}$ and $\mathcal{O}^{b_1}_{S^\perp+u}$ to $\mathcal{B}_R$.
4. $\mathcal{B}_L$ returns b^l_0, $\mathcal{B}_R$ returns b^r_1. The adversary wins if $b^l_0 = b_0$ and $b^r_1 = b_1$.

$\mathcal{B}$ is constructed as follows (shown in Figure 8):

1. $\mathcal{B}_M$ receives 🏴‍☠️ from the challenger, it calls $\mathcal{A}^0_M$ to obtain $\mathbf{LMR}$. It aborts if $\mathcal{A}^0_M$ aborts. It then samples b'_0 by itself, simulates $\mathcal{O}^{b'_0}_{T+v}$ on its own using $\mathcal{O}_{T+v}$, calls $\mathcal{A}^1_M$ to obtain $\mathbf{M}_L\mathbf{M}'\mathbf{M}_R$. It gives $\mathbf{LM}_L$ (as the $\mathbf{L}$ register in AsymmetricOracleMonogamy) to $\mathcal{B}_L$ and gives $\mathbf{M}'\mathbf{M}_R\mathbf{R}$ (as the $\mathbf{R}$ register in AsymmetricOracleMonogamy) to $\mathcal{B}_R$.
2. $\mathcal{B}_L$ with access to $\mathcal{O}^{b_0}_{T+v}$ calls $\mathcal{A}^0_L$. It forwards the output of $\mathcal{A}^0_L$ as b^l_0.
3. $\mathcal{B}_R$ calls $\mathcal{A}^2_M$ with access to $\mathcal{O}^{b_0}_{T+v}$ and $\mathcal{O}^{b_1}_{S^\perp+u}$ on $\mathbf{M}'$ to obtain $\mathbf{M}'_L\mathbf{M}'_R$. Then it calls $\mathcal{A}^0_R$ on $\mathbf{RM}_R$ with access to $\mathcal{O}^{b_0}_{T+v}$ to obtain $\mathbf{R}'$. Finally it calls $\mathcal{A}^1_R$ on $\mathbf{M}'_R\mathbf{R}'$ with access to $\mathcal{O}^{b_0}_{T+v}, \mathcal{O}^{b_1}_{S^\perp+u}$ and forwards the output of $\mathcal{A}^1_R$ as b^r_1.

The same adversary $\mathcal{B} = (\mathcal{B}_M, \mathcal{B}_L, \mathcal{B}_R)$ also wins the following game with probability at least $\frac{1}{2} + \delta(\lambda) - \mathsf{negl}$ conditioned on not aborting, and the non-aborting probability is at least $\delta(\lambda) - \mathsf{negl}(\lambda)$.

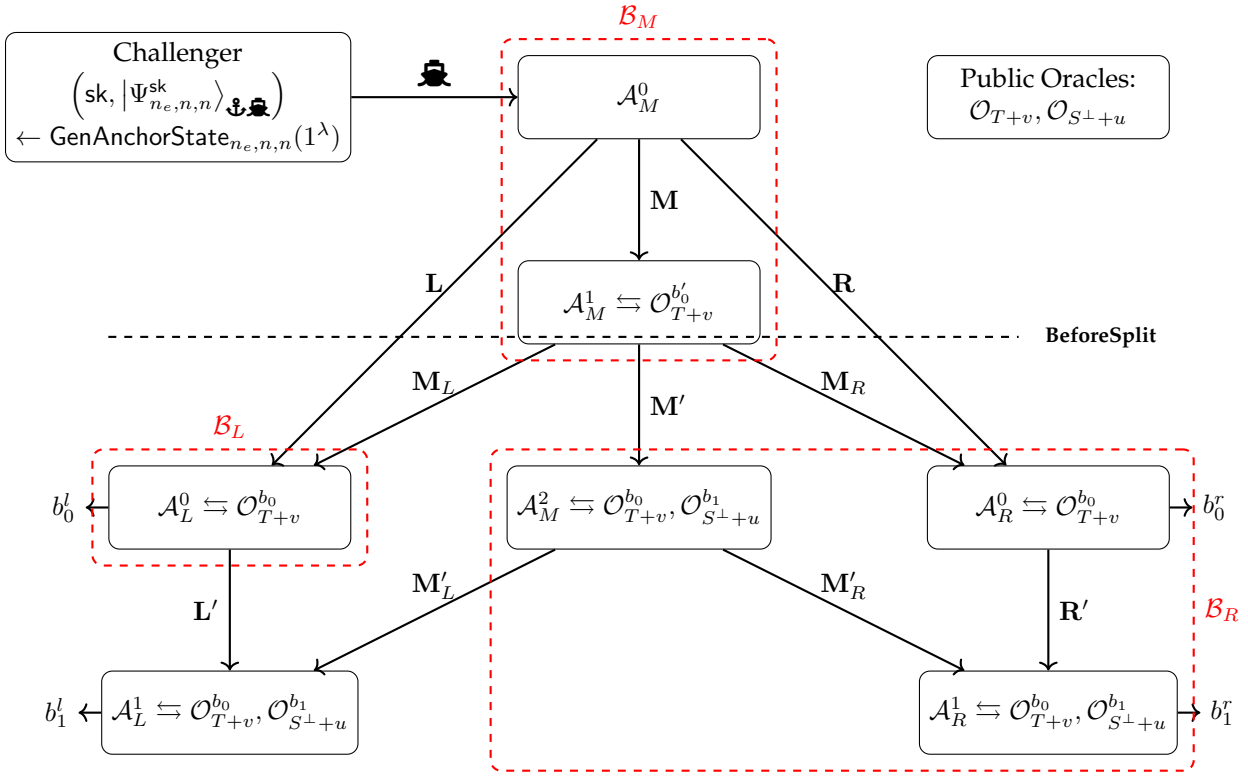

Figure 8: The adversary $\mathcal{B} = (\mathcal{B}_M, \mathcal{B}_L, \mathcal{B}_R)$.

1. The challenger generates $\left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}_{n_e,n,n}\right\rangle_{\text{⚓🏴‍☠️}}\right) \leftarrow \mathsf{GenAnchorState}_{n_e,n,n}(1^\lambda)$ and gives the register 🏴‍☠️ to $\mathcal{B}_M$. It samples uniformly random subspace T^* of dimension $n_e + 7n/4$ such that $T \le T^* \le \mathbb{F}_2^{n_e+2n}$. It also samples uniformly random subspace $S^* \le S$ of dimension $n/4$. Membership oracles $\mathcal{O}_{T^*+v}$ and $\mathcal{O}_{S^{*\perp}+u}$ are made public.
2. $\mathcal{B}_M$ can choose to abort in this step. If it aborts, the output of the game is $\perp$. Otherwise, it generates a bipartite state on $\mathbf{LR}$. It sends $\mathbf{L}$ to $\mathcal{B}_L$ and sends $\mathbf{R}$ to $\mathcal{B}_R$.
3. The challenger samples $b_0 \xleftarrow{\$} \mathbb{F}_2$ and $b_1 \xleftarrow{\$} \mathbb{F}_2^{m_1}$. It gives $\mathcal{O}^{b_0}_{T+v}$ to $\mathcal{B}_L$. Then it gives both $\mathcal{O}^{b_0}_{T^*+v}$ and $\mathcal{O}^{b_1}_{S^\perp+u}$ to $\mathcal{B}_R$.
4. $\mathcal{B}_L$ returns b^l_0, $\mathcal{B}_R$ returns b^r_1. The adversary wins if $b^l_0 = b_0$ and $b^r_1 = b_1$.

This is because that we can apply hybrid argument and use Lemma 5.1 multiple times on $\mathcal{O}_{T+v}$, $\mathcal{O}_{S^\perp+u}$, and $\mathcal{O}^{b_0}_{T+v}$ given to $\mathcal{B}_R$. Now using $\mathcal{B}$ we construct an adversary $\mathcal{C} = (\mathcal{C}_M, \mathcal{C}_L, \mathcal{C}_R)$ that wins $\mathsf{AsymmetricOracleMonogamy}$ (defined in Lemma 6.7) with probability $\frac{1}{2} + \delta(\lambda) - \mathsf{negl}$ conditioned on not aborting, and the non-aborting probability is at least $\delta(\lambda) - \mathsf{negl}(\lambda)$.

1. $\mathcal{C}_M$ receives 🏴‍☠️ and runs $\mathsf{sk}^* \leftarrow \mathsf{DomainExtension}_{n/4,n/4}(1^\lambda)$ on 🏴‍☠️. It runs $\mathcal{B}_M$ on 🏴‍☠️* to obtain $\mathbf{LR}$. It aborts if $\mathcal{B}_M$ aborts. It gives $\mathsf{sk}^*, \mathbf{L}$ to $\mathcal{B}_L$ and $\mathsf{sk}^*, \mathbf{R}$ to $\mathcal{B}_R$. All queries to $\mathcal{O}_{T^*+v}$ is equivalent to $\mathcal{O}_{T^*+v^*}$ and can be simulated by sk^*. So do queries to $\mathcal{O}_{S^{*\perp}+u}$ and other oracles below.
2. $\mathcal{C}_L$ runs $\mathcal{B}_L$ on $\mathbf{L}$ with oracle access to $\mathcal{O}^{b_0}_{T_{\mathsf{ext}}+v_{\mathsf{ext}}}$. It outputs b^l_0, the output of $\mathcal{B}_L$.
3. $\mathcal{C}_R$ runs $\mathcal{B}_R$ on $\mathbf{R}$ with oracle access to $\mathcal{O}^{b_0}_{T^*+v}$ (simulated by sk^* and b_0 given by the challenger) and $\mathcal{O}^{b_1}_{S^\perp_{\mathsf{ext}}+u_{\mathsf{ext}}}$. It outputs b^r_1, the output of $\mathcal{B}_R$.

By contradiction, such adversary $\mathcal{B}$ cannot exist and hence $\mathcal{A}$ also doesn't exist. □

Finally, combining Claim 6.18 and Claim 6.19 we get

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathsf{sk},x}\left[\left|\left\langle\psi^{\mathsf{sk}}_{x,0}|\psi^{\mathsf{sk}}_{x,1}\right\rangle\right|^2\right]\\
\leq&4\left(\frac{\Pr\left[\mathsf{ModifiedMultiStageDecisionMonogamy}(\mathcal{A},1^\lambda)=1\right]}{\Pr\left[\mathsf{ModifiedMultiStageDecisionMonogamy}(\mathcal{A},1^\lambda)\neq\bot\right]}+\frac{1}{2}-\frac{\Pr\left[\mathsf{MultiStageDecisionMonogamy}(\mathcal{A},1^\lambda)=1\right]}{\Pr\left[\mathsf{MultiStageDecisionMonogamy}(\mathcal{A},1^\lambda)\neq\bot\right]}\right)\\
=&4\left(\frac{\Pr\left[\mathsf{ModifiedMultiStageDecisionMonogamy}(\mathcal{A},1^\lambda)=1\right]}{\Pr\left[\mathsf{ModifiedMultiStageDecisionMonogamy}(\mathcal{A},1^\lambda)\neq\bot\right]}+\frac{1}{2}-(1-\varepsilon(\lambda))\right)\\
\leq&4\left(\frac{1}{2}+\mathsf{negl}(\lambda)+\frac{1}{2}-(1-\varepsilon(\lambda))\right)\\
\leq&4\varepsilon(\lambda)+\mathsf{negl}(\lambda).
\end{aligned}$$

□

Lemma 6.20. *Take any efficient adversary $\mathcal{A}$ in the multi-stage monogamy-of-entanglement game described in Definition 6.1 such that*

$$\Pr\left[\mathsf{MultiStageSearchMonogamy}(\mathcal{A},1^\lambda)\neq\bot\right]=\varepsilon'(\lambda)$$

and

$$\frac{\Pr\left[\mathsf{MultiStageSearchMonogamy}(\mathcal{A},1^\lambda)=1\right]}{\Pr\left[\mathsf{MultiStageSearchMonogamy}(\mathcal{A},1^\lambda)\neq\bot\right]}=1-\varepsilon(\lambda)$$

*for any function $\varepsilon(\lambda)$ and any inverse polynomial $\varepsilon'(\lambda)$.. Consider the state $🔑⚓\mathbf{LMRB}$ at **BeforeSplit** conditioned on $\mathcal{A}^0_M$ not aborting, which we can write as (we also purify sk in register $🔑$ here):*

$$|\psi\rangle_{🔑⚓\mathbf{LMRB}}=\sum_{\mathsf{sk}}\sum_{x}\sum_{b_0\in\mathbb{F}_2^{m_0}}\frac{\alpha_{\mathsf{sk},x}}{\sqrt{2^{m_0}}}|\mathsf{sk}\rangle_{🔑}|x\rangle_{⚓}\left|\psi^{\mathsf{sk}}_{x,b_0}\right\rangle_{\mathbf{LMR}}|b_0\rangle_{\mathbf{B}}.$$

where $\left|\psi^{\mathsf{sk}}_{x,b_0}\right\rangle$ for all b_0 are normalized states. We have,

$$\mathrm{Tr}\left(|+_{\mathbf{B}}\rangle\langle+_{\mathbf{B}}|_{\mathbf{B}}\cdot|\psi\rangle\langle\psi|_{🔑⚓\mathbf{LMRB}}\right)\leq\frac{1}{2^{m_0}}+2\sqrt{2}\varepsilon^{1/2}+\mathsf{negl}(\lambda)$$

where $|+_{\mathbf{B}}\rangle_{\mathbf{B}}=\frac{1}{\sqrt{2^{m_0}}}\sum_{b_0\in\mathbb{F}_2^{m_0}}|b_0\rangle_{\mathbf{B}}$ is the uniform superposition over b_0.

Proof. Define the probability that the adversary $\mathcal{A}$ wins (both parties output the correct b_0) when $b_0=b_0^*$ conditioned on not aborting as $1-\varepsilon_{b_0^*}(\lambda)$. We have,

$$\varepsilon=\mathop{\mathbb{E}}_{b_0}\left[\varepsilon_{b_0}\right].$$

For any different $b_0',b_0''\in\mathbb{F}_2^{m_0}$, we must have

$$\mathop{\mathbb{E}}_{\mathsf{sk},x}\left[\left|\left\langle\psi^{\mathsf{sk}}_{x,b_0'}|\psi^{\mathsf{sk}}_{x,b_0''}\right\rangle\right|^2\right]\leq2\left(\varepsilon_{b_0'}(\lambda)+\varepsilon_{b_0''}(\lambda)\right)+\mathsf{negl}(\lambda).\tag{1}$$

To prove this we construct an adversary $\mathcal{A}'$ in Lemma 6.16. $\mathcal{A}'$ runs $\mathcal{A}$ and reprograms all the oracle outputs: reprograms 0 to b'_0, 1 to b''_0, and $\perp$ unchanged. Finally, $\mathcal{A}'$ also reprograms the output of $\mathcal{A}$: reprograms b'_0 to 0 and b''_0 to 1. For this $\mathcal{A}'$,

$$\frac{\Pr\left[\mathsf{MultiStageDecisionMonogamy}(\mathcal{A}', 1^\lambda) = 1\right]}{\Pr\left[\mathsf{MultiStageDecisionMonogamy}(\mathcal{A}', 1^\lambda) \neq \perp\right]} = 1 - \frac{1}{2}\left(\varepsilon_{b'_0}(\lambda) + \varepsilon_{b''_0}(\lambda)\right).$$

By Lemma 6.16, the state of $\mathcal{A}'$ at **BeforeSplit** satisfies

$$\mathop{\mathbb{E}}_{\mathsf{sk},x}\left[\left|\left\langle \psi^{\mathsf{sk}}_{x,0} | \psi^{\mathsf{sk}}_{x,1} \right\rangle\right|^2\right] \leq 2\left(\varepsilon_{b'_0}(\lambda) + \varepsilon_{b''_0}(\lambda)\right) + \mathsf{negl}(\lambda).$$

Hence we proved Equation (1). Combine all above, we have

$$\begin{aligned}
&\mathrm{Tr}\left(|+_{\mathbf{B}}\rangle\langle+_{\mathbf{B}}|_{\mathbf{B}} \cdot |\psi\rangle\langle\psi|_{\mathbf{LMRB}}\right)\\
\leq&\frac{1}{2^{2m_0}}\mathop{\mathbb{E}}_{\mathsf{sk},x}\left[\sum_{b'_0,b''_0}\left|\left\langle \psi^{\mathsf{sk}}_{x,b'_0} | \psi^{\mathsf{sk}}_{x,b''_0} \right\rangle\right|\right]\\
=&\frac{1}{2^{2m_0}}\mathop{\mathbb{E}}_{\mathsf{sk},x}\left[2^{m_0} + \sum_{b'_0\neq b''_0}\left|\left\langle \psi^{\mathsf{sk}}_{x,b'_0} | \psi^{\mathsf{sk}}_{x,b''_0} \right\rangle\right|\right]\\
=&\frac{1}{2^{m_0}} + \frac{1}{2^{2m_0}}\sum_{b'_0\neq b''_0}\mathop{\mathbb{E}}_{\mathsf{sk},x}\left[\left|\left\langle \psi^{\mathsf{sk}}_{x,b'_0} | \psi^{\mathsf{sk}}_{x,b''_0} \right\rangle\right|\right]\\
\leq&\frac{1}{2^{m_0}} + \frac{1}{2^{2m_0}}\sum_{b'_0\neq b''_0}\sqrt{\mathop{\mathbb{E}}_{\mathsf{sk},x}\left[\left|\left\langle \psi^{\mathsf{sk}}_{x,b'_0} | \psi^{\mathsf{sk}}_{x,b''_0} \right\rangle\right|^2\right]}\\
\leq&\frac{1}{2^{m_0}} + \frac{1}{2^{2m_0}}\sum_{b'_0\neq b''_0}\sqrt{2\left(\varepsilon_{b'_0}(\lambda) + \varepsilon_{b''_0}(\lambda)\right) + \mathsf{negl}(\lambda)}\\
\leq&\frac{1}{2^{m_0}} + \frac{1}{2^{2m_0}}\sum_{b'_0\neq b''_0}\left(\sqrt{2\varepsilon_{b'_0}(\lambda) + \mathsf{negl}(\lambda)} + \sqrt{2\varepsilon_{b''_0}(\lambda) + \mathsf{negl}(\lambda)}\right)\\
\leq&\frac{1}{2^{m_0}} + 2\sqrt{2}\mathop{\mathbb{E}}_{b_0}\left[\sqrt{\varepsilon_{b_0}(\lambda) + \mathsf{negl}(\lambda)}\right]\\
\leq&\frac{1}{2^{m_0}} + 2\sqrt{2}\sqrt{\mathop{\mathbb{E}}_{b_0}\left[\varepsilon_{b_0}(\lambda) + \mathsf{negl}(\lambda)\right]}\\
\leq&\frac{1}{2^{m_0}} + 2\sqrt{2}\varepsilon^{1/2} + \mathsf{negl}(\lambda)
\end{aligned}$$

The inequality between line 4 and line 5 uses $\mathbf{Var}(X) = \mathbb{E}\left[X^2\right] - \mathbb{E}\left[X\right]^2 \geq 0$, as well as the inequality between line 8 and line 9. □

7 Entanglement Extraction

7.1 Extracting Incompatible Measurement Results via Collapsing Oracles

In later sections, we consider $n_e(\lambda) = n(\lambda)$ if not specified.

Lemma 7.1. *Let $n(\lambda) \geq \lambda$ and $m_0(\lambda) = m_1(\lambda) = m(\lambda) \geq \lambda$ be polynomials of λ. There exists an efficient extractor $\mathcal{E}_S$ with access to* sk *and the register* $\mathbf{M}$ *such that the following holds. Take any efficient adversary $\mathcal{A}$ in the multi-stage monogamy-of-entanglement game described in Definition 6.1 such that*

$$\Pr\left[\mathsf{MultiStageSearchMonogamy}(\mathcal{A}, 1^\lambda) \neq \bot\right] = \varepsilon'(\lambda)$$

and

$$\frac{\Pr\left[\mathsf{MultiStageSearchMonogamy}(\mathcal{A}, 1^\lambda) = 1\right]}{\Pr\left[\mathsf{MultiStageSearchMonogamy}(\mathcal{A}, 1^\lambda) \neq \bot\right]} = 1 - \varepsilon(\lambda)$$

for any function $\varepsilon(\lambda)$ and any inverse polynomial $\varepsilon'(\lambda)$.. Then running $\mathcal{E}_S^{\mathcal{A}_M^1}$ on the state $\mathbf{M}$ conditioned on $\mathcal{A}_M^0$ not aborting gives x, the standard basis measurement result on $\textcolor{black}{\text{⚓}}$, with probability at least $1 - 2\sqrt{2}\varepsilon^{1/2} - \mathsf{negl}(\lambda)$. That is, with $\mathbf{X}$ denoting the output register of $\mathcal{E}_S^{\mathcal{A}_M^1}$,

$$\mathbb{E}\left[\mathsf{Tr}\left(\Pi_{\mathsf{eq}}\mathcal{E}_S^{\mathcal{A}_M^1}\rho_{\text{⚓}\mathbf{M}}\left(\mathcal{E}_S^{\mathcal{A}_M^1}\right)^\dagger\Pi_{\mathsf{eq}}\right)\middle|\begin{array}{c}\left(\mathsf{sk},|\Psi^{\mathsf{sk}}\rangle_{\text{⚓☠}}\right)\leftarrow\mathsf{GenAnchorState}(1^\lambda)\\ \mathcal{A}_M^0 \textit{ does not abort and produces } \rho_{\text{⚓}\mathbf{M}} \textit{ on } \mathbf{XM}\end{array}\right] \geq 1 - 2\sqrt{2}\varepsilon^{1/2} - \mathsf{negl}(\lambda)$$

where $\Pi_{\mathsf{eq}} = \sum_x |x\rangle_{\text{⚓}}\langle x| \otimes |x\rangle_{\mathbf{X}}\langle x|$ is the projector that tests whether the output is correct. Furthermore, $\mathcal{E}_S^{\mathcal{A}_M^1}$ only makes black-box access to $\mathcal{A}_M^1$ and acts on $\mathbf{M}$. When we say $\mathcal{A}_M^1$, we refer to the process only up to ***BeforeSplit****, excluding the renaming and splitting procedure.*

Proof. In this proof we consider all notations aligned with Lemma 5.3 with $a = n, b = 2n$ and $c = 3n$. In the following proof, all procedures are conditioned on $\mathcal{A}_M^0$ not aborting. Here is the construction of $\mathcal{E}_S^{\mathcal{A}_M^1}$:

1. Set up a random oracle $\mathcal{O}_{\mathsf{random}}$ using a compress oracle with the database on $|D\rangle_{\mathbf{D}}$. Run $\mathcal{A}_M^1$ in $\mathsf{MultiStageSearchMonogamy}$ on $\mathbf{M}$. Simulate $\mathcal{O}_{S^\perp+u}$ using sk. Replace $\mathcal{O}_{T+v}^{b_0}$ with
   - $\mathcal{O}_{T+v}^{\mathsf{col},\mathcal{O}_{\mathsf{random}}} : \mathbb{F}_2^{3n} \to \mathbb{F}_2^{m_0} \cup \{\bot\}$ on input z checks if it is in $T + v$. It returns $\bot$ if it is not. Otherwise, it returns $\mathcal{O}_{\mathsf{random}}(\mathsf{Can}_S(z))$.

   And also simulate $\mathcal{O}_{T+v}$ using the above oracle.
2. Measure the database register, if there is no non-$\bot$ entry or more than one non-$\bot$ entry, then output $\bot$. Otherwise, let the unique entry be (w', y). Output the middle n bits of $U_{\mathsf{shift}}^{-1}w'$.

We calculate the success probability of $\mathcal{E}_S^{\mathcal{A}_M^1}$ using a hybrid argument. In the following hybrids, when we say compute something based on x, the standard basis value on $\textcolor{black}{\text{⚓}}$, we mean coherently do that controlled by x on $\textcolor{black}{\text{⚓}}$.

Hybrid 0: This hybrid corresponds to running $\mathcal{E}_S^{\mathcal{A}_M^1}$ on $\mathbf{M}$.

Hybrid 1: This hybrid corresponds to running $\mathcal{E}_{S,\mathsf{ncol}}^{\mathcal{A}_M^1}$ on $\mathbf{M}$ where $\mathcal{E}_{S,\mathsf{ncol}}^{\mathcal{A}_M^1}$ is controlled by $\textcolor{black}{\text{⚓}}$ and is constructed as follows:

1. Set up a random oracle $\mathcal{O}_{\mathsf{random}}$ using a compress oracle with the database on $|D\rangle_{\mathbf{D}}$. Run $\mathcal{A}_M^1$ in $\mathsf{MultiStageSearchMonogamy}$ on $\mathbf{M}$. Simulate $\mathcal{O}_{S^\perp+u}$ using sk. Sample $w_x \xleftarrow{\$} U_{\mathsf{shift}}(\{0,1\}^n \times x \times 0^n)+v$ where x is the standard basis value on $\textcolor{black}{\text{⚓}}$. Replace $\mathcal{O}_{T+v}^{b_0}$ with
   - $\mathcal{O}_{T+w_x}^{\mathsf{ncol},\mathcal{O}_{\mathsf{random}}} : \mathbb{F}_2^{3n} \to \mathbb{F}_2^{m_0} \cup \{\bot\}$ on input z checks if it is in $T + w_x$. It returns $\bot$ if it is not. Otherwise, it returns $\mathcal{O}_{\mathsf{random}}(\mathsf{Can}_S(w_x))$.

   And also simulate $\mathcal{O}_{T+v}$ using the above oracle.

2. Measure the database register, if there is no non-$\perp$ entry or more than one non-$\perp$ entry, then output $\perp$. Otherwise, let the unique entry be (w', y). Output the middle n bits of $U_{\mathsf{shift}}^{-1} w'$.

