

Classically Verifiable NIZK for QMA with Preprocessing

Tomoyuki Morimae¹ and Takashi Yamakawa^{1,2}

¹Yukawa Institute for Theoretical Physics, Kyoto University, Kyoto, Japan

²NTT Social Informatics Laboratories, Tokyo, Japan

September 6, 2022

Abstract

We propose three constructions of classically verifiable non-interactive zero-knowledge proofs and arguments (CV-NIZK) for **QMA** in various preprocessing models.

1. We construct a CV-NIZK for **QMA** in the quantum secret parameter model where a trusted setup sends a quantum proving key to the prover and a classical verification key to the verifier. It is information theoretically sound and zero-knowledge.
2. Assuming the quantum hardness of the learning with errors problem, we construct a CV-NIZK for **QMA** in a model where a trusted party generates a CRS and the verifier sends an instance-independent quantum message to the prover as preprocessing. This model is the same as one considered in the recent work by Coladangelo, Vidick, and Zhang (CRYPTO '20). Our construction has the so-called dual-mode property, which means that there are two computationally indistinguishable modes of generating CRS, and we have information theoretical soundness in one mode and information theoretical zero-knowledge property in the other. This answers an open problem left by Coladangelo et al, which is to achieve either of soundness or zero-knowledge information theoretically. To the best of our knowledge, ours is the first dual-mode NIZK for **QMA** in any kind of model.
3. We construct a CV-NIZK for **QMA** with quantum preprocessing in the quantum random oracle model. This quantum preprocessing is the one where the verifier sends a random Pauli-basis states to the prover. Our construction uses the Fiat-Shamir transformation. The quantum preprocessing can be replaced with the setup that distributes Bell pairs among the prover and the verifier, and therefore we solve the open problem by Broadbent and Grilo (FOCS '20) about the possibility of NIZK for **QMA** in the shared Bell pair model via the Fiat-Shamir transformation.

1 Introduction

1.1 Background

The zero-knowledge [GMR89], which ensures that the verifier learns nothing beyond the statement proven by the prover, is one of the most central concepts in cryptography. Recently, there have been many works that constructed non-interactive zero-knowledge (NIZK) [BFM88] proofs or arguments for **QMA**, which is the “quantum counterpart” of **NP**, in various kind of models [ACGH20, CVZ20, BG20, Shm21, BCKM21, BM21]. We note that we require the honest prover to run in quantum polynomial-time receiving sufficiently many copies of a witness when we consider NIZK proofs or arguments for **QMA**. All known protocols except for the protocol of Broadbent and Grilo [BG20]

only satisfy computational soundness. The protocol of [BG20] satisfies information theoretical soundness and zero-knowledge in the *secret parameter (SP) model* [Ps05] where a trusted party generates proving and verification keys and gives them to the corresponding party while keeping it secret to the other party as setup.¹ A drawback of their protocol is that the prover sends a quantum proof to the verifier, and thus the verifier should be quantum. Therefore it is natural to ask the following question.

*Can we construct a NIZK proof for **QMA** with classical verification assuming a trusted party that generates proving and verification keys?*

In addition, the SP model is not a very desirable model since it assumes a strong trust in the setup. In the classical literature, there are constructions of NIZK proofs for **NP** in the common reference string (CRS) model [BFM88, FLS99, PS19] where the only trust in the setup is that a classical string is chosen according to a certain distribution and then published. Compared to the SP model, we need to put much less trust in the setup in the CRS model. Indeed, several works [BG20, CVZ20, Shm21] mention it as an open problem to construct a NIZK proofs (or even arguments) for **QMA** in the CRS model. Though this is still open, there are several constructions of NIZKs for **QMA** in different models that assume less trust in the setup than in the SP model [CVZ20, Shm21, BCKM21]. However, all of them are arguments. Therefore, we ask the following question.

*Can we construct a NIZK proof for **QMA** with classical verification in a model that assumes less trust in the setup than in the SP model?*

The Fiat-Shamir transformation [FS87] is one of the most important techniques in cryptography that have many applications. In particular, NIZK can be constructed from a Σ protocol: the prover generates the verifier’s challenge β by itself by applying a random oracle H on the prover’s first message α , and then the prover issues the proof $\pi = (\alpha, \gamma)$, where γ is the third message generated from α and $\beta = H(\alpha)$. It is known that Fiat-Shamir transform works in the post-quantum setting where we consider classical protocols secure against quantum adversaries [LZ19, DFMS19, DFM20]. On the other hand, it is often pointed out that (for example, [Shm21, BG20]) this standard technique cannot be used in the fully quantum setting. In particular, due to the no-cloning, the application of random oracle on the first message does not work when the first message is quantum like so-called the Ξ -protocol constructed by Broadbent and Grilo [BG20]. Broadbent and Grilo left the following open problem:

*Is it possible to construct NIZK for **QMA** in the CRS model (or shared Bell pair model) via the Fiat-Shamir transformation?*

Note that the shared Bell pair model is the setup model where the setup distributes Bell pairs among the prover and the verifier. It can be considered as a “quantum analogue” of the CRS [Kob03].

1.2 Our Results

We answer the above questions affirmatively.

1. We construct a classically verifiable NIZK (CV-NIZK) for **QMA** in the QSP model where a trusted party generates a quantum proving key and classical verification key and gives

¹The SP model is also often referred to as *preprocessing model* [DMP90].

them to the corresponding parties. We do not rely on any computational assumption for this construction either, and thus both soundness and the zero-knowledge property are satisfied information theoretically. This answers our first question. Compared with [BG20], ours has an advantage that verification is classical at the cost of making the proving key quantum. The proving key is a very simple state, i.e., a tensor product of randomly chosen Pauli X , Y , or Z basis states. We note that we should not let the verifier play the role of the trusted party for this construction since that would break the zero-knowledge property.

2. Assuming the quantum hardness of the learning with errors problem (the LWE assumption) [Reg09], we construct a CV -NIZK for QMA in a model where a trusted party generates a CRS and the verifier sends an instance-independent quantum message to the prover as preprocessing. We note that the CRS is reusable for generating multiple proofs but the quantum message in the preprocessing is not reusable. In this model, we only assume a trusted party that just generates a CRS once, and thus this answers our second question. This model is the same as one considered in [CVZ20] recently, and we call it the $CRS + (V \rightarrow P)$ model. Compared to their work, our construction has the following advantages.

- (a) In their protocol, both soundness and the zero-knowledge property hold only against quantum polynomial-time adversaries, and they left it open to achieve either of them information theoretically. We answer the open problem. Indeed, our construction has the so-called dual-mode property [GOS12, PS19], which means that there are two computationally indistinguishable modes of generating CRS, and we have information theoretical soundness in one mode and information theoretical zero-knowledge property in the other. To the best of our knowledge, ours is the first dual-mode NIZK for QMA in any kind of model.
- (b) Our protocol uses underlying cryptographic primitives (which are lossy encryption and oblivious transfer with certain security) only in a black-box manner whereas their protocol heavily relies on non-black-box usage of the underlying primitives. Indeed, their protocol uses fully homomorphic encryption to homomorphically runs the proving algorithm of a NIZK for NP , which would make the protocol extremely inefficient. On the other hand, our construction uses the underlying primitives only in a black-box manner, which results in a much more efficient construction. We note that black-box constructions have been considered desirable for both theoretical and practical reasons in the cryptography community (e.g., see introduction of [IKLP06]).
- (c) The verifier's quantum operation in our preprocessing is simpler than that in theirs: in the preprocessing of our protocol, the verifier has only to do single-qubit gate operations (Hadamard, bit-flip or phase gates), while in the preprocessing of their protocol, the verifier has to do five-qubit (entangled) Clifford operations. In their paper [CVZ20], they left the following open problem: how far their preprocessing phase could be weakened? Our construction with the weaker verifier therefore partially answers the open problem.

On the other hand, Coladangelo et al. [CVZ20] proved that their protocol is also an *argument of quantum knowledge (AoQK)*. We leave it open to study if ours is also a proof/argument of knowledge.

3. We construct a CV -NIZK for QMA with quantum preprocessing in the quantum random oracle model. This quantum preprocessing is the one where the verifier sends a random Pauli-basis states to the prover. Our construction uses the Fiat-Shamir transformation. Importantly, the quantum preprocessing can be replaced with the setup that distributes Bell

Table 1: Comparison of NIZKs for **QMA**.

Reference	Soundness	ZK	Ver.	Model	Assumption	Misc
[ACGH20]	comp.	comp.	C	SP	LWE + QRO	
[CVZ20]	comp.	comp.	Q+C	CRS + ($V \rightarrow P$)	LWE	AoQK
[BG20]	stat.	stat.	Q	SP	None	
[Shm21]	comp.	comp.	Q	MDV	LWE	reusable
[BCKM21]	comp.	comp.	Q	MDV	LWE	reusable and single-witness
[BM21]	comp.	stat.	C	CRS	iO + QRO (heuristic)	
Section 3	stat.	stat.	C	QSP	None	
Section 4	stat. comp.	comp. stat.	Q+C	CRS + ($V \rightarrow P$)	LWE	dual-mode
Section 5	comp. (query)	comp. (query)	C	$V \rightarrow P$ /Bell pair	QRO	

In column “Soundness” (resp. “ZK”), stat., and comp. mean statistical, and computational soundness (resp. zero-knowledge), respectively. Also, comp.(query) means that only the number of queries should be polynomial. In column “Ver.,” “Q” and “C” mean that the verification is quantum and classical, respectively, and “Q+C” means that the verifier needs to send a quantum message in preprocessing but the online phase of verification is classical. QRO means the quantum random oracle.

pairs among the prover and the verifier. The distribution of Bell pairs by the setup can be considered as a “quantum analogue” of the CRS. This result gives an answer to our third question (and the second question as well). (Note that both the soundness and zero-knowledge property of the construction are computational one, but it does not mean that we use some computational assumptions: just the oracle query is restricted to be polynomial time.)

Comparison among NIZKs for **QMA.** We give more comparisons among our and known constructions of NIZKs for **QMA**. Since we already discuss comparisons with ours and [BG20, CVZ20], we discuss comparisons with other works. A summary of the comparisons is given in Table 1.

Alagic et al. [ACGH20] gave a construction of a NIZK for **QMA** in the SP model. Their protocol has an advantage that both the trusted party and verifier are completely classical. On the other hand, the drawback is that only computational soundness and zero-knowledge are achieved, whereas our first two constructions achieve (at least) either statistical soundness or zero-knowledge. Their protocol also uses the Fiat-Shamir transformation with quantum random oracle like our third result, but their setup is the secret parameter model, whereas ours can be the sharing Bell pair model, which is a quantum analogue of the CRS model.

Shmueli [Shm21] gave a construction of a NIZK for **QMA** in the malicious designated-verifier (MDV) model, where a trusted party generates a CRS and the verifier sends an instance-independent classical message to the prover as preprocessing. In this model, the preprocessing is *reusable*, i.e., a single preprocessing can be reused to generate arbitrarily many proofs later. This is a crucial advantage of their construction compared to ours. On the other hand, in their protocol, proofs are quantum and thus the verifier should perform quantum computations in the online phase whereas the online phase of the verifier is classical in our constructions. Also, their protocol only satisfies computational soundness and zero-knowledge whereas we can achieve (at least) either of them statistically.

Recently, Bartusek et al. [BCKM21] gave another construction of a NIZK for **QMA** in the MDV model that has an advantage that the honest prover only uses a single copy of a witness. (Note that all other NIZKs for **QMA** including ours require the honest prover to take multiple copies of

a witness if we require negligible completeness and soundness errors.) However, their construction also requires quantum verifier in the online phase and only achieves computational soundness and zero-knowledge similarly to [Shm21].

Subsequently to our work, Bartusek and Malavolta [BM21] recently constructed the first CV-NIZK argument for **QMA** in the CRS model assuming the LWE assumption and ideal obfuscation for classical circuits. An obvious drawback is the usage of ideal obfuscation, which has no provably secure instantiation.² They also construct a witness encryption scheme for **QMA** under the same assumptions. They use the verification protocol of Mahadev [Mah18] and therefore the LWE assumption is necessary. If our CV-NIZK in the QSP model is used, instead, a witness encryption for **QMA** (with quantum ciphertext) would be constructed without the LWE assumption, which is one interesting application of our results.

1.3 Technical Overview

Classically verifiable NIZK for QMA in the QSP model. Our starting point is the NIZK for **QMA** in [BG20], which is based on the fact that a **QMA** language can be reduced to the 5-local Hamiltonian problem with *locally simulatable* history states [BG20, GSY19]. (We will explain later the meaning of “locally simulatable”.) An instance \mathbf{x} corresponds to an N -qubit Hamiltonian $\mathcal{H}_{\mathbf{x}}$ of the form $\mathcal{H}_{\mathbf{x}} = \sum_{i=1}^M p_i \frac{I+s_i P_i}{2}$, where $N = \text{poly}(|\mathbf{x}|)$, $M = \text{poly}(|\mathbf{x}|)$, $s_i \in \{+1, -1\}$, $p_i > 0$, $\sum_{i=1}^M p_i = 1$, and P_i is a tensor product of Pauli operators (I, X, Y, Z) with at most 5 nontrivial Pauli operators (X, Y, Z). There are $0 < \alpha < \beta < 1$ with $\beta - \alpha = 1/\text{poly}(|\mathbf{x}|)$ such that if \mathbf{x} is a yes instance, then there exists a state ρ_{hist} (called the *history state*) such that $\text{Tr}(\rho_{\text{hist}} \mathcal{H}_{\mathbf{x}}) \leq \alpha$, and if \mathbf{x} is a no instance, then for any state ρ , we have $\text{Tr}(\rho \mathcal{H}_{\mathbf{x}}) \geq \beta$.

The completeness and the soundness of the NIZK for **QMA** in [BG20] is based on the posthoc verification protocol [FHM18], which is explained as follows. To prove that \mathbf{x} is a yes instance, the prover sends the history state to the verifier. The verifier first chooses P_i with probability p_i , and measures each qubit in the Pauli basis corresponding to P_i . Let $m_j \in \{0, 1\}$ be the measurement result on j th qubit. The verifier accepts if $(-1)^{\oplus_j m_j} = -s_i$ and rejects otherwise. The probability that the verifier accepts is $1 - \text{Tr}(\rho \mathcal{H}_{\mathbf{x}})$ when the prover’s quantum message is ρ , and therefore the verifier accepts with probability at least $1 - \alpha$ if \mathbf{x} is a yes instance and the prover is honest whereas it accepts with probability at most $1 - \beta$ if \mathbf{x} is a no instance. (See Lemma 2.5 and [FHM18].) The gap between completeness and soundness can be amplified by simple parallel repetitions.