Claim 7.2. *We view x on ⚓ as a classical control string. Let $P_{D,x}$ be the distribution of database register $|D\rangle_{\mathbf{D}}$ in **Hybrid 0** conditioned on the value on ⚓ is x. Let $P'_{D,x}$ be the corresponding distribution for **Hybrid 1**. For a uniform random x, the expected statistical distance between $P_{D,x}$ and $P'_{D,x}$ is negligible.*

Proof. First note that w_x is an independent random vector given S and T is an independent subspace containing S given S, w_x. To use Lemma 5.3 we need to show that we can construct the correct state on 🏴‍☠️ conditioned on the standard basis value x on ⚓ given S, w_x so that we describe the whole hybrid as an isometry on $|S, w_x\rangle$ for uniform random x (if we trace out ⚓). Decompose w_x as $u_x + U_{\mathsf{shift}}(0^n \times x \times 0^n) + v$. We need to construct

$$\propto \sum_{x\in\mathbb{F}_2^n} |x\rangle_{⚓} \sum_{s\in S} (-1)^{\langle s,u\rangle} \left|s + U_{\mathsf{shift}}(0^{n_h} \times x \times 0^{n_s}) + v\right\rangle_{🏴‍☠️}$$

from

$$\propto \sum_{x\in\mathbb{F}_2^n} |x\rangle_{⚓} |S, w_x\rangle$$

by operating on 🏴‍☠️. We can generate the state by first generating the state on 🏴‍☠️ assuming $x = 0$ on ⚓ with a random u and with $v = 0$.

$$\propto \sum_{x\in\mathbb{F}_2^n} |x\rangle_{⚓} \sum_{s\in S} (-1)^{\langle s,u\rangle} |s\rangle .$$

Then we XOR w_x on the standard basis.

$$\begin{aligned}
&\propto \sum_{x\in\mathbb{F}_2^n} |x\rangle_{⚓} \sum_{s\in S} (-1)^{\langle s,u\rangle} |s + w_x\rangle \\
&\propto \sum_{x\in\mathbb{F}_2^n} |x\rangle_{⚓} \sum_{s\in S} (-1)^{\langle s,u\rangle} |s + u_x + U_{\mathsf{shift}}(0^n \times x \times 0^n) + v\rangle \\
&\propto \sum_{x\in\mathbb{F}_2^n} (-1)^{\langle u_x,u\rangle} |x\rangle_{⚓} \sum_{s'\in S} (-1)^{\langle s',u\rangle} \left|s' + U_{\mathsf{shift}}(0^n \times x \times 0^n) + v\right\rangle .
\end{aligned}$$

In this way we get the correct state except there is a relative phase between ⚓ and 🏴‍☠️ depending on the standard basis value on ⚓ if you consider the whole purified anchor state. But note that the standard basis value on ⚓ only acts as a control so this relative phase doesn't affect the output distribution of the hybrid. □

Hybrid 2: This hybrid corresponds to running $\mathcal{A}_M^1$ on $\mathbf{M}$ but we replace $\mathcal{O}_{T+v}^{b_0}$ with a purified version of it:

1. Set up a random oracle $\mathcal{O}_{\mathsf{random}}$ using a compress oracle with the database on $|D\rangle_{\mathbf{D}}$. Run $\mathcal{A}_M^1$ in MultiStageSearchMonogamy on $\mathbf{M}$. Simulate $\mathcal{O}_{S^\perp+u}$ using sk. Sample $w_x \xleftarrow{\$} U_{\mathsf{shift}}(\{0,1\}^n \times x \times 0^n) + v$ where x is the standard basis value on ⚓. But rename $\mathbf{D}_{\mathsf{Can}_S(w_x)}$ as $\mathbf{B}$ that stores the purified b_0. And also simulate $\mathcal{O}_{T+v}$ with $\mathcal{O}_{T+v}^{b_0}$.
2. Measure the database register, if there is no non-$\perp$ entry or more than one non-$\perp$ entry, then output $\perp$. Otherwise, let the unique entry be (w', y). Output the middle n bits of $U_{\mathsf{shift}}^{-1} w'$.

Claim 7.3. *Let $P''_{D,x}$ be the distribution of database register $|D\rangle_{\mathbf{D}}$ in **Hybrid 2** conditioned on the value on ⚓ is x. For all x, $P'_{D,x}$ and $P''_{D,x}$ are identical.*

Proof. $\mathcal{O}_{\mathsf{random}}(\mathsf{Can}_S(w_x))$ is just a uniform random string identically distributed as b_0. □

Note that the probability of outputting the correct x in **Hybrid 2** is at least $1-\frac{1}{2^m}-2\sqrt{2}\varepsilon^{1/2}-\mathsf{negl}(\lambda)$ because:

- There is at most one non-$\perp$ entry in the database because the only place that can be accessed is $\mathsf{Can}_S(w_x)$.
- The probability that $\mathsf{Can}_S(w_x)$ is also a $\perp$-entry is at most $\frac{1}{2^m}+2\sqrt{2}\varepsilon^{1/2}+\mathsf{negl}$ by Lemma 6.20.

Thus by hybrid argument, the probability that in **Hybrid 0** it outputs x is at least $1-\frac{1}{2^m}-2\sqrt{2}\varepsilon^{1/2}-\mathsf{negl}(\lambda)$. □

Now we look at a similar game DualMultiStageSearchMonogamy that reverses the order of cosets given in the challenge.

Definition 7.4 (Dual Multi-Stage Search Monogamy-of-Entanglement Game)**.** *Let $\lambda\in\mathbb{N}$ be the security parameter. Let $n(\lambda)\geq\lambda$ and $n_e(\lambda),m_0(\lambda),m_1(\lambda)$ be polynomials. Consider the following game between the challenger and an adversary $\mathcal{A}=(\mathcal{A}^0_M,\mathcal{A}^1_M,\mathcal{A}^2_M,\mathcal{A}^0_L,\mathcal{A}^1_L,\mathcal{A}^0_R,\mathcal{A}^1_R)$:*

1. *The challenger generates $\left(\mathsf{sk},\left|\Psi^{\mathsf{sk}}_{n_e,n,n}\right\rangle_{⚓🏴‍☠️}\right)\leftarrow\mathsf{GenAnchorState}_{n_e,n,n}(1^\lambda)$ and gives the register 🏴‍☠️ to $\mathcal{A}^0_M$. All parties of the adversary are given oracle access to $\mathcal{O}_{T+v}$ and $\mathcal{O}_{S^\perp+u}$.*
2. *$\mathcal{A}^0_M$ can choose to abort in this step. If it aborts, the output of the game is $\perp$. Otherwise, it generates a tripartite state on $\mathbf{LMR}$. It sends $\mathbf{L}$ to $\mathcal{A}^0_L$, sends $\mathbf{M}$ to $\mathcal{A}^1_M$ and sends $\mathbf{R}$ to $\mathcal{A}^0_R$.*
3. *$\mathcal{A}^1_M$ is given access to $\mathcal{O}^{b_0}_{S^\perp+u}$ for random $b_0\xleftarrow{\$}\{0,1\}^{m_0(\lambda)}$. $\mathcal{A}^1_M$ with access to this oracle and $\mathbf{M}$ generates a tripartite state on $\mathbf{M}_L\mathbf{M}'\mathbf{M}_R$. It sends $\mathbf{M}_L$ to $\mathcal{A}^0_L$, sends $\mathbf{M}'$ to $\mathcal{A}^2_M$ and sends $\mathbf{M}_R$ to $\mathcal{A}^0_R$. We will name the time just after all operations of $\mathcal{A}^1_M$ are done and before renaming the registers and splitting them as **BeforeSplit**.*
4. *$\mathcal{A}^0_L$ and $\mathcal{A}^0_R$ are given access to $\mathcal{O}^{b_0}_{S^\perp+u}$. $\mathcal{A}^0_L$ on $\mathbf{LM}_L$ produces answer b^l_0 and a state on register $\mathbf{L}'$ that is sent to $\mathcal{A}^1_L$. $\mathcal{A}^0_R$ on $\mathbf{RM}_R$ produces answer b^r_0 and a state on register $\mathbf{R}'$ that is sent to $\mathcal{A}^1_R$.*
5. *$\mathcal{A}^2_M$ is given access to $\mathcal{O}^{b_0}_{S^\perp+u}$ and $\mathcal{O}^{b_1}_{T+v}$ for random $b_1\xleftarrow{\$}\{0,1\}^{m_1(\lambda)}$. It generates a bipartite state on $\mathbf{M}'_L\mathbf{M}'_R$. $\mathbf{M}'_L$ is given to $\mathcal{A}^1_L$ and $\mathbf{M}'_R$ is given to $\mathcal{A}^1_R$.*
6. *$\mathcal{A}^1_L$ and $\mathcal{A}^1_R$ are given access to $\mathcal{O}^{b_0}_{S^\perp+u}$ and $\mathcal{O}^{b_1}_{T+v}$. $\mathcal{A}^1_L$ on $\mathbf{L}'\mathbf{M}'_L$ generates the answer b^l_1. $\mathcal{A}^1_R$ on $\mathbf{R}'\mathbf{M}'_R$ generates the answer b^r_1. The adversary wins iff $b^l_0=b^r_0=b_0$ and $b^l_1=b^r_1=b_1$.*

This game is displayed in Figure 9. Let $\mathsf{DualMultiStageSearchMonogamy}(\mathcal{A},1^\lambda)$ be the random variable that takes value $1/0/\perp$ if the adversary $\mathcal{A}$ wins/loses/aborts in the above game, respectively.

Corollary 7.5. *Let $n(\lambda)\geq\lambda$ and $m_0(\lambda)=m_1(\lambda)=m(\lambda)\geq\lambda$ be polynomials of λ. There exists an efficient extractor $\mathcal{E}_H$ with access to sk and the register $\mathbf{M}$ such that the following holds. Take any efficient adversary $\mathcal{A}$ in the multi-stage monogamy-of-entanglement game described in Definition 7.4 such that*

$$\Pr\left[\mathsf{DualMultiStageSearchMonogamy}(\mathcal{A},1^\lambda)\neq\perp\right]=\varepsilon'(\lambda)$$

and

$$\frac{\Pr\left[\mathsf{DualMultiStageSearchMonogamy}(\mathcal{A},1^\lambda)=1\right]}{\Pr\left[\mathsf{DualMultiStageSearchMonogamy}(\mathcal{A},1^\lambda)\neq\perp\right]}=1-\varepsilon(\lambda)$$

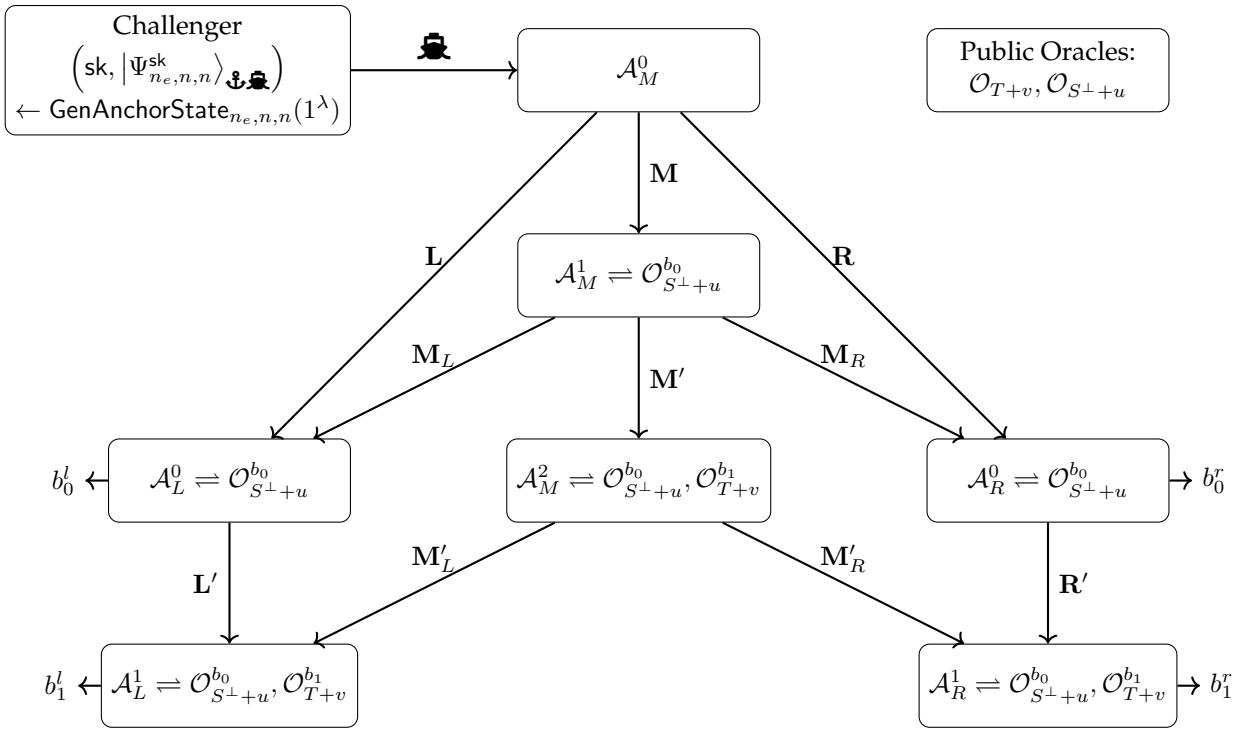

Figure 9: The dual multi-stage monogamy-of-entanglement game DualMultiStageSearchMonogamy.

for any function $\varepsilon(\lambda)$ and any inverse polynomial $\varepsilon'(\lambda)$.. Then running $\mathcal{E}_H^{\mathcal{A}^1_M}$ on the state $\mathbf{M}$ conditioned on $\mathcal{A}^0_M$ not aborting gives x, the Hadamard basis measurement result on ⚓, with probability at least $1 - 2\sqrt{2}\varepsilon^{1/2} - \mathsf{negl}(\lambda)$. That is, with $\mathbf{X}$ denoting the output register of $\mathcal{E}_H^{\mathcal{A}^1_M}$,

$$\mathbb{E}\left[\mathsf{Tr}\left(\Pi_{\mathsf{eq}}\mathcal{E}_H^{\mathcal{A}^1_M}\rho_{\text{⚓}\mathbf{M}}\left(\mathcal{E}_H^{\mathcal{A}^1_M}\right)^\dagger\Pi_{\mathsf{eq}}\right)\middle|_{\mathcal{A}^0_M \text{ does not abort and produces } \rho_{\text{⚓}\mathbf{M}} \text{ on } \mathbf{XM}}^{\left(\mathsf{sk},\left|\Psi^{\mathsf{sk}}\right\rangle_{\text{⚓🏴‍☠️}}\right)\leftarrow\mathsf{GenAnchorState}(1^\lambda)}\right] \geq 1 - 2\sqrt{2}\varepsilon^{1/2} - \mathsf{negl}(\lambda)$$

*where $\Pi_{\mathsf{eq}} = \sum_x H^{\otimes n}\left|x\right\rangle_{\text{⚓}}\left\langle x\right|H^{\otimes n} \otimes \left|x\right\rangle_{\mathbf{X}}\left\langle x\right|$ is the projector that tests whether the output is correct. Furthermore, $\mathcal{E}_H^{\mathcal{A}^1_M}$ only makes black-box access to $\mathcal{A}^1_M$ and acts on $\mathbf{M}$. When we say $\mathcal{A}^1_M$, we refer to the process only up to **BeforeSplit**, excluding the renaming and splitting procedure.*

Proof. Similar to Lemma 7.1. By the symmetric structure of $\left|\Psi^{\mathsf{sk}}\right\rangle_{\text{⚓🏴‍☠️}}$. □

7.2 Entanglement Rigidity in Multi-Stage Monogamy-of-Entanglement Games

Define the Pauli operators

$$\sigma_X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \sigma_Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \sigma_Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Lemma 7.6 (EPR Pair Rigidity, [VZ21], Lemma 5.4)**.** *Let $|\psi\rangle_{\mathbf{AB}} \in (\mathbb{C}^2)^{\otimes n}_{\mathbf{A}} \otimes \mathcal{H}_{\mathbf{B}}$, where $\mathcal{H}_{\mathbf{B}}$ is arbitrary. Suppose that for every $a \in \{0,1\}^n$ there exist observables $X^{\mathbf{B}}(a)$ and $Z^{\mathbf{B}}(a)$ on $\mathcal{H}_{\mathbf{B}}$ such that*

$$\forall W \in \{X, Z\}, \quad \mathop{\mathbb{E}}_a\left[\left\|\left(\sigma^{\mathbf{A}}_W(a) - W^{\mathbf{B}}(a)\right)|\psi\rangle\right\|^2\right] \leq \varepsilon,$$

for some $0 \leq \varepsilon \leq 1$ where $\sigma_W^{\mathbf{A}}(a) = \sigma_W^{a_1} \otimes \sigma_W^{a_2} \otimes \cdots \otimes \sigma_W^{a_n}$ and the expectation is under the uniform distribution over $a \in \{0,1\}^n$. Then there exists an isometry

$$\Phi_{\mathbf{B}} : \mathcal{H}_{\mathbf{B}} \to \left(\left(\mathbb{C}^2\right)^{\otimes n}\right)_{\mathbf{B}'} \otimes \mathcal{H}_{\hat{\mathbf{B}}}$$

such that

$$\mathsf{Tr}\left(\langle \mathsf{EPR}|^{\otimes n}_{\mathbf{AB}'} \left((I_{\mathbf{A}} \otimes \Phi_{\mathbf{B}}\right)\left(|\psi\rangle_{\mathbf{AB}} \langle\psi|\right)\right) |\mathsf{EPR}\rangle^{\otimes n}_{\mathbf{AB}'}\right) = 1 - O\left(\varepsilon^{1/2}\right).$$

Moreover, the isometry $\Phi_{\mathbf{B}}$ can be implemented as an $O(n)$-size quantum circuit acting on $\mathcal{H}_{\mathbf{B}}$ as well as some ancilla qubits, and that uses controlled gates for $X^{\mathbf{B}}(a)$ and $Z^{\mathbf{B}}(b)$ (controlled on an ancilla register of dimension 2^n that contains a or b) as black boxes. Precisely, $\Phi^{\mathbf{B}}$ is defined by

$$\Phi_{\mathbf{B}} |\phi\rangle_{\mathbf{B}} = \left(\frac{1}{2^n} \sum_{a,b} X^{\mathbf{B}}(a) Z^{\mathbf{B}}(b) \otimes \sigma_X(a)\sigma_Z(b) \otimes I\right) |\phi\rangle_{\mathbf{B}} |\mathsf{EPR}\rangle^{\otimes n},$$

and letting $\mathcal{H}_{\hat{\mathbf{B}}} = \mathcal{H}_{\mathbf{B}} \otimes \left(\mathbb{C}^2\right)^{\otimes n}$, with register $\mathbf{B}'$ associated with the last n copies of $\mathbb{C}^2$.

Corollary 7.7 (EPR Pair Rigidity (corollary))**.** *Let $|\psi\rangle_{\mathbf{AB}} \in \left(\mathbb{C}^2\right)^{\otimes n}_{\mathbf{A}} \otimes \mathcal{H}_{\mathbf{B}}$, where $\mathcal{H}_{\mathbf{B}}$ is arbitrary. Let $\widetilde{\mathbf{B}}$ be the sub-register of $\mathbf{B}$ containing the first n qubits and define:*

$$\Pi_{\mathsf{eq}} = \sum_{x \in \mathbb{F}_2^n} |x\rangle_{\mathbf{A}} \langle x| \otimes |x\rangle_{\widetilde{\mathbf{B}}} \langle x| \otimes I_{\widetilde{\mathbf{B}}'}.$$

Suppose that there exists two isometries $\mathcal{U}_X^{\mathbf{B}}, \mathcal{U}_Z^{\mathbf{B}} : \mathcal{H}_{\mathbf{B}} \to \mathcal{H}_{\widetilde{\mathbf{B}}} \otimes \mathcal{H}_{\widetilde{\mathbf{B}}'}$ such that

$$\left\|\Pi_{\mathsf{eq}}(I_{\mathbf{A}} \otimes \mathcal{U}_Z^{\mathbf{B}}) |\psi\rangle\right\|^2 \geq 1 - \varepsilon \quad \text{and} \quad \left\|\Pi_{\mathsf{eq}}(H_{\mathbf{A}}^{\otimes n} \otimes \mathcal{U}_X^{\mathbf{B}}) |\psi\rangle\right\|^2 \geq 1 - \varepsilon.$$

Then there exists an isometry

$$\Phi_{\mathbf{B}} : \mathcal{H}_{\mathbf{B}} \to \left(\left(\mathbb{C}^2\right)^{\otimes n}\right)_{\mathbf{B}'} \otimes \mathcal{H}_{\hat{\mathbf{B}}}$$

such that

$$\mathsf{Tr}\left(\langle \mathsf{EPR}|^{\otimes n}_{\mathbf{AB}'} \left((I_{\mathbf{A}} \otimes \Phi_{\mathbf{B}}\right)\left(|\psi\rangle_{\mathbf{AB}} \langle\psi|\right)\right) |\mathsf{EPR}\rangle^{\otimes n}_{\mathbf{AB}'}\right) = 1 - O\left(\varepsilon^{1/2}\right).$$

Moreover, the isometry $\Phi_{\mathbf{B}}$ can be implemented as an $O(n)$-size quantum circuit acting on $\mathcal{H}_{\mathbf{B}}$ as well as some ancilla qubits, and that uses controlled gates for $\mathcal{U}_X^{\mathbf{B}}$ and $\mathcal{U}_Z^{\mathbf{B}}$ as black boxes.

Proof. We construct observables $W^{\mathbf{B}}(a)$ in Lemma 7.6 using $\mathcal{U}_X^{\mathbf{B}}$ and $\mathcal{U}_Z^{\mathbf{B}}$. Let (we abuse the notation of $\mathcal{U}^\dagger$ to mean the reverse procedure of $\mathcal{U}$)

$$W^{\mathbf{B}}(a) = \left(\mathcal{U}_W^{\mathbf{B}}\right)^\dagger \left(\sigma_Z^{\widetilde{\mathbf{B}}}(a) \otimes I_{\widetilde{\mathbf{B}}'}\right) \mathcal{U}_W^{\mathbf{B}} \quad \forall W \in \{X, Z\}.$$

We have

$$\|\Pi_{\mathsf{eq}}(I_{\mathbf{A}} \otimes \mathcal{U}_Z) |\psi\rangle\|^2 \geq 1 - \varepsilon \Rightarrow \mathop{\mathbb{E}}_a \left[\left\|\left(\sigma_Z^{\mathbf{A}}(a) - Z^{\mathbf{B}}(a)\right) |\psi\rangle\right\|^2\right] \leq 2\varepsilon$$
$$\left\|\Pi_{\mathsf{eq}}(H_{\mathbf{A}}^{\otimes n} \otimes \mathcal{U}_X) |\psi\rangle\right\|^2 \geq 1 - \varepsilon \Rightarrow \mathop{\mathbb{E}}_a \left[\left\|\left(\sigma_X^{\mathbf{A}}(a) - X^{\mathbf{B}}(a)\right) |\psi\rangle\right\|^2\right] \leq 2\varepsilon.$$

By Lemma 7.6, we obtain the result. □

Definition 7.8 (Multi-Stage Independent Monogamy-of-Entanglement Game)**.** *Let $\lambda \in \mathbb{N}$ be the security parameter and let $n(\lambda)$, $m(\lambda)$ be polynomials. Consider the following game between the challenger and an adversary $\mathcal{A} = (\mathcal{A}_M^0, \mathcal{A}_M^1, \mathcal{A}_M^2, \mathcal{A}_L^0, \mathcal{A}_L^1, \mathcal{A}_R^0, \mathcal{A}_R^1)$:*

1. *The challenger generates $\left(\mathsf{sk}, |\Psi^{\mathsf{sk}}\rangle_{\anchor\pirate}\right) \leftarrow \mathsf{GenAnchorState}(1^\lambda)$ and gives the register $\pirate$ to $\mathcal{A}_M^0$. All parties of the adversary are given oracle access to $\mathcal{O}_{T+v}$ and $\mathcal{O}_{S^\perp+u}$. Let $C_0 = T + v$ and $C_1 = S^\perp + u$.*
2. *$\mathcal{A}_M^0$ can choose to abort in this step. If it aborts, the output of the game is $\perp$. Otherwise, it generates a tripartite state on $\mathbf{LMR}$. It sends $\mathbf{L}$ to $\mathcal{A}_L^0$, sends $\mathbf{M}$ to $\mathcal{A}_M^1$ and sends $\mathbf{R}$ to $\mathcal{A}_R^0$.*
3. *The challenger samples $\theta_0 \xleftarrow{\$} \{0,1\}$ and gives θ_0 to $\mathcal{A}_M^1$. It also samples $b_0 \xleftarrow{\$} \mathbb{F}_2^m$, and gives $\mathcal{A}_M^1$ oracle access to $\mathcal{O}_{C_{\theta_0}}^{b_0}$. Then, $\mathcal{A}_M^1$ on $\mathbf{M}$ generates a tripartite state on $\mathbf{M}_L\mathbf{M}'\mathbf{M}_R$. It sends $\mathbf{M}_L$ to $\mathcal{A}_L^0$, sends $\mathbf{M}'$ to $\mathcal{A}_M^2$ and sends $\mathbf{M}_R$ to $\mathcal{A}_R^0$. We will name the time just after all operations of $\mathcal{A}_M^1$ are done and before renaming the registers and splitting them as **BeforeSplit**.*
4. *$\mathcal{A}_L^0$ and $\mathcal{A}_R^0$ are given access to $\mathcal{O}_{C_{\theta_0}}^{b_0}$. $\mathcal{A}_L^0$ on $\mathbf{LM}_L$ produces answer b_0^l and a state on register $\mathbf{L}'$ that is sent to $\mathcal{A}_L^1$. $\mathcal{A}_R^0$ on $\mathbf{RM}_R$ produces answer b_0^r and a state on register $\mathbf{R}'$ that is sent to $\mathcal{A}_R^1$.*
5. *The challenger samples $\theta_1 \xleftarrow{\$} \{0,1\}$ and gives θ_1 to $\mathcal{A}_M^2$. It also samples $b_1 \xleftarrow{\$} \mathbb{F}_2^m$ gives $\mathcal{A}_M^2$ oracle access to $\mathcal{O}_{C_{\theta_0}}^{b_0}$ and $\mathcal{O}_{C_{\theta_1}}^{b_1}$. It generates a bipartite state on $\mathbf{M}'_L\mathbf{M}'_R$. $\mathbf{M}'_L$ is given to $\mathcal{A}_L^1$ and $\mathbf{M}'_R$ is given to $\mathcal{A}_R^1$.*
6. *$\mathcal{A}_L^1$ and $\mathcal{A}_R^1$ are given access to $\mathcal{O}_{C_{\theta_0}}^{b_0}$ and $\mathcal{O}_{C_{\theta_1}}^{b_1}$. $\mathcal{A}_L^1$ on $\mathbf{L}'\mathbf{M}'_L$ generates the answer b_1^l. $\mathcal{A}_R^1$ on $\mathbf{R}'\mathbf{M}'_R$ generates the answer b_1^r. The adversary wins iff $b_0^l = b_0^r = b_0$ and $b_1^l = b_1^r = b_1$.*

Let $\mathsf{MultiStageIndependentMonogamy}(\mathcal{A}, 1^\lambda)$ be the random variable that takes value $1/0/\perp$ if the adversary $\mathcal{A}$ wins/loses/aborts in the above game, respectively.

Theorem 7.9. *Let $n(\lambda) \geq \lambda$ and $m_0(\lambda) = m_1(\lambda) = m(\lambda) \geq \lambda$ be polynomials of λ. There exists an efficient extractor $\mathcal{E}$ with access to sk and the register $\mathbf{M}$ such that the following holds. Take any efficient adversary $\mathcal{A}$ in the multi-stage monogamy-of-entanglement game described in Definition 7.8 such that*

$$\Pr\left[\mathsf{MultiStageIndependentMonogamy}(\mathcal{A}, 1^\lambda) \neq \perp\right] = \varepsilon'(\lambda)$$

and

$$\frac{\Pr\left[\mathsf{MultiStageIndependentMonogamy}(\mathcal{A}, 1^\lambda) = 1\right]}{\Pr\left[\mathsf{MultiStageIndependentMonogamy}(\mathcal{A}, 1^\lambda) \neq \perp\right]} = 1 - \varepsilon(\lambda)$$

for any function $\varepsilon(\lambda)$ and any inverse polynomial $\varepsilon'(\lambda)$.. Then running $\mathcal{E}^{\mathcal{A}_M^1}$ on the state $\mathbf{M}$ conditioned on $\mathcal{A}_M^0$ not aborting extracts a state on $\anchor\mathbf{X}$ that is very close to n EPR-pairs. More specifically, with $\mathbf{X}$ denoting the output register of $\mathcal{E}^{\mathcal{A}_M^1}$,

$$\mathbb{E}\left[\mathsf{Tr}\left(\langle\mathsf{EPR}|^{\otimes n}_{\anchor\mathbf{X}} \mathcal{E}^{\mathcal{A}_M^1}\rho_{\anchor\mathbf{M}}\left(\mathcal{E}^{\mathcal{A}_M^1}\right)^\dagger|\mathsf{EPR}\rangle^{\otimes n}_{\anchor\mathbf{X}}\right)\middle|\substack{\left(\mathsf{sk},|\Psi^{\mathsf{sk}}\rangle_{\anchor\pirate}\right)\leftarrow\mathsf{GenAnchorState}(1^\lambda)\\ \mathcal{A}_M^0 \text{ does not abort and produces } \rho_{\anchor\mathbf{M}} \text{ on } \mathbf{XM}}\right] \geq 1 - O\left(\varepsilon^{1/4}\right) - \mathsf{negl}(\lambda).$$

*Furthermore, $\mathcal{E}^{\mathcal{A}_M^1}$ only makes black-box access to $\mathcal{A}_M^1$ and acts on $\mathbf{M}$. When we say $\mathcal{A}_M^1$, we refer to the process only up to **BeforeSplit**, excluding the renaming and splitting procedure.*

Proof. Note that the game MultiStageIndependentMonogamy is just a MultiStageSearchMonogamy with $\frac{1}{4}$ probability, DualMultiStageSearchMonogamy with $\frac{1}{4}$ and something else with $\frac{1}{2}$ probability. So such adversary $\mathcal{A}$ must succeed in MultiStageSearchMonogamy and DualMultiStageSearchMonogamy with probability at least $1 - 4\varepsilon$. We can obtain two extractors $\mathcal{E}_S^{\mathcal{A}_M^1}$ and $\mathcal{E}_H^{\mathcal{A}_M^1}$ that, with

probability $1-O\left(\varepsilon^{1/2}\right)-\mathsf{negl}(\lambda)$, extract the standard basis measurement and the Hadamard basis measurement by Lemma 7.1 and Corollary 7.5. Then by Corollary 7.7, we can construct $\mathcal{E}^{\mathcal{A}_M^1}$ from $\mathcal{E}_S^{\mathcal{A}_M^1}$ and $\mathcal{E}_H^{\mathcal{A}_M^1}$. □

8 Warmup: Localizing Entanglement in the High-Success-Probability Regime

In this section, we present a warmup to the main results of the paper — a protocol allowing us to localize EPR-pair halves to within some small region of spacetime, built and proven secure using the techniques developed in the previous sections. This is a simplified variant of the entanglement localization protocol in Section 9.1, with the latter being a sequentially repeated version of the former. Sequential repetition will allow us to amplify extraction fidelity, even in the low-success-probability regime.

The definition and subsequent construction in this section will have a weaker form of extraction soundness than in the main result. In Section 9.1, we give a stronger, more general definition accompanied by a construction which satisfies it, needing no additional assumptions.

Definition 8.1 (Entanglement Localization, warmup version)**.** *An entanglement localization scheme consists of the following syntax:*

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, \mathbf{A}, \mathbf{B}$*: The setup algorithm takes the security parameter* 1^λ *and outputs the public parameter* pp*, the secret parameter* sp*, the extraction key* ek*, and a state on the bipartite register* $\mathbf{AB}$*.*
- $\mathsf{Localize}\left(\mathcal{P}(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp})\right)(\mathsf{pp}, L, t) \to \mathbf{B}', b$*: This is a positional protocol between a QPT prover holding a quantum register* $\mathbf{B}$ *and a PPT*[11] *verifier with input* sp*, further specified by public inputs* pp *and the claimed location* (L, t)*. The prover's output is a quantum register* $\mathbf{B}'$ *and the verifier's output is an acceptance indicator* $b \in \{\top, \bot\}$*.*

It should satisfy the following properties — parameterized by completeness probability $\alpha(\lambda)$*, extraction fidelity* $\beta(\lambda, \eta)$*, and localization parameter* $\Delta(\lambda)$ *— for all* $\lambda \in \mathbb{N}$*.*

- α-***Completeness****: For any location* L *and time* t*, there is a QPT prover* $\mathcal{P}$ *at position* (L, t) *such that*

$$\Pr\left[b = \top : \begin{array}{r} \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, \mathbf{A}, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathbf{B}', b \leftarrow \mathsf{Localize}\left(\mathcal{P}(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp})\right)(\mathsf{pp}, L, t) \end{array}\right] \geq \alpha(\lambda).$$

- (β, Δ)-***Extraction Soundness****: For any* L, t *and QPT prover* $\mathcal{P}^*$*, let*

$$1 - \eta(\lambda) := \Pr\left[b = \top : \begin{array}{r} \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, \mathbf{A}, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathbf{B}', b \leftarrow \mathsf{Localize}\left(\mathcal{P}^*(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp})\right)(\mathsf{pp}, L, t) \end{array}\right]$$

be the probability that $\mathcal{P}^*$ *passes the verification. Then there exists a local extractor* $\mathcal{E}$ *that extracts* $|\mathsf{EPR}\rangle^{\otimes n}$ *from registers near location* L *at time* t *with fidelity at least* $\beta(\lambda, \eta)$*:*

$$\mathbb{E}\left[\langle \mathsf{EPR}|^{\otimes n}_{\mathbf{AB}^*} \rho_{\mathbf{AB}^*} |\mathsf{EPR}\rangle^{\otimes n}_{\mathbf{AB}^*} : \begin{array}{r} \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, \mathbf{A}, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathbf{B}^* \leftarrow \mathsf{register}[L_\Delta \mathbin{@} t]\left(\mathsf{Localize}\left(\mathcal{P}^*(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp})\right)(\mathsf{pp}, L, t)\right) \\ \mathbf{B}^* \leftarrow \mathcal{E}(\mathsf{pp}, \mathsf{ek}, \mathbf{B}^*) \end{array}\right] \geq \beta(\lambda, \eta)\,,$$

[11] In general, we should allow any quantum verifier. Nevertheless, our construction achieves classical verification.

where $\rho_{\mathbf{AB}^*}$ is the joint state on $\mathbf{AB}^*$ after running the extractor, and $L_\Delta = [L-\Delta, L+\Delta]$. Here we overload the notation $\mathbf{B}^*$, meaning that we update the register $\mathbf{B}^*$ by applying a channel to it.

We say that the entanglement localization scheme is non-destructive *if it additionally satisfies the following property.*

- ***Non-destructive**: For any spatial location L and time t,*

$$\mathbb{E}\left[\mathrm{TD}\left(\rho_{\mathbf{AB}}, \rho_{\mathbf{AB}'}\right) : \begin{array}{r}\mathsf{pp}, \mathsf{sp}, \mathsf{ek}, \mathbf{A}, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathbf{B}', b \leftarrow \mathsf{Localize}\left(\mathcal{P}(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp})\right)(\mathsf{pp}, L, t)\end{array}\right] \leq \mathsf{negl}(\lambda)\ ,$$

where $\rho_{\mathbf{AB}}$ is the joint state on $\mathbf{AB}$ after Setup *and $\rho_{\mathbf{AB}'}$ is the joint state on $\mathbf{AB}'$ later after* Localize*.*

Remark 8.2. *Note that this definition does not require the extractor $\mathcal{E}$ to be efficient. That is, $\mathcal{E}$ can run for unbounded time, and, in the oracle model, can make unbounded queries to its oracle. This is still meaningful, as entanglement is an information-theoretic notion that cannot be duplicated even given unbounded time. Moreover, the extractor never runs in the real world — only in the analysis. Nevertheless, we note that our extractor* is *efficient and requires an "extraction key"* ek *that is sampled by* Setup *(but not known to the prover). In our construction,* $\mathsf{ek} = (U_{\mathsf{shift}}, u, v)$*. We mention this here because efficient extraction given* ek *may be a useful feature, depending on the application.*

Construction 8.3 (Entanglement Localization, warmup version)**.** *Let $\lambda \in \mathbb{N}$ be the security parameter and $n(\lambda), m(\lambda) \geq \lambda$ be polynomials. Take some localization parameter $\Delta(\lambda)$. We consider the following non-destructive entanglement localization scheme:*

- $\mathsf{Setup}(1^\lambda)$*:*
  1. *Generate $\left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}\right\rangle_{\mathbf{AB}}\right) \leftarrow \mathsf{GenAnchorState}(1^\lambda)$, and let $U_{\mathsf{shift}}, S, T, u, v$ be the basis-matrix, subspaces, and shifts determined by* sk*.*
  2. *Sample the public challenge oracle $\mathcal{O}_{\mathsf{Chal}} : \mathbb{F}_2^\lambda \to \{0,1\}$.*
  3. *Sample the public oracle $\mathcal{O} : \{0,1\} \times \mathbb{F}_2^\lambda \times \mathbb{F}_2^{3n} \to \mathbb{F}_2^m \cup \{\bot\}$ is based on a random oracle $\mathcal{O}_{\mathsf{random}} : \{0,1\} \times \mathbb{F}_2^\lambda \to \mathbb{F}_2^m$ that is **not** publicly available.*

$$\mathcal{O}(\theta, x, z) = \begin{cases} \mathcal{O}_{\mathsf{random}}(\theta, x) & \text{if } \theta = 0 \text{ and } z \in T + v, \\ \mathcal{O}_{\mathsf{random}}(\theta, x) & \text{if } \theta = 1 \text{ and } z \in S^\perp + u, \\ \bot & \text{otherwise} \end{cases}$$

  Then it generates $\left|\Psi^{\mathsf{sk}}\right\rangle_{\mathbf{AB}}$ and outputs:
  - *the public parameter $\mathsf{pp} = (\mathcal{O}_{\mathsf{Chal}}, \mathcal{O})$.*
  - *The secret parameters $\mathsf{sp} = (u, v)$. This is given to the verifiers $\mathcal{V}_L$ and $\mathcal{V}_R$.*
  - *The extraction key $\mathsf{ek} = (U_{\mathsf{shift}}, u, v)$.*
  - *The anchor state $\left|\Psi^{\mathsf{sk}}\right\rangle_{\mathbf{AB}'}$, with ⚓ renamed $\mathbf{A}$ and 🏴‍☠️ renamed $\mathbf{B}$.*
- $\mathsf{Localize}\left(\mathcal{P}(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp})\right)(\mathsf{pp}, L, t)$*: Let $\delta = \Delta/2$.*
  - *For $i = 0, 1$, $\mathcal{V}_L$ samples $x_{L,i} \xleftarrow{\$} \mathbb{F}_2^\lambda$ and broadcasts it at time $t + i\delta - (L+1)$. It expects responses $y_{L,i}$ at time $t + i\delta + (L+1)$.*
  - *For $i = 0, 1$, $\mathcal{V}_R$ samples $x_{R,i} \xleftarrow{\$} \mathbb{F}_2^\lambda$ and broadcasts it at time $t + i\delta - (1-L)$. It expects responses $y_{R,i}$ at time $t + i\delta + (1-L)$.*
  - *If any of these responses is missing, the verifier outputs $\bot$. Otherwise, the verifier outputs $\top$ if and only if for each $i = 0, 1$, one of the following holds:*

    * $\theta_i := \mathcal{O}_{\mathsf{Chal}}(x_{L,i} \oplus x_{R,i}) = 0$ *and* $y_{L,i} = y_{R,i} = \mathcal{O}(0, x_{L,i} \oplus x_{R,i}, v)$.
    * $\theta_i := \mathcal{O}_{\mathsf{Chal}}(x_{L,i} \oplus x_{R,i}) = 1$ *and* $y_{L,i} = y_{R,i} = \mathcal{O}(1, x_{L,i} \oplus x_{R,i}, u)$.
- –
- – *The honest prover consists of a single party always holding the* $\mathbf{B}$ *register at location* L*. For each* $i = 0, 1$*, it receives* $x_{L,i}, x_{R,i}$ *at time* $t + i\delta$*. If* $\theta_i = 0$*, it coherently evaluates* $\mathcal{O}(0, x_{L,i} \oplus x_{R,i}, \cdot)$ *on the* $\mathbf{B}$ *register in the standard basis and broadcasts the result. If* $\theta_i = 1$*, it coherently evaluates* $\mathcal{O}(1, x_{L,i} \oplus x_{R,i}, \cdot)$ *on the* $\mathbf{B}$ *register in the Hadamard basis and broadcasts the result.*

Theorem 8.4 (NDEL security, warmup version)**.** *The construction in Construction 8.3 is a non-destructive entanglement localization scheme with* 1-***Completeness*** *and* (β, Δ)-***Extraction Soundness****, where*

$$\beta(\lambda, \eta) = 1 - O\left(\eta(\lambda)^{1/4}\right) - \mathsf{negl}(\lambda).$$

Proof. For completeness, by Fact 4.5, the standard basis measurement on $\mathbf{B}$ will always give a vector in $T + v$ and the Hadamard basis measurement on $\mathbf{B}$ will always give a vector in $S^\perp + u$. Thus, the coherent query result will be $y_i = \mathcal{O}_{\mathsf{random}}(\theta_i, x_{L,i} \oplus x_{R,i})$ with probability 1. From there, non-destructiveness also follows, by an application of gentle measurement.

For extraction soundness, suppose that there exists a prover $\mathcal{P}^*$ that passes localize experiment at a point (L, t_0) with probability $1-\eta$. We will turn $\mathcal{P}^*$ into a prover $\mathcal{A} = (\mathcal{A}^0_M, \mathcal{A}^1_M, \mathcal{A}^0_L, \mathcal{A}^0_R, \mathcal{A}^2_M, \mathcal{A}^1_L, \mathcal{A}^1_R)$ for the MultiStageIndependentMonogamy game, by *chunking* the execution of $\mathcal{P}^*$ according to its spacetime geometry. This is pictured in Figure 10. Let $r(\lambda)$ be the total number of queries to all oracles that $\mathcal{P}^*$ makes (from any location and at any time), then let $\left\{F_k : \{\mathcal{O}_{\mathsf{Chal}}(\cdot), \mathcal{O}(0, \cdot, \cdot), \mathcal{O}(1, \cdot, \cdot)\} \times \mathbb{F}_2^\lambda \to \mathbb{F}_2^m\right\}_k$ be a family of $2r$-wise independent hash functions. The first entry of the input is the 'name' of the oracle that F_k is used to simulate. Let $\delta := \Delta/2$, $t_0 = t$, and $t_1 := t + \delta$. For notational clarity, we overload p, t to refer to position and time axes, respectively. We construct each component of $\mathcal{A}$ as follows:

1. $\mathcal{A}^0_M$ receives register $\mathbf{B} :=$ , membership oracles $\mathcal{O}_{C_0}$ and $\mathcal{O}_{C_1}$ (where $C_0 = T + v$ and $C_1 = S^\perp + u$, as in Definition 7.8) from the challenger. It then prepares to simulate the localization experiment by sampling a hash function F_k and secret-shared challenges $x_{L,0}, x_{R,0}, x_{L,1}, x_{R,1}$. Define $\mathsf{info} := (k, x_{L,0}, x_{R,0}, x_{L,1}, x_{R,1})$.
   $\mathcal{A}^0_M$ then begins simulating $\mathcal{P}^*$ on input $\mathbf{B}$, answering all of $\mathcal{P}^*$'s queries to $\mathcal{O}$ as follows: a query (θ, x, z) is answered by checking membership in the corresponding coset using $\mathcal{O}_{C_\theta}$, and then outputting $F_k(\theta, x)$ in place of $\mathcal{O}_{\mathsf{random}}(\theta, x)$ if the membership check passed. Crucially, $\mathcal{P}^*$ is only simulated over a carefully chosen region of spacetime: $\mathcal{A}^0_M$ freezes each of the simulated prover's computations and messages *just before* they would touch the spacetime line $t - p = t_0 - L$ (on the right of L) or $t + p = t_0 + L$ (on the left of L). This cutoff is illustrated by a pair of dashed lines in Figure 10. Let $\mathbf{L}$ be the register that stores info and all simulated parties/messages with $p \in [-1, L - \delta]$ when the simulation ends. Similarly, let $\mathbf{R}$ be the register that stores info and all simulated parties/messages with $p \in [L + \delta, 1]$ when the simulation ends. Finally, let $\mathbf{M}$ be the register that stores info and all parties/messages with $p \in [L - \delta, L + \delta]$ when the simulation ends.
2. $\mathcal{A}^1_M$ receives $\mathbf{M}$ from $\mathcal{A}^0_M$, basis θ_0, and oracle access to $\mathcal{O}^{b_0}_{C_{\theta_0}}$ from the challenger. It then resumes simulation of all of $\mathcal{P}^*$'s (real-time) computations and messages which would take place in the spacetime triangle – colored green in Figure 10 – described by the constraints
   - $t - p \geq t_0 - L$;
   - $t + p \geq t_0 + L$;
   - $t < t_1$,

and halts simulation just before exiting this region. During this phase, the simulated oracle $\mathcal{O}$ is reprogrammed so that on a query (θ, x, z) with $(\theta, x) = (\theta_0, x_{L,0} \oplus x_{R,0})$, $\mathcal{A}^1_M$ answers with $\mathcal{O}^{b_0}_{C_{\theta_0}}(z)$. The other simulated oracle $\mathcal{O}_{\mathsf{Chal}}$ is reprogrammed so that on the query $x = x_{L,0} \oplus x_{R,0}$ it outputs θ_0. Other oracle outputs not mentioned here are still simulated using the membership oracles and F_k. Let $\mathbf{M}_L$ be the register that stores θ_0 and all parties/messages with $p \in [L - \delta, L)$ when the simulation ends. Similarly, let $\mathbf{M}_R$ be the register that stores θ_0 and all parties/messages with $p \in (L, L + \delta]$ when the simulation ends. Finally, append θ_0 to info and let $\mathbf{M}'$ be the register that stores info and all parties/messages at the spacetime point (L, t_1).

3. $\mathcal{A}^0_L$ receives $\mathbf{L}$ from $\mathcal{A}^0_M$, $\mathbf{M}_L$ from $\mathcal{A}^1_M$, and oracle access to $\mathcal{O}^{b_0}_{C_{\theta_0}}$ from the challenger. It then resumes simulation of all of $\mathcal{P}^*$'s real-time computations and messages which would take place in the spacetime trapezoid – colored red in Figure 10 – described by the constraints
   - $p \geq -1$;
   - $t \geq t_1$;
   - $t + p \geq t_0 + L$;
   - $t + p < t_1 + L$,

   and halts simulation just before exiting this region. During this simulation, $\mathcal{O}$ is reprogrammed in the same way as for $\mathcal{A}^1_M$. If the simulated prover did not send a classical message which would be received by $\mathcal{V}_L$ at time $t_0 + L + 1$, $\mathcal{A}^0_L$ outputs $\perp$. Otherwise, $\mathcal{A}^0_L$ uses $\mathcal{P}^*$'s response message sent to $\mathcal{V}_L$ as its output b^L_0. Let $\mathbf{L}'$ be the register that stores info and all parties/messages with $p \in [-1, L)$ when $\mathcal{A}^0_L$'s simulation ends.
4. $\mathcal{A}^0_R$ receives $\mathbf{R}$ from $\mathcal{A}^0_M$, $\mathbf{M}_R$ from $\mathcal{A}^1_M$, and oracle access to $\mathcal{O}^{b_0}_{C_{\theta_0}}$ from the challenger. It then resumes simulation of all of $\mathcal{P}^*$'s real-time computations and messages which would take place in the spacetime trapezoid – colored blue in Figure 10 – described by the constraints
   - $p \leq 1$;
   - $t \geq t_1$;
   - $t - p \geq t_0 - L$;
   - $t - p < t_1 - L$,

   and halts simulation just before exiting this region. During this simulation, $\mathcal{O}$ is reprogrammed in the same way as for $\mathcal{A}^1_M$. If the simulated prover did not send a classical message which would be received by $\mathcal{V}_R$ at time $t_0 + 1 - L$, $\mathcal{A}^0_R$ outputs $\perp$. Otherwise, $\mathcal{A}^0_R$ uses $\mathcal{P}^*$'s response message sent to $\mathcal{V}_R$ as its output b^R_0. Let $\mathbf{R}'$ be the register that stores info and all parties/messages with $p \in (L, 1]$ when $\mathcal{A}^0_R$'s simulation ends.
5. $\mathcal{A}^2_M$ receives $\mathbf{M}'$ from $\mathcal{A}^1_M$, basis θ_1, and oracle access to $\mathcal{O}^{b_0}_{C_{\theta_0}}$ and $\mathcal{O}^{b_1}_{C_{\theta_1}}$ from the challenger. It then resumes simulation of all of $\mathcal{P}^*$'s real-time computations and messages which would take place exactly at the point (L, t_1). On top of reprogramming the oracle $\mathcal{O}$ in the same way as for $\mathcal{A}^1_M$, we add two additional reprograms:
   - On a query (θ, x, z) to $\mathcal{O}$ with $(\theta, x) = (\theta_1, x_{L,1} \oplus x_{R,1})$, it answers with $\mathcal{O}^{b_1}_{C_{\theta_1}}(z)$.
   - On the query $x = x_{L,1} \oplus x_{R,1}$ to $\mathcal{O}_{\mathsf{Chal}}$ it outputs θ_1.

   Let $\mathbf{M}'_L$ be the register that stores θ_1 and all messages sent to the left from L at time t_1. Similarly, let $\mathbf{M}'_R$ be the register that stores θ_1 and all messages sent to the right from L at time t_1.
6. $\mathcal{A}^1_L$ receives $\mathbf{L}'$ from $\mathcal{A}^0_L$, $\mathbf{M}'_L$ from $\mathcal{A}^2_M$, and oracle access to $\mathcal{O}^{b_0}_{C_{\theta_0}}$ and $\mathcal{O}^{b_1}_{C_{\theta_1}}$ from the challenger. It then resumes simulation of all of $\mathcal{P}^*$'s real-time computations and messages which would take place in the spacetime line segment described by

- $p \in [-1, L)$;
- $t + p = t_1 + L$,

and halts simulation just before exiting this region. $\mathcal{O}$ is reprogrammed in the same way as for $\mathcal{A}_M^2$. If the simulated prover did not send a classical message which would be received by $\mathcal{V}_L$ at time $t_1 + L + 1$, $\mathcal{A}_L^1$ outputs $\perp$. Otherwise, $\mathcal{A}_L^1$ uses that response message as its output b_1^L.

7. $\mathcal{A}_R^1$ receives $\mathbf{R}'$ from $\mathcal{A}_R^0$, $\mathbf{M}'_R$ from $\mathcal{A}_M^2$, and oracle access to $\mathcal{O}_{C_{\theta_0}}^{b_0}$ and $\mathcal{O}_{C_{\theta_1}}^{b_1}$ from the challenger. It then resumes simulation of all of $\mathcal{P}^*$'s real-time computations and messages which would take place in the spacetime line segment described by

   - $p \in (L, 1]$;
   - $t - p = t_1 - L$,

   and halts simulation just before exiting this region. $\mathcal{O}$ is reprogrammed in the same way as for $\mathcal{A}_M^2$. If the simulated prover did not send a classical message which would be received by $\mathcal{V}_R$ at time $t_1 + 1 - L$, $\mathcal{A}_R^1$ outputs $\perp$. Otherwise, $\mathcal{A}_R^1$ uses that response message as its output b_1^R.