The verifier in the posthoc protocol is, however, not classical, because it has to receive a quantum state and measure each qubit. Our first idea to make the verifier classical is to use the quantum teleportation. Suppose that the prover and verifier share sufficiently many Bell pairs at the beginning. Then the prover can send the history state to the verifier with classical communication by the quantum teleportation. Though this removes the necessity of quantum communication, the verifier still needs to be quantum since it has to keep halves of Bell pairs and perform a measurement after receiving a proof.

To solve the problem, we utilize our observation that the verifier’s measurement and the prover’s measurement commute with each other, which is our second idea. In other words, we can let the verifier perform the measurement at the beginning without losing completeness or soundness. In the above quantum-teleportation-based protocol, when the prover sends its measurement outcomes $\{(x_j, z_j)\}_{j \in [N]}$ to the verifier, the verifier’s state collapses to $X^x Z^z \rho_{\text{hist}} Z^z X^x$ where ρ_{hist} denotes the history state and $X^x Z^z$ means $\prod_{j=1}^N X_j^{x_j} Z_j^{z_j}$. Then the verifier applies the Pauli correction

²In the latest version, they give a candidate instantiation based on indistinguishability obfuscation and random oracles. However, the instantiation is heuristic since they obfuscate circuits that involve the random oracle, which cannot be done in the quantum random oracle model.

$X^x Z^z$ and then measures each qubit in a Pauli basis. We observe that the Pauli correction can be applied even after the verifier measures each qubit because $X_j^{x_j} Z_j^{z_j}$ before a Pauli measurement on the j th qubit has the same effect as XOR by z_j or x_j after the measurement (see Lemma 2.2). Therefore, if a trusted party generates Bell pairs and measures half of them in random Pauli basis and gives the unmeasured halves to the prover as a proving key while the measurement outcomes to the verifier as a verification key, a completely classical verifier can verify the **QMA** promise problem.

The last remaining issue is that the distribution of bases that appear in P_i depends on the instance \mathbf{x} , and thus we cannot sample the distribution at the setup phase where \mathbf{x} is not decided yet. To resolve this issue, we use the following idea (which was also used in [ACGH20]). The trusted party just chooses random bases, and the verifier just accepts if they are inconsistent to P_i chosen by the verifier in the online phase. Since there are only 3 possible choices of the bases and P_i non-trivially acts on at most 5 qubits, the probability that the randomly chosen bases are consistent to P_i is at least 3^{-5} .³ Therefore we can still achieve inverse-polynomial gap between completeness and soundness.

The zero-knowledge property of the NIZK for **QMA** in [BG20] uses the local simulatability of the history state. It roughly means that a classical description of the reduced density matrix of the history state for any 5-qubit subsystem can be efficiently computable without knowing the witness. Broadbent and Grilo [BG20] used this local simulatability to achieve the zero-knowledge property as follows. A trusted party randomly chooses $(\hat{x}, \hat{z}) \leftarrow^{\$} \{0, 1\}^N \times \{0, 1\}^N$, and randomly picks a random subset $S_V \subseteq [N]$ such that $1 \leq |S_V| \leq 5$. Then it gives (\hat{x}, \hat{z}) to the prover as a proving key and gives $\{(\hat{x}_j, \hat{z}_j)\}_{j \in S_V}$ to the verifier as a verification key where \hat{x}_j and \hat{z}_j denote the j -th bits of \hat{x} and \hat{z} , respectively. The prover generates the history state ρ_{hist} and sends $\rho' = X^{\hat{x}} Z^{\hat{z}} \rho_{\text{hist}} Z^{\hat{z}} X^{\hat{x}}$ to the verifier as a proof. The verifier then measures each qubit as is done in the posthoc verification protocol. This needs the quantum verifier, but as we have explained, we can make the verifier classical by using the teleportation technique.

An intuitive explanation of why it is zero-knowledge is that the verifier can access at most five qubits of the history state, because other qubits are quantum one-time padded. Due to the local simulatability of the history state, the information that the verifier gets can be classically simulated without the witness. This results in our classically verifiable NIZK for **QMA** in the QSP model. In our QSP model, the trusted setup sends random Pauli basis states to the prover and their classical description to the verifier. Furthermore, the trusted setup also sends randomly chosen $(\hat{x}, \hat{z}) \leftarrow^{\$} \{0, 1\}^N \times \{0, 1\}^N$ to the prover, and $\{(\hat{x}_j, \hat{z}_j)\}_{j \in S_V}$ to the verifier with randomly chosen subset S_V .

Classically verifiable NIZK for QMA in the CRS + $(V \rightarrow P)$ model. We want to reduce the trust in the setup, so let us first examine what happens if the verifier runs the setup as preprocessing. Unfortunately, such a construction is not zero-knowledge since the verifier can know whole bits of (\hat{x}, \hat{z}) and thus it may obtain information of qubits of ρ_{hist} that are outside of S_V , in which case we cannot rely on the local simulatability. Therefore, for ensuring the zero-knowledge property, we have to make sure that the verifier only knows $\{(\hat{x}_j, \hat{z}_j)\}_{j \in S_V}$. Then suppose that the prover chooses (\hat{x}, \hat{z}) whereas other setups are still done by the verifier. Here, the problem is how to let the verifier know $\{(\hat{x}_j, \hat{z}_j)\}_{j \in S_V}$. A naive solution is that the verifier sends S_V to the prover and then the prover returns $\{(\hat{x}_j, \hat{z}_j)\}_{j \in S_V}$. However, such a construction is not sound since it is essential that the prover “commits” to a single quantum state independently of S_V when reducing

³There is a subtle issue that the probability depends on the number of qubits on which P_i non-trivially acts. We adjust this by an additional biased coin flipping.

soundness to the local Hamiltonian problem. So what we need is a protocol between the prover and verifier where the verifier only gets $\{(\hat{x}_j, \hat{z}_j)\}_{j \in S_V}$ and the prover does not learn S_V . We observe that this is exactly the functionality of 5-out-of- N oblivious transfer [BCR87].

Though it may sound easy to solve the problem by just using a known two-round 5-out-of- N oblivious transfer, there is still some subtlety. For example, if we use an oblivious transfer that satisfies only indistinguishability-based notion of receiver’s security (e.g., [NP01, BD18]),⁴ which just says that the sender cannot know indices chosen by the receiver, we cannot prove soundness. Intuitively, this is because the indistinguishability-based receiver’s security does not prevent a malicious sender from generating a malicious message such that the message derived on the receiver’s side depends on the chosen indices, which does not force the prover to “commit” to a single state.

If we use a *fully-simulatable* [Lin08] oblivious transfer, the above problem does not arise and we can prove both soundness and zero-knowledge. However, the problem is that we are not aware of any efficient fully-simulatable 5-out-of- N oblivious transfer based on post-quantum assumptions (in the CRS model). The LWE-based construction of [PVW08] does not suffice for our purpose since a CRS can be reused only a bounded number of times in their construction. Recently, Quach [Qua20] resolved this issue, and proposed an efficient fully-simulatable 1-out-of-2 oblivious transfer based on the LWE assumption.⁵ We can extend his construction to a fully-simulatable 1-out-of- N oblivious transfer efficiently. However, we do not know how to convert this into 5-out-of- N one efficiently without losing the full-simulatability. We note that a conversion from 1-out-of- N to 5-out-of- N oblivious transfer by a simple 5-parallel repetition loses the full-simulatability against malicious senders since a malicious sender can send different inconsistent messages in different sessions, which should be considered as an attack against the full-simulatability. One possible way to prevent such an inconsistent message attack is to let the sender prove that the messages in all sessions are consistent by using (post-quantum) CRS-NIZK for NP [PS19]. However, such a construction is very inefficient since it uses the underlying 1-out-of- N oblivious transfer in a non-black-box manner, which we want to avoid.

We note that the parallel repetition construction preserves indistinguishability-based receiver’s security and fully-simulatable sender’s security for two-round protocols. Therefore, we have an efficient (black-box) construction of 5-out-of- N oblivious transfer if we relax the receiver’s security to the indistinguishability-based one. As already explained, such a security does not suffice for proving soundness. To resolve this issue, we add an additional mechanism to force the prover to “commit” to a single state. Specifically, instead of directly sending (x, z) by a 5-out-of- N oblivious transfer, the prover sends a commitment of (x, z) and then sends (x, z) and the corresponding randomness used in the commitment by a 5-out-of- N oblivious transfer. When the verifier receives $\{x_j, z_j\}_{j \in S_V}$ and corresponding randomness, it checks if it is consistent to the commitment by recomputing it, and immediately rejects if not. This additional mechanism prevents a malicious prover’s inconsistent behavior, which resolves the problem in the proof of soundness.

Finally, our construction satisfies the dual-mode property if we assume appropriate dual-mode properties for building blocks. A dual-mode oblivious transfer (in the CRS model) has two modes of generating a CRS and it satisfies statistical (indistinguishability-based) receiver’s security in one mode and statistical (full-simulation-based) sender’s security in the other mode. The construction of [Qua20] is an instantiation of a 1-out-of-2 oblivious transfer with such a dual-mode property, and this can be converted into 5-out-of- N one as explained above. We stress again that it is important to relax the receiver’s security to the indistinguishability-based one to make the conversion work.

⁴The indistinguishability-based receiver’s security is also often referred to as half-simulation security [CNs07].

⁵Actually, his construction satisfies a stronger UC-security [Can20, PVW08].

A dual-mode commitment (in the CRS model) has two modes of generating a CRS and it is statistically binding in one mode and statistically hiding in the other mode. We can use lossy encryption [BHY09, Reg09] as an instantiation of such a dual-mode commitment. Both of dual-mode 5-out-of- N oblivious transfer and lossy encryption are based on the LWE assumption (with super-polynomial modulus for the former) and fairly efficient in the sense that they do not rely on non-black-box techniques. Putting everything together, we obtain a fairly efficient (black-box) construction of a dual-mode NIZK for **QMA** in the CRS + $(V \rightarrow P)$ model.

NIZK for QMA via Fiat-Shamir transformation. Finally, let us explain our construction of NIZK for **QMA** via the Fiat-Shamir transformation. It is based on so-called the Ξ -protocol for **QMA** [BG20], which is equal to the standard Σ -protocol except that the first message is quantum. Because the first message is quantum, the Fiat-Shamir technique cannot be directly applied. Our idea is again to use the teleportation technique: if we introduce a setup that sends random Pauli basis states to the prover and their classical description to the verifier, the first message can be classical. We thus obtain a (classical) Σ -protocol in the QSP model, where the trusted setup sends random Pauli basis states to the prover and their classical description to the verifier. This task can be, actually, done by the verifier, not the trusted setup, unlike our first construction. We therefore obtain a (classical) Σ -protocol with quantum preprocessing (Definition 5.3), where the verifier sends random Pauli basis states to the prover as the preprocessing.

We then apply the (classical) Fiat-Shamir transformation to the Σ -protocol with quantum preprocessing, and obtain the CV-NIZK for **QMA** in the quantum random oracle plus $V \rightarrow P$ model (Definition 5.1), where $V \rightarrow P$ means the communication from the verifier to the prover as the preprocessing. Note that we are considering a classical Σ -protocol with quantum preprocessing differently from previous works. By a close inspection, we show that an existing security proof for classical Σ -protocol in the QROM [DFM20] also works in our setting.

Importantly, in this case, unlike the previous two constructions, the quantum preprocessing can be replaced with the setup that distributes Bell pairs among the prover and the verifier. As a corollary, we therefore obtain NIZK for **QMA** in the shared Bell pair model (plus quantum random oracle). The distribution of Bell pairs by a trusted setup can be considered as a “quantum analogue” of the CRS, and therefore we can say that we obtain NIZK for **QMA** in the “quantum CRS” model via the Fiat-Shamir transformation.

1.4 Related Work

More related works on quantum NIZKs. Kobayashi [Kob03] studied (statistically sound and zero-knowledge) NIZKs in a model where the prover and verifier share Bell pairs, and gave a complete problem in this setting. It is unlikely that the complete problem contains (even a subclass of) **NP** [MW18] and thus even a NIZK for all **NP** languages is unlikely to exist in this model. Note that if we consider the prover and verifier sharing Bell pairs in advance like this model, the verifier’s preprocessing message of our protocols (and the protocol of [CVZ20]) becomes classical. Chailloux et al. [CCKV08] showed that there exists a (statistically sound and zero-knowledge) NIZK for all languages in **QSZK** in the help model where a trusted party generates a pure state *depending on the statement to be proven* and gives copies of the state to both prover and verifier.

Interactive zero-knowledge for QMA. There are several works of interactive zero-knowledge proofs/arguments for **QMA**. The advantage of these constructions compared to non-interactive ones is that they do not require any trusted setup. Broadbent, Ji, Song, and Watrous [BJSW20] gave the first construction of a zero-knowledge proof for **QMA**. Broadbent and Grilo [BG20] gave

an alternative simpler construction. Bitansky and Shmueli [BS20] gave the first constant round zero-knowledge argument for **QMA** with negligible soundness error. Brakerski and Yuen [BY20] gave a construction of 3-round *delayed-input* zero-knowledge proof for **QMA** where the prover needs to know the statement and witness only for generating its last message. By considering the first two rounds as preprocessing, we can view this construction as a NIZK in a certain kind of preprocessing model. However, their protocol has a constant soundness error, and it seems difficult to prove the zero-knowledge property for the parallel repetition version of it.

2 Preliminaries

Notations. We use λ to denote the security parameter throughout the paper. For a positive integer N , $[N]$ means the set $\{1, 2, \dots, N\}$. For a probabilistic classical or quantum algorithm \mathcal{A} , we denote by $y \stackrel{\$}{\leftarrow} \mathcal{A}(x)$ to mean \mathcal{A} runs on input x and outputs y . For a finite set S of classical strings, $x \stackrel{\$}{\leftarrow} S$ means that x is uniformly randomly chosen from S . For a classical string x , x_i denotes the i -th bit of x . For classical strings x and y , $x||y$ denotes the concatenation of x and y . We write poly to mean an unspecified polynomial and negl to mean an unspecified negligible function. We use PPT to stand for (classical) probabilistic polynomial time and QPT to stand for quantum polynomial time. When we say that an algorithm is non-uniform QPT, it is expressed as a family of polynomial size quantum circuits with quantum advice.

2.1 Quantum Computation Preliminaries

Here, we briefly review basic notations and facts on quantum computations.

For any quantum state ρ over registers **A** and **B**, $\text{Tr}_{\mathbf{A}}(\rho)$ is the partial trace of ρ over **A**. We use I to mean the identity operator. (For simplicity, we use the same I for all identity operators with different dimensions, because the dimension of an identity operator is clear from the context.) We

use X , Y , and Z to mean Pauli operators i.e., $X := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $Z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, and $Y := iXZ$.

We use H to mean Hadamard operator, i.e., $H := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. We also define the T operator

by $T := \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$. The $CNOT := |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X$ is the controlled-NOT operator.

We define $V(Z) := I$, $V(X) := H$, and $V(Y) := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix}$ so that for each $W \in \{X, Y, Z\}$, $V(W)|0\rangle$ and $V(W)|1\rangle$ are the eigenvectors of W with eigenvalues $+1$ and -1 , respectively. For each $W \in \{X, Y, Z\}$, we call $\{V(W)|0\rangle, V(W)|1\rangle\}$ the W -basis.

When we consider an N -qubit system, for a Pauli operator $Q \in \{X, Y, Z\}$, Q_j denotes the operator that acts on j -th qubit as Q and trivially acts on all the other qubits. Similarly, $V_j(W)$ denotes the operator that acts on j -th qubit as $V(W)$ and trivially acts on all the other qubits. For any $x \in \{0, 1\}^N$ and $z \in \{0, 1\}^N$, $X^x Z^z$ means $\prod_{j=1}^N X_j^{x_j} Z_j^{z_j}$.

We call the state $\frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle)$ the Bell pair. We call the set $\{|\phi_{x,z}\rangle\}_{(x,z) \in \{0,1\}^2}$ the Bell basis where $|\phi_{x,z}\rangle := (X^x Z^z \otimes I) \frac{|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle}{\sqrt{2}}$. Let us define $U(X) := V(X)$, $U(Y) := V(Y)X$, and $U(Z) := V(Z)$.

Lemma 2.1 (State Collapsing). *If we project one qubit of a Bell pair onto $V(W)|m\rangle$ with $W \in \{X, Y, Z\}$ and $m \in \{0, 1\}$, the other qubit collapses to $U(W)|m\rangle$.*

Lemma 2.2 (Effect of $X^x Z^z$ before measurement). *For any N -qubit state ρ , $(W_1, \dots, W_N) \in \{X, Y, Z\}^N$, and $(x, z) \in \{0, 1\}^N \times \{0, 1\}^N$, the distributions of (m'_1, \dots, m'_n) sampled in the following two ways are identical.*

1. For $j \in [N]$, measure j -th qubit of ρ in W_j basis, let $m_j \in \{0, 1\}$ be the outcome, and set

$$m'_j := \begin{cases} m_j \oplus x_j & (W_j = Z), \\ m_j \oplus z_j & (W_j = X), \\ m_j \oplus x_j \oplus z_j & (W_j = Y). \end{cases}$$

2. For $j \in [N]$, measure j -th qubit of $X^x Z^z \rho Z^z X^x$ in W_j basis and let $m'_j \in \{0, 1\}$ be the outcome.

The proofs of the above lemmas are straightforward.

Lemma 2.3 (Pauli Mixing). *Let ρ be an arbitrary quantum state over registers \mathbf{A} and \mathbf{B} , and let N be the number of qubits in \mathbf{A} . Then we have*

$$\frac{1}{2^{2N}} \sum_{x \in \{0,1\}^N, z \in \{0,1\}^N} (X^x Z^z \otimes I_{\mathbf{B}}) \rho (Z^z X^x \otimes I_{\mathbf{B}}) = \frac{1}{2^N} I_{\mathbf{A}} \otimes \text{Tr}_{\mathbf{A}}(\rho).$$

This is well-known, and one can find a proof in e.g., [Mah18].

Lemma 2.4 (Quantum Teleportation). *Suppose that we have N Bell pairs between registers \mathbf{A} and \mathbf{B} , i.e., $\frac{1}{\sqrt{2}} \sum_{s \in \{0,1\}^N} |s\rangle_{\mathbf{A}} \otimes |s\rangle_{\mathbf{B}}$, and let ρ be an arbitrary N -qubit quantum state in register \mathbf{C} . Suppose that we measure j -th qubits of \mathbf{C} and \mathbf{A} in the Bell basis and let (x_j, z_j) be the measurement outcome for all $j \in [N]$. Let $x := x_1 \| x_2 \| \dots \| x_N$ and $z := z_1 \| z_2 \| \dots \| z_N$. Then the (x, z) is uniformly distributed over $\{0, 1\}^N \times \{0, 1\}^N$. Moreover, conditioned on the measurement outcome (x, z) , the resulting state in \mathbf{B} is $X^x Z^z \rho Z^z X^x$.*

This is also well-known, and one can find a proof in e.g., [NC00].

The following lemma is implicit in previous works e.g., [MNS18, FHM18].

Lemma 2.5. *Let*

$$\mathcal{H} := \frac{I + s(\prod_{j \in S_X} X_j)(\prod_{j \in S_Y} Y_j)(\prod_{j \in S_Z} Z_j)}{2}$$

be an N -qubit projection operator, where $s \in \{+1, -1\}$, and S_X, S_Y , and S_Z are disjoint subsets of $[N]$. For any N -qubit quantum state ρ , suppose that for all $j \in S_W$, where $W \in \{X, Y, Z\}$, we measure j -th qubit of ρ in the W -basis, and let $m_j \in \{0, 1\}$ be the outcome. Then we have

$$\Pr \left[(-1)^{\bigoplus_{j \in S_X \cup S_Y \cup S_Z} m_j} = -s \right] = 1 - \text{Tr}(\rho \mathcal{H}).$$

Proof of Lemma 2.5. Let us define $V := (\prod_{j \in S_X} V_j(X))(\prod_{j \in S_Y} V_j(Y))(\prod_{j \in S_Z} V_j(Z))$, and $|m\rangle :=$

$\bigotimes_{j=1}^N |m_j\rangle$. Then,

$$\begin{aligned}
\Pr \left[(-1)^{\bigoplus_{j \in S_X \cup S_Y \cup S_Z} m_j} = -s \right] &= \sum_{m \in \{0,1\}^N} \langle m | V^\dagger \rho V | m \rangle \frac{1 - s(-1)^{\bigoplus_{j \in S_X \cup S_Y \cup S_Z} m_j}}{2} \\
&= \sum_{m \in \{0,1\}^N} \langle m | V^\dagger \rho V \frac{I - s \prod_{j \in S_X \cup S_Y \cup S_Z} Z_j}{2} | m \rangle \\
&= \text{Tr} \left[V^\dagger \rho V \frac{I - s \prod_{j \in S_X \cup S_Y \cup S_Z} Z_j}{2} \right] \\
&= \text{Tr} \left[\rho V \frac{I - s \prod_{j \in S_X \cup S_Y \cup S_Z} Z_j}{2} V^\dagger \right] \\
&= \text{Tr} \left[\rho (I - \mathcal{H}) \right] \\
&= 1 - \text{Tr}(\rho \mathcal{H}).
\end{aligned}$$

□

2.2 QMA and Local Hamiltonian Problem

Definition 2.6 (QMA). We say that a promise problem $L = (L_{\text{yes}}, L_{\text{no}})$ is in **QMA** if there is a polynomial ℓ and a QPT algorithm V such that the following is satisfied:

- For any $\mathbf{x} \in L_{\text{yes}}$, there exists a quantum state \mathbf{w} of $\ell(|\mathbf{x}|)$ -qubit (called a witness) such that we have $\Pr[V(\mathbf{x}, \mathbf{w}) = 1] \geq 2/3$.
- For any $\mathbf{x} \in L_{\text{no}}$ and any quantum state \mathbf{w} of $\ell(|\mathbf{x}|)$ -qubit, we have $\Pr[V(\mathbf{x}, \mathbf{w}) = 1] \leq 1/3$.

For any $\mathbf{x} \in L$, we denote by $R_L(\mathbf{x})$ to mean the (possibly infinite) set of all quantum states \mathbf{w} such that $\Pr[V(\mathbf{x}, \mathbf{w}) = 1] \geq 2/3$.

Recently, Broadbent and Grilo [BG20] showed that any **QMA** problem can be reduced to a 5-local Hamiltonian problem with *local simulatability*. (See also [GSY19].) Moreover, it is easy to see that we can make the Hamiltonian $\mathcal{H}_{\mathbf{x}}$ be of the form $\mathcal{H}_{\mathbf{x}} = \sum_{i=1}^M p_i \frac{I + s_i P_i}{2}$ where $s_i \in \{+1, -1\}$, $p_i \geq 0$, $\sum_{i=1}^M p_i = 1$, and P_i is a tensor product of Pauli operators (I, X, Z, Y) with at most 5 nontrivial Pauli operators (X, Y, Z) . See Appendix A for more details. Then we have the following lemma.

Lemma 2.7 (QMA-completeness of 5-local Hamiltonian problem with local simulatability [BG20]). For any **QMA** promise problem $L = (L_{\text{yes}}, L_{\text{no}})$, there is a classical polynomial-time computable deterministic function that maps $\mathbf{x} \in \{0, 1\}^*$ to an N -qubit Hamiltonian $\mathcal{H}_{\mathbf{x}}$ of the form $\mathcal{H}_{\mathbf{x}} = \sum_{i=1}^M p_i \frac{I + s_i P_i}{2}$, where $N = \text{poly}(|\mathbf{x}|)$, $M = \text{poly}(|\mathbf{x}|)$, $s_i \in \{+1, -1\}$, $p_i > 0$, $\sum_{i=1}^M p_i = 1$, and P_i is a tensor product of Pauli operators (I, X, Y, Z) with at most 5 nontrivial Pauli operators (X, Y, Z) , and satisfies the following: There are $0 < \alpha < \beta < 1$ such that $\beta - \alpha = 1/\text{poly}(|\mathbf{x}|)$ and

- if $\mathbf{x} \in L_{\text{yes}}$, then there exists an N -qubit state ρ such that $\text{Tr}(\rho \mathcal{H}_{\mathbf{x}}) \leq \alpha$, and
- if $\mathbf{x} \in L_{\text{no}}$, then for any N -qubit state ρ , we have $\text{Tr}(\rho \mathcal{H}_{\mathbf{x}}) \geq \beta$.

Moreover, for any $\mathbf{x} \in L_{\text{yes}}$, we can convert any witness $\mathbf{w} \in R_L(\mathbf{x})$ into a state ρ_{hist} , called the *history state*, such that $\text{Tr}(\rho_{\text{hist}} \mathcal{H}_{\mathbf{x}}) \leq \alpha$ in quantum polynomial time. Moreover, there exists a classical deterministic polynomial time algorithm Sim_{hist} such that for any $\mathbf{x} \in L_{\text{yes}}$ and any subset

$S \subseteq [N]$ with $|S| \leq 5$, $\text{Sim}_{\text{hist}}(\mathbf{x}, S)$ outputs a classical description of an $|S|$ -qubit density matrix ρ_S such that $\|\rho_S - \text{Tr}_{[N] \setminus S} \rho_{\text{hist}}\|_{tr} = \text{negl}(\lambda)$ where $\text{Tr}_{[N] \setminus S} \rho_{\text{hist}}$ is the state of ρ_{hist} in registers corresponding to S tracing out all other registers.

2.3 Classically-Verifiable Non-Interactive Zero-knowledge Proofs

Definition 2.8 (CV-NIZK in the QSP model). *A classically-verifiable non-interactive zero-knowledge proof (CV-NIZK) for a QMA promise problem $L = (L_{\text{yes}}, L_{\text{no}})$ in the quantum secret parameter (QSP) model consists of algorithms $\Pi = (\text{Setup}, \text{Prove}, \text{Verify})$ with the following syntax:*

Setup(1^λ): *This is a QPT algorithm that takes the security parameter 1^λ as input and outputs a quantum proving key k_P and a classical verification key k_V .*

Prove($k_P, \mathbf{x}, \mathbf{w}^{\otimes k}$): *This is a QPT algorithm that takes the proving key k_P , a statement \mathbf{x} , and $k = \text{poly}(\lambda)$ copies $\mathbf{w}^{\otimes k}$ of a witness $\mathbf{w} \in R_L(\mathbf{x})$ as input and outputs a classical proof π .*

Verify(k_V, \mathbf{x}, π): *This is a PPT algorithm that takes the verification key k_V , a statement \mathbf{x} , and a proof π as input and outputs \top indicating acceptance or \perp indicating rejection.*

We require Π to satisfy the following properties for some $0 < s < c < 1$ such that $c - s > 1/\text{poly}(\lambda)$. Especially, when we do not specify c and s , they are set as $c = 1 - \text{negl}(\lambda)$ and $s = \text{negl}(\lambda)$.

c -Completeness. *For all $\mathbf{x} \in L_{\text{yes}} \cap \{0, 1\}^\lambda$, and $\mathbf{w} \in R_L(\mathbf{x})$, we have*

$$\Pr \left[\text{Verify}(k_V, \mathbf{x}, \pi) = \top : (k_P, k_V) \stackrel{\$}{\leftarrow} \text{Setup}(1^\lambda), \pi \stackrel{\$}{\leftarrow} \text{Prove}(k_P, \mathbf{x}, \mathbf{w}^{\otimes k}) \right] \geq c.$$

(Adaptive Statistical) s -Soundness. *For all unbounded-time adversary \mathcal{A} , we have*

$$\Pr \left[\mathbf{x} \in L_{\text{no}} \wedge \text{Verify}(k_V, \mathbf{x}, \pi) = \top : (k_P, k_V) \stackrel{\$}{\leftarrow} \text{Setup}(1^\lambda), (\mathbf{x}, \pi) \stackrel{\$}{\leftarrow} \mathcal{A}(k_P) \right] \leq s.$$

(Adaptive Statistical Single-Theorem) Zero-Knowledge. *There exists a PPT simulator Sim such that for any unbounded-time distinguisher \mathcal{D} , we have*

$$\left| \Pr \left[\mathcal{D}^{\mathcal{O}_P(k_P, \cdot, \cdot)}(k_V) = 1 \right] - \Pr \left[\mathcal{D}^{\mathcal{O}_S(k_V, \cdot, \cdot)}(k_V) = 1 \right] \right| = \text{negl}(\lambda)$$

where $(k_P, k_V) \stackrel{\$}{\leftarrow} \text{Setup}(1^\lambda)$, \mathcal{D} can make at most one query, which should be of the form $(\mathbf{x}, \mathbf{w}^{\otimes k})$ where $\mathbf{w} \in R_L(\mathbf{x})$ and $\mathbf{w}^{\otimes k}$ is unentangled with \mathcal{D} 's internal registers,⁶ $\mathcal{O}_P(k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$ returns $\text{Prove}(k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$, and $\mathcal{O}_S(k_V, \mathbf{x}, \mathbf{w}^{\otimes k})$ returns $\text{Sim}(k_V, \mathbf{x})$.