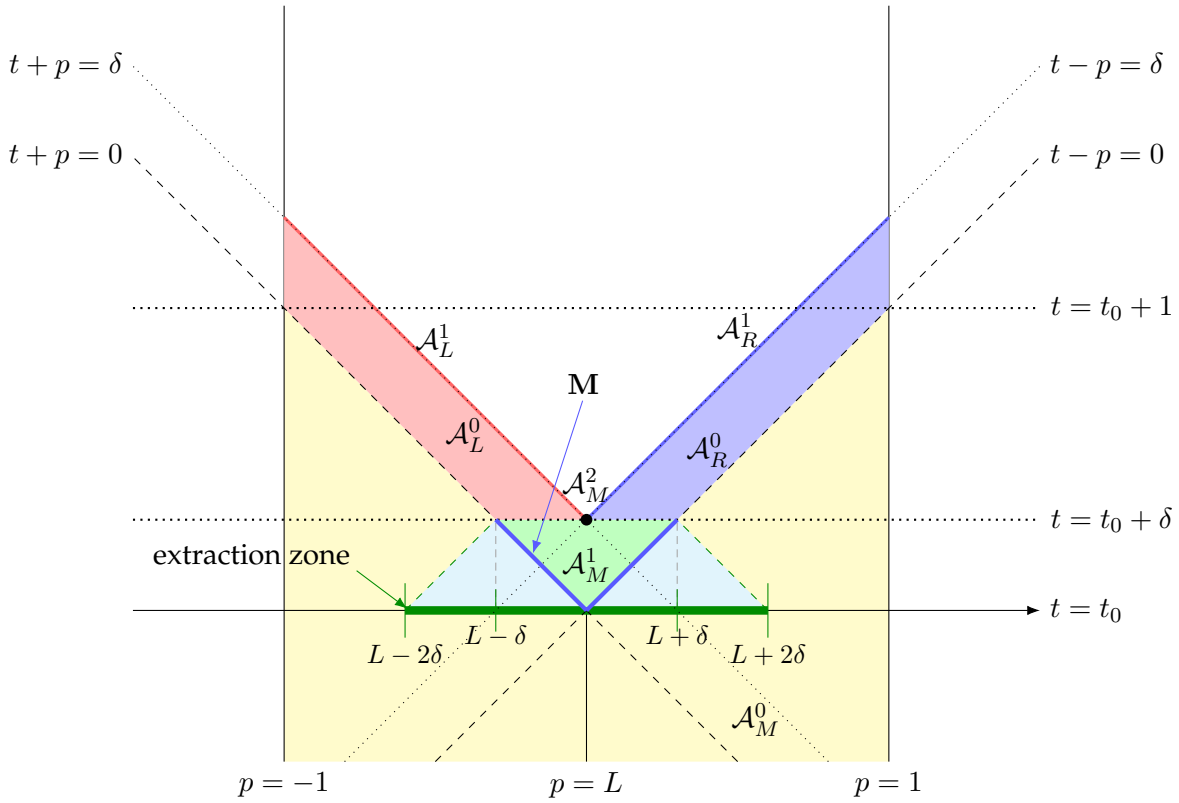

Figure 10: Spacetime decomposition of the prover $\mathcal{A}$ for our localization reduction, shown with $L = 0$.

Claim 8.5. *For any prover $\mathcal{P}^*$ that passes with probability $1 - \eta$, the probability that $\mathcal{A}$ wins Definition 7.8 is at least $1 - \eta -$* negl.

Proof. By [Zha12, Theorem 3.1], the $2r$-wise independent hash function perfectly simulates the random oracle for $\mathcal{P}^*$. Also we can see that $\mathcal{A}$ perfectly simulates $\mathcal{P}^*$, since for any event simulated

in $\mathcal{A}$, all other events that can have a causal effect on that event have already been simulated. The only difference is that the oracles $\mathcal{O}_{\mathsf{Chal}}, \mathcal{O}$ are both reprogrammed twice, but by Lemma 3.1, the resulting experiment is negligibly close to the actual Localize experiment. And there, $\mathcal{P}^*$ eventually outputs $y_{L,0} = y_{R,0} = \mathcal{O}_{\mathsf{random}}(\theta_0, x_{L,0} \oplus x_{R,0})$ and $y_{L,1} = y_{R,1} = \mathcal{O}_{\mathsf{random}}(\theta_1, x_{L,1} \oplus x_{R,1})$ with probability $1 - \eta$. Therefore $\mathcal{A}$ will produce the correct outputs $b_0^L, b_0^R, b_1^L, b_1^R$ with probability $1 - \eta - \mathsf{negl}(\lambda)$. □

By the above claim and Theorem 7.9, there exists an extractor $\mathcal{E}^{\mathcal{A}_M^1}$ acting on $\mathbf{M}$ with access to $\mathcal{A}_M^1$ that extracts n EPR pairs with expected fidelity

$$1 - O\left(\eta^{1/4}\right) - \mathsf{negl}.$$

Recall that $\mathbf{M}$ is the joint register of all parties/messages with $p \in [L-\delta, L+\delta]$ with $t - t_0 = |p - L|$, corresponding to the two bottom boundaries of the green triangle in Figure 10. These registers can be derived from access to all parties/messages of $\mathcal{P}^*$ in the spacetime region $[L-2\delta, L+2\delta] \times \{t_0\}$ by forward time-evolution, as pictured in the diagram. Note that this is exactly the extraction region $L_\Delta \mathbin{@} t_0$ from the definition, since $2\delta = \Delta$. Now, let us define our final extractor $\mathcal{E}$ that takes as input pp, ek, and $\mathsf{register}[L_\Delta \mathbin{@} t_0]$ and outputs something close to $|\mathsf{EPR}\rangle^{\otimes n}$. It does the following:

- Recover the joint register $\mathbf{M}$ via forward-simulation.
- Use the extraction key $\mathsf{ek} = (U_{\mathsf{shift}}, u, v)$ and run $\mathcal{E}^{\mathcal{A}_M^1}$ on $(\mathsf{ek}, \mathbf{M})$ with oracle access to oracles in pp.

□

9 Localizing Quantum Information

In this section, we present our formal definitions and constructions of the localization primitives mentioned in the introduction: entanglement localization, trajectory verification, state localization, and functionality localization. These definitions will have a stronger extraction soundness condition than was shown in the warmup, which is much more meaningful against low-success-probability adversaries. To achieve this stronger soundness, our constructions in this section employ a sequential repetition technique — each protocol is the sequential chaining of $O(\gamma)$-many two-round protocols.

9.1 Entanglement Localization

Definition 9.1 (Entanglement Localization)**.** *An entanglement localization scheme is parameterized by the security parameter λ and a bipartite state σ[12]. It consists of the following syntax:*

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, \mathbf{A}, \mathbf{B}$*: The setup algorithm takes the security parameter 1^λ and outputs the public parameter* pp*, the secret parameter* sp*, the extraction key* ek*, and a state on the bipartite register* $\mathbf{AB}$*.*
- $\mathsf{Localize}\left(\mathcal{P}(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp})\right)(\mathsf{pp}, L, t) \to \mathbf{B}', b$*: This is a positional protocol between a QPT prover holding a quantum register* $\mathbf{B}$ *and a PPT verifier with input* sp*, further specified by the following*

[12] In reality, we localize a family of bipartite states indexed by λ, i.e. $\Sigma = \{\sigma_\lambda\}_\lambda$. As is standard in these contexts, we suppress the dependence on λ for notational convenience, and simply write σ.

public inputs: public parameters pp*, spatial location* $L \in [-1, 1]$*, and time* t*. The prover's output is a quantum register* $\mathbf{B}'$ *and the verifier's output is an acceptance indicator* $b \in \{\top, \bot\}$*.*

It should satisfy the following properties — parameterized by completeness probability $\alpha(\lambda)$*, extraction fidelity* $\beta(\lambda, \eta)$*, extraction probability* $\beta'(\lambda, \eta)$*, and localization parameter* $\Delta(\lambda)$ *— for all* $\lambda \in \mathbb{N}$*.*

- α-***Completeness****: For any location* L *and time* t*, there is a QPT prover* $\mathcal{P}$ *at position* (L, t) *such that*

$$\Pr\left[b = \top : \begin{array}{r} \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, \mathbf{A}, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathbf{B}', b \leftarrow \mathsf{Localize}\left(\mathcal{P}(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp})\right)(\mathsf{pp}, L, t) \end{array}\right] \geq \alpha(\lambda).$$

- (β, β', Δ)-***Extraction Soundness****: For any* L, t *and QPT prover* $\mathcal{P}^*$*, let*

$$\eta(\lambda) := \Pr\left[b = \top : \begin{array}{r} \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, \mathbf{A}, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathbf{B}', b \leftarrow \mathsf{Localize}\left(\mathcal{P}^*(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp})\right)(\mathsf{pp}, L, t) \end{array}\right]$$

be $\mathcal{P}^*$*'s success probability. Assume* $\eta(\lambda) \geq 1/p(\lambda)$ *for some polynomial* p*. Then there exists a local extractor* $\mathcal{E}$ *acting on registers near location* L *at time* t *such that the probability of extracting* σ_λ *with fidelity at least* $\beta(\lambda, \eta)$ *is at least* $\beta'(\lambda, \eta)$*:*

$$\Pr\left[F\left(\sigma_\lambda, \rho_{\mathbf{AB}^*}\right) \geq \beta(\lambda, \eta) \ : \ \begin{array}{r} \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, \mathbf{A}, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathbf{B}^* \leftarrow \mathsf{register}[L_\Delta \mathbin{@} t]\left(\mathsf{Localize}\left(\mathcal{P}^*(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp})\right)(\mathsf{pp}, L, t)\right) \\ \mathbf{B}^* \leftarrow \mathcal{E}(\mathsf{pp}, \mathsf{ek}, \mathbf{B}^*) \end{array}\right] \geq \beta'(\lambda, \eta)\,,$$

where $\rho_{\mathbf{AB}^*}$ *is the joint state on* $\mathbf{AB}^*$ *after running the extractor, and* $L_\Delta = [L - \Delta, L + \Delta]$*.*

We say that the entanglement localization scheme is non-destructive *if it additionally satisfies the following property.*

- ***Non-destructive****: For any spatial location* L *and time* t*,*

$$\mathbb{E}\left[\mathrm{TD}\left(\rho_{\mathbf{AB}}, \rho_{\mathbf{AB}'}\right) : \begin{array}{r} \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, \mathbf{A}, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathbf{B}', b \leftarrow \mathsf{Localize}\left(\mathcal{P}(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp})\right)(\mathsf{pp}, L, t) \end{array}\right] \leq \mathsf{negl}(\lambda)\,,$$

where $\rho_{\mathbf{AB}}$ *is the joint state on* $\mathbf{AB}$ *after* Setup *and* $\rho_{\mathbf{AB}'}$ *is the joint state on* $\mathbf{AB}'$ *later after* $\mathsf{Localize}$*.*

Construction 9.2 (Entanglement Localization Scheme)**.** *Let* $\lambda \in \mathbb{N}$ *be the security parameter and* $n(\lambda), m(\lambda) \geq \lambda$ *be polynomials. Take a localization parameter* $\Delta(\lambda) \leq 1$ *and a repetition factor* $\gamma(\lambda)$*. We consider the following non-destructive entanglement localization scheme:*

- *There are two verifiers:* $\mathcal{V}_L$ *at* -1 *and* $\mathcal{V}_R$ *at* 1*.*
- $\mathsf{Setup}(1^\lambda)$*:*
  1. *Generate* $\left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}\right\rangle_{\mathbf{AB}}\right) \leftarrow \mathsf{GenAnchorState}(1^\lambda)$*, and let* $U_{\mathsf{shift}}, S, T, u, v$ *be the basis-matrix, subspaces, and shifts determined by* sk*.*
  2. *Sample the public challenge oracle* $\mathcal{O}_{\mathsf{Chal}} : \mathbb{F}_2^\lambda \to \{0, 1\}$*.*
  3. *Sample the public oracle* $\mathcal{O} : \{0, 1\} \times \mathbb{F}_2^\lambda \times \mathbb{F}_2^{3n} \to \mathbb{F}_2^m \cup \{\bot\}$ *is defined with respect to a random oracle* $\mathcal{O}_{\mathsf{random}} : \{0, 1\} \times \mathbb{F}_2^\lambda \to \mathbb{F}_2^m$ *that is* ***not*** *publicly available:*

$$\mathcal{O}(\theta, x, z) = \begin{cases} \mathcal{O}_{\mathsf{random}}(\theta, x) & \text{if } \theta = 0 \text{ and } z \in T + v, \\ \mathcal{O}_{\mathsf{random}}(\theta, x) & \text{if } \theta = 1 \text{ and } z \in S^\perp + u, \\ \bot & \text{otherwise.} \end{cases}$$

Then Setup *outputs:*

- *the public parameter* $\mathsf{pp} = (\mathcal{O}_{\mathsf{Chal}}, \mathcal{O})$.
- *The secret parameter* $\mathsf{sp} = (u, v)$
- *The extraction key* $\mathsf{ek} = (U_{\mathsf{shift}}, u, v)$.
- *The anchor state* $\left|\Psi^{\mathsf{sk}}\right\rangle_{\mathbf{AB}'}$, *with* ⚓ *renamed* $\mathbf{A}$ *and* 🏴‍☠️ *renamed* $\mathbf{B}$.

- $\mathsf{Localize}\,(\mathcal{P}(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp}))\,(\mathsf{pp}, L, t)$: *Let* $\delta = \Delta/(\gamma+2)$ *and define checkpoints* $t_i = t + i\delta$ *for* $i = 0, 1, \ldots, \gamma$.
  - *For* $i = 0, 1, \cdots, \gamma$, $\mathcal{V}_L$ *samples* $x_{L,i} \xleftarrow{\$} \mathbb{F}_2^\lambda$ *and broadcasts it at time* $t_i - (L+1)$. *It expects responses* $y_{L,i}$ *at time* $t_i + (L+1)$.
  - *For* $i = 0, 1, \cdots, \gamma$, $\mathcal{V}_R$ *samples* $x_{R,i} \xleftarrow{\$} \mathbb{F}_2^\lambda$ *and broadcasts it at time* $t_i - (1-L)$. *It expects responses* $y_{R,i}$ *at time* $t_i + (1-L)$.
  - *If any of these responses is missing, the verifier outputs* $\perp$. *Otherwise, the verifier outputs* $\top$ *if and only if for each* $i = 0, 1, \cdots, \gamma$, *one of the following holds:*
    - $\theta_i := \mathcal{O}_{\mathsf{Chal}}(x_{L,i} \oplus x_{R,i}) = 0$ *and* $y_{L,i} = y_{R,i} = \mathcal{O}(0, x_{L,i} \oplus x_{R,i}, v)$.
    - $\theta_i := \mathcal{O}_{\mathsf{Chal}}(x_{L,i} \oplus x_{R,i}) = 1$ *and* $y_{L,i} = y_{R,i} = \mathcal{O}(1, x_{L,i} \oplus x_{R,i}, u)$.
  - *The honest prover consists of a single party always holding the* $\mathbf{B}$ *register at location* L. *For each* $i = 0, 1, \ldots, \gamma$, *it receives* $x_{L,i}, x_{R,i}$ *at time* t_i. *If* $\theta_i = 0$, *it coherently evaluates* $\mathcal{O}(0, x_{L,i} \oplus x_{R,i}, \cdot)$ *on the* $\mathbf{B}$ *register in the standard basis and broadcasts the result. If* $\theta_i = 1$, *it coherently evaluates* $\mathcal{O}(1, x_{L,i} \oplus x_{R,i}, \cdot)$ *on the* $\mathbf{B}$ *register in the Hadamard basis and broadcasts the result.*

Theorem 9.3. *The construction in Construction 9.2 is a non-destructive entanglement localization scheme for the state* $\sigma = |\mathsf{EPR}\rangle\langle\mathsf{EPR}|^{\otimes n}$ *in the classical oracle model with* 1-***Completeness*** *and* (β, β', Δ)-***Extraction Soundness***, *where*

$$\beta(\lambda, \eta) = 1 - O\left(\left(\frac{\log(1/\eta)}{\gamma}\right)^{1/4}\right) - \mathsf{negl}(\lambda), \qquad \beta'(\lambda, \eta) = 0.99\eta(\lambda).^{13}$$

In particular, since we only consider adversaries for which $1/\eta$ *is at most polynomial in* λ, *for any polynomial* $t(\lambda)$, *we can set* $\gamma = c\lambda t^4$ *for some constant* c *to achieve*

$$\beta(\lambda, \eta) \geq 1 - 1/t(\lambda)\,.$$

Proof. Completeness follows from Fact 4.5: the standard-basis measurement on $\mathbf{B}$ always gives a vector in $T + v$, and the Hadamard-basis measurement on $\mathbf{B}$ always gives a vector in $S^\perp + u$. Therefore the honest coherent query in every checkpoint returns $\mathcal{O}_{\mathsf{random}}(\theta_i, x_{L,i} \oplus x_{R,i})$, and the verifier accepts. The same fact implies non-destructiveness, since the oracle response is constant on the coset supporting the honest prover's state in the queried basis.

We prove extraction soundness. Let $\mathcal{P}^*$ be accepted with probability η. Let q_i denote the probability that $\mathcal{P}^*$ passes checkpoint i, conditioned on passing checkpoints $0, \ldots, i-1$. Since $\prod_{i=0}^{\gamma} q_i = \eta$, there exists an adjacent pair of checkpoints $i^\star, i^\star + 1$ such that

$$q_{i^\star} q_{i^\star+1} \geq \eta^{2/\gamma}.$$

Write $t_0 = t + i^\star\delta$ and $t_1 = t_0 + \delta$. Conditioned on passing checkpoints $0, \ldots, i^\star - 1$, the checkpoints $i^\star$ and $i^\star + 1$ form an identical experiment to the two-round localization test from Construction 8.3. We therefore apply the same spacetime chunking reduction from the proof of Theorem 8.4, with

[13] This 0.99 factor can be an arbitrary constant in $(0, 1)$, it will affect the constant in the big-O notation of $\beta(\lambda, \eta)$.

the two target challenges at these checkpoints. As usual, the reduction $\mathcal{A}$ simulates the hidden random oracle $\mathcal{O}_{\mathsf{random}}$ and the challenge oracle $\mathcal{O}_{\mathsf{Chal}}$ by a $2r(\lambda)$-wise independent function F_k, where r is the number of oracle queries $\mathcal{P}^*$ makes.

Importantly, due to the repetition, the first-stage reduction component $\mathcal{A}_M^0$ must simulate the Localize experiment from the beginning of time up through checkpoint $i^\star - 1$, thus ensuring that $\mathcal{A}_M^0$ is non-aborting iff the simulated prover $\mathcal{P}^*$ makes it to checkpoint $i^\star$. To do this, $\mathcal{A}_M^0$ samples challenges $(x_{L,i}, x_{R,i})$ for $i < i^*$ on its own and simulates the verifiers' interactions with the prover as in Localize. If the prover responds with $y_{L,i}, y_{R,i}$ such that $y_{L,i} \neq y_{R,i}$ or if $y_{L,i} \neq F_k(\theta_i, x_{L,i} \oplus x_{R,i})$, $\mathcal{A}_M^0$ aborts. The reduction reprograms $\mathcal{O}_{\mathsf{Chal}}$ at both $x_{L,i^\star} \oplus x_{R,i^\star}$ and $x_{L,i^\star+1} \oplus x_{R,i^\star+1}$, reprograms $\mathcal{O}$ at both $(\theta_{i^\star}, x_{L,i^\star} \oplus x_{R,i^\star}, \cdot)$ and $(\theta_{i^\star+1}, x_{L,i^\star+1} \oplus x_{R,i^\star+1}, \cdot)$; by Lemma 3.1, this changes the experiment by only a negligible amount. Thus the resulting Multi-StageIndependentMonogamy adversary does not abort with probability at least η and wins, conditioned on not aborting, with probability at least $q_{i^\star} q_{i^\star+1} - \mathsf{negl}(\lambda)$.

Then by Theorem 7.9, there exists an extractor $\mathcal{E}_{i^\star}^{\mathcal{A}_M^1}$ acting on the middle register $\mathbf{M}$ for checkpoints $i^\star, i^\star + 1$, which with probability $\geq \eta$ (correspond to the case that the prover does not abort before i^* checkpoint) outputs a state having the following **expected** fidelity with $|\mathsf{EPR}\rangle^{\otimes n}$:

$$1 - O\Big((1 - q_{i^\star} q_{i^\star+1})^{1/4}\Big) - \mathsf{negl}.$$

Since $q_{i^\star} q_{i^\star+1} \geq \eta^{2/\gamma}$, this expected fidelity is at least

$$1 - O\left(\left(1 - \eta^{2/\gamma}\right)^{1/4}\right) - \mathsf{negl}(\lambda) = 1 - O\left(\left(\frac{\log(1/\eta)}{\gamma}\right)^{1/4}\right) - \mathsf{negl}(\lambda),$$

where the last step uses

$$1 - \eta^{2/\gamma} = 1 - \exp\left(-\frac{2\log(1/\eta)}{\gamma}\right) \leq \frac{2\log(1/\eta)}{\gamma}.$$

By Markov inequality, with probability $\geq 0.99\eta$ the actual fidelity is

$$1 - O\left(\left(\frac{\log(1/\eta)}{\gamma}\right)^{1/4}\right) - \mathsf{negl}(\lambda)$$

with a different constant in the big-O notation.

It remains to translate the middle-register extractor into the extraction statement at the original time t. Let $\mathsf{Exp} = \mathsf{Localize}(\mathcal{P}^*(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp}))(\mathsf{pp}, L, t)$ be the real localization experiment. The middle register $\mathbf{M}$ is the same one used in the warmup proof: it lies on the two lower boundaries of the green triangle in Figure 10 for the time t_0. These registers can be derived from $\mathsf{register}[L_{2\delta} \mathbin{@} t_0](\mathsf{Exp})$ by forward simulation. Since $t_0 - t \leq \gamma\delta$, recovering this 2δ neighborhood at time t_0 from time t requires the region

$$\mathsf{register}[L_{(\gamma+2)\delta} \mathbin{@} t](\mathsf{Exp}) = \mathsf{register}[L_\Delta \mathbin{@} t](\mathsf{Exp}).$$

The final extractor performs this forward simulation and then runs $\mathcal{E}_{i^\star}^{\mathcal{A}_M^1}$. □

9.2 Trajectory Verification

Definition 9.4 (Trajectory Verification)**.** *A trajectory verification scheme is parameterized by the security parameter λ and a bipartite state σ[14]. It consists of the following syntax:*

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, \mathbf{A}, \mathbf{B}$*: The setup algorithm takes the security parameter 1^λ and outputs the public parameter* pp*, the secret parameter* sp*, the extraction key* ek*, and a state on the bipartite register* $\mathbf{AB}$*.*
- $\mathsf{Localize}\left(\mathcal{P}(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp})\right)(\mathsf{pp}, L) \to \mathbf{B}', b$*: The interactive verification protocol takes place between a QPT prover holding a quantum register* $\mathbf{B}$ *and a PPT verifier with input* sp*. The public inputs are* pp *and the description of a continuous trajectory $L : [0, \tau] \to [-1, 1]$ with $\left|\frac{\mathrm{d}L}{\mathrm{d}t}\right| < 1$. $L(t)$ is the prover's purported location at any time $0 \leq t \leq \tau$, where τ is some time limit included in the description of L. The prover's output is a quantum register $\mathbf{B}'$ and the verifier's output is an acceptance indicator $b \in \{\top, \bot\}$.*

It should satisfy the following properties — parameterized by completeness probability $\alpha(\lambda)$, extraction fidelity $\beta(\lambda, \eta)$, extraction probability $\beta'(\lambda, \eta)$, and localization parameter $\Delta(\lambda)$ — for all $\lambda \in \mathbb{N}$.

- α-***Completeness****: For any valid trajectory L, there is a QPT prover $\mathcal{P}$ moving along L such that*

$$\Pr\left[b = \top : \begin{array}{r} \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, \mathbf{A}, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathbf{B}', b \leftarrow \mathsf{Localize}\left(\mathcal{P}(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp})\right)(\mathsf{pp}, L) \end{array}\right] \geq \alpha(\lambda).$$

- (β, β', Δ)-***Extraction Soundness****: For any trajectory L and QPT prover $\mathcal{P}^*$, let*

$$\eta(\lambda) := \Pr\left[b = \top : \begin{array}{r} \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, \mathbf{A}, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathbf{B}', b \leftarrow \mathsf{Localize}\left(\mathcal{P}^*(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp})\right)(\mathsf{pp}, L) \end{array}\right]$$

be $\mathcal{P}^*$'s success probability in $\mathsf{Localize}$*. Assume $\eta(\lambda) \geq 1/p(\lambda)$ for some polynomial p. Then for any time $t \in [0, \tau]$, there exists a (potentially unbounded, see Remark 8.2) local extractor $\mathcal{E}$ acting on registers near the claimed trajectory at time t such that the probability of extracting σ with fidelity at least $\beta(\lambda, \eta)$ is at least $\beta'(\lambda, \eta)$:*

$$\Pr\left[F\left(\sigma, \rho_{\mathbf{AB}^*}\right) \geq \beta(\lambda, \eta) : \begin{array}{r} \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, \mathbf{A}, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathbf{B}^* \leftarrow \mathsf{register}[(L(t))_\Delta \mathbin{@} t]\left(\mathsf{Localize}\left(\mathcal{P}^*(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp})\right)(\mathsf{pp}, L)\right) \\ \mathbf{B}^* \leftarrow \mathcal{E}(\mathsf{pp}, \mathsf{ek}, \mathbf{B}^*) \end{array}\right] \geq \beta'(\lambda, \eta)\,,$$

where $\rho_{\mathbf{AB}^*}$ is the joint state on $\mathbf{AB}^*$ after running the extractor $\mathcal{E}$, and $(L(t))_\Delta = [L(t) - \Delta, L(t) + \Delta]$.

Now we present a construction for trajectory verification.

Construction 9.5 (Trajectory Verification Scheme)**.** *Let $\lambda \in \mathbb{N}$ be the security parameter and $n(\lambda), m(\lambda) \geq \lambda$ be polynomials. Take some localization parameter $\Delta(\lambda) \leq 1$ and a repetition factor $\gamma(\lambda)$. We consider the following trajectory verification scheme:*

- *There are two verifiers: $\mathcal{V}_L$ at -1 and $\mathcal{V}_R$ at 1.*
- $\mathsf{Setup}(1^\lambda)$*:*
  1. *Generate $\left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}\right\rangle_{\mathbf{AB}}\right) \leftarrow \mathsf{GenAnchorState}(1^\lambda)$, and let $U_{\mathsf{shift}}, S, T, u, v$ be the basis-matrix, subspaces, and shifts determined by* sk*.*

[14] Technically, a sequence of states indexed by λ – see Footnote 12.

2. *Sample the public challenge oracle* $\mathcal{O}_{\mathsf{Chal}} : \mathbb{F}_2^\lambda \to \{0,1\}$.
3. *Sample the public oracle* $\mathcal{O} : \{0,1\} \times \mathbb{F}_2^\lambda \times \mathbb{F}_2^{3n} \to \mathbb{F}_2^m \cup \{\bot\}$ *is defined with respect to a random oracle* $\mathcal{O}_{\mathsf{random}} : \{0,1\} \times \mathbb{F}_2^\lambda \to \mathbb{F}_2^m$ *that is* ***not*** *publicly available.*

$$\mathcal{O}(\theta, x, z) = \begin{cases} \mathcal{O}_{\mathsf{random}}(\theta, x) & \text{if } \theta = 0 \text{ and } z \in T + v, \\ \mathcal{O}_{\mathsf{random}}(\theta, x) & \text{if } \theta = 1 \text{ and } z \in S^\perp + u, \\ \bot & \text{otherwise} \end{cases}$$

Then Setup *outputs:*

- *the public parameter* $\mathsf{pp} = (\mathcal{O}_{\mathsf{Chal}}, \mathcal{O})$.
- *The secret parameter* $\mathsf{sp} = (u, v)$
- *The extraction key* $\mathsf{ek} = (U_{\mathsf{shift}}, u, v)$.
- *The anchor state* $\left|\Psi^{\mathsf{sk}}\right\rangle_{\mathbf{AB}'}$, *with* ⚓ *renamed* $\mathbf{A}$ *and* 🏴‍☠️ *renamed* $\mathbf{B}$.