It is easy to see that we can amplify the gap between completeness and soundness thresholds by a simple parallel repetition. Moreover, we can see that this does not lose the zero-knowledge property. Therefore, we have the following lemma.

Lemma 2.9 (Gap Amplification for CV-NIZK). *If there exists a CV-NIZK for L in the QSP model that satisfies c -completeness and s -soundness, for some $0 < s < c < 1$ such that $c - s > 1/\text{poly}(\lambda)$, then there exists a CV-NIZK for L in the QSP model (with $(1 - \text{negl}(\lambda))$ -completeness and $\text{negl}(\lambda)$ -soundness).*

⁶Though our protocols are likely to remain secure even if they can be entangled, we assume that they are unentangled for simplicity. To the best of our knowledge, none of existing works on interactive or non-interactive zero-knowledge for QMA [BJSW20, CVZ20, BS20, BG20, Shm21, BCKM21] considered entanglement between a witness and distinguisher's internal register.

Proof. Let $\Pi = (\text{Setup}, \text{Prove}, \text{Verify})$ be a CV-NIZK for L in the SP model that satisfies c -completeness, s -soundness, and the zero-knowledge property for some $0 < s < c < 1$ such that $c - s > 1/\text{poly}(\lambda)$. Let k be the number of copies of a witness Prove takes as input. For any polynomial $N = \text{poly}(\lambda)$, $\Pi^N = (\text{Setup}^N, \text{Prove}^N, \text{Verify}^N)$ be the N -parallel version of Π . That is, Setup^N and Prove^N run Setup and Prove N times parallelly and outputs tuples consisting of outputs of each execution, respectively where Prove^N takes Nk copies of the witness as input. Verify^N takes N -tuple of the verification key and proof, runs Verify to verify each of them separately, and outputs \top if the number of executions of Verify that outputs \top is larger than $\frac{N(\alpha+\beta)}{2}$. By Hoeffding's inequality, it is easy to see that we can take $N = O\left(\frac{\log^2 \lambda}{(\alpha-\beta)^2}\right)$ so that Π^N satisfies $(1 - \text{negl}(\lambda))$ -completeness and $\text{negl}(\lambda)$ -soundness.

What is left is to prove that Π^N satisfies the zero-knowledge property. This can be reduced to the zero-knowledge property of Π by a standard hybrid argument. More precisely, for each $i \in \{0, \dots, N\}$, let \mathcal{O}_i be the oracle that works as follows where k'_P and k'_V denote the proving and verification keys of Π^N , respectively.

$\mathcal{O}_i(k'_P = (k_P^1, \dots, k_P^N), k'_V = (k_V^1, \dots, k_V^N), \mathbf{x}, \mathbf{w}^{\otimes Nk})$: It works as follows:

- For $1 \leq j \leq i$, it computes $\pi_j \stackrel{\$}{\leftarrow} \text{Sim}(k_V^j, \mathbf{x})$.
- For $i < j \leq N$, it computes $\pi_j \stackrel{\$}{\leftarrow} \text{Prove}(k_P^j, \mathbf{x}, \mathbf{w}^{\otimes k})$ where it uses the $(k(j-1) + 1)$ -th to kj -th copies of \mathbf{w} .
- Output $\pi := (\pi_1, \dots, \pi_N)$.

Clearly, we have $\mathcal{O}_0(k'_P, k'_V, \cdot, \cdot) = \mathcal{O}_P(k'_P, \cdot, \cdot)$ and $\mathcal{O}_N(k'_P, k'_V, \cdot, \cdot) = \mathcal{O}_S(k'_V, \cdot, \cdot)$.⁷ Therefore, it suffices to prove that no distinguisher can distinguish $\mathcal{O}_i(k'_P, k'_V, \cdot, \cdot)$ and $\mathcal{O}_{i+1}(k'_P, k'_V, \cdot, \cdot)$ for any $i \in \{0, 1, \dots, N-1\}$. For the sake of contradiction, suppose that there exists a distinguisher \mathcal{D}' that distinguishes $\mathcal{O}_i(k'_P, k'_V, \cdot, \cdot)$ and $\mathcal{O}_{i+1}(k'_P, k'_V, \cdot, \cdot)$ with a non-negligible advantage by making one query of the form $(\mathbf{x}, \mathbf{w}^{\otimes Nk})$. Then we construct a distinguisher \mathcal{D} that breaks the zero-knowledge property of Π as follows:

$\mathcal{D}^{\mathcal{O}}(k_V)$: \mathcal{D} takes k_V as input and is given a single oracle access to \mathcal{O} , which is either $\mathcal{O}_P(k_P, \cdot, \cdot)$ or $\mathcal{O}_S(k_V, \cdot, \cdot)$ where k_P is the proving key corresponding to k_V .⁸ (Remark that \mathcal{D} is not given k_P .) It sets $k_V^{i+1} := k_V$ (which implicitly defines $k_P^{i+1} := k_P$) and generates $(k_P^j, k_V^j) \stackrel{\$}{\leftarrow} \text{Setup}(1^\lambda)$ for all $j \in [N] \setminus \{i+1\}$. It sets $k'_V := (k_V^1, \dots, k_V^N)$ and runs $\mathcal{D}'^{\mathcal{O}'}(k'_V)$ where when \mathcal{D}' makes a query $(\mathbf{x}, \mathbf{w}^{\otimes Nk})$ to \mathcal{O}' , \mathcal{D} simulates the oracle \mathcal{O}' for \mathcal{D}' as follows:

- For $1 \leq j \leq i$, \mathcal{D} computes $\pi_j \stackrel{\$}{\leftarrow} \text{Sim}(k_V^j, \mathbf{x})$.
- For $j = i+1$, \mathcal{D} queries $(\mathbf{x}, \mathbf{w}^{\otimes k})$ to the external oracle \mathcal{O} where it uses the $(ki+1)$ -th to $k(i+1)$ -th copies of \mathbf{w} as part of its query, and lets π_{i+1} be the oracle's response.
- For $i+1 < j \leq N$, it computes $\pi_j \stackrel{\$}{\leftarrow} \text{Prove}(k_P^j, \mathbf{x}, \mathbf{w}^{\otimes k})$ where it uses the $(k(j-1) + 1)$ -th to kj -th copies of \mathbf{w} . We note that this can be simulated by \mathcal{D} since it knows k_P^j for $j \neq i+1$.
- \mathcal{D} returns $\pi' := (\pi_1, \dots, \pi_N)$ to \mathcal{D}' as a response from the oracle \mathcal{O}' .

Finally, when \mathcal{D}' outputs b , \mathcal{D} also outputs b .

⁷ $\mathcal{O}_P(k'_P, \cdot, \cdot)$ and $\mathcal{O}_S(k'_V, \cdot, \cdot)$ mean the corresponding oracles for Π^N .

⁸ $\mathcal{O}_P(k_P, \cdot, \cdot)$ and $\mathcal{O}_S(k_V, \cdot, \cdot)$ mean the corresponding oracles for Π by abuse of notation.

We can see that the oracle \mathcal{O}' simulated by \mathcal{D} works similarly to $\mathcal{O}_i(k'_P, k'_V, \cdot, \cdot)$ when \mathcal{O} is $\mathcal{O}_P(k_P, \cdot, \cdot)$ and works similarly to $\mathcal{O}_{i+1}(k'_P, k'_V, \cdot, \cdot)$ when \mathcal{O} is $\mathcal{O}_S(k_V, \cdot, \cdot)$ where $k'_P = (k_P^1, \dots, k_P^N)$. Therefore, by the assumption that \mathcal{D}' distinguishes $\mathcal{O}_i(k'_P, k'_V, \cdot, \cdot)$ and $\mathcal{O}_{i+1}(k'_P, k'_V, \cdot, \cdot)$ with a non-negligible advantage, \mathcal{D} distinguishes $\mathcal{O}_P(k_P, \cdot, \cdot)$ and $\mathcal{O}_S(k_V, \cdot, \cdot)$ with a non-negligible advantage. However, this contradicts the zero-knowledge property of Π . Therefore, such \mathcal{D}' does not exist, which completes the proof of Lemma 2.9. \square

3 CV-NIZK in the QSP model

In this section, we construct a CV-NIZK in the QSP model (Definition 2.8). Specifically, we prove the following theorem.

Theorem 3.1. *There exists a CV-NIZK for QMA in the QSP model (without any computational assumption).*

Our construction of a CV-NIZK for a QMA promise problem L is given in Figure 1 where \mathcal{H}_x , N , M , p_i , s_i , P_i , α , β , and ρ_{hist} are as in Lemma 2.7 for L and $V_j(W_j)$ is as defined in Section 2.1.

We note that there is a slightly simpler construction of CV-NIZK as shown in Figure 8 in Appendix C. However, we consider the construction given in Figure 1 as our main construction since this is more convenient to extend to the computationally secure construction given in Section 4.

Moreover, if we require only the completeness and the soundness, there is a much simpler construction. For details, see Appendix D.

To show Theorem 3.1, we prove the following lemmas.

Lemma 3.2 (Completeness and Soundness). Π_{NIZK} satisfies $(1 - \frac{\alpha}{N'})$ -completeness and $(1 - \frac{\beta}{N'})$ -soundness where $N' := 3^5 \sum_{i=1}^5 \binom{N}{i}$.

Lemma 3.3 (Zero-Knowledge). Π_{NIZK} satisfies the zero-knowledge property.

Since $(1 - \frac{\alpha}{N'}) - (1 - \frac{\beta}{N'}) = \frac{\beta - \alpha}{N'} \geq 1/\text{poly}(\lambda)$, by combining Lemmas 2.9, 3.2 and 3.3, Theorem 3.1 follows.

In the following, we give proofs of Lemmas 3.2 and 3.3.

Proof of Lemma 3.2. We prove this lemma by considering virtual protocols that do not change completeness and soundness. For more details, see Appendix B. First, we consider the virtual protocol 1 described in Figure 2. There are two differences from the original protocol. The first is that k_V includes the whole (\hat{x}, \hat{z}) instead of $\{\hat{x}_j, \hat{z}_j\}_{j \in S_V}$. This difference does not change the (possibly malicious) prover's view since k_V is not given to the prover. The second is that the setup algorithm generates N Bell pairs and gives each halves to the prover and verifier, and the verifier obtains (m_1, \dots, m_N) by measuring his halves in Pauli basis. Because the verifier's measurement and the prover's measurement commute with each other, in the virtual protocol 1, the verifier's acceptance probability does not change even if the verifier chooses (W_1, \dots, W_N) and measures ρ_V in the corresponding basis to obtain outcomes (m_1, \dots, m_N) before ρ_P is given to the prover. Moreover, conditioned on the above measurement outcomes, the state in \mathbf{P} collapses to $\bigotimes_{j=1}^N (U(W_j)|m_j\rangle)$ (See Lemma 2.1). Therefore, the virtual protocol 1 is exactly the same as the original protocol from the prover's view, and the verifier's acceptance probability of the virtual protocol 1 is the same as that of the original protocol Π_{NIZK} for any possibly malicious prover.

Next, we further modify the protocol to define the virtual protocol 2 described in Figure 3. The difference from the virtual protocol 1 is that instead of setting m'_j , the verification algorithm applies

Setup(1^λ): The setup algorithm chooses $(W_1, \dots, W_N) \xleftarrow{\$} \{X, Y, Z\}^N$, $(m_1, \dots, m_N) \xleftarrow{\$} \{0, 1\}^N$, $(\hat{x}, \hat{z}) \xleftarrow{\$} \{0, 1\}^N \times \{0, 1\}^N$, and a uniformly random subset $S_V \subseteq [N]$ such that $1 \leq |S_V| \leq 5$, and outputs a proving key $k_P := (\rho_P := \bigotimes_{j=1}^N (U(W_j)|m_j\rangle), \hat{x}, \hat{z})$ and a verification key $k_V := (W_1, \dots, W_N, m_1, \dots, m_N, S_V, \{\hat{x}_j, \hat{z}_j\}_{j \in S_V})$.

Prove($k_P, \mathbf{x}, \mathbf{w}$): The proving algorithm parses $(\rho_P, \hat{x}, \hat{z}) \leftarrow k_P$, generates the history state ρ_{hist} for \mathcal{H}_x from \mathbf{w} , and computes $\rho'_{\text{hist}} := X^{\hat{x}} Z^{\hat{z}} \rho_{\text{hist}} Z^{\hat{z}} X^{\hat{x}}$. It measures j -th qubits of ρ'_{hist} and ρ_P in the Bell basis for $j \in [N]$. Let $x := x_1 \| x_2 \| \dots \| x_N$, and $z := z_1 \| z_2 \| \dots \| z_N$ where $(x_j, z_j) \in \{0, 1\}^2$ denotes the outcome of j -th measurement. It outputs a proof $\pi := (x, z)$.

Verify(k_V, \mathbf{x}, π): The verification algorithm parses $(W_1, \dots, W_N, m_1, \dots, m_N, S_V, \{\hat{x}_j, \hat{z}_j\}_{j \in S_V}) \leftarrow k_V$ and $(x, z) \leftarrow \pi$, chooses $i \in [M]$ according to the probability distribution defined by $\{p_i\}_{i \in [M]}$ (i.e., chooses i with probability p_i). Let

$$S_i := \{j \in [N] \mid j\text{th Pauli operator of } P_i \text{ is not } I\}.$$

We note that we have $1 \leq |S_i| \leq 5$ by the 5-locality of \mathcal{H}_x . We say that P_i is consistent to $(S_V, \{W_j\}_{j \in S_V})$ if and only if $S_i = S_V$ and the j th Pauli operator of P_i is W_j for all $j \in S_i$. If P_i is not consistent to $(S_V, \{W_j\}_{j \in S_V})$, it outputs \perp . If P_i is consistent to $(S_V, \{W_j\}_{j \in S_V})$, it flips a biased coin that heads with probability $1 - 3^{|S_i| - 5}$. If heads, it outputs \top . If tails, it defines

$$m'_j := \begin{cases} m_j \oplus x_j \oplus \hat{x}_j & (W_j = Z), \\ m_j \oplus z_j \oplus \hat{z}_j & (W_j = X), \\ m_j \oplus x_j \oplus \hat{x}_j \oplus z_j \oplus \hat{z}_j & (W_j = Y) \end{cases}$$

for $j \in S_i$, and outputs \top if $(-1)^{\bigoplus_{j \in S_i} m'_j} = -s_i$ and \perp otherwise.

Figure 1: CV-NIZK Π_{NIZK} in the QSP model.

a corresponding Pauli $X^{x \oplus \hat{x}} Z^{z \oplus \hat{z}}$ on ρ_V , and then measures it to obtain m'_j . By Lemma 2.2, this does not change the distribution of (m'_1, \dots, m'_N) . Therefore, the verifier's acceptance probability of the virtual protocol 2 is the same as that of the virtual protocol 1 for any possibly malicious prover.

Therefore, it suffices to prove $(1 - \frac{\alpha}{N^r})$ -completeness and $(1 - \frac{\beta}{N^r})$ -soundness for the virtual protocol 2. When $\mathbf{x} \in L_{\text{yes}}$ and π is honestly generated, then ρ'_V is the history state ρ_{hist} , which satisfies $\text{Tr}(\rho_{\text{hist}} \mathcal{H}_x) \leq \alpha$, by the correctness of quantum teleportation (Lemma 2.4). For any fixed P_i , the probability that P_i is consistent to $(S_V, \{W_j\}_{j \in S_V})$ and the coin tails is $\frac{1}{N^r}$. Therefore, by Lemma 2.5 and Lemma 2.7, the verifier's acceptance probability is $1 - \frac{1}{N^r} \text{Tr}(\rho_{\text{hist}} \mathcal{H}_x) \geq 1 - \frac{\alpha}{N^r}$.

Let \mathcal{A} be an adaptive adversary against soundness of virtual protocol 2. That is, \mathcal{A} is given k_P and outputs (\mathbf{x}, π) . We say that \mathcal{A} wins if $\mathbf{x} \in L_{\text{no}}$ and $\text{Verify}(k_V, \mathbf{x}, \pi) = \top$. For any \mathbf{x} , let E_x be the event that the statement output by \mathcal{A} is \mathbf{x} , and $\rho'_{V, \mathbf{x}}$ be the state in \mathbf{V} right before the measurement by Verify conditioned on E_x . Similarly to the analysis for the completeness, by Lemma 2.5 and

Setup_{vir-1}(1^λ): The setup algorithm generates N Bell-pairs between registers \mathbf{P} and \mathbf{V} and lets ρ_P and ρ_V be quantum states in registers \mathbf{P} and \mathbf{V} , respectively. It chooses $(\hat{x}, \hat{z}) \stackrel{\$}{\leftarrow} \{0, 1\}^N \times \{0, 1\}^N$. It chooses a uniformly random subset $S_V \subseteq [N]$ such that $1 \leq |S_V| \leq 5$, and outputs a proving key $k_P := (\rho_P, \hat{x}, \hat{z})$ and a verification key $k_V := (\rho_V, S_V, \hat{x}, \hat{z})$.

Prove_{vir-1}($k_P, \mathbf{x}, \mathbf{w}$): This is the same as **Prove**($k_P, \mathbf{x}, \mathbf{w}$) in Figure 1.

Verify_{vir-1}(k_V, \mathbf{x}, π): The verification algorithm chooses $(W_1, \dots, W_N) \stackrel{\$}{\leftarrow} \{X, Y, Z\}^N$, and measures j -th qubit of ρ_V in the W_j basis for all $j \in [N]$, and lets (m_1, \dots, m_N) be the measurement outcomes. The rest of this algorithm is the same as **Verify**(k_V, \mathbf{x}, π) given in Figure 1.

Figure 2: The virtual protocol 1 for Π_{NIZK}

Setup_{vir-2}(1^λ): This is the same as **Setup_{vir-1}**(1^λ) in Figure 2.

Prove_{vir-2}($k_P, \mathbf{x}, \mathbf{w}$): This is the same as **Prove**($k_P, \mathbf{x}, \mathbf{w}$) in Figure 1.

Verify_{vir-2}(k_V, \mathbf{x}, π): The verification algorithm parses $(\rho_V, S_V, \hat{x}, \hat{z}) \leftarrow k_V$ and $(x, z) \leftarrow \pi$, computes $\rho'_V := X^{x \oplus \hat{x}} Z^{z \oplus \hat{z}} \rho_V Z^{z \oplus \hat{z}} X^{x \oplus \hat{x}}$, chooses $(W_1, \dots, W_N) \stackrel{\$}{\leftarrow} \{X, Y, Z\}^N$, measures j -th qubit of ρ'_V in the W_j basis for all $j \in [N]$, and lets (m'_1, \dots, m'_N) be the measurement outcomes.

It chooses $i \in [M]$ and defines $S_i \subseteq [N]$ similarly to **Verify**(k_V, \mathbf{x}, π) in Figure 1. If P_i is not consistent to $(S_V, \{W_j\}_{j \in S_V})$, it outputs \perp . If P_i is consistent to $(S_V, \{W_j\}_{j \in S_V})$, it flips a biased coin that heads with probability $1 - 3^{|S_i| - 5}$. If heads, it outputs \top . If tails, it outputs \top if $(-1)^{\bigoplus_{j \in S_i} m'_j} = -s_i$ and \perp otherwise.

Figure 3: The virtual protocol 2 for Π_{NIZK}

Lemma 2.7, we have

$$\Pr[\mathcal{A} \text{ wins}] = \sum_{\mathbf{x} \in L_{\text{no}}} \Pr[\mathbf{E}_{\mathbf{x}}] \left(1 - \frac{1}{N'} \text{Tr}(\rho'_{V, \mathbf{x}} \mathcal{H}_{\mathbf{x}}) \right) \leq \sum_{\mathbf{x} \in L_{\text{no}}} \Pr[\mathbf{E}_{\mathbf{x}}] \left(1 - \frac{\beta}{N'} \right) \leq 1 - \frac{\beta}{N'}.$$

□

Proof of Lemma 3.3. We describe the simulator **Sim** below.

Sim(k_V, \mathbf{x}): The simulator parses $(W_1, \dots, W_N, m_1, \dots, m_N, S_V, \{\hat{x}_j, \hat{z}_j\}_{j \in S_V}) \leftarrow k_V$ and does the following.

1. Generate the classical description of the density matrix $\rho_{S_V} := \text{Sim}_{\text{hist}}(\mathbf{x}, S_V)$ where Sim_{hist} is as in Lemma 2.7.
2. Sample $\{x_j, z_j\}_{j \in S_V}$ according to the probability distribution of outcomes of the Bell-basis measurements of the corresponding pairs of qubits of $\left(\prod_{j \in S_V} X_j^{\hat{x}_j} Z_j^{\hat{z}_j} \right) \rho_{S_V} \left(\prod_{j \in S_V} Z_j^{\hat{z}_j} X_j^{\hat{x}_j} \right)$

and $\bigotimes_{j \in S_V} (U(W_j) |m_j\rangle)$. We emphasize that this measurement can be simulated in a classical probabilistic polynomial time since $|S_V| \leq 5$.

3. Choose $(x_j, z_j) \stackrel{\$}{\leftarrow} \{0, 1\}^2$ for all $j \in [N] \setminus S_V$.
4. Output $\pi := (x, z)$ where $x := x_1 \| x_2 \| \dots \| x_N$ and $z := z_1 \| z_2 \| \dots \| z_N$.

We prove that the output of this simulator is indistinguishable from the real proof. For proving this, we consider the following sequences of modified simulators. We note that these simulators may perform quantum computations unlike the real simulator.

Sim₁(k_V, \mathbf{x}): The simulator parses $(W_1, \dots, W_N, m_1, \dots, m_N, S_V, \{\hat{x}_j, \hat{z}_j\}_{j \in S_V}) \leftarrow k_V$ and does the following.

1. Generate the classical description of the density matrix $\rho_{S_V} := \text{Sim}_{\text{hist}}(\mathbf{x}, S_V)$ where Sim_{hist} is as in Lemma 2.7. (This step is the same as the step 1 of $\text{Sim}(k_V, \mathbf{x})$.)
2. Generate $\tilde{\rho}'_{\text{hist}} := \left(\prod_{j \in S_V} X_j^{\hat{x}_j} Z_j^{\hat{z}_j} \right) \rho_{S_V} \left(\prod_{j \in S_V} Z_j^{\hat{z}_j} X_j^{\hat{x}_j} \right) \otimes \frac{I_{[N] \setminus S_V}}{2^{|[N] \setminus S_V|}}$.
3. Measure j -th qubits of $\tilde{\rho}'_{\text{hist}}$ and $\rho_P := \bigotimes_{j=1}^N (U(W_j) |m_j\rangle)$ in the Bell basis for $j \in [N]$, and let (x_j, z_j) be the j -th measurement result.
4. Output $\pi := (x, z)$ where $x := x_1 \| x_2 \| \dots \| x_N$ and $z := z_1 \| z_2 \| \dots \| z_N$.

Clearly, the distributions of $\{x_j, z_j\}_{j \in S_V}$ output by $\text{Sim}(k_V, \mathbf{x})$ and $\text{Sim}_1(k_V, \mathbf{x})$ are the same. Moreover, the distributions of $\{x_j, z_j\}_{j \in [N] \setminus S_V}$ output by $\text{Sim}(k_V, \mathbf{x})$ and $\text{Sim}_1(k_V, \mathbf{x})$ are both uniformly and independently random. Therefore, output distributions of $\text{Sim}(k_V, \mathbf{x})$ and $\text{Sim}_1(k_V, \mathbf{x})$ are exactly the same.

Next, we consider the following modified simulator that takes a witness $\mathbf{w} \in R_L(\mathbf{x})$ as input.

Sim₂($k_V, \mathbf{x}, \mathbf{w}$): The simulator parses $(W_1, \dots, W_N, m_1, \dots, m_N, S_V, \{\hat{x}_j, \hat{z}_j\}_{j \in S_V}) \leftarrow k_V$ and does the following.

1. Generate the history state ρ_{hist} for $\mathcal{H}_{\mathbf{x}}$ from \mathbf{w} .
2. Generate $(\hat{x}_j, \hat{z}_j) \stackrel{\$}{\leftarrow} \{0, 1\}^2$ for $j \in [N] \setminus S_V$ and let $\hat{x} := \hat{x}_1 \| \dots \| \hat{x}_N$ and $\hat{z} := \hat{z}_1 \| \dots \| \hat{z}_N$.
3. Compute $\rho'_{\text{hist}} := X^{\hat{x}} Z^{\hat{z}} \rho_{\text{hist}} Z^{\hat{z}} X^{\hat{x}}$.
4. Measure j -th qubits of ρ'_{hist} and $\rho_P := \bigotimes_{j=1}^N (U(W_j) |m_j\rangle)$ in the Bell basis for $j \in [N]$, and let (x_j, z_j) be the j -th measurement result.
5. Output $\pi := (x, z)$ where $x := x_1 \| x_2 \| \dots \| x_N$ and $z := z_1 \| z_2 \| \dots \| z_N$.

By Lemma 2.3, we have $\rho'_{\text{hist}} = \left(\prod_{j \in S_V} X_j^{\hat{x}_j} Z_j^{\hat{z}_j} \right) \text{Tr}_{N \setminus S_V}[\rho_{\text{hist}}] \left(\prod_{j \in S_V} Z_j^{\hat{z}_j} X_j^{\hat{x}_j} \right) \otimes \frac{I_{[N] \setminus S_V}}{2^{|[N] \setminus S_V|}}$ from the view of a distinguisher that has no information on $\{\hat{x}_j, \hat{z}_j\}_{j \in [N] \setminus S_V}$. By Lemma 2.7, we have $\|\rho_{S_V} - \text{Tr}_{[N] \setminus S_V} \rho_{\text{hist}}\|_{tr} = \text{negl}(\lambda)$. Therefore, we have $\|\tilde{\rho}'_{\text{hist}} - \rho'_{\text{hist}}\|_{tr} = \text{negl}(\lambda)$. This means that $\text{Sim}_1(k_V, \mathbf{x})$ and $\text{Sim}_2(k_V, \mathbf{x}, \mathbf{w})$ are statistically indistinguishable from the view of a distinguisher that makes at most one query.

Finally, noting that the output distribution of $\text{Sim}_2(k_V, \mathbf{x}, \mathbf{w})$ is exactly the same as that of $\text{Prove}(k_P, \mathbf{x}, \mathbf{w})$, the proof of Lemma 3.3 is completed. □

4 Dual-Mode CV-NIZK with Preprocessing

In this section, we extend the CV-NIZK given in Section 3 to reduce the amount of trust in the setup at the cost of introducing a quantum preprocessing and relying on a computational assumption. In the construction in Section 3, we assume that the trusted setup algorithm honestly generates proving and verification keys, which are correlated with each other, and sends them to the prover and verifier, respectively, without revealing them to the other party. Here, we give a construction of CV-NIZK with preprocessing that consists of the generation of common reference string by a trusted party and a single instance-independent quantum message from the verifier to the prover. We call such a model the $\text{CRS} + (V \rightarrow P)$ model. We note this is the same model as is considered in [CVZ20]. Moreover, our construction has a nice feature called the dual-mode property, which has been considered for NIZKs for NP [GS12, GOS12, PS19]. The dual-mode property requires that there are two computationally indistinguishable modes of generating a common reference string, one of which ensures statistical soundness (and computational zero-knowledge) while the other ensures statistical zero-knowledge (and computational soundness). To the best of our knowledge, ours is the first construction of a dual-mode NIZK for QMA in any kind of model.