- Localize $(\mathcal{P}(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp}))\,(\mathsf{pp}, L)$: *Suppose the claimed trajectory is* $L(t)$ *with* $t \in [0, \tau]$. *Let* $\delta = \Delta/(2\gamma + 2)$, *let* $N := \lceil \tau/\delta \rceil + \gamma$, *and define an extended trajectory*[15] $\widetilde{L}(s) := L(\min\{s, \tau\})$.
  - *For* $i = 0, 1, \cdots, N$, $\mathcal{V}_L$ *samples* $x_{L,i} \xleftarrow{\$} \mathbb{F}_2^\lambda$ *and broadcasts it at time* $i\delta - (\widetilde{L}(i\delta) + 1)$. *It expects responses* $y_{L,i}$ *at time* $i\delta + (\widetilde{L}(i\delta) + 1)$.
  - *For* $i = 0, 1, \cdots, N$, $\mathcal{V}_R$ *samples* $x_{R,i} \xleftarrow{\$} \mathbb{F}_2^\lambda$ *and broadcasts it at time* $i\delta - (1 - \widetilde{L}(i\delta))$. *It expects responses* $y_{R,i}$ *at time* $i\delta + (1 - \widetilde{L}(i\delta))$.
  - *If any of these responses is missing, the verifier outputs* $\bot$. *Otherwise, the verifier outputs* $\top$ *if and only if for each* $i = 0, 1, \cdots, N$, *one of the following holds:*
    - $\theta_i := \mathcal{O}_{\mathsf{Chal}}(x_{L,i} \oplus x_{R,i}) = 0$ *and* $y_{L,i} = y_{R,i} = \mathcal{O}(0, x_{L,i} \oplus x_{R,i}, v)$.
    - $\theta_i := \mathcal{O}_{\mathsf{Chal}}(x_{L,i} \oplus x_{R,i}) = 1$ *and* $y_{L,i} = y_{R,i} = \mathcal{O}(1, x_{L,i} \oplus x_{R,i}, u)$.
  - *The honest prover consists of a single party always holding the* $\mathbf{B}$ *register while moving along* L *and then remaining at* $L(\tau)$ *for the final* γ *checkpoints. For* $i = 0, 1, \cdots, N$, *it does the following:*
    - *Receive* $x_{L,i}, x_{R,i}$ *at time* $i\delta$. *If* $\theta_i = 0$, *it coherently evaluates* $\mathcal{O}(\theta_i, x_{L,i} \oplus x_{R,i}, \cdot)$ *on the* $\mathbf{B}$ *register in the standard basis and sends the result to* $\mathcal{V}_L, \mathcal{V}_R$. *If* $\theta_i = 1$, *it coherently evaluates* $\mathcal{O}(\theta_i, x_{L,i} \oplus x_{R,i}, \cdot)$ *on the* $\mathbf{B}$ *register in the Hadamard basis and sends the result to* $\mathcal{V}_L, \mathcal{V}_R$.

Theorem 9.6. *The construction in Construction 9.5 is a trajectory verification scheme for the state* $\sigma = |\mathsf{EPR}\rangle\langle\mathsf{EPR}|^{\otimes n}$ *in the classical oracle model with* 1-***Completeness*** *and* (β, β', Δ)-***Extraction Soundness*** *where*

$$\beta(\lambda, \eta) = 1 - O\left(\left(\frac{\log(1/\eta)}{\gamma}\right)^{1/4}\right) - \mathsf{negl}(\lambda), \qquad \beta'(\lambda, \eta) = 0.99\eta(\lambda).$$

In particular, since we only consider adversaries for which $1/\eta$ *is at most polynomial in* λ, *for any polynomial* $t(\lambda)$, *we can set* $\gamma = c\lambda t^4$ *for some constant* c *to achieve*

$$\beta(\lambda, \eta) \geq 1 - 1/t(\lambda)\ .$$

Proof. Completeness is identical to the completeness argument for Theorem 8.4: by Fact 4.5, the

[15] To enable extraction from all points along the trajectory, we must send a round of challenges *after* the prover has already reached the endpoint $(L(\tau), \tau)$. We define the extension $\widetilde{L}$ for notational convenience when sending these post-trajectory challenges.

honest prover's coherent query returns $\mathcal{O}_{\mathsf{random}}(\theta_i, x_{L,i} \oplus x_{R,i})$ in every checkpoint and the vessel state is not disturbed.

We prove extraction soundness by treating the protocol as a chained execution of the simple two-round localization scheme of Construction 8.3. Take a trajectory L, a QPT prover $\mathcal{P}^*$ succeeding in Localize with probability η, and fix a time $t \in [0, \tau]$. Let N and $\widetilde{L}$ be as in the construction. Let q_i denote the probability that $\mathcal{P}^*$ passes checkpoint i, conditioned on passing checkpoints $0, \ldots, i-1$, so $\prod_{i=0}^{N} q_i = \eta$. Set $k := \lceil t/\delta \rceil$ and consider the future window of adjacent checkpoint indices

$$I_t := \{k, k+1, \ldots, k+\gamma-1\}.$$

Since $N = \lceil \tau/\delta \rceil + \gamma$, this window is contained in $\{0, \ldots, N-1\}$, and corresponds to the checkpoint times in $[t, t+\gamma\delta]$. Since all $q_i \leq 1$, we have $\prod_{r=k}^{k+\gamma} q_r \geq \eta$. Therefore, there exists $i^\star \in I_t$ such that, conditioned on passing checkpoints $0, \ldots, i^\star - 1$, the probability of passing checkpoints $i^\star$ and $i^\star + 1$ is at least $\eta^{2/\gamma}$.

The reduction to MultiStageIndependentMonogamy is the same reduction as in the proof of Theorem 8.4, with the following trajectory-specific changes. First, the two localization challenge points are now $(\widetilde{L}(i^\star\delta), i^\star\delta)$ and $(\widetilde{L}((i^\star+1)\delta), (i^\star+1)\delta)$ rather than two points on a stationary worldline. Thus the middle/left/right components are obtained by cutting the prover's spacetime computation along the corresponding light-cone boundaries. This moving-prover setting for the spacetime decomposition is pictured in Figure 11. Concretely, in the seven-component adversary $\mathcal{A}$ from Theorem 8.4, replace t_0, t_1 by $i^\star\delta, (i^\star+1)\delta$, replace the first target location L by $\widetilde{L}(i^\star\delta)$, and replace the second target location L by $\widetilde{L}((i^\star+1)\delta)$ in the splitting of $\mathbf{M}_L, \mathbf{M}_R, \mathbf{L}', \mathbf{R}'$ and the final response lines.

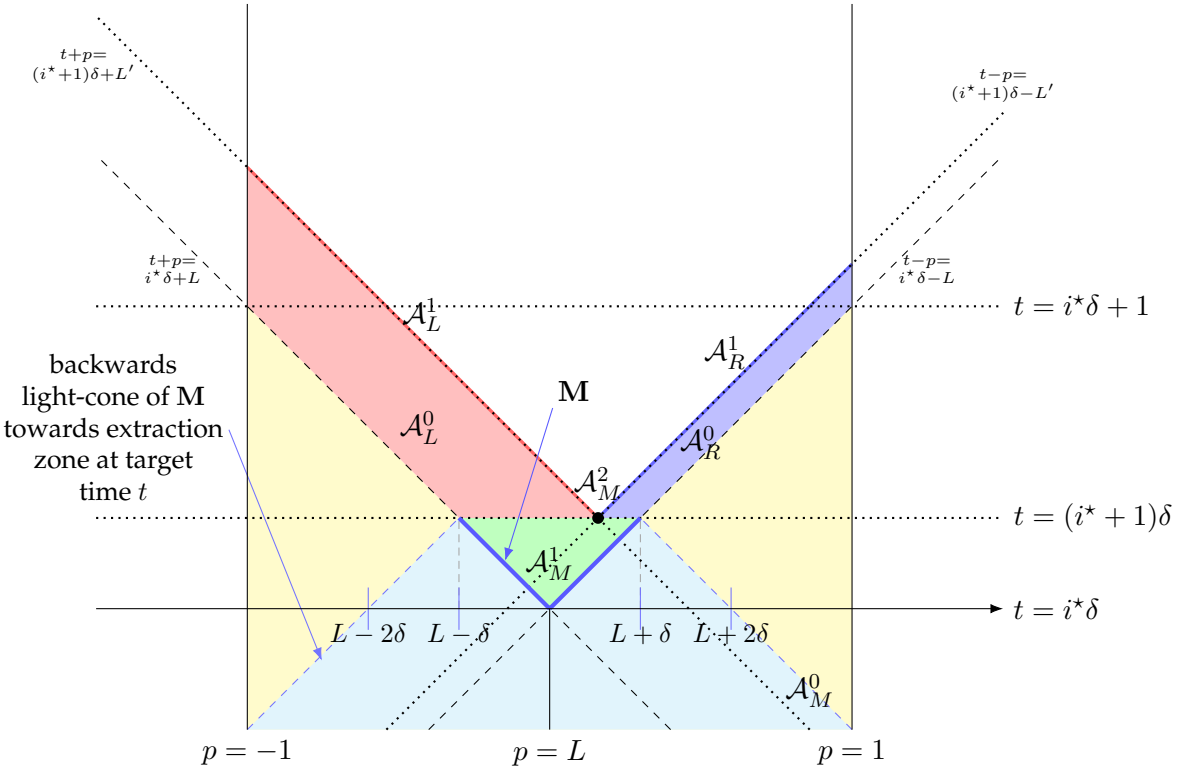

Figure 11: Spacetime decomposition of the reduction $\mathcal{A}$ for trajectory verification, where the prover is moving between checkpoints. In this picture, $L = 0$ and $L' = 0.16$.

As usual, the reduction simulates $\mathcal{O}_{\mathsf{Chal}}$ and the hidden random oracle $\mathcal{O}_{\mathsf{random}}$ by a $2r(\lambda)$-wise independent function F_k, where r is the number of oracle queries $\mathcal{P}^*$ makes. Importantly, due to the repetition, the first-stage reduction component must simulate the Localize experiment from the beginning of time up through checkpoint $i^\star - 1$, thus ensuring that it is non-aborting iff the simulated prover $\mathcal{P}^*$ makes it to checkpoint $i^\star$. To do this, it samples challenges $(x_{L,i}, x_{R,i})$ for $i < i^*$ on its own and simulates the verifiers' interactions with the prover as in Localize. If the prover responds with $y_{L,i}, y_{R,i}$ such that $y_{L,i} \neq y_{R,i}$ or if $y_{L,i} \neq F_k(\theta_i, x_{L,i} \oplus x_{R,i})$, the reduction aborts. The oracle $\mathcal{O}$ is reprogrammed only at the two challenge points $(\theta_{i^\star}, x_{L,i^\star} \oplus x_{R,i^\star})$ and $(\theta_{i^\star+1}, x_{L,i^\star+1} \oplus x_{R,i^\star+1})$. The basis oracle $\mathcal{O}_{\mathsf{Chal}}$ is also reprogrammed at $x_{L,i^\star} \oplus x_{R,i^\star}$ and $x_{L,i^\star+1} \oplus x_{R,i^\star+1}$. The position-verification reprogramming lemma (Lemma 3.1) applies exactly as in the localization proof because the two additive shares of the challenge become jointly available only at the claimed spacetime point for each checkpoint.

Hence, the constructed monogamy adversary does not abort with probability at least $\prod_{r<i^\star} q_r \geq \eta$, and conditioned on not aborting it wins with probability at least $\eta^{2/\gamma} - \mathsf{negl}(\lambda)$. Applying the extractor from Theorem 7.9, exactly as in Theorem 8.4, there exists an extractor which with probability at least 0.99η outputs a state having the following fidelity with $|\mathsf{EPR}\rangle^{\otimes n}$:

$$1 - O\left(\left(1 - \eta^{2/\gamma}\right)^{1/4}\right) - \mathsf{negl}(\lambda) = 1 - O\left(\left(\frac{\log(1/\eta)}{\gamma}\right)^{1/4}\right) - \mathsf{negl}(\lambda).$$

It remains only to translate the checkpoint extractor into the extraction statement at the target time. The input to the monogamy extractor is the middle register on the two lower boundaries of the green triangle in Figure 11; this register can be recovered by forward simulation from $\mathsf{register}[\widetilde{L}(i^\star\delta)_{2\delta} \mathbin{@} i^\star\delta](\mathsf{Exp})$, where $\mathsf{Exp} = \mathsf{Localize}(\mathcal{P}^*(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp}))(\mathsf{pp}, L)$. By construction, $0 \leq i^\star\delta - t \leq \gamma\delta$. Since the valid trajectory has speed at most one, the backwards light cone of $\mathsf{register}[\widetilde{L}(i^\star\delta)_{2\delta} \mathbin{@} i^\star\delta]$ at time t is contained in $\mathsf{register}[L(t)_{(2\gamma+2)\delta} \mathbin{@} t] = \mathsf{register}[L(t)_\Delta \mathbin{@} t]$. Thus the final extractor acts on $\mathsf{register}[L(t)_\Delta \mathbin{@} t]$, forward-simulates to the appropriate future checkpoint neighborhood, and then runs the monogamy extractor. This gives (β, β', Δ)-soundness with

$$\beta(\lambda, \eta) = 1 - O\left(\left(\frac{\log(1/\eta)}{\gamma}\right)^{1/4}\right) - \mathsf{negl}(\lambda), \qquad \beta'(\lambda, \eta) = 0.99\eta(\lambda).$$

□

9.3 State Localization

Note that in the entanglement localization scheme, if we ignore register $\mathbf{A}$, then we are localizing the reduced state on register $\mathbf{B}$. This idea extends to the concept of state localization.

Definition 9.7 (State Localization)**.** *A state localization scheme is parameterized by the security parameter λ and a state family $\{|\psi_i\rangle\}_{i\in[N]}$.*[16] *It consists of the following syntax.*

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, i, \mathbf{B}$*: The Setup algorithm takes the security parameter 1^λ and outputs the public parameter* pp*, the secret parameter* sp*, the extraction key* ek*, an index i, and a state on register $\mathbf{B}$.*
- $\mathsf{Localize}\,(\mathcal{P}(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp}))\,(\mathsf{pp}, L, t) \to \mathbf{B}', b$*: This is a positional protocol between a QPT prover holding a quantum register $\mathbf{B}$ and a PPT verifier with input* sp*, further specified by the following*

[16] Technically, we have one state family per security parameter λ – see Footnote 12

public inputs: public parameters pp*, spatial location* $L \in [-1,1]$*, and time* t*. The prover's output is* ρ' *and the verifier's output is an acceptance indicator* $b \in \{\top, \bot\}$*.*

It should satisfy the following properties — parameterized by completeness probability $\alpha(\lambda)$*, extraction fidelity* $\beta(\lambda,\eta)$*, extraction probability* $\beta'(\lambda,\eta)$*, uniqueness error* $\zeta(\lambda)$*, and localization parameter* $\Delta(\lambda)$ *— for all* $\lambda \in \mathbb{N}$*.*

- α***-Completeness****: For any location* L *and time* t*, there is a QPT prover* $\mathcal{P}$ *at* (L,t) *such that*

$$\Pr\left[b = \top : \begin{array}{r} \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, i, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ (\mathbf{B}', b) \leftarrow \mathsf{Localize}\big(\mathcal{P}(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp})\big)(\mathsf{pp}, L, t) \end{array}\right] \geq \alpha(\lambda).$$

- (β, β', Δ)***-Extraction Soundness****: For any QPT prover* $\mathcal{P}^*$*, spatial location* L*, and time* t*, let*

$$\eta(\lambda) := \Pr\left[b = \top : \begin{array}{r} \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, i, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ (\mathbf{B}', b) \leftarrow \mathsf{Localize}\big(\mathcal{P}^*(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp})\big)(\mathsf{pp}, L, t) \end{array}\right]$$

be $\mathcal{P}^*$*'s success probability. Assume* $\eta(\lambda) \geq 1/p(\lambda)$ *for some polynomial* p*. Then there exists a (potentially unbounded, see Remark 8.2) local extractor*[17] $\mathcal{E}$ *acting on registers near location* L *at time* t *such that the probability of extracting* $|\psi_i\rangle$ *with fidelity at least* $\beta(\lambda,\eta)$ *is at least* $\beta'(\lambda,\eta)$*:*

$$\Pr\left[\langle\psi_i|\,\rho_{\mathbf{B}^*}\,|\psi_i\rangle \geq \beta(\lambda,\eta) : \begin{array}{r} \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, i, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathbf{B}^* \leftarrow \mathsf{register}[L_\Delta \mathbin{@} t]\left(\mathsf{Localize}\left(\mathcal{P}^*(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp})\right)(\mathsf{pp}, L, t)\right) \\ \mathbf{B}^* \leftarrow \mathcal{E}(\mathsf{pp}, \mathsf{ek}, \mathbf{B}^*) \end{array}\right] \geq \beta'(\lambda,\eta)\,,$$

where $\rho_{\mathbf{B}^*}$ *is the state on* $\mathbf{B}^*$ *after running the extractor, and* $L_\Delta := [L - \Delta(\lambda), L + \Delta(\lambda)]$*.*

- ζ***-Uniqueness****: For any (potentially unbounded) cloner* $\mathcal{A}$*,*

$$\mathbb{E}\left[\langle\psi_i|\,\langle\psi_i|\,\rho_{\mathbf{B}_0^*\mathbf{B}_1^*}\,|\psi_i\rangle\,|\psi_i\rangle : \begin{array}{r} \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, i, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathbf{B}_0^*, \mathbf{B}_1^* \leftarrow \mathcal{A}\left(\mathsf{pp}, \mathsf{sp}, \mathsf{ek}, \mathbf{B}\right) \end{array}\right] \leq \zeta(\lambda)\,,$$

where $\rho_{\mathbf{B}_0^*\mathbf{B}_1^*}$ *is the state on* $\mathbf{B}_0^*\mathbf{B}_1^*$ *after running* $\mathcal{A}$*.*

We say that the state localization scheme is non-destructive *if it additionally satisfies the following property.*

- ***Non-destructive****: For any spatial location* L *and time* t*,*

$$\mathbb{E}\left[\mathrm{TD}\left(\rho_{\mathbf{B}}, \rho_{\mathbf{B}'}\right) : \begin{array}{r} \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, i, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathbf{B}', b \leftarrow \mathsf{Localize}\left(\mathcal{P}(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp})\right)(\mathsf{pp}, L, t) \end{array}\right] \leq \mathsf{negl}(\lambda)\,,$$

where $\rho_{\mathbf{B}}$ *is the state on* $\mathbf{B}$ *after* Setup *and* $\rho_{\mathbf{B}'}$ *is the state on* $\mathbf{B}'$ *later after* Localize*.*

Construction 9.8 (Non-Destructive State Localization Scheme)**.** *Let* $\lambda \in \mathbb{N}$ *be the security parameter and* $n(\lambda), m(\lambda) \geq \lambda$ *be polynomials. Take some localization parameter* $\Delta(\lambda) \leq 1$ *and a repetition factor* $\gamma(\lambda)$*. Take a state family* $\{|\psi_i\rangle\}_{i\in[N]}$*. We consider the following non-destructive state localization scheme:*

- *There are two verifiers:* $\mathcal{V}_L$ *at* -1 *and* $\mathcal{V}_R$ *at* 1*.*
- $\mathsf{Setup}(1^\lambda)$*:*
  1. *Generate* $\left(\mathsf{sk}, \left|\Psi^{\mathsf{sk}}\right\rangle_{\mathbf{AB}}\right) \leftarrow \mathsf{GenAnchorState}(1^\lambda)$*, and let* $U_{\mathsf{shift}}, S, T, u, v$ *be the basis-matrix, subspaces, and shifts determined by* sk*.*

[17] It is crucial that this extractor does not know i.

2. *Sample $i \leftarrow [N]$ and define the isometry*

$$\mathsf{Enc}_{U_{\mathsf{shift}},u,v} : |x\rangle \to \frac{1}{\sqrt{2^n}} \sum_{s\in S} (-1)^{s\cdot u} |s + U_{\mathsf{shift}}(0^n \times x \times 0^n) + v\rangle ,$$

and compute

$$\rho = \mathsf{Enc}_{U_{\mathsf{shift}},u,v} |\psi_i\rangle\langle\psi_i| \mathsf{Enc}^\dagger_{U_{\mathsf{shift}},u,v}.$$

3. *Sample the public challenge oracle $\mathcal{O}_{\mathsf{Chal}} : \mathbb{F}_2^\lambda \to \{0,1\}$.*
4. *Sample the public oracle $\mathcal{O} : \{0,1\} \times \mathbb{F}_2^\lambda \times \mathbb{F}_2^{3n} \to \mathbb{F}_2^m \cup \{\bot\}$ based on random oracle $\mathcal{O}_{\mathsf{random}} : \{0,1\} \times \mathbb{F}_2^\lambda \to \mathbb{F}_2^m$ that is **not** publicly available.*

$$\mathcal{O}(\theta,x,z) = \begin{cases} \mathcal{O}_{\mathsf{random}}(\theta,x) & \text{if } \theta = 0 \text{ and } z \in T + v, \\ \mathcal{O}_{\mathsf{random}}(\theta,x) & \text{if } \theta = 1 \text{ and } z \in S^\perp + u, \\ \bot & \text{otherwise} \end{cases}$$

Then Setup *outputs:*

- *the public parameter* $\mathsf{pp} = (\mathcal{O}_{\mathsf{Chal}}, \mathcal{O})$.
- *The secret parameter* $\mathsf{sp} = (u,v)$
- *The extraction key* $\mathsf{ek} = (U_{\mathsf{shift}}, u, v)$.
- *The sampled state and the index* (i,ρ).

- $\mathsf{Localize}\,(\mathcal{P}(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp}))\,(\mathsf{pp}, L, t)$*: Let* $\delta = \Delta/(\gamma+2)$ *and define checkpoints* $t_i = t + i\delta$ *for* $i = 0,1,\ldots,\gamma$.
  - *For* $i = 0,1,\cdots,\gamma$, $\mathcal{V}_L$ *samples* $x_{L,i} \xleftarrow{\$} \mathbb{F}_2^\lambda$ *and broadcasts it at time* $t_i - (L+1)$. *It expects responses* $y_{L,i}$ *at time* $t_i + (L+1)$.
  - *For* $i = 0,1,\cdots,\gamma$, $\mathcal{V}_R$ *samples* $x_{R,i} \xleftarrow{\$} \mathbb{F}_2^\lambda$ *and broadcasts it at time* $t_i - (1-L)$. *It expects responses* $y_{R,i}$ *at time* $t_i + (1-L)$.
  - *If any of these responses is missing, the verifier outputs* $\bot$. *Otherwise, the verifier outputs* $\top$ *if and only if for each* $i = 0,1,\cdots,\gamma$, *one of the following holds:*
    - $\theta_i := \mathcal{O}_{\mathsf{Chal}}(x_{L,i} \oplus x_{R,i}) = 0$ *and* $y_{L,i} = y_{R,i} = \mathcal{O}(0, x_{L,i} \oplus x_{R,i}, v)$.
    - $\theta_i := \mathcal{O}_{\mathsf{Chal}}(x_{L,i} \oplus x_{R,i}) = 1$ *and* $y_{L,i} = y_{R,i} = \mathcal{O}(1, x_{L,i} \oplus x_{R,i}, u)$.
  - *The honest prover consists of a single party always holding the* $\mathbf{B}$ *register at location* L. *For each* $i = 0,1,\ldots,\gamma$, *it receives* $x_{L,i}, x_{R,i}$ *at time* t_i. *If* $\theta_i = 0$, *it coherently evaluates* $\mathcal{O}(0, x_{L,i} \oplus x_{R,i}, \cdot)$ *on the* $\mathbf{B}$ *register in the standard basis and broadcasts the result. If* $\theta_i = 1$, *it coherently evaluates* $\mathcal{O}(1, x_{L,i} \oplus x_{R,i}, \cdot)$ *on the* $\mathbf{B}$ *register in the Hadamard basis and broadcasts the result.*

Theorem 9.9. *Let* $\mathcal{S} = \{|\psi_i\rangle\}_{i\in[N]}$ *be any set of* n*-qubit states that consist of a union of orthonormal bases. That is* $\mathcal{S} = \bigcup_j \mathcal{B}_j$, *where each* $\mathcal{B}_j$ *consists of* 2^n *orthonormal states.*

Then the construction in Construction 9.8 is a non-destructive state localization scheme for $\mathcal{S}$ *in the classical oracle model with* $\alpha(\lambda) = 1$, (β, β', Δ)-***Extraction Soundness*** *for*

$$\beta(\lambda,\eta) = 1 - O\left(\left(\frac{\log(1/\eta)}{\gamma}\right)^{1/4}\right) - \mathsf{negl}(\lambda), \qquad \beta'(\lambda,\eta) = 0.98\eta(\lambda),$$

and uniqueness parameter

$$\zeta(\lambda) := \sup_{\mathcal{A}} \left\{ \mathbb{E}\left[\langle\psi_i| \langle\psi_i| \rho |\psi_i\rangle |\psi_i\rangle : \begin{array}{c} |\psi_i\rangle \leftarrow \mathcal{S} \\ \rho \leftarrow \mathcal{A}(|\psi_i\rangle) \end{array} \right] \right\}.$$

In particular, since we only require extraction soundness against adversaries for which $1/\eta$ is at most polynomial in λ, for any polynomial $t(\lambda)$, we can set $\gamma = c\lambda t^4$ for some constant c to achieve

$$\beta(\lambda, \eta) \geq 1 - 1/t(\lambda) \,.$$

Before coming to the proof, we give two examples of state families that are covered by our theorem. Note that we can write any $\mathcal{S}$ from the above theorem statement as

$$\mathcal{S} = \bigcup_{U \in \mathcal{U}} \{U |x\rangle\}_{x \in \{0,1\}^n}$$

for some set of unitaries $\mathcal{U}$.

- **BB84 states**: $\mathcal{U}$ is the set of n-qubit unitaries $H^{\theta_1} \otimes \cdots \otimes H^{\theta_n}$ for each $\theta \in \{0,1\}^n$.
- **Coset states**: $\mathcal{U}$ is the set of unitaries that first map $|x\rangle \to |Bx\rangle$ for some full rank $B \in \mathbb{F}_2^{n \times n}$ and then apply H to the final $n/2$ qubits.