4.1 Definition

We give a formal definition of a dual-mode CV-NIZK in the $\text{CRS} + (V \rightarrow P)$ model.

Definition 4.1 (Dual-Mode CV-NIZK in the $\text{CRS} + (V \rightarrow P)$ Model). *A dual-mode CV-NIZK for a QMA promise problem $L = (L_{\text{yes}}, L_{\text{no}})$ in the $\text{CRS} + (V \rightarrow P)$ model consists of algorithms $\Pi = (\text{CRSGen}, \text{Preprocess}, \text{Prove}, \text{Verify})$ with the following syntax:*

$\text{CRSGen}(1^\lambda, \text{mode})$: *This is a PPT algorithm that takes the security parameter 1^λ and a mode $\text{mode} \in \{\text{binding}, \text{hiding}\}$ as input and outputs a classical common reference string crs . We note that crs can be reused and thus this algorithm is only needed to run once by a trusted third party.*

$\text{Preprocess}(\text{crs})$: *This is a QPT algorithm that takes the common reference string crs as input and outputs a quantum proving key k_P and a classical verification key k_V . We note that this algorithm is supposed to be run by the verifier as preprocessing, and k_P is supposed to be sent to the prover while k_V is supposed to be kept on verifier's side in secret. We also note that they can be used only once and cannot be reused unlike crs .*

$\text{Prove}(\text{crs}, k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$: *This is a QPT algorithm that takes the common reference string crs , the proving key k_P , a statement \mathbf{x} , and $k = \text{poly}(\lambda)$ copies $\mathbf{w}^{\otimes k}$ of a witness $\mathbf{w} \in R_L(\mathbf{x})$ as input and outputs a classical proof π .*

$\text{Verify}(\text{crs}, k_V, \mathbf{x}, \pi)$: *This is a PPT algorithm that takes the common reference string crs , the verification key k_V , a statement \mathbf{x} , and a proof π as input and outputs \top indicating acceptance or \perp indicating rejection.*

We require Π to satisfy the following properties for some $0 < s < c < 1$ such that $c - s > 1/\text{poly}(\lambda)$. Especially, when we do not specify c and s , they are set as $c = 1 - \text{negl}(\lambda)$ and $s = \text{negl}(\lambda)$.

c-Completeness. *For all $\text{mode} \in \{\text{binding}, \text{hiding}\}$, $\mathbf{x} \in L_{\text{yes}} \cap \{0, 1\}^\lambda$, and $\mathbf{w} \in R_L(\mathbf{x})$, we have*

$$\Pr \left[\begin{array}{l} \text{Verify}(\text{crs}, k_V, \mathbf{x}, \pi) = \top : \\ \text{crs} \xleftarrow{s} \text{CRSGen}(1^\lambda, \text{mode}) \\ (k_P, k_V) \xleftarrow{s} \text{Preprocess}(\text{crs}) \\ \pi \xleftarrow{s} \text{Prove}(\text{crs}, k_P, \mathbf{x}, \mathbf{w}^{\otimes k}) \end{array} \right] \geq c.$$

(Adaptive) Statistical s -Soundness in the Binding Mode For all unbounded-time adversary \mathcal{A} , we have

$$\Pr \left[\begin{array}{l} \mathbf{x} \in L_{\text{no}} \wedge \text{Verify}(\text{crs}, k_V, \mathbf{x}, \pi) = \top : \\ \text{crs} \stackrel{\$}{\leftarrow} \text{CRSGen}(1^\lambda, \text{binding}) \\ (k_P, k_V) \stackrel{\$}{\leftarrow} \text{Preprocess}(\text{crs}) \\ (\mathbf{x}, \pi) \stackrel{\$}{\leftarrow} \mathcal{A}(\text{crs}, k_P) \end{array} \right] \leq s.$$

(Adaptive Multi-Theorem) Statistical Zero-Knowledge in the Hiding Mode. There exists a PPT simulator Sim_0 and a QPT simulator Sim_1 such that for any unbounded-time distinguisher \mathcal{D} , we have

$$\left| \Pr \left[\mathcal{D}^{\mathcal{O}_P(\text{crs}, \cdot, \cdot)}(\text{crs}) = 1 : \text{crs} \stackrel{\$}{\leftarrow} \text{CRSGen}(1^\lambda, \text{hiding}) \right] - \Pr \left[\mathcal{D}^{\mathcal{O}_S(\text{td}, \cdot, \cdot)}(\text{crs}) = 1 : (\text{crs}, \text{td}) \stackrel{\$}{\leftarrow} \text{Sim}_0(1^\lambda) \right] \right| \leq \text{negl}(\lambda)$$

where \mathcal{D} can make $\text{poly}(\lambda)$ queries, which should be of the form $(k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$ where $\mathbf{w} \in R_L(\mathbf{x})$ and $\mathbf{w}^{\otimes k}$ is unentangled with \mathcal{D} 's internal registers,⁹ $\mathcal{O}_P(\text{crs}, k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$ returns $\text{Prove}(\text{crs}, k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$, and $\mathcal{O}_S(\text{td}, k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$ returns $\text{Sim}_1(\text{td}, k_P, \mathbf{x})$.

Computational Mode Indistinguishability. For any non-uniform QPT distinguisher \mathcal{D} , we have

$$\left| \Pr \left[\mathcal{D}(\text{crs}) = 1 : \text{crs} \stackrel{\$}{\leftarrow} \text{CRSGen}(1^\lambda, \text{binding}) \right] - \Pr \left[\mathcal{D}(\text{crs}) = 1 : \text{crs} \stackrel{\$}{\leftarrow} \text{CRSGen}(1^\lambda, \text{hiding}) \right] \right| \leq \text{negl}(\lambda).$$

Remark 1 (On definition of zero-knowledge property). By considering a combination of CRSGen (for a fixed mode) and Preprocess as a setup algorithm, (dual-mode) CV-NIZK in the $\text{CRS} + (V \rightarrow P)$ model can be seen as a CV-NIZK in the QSP model in a syntactical sense. However, it seems difficult to prove that this satisfies (even a computational variant of) the zero-knowledge property defined in Definition 2.8 due to the following reasons:

1. In Definition 4.1, Sim_1 is quantum, whereas a simulator is required to be classical in Definition 2.8. We observe that this seems unavoidable in the above model: If k_P is quantum, then a classical simulator cannot even take k_P as input. On the other hand, if k_P is classical, then that implies $L \in \mathbf{AM}$ similarly to the final paragraph of Appendix D.
2. A simulator in Definition 4.1 can embed a trapdoor td behind the common reference string crs whereas a simulator in Definition 2.8 just takes an honestly generated verification key k_V as input. We remark that this also seems unavoidable since k_V may be maliciously generated when the verifier is malicious, in which case just taking k_V as input would be useless for the simulation.

On the other hand, the definition in Definition 4.1 allows a distinguisher (that plays the role of a malicious verifier) to maliciously generate k_P , which is a stronger capability than that of a distinguisher in Definition 2.8. Therefore, the zero-knowledge properties in Definition 4.1 and Definition 2.8 are incomparable. We believe that the definition of the zero-knowledge property in Definition 4.1 ensures meaningful security. It roughly means that any malicious verifier cannot learn anything beyond what could be computed in quantum polynomial time by itself even if it is allowed to interact with many sessions of honest provers under maliciously generated proving keys and the reused honestly generated common reference string. While this does not seem very meaningful when $L \in \mathbf{BQP}$, we can ensure a meaningful privacy of the witness when $L \in \mathbf{QMA}$. Finally we remark that our definition is essentially the same as that in [CVZ20] (except for the dual-mode property).

⁹We remark that k_P is allowed to be entangled with \mathcal{D} 's internal registers unlike $\mathbf{w}^{\otimes k}$. See also footnote 6.

Remark 2 (Comparison to NIZK in the malicious designated verifier model). A CV-NIZK for QMA in the CRS + (V → P) model as defined above is syntactically very similar to the NIZK for QMA in the malicious designated verifier model as introduced in [Shm21]. However, a crucial difference is that the proving key k_P is a quantum state in our case and cannot be reused whereas that is classical and can be reused for proving multiple statements in [Shm21]. On the other hand, a CV-NIZK in the CRS + (V → P) model has two nice features that the NIZK of [Shm21] does not have: one is that verification can be done classically in the online phase and the other is the dual-mode property.

Though Definition 4.1 does not explicitly require anything on soundness in the hiding mode or the zero-knowledge property in the binding mode, we can easily prove that they are satisfied in a computational sense. Specifically, we have the following lemma.

Lemma 4.2. *If a dual-mode CV-NIZK $\Pi = (\text{CRSGen}, \text{Preprocess}, \text{Prove}, \text{Verify})$ for a QMA promise problem L satisfies statistical s -soundness in the binding mode, statistical zero-knowledge property in the hiding mode, and computational mode indistinguishability, then it also satisfies the following properties.*

(Exclusive-Adaptive) Computational $(s + \text{negl}(\lambda))$ -Soundness in the Hiding Mode For all non-uniform QPT adversaries \mathcal{A} , we have

$$\Pr \left[\begin{array}{l} \text{Verify}(\text{crs}, k_V, \mathbf{x}, \pi) = \top : \\ \text{crs} \stackrel{\$}{\leftarrow} \text{CRSGen}(1^\lambda, \text{hiding}) \\ (k_P, k_V) \stackrel{\$}{\leftarrow} \text{Preprocess}(\text{crs}) \\ (\mathbf{x}, \pi) \stackrel{\$}{\leftarrow} \mathcal{A}(\text{crs}, k_P) \end{array} \right] \leq s + \text{negl}(\lambda).$$

where \mathcal{A} 's output must always satisfy $\mathbf{x} \in L_{\text{no}}$.

(Adaptive Multi-Theorem) Computational Zero-Knowledge in the Binding Mode. There exists a PPT simulator Sim_0 and QPT simulator Sim_1 such that for any non-uniform QPT distinguisher \mathcal{D} , we have

$$\left| \Pr \left[\mathcal{D}^{\mathcal{O}_P(\text{crs}, \cdot, \cdot)}(\text{crs}) = 1 : \text{crs} \stackrel{\$}{\leftarrow} \text{CRSGen}(1^\lambda, \text{binding}) \right] - \Pr \left[\mathcal{D}^{\mathcal{O}_S(\text{td}, \cdot, \cdot)}(\text{crs}) = 1 : (\text{crs}, \text{td}) \stackrel{\$}{\leftarrow} \text{Sim}_0(1^\lambda) \right] \right| \leq \text{negl}(\lambda)$$

where \mathcal{D} can make $\text{poly}(\lambda)$ queries, which should be of the form $(k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$ where $\mathbf{w} \in R_L(\mathbf{x})$ and $\mathbf{w}^{\otimes k}$ is unentangled with \mathcal{D} 's internal registers, $\mathcal{O}_P(\text{crs}, k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$ returns $\text{Prove}(\text{crs}, k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$, and $\mathcal{O}_S(\text{td}, k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$ returns $\text{Sim}_1(\text{td}, k_P, \mathbf{x})$.

Intuitively, the above lemma holds because soundness and zero-knowledge should transfer from one mode to the other by the mode indistinguishability since otherwise we can distinguish the two modes. Here, security degrades to computational ones as the mode indistinguishability only holds against QPT distinguishers. We omit a formal proof since this is easy and can be proven similarly to a similar statement for dual-mode NIZKs for NP, which has been folklore and formally proven recently [AB20].

Remark 3. Remark that soundness in the hiding mode is defined in the “exclusive style” where \mathcal{A} should always output $\mathbf{x} \in L_{\text{no}}$. This is weaker than soundness in the “penalizing style” as in Definition 4.1 where \mathcal{A} is allowed to also output $\mathbf{x} \in L_{\text{yes}}$ and we add $\mathbf{x} \in L_{\text{no}}$ as part of the adversary’s winning condition. This is because the adaptive soundness in the penalizing style does not transfer well through the mode change while the adaptive soundness in the exclusive style does.

This was formally proven for NIZK for **NP** in the common reference string model in [AB20], and easily extends to CV-NIZK for **QMA** in the $\text{CRS} + (V \rightarrow P)$ model. This is justified by the impossibility of penalizing-adaptively (computational) sound and statistically zero-knowledge NIZK for **NP** in the classical setting (under falsifiable assumptions) [Pas13]. We leave it open to study if a similar impossibility holds for dual-mode CV-NIZK for **QMA** in the $\text{CRS} + (V \rightarrow P)$ model.

Finally, we note that we can amplify the gap between the thresholds for completeness and soundness by parallel repetitions similarly to CV-NIZK in the QSP model as discussed in Section 2.3. As a result, we obtain the following lemma.

Lemma 4.3 (Gap amplification for dual-mode CV-NIZK in the $\text{CRS} + (V \rightarrow P)$ model). *If there exists a dual-mode CV-NIZK for L in the $\text{CRS} + (V \rightarrow P)$ model that satisfies c -completeness and s -soundness, for some $0 < s < c < 1$ such that $c - s > 1/\text{poly}(\lambda)$, then there exists a dual-mode CV-NIZK for L in the $\text{CRS} + (V \rightarrow P)$ model (with $(1 - \text{negl}(\lambda))$ -completeness and $\text{negl}(\lambda)$ -soundness).*

Since this can be proven similarly to Lemma 2.9, we omit a proof.

4.2 Building Blocks

We introduce two cryptographic building blocks for our dual-mode CV-NIZK in the $\text{CRS} + (V \rightarrow P)$ model.

Lossy Encryption The first building block is lossy encryption [BHY09]. Intuitively, a lossy encryption scheme is a public key encryption scheme with a special property that we can generate a *lossy key* that is computationally indistinguishable from an honestly generated public key, for which there is no corresponding decryption key.

Definition 4.4 (Lossy Encryption). *A lossy encryption scheme over the message space \mathcal{M} and the randomness space \mathcal{R} consists of PPT algorithms $\Pi_{\text{LE}} = (\text{InjGen}, \text{LossyGen}, \text{Enc}, \text{Dec})$ with the following syntax.*

InjGen(1^λ): *The injective key generation algorithm takes the security parameter 1^λ as input and outputs an injective public key pk and a secret key sk .*

LossyGen(1^λ): *The lossy key generation algorithm takes the security parameter 1^λ as input and outputs a lossy public key pk .*

Enc(pk, μ): *The encryption algorithm takes the public key pk and a message $\mu \in \mathcal{M}$ as input and outputs a ciphertext ct . This algorithm uses a randomness $R \in \mathcal{R}$. We denote by $\text{Enc}(\text{pk}, \mu; R)$ to mean that we run **Enc** on input pk and μ and randomness R when we need to clarify the randomness.*

Dec(sk, ct): *The decryption algorithm takes the secret key sk and a ciphertext ct as input and outputs a message μ .*

We require Π_{LE} to satisfy the following properties.

Correctness on Injective Keys For all $\mu \in \mathcal{M}$, we have

$$\Pr \left[\text{Dec}(\text{sk}, \text{ct}) = \mu : \begin{array}{l} (\text{pk}, \text{sk}) \xleftarrow{\$} \text{InjGen}(1^\lambda) \\ \text{ct} \xleftarrow{\$} \text{Enc}(\text{pk}, \mu) \end{array} \right] = 1.$$

Lossiness on Lossy Keys With overwhelming probability over $\text{pk} \xleftarrow{\$} \text{LossyGen}(1^\lambda)$, for all $\mu_0, \mu_1 \in \mathcal{M}$ and all unbounded-time distinguisher \mathcal{D} , we have

$$\left| \Pr \left[\mathcal{D}(\text{ct}) = 1 : \text{ct} \xleftarrow{\$} \text{Enc}(\text{pk}, \mu_0) \right] - \Pr \left[\mathcal{D}(\text{ct}) = 1 : \text{ct} \xleftarrow{\$} \text{Enc}(\text{pk}, \mu_1) \right] \right| \leq \text{negl}(\lambda).$$

Computational Mode Indistinguishability For any non-uniform QPT distinguisher \mathcal{D} , we have

$$\left| \Pr \left[\mathcal{D}(\text{pk}) = 1 : (\text{pk}, \text{sk}) \xleftarrow{\$} \text{InjGen}(1^\lambda) \right] - \Pr \left[\mathcal{D}(\text{pk}) = 1 : \text{pk} \xleftarrow{\$} \text{LossyGen}(1^\lambda) \right] \right| \leq \text{negl}(\lambda).$$

It is well-known that Regev’s encryption [Reg09] is lossy encryption under the LWE assumption with a negligible correctness error. We can modify the scheme to achieve perfect correctness by a standard technique. Then we have the following lemma.

Lemma 4.5. *If the LWE assumption holds, then there exists a lossy encryption scheme.*

Dual-Mode Oblivious Transfer The second building block is a k -out-of- n *dual-mode oblivious transfer*. Though this is a newly introduced definition in this paper, 1-out-of-2 case is already implicit in existing works on universally composable (UC-secure) [Can20] oblivious transfers [PVW08, Qua20].

Definition 4.6 (Dual-mode oblivious transfer). *A (2-round) k -out-of- n dual-mode oblivious transfer with a message space \mathcal{M} consists of PPT algorithms $\Pi_{\text{OT}} = (\text{CRSGen}, \text{Receiver}, \text{Sender}, \text{Derive})$.*

CRSGen($1^\lambda, \text{mode}$): *This is an algorithm supposed to be run by a trusted third party that takes the security parameter 1^λ and a mode $\text{mode} \in \{\text{binding}, \text{hiding}\}$ as input and outputs a common reference string crs .*

Receiver(crs, J): *This is an algorithm supposed to be run by a receiver that takes the common reference string crs and an ordered set of k indices $J \in [n]^k$ as input and outputs a first message ot_1 and a receiver’s state st .*

Sender($\text{crs}, \text{ot}_1, \boldsymbol{\mu}$): *This is an algorithm supposed to be run by a sender that takes the common reference string crs , a first message ot_1 sent from a receiver and a tuple of messages $\boldsymbol{\mu} \in \mathcal{M}^n$ as input and outputs a second message ot_2 .*

Derive($\text{crs}, \text{st}, \text{ot}_2$): *This is an algorithm supposed to be run by a receiver that takes a receiver’s state st and a second message ot_2 as input and outputs a tuple of messages $\boldsymbol{\mu}' \in \mathcal{M}^k$.*

We require the following properties.

Correctness For all $\text{mode} \in \{\text{binding}, \text{hiding}\}$, $J = (j_1, \dots, j_k) \in [n]^k$, and $\boldsymbol{\mu} = (\mu_1, \dots, \mu_n) \in \mathcal{M}^n$, we have

$$\Pr \left[\text{Derive}(\text{crs}, \text{st}, \text{ot}_2) = (\mu_{j_1}, \dots, \mu_{j_k}) : \begin{array}{l} \text{crs} \xleftarrow{\$} \text{CRSGen}(1^\lambda, \text{mode}) \\ (\text{ot}_1, \text{st}) \xleftarrow{\$} \text{Receiver}(\text{crs}, J) \\ \text{ot}_2 \xleftarrow{\$} \text{Sender}(\text{crs}, \text{ot}_1, \boldsymbol{\mu}) \end{array} \right] \geq 1 - \text{negl}(\lambda).$$

Statistical Receiver’s Security in the Binding Mode Intuitively, this security requires that the indices chosen by a receiver are information theoretically hidden from a sender in the binding

mode. Formally, we require that there is a PPT algorithm Sim_{rec} such that for any unbounded-time distinguisher \mathcal{D} and $J \in [n]^k$, we have

$$\left| \Pr \left[\mathcal{D}(\text{crs}, \text{ot}_1) = 1 : \begin{array}{l} \text{crs} \xleftarrow{\$} \text{CRSGen}(1^\lambda, \text{binding}) \\ (\text{ot}_1, \text{st}) \xleftarrow{\$} \text{Receiver}(\text{crs}, J) \end{array} \right] - \Pr \left[\mathcal{D}(\text{crs}, \text{ot}_1) = 1 : \begin{array}{l} \text{crs} \xleftarrow{\$} \text{CRSGen}(1^\lambda, \text{binding}) \\ \text{ot}_1 \xleftarrow{\$} \text{Sim}_{\text{rec}}(\text{crs}) \end{array} \right] \right| \leq \text{negl}(\lambda).$$

Statistical Sender's Security in the Hiding Mode Intuitively, this security requires that we can extract the indices of messages which a (possibly malicious) receiver tries to learn by using a trapdoor in the hiding mode. Formally, there are PPT algorithms Sim_{CRS} and Sim_{sen} and a deterministic classical polynomial-time algorithm Open_{rec} such that the following two properties are satisfied.

- For any unbounded-time distinguisher \mathcal{D} , we have

$$\left| \Pr \left[\mathcal{D}(\text{crs}) = 1 : \text{crs} \xleftarrow{\$} \text{CRSGen}(1^\lambda, \text{hiding}) \right] - \Pr \left[\mathcal{D}(\text{crs}) = 1 : (\text{crs}, \text{td}) \xleftarrow{\$} \text{Sim}_{\text{CRS}}(1^\lambda) \right] \right| \leq \text{negl}(\lambda).$$

- For any unbounded-time adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ (that plays the role of a malicious receiver) and $\boldsymbol{\mu} = (\mu_1, \dots, \mu_n)$, we have

$$\left| \Pr \left[\mathcal{A}_1(\text{st}_{\mathcal{A}}, \text{ot}_2) = 1 : \begin{array}{l} (\text{crs}, \text{td}) \xleftarrow{\$} \text{Sim}_{\text{CRS}}(1^\lambda) \\ (\text{ot}_1, \text{st}_{\mathcal{A}}) \xleftarrow{\$} \mathcal{A}_0(\text{crs}, \text{td}) \\ \text{ot}_2 \xleftarrow{\$} \text{Sender}(\text{crs}, \text{ot}_1, \boldsymbol{\mu}) \end{array} \right] - \Pr \left[\mathcal{A}_1(\text{st}_{\mathcal{A}}, \text{ot}_2) = 1 : \begin{array}{l} (\text{crs}, \text{td}) \xleftarrow{\$} \text{Sim}_{\text{CRS}}(1^\lambda) \\ (\text{ot}_1, \text{st}_{\mathcal{A}}) \xleftarrow{\$} \mathcal{A}_0(\text{crs}, \text{td}) \\ J := \text{Open}_{\text{rec}}(\text{td}, \text{ot}_1) \\ \text{ot}_2 \xleftarrow{\$} \text{Sim}_{\text{sen}}(\text{crs}, \text{ot}_1, J, \boldsymbol{\mu}_J) \end{array} \right] \right| \leq \text{negl}(\lambda)$$

where the output of Open_{rec} always satisfies $J \in [n]^k$ and $\boldsymbol{\mu}_J := (\mu_{j_1}, \dots, \mu_{j_k})$ for $J = (j_1, \dots, j_k)$.

Computational Mode Indistinguishability. For any non-uniform QPT distinguisher \mathcal{D} , we have

$$\left| \Pr \left[\mathcal{D}(\text{crs}) = 1 : \text{crs} \xleftarrow{\$} \text{CRSGen}(1^\lambda, \text{binding}) \right] - \Pr \left[\mathcal{D}(\text{crs}) = 1 : \text{crs} \xleftarrow{\$} \text{CRSGen}(1^\lambda, \text{hiding}) \right] \right| \leq \text{negl}(\lambda).$$

Remark 4 (On security definition of dual-mode oblivious transfer). We remark that security of a k -out-of- n dual-mode oblivious transfer as defined in Definition 4.6 does not imply UC-security [Can20, PVW08, Qua20] or even full-simulation security in the standard stand-alone simulation-based definition [Lin08]. This is because the receiver's security in Definition 4.6 only ensures privacy of J and does not prevent a malicious sender from generating ot_2 so that he can manipulate the message derived on the receiver's side depending on J . The security with such a weaker receiver's security is often referred to as half-simulation security [CNs07]. We define the security in this way due to the following reasons:

1. This definition is sufficient for constructing a dual-mode CV-NIZK in the $\text{CRS} + (V \rightarrow P)$ model given in Section 4.3 by additionally relying on lossy encryption.

2. We are not aware of an efficient construction of a k -out-of- n oblivious transfer that satisfies full-simulation security under a post-quantum assumption (even if we ignore the dual-mode property). We note that Quach [Qua20] gave a construction of a 1-out-of-2 oblivious transfer with full-simulation security based on LWE and we can extend it to 1-out-of- n one.¹⁰ However, we are not aware of an efficient way to convert this into k -out-of- n one without losing the full-simulation security. We note that a conversion from 1-out-of- n to k -out-of- n oblivious transfer by a simple k -parallel repetition does not work if we require the full-simulation security since a malicious sender can send different inconsistent messages in different sessions, which should be considered as an attack against full-simulation security. One possible way to prevent such an inconsistent message attack is to let the sender prove that the messages in all sessions are consistent by using (post-quantum) NIZK for **NP** in the common reference string model [PS19]. However, such a construction is very inefficient since it uses the underlying 1-out-of- n oblivious transfer in a non-black-box manner. On the other hand, the half-simulation security is preserved under parallel repetitions as shown in Appendix E, and thus we can achieve this much more efficiently.

Lemma 4.7. *If the LWE assumption holds, then there exists k -out-of- n dual-mode oblivious transfer for arbitrary $0 < k < n$ that are polynomial in λ .*

Proof (sketch). First, we can see that the LWE-based UC-secure OT by Quach [Qua20] can be seen as a 1-out-of-2 dual-mode oblivious transfer. This construction can be converted into 1-out-of- n dual-mode oblivious transfer by using the generic conversion for an ordinary oblivious transfer given in [BCR86] observing that the conversion preserves the dual-mode property.¹¹ By k -parallel repetition of the 1-out-of- n dual-mode oblivious transfer, we obtain k -out-of- n dual-mode oblivious transfer. The full proof can be found in Appendix E. \square

4.3 Construction

In this section, we construct a dual-mode CV-NIZK in the $\text{CRS} + (V \rightarrow P)$ model. As a result, we obtain the following theorem.

Theorem 4.8. *If the LWE assumption holds, then there exists a dual-mode CV-NIZK in the $\text{CRS} + (V \rightarrow P)$ model.*

Let L be a **QMA** promise problem, and \mathcal{H}_x , N , M , p_i , s_i , P_i , α , β , and ρ_{hist} be as in Lemma 2.7 for the language L . We let $N' := 3^5 \sum_{i=1}^5 \binom{N}{i}$ similarly to Lemma 3.2. Let $\Pi_{\text{LE}} = (\text{InjGen}_{\text{LE}}, \text{LossyGen}_{\text{LE}}, \text{Enc}_{\text{LE}}, \text{Dec}_{\text{LE}})$ be a lossy encryption scheme over the message space $\mathcal{M}_{\text{LE}} = \{0, 1\}^2$ and the randomness space \mathcal{R}_{LE} as defined in Definition 4.4. Let $\Pi_{\text{OT}} = (\text{CRSGen}_{\text{OT}}, \text{Receiver}_{\text{OT}}, \text{Sender}_{\text{OT}}, \text{Derive}_{\text{OT}})$ be a 5-out-of- N dual-mode oblivious transfer over the message space $\mathcal{M}_{\text{OT}} = \mathcal{M}_{\text{LE}} \times \mathcal{R}_{\text{LE}}$ as defined in Definition 4.6. Then our dual-mode CV-NIZK $\Pi_{\text{DM}} = (\text{CRSGen}_{\text{DM}}, \text{Preprocess}_{\text{DM}}, \text{Prove}_{\text{DM}}, \text{Verify}_{\text{DM}})$ for L is described in Figure 4.

Then we prove the following lemmas.

Lemma 4.9. Π_{DM} satisfies $(1 - \frac{\alpha}{N'} - \text{negl}(\lambda))$ -completeness.