Proof. 1-completeness and non-destructiveness follow by observation, and ζ-uniqueness follows directly from the definition of ζ given in the theorem statement.

Thus, it remains to argue extraction soundness. To do so, we use exactly the same extractor $\mathcal{E}$ as in the proof of Theorem 9.3. Now, note that our extraction experiment is equivalent to the following purified version.

- In Setup, rather sampling $i \leftarrow [N]$ and applying $\mathsf{Enc}_{U_{\mathsf{shift}},u,v}$ to $|\psi_i\rangle$, instead prepare

$$|\mathsf{EPR}\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle_{\mathbf{A}} |x\rangle_{\mathbf{B}} \,,$$

  and let $\rho_{\mathbf{B}}$ be the result of applying $\mathsf{Enc}_{U_{\mathsf{shift}},u,v}$ to register $\mathbf{B}$.
- After $\mathcal{E}$ outputs $\rho_{\mathbf{B}^*}$, sample $U \leftarrow \mathcal{U}$ (where $\mathcal{U}$ is defined based on $\mathcal{S}$ as above), apply U^t to register $\mathbf{A}$, and then measure $\mathbf{A}$ in the standard basis to obtain x. Let $|\psi_i\rangle := |\psi_{U,x}\rangle = U|x\rangle$ be the state[18] that is used in the expression $\langle\psi_i| \rho_{\mathbf{B}^*} |\psi_i\rangle$.

For any choice of U, the expected success probability of extracting the state is

$$\begin{aligned}
\mathop{\mathbb{E}}_x \left[\langle\psi_{U,x}| \rho_{\mathbf{B}^*} |\psi_{U,x}\rangle\right] &= \mathrm{Tr}\left[\left(\sum_x |x\rangle\langle x|_{\mathbf{A}} \otimes |\psi_{U,x}\rangle\langle\psi_{U,x}|_{\mathbf{B}^*}\right) U_{\mathbf{A}}^t \rho_{\mathbf{AB}^*} U_{\mathbf{A}}^*\right] \\
&= \mathrm{Tr}\left[\left(\sum_x U^* |x\rangle\langle x| U_{\mathbf{A}}^t \otimes U |x\rangle\langle x| U_{\mathbf{B}^*}^\dagger\right) \rho_{\mathbf{AB}^*}\right] \\
&= \mathrm{Tr}\left[\left((U^* \otimes U)\left(\sum_x |x\rangle\langle x| \otimes |x\rangle\langle x|\right)\left(U^t \otimes U^\dagger\right)\right) \rho_{\mathbf{AB}^*}\right] \\
&\geq \mathrm{Tr}\left[\left((U^* \otimes U)\left(|\mathsf{EPR}\rangle\langle\mathsf{EPR}|^{\otimes n}\right)\left(U^t \otimes U^\dagger\right)\right) \rho_{\mathbf{AB}^*}\right] \\
&= \mathrm{Tr}\left[\left(|\mathsf{EPR}\rangle\langle\mathsf{EPR}|^{\otimes n}\right) \rho_{\mathbf{AB}^*}\right] \\
&\geq \beta(\lambda, \eta),
\end{aligned}$$

[18] In this purified version, the family of states is indexed by U, x. There is a one-to-one correspondence between U, x and, i which we used in the original definition.

where the last inequality holds with probability at least $0.99\eta(\lambda)$ by Theorem 9.3. By an additional Markov inequality on expectation over x we get $\beta'(\lambda,\eta) = 0.98\eta(\lambda)$. This gives the claimed extraction soundness parameters. □

Remark 9.10. *One could strengthen the completeness definition of State Localization to demand that $\rho = |\psi_i\rangle\langle\psi_i|$. That is, the prover obtains the state $|\psi_i\rangle$ itself rather than an encoded version of it.*

We can achieve this stronger completeness guarantee for the set of $3n$-qubit coset states with subspaces of dimension $3n/2$. Indeed, one can sample a uniformly random $3n$-qubit coset state by first sampling an n-qubit coset state $|\psi\rangle$ with subspace of dimension $n/2$, and then outputting the $3n$-qubit state $\mathsf{Enc}_{U_{\mathsf{shift}},u,v}|\psi\rangle$, where U_{shift}, u, v are sampled by the Setup algorithm given above. The resulting $3n$-qubit state is still unclonable even given U_{shift}, u, v, and thus uniqueness continues to hold.

9.4 Functionality Localization

We first recall the notion of quantum copy-protection from [Aar09]. Our definition of functionality localization, stated later, will be a strengthening of copy-protection.

Definition 9.11 (Copy-protectable functionality)**.** *A family of functions $\mathcal{F} = \{f\}_f$ is copy-protectable with security $\zeta = \zeta(\lambda)$ if there exists a distribution $\mathcal{D}$ on its domain and algorithms*

- $\mathsf{Protect}(1^\lambda, f) \to \rho_f$
- $\mathsf{Eval}(\rho_f, x) \to y$

such that the following properties hold.

- ***Correctness**: For any $f \in \mathcal{F}$ and $x \in \mathcal{X}$,*

$$\Pr[\mathsf{Eval}(\rho_f, x) = f(x) : \rho_f \leftarrow \mathsf{Protect}(1^\lambda, f)] = 1 - \mathsf{negl}(\lambda).$$

- ***Security**: For any QPT tri-partite adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$,*

$$\Pr\left[\begin{array}{l} y_B = f(x_B) \wedge y_C = f(x_C) : \\ \end{array} \begin{array}{r} f \leftarrow \mathcal{F} \\ \rho_f \leftarrow \mathsf{Protect}(1^\lambda, f) \\ \rho_{B,C} \leftarrow \mathcal{A}(\rho_f) \\ x_B, x_C \leftarrow \mathcal{D} \\ y_B \leftarrow \mathcal{B}(x_B, B), y_C \leftarrow \mathcal{C}(x_C, C) \end{array}\right] \leq \zeta(\lambda).$$

We also recall quantum state obfuscation, which will be used later on for our construction of functionality localization. The following definition is adapted from [BBV24], but for simplicity we have significantly weakened the definition to work only for classical functionalities.

Definition 9.12 (Quantum State Obfuscation ([CG24],[BBV24]))**.** *Let $\mathcal{F} = \{f\}_f$ be a classical function family, where for each $f \in \mathcal{F}$ there exists a quantum program $(C_f, |\psi_f\rangle)$ such that $C_f(x, |\psi\rangle_f) = f(x)$ deterministically. A quantum state obfuscator for $\mathcal{F}$ is a pair of QPT algorithms $(\mathsf{QObf}, \mathsf{QEval})$ with the following syntax.*

- $\mathsf{QObf}\left(1^\lambda, |\psi\rangle, C\right) \to \left|\widetilde{\psi}\right\rangle$*: The obfuscator takes as input the security parameter 1^λ and a quantum program $(C, |\psi\rangle)$, and outputs an obfuscated state $\left|\widetilde{\psi}\right\rangle$.*
- $\mathsf{QEval}\left(x, \left|\widetilde{\psi}\right\rangle\right) \to y$*: The evaluation algorithm takes an input $x \in \{0,1\}^{m(\lambda)}$ and an obfuscated state $\left|\widetilde{\psi}\right\rangle$, and outputs $y \in \{0,1\}^{m'(\lambda)}$.*

Correctness *is defined as follows for any $f \in \mathcal{F}$.*

$$\forall x \in \{0,1\}^{m(\lambda)}, \Pr\left[\mathsf{QEval}\left(x, \left|\widetilde{\psi}\right\rangle\right) = C_f(x, |\psi_f\rangle) : \left|\widetilde{\psi}\right\rangle \leftarrow \mathsf{QObf}\left(1^\lambda, |\psi_f\rangle, C_f\right)\right] = 1 - \mathsf{negl}(\lambda).$$

Ideal Obfuscation*: For any QPT adversary $\mathcal{A}$, there exists a QPT simulator* Sim *such that for any $f \in \mathcal{F}$ and QPT distinguisher $\mathcal{D}$,*

$$\left|\Pr\left[1 \leftarrow \mathcal{D}\left(\mathcal{A}\left(\mathsf{QObf}\left(1^\lambda, |\psi_f\rangle, C_f\right)\right)\right)\right] - \Pr\left[1 \leftarrow \mathcal{D}\left(\mathsf{Sim}^f\left(1^\lambda\right)\right)\right]\right| = \mathsf{negl}(\lambda).$$

Theorem 9.13 (Quantum State Ideal Obfuscation in the Oracle Model, [BBV24])**.** *There exists a quantum state ideal obfuscator for any classical functionality, in the classical oracle model.*

Now we state the main definition of this section, and present our construction of it from quantum state obfuscation and signatures.

Definition 9.14 (Functionality localization)**.** *A functionality localization scheme is parameterized by a function family $\mathcal{F} = \{f : \mathcal{X} \to \mathcal{Y}\}_f$[19], and a distribution $\mathcal{D}$ on its domain. It consists of the following syntax.*

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, f, \mathbf{B}$*: The setup algorithm takes the security parameter 1^λ and outputs the public parameter* pp*, the secret parameter* sp*, the extraction key* ek*, a function $f \in \mathcal{F}$, and a state on register $\mathbf{B}$.*
- $\mathsf{Localize}\big(\mathcal{P}(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp}, f)\big)(\mathsf{pp}, L, t) \to \mathbf{B}', b$*: The Localize protocol is an interaction between a QPT prover whose input is register $\mathbf{B}$, and a PPT verifier with inputs* sp *and f, further specified by the following public inputs: public parameters* pp*, spatial location $L \in [-1, 1]$, and time t. The prover's output is $\mathbf{B}'$ and the verifier's output is an acceptance indicator $b \in \{\top, \bot\}$.*

It should satisfy the following properties — parameterized by completeness probability $\alpha(\lambda)$, extraction success $\beta(\lambda, \eta)$, uniqueness error $\zeta(\lambda)$, and localization parameter $\Delta(\lambda)$ — for all $\lambda \in \mathbb{N}$.

- ***Functionality Preservation****: There exists a procedure* Eval *such that for any $x \in \mathcal{X}$,*

$$\Pr\left[\mathsf{Eval}(\mathbf{B}, x) = f(x) : \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, f, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda)\right] = 1 - \mathsf{negl}(\lambda)$$

- α***-Completeness****: For any location L and time t, there is a QPT prover $\mathcal{P}$ at (L, t) such that*

$$\Pr\left[b = \top : \begin{array}{r}\mathsf{pp}, \mathsf{sp}, \mathsf{ek}, f, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ (\mathbf{B}', b) \leftarrow \mathsf{Localize}\big(\mathcal{P}(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp}, f)\big)(\mathsf{pp}, L, t)\end{array}\right] \geq \alpha(\lambda).$$

- (β, Δ)***-Extraction Soundness****: For any QPT prover $\mathcal{P}^*$, spatial location L, and time t, let*

$$\eta(\lambda) := \Pr\left[b = \top : \begin{array}{r}\mathsf{pp}, \mathsf{sp}, \mathsf{ek}, f, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ (\mathbf{B}', b) \leftarrow \mathsf{Localize}\big(\mathcal{P}^*(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp}, f)\big)(\mathsf{pp}, L, t)\end{array}\right]$$

be $\mathcal{P}^*$'s success probability. Assume $\eta(\lambda) \geq 1/p(\lambda)$ for some polynomial p, and let spatial region $L_\Delta := [L - \Delta(\lambda), L + \Delta(\lambda)]$. Then there exists a QPT extractor $\mathcal{E}$ acting on registers near location

[19] Technically, we have one function family per security parameter λ – see Footnote 12

L at time t such that

$$\Pr\left[y = f(x) : \begin{array}{r} \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, f, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathbf{B}^* \leftarrow \mathsf{register}[L_\Delta \mathbin{@} t]\,(\mathsf{Localize}(\mathcal{P}^*(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp}, f))\,(\mathsf{pp}, L, t)) \\ x \leftarrow \mathcal{D} \\ y \leftarrow \mathcal{E}(\mathsf{pp}, \mathsf{ek}, x, \mathbf{B}^*) \end{array}\right] \geq \beta(\lambda, \eta).$$

- *ζ-**Uniqueness**: For any QPT cloners $(\mathcal{A}, \mathcal{B}, \mathcal{C})$,*

$$\Pr\left[\begin{array}{l} y_0 = f(x_0), \\ y_1 = f(x_1) \end{array} : \begin{array}{r} \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, f, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathbf{B}_0^*, \mathbf{B}_1^* \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{sp}, \mathsf{ek}, \mathbf{B}) \\ x_0 \leftarrow \mathcal{D}, x_1 \leftarrow \mathcal{D} \\ y_0 \leftarrow \mathcal{B}(x_0, \mathbf{B}_0^*), y_1 \leftarrow \mathcal{C}(x_1, \mathbf{B}_1^*) \end{array}\right] \leq \zeta(\lambda).$$

We say that the functionality localization scheme is non-destructive *if it additionally satisfies the following property.*

- ***Non-destructive**: For any spatial location L and time t,*

$$\mathbb{E}\left[\mathrm{TD}\left(\rho_{\mathbf{B}}, \rho_{\mathbf{B}'}\right) : \begin{array}{r} \mathsf{pp}, \mathsf{sp}, \mathsf{ek}, f, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda) \\ (\mathbf{B}', b) \leftarrow \mathsf{Localize}\big(\mathcal{P}(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp}, f)\big)(\mathsf{pp}, L, t) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

where $\rho_{\mathbf{B}}$ is the state on $\mathbf{B}$ after Setup *and $\rho_{\mathbf{B}'}$ is the state on $\mathbf{B}'$ later after* Localize*.*

Remark 9.15. *Note that in contrast to entanglement and state localization, here we require the extractor $\mathcal{E}$ to be efficient (QPT). This is because the uniqueness property may only hold against computationally-bounded cloners.*

Construction 9.16 (Non-Destructive Functionality Localization Scheme)**.** *Let $\mathcal{F}$ be a function family. Let $n(\lambda), m(\lambda) \geq \lambda$ be polynomials. Take a localization parameter $\Delta(\lambda) \leq 1$ and a repetition factor $\gamma(\lambda)$. Let* QSO *be a quantum state obfuscator, and let* $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Ver})$ *be a digital signature scheme, with message space $\{0,1\}^*$ and signature space $\mathcal{S}$. We define the following functionality localization scheme.*

- $\mathsf{Setup}(1^\lambda)$*:*
  1. *Sample $f \leftarrow \mathcal{F}$.*
  2. *Sample a uniformly random subspace $S \leq \mathbb{F}_2^n$ of dimension $n/2$, sample $v \xleftarrow{\$} \mathsf{CS}(S)$ and $u \xleftarrow{\$} \mathsf{CS}(S^\perp)$, and prepare the coset state $X^v Z^u |S\rangle$.*
  3. *Sample $\mathsf{sk}, \mathsf{vk} \leftarrow \mathsf{KeyGen}(1^\lambda)$.*
  4. *Define the signed functionality*

$$\widehat{f}_{\mathsf{sk}}(x) := \big(f(x), \mathsf{Sign}(\mathsf{sk}, (x, f(x)))\big), \tag{2}$$

  and obfuscate it using the ideal obfuscator:

$$\rho_{\mathsf{QSO}} \leftarrow \mathsf{QSO.QObf}(1^\lambda, \widehat{f}_{\mathsf{sk}}).$$

  5. *Sample a public random oracle $\mathcal{O}_{\mathsf{chal}} : \mathbb{F}_2^m \to \mathbb{F}_2^m$.*
  6. *Let $C_0 := S + v$ and $C_1 := S^\perp + u$. Sample a random oracle $\mathcal{O}_{\mathsf{random}} : \mathbb{F}_2^m \times \mathcal{X}_\lambda \times \mathcal{Y}_\lambda \times \mathbb{F}_2 \to \mathbb{F}_2^m$ and define the public oracle $\mathcal{O}_{S,T,u,v,\mathsf{vk}} : \mathbb{F}_2^m \times \mathcal{X} \times \mathcal{Y} \times \mathcal{S} \times \{0,1\} \times \mathbb{F}_2^n \to \mathbb{F}_2^m \cup \{\bot\}$ as follows.*

$$\mathcal{O}_{S,T,u,v,\mathsf{vk}}(h, x, y, \sigma, c, z) = \begin{cases} \mathcal{O}_{\mathsf{random}}(h, x, y, c) & \text{if } \mathsf{Ver}(\mathsf{vk}, (x, y), \sigma) = \top \text{ and } z \in C_c, \\ \bot & \text{otherwise} \end{cases}$$

*Note that $\mathcal{O}_{\mathsf{random}}$ is **not** made public.*

Then Setup *outputs:*

- *The public parameter* $\mathsf{pp} = (\mathcal{O}, \mathcal{O}_{\mathsf{chal}}, \mathsf{vk})$.
- *The secret parameter* $\mathsf{sp} = (u, v, \mathsf{sk})$.
- *The extraction key* $\mathsf{ek} = \emptyset$.
- *The functionality* f.
- *The quantum state* $\rho = (\rho_{\mathsf{QSO}}, X^v Z^u |S\rangle)$.

- Localize $(\mathcal{P}(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp}, f))\,(\mathsf{pp}, L, t)$*: Let* $\delta = \Delta/(2\gamma+2)$ *and define checkpoint times* $t_{2i} = t + 2i\delta$ *and* $t_{2i+1} = t + (2i+1)\delta$ *for* $i = 0, 1, \ldots, \gamma - 1$.

  1. *For each* $i \in \{0, \ldots, \gamma - 1\}$*, the verifiers sample* $x_i \leftarrow \mathcal{D}$*, then sample the challenges*

  $$\hat{x}_{L,i}, \hat{x}_{R,i}, h_{L,2i}, h_{R,2i}, h_{L,2i+1}, h_{R,2i+1} \leftarrow \{0,1\}^m,$$

  for rounds $2i$ *and* $2i+1$*. Let* $\bar{x}_i := x_i \oplus \mathcal{O}_{\mathsf{chal}}(\hat{x}_{L,i} \oplus \hat{x}_{R,i})$*.* $\mathcal{V}_L$ *and* $\mathcal{V}_R$ *send* $(\bar{x}_i, \hat{x}_{L,i}, h_{L,2i})$ *and* $(\hat{x}_{R,i}, h_{R,2i})$*, respectively, so that they arrive at location* L *at time* t_{2i}*. They send* $h_{L,2i+1}$ *and* $h_{R,2i+1}$ *so that they arrive at* L *at time* t_{2i+1}*.*
  2. *For each* i*, upon receiving the messages for checkpoint* $2i$ *at location* L*, the honest prover constructs the real challenges* x_i *and* h_{2i} *as follows:*

  $$x_i := \bar{x}_i \oplus \mathcal{O}_{\mathsf{chal}}(\hat{x}_{L,i} \oplus \hat{x}_{R,i})$$
  $$h_{2i} := h_{L,2i} \oplus h_{R,2i}$$

  It then evaluates

  $$(y_i, \sigma_i) \leftarrow \mathsf{QSO.QEval}(x_i, \rho_{\mathsf{QSO}}),$$

  and coherently queries $\mathcal{O}(h_{2i}, x_i, y_i, \sigma_i, 0, \cdot)$ *on the coset state in the standard basis, obtaining an answer* a_{2i}*, which it immediately broadcasts.*
  3. *For each* i*, upon receiving the messages for checkpoint* $2i+1$ *at location* L*, the honest prover reconstructs* $h_{2i+1} := h_{L,2i+1} \oplus h_{R,2i+1}$ *and coherently queries* $\mathcal{O}(h_{2i+1}, x_i, y_i, \sigma_i, 1, \cdot)$ *on the same coset state in the Hadamard basis, obtaining an answer* a_{2i+1}*, which it immediately broadcasts.*
  4. *If any response is missing, the verifiers output* $\perp$*. Otherwise, for each* i*, the verifiers set* $h_{2i} := h_{L,2i} \oplus h_{R,2i}$ *and* $h_{2i+1} := h_{L,2i+1} \oplus h_{R,2i+1}$*, compute*

  $$\sigma_{x_i} := \mathsf{Sign}(\mathsf{sk}, (x_i, f(x_i)))$$

  and accept iff for every $i \in \{0, \ldots, \gamma - 1\}$*,*

  $$a_{2i} = \mathcal{O}(h_{2i}, x_i, f(x_i), \sigma_{x_i}, 0, v) \qquad \text{and} \qquad a_{2i+1} = \mathcal{O}(h_{2i+1}, x_i, f(x_i), \sigma_{x_i}, 1, u).$$

Remark 9.17. *Note that unlike the constructions of trajectory verification, entanglement localization, and state localization, Construction 9.16 does not randomize the basis of each challenge. Instead it uses a fixed ordering — standard, then Hadamard. This simplification is possible because here we are no longer relying on the EPR extraction machinery of [VZ21], and instead all we need is to show that the prover must have queried at the designated position. This is already a property we establish for the standard-then-Hadamard monogamy game in Definition 6.1, and by reducing to this game instead of* MultiStageIndependentMonogamy *we get a quadratically better loss in* $\beta(\lambda, \eta)$ *than for the other primitives.*

Lemma 9.18 (Functionality Preservation)**.** *Construction 9.16 satisfies* ***Functionality Preservation****.*

Proof. We define $\mathsf{Eval}(\mathbf{B}, x)$ as follows. Take the state on $\mathbf{B}$, $\rho_{\mathbf{B}} = (\rho_{\mathsf{QSO}}, X^v Z^u |S\rangle)$, run $(y, \sigma) \leftarrow \mathsf{QSO.QEval}(\rho_{\mathsf{QSO}}, x)$, and output y. The property follows immediately from correctness of QSO. □

Lemma 9.19 (Completeness, non-destructiveness)**.** *Construction 9.16 satisfies* $(1-\mathsf{negl}(\lambda))$*-**Completeness**. Furthermore, it is non-destructive.*

Proof. By correctness of QSO, in every checkpoint pair the honest prover obtains $(f(x_i), \sigma_{x_i})$ except with negligible probability, and the state ρ_{QSO} is preserved by gentle measurement. For each i, on checkpoint $2i$, $\mathcal{O}(h_{2i}, x_i, f(x_i), \sigma_{x_i}, 0, \cdot)$ is constant on the coset $S + v$, so coherently querying it on $X^v Z^u |S\rangle$ in the standard basis returns $\mathcal{O}(h_{2i}, x_i, f(x_i), \sigma_{x_i}, 0, v)$ while leaving the coset state unchanged. Similarly, $\mathcal{O}(h_{2i+1}, x_i, f(x_i), \sigma_{x_i}, 1, \cdot)$ is constant on the coset $S^\perp + u$, so coherently querying it in the Hadamard basis returns $\mathcal{O}(h_{2i+1}, x_i, f(x_i), \sigma_{x_i}, 1, u)$ while again leaving the coset state unchanged. A union bound over the polynomially many checkpoints gives acceptance probability $1 - \mathsf{negl}(\lambda)$ and final prover state negligibly close to the initial state. □

Lemma 9.20 (Extraction Soundness)**.** *If* $(\mathsf{Sign}, \mathsf{Ver})$ *is a secure MAC scheme and* QSO *is an ideal obfuscator, then Construction 9.16 satisfies* (β, Δ)*-**Extraction soundness**, where*

$$\beta(\lambda, \eta) = \eta(\lambda) \left(1 - O\left(\sqrt{\frac{\log(1/\eta)}{\gamma}}\right) - \mathsf{negl}(\lambda)\right).$$

Proof. The proof proceeds in several steps, which we outline below.

Removing the Signature To analyze the construction, we first consider modifying the public oracle to reject any queries not output by the signed functionality. Define

$$\mathcal{O}_f(h, x, y, \sigma, c, z) = \begin{cases} \mathcal{O}(h, x, y, \sigma, c, z) & \text{if } y = f(x), \\ \perp & \text{otherwise.} \end{cases} \tag{3}$$

Claim 9.21. *No QPT distinguisher has more than negligible advantage in telling apart a setup which outputs $\mathcal{O}$, from a setup which outputs $\mathcal{O}_f$.*

Proof. The two worlds differ only on queries (h, x, y, σ, c, z) such that $\mathsf{Ver}(\mathsf{vk}, (x, y), \sigma) = \top$ and $y \neq f(x)$. Any such query contains a valid signature on a message (x, y) that is never signed by the functionality $\widehat{f}_{\mathsf{sk}}$, since the only signatures returned by $\widehat{f}_{\mathsf{sk}}$ are on messages of the form $(x, f(x))$. If there were a distinguisher between the two worlds, then by the security of QSO we could replace the obfuscated state by black-box access to the functionality $\widehat{f}_{\mathsf{sk}}$ without losing more than negligible advantage. A standard BBBV-type search-to-distinguishing reduction would then find, with non-negligible probability, a query on which $\mathcal{O}$ and $\mathcal{O}_f$ differ, yielding a valid forgery for the MAC scheme. □

We may therefore analyze the experiment with $\mathcal{O}_f$ in place of $\mathcal{O}$, losing only a negligible term.

Finding a Successful Checkpoint Pair Write $\delta = \Delta/(2\gamma + 2)$ as in the construction. Fix L, t, and a QPT prover $\mathcal{P}^*$ succeeding in Localize with probability η. Let q_i denote the probability that

$\mathcal{P}^*$ passes the i-th checkpoint pairs condition on passing all previous checkpoints. There exists $i^\star \in \{0, \ldots, \gamma - 1\}$ such that the checkpoint pair $2i^\star, 2i^\star + 1$ is passed with probability

$$q_{i^\star} \geq \eta^{1/\gamma}$$

conditioned on passing checkpoint pairs $0, \ldots, i^\star - 1$. Let time $t^\star = t + 2i^\star\delta$ be the first challenge point of the pair, and denote the failure probability by $\varepsilon := 1 - q_{i^\star}$ for convenience.

Extracting the Functionality. We now define the implementation extractor $\mathcal{E}$. In addition to a sampled input x, the extractor takes as input the public setup parameters pp, and the local register $\mathbf{B}^*$ from which it will extract. For our construction, $\mathsf{ek} = \emptyset$. We will split $\mathcal{E}$ into two parts $\mathcal{E}_0$ and $\mathcal{E}_1$, which also act on different parts of the input, for convenience later on in the proof.

$\mathcal{E}_0(\mathsf{pp}, \mathsf{ek}, \mathbf{B}^*)$: The extractor first forward-simulates $\mathbf{B}^* = \mathsf{register}[L_\Delta \mathbin{@} t]$ to obtain $\mathsf{register}[L_{2\delta} \mathbin{@} t^\star]$, which we can see is possible by checking the backwards light-cone: $t^\star - t \leq 2\gamma\delta = \Delta - 2\delta$. Note that when $\mathcal{E}_0$ is doing this forward-simulation, it is not sampling challenges or answering oracle queries on its own. Instead, it is using all information contained in $\mathbf{B}^*$ (including inflight verifier messages, etc) along with the code of $\mathcal{P}^*$, and access to the oracles in pp, to simulate the prover perfectly.

Next, in the same way, $\mathcal{E}_0$ simulates $\mathcal{P}^*$ inside the spacetime region for checkpoint $2i^\star$, essentially corresponding to the light blue-shaded region of Figure 10). Let $R_\vee$ be the vee-shaped top border of this region (bottom border of the green triangle in Figure 10), and note that its backwards light-cone is $L_{2\delta}$ at time $t^\star$. The output of $\mathcal{E}_0$ is

$$\mathbf{M} := \mathsf{register}[R_\vee](\text{Forward time-evolution of } \mathbf{B}^*)$$

$\mathcal{E}_1(\mathsf{pp}, \mathsf{ek}, \mathbf{M}, x)$: Continuing from $\mathcal{E}_0$, and now having additional input x, the extractor now recovers $\mathcal{V}$'s messages for checkpoints $2i^\star, 2i^\star + 1$,

$$\bar{x}_{i^\star}, \hat{x}_{L,i^\star}, \hat{x}_{R,i^\star}, h_{L,2i^\star}, h_{R,2i^\star}, h_{L,2i^\star+1}, h_{R,2i^\star+1}$$

from the transcript contained in the forward simulation. Let $\hat{x}_{i^\star} := \hat{x}_{L,i^\star} \oplus \hat{x}_{R,i^\star}$, $h_{2i^\star} := h_{L,2i^\star} \oplus h_{R,2i^\star}$, and $h_{2i^\star+1} := h_{L,2i^\star+1} \oplus h_{R,2i^\star+1}$.

It then simulates $\mathcal{P}^*$ inside the spacetime triangle ∇, corresponding to the green triangle in Figure 10. For convenience, we use $\mathcal{P}^*_\nabla$ to refer to the portion of $\mathcal{P}^*$'s circuits contained within the region ∇. During its simulation of $\mathcal{P}^*_\nabla$, $\mathcal{E}$ intercepts all oracle queries and responds in the following way.

- On query $\mathcal{O}_{\mathsf{chal}}(x')$:
  - If $x' = \hat{x}_{i^\star}$, respond with $x \oplus \bar{x}_{i^\star}$;
  - otherwise, forward to the true oracle $\mathcal{O}_{\mathsf{chal}}$.
- On query $\mathcal{O}(h, x', y, \sigma, c, z)$:
  - If $(h, x', c) = (h_{2i^\star}, x, 0)$, first compute $w = \mathcal{O}(h_{2i^\star}, x, y, \sigma, c, z)$.
    - If $w \neq \perp$, respond with $H(y)$, where H is a compressed oracle;
    - otherwise, respond with $\perp$.
  - otherwise, forward to the true oracle $\mathcal{O}$;

At the end of the simulation, $\mathcal{E}$ measures the database of the compressed oracle H. If there is a unique non-$\perp$ entry y, it outputs y; otherwise it outputs $\perp$.

We now analyze $\mathcal{E}$, which proceeds very similarly to the analysis in Theorem 8.4.

Claim 9.22. *In the extraction soundness experiment, $\mathcal{E}$ outputs $y = f(x)$ with probability at least $\eta(1 - 2\sqrt{2\varepsilon}) - \mathsf{negl}(\lambda)$.*

Proof. We will prove this by first constructing a reduction from the Localize experiment to the MultiStageSearchMonogamy game of Definition 6.1, and then showing how to view our extractor $\mathcal{E}$ as a component in that reduction.

The reduction will be a seven-part adversary

$$\mathcal{A} = (\underbrace{\mathcal{A}^0_M}_{\text{stage 0}}, \underbrace{\mathcal{A}^1_M, \mathcal{A}^0_L, \mathcal{A}^0_R}_{\text{stage 1}}, \underbrace{\mathcal{A}^2_M, \mathcal{A}^1_L, \mathcal{A}^1_R}_{\text{stage 2}})$$

for MultiStageSearchMonogamy, where $S = T$ because the parameter n_e is set to 0. The challenger samples (S, u, v) and provides the initial coset-state register $X^v Z^u |S\rangle$ to $\mathcal{A}^0_M$. In stage 1 it gives oracle access to $\mathcal{O}^{b_0}_{S+v}$ for a uniformly random hidden answer $b_0 \in \mathbb{F}_2^m$, and in stage 2 it additionally gives oracle access to $\mathcal{O}^{b_1}_{S^\perp+u}$ for an independent uniformly random hidden answer $b_1 \in \mathbb{F}_2^m$.

The adversary $\mathcal{A}$ is defined just as in Theorem 8.4, excepting a few differences specific to the functionality localization setting which we highlight below. Let $r(\lambda)$ be an upper bound on the number of oracle queries made by $\mathcal{P}^*$.

1. Once $\mathcal{A}^0_M$ receives the coset-state $X^v Z^u |S\rangle$ along with membership oracles $\mathcal{O}_{S+v}$ and $\mathcal{O}_{S^\perp+u}$ from the challenger, it additionally samples the rest of Setup:

   $$f \leftarrow \mathcal{F}, \qquad \mathsf{sk}, \mathsf{vk} \leftarrow \mathsf{KeyGen}(1^\lambda), \qquad \rho_{\mathsf{QSO}} \leftarrow \mathsf{QSO.QObf}(1^\lambda, \widehat{f_{\mathsf{sk}}}).$$

   Letting $\mathbf{B}$ be a register with state $\rho = (\rho_{\mathsf{QSO}}, X^v Z^u |S\rangle)$, the simulated prover is initialized with input $\mathbf{B}$. $\mathcal{A}^0_M$ also initializes two $2r(\lambda)$-wise independent functions F_k and $G_{k'}$ (for simulating $\mathcal{O}_{\mathsf{random}}$ and $\mathcal{O}_{\mathsf{chal}}$, respectively), and samples the following challenges for each $i \in \{0, \ldots, i^\star\}$

   $$x_i, \bar{x}_i, \hat{x}_{L,i}, \hat{x}_{R,i}, h_{L,2i}, h_{R,2i}, h_{L,2i+1}, h_{R,2i+1}$$

   where $x_i \leftarrow \mathcal{D}$, and the rest are uniform subject to $x_i = \bar{x}_i \oplus G_{k'}(\hat{x}_{L,i} \oplus \hat{x}_{R,i})$. Let $h_{2i^\star} := h_{L,2i^\star} \oplus h_{R,2i^\star}$ and $h_{2i^\star+1} := h_{L,2i^\star+1} \oplus h_{R,2i^\star+1}$. The first $i^\star$ of these challenge tuples are used, along with the hash functions F_k and $G_{k'}$, to simulate the prover through rounds $0, \ldots, 2i^\star - 1$ of the Localize experiment. If the prover responds incorrectly in any of these rounds, $\mathcal{A}^0_M$ aborts. The challenges for rounds $2i^\star$ and $2i^\star + 1$ are placed in the info register which is copied along between the reduction components. In particular, info contains $x_{i^\star}$, $h_{2i^\star}$, $h_{2i^\star+1}$, and f. Like in Theorem 9.6, access to the oracle $\mathcal{O}$ is simulated using the sampled components of Setup, along with F_k for the random oracle and $\mathcal{O}_{S+v}$ and $\mathcal{O}_{S^\perp+u}$ for checking coset membership. Access to $\mathcal{O}_{\mathsf{chal}}$ is similarly simulated with $G_{k'}$.
2. For the simulation of all parts of $\mathcal{A}$ in stages 1 and 2, we reprogram the oracle as follows: all queries of the form $(h_{2i^\star}, x_{i^\star}, f(x_{i^\star}), \sigma, 0, z)$ are remapped to $\mathcal{O}^{b_0}_{S+v}(z)$. In other words, $\mathcal{O}_{\mathsf{random}}$ is reprogrammed at the lone point $(h_{2i^\star}, x_{i^\star}, f(x_{i^\star}), 0)$.
3. For the simulation of stage 2, we additionally reprogram the oracle as follows: all queries of the form $(h_{2i^\star+1}, x_{i^\star}, f(x_{i^\star}), \sigma, 1, z)$ are remapped to $\mathcal{O}^{b_1}_{S^\perp+u}(z)$. In other words, $\mathcal{O}_{\mathsf{random}}$ is reprogrammed at the lone point $(h_{2i^\star+1}, x_{i^\star}, f(x_{i^\star}), 1)$.

Claim 9.23.

$$\Pr\left(\mathsf{MultiStageSearchMonogamy}(\mathcal{A}, 1^\lambda) \neq \bot\right) \geq \eta, \text{ and}$$
$$\frac{\Pr\left(\mathsf{MultiStageSearchMonogamy}(\mathcal{A}, 1^\lambda) = 1\right)}{\Pr\left(\mathsf{MultiStageSearchMonogamy}(\mathcal{A}, 1^\lambda) \neq \bot\right)} \geq 1 - \varepsilon - \mathsf{negl}(\lambda).$$

Proof. The component $\mathcal{A}_M^0$ aborts exactly if the simulated prover fails one of checkpoint pairs $0, \ldots, i^\star - 1$, giving the first bound. Since the tuple $(h_{2i^\star}, x_{i^\star})$ is not jointly available before space-time point $(L, t^\star)$, and the tuple $(h_{2i^\star+1}, x_{i^\star})$ is not jointly available before $(L, t^\star + \delta)$, the same oracle-reprogramming argument used in Theorem 9.6 (invoking Lemma 3.1) implies the simulation of checkpoints $2i^\star, 2i^\star + 1$ is negligibly close to the real-world execution of adversary $\mathcal{P}^*$ in the Localize experiment (when the oracle is $\mathcal{O}_f$ instead of $\mathcal{O}$), conditioned on reaching checkpoint pair $i^\star$. Then, the second inequality follows from Claim 9.21. □

We now introduce a sequence of hybrids, where the first corresponds to the MultiStageSearchMonogamy reduction, and the last corresponds to the extraction experiment. Throughout these hybrids, probabilities are conditioned on $\mathcal{A}_M^0$ not aborting, corresponding to the event where the real execution of Localize reaches checkpoint pair $i^\star$.

Hybrid 1: This denotes the following experiment.

1. Run $\mathcal{A}_M^0$ from the reduction above, and take its middle output $\mathbf{M}' = (\mathsf{info}, \mathbf{M})$. Receive $\mathcal{O}_{S+v}^{b_0}$ from the MultiStageSearchMonogamy challenger. Take $x_{i^\star}, h_{2i^\star}, h_{2i^\star+1}, f \leftarrow \mathsf{info}$.
2. Simulate $\mathcal{P}_\nabla^*$ (defined above, in the description of $\mathcal{E}$) on input $\mathbf{M}$. In this simulation, oracle queries to $\mathcal{O}$ are answered with $\mathcal{O}'$, which depends on $x_{i^\star}, h_{2i^\star}, f$ and is described below. Here, H is a compressed oracle on $\mathcal{Y}_\lambda \to \{0,1\}^m$.

$$\mathcal{O}'(h, x', y, \sigma, c, z) = \begin{cases} H(f(x')) & \text{if } c = 0 \wedge x' = x_{i^\star} \wedge h = h_{2i^\star} \wedge y = f(x') \wedge \mathcal{O}(h, x', y, \sigma, c, z) \neq \bot, \\ \mathcal{O}(h, x', y, \sigma, c, z) & \text{otherwise.} \end{cases}$$

3. After simulating $\mathcal{P}_\nabla^*$, measure the compressed oracle database of H.
4. If there is exactly one non-$\bot$ entry y, output y. Otherwise, output $\bot$.

Claim 9.24. *Let p_1 denote the probability that the output of **Hybrid 1** is exactly $f(x_{i^\star})$. Then,*

$$p_1 \geq 1 - 2\sqrt{2\varepsilon} - \mathsf{negl}(\lambda)$$

Proof. Consider the measurement of the H database. Clearly there can be at most one non-$\bot$ entry, since H can only be accessed at $f(x_{i^\star})$. Thus we must now argue that the probability that the measurement of the $H(f(x_{i^\star}))$ database entry is $\bot$ is at most $2\sqrt{2\varepsilon} + \mathsf{negl}(\lambda)$. We will do this as follows:

First, we notice that steps 1 and 2 of **Hybrid 1** are the same experiment as running $\mathcal{A}_M^0$ and then $\mathcal{A}_M^1$ in the MultiStageSearchMonogamy game, except that the queries $\mathcal{O}(h_{2i^\star}, x_{i^\star}, f(x_{i^\star}), \sigma, 0, z)$ which $\mathcal{A}_M^1$ would reprogram to $\mathcal{O}_{S+v}^{b_0}(z)$ are now instead reprogrammed to $H(f(x_{i^\star}))$. In other words, $\mathcal{O}_{\mathsf{random}}(h_{2i^\star}, x_{i^\star}, f(x_{i^\star}), 0)$ is now reprogrammed to $H(f(x_{i^\star}))$ instead of b_0. These two experiments are identical, since b_0 and $H(f(x_{i^\star}))$ are both uniformly random. Then, by Claim 9.23, if we run the MultiStageSearchMonogamy reduction but replace $\mathcal{A}_M^0$ and $\mathcal{A}_M^1$ with steps 1 and 2 of **Hybrid 1**, it would succeed conditioned on non-abort with probability $1 - \varepsilon - \mathsf{negl}(\lambda)$.

Finally, let $\mathbf{B}$ denote the $f(x_{i^\star})^{th}$ entry in the compressed oracle database of H, which will initially be in the uniform superposition over m-bit output strings. We notice that the simulated

prover has the following access to $\mathbf{B}$:

$$\mathcal{O}(h_{2i^\star}, x_{i^\star}, f(x_{i^\star}), \sigma, 0, \cdot) = \sum_{b_0 \in \mathbb{F}_2^{m_0}} |b_0\rangle_{\mathbf{B}} \langle b_0| \otimes \mathcal{O}_{S+v}^{b_0}.$$

and that there is no other dependence that the prover can have on $\mathbf{B}$. This is the same as the interface given to the adversary in Definition 6.1. Now, given the last paragraph, Lemma 6.20 applies: if we write the purified state of the entire **Hybrid 1** experiment, after step 2, as

$$|\psi\rangle_{\mathbf{AB}} = \sum_{\mathsf{sk}} \sum_{b_0 \in \mathbb{F}_2^m} \left|\psi_{b_0}^{\mathsf{sk}}\right\rangle_{\mathbf{A}} |b_0\rangle_{\mathbf{B}} \ ,$$

where $\mathsf{sk} = (S, u, v)$, $\mathbf{A}$ is all of $\mathcal{P}^*$'s registers and messages at the top edge of the region ∇, and $\mathbf{B}$ is the purified choice of b_0 from Definition 6.1, then,

$$\mathrm{Tr}\left(|+_{\mathbf{B}}\rangle_{\mathbf{B}} \langle +_{\mathbf{B}}| \cdot |\psi\rangle_{\mathbf{AB}} \langle \psi|\right) \leq 2\sqrt{2\varepsilon} + \mathsf{negl}(\lambda).$$

Thus, if we measure this register in the Hadamard basis, it will be $\perp$ with probability $\leq 2\sqrt{2\varepsilon} + \mathsf{negl}(\lambda)$. This proves the claim. □

Hybrid 2: This denotes the following experiment:

1. Run $\mathcal{A}_M^0$ from the reduction above, and take its middle output $\mathbf{M}' = (\mathsf{info}, \mathbf{M})$. Receive $\mathcal{O}_{S+v}^{b_0}$ from the MultiStageSearchMonogamy challenger. Take $x_{i^\star}, h_{2i^\star}, h_{2i^\star+1} \leftarrow \mathsf{info}$ (discarding f).
2. Simulate $\mathcal{P}_\nabla^*$ on input $\mathbf{M}$. In this simulation, oracle queries to $\mathcal{O}$ are answered with $\mathcal{O}''$, which depends on $x_{i^\star}, h_{2i^\star}$ and is described below. Here, H is a compressed oracle on $\mathcal{Y}_\lambda \to \{0,1\}^m$.

$$\mathcal{O}''(h, x', y, \sigma, c, z) = \begin{cases} H(y) & \text{if } c = 0 \wedge x' = x_{i^\star} \wedge h = h_{2i^\star} \wedge \mathcal{O}(h, x', y, \sigma, c, z) \neq \perp, \\ \mathcal{O}(h, x', y, \sigma, c, z) & \text{otherwise.} \end{cases}$$

3. After simulating $\mathcal{P}_\nabla^*$, measure the compressed oracle database of H.
4. If there is exactly one non-$\perp$ entry y, output y. Otherwise, output $\perp$.

Claim 9.25. *Let p_2 denote the probability that the output of **Hybrid 2** is exactly $f(x_{i^\star})$. Then,*

$$|p_2 - p_1| \leq \mathsf{negl}(\lambda).$$

Proof. We can first swap $\mathcal{O}$ in **Hybrid 1** for $\mathcal{O}_f$, which is distinguishable with only negligible probability by Claim 9.21. Then, the check for $y = f(x_{i^\star})$ becomes redundant, since it already exists in $\mathcal{O}_f(h, x', y, \sigma, c, z) \neq \perp$, so we can remove it from the definition of $\mathcal{O}'$. Then, we switch back to $\mathcal{O}$ by another application of Claim 9.21, and we get the definition of $\mathcal{O}''$. Since the oracle is now independent of f, we can discard f from info. □

Hybrid 3: This denotes the following experiment.

1. Run $\mathsf{pp}, \mathsf{sp}, \mathsf{ek}, f, \rho \leftarrow \mathsf{Setup}(1^\lambda)$.
2. Start the procedure $\mathsf{Localize}\left(\mathcal{P}^*(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp}, f)\right)(\mathsf{pp}, L, t)$, but freeze all circuits and messages at time t. Let $\mathbf{B}^* := \mathsf{register}[L_\Delta \mathbin{@} t]\big(\mathsf{Localize}\left(\mathcal{P}^*(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp}, f)\right)(\mathsf{pp}, L, t)\big)$.
3. Use the code of $\mathcal{P}^*$, along with the oracles $(\mathcal{O}, \mathcal{O}_{\mathsf{chal}}) = \mathsf{pp}$, to evolve $\mathbf{B}^*$ forward to $\mathsf{register}[L_{2\delta} \mathbin{@} t^\star]$ and then find the state on all registers and messages in the region described by the following:
   (a) $s - p = t^\star - L$;

(b) $s + p = t^\star + L$;
(c) $p \in [L - 2\delta, L + 2\delta]$,

which we call $R_\vee$. The output of this step is

$$\mathbf{M} := \mathsf{register}[R_\vee](\text{Forward time-evolution of } \mathbf{B}^*)$$

4. Compute $\mathsf{info} = (x_{i^\star}, h_{2i^\star}, h_{2i^\star+1})$ for checkpoint pair $i^\star$ from the transcript of verifier messages in the forward simulation.
5. Run steps 2-4 of **Hybrid 2** with inputs $(\mathsf{info}, \mathbf{M})$.
   - Simulate $\mathcal{P}^*_\nabla$ on input $\mathbf{M}$. In this simulation, oracle queries to $\mathcal{O}$ are answered with $\mathcal{O}''$, which depends on $x_{i^\star}, h_{2i^\star}$ and is described in **Hybrid 2**.
   - After simulating $\mathcal{P}^*_\nabla$, measure the compressed oracle database of H.
   - If there is exactly one non-$\perp$ entry y, output y. Otherwise, output $\perp$.

Claim 9.26. *Let p_3 denote the probability that the output of **Hybrid 3** is exactly $f(x_{i^\star})$. Then,*

$$p_3 = p_2 \,.$$

Proof. By examining the definition of $\mathcal{A}^0_M$, one can see that the output

$$(\mathsf{info} = (x_{i^\star}, h_{2i^\star}, h_{2i^\star+1}), \mathbf{M}) \leftarrow \mathcal{A}^0_M()$$

is equal to the output of the simulation in steps 1-4 above. In other words, we argue that since the reduction $\mathcal{A}$ is specified in such a way as to perfectly simulate the Localize experiment, **Hybrid 2** and **Hybrid 3** are the same experiment up to rewriting of steps. □

Hybrid 4: This denotes the following experiment.

1. Run $\mathsf{pp}, \mathsf{sp}, \mathsf{ek}, f, \mathbf{B} \leftarrow \mathsf{Setup}(1^\lambda)$, and sample $x \leftarrow \mathcal{D}$.
2. Start the procedure $\mathsf{Localize}\,(\mathcal{P}^*(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp}, f))\,(\mathsf{pp}, L, t)$, but freeze all circuits and messages at time t. Let $\mathbf{B}^* := \mathsf{register}[L_\Delta \mathbin{@} t]\big(\mathsf{Localize}\,(\mathcal{P}^*(\mathbf{B}) \rightleftharpoons \mathcal{V}(\mathsf{sp}, f))\,(\mathsf{pp}, L, t)\big)$.
3. Run $\mathbf{M} \leftarrow \mathcal{E}_0(\mathsf{pp}, \mathsf{ek}, \mathbf{B}^*)$.
4. Run $y \leftarrow \mathcal{E}_1(\mathsf{pp}, \mathsf{ek}, \mathbf{M}, x)$, and output y.

Claim 9.27. *Let p_4 denote the probability that the output of **Hybrid 4** is exactly $f(x)$. Then,*

$$|p_4 - p_3| \leq \mathsf{negl}(\lambda)$$

Proof. We notice that step 3 of **Hybrid 3** is definitionally equivalent to $\mathbf{M} \leftarrow \mathcal{E}_0(\mathsf{pp}, \mathbf{B}^*)$. Also, the fresh input x in **Hybrid 4** has the same distribution as $x_{i^\star}$ in **Hybrid 3**. Thus, the remaining steps are exactly the same procedure as $y \leftarrow \mathcal{E}_1(\mathsf{pp}, \mathbf{M}, x)$, except that $\mathcal{E}_1$ reprograms $\mathcal{O}_{\mathsf{chal}}$ at $\hat{x}_{i^\star}$ so the selected checkpoint challenge is this fresh x rather than the original $x_{i^\star}$. By Lemma 3.1, this reprogramming can be detected only with negligible probability. □

Now, all that is left to notice is that **Hybrid 4** is precisely the extraction experiment for checkpoint pair $i^\star$. Also, $\mathcal{E}$ is efficient. Thus, as a consequence of Claim 9.24 through Claim 9.27, the extractor succeeds with probability $1 - 2\sqrt{2\varepsilon} - \mathsf{negl}(\lambda)$ conditioned on reaching checkpoint pair $i^\star$. The overall extraction experiment succeeds with probability at least

$$\eta(\lambda)\left(1 - 2\sqrt{2\varepsilon} - \mathsf{negl}(\lambda)\right).$$

□

Finally, $\varepsilon = 1 - q_{i^\star} \leq 1 - \eta^{1/\gamma} \leq \log(1/\eta)/\gamma$. The loss term $2\sqrt{2\varepsilon}$ is

$$O\left(\sqrt{\frac{\log(1/\eta)}{\gamma}}\right).$$

□

Lemma 9.28 (Uniqueness)**.** *Assume that $\mathcal{F}$ is a copy-protectable functionality with security $\zeta = \zeta(\lambda)$, and that* QSO *is an ideal obfuscator. Then, Construction 9.16 satisfies ζ-**Uniqueness**.*

Proof. As noted in [CG24], QSO is a best-possible copy-protector. In other words, if $\mathcal{F}$ is copy-protectable with security α, then $(\mathsf{QSO.QObf}, \mathsf{QSO.QEval})$ is a copy-protection scheme with security at least α.

□

Theorem 9.29. *Let $\mathcal{F}$ be a classical functionality which is copy-protectable with security $\zeta(\lambda)$. Then there exists a non-destructive functionality localization scheme for $\mathcal{F}$ with $(1 - \mathsf{negl}(\lambda))$-**Completeness**, (β, Δ)-**Extraction Soundness** for*

$$\beta(\lambda, \eta) = \eta(\lambda)\left(1 - O\left(\sqrt{\frac{\log(1/\eta)}{\gamma}}\right) - \mathsf{negl}(\lambda)\right)$$

*and $\zeta(\lambda)$-**Uniqueness** in the classical oracle model. In particular, since we only consider adversaries for which $1/\eta$ is at most polynomial in λ, for any polynomial $t(\lambda)$, we can set $\gamma = c\lambda t^2$ for some constant c to achieve*

$$\beta(\lambda, \eta) = \eta(\lambda)\left(1 - 1/t(\lambda)\right) .$$

Proof. Instantiating Construction 9.16 in the classical oracle model gives such a scheme as claimed, with the properties following from Lemma 9.19, Lemma 9.20, and Lemma 9.28, respectively. □

10 Discussion

10.1 Publicly-Verifiable Localization

All of our constructions in Section 9 utilize some secret parameters sp generated by Setup and then held privately by the verifiers. The purpose of sp is, generally speaking, to serve as a verification key for the protocol – i.e. the eventual accept/reject decision is a deterministic predicate of two things, 1. the (classical) protocol transcript and 2. the secret parameters sp. It is natural to see if these can be made public, a la publicly-verifiable digital signatures. However unfortunately, in our constructions sp contains the coset descriptors u, v – meaning that a prover knowing sp, even if he completely discards ρ, could respond to all challenges perfectly, nullifying any extraction guarantees.

An optimistic reader might hope for alternative solutions, and therefore ask:
*Are there sound localization schemes in which the verifiers hold no secret information? I.e., schemes where **anyone** can play the role of the verifiers, and achieve an extraction guarantee?*

In this section, we will argue that we can indeed achieve this for entanglement localization, state localization, and trajectory verification, by making a slight modification to our existing constructions. Firstly, we introduce the formal notion of public verifiability.

Definition 10.1. *We say that a localization scheme (e.g. one of Definitions 9.1,9.4,9.7, or 9.14) is* ***publicly verifiable*** *if the* Localize *protocol takes no private verifier input* sp*. In other words, if the interface for the* Localize *protocol can be written as* $\mathsf{Localize}\,(\mathcal{P}(\mathbf{B}) \rightleftharpoons \mathcal{V})\,(\mathsf{pp}, \dots) \to \mathbf{B}', b$.

Next, we will show an idea for how to augment our constructions of each aforementioned localization scheme, so that the verifiers can check the protocol transcript without knowledge of u, v. In order to give the general idea and intuition without getting too deep into the implementation details for each specific protocol, we will keep this explanation relatively high-level.

Theorem 10.2 (informal)**.** *Each of Construction 9.2, Construction 9.8, and Construction 9.5 can be made publicly verifiable*[20]*.*

Proof. To do this, we will apply the same, simple technique to each construction. As an initial (flawed) attempt, consider adding a public verification oracle like the following:

$$\mathcal{O}_{\mathsf{ver}}(\{\mathsf{challenge}_i\}_i, \{\mathsf{response}_i\}_i) = \begin{cases} \top & \text{if } \mathsf{response}_i = \mathcal{O}_{\mathsf{random}}(\mathsf{challenge}_i)\ \ \forall i, \\ \bot & \text{otherwise.} \end{cases}$$

This is essentially an oracle for the verifiers' accept/reject decision: it takes as input the entire protocol transcript, and one can check that its acceptance predicate is logically equivalent to the one computed by the verifiers in each of our constructions (under the proper interpretations of "challenge" and "response").

However, we cannot add such a public oracle – in our reduction to the monogamy game, the simulated prover might play a malicious trick on us: before sending a response, check if it's correct. If yes, send the response, and if not, instead send garbage. By doing so, a prover has not changed its success probability whatsoever. However, the monogamy game reduction is required to answer queries to this verification oracle – without knowing what the right answers are! So a prover can succeed with very high probability in the real experiment, but might (maliciously) fail to produce correct answers in any reduction where the verification oracle cannot be faithfully simulated.

To get around this, we use the following idea: at the start of Localize, the verifier will generate a random "key" which somehow gates access to the public verification oracle on a certain set of inputs, and release this key at the very end of the protocol. Challenges are then generated with respect to this key, meaning that the prover is unable to use the public verification oracle on the active protocol transcript until after it has produced all of its responses, circumventing the malicious behavior described above.

One such gating mechanism can be made with a simple hashing trick. Let H be a new public random oracle, such that H and $\mathcal{O}_{\mathsf{ver}}$ are now both generated during setup and output as part of the public parameters pp. The verifier will generate the key and challenges jointly at the start of Localize as follows:

$$\mathsf{key} \leftarrow \{0,1\}^n\ ,$$

$$\mathsf{challenge}_n = H(\mathsf{key}),\ \ \mathsf{challenge}_{n-1} = H(\mathsf{challenge}_n) = H^{(2)}(\mathsf{key}),\ \ \cdots\ , \mathsf{challenge}_1 = H^{(n)}(\mathsf{key})\ .$$

These challenges will then be directly used in place of the randomly-generated challenges from our original, privately-verifiable protocols. Additionally, we will now swap out the faulty verification oracle from above, and use the following new one which will be generated in Setup jointly

[20] We leave out functionality localization from this statement, because it is unclear how public verifiability should be defined with respect to the private functionality f. If f itself has a natural notion of public verification (e.g. if f is a copy-protectable signing algorithm for a public-key signature scheme), then we could potentially hope to extend the techniques in this section to allow such a functionality f to be localized in a publicly verifiable way.

with H:

$$\mathcal{O}_{\mathsf{ver}}(\mathsf{key}, \{\mathsf{challenge}_i\}_i, \{\mathsf{response}_i\}_i) = \begin{cases} \top & \text{if } H^{(n-i+1)}(\mathsf{key}) = \mathsf{challenge}_i \ \ \forall i \\ & \wedge \ \mathsf{response}_i = \mathcal{O}_{\mathsf{random}}(\mathsf{challenge}_i) \ \ \forall i, \\ \bot & \text{otherwise.} \end{cases}$$

Finally, the verifier broadcasts the key in a secret-shared manner (like all of the challenges) after all prover responses have been received, thus enabling public verification of the transcript.

To argue that this transformation preserves extraction soundness, we will describe the reduction from a given round i of the publicly-verifiable protocol to the MultiStageIndependentMonogamy game — an updated version of the reduction $\mathcal{A}$ defined in Theorem 8.4. The reduction will work in the same way as for the original schemes, except that it is now tasked with simulating the new oracles $\mathcal{O}_{\mathsf{ver}}$ and H, as well as generating challenges and the key in the updated way. It uses a fresh $2q$-wise independent hash function to simulate H in the same way as was done for $\mathcal{O}_{\mathsf{random}}$, and generates simulated verifier challenges and key according to H just as in the modified construction. It remains to show that $\mathcal{O}_{\mathsf{ver}}$ can be simulated. On a query $(\mathsf{key}', \{\mathsf{challenge}'_i\}_i, \{\mathsf{response}'_i\}_i)$, $\mathcal{A}$ first checks if $\mathsf{challenge}'_i = (\theta_0, x_{L,0} \oplus x_{R,0})$ or $\mathsf{challenge}'_i = (\theta_1, x_{L,1} \oplus x_{R,1})$ for any i. If yes, then it outputs $\bot$, otherwise it checks the predicate according to its simulated $\mathcal{O}_{\mathsf{random}}$ and H and outputs appropriately. We then use a standard one-way to hiding argument to show that this simulation is computationally indistinguishable from the real experiment, since it is clearly hard to find any value key' such that $x_{L,0} \oplus x_{R,0}$ or $x_{L,1} \oplus x_{R,1}$ is an iterate of key' under H. Thus, the reduction still wins with only negligible loss, and extraction proceeds as usual. □

10.1.1 Applications to Black-Box Trajectory Verification

Aside from being an interesting and desirable property on its own, public verifiability of localization protocols has a secondary benefit: allowing trajectory verification to be constructed in a relatively black-box way from a state or entanglement localization protocol. The construction is simple: run Setup as normal, and then execute back-to-back instances of Localize for each checkpoint along the trajectory. The details are somewhat tedious, so we will argue about this construction at a very high level.

For completeness to hold, we must be careful that completeness of the localization protocol still holds when the prover is moving along the trajectory rather than staying in place (a la Figure 11), and that non-destructiveness of Localize applies over a short enough timescale such that the prover's state ρ can be plugged into the next instance right away.

As for soundness, public verifiability allows for a simple black-box reduction: given an adversary for trajectory verification, simulate rounds 1 through i via the public verification algorithm. Then, forward the real Localize challenges to the adversary for the given round, which it should win with high probability.

10.2 Position-Based Cryptography

Position-based authentication (PBA) is a position-based cryptographic primitive that naturally extends plain position verification, and was initially introduced together with the original construction of QPV, in Buhrman et al. ([BCF+14]). In a PBA protocol, the honest prover decides on a message m to relay to the verifiers, and the security guarantee is twofold: 1. When the honest prover is positioned in the secure location, no coalition of malicious parties located elsewhere can successfully tamper with the authenticated message, and 2. A coalition of parties with nobody

in the secure location cannot pass the authentication protocol, regardless of the message. The applications of such a scheme are quite numerous and natural, e.g. authenticating military orders from a secure command center, arguably even more so than for plain position verification. As for building PBA, [BCF+14] gave a construction based on any black-box quantum position verification scheme, and Unruh showed that their QROM f-BB84 position verification scheme could be straightforwardly extended to a PBA scheme secure against all QPT attackers in the QROM ([Unr14]).

However, we will now point out that the security notion described above, under which all PBA schemes from prior literature are proven secure, seems unsatisfying when put under scrutiny. Points 1 and 2 above rule out attacks where nobody is in the secure location, or where the honest prover algorithm runs in the presence of malicious tampering parties, but *does not* rule out attacks where there might be a distributed attack involving a (potentially unwitting) party in the secure location. In particular, the distributed attack on f-BB84 described in Section 1.5 (and depicted in Figure 2) is an example of such an attack on the PBA scheme of Buhrman et al., when their scheme is instantiated with the f-BB84 QPV scheme.

10.2.1 An explicit "attack" scenario

We briefly describe an adversarial strategy for the PBA construction of [BCF+14], which does not constitute an attack according to the standard security definition, but is nonetheless a concerning scenario for any real-world application of PBA. To describe this attack we'll first give a summary of their construction.

Construction 10.3 ([BCF+14])**.** *Take a QPV protocol whose routines are described as* $(\mathsf{Chal}, \mathsf{Resp}, \mathsf{Ver})$*:* Chal *generates verifier challenges, and* Resp *computes the prover's response, which is finally checked by* Ver *(including timings). Now, suppose the prover wishes to authenticate a bit* $c \in \{0,1\}$*.*

1. *First, the prover computes* $m = \mathsf{Enc}(c)$*, according to some carefully chosen multi-bit encoding* Enc*.*
2. *Next, for each bit* m_i*, the prover and verifier run the following protocol:*
   - *Verifiers send challenges* $c_L, c_R \leftarrow \mathsf{Chal}(1^\lambda)$*.*
   - *If* $m_i = 1$*, the prover responds to both verifiers with* $r_i \leftarrow \mathsf{Resp}(c_L, c_R)$*. If* $m_i = 0$*, however, the prover sends* $r_i = \perp$ *to both verifiers with some probability* q*, and otherwise responds honestly.*
3. *The verifiers compute* $b \leftarrow \mathsf{Dec}(\{i : \mathsf{Ver}(r_i) = \top\})$*.*

Intuitively, this protocol is secure in their model because if nobody is at the purported location, then indeed there is nothing a malicious prover can do to inject 1s into the encoding of the message. Injecting 0s is easier, since anyone can send $\perp$, but the encoding prevents this from ever succeeding to tamper with the true message bit.

We now get to describing our "attack". Imagine that there are three provers, P_L, P_M, P_R, where only P_M is at the true location. They participate in the scheme of Construction 10.3 where the QPV scheme is f-BB84, and have the following strategy. Say, without loss of generality, that the BB84 qubit comes from the left.

- P_L and P_R pre-determine a message bit $c \in \{0,1\}$. They compute the encoding $\mathsf{Enc}(c)$, and execute the following strategy for each index i of the encoded string:
  1. When P_L receives the qubit $H^\theta |b\rangle$, it applies a quantum one-time pad $X^r Z^s$, where the one-time pad key (r,s) belongs to private shared randomness of P_L and P_R. Importantly, P_M never sees this key. P_L forwards the scrambled qubit to P_M.

2. When P_M receives the qubit, it measures in the correct basis determined by the f-BB84 challenges. It sends the measurement outcome b' to P_L and P_R.
3. P_L and P_R receive this measurement outcome, and by this time they also have full knowledge of the f-BB84 challenges. Namely, they know which basis the middle prover measured in, and thus which correction to apply from the quantum one-time pad in order to find the intended measurement outcome b.
4. If $m_i = 1$, then P_L and P_R send the outcome b to their respective nearest provers. Otherwise, if $m_i = 0$, then with probability q they instead send $\perp$ to each prover.

It is clear that this attack would not violate the typical security definition, since it involves active participation from a prover P_M who is in the secure location. However, there is a very concerning property of this scheme – namely, that P_M is *entirely unaware* of the message being authenticated. In other words, there might be a prover in the right location who was simply tricked into participating, and the authenticated message in no way originates from the secure location.

10.2.2 Towards stronger security notions

To address this attack, we give some informal directions for tackling this weakness in the security definition of PBA. We propose that the localization techniques developed in this paper, for example that of functionality localization, could be used to rule out this class of cheating strategies where the middle prover is acting blindly, without knowledge of the message.

A concrete idea is the following (informal) PBA scheme:

- Let $\mathcal{F} = \{f_k\}_k$ be a PRF. Take $(\mathsf{Setup}, \mathsf{Localize}, \mathsf{Eval})$ from the functionality localization scheme in Construction 9.16 instantiated for $\mathcal{F}$. Let H be a public random oracle.
- The verifiers run $\mathsf{pp}, \mathsf{sp}, \mathsf{ek}, f_k, \rho \leftarrow \mathsf{Setup}(1^\lambda)$, and send pp, ρ to the prover. Next, they sample two random challenge strings $c_L, c_R \in \{0,1\}^\lambda$.
- Two routines are executed simultaneously:
  - The prover and verifiers execute
  $$\mathsf{Localize}\big(P(\rho) \rightleftharpoons V(\mathsf{sp}, f)\big)(\mathsf{pp}, L, t)$$
  - The left and right verifiers send c_L and c_R, respectively, towards (L, t). The prover receives these, computes $c := c_L \oplus c_R$ and runs $y \leftarrow \mathsf{Eval}(\rho, H(m||c))$. The prover sends (m, y) to both verifiers simultaneously.
- The verifiers accept iff $\mathsf{Localize}$ accepted and $y = f_k(H(m||(c_L \oplus c_R)))$.

Intuitively, the point of this construction is to authenticate the prover's message with the copy-protected PRF, which is simultaneously being localized to a single position. If a group of provers pass this protocol with high probability, then soundness of functionality localization guarantees that we can extract the ability to sign (say, uniformly random) messages from the circuits at the secure location. Uniqueness also guarantees that this cannot be happening anywhere else at time t. Thus, when the verifiers receive the signed message $m, f_k(H(m||c))$, we know that this evaluation must have taken place locally at the secure location, which additionally tells us that the prover queried the oracle value $H(m||c)$ at this location at time t. The techniques from our localization constructions would then allow this query to be recorded and "extracted" from the prover's circuitry at that position.

We leave further exploration of this direction to future work, e.g. formalize a stronger notion of security for position-based authentication and analyze the above construction in terms of said notion.

References

[Aar09] Scott Aaronson. "Quantum Copy-Protection and Quantum Money". In: *Proceedings of the 2009 24th Annual IEEE Conference on Computational Complexity*. 2009, pp. 229–242. DOI: 10.1109/CCC.2009.42. URL: https://doi.org/10.1109/CCC.2009.42 (cit. on pp. 9, 72).

[AB24] Prabhanjan Ananth and Amit Behera. "A Modular Approach to Unclonable Cryptography". In: *Advances in Cryptology – CRYPTO 2024: 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2024, Proceedings, Part VII*. 2024, pp. 3–37. DOI: 10.1007/978-3-031-68394-7_1. URL: https://doi.org/10.1007/978-3-031-68394-7_1 (cit. on p. 9).

[ABH+26] Prabhanjan Ananth, Amit Behera, Zikuan Huang, Fuyuki Kitagawa, and Takashi Yamakawa. "Copy-Protection from Unclonable Puncturable Obfuscation, Revisited". In: *Advances in Cryptology – EUROCRYPT 2026*. 2026, pp. 541–570 (cit. on p. 9).

[AC12] Scott Aaronson and Paul Christiano. "Quantum money from hidden subspaces". In: *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*. 2012, pp. 41–60. DOI: 10.1145/2213977.2213983. URL: https://doi.org/10.1145/2213977.2213983 (cit. on p. 10).

[ACCM25] Vahid Asadi, Richard Cleve, Eric Culf, and Alex May. "Linear gate bounds against natural functions for position-verification". In: *Quantum* 9 (2025), p. 1604. DOI: 10.22331/q-2025-01-21-1604. URL: http://dx.doi.org/10.22331/q-2025-01-21-1604 (cit. on p. 3).

[ACM25] Vahid R. Asadi, Eric Culf, and Alex May. *Rank lower bounds on non-local quantum computation*. 2025. arXiv: 2402.18647 [quant-ph]. URL: https://arxiv.org/abs/2402.18647 (cit. on p. 3).

[ALL+21] Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. "New Approaches for Quantum Copy-Protection". In: *Advances in Cryptology – CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part I*. 2021, pp. 526–555. DOI: 10.1007/978-3-030-84242-0_19. URL: https://doi.org/10.1007/978-3-030-84242-0_19 (cit. on pp. 9, 10).

[BBV24] James Bartusek, Zvika Brakerski, and Vinod Vaikuntanathan. "Quantum State Obfuscation from Classical Oracles". In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*. 2024, pp. 1009–1017. DOI: 10.1145/3618260.3649673. URL: https://doi.org/10.1145/3618260.3649673 (cit. on pp. 9, 10, 20, 72, 73).

[BCF+14] Harry Buhrman, Nishanth Chandran, Serge Fehr, Ran Gelles, Vipul Goyal, Rafail Ostrovsky, and Christian Schaffner. "Position-Based Quantum Cryptography: Impossibility and Constructions". In: *SIAM J. Comput.* 43.1 (2014), pp. 150–178. DOI: 10.1137/130913687 (cit. on pp. 3, 12, 84, 85).

[BCS21] Andreas Bluhm, Matthias Christandl, and Florian Speelman. *Position-based cryptography: Single-qubit protocol secure against multi-qubit attacks*. 2021. arXiv: 2104.06301v2 [quant-ph] (cit. on p. 3).

[BGI+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. "On the (im)possibility of obfuscating programs". In: *J. ACM* 59.2 (2012), 6:1–6:48 (cit. on p. 10).

[BJRT25] James Bartusek, Ruta Jawale, Justin Raizes, and Kabir Tomer. *A New Approach to Arguments of Quantum Knowledge*. 2025. arXiv: 2510.05316 [quant-ph]. URL: https://arxiv.org/abs/2510.05316 (cit. on pp. 10, 12).

[BKNY23] James Bartusek, Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. "Obfuscation of Pseudo-Deterministic Quantum Circuits". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*. 2023, pp. 1567–1578. DOI: 10.1145/3564246.3585179. URL: https://doi.org/10.1145/3564246.3585179 (cit. on p. 10).

[BM22] James Bartusek and Giulio Malavolta. "Indistinguishability Obfuscation of Null Quantum Circuits and Applications". In: *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*. Vol. 215. 2022, 15:1–15:13. DOI: 10.4230/LIPIcs.ITCS.2022.15. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2022.15 (cit. on p. 10).

[BS23] Shalev Ben-David and Or Sattath. "Quantum Tokens for Digital Signatures". In: *Quantum* 7 (2023), p. 901. DOI: 10.22331/q-2023-01-19-901. URL: https://doi.org/10.22331/q-2023-01-19-901 (cit. on p. 10).

[CG24] Andrea Coladangelo and Sam Gunn. "How to Use Quantum Indistinguishability Obfuscation". In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*. 2024, pp. 1003–1008. DOI: 10.1145/3618260.3649779. URL: https://doi.org/10.1145/3618260.3649779 (cit. on pp. 9, 20, 72, 82).

[ÇG26] Alper Çakan and Vipul Goyal. "How to Copy-Protect Malleable-Puncturable Cryptographic Functionalities Under Arbitrary Challenge Distributions: A Unified Solution to Quantum Protection". In: *Advances in Cryptology – EUROCRYPT 2026*. 2026, pp. 481–509 (cit. on p. 9).

[CGMO09] Nishanth Chandran, Vipul Goyal, Ryan Moriarty, and Rafail Ostrovsky. "Position based cryptography". In: *Annual International Cryptology Conference*. Springer. 2009, pp. 391–407 (cit. on p. 3).

[CGMO14] Nishanth Chandran, Vipul Goyal, Ryan Moriarty, and Rafail Ostrovsky. "Position-Based Cryptography". In: *SIAM J. Comput.* 43.4 (2014), pp. 1291–1341. DOI: 10.1137/100805005 (cit. on p. 21).

[CLLZ21] Andrea Coladangelo, Jiahui Liu, Qipeng Liu, and Mark Zhandry. "Hidden Cosets and Applications to Unclonable Cryptography". In: *Advances in Cryptology – CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part I*. 2021, pp. 556–584. DOI: 10.1007/978-3-030-84242-0_20. URL: https://doi.org/10.1007/978-3-030-84242-0_20 (cit. on p. 9).

[CMP24] Andrea Coladangelo, Christian Majenz, and Alexander Poremba. "Quantum copy-protection of compute-and-compare programs in the quantum random oracle model". In: *Quantum* 8 (2024), p. 1330. DOI: 10.22331/q-2024-05-02-1330. URL: https://doi.org/10.22331/q-2024-05-02-1330 (cit. on p. 9).

[FSZ+26] Guan-Jie Fan-Yuan, Yang-Guang Shan, Cong Zhang, Yu-Long Wang, Yu-Xuan Fan, Wei-Xin Xie, De-Yong He, Shuang Wang, Zhen-Qiang Yin, Wei Chen, Song-Nian Fu, Guang-Can Guo, and Zheng-Fu Han. "Relativistic position verification with coherent states". In: *Nature Physics* (2026). DOI: 10.1038/s41567-026-03439-5. URL: https://doi.org/10.1038/s41567-026-03439-5 (cit. on p. 3).

[GGG+26] Uma Girish, Greg Gluch, Shafi Goldwasser, Tal Malkin, Leo Orshansky, and Henry Yuen. "Private proofs of when and where". In: *arXiv preprint arXiv:2601.18961* (2026) (cit. on pp. 3, 12, 21).

[HHL26] Zihan Hao, Zikuan Huang, and Qipeng Liu. *On the Need for (Quantum) Memory with Short Outputs*. 2026. arXiv: 2602.23763 [cs.CC]. URL: https://arxiv.org/abs/2602.23763 (cit. on p. 24).

[HT25] Mi-Ying Miryam Huang and Er-Cheng Tang. " Obfuscation of Unitary Quantum Programs ". In: *2025 IEEE 66th Annual Symposium on Foundations of Computer Science (FOCS)*. 2025, pp. 1665–1671. DOI: 10.1109/FOCS63196.2025.00088. URL: https://doi.ieeecomputersociety.org/10.1109/FOCS63196.2025.00088 (cit. on p. 10).

[HT26] Miryam Mi-Ying Huang and Er-Cheng Tang. *Obfuscation of Arbitrary Quantum Circuits*. 2026. arXiv: 2601.08969 [quant-ph]. URL: https://arxiv.org/abs/2601.08969 (cit. on p. 10).

[JLLW23] Aayush Jain, Huijia Lin, Ji Luo, and Daniel Wichs. "The Pseudorandom Oracle Model and Ideal Obfuscation". In: *Advances in Cryptology – CRYPTO 2023*. 2023, pp. 233–262 (cit. on p. 10).

[KKR26] Yael Tauman Kalai, Dakshita Khurana, and Justin Raizes. *How to Classically Verify a Quantum Cat without Killing It*. 2026. arXiv: 2602.09282 [quant-ph]. URL: https://arxiv.org/abs/2602.09282 (cit. on p. 12).

[KLC+25] Fatih Kaleoglu, Minzhao Liu, Kaushik Chakraborty, David Cui, Omar Amer, Marco Pistoia, and Charles Lim. *On the Equivalence between Classical Position Verification and Certified Randomness*. 2025. arXiv: 2410.03982 [quant-ph]. URL: https://arxiv.org/abs/2410.03982 (cit. on p. 12).

[KMS11] Adrian Kent, William J. Munro, and Timothy P. Spiller. "Quantum tagging: Authenticating location via quantum information and relativistic signaling constraints". In: *Phys. Rev. A* 84 (1 2011), p. 012326. DOI: 10.1103/PhysRevA.84.012326 (cit. on p. 3).

[KPB+25] Kirsten Kanneworff, Mio Poortvliet, Dirk Bouwmeester, Rene Allerstorfer, Philip Verduyn Lunel, Florian Speelman, Harry Buhrman, Petr Steindl, and Wolfgang Löffler. "Towards experimental demonstration of quantum position verification using single photons". In: *Quantum Science and Technology* 10.4 (2025), p. 045004. DOI: 10.1088/2058-9565/adf2da. URL: https://doi.org/10.1088/2058-9565/adf2da (cit. on p. 3).

[KY25] Fuyuki Kitagawa and Takashi Yamakawa. *Foundations of Single-Decryptor Encryption*. Cryptology ePrint Archive, Paper 2025/1219. 2025. URL: https://eprint.iacr.org/2025/1219 (cit. on pp. 33, 34).

[KZG+26] Gautam A. Kavuri, Yanbao Zhang, Abigail R. Gookin, Soumyadip Patra, Joshua C. Bienfang, Honghao Fu, Yusuf Alnawakhtha, Dileep V. Reddy, Michael D. Mazurek, Carlos Abellán, Waldimar Amaya, Morgan W. Mitchell, Sae Woo Nam, Carl A. Miller, Richard P. Mirin, Martin J. Stevens, Scott Glancy, Emanuel Knill, and Lynden K. Shalm. *Quantum Position Verification with Remote Untrusted Devices*. 2026. arXiv: 2601.16892 [quant-ph]. URL: https://arxiv.org/abs/2601.16892 (cit. on p. 3).

[LLQ21] Jiahui Liu, Qipeng Liu, and Luowen Qian. "Beating classical impossibility of position verification". In: *arXiv preprint arXiv:2109.07517* (2021) (cit. on pp. 9, 12).

[LLQZ22] Jiahui Liu, Qipeng Liu, Luowen Qian, and Mark Zhandry. "Collusion Resistant Copy-Protection for Watermarkable Functionalities". In: *Theory of Cryptography: 20th International Conference, TCC 2022, Chicago, IL, USA, November 7–10, 2022, Proceedings, Part I*. 2022, pp. 294–323. DOI: 10.1007/978-3-031-22318-1_11. URL: https://doi.org/10.1007/978-3-031-22318-1_11 (cit. on p. 9).

[TFKW13] Marco Tomamichel, Serge Fehr, Jędrzej Kaniewski, and Stephanie Wehner. "A monogamy-of-entanglement game with applications to device-independent quantum cryptography". In: *New Journal of Physics* 15.10 (2013), p. 103002. DOI: 10.1088/1367-2630/15/10/103002 (cit. on p. 3).

[Unr14] Dominique Unruh. "Quantum Position Verification in the Random Oracle Model". In: *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*. Vol. 8617. 2014, pp. 1–18. DOI: 10.1007/978-3-662-44381-1_1 (cit. on pp. 3, 10–12, 21, 23, 85).

[VZ21] Thomas Vidick and Tina Zhang. *Classical proofs of quantum knowledge*. 2021. arXiv: 2005.01691 [quant-ph]. URL: https://arxiv.org/abs/2005.01691 (cit. on pp. 13, 17, 53, 75).

[Zha12] Mark Zhandry. "Secure Identity-Based Encryption in the Quantum Random Oracle Model". In: *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*. Vol. 7417. 2012, pp. 758–775. DOI: 10.1007/978-3-642-32009-5_44 (cit. on p. 60).

[Zha19] Mark Zhandry. "How to record quantum queries, and applications to quantum indifferentiability". In: *Annual International Cryptology Conference*. Springer. 2019, pp. 239–268 (cit. on pp. 19, 24, 25).

[Zha21] Mark Zhandry. "How to Construct Quantum Random Functions". In: *J. ACM* 68.5 (2021). DOI: 10.1145/3450745. URL: https://doi.org/10.1145/3450745 (cit. on p. 10).