Proof. By the correctness of Π_{OT} , it is easy to see that the probability that an honestly generated proof passes the verification differs from that in Π_{NIZK} in Figure 1 only by $\text{negl}(\lambda)$. Since Π_{NIZK} satisfies $(1 - \frac{\alpha}{N'})$ -completeness as shown in Lemma 3.2, Π_{DM} satisfies $(1 - \frac{\alpha}{N'} - \text{negl}(\lambda))$ -completeness. \square

¹⁰His construction further satisfies UC-security, which is stronger than full-simulation security.

¹¹Alternatively, it may be possible to directly construct 1-out-of- n dual-mode oblivious transfer by appropriately modifying the construction by Quach [Qua20].

$\text{CRSGen}_{\text{DM}}(1^\lambda, \text{mode})$: The CRS generation algorithm generates $\text{crs}_{\text{OT}} \xleftarrow{\$} \text{CRSGen}_{\text{OT}}(1^\lambda, \text{mode})$.

- If $\text{mode} = \text{binding}$, then it generates $(\text{pk}, \text{sk}) \xleftarrow{\$} \text{InjGen}_{\text{LE}}(1^\lambda)$.
- If $\text{mode} = \text{hiding}$, then it generates $\text{pk} \xleftarrow{\$} \text{LossyGen}_{\text{LE}}(1^\lambda)$.

Then it outputs $\text{crs}_{\text{DM}} := (\text{crs}_{\text{OT}}, \text{pk})$.

$\text{Preprocess}_{\text{DM}}(\text{crs}_{\text{DM}})$: The preprocessing algorithm parses $(\text{crs}_{\text{OT}}, \text{pk}) \leftarrow \text{crs}_{\text{DM}}$ and chooses $(W_1, \dots, W_N) \xleftarrow{\$} \{X, Y, Z\}^N$, $(m_1, \dots, m_N) \xleftarrow{\$} \{0, 1\}^N$, and a uniformly random subset $S_V \subseteq [N]$ such that $1 \leq |S_V| \leq 5$. Let $J = (j_1, \dots, j_5) \in [N]^5$ be the elements of S_V in the ascending order where we append arbitrary indices when $|S_V| < 5$. It generates $(\text{ot}_1, \text{st}) \xleftarrow{\$} \text{Receiver}_{\text{OT}}(\text{crs}_{\text{OT}}, J)$ and outputs a proving key $k_P := (\rho_P := \bigotimes_{j=1}^N (U(W_j)|m_j\rangle), \text{ot}_1)$ and a verification key $k_V := (W_1, \dots, W_N, m_1, \dots, m_N, S_V, \text{st})$.

$\text{Prove}_{\text{DM}}(\text{crs}_{\text{DM}}, k_P, \mathbf{x}, \mathbf{w})$: The proving algorithm parses $(\text{crs}_{\text{OT}}, \text{pk}) \leftarrow \text{crs}_{\text{DM}}$ and $(\rho_P, \text{ot}_1) \leftarrow k_P$, generates $(\hat{x}, \hat{z}) \xleftarrow{\$} \{0, 1\}^N \times \{0, 1\}^N$, generates the history state ρ_{hist} for \mathcal{H}_x from \mathbf{w} , and computes $\rho'_{\text{hist}} := X^{\hat{x}} Z^{\hat{z}} \rho_{\text{hist}} Z^{\hat{z}} X^{\hat{x}}$. It measures j -th qubits of ρ'_{hist} and ρ_P in the Bell basis for $j \in [N]$. Let $x := x_1 || x_2 || \dots || x_N$, and $z := z_1 || z_2 || \dots || z_N$ where (x_j, z_j) denotes the outcome of j -th measurement. For $j \in [N]$, it generates $\text{ct}_j := \text{Enc}_{\text{LE}}(\text{pk}, (\hat{x}_j, \hat{z}_j); R_j)$ where $R_j \xleftarrow{\$} \mathcal{R}_{\text{LE}}$ and \hat{x}_j and \hat{z}_j denote the j -th bits of \hat{x} and \hat{z} , respectively. It sets $\mu_j := ((\hat{x}_j, \hat{z}_j), R_j)$ for $j \in [N]$ and generates $\text{ot}_2 \xleftarrow{\$} \text{Sender}_{\text{OT}}(\text{crs}_{\text{OT}}, \text{ot}_1, (\mu_1, \dots, \mu_N))$. It outputs a proof $\pi := (x, z, \{\text{ct}_j\}_{j \in [N]}, \text{ot}_2)$.

$\text{Verify}_{\text{DM}}(\text{crs}_{\text{DM}}, k_V, \mathbf{x}, \pi)$: The verification algorithm parses $(\text{crs}_{\text{OT}}, \text{pk}) \leftarrow \text{crs}_{\text{DM}}$, $(W_1, \dots, W_N, m_1, \dots, m_N, S_V, \text{st}) \leftarrow k_V$, and $(x, z, \{\text{ct}_j\}_{j \in [N]}, \text{ot}_2) \leftarrow \pi$. It runs $\mu' \xleftarrow{\$} \text{Derive}_{\text{OT}}(\text{crs}_{\text{OT}}, \text{st}, \text{ot}_2)$ and parses $((\hat{x}'_1, \hat{z}'_1), R'_1), \dots, ((\hat{x}'_5, \hat{z}'_5), R'_5) \leftarrow \mu'$. If $\text{Enc}_{\text{LE}}(\text{pk}, (\hat{x}'_i, \hat{z}'_i); R'_i) \neq \text{ct}_{j_i}$ for some $i \in [5]$, it outputs \perp . Otherwise, it recovers $\{\hat{x}_j, \hat{z}_j\}_{j \in S_V}$ by setting $(\hat{x}_{j_i}, \hat{z}_{j_i}) := (\hat{x}'_i, \hat{z}'_i)$ for $i \in [|S_V|]$. It chooses $i \in [M]$ according to the probability distribution defined by $\{p_i\}_{i \in [M]}$ (i.e., chooses i with probability p_i). Let

$$S_i := \{j \in [N] \mid j\text{th Pauli operator of } P_i \text{ is not } I\}.$$

We note that we have $1 \leq |S_i| \leq 5$ by the 5-locality of \mathcal{H}_x . We say that P_i is consistent to $(S_V, \{W_j\}_{j \in S_V})$ if and only if $S_i = S_V$ and the j th Pauli operator of P_i is W_j for all $j \in S_i$. If P_i is not consistent to $(S_V, \{W_j\}_{j \in S_V})$, it outputs \top . If P_i is consistent to $(S_V, \{W_j\}_{j \in S_V})$, it flips a biased coin that heads with probability $1 - 3^{|S_i| - 5}$. If heads, it outputs \top . If tails, it defines

$$m'_j := \begin{cases} m_j \oplus x_j \oplus \hat{x}_j & (W_j = Z), \\ m_j \oplus z_j \oplus \hat{z}_j & (W_j = X), \\ m_j \oplus x_j \oplus \hat{x}_j \oplus z_j \oplus \hat{z}_j & (W_j = Y) \end{cases}$$

for $j \in S_i$, and outputs \top if $(-1)^{\bigoplus_{j \in S_i} m'_j} = -s_i$ and \perp otherwise.

Figure 4: Dual-Mode CV-NIZK Π_{DM} .

Lemma 4.10. Π_{DM} satisfies the computational mode indistinguishability.

Proof. This can be reduced to the computational mode indistinguishability of Π_{OT} and Π_{LE} in a straightforward manner. \square

Lemma 4.11. Π_{DM} satisfies statistical $\left(1 - \frac{\beta}{N'} + \text{negl}(\lambda)\right)$ -soundness in the binding mode.

Lemma 4.12. Π_{DM} satisfies the statistical zero-knowledge property in the hiding mode.

By combining Lemma 4.5, Lemmas 4.3, 4.7 and 4.9 to 4.12 and

$$\left(1 - \frac{\alpha}{N'} - \text{negl}(\lambda)\right) - \left(1 - \frac{\beta}{N'} + \text{negl}(\lambda)\right) = \frac{\beta - \alpha}{N'} - \text{negl}(\lambda) = \frac{1}{\text{poly}(\lambda)},$$

we obtain Theorem 4.8.

In the following, we prove Lemmas 4.11 and 4.12.

Proof of Lemma 4.11 (Soundness). For any adversary \mathcal{A} , we consider the following sequence of games between \mathcal{A} and the challenger where we denote by Win_i the event that the challenger returns \top in Game_i .

Game₁: This game is the original soundness game in the binding game. That is, it works as follows:

1. The challenger generates $\text{crs}_{\text{OT}} \xleftarrow{\$} \text{CRSGen}_{\text{OT}}(1^\lambda, \text{binding})$ and $(\text{pk}, \text{sk}) \xleftarrow{\$} \text{InjGen}_{\text{LE}}(1^\lambda)$.
2. The challenger generates $(W_1, \dots, W_N) \xleftarrow{\$} \{X, Y, Z\}^N$, $(m_1, \dots, m_N) \xleftarrow{\$} \{0, 1\}^N$, and $\rho_P := \bigotimes_{j=1}^N (U(W_j) | m_j)$.
3. The challenger generates S_V and $J = (j_1, \dots, j_5)$ similarly to $\text{Preprocess}_{\text{DM}}$.
4. The challenger generates $(\text{ot}_1, \text{st}) \xleftarrow{\$} \text{Receiver}_{\text{OT}}(\text{crs}_{\text{OT}}, J)$.
5. The challenger gives crs_{DM} and a proving key $k_P := (\rho_P, \text{ot}_1)$ to \mathcal{A} , and \mathcal{A} outputs $(\mathbf{x}, \pi = (x, z, \{\text{ct}_j\}_{j \in [N]}, \text{ot}_2))$. If $\mathbf{x} \in L_{\text{yes}}$, the challenger outputs \perp and immediately halts.
6. The challenger runs $\boldsymbol{\mu}' \xleftarrow{\$} \text{Derive}_{\text{OT}}(\text{crs}_{\text{OT}}, \text{st}, \text{ot}_2)$ and parses $((\hat{x}'_1, \hat{z}'_1), R'_1), \dots, ((\hat{x}'_5, \hat{z}'_5), R'_5) \leftarrow \boldsymbol{\mu}'$. If $\text{Enc}_{\text{LE}}(\text{pk}, (\hat{x}'_i, \hat{z}'_i); R'_i) \neq \text{ct}_{j_i}$ for some $i \in [5]$, it outputs \perp and immediately halts. Otherwise, it recovers $\{\hat{x}_j, \hat{z}_j\}_{j \in S_V}$ by setting $(\hat{x}_{j_i}, \hat{z}_{j_i}) := (\hat{x}'_i, \hat{z}'_i)$ for $i \in [5]$.
7. The challenger samples i and defines S_i and P_i similarly to $\text{Verify}_{\text{DM}}$. If P_i is not consistent to $(S_V, \{W_j\}_{j \in S_V})$, it outputs \top . If P_i is consistent to $(S_V, \{W_j\}_{j \in S_V})$, it flips a biased coin that heads with probability $1 - 3^{|S_i| - 5}$. If heads, it outputs \top . If tails, it defines m'_j for $j \in S_i$ similarly to $\text{Verify}_{\text{DM}}$ and outputs \top if $(-1)^{\bigoplus_{j \in S_i} m'_j} = -s_i$ and \perp otherwise.

Our goal is to prove $\Pr[\text{Win}_1] \leq 1 - \frac{\beta}{N'} + \text{negl}(\lambda)$.

Game₂: This game is identical to the previous game except that Step 6 is replaced with Step 6' described as follows.

- 6'. The challenger computes $(\hat{x}_j, \hat{z}_j) \xleftarrow{\$} \text{Dec}_{\text{LE}}(\text{sk}, \text{ct}_j)$ for $j \in [N]$.

If the challenger does not output \perp in Step 6, then we have $\text{Enc}_{\text{LE}}(\text{pk}, (\hat{x}'_i, \hat{z}'_i); R'_i) = \text{ct}_{j_i}$ for all $i \in [5]$. In this case, we have $\text{Dec}_{\text{LE}}(\text{sk}, \text{ct}_{j_i}) = (\hat{x}'_i, \hat{z}'_i)$ by correctness of Π_{LE} . Therefore, the values of $\{\hat{x}_j, \hat{z}_j\}_{j \in S_V}$ computed in Step 6 and 6' are identical conditioned on that the challenger does not output \perp in Step 6. Noting that Step 7 only uses the values of (\hat{x}_j, \hat{z}_j) for $j \in S_V$, we have $\Pr[\text{Win}_1] \leq \Pr[\text{Win}_2]$.

Game₃: This game is identical to the previous game except that Step 4 is replaced with Step 4' described as follows.

4' The challenger generates $\text{ot}_1 \xleftarrow{\$} \text{Sim}_{\text{rec}}(\text{crs}_{\text{OT}})$.

By statistical receiver's security in the binding mode of Π_{OT} , it is clear that we have $|\Pr[\text{Win}_3] - \Pr[\text{Win}_2]| \leq \text{negl}(\lambda)$.

Game₄: This game is identical to the previous game except that Step 2 is replaced with Step 2' described below.

2'. The challenger generates N Bell-pairs between registers \mathbf{P} and \mathbf{V} and lets ρ_P and ρ_V be quantum states in registers \mathbf{P} and \mathbf{V} , respectively. Then it chooses $(W_1, \dots, W_N) \xleftarrow{\$} \{X, Y, Z\}^N$, and measures j -th qubit of ρ_V in the W_j basis for all $j \in [N]$, and lets (m_1, \dots, m_N) be the measurement outcomes.

By Lemma 2.1, the joint distributions of $(\rho_P, (W_1, \dots, W_N, m_1, \dots, m_N))$ in **Game₃** and **Game₄** are identical, and thus we have $\Pr[\text{Win}_4] = \Pr[\text{Win}_3]$.

Game₅: This game is identical to the previous game except that the measurement of ρ_V in Step 2' is omitted and the way of generating $\{m'_j\}_{j \in S_i}$ in Step 7 is modified as follows.

- The challenger computes $\rho'_V := X^{x \oplus \hat{x}} Z^{z \oplus \hat{z}} \rho_V Z^{z \oplus \hat{z}} X^{x \oplus \hat{x}}$. For all $j \in S_i$, it measures j -th qubit of ρ'_V in W_j basis, and lets m'_j be the measurement outcome.

By Lemma 2.2, this does not change the distribution of $\{m'_j\}_{j \in S_i}$. Therefore, we have $\Pr[\text{Win}_5] = \Pr[\text{Win}_4]$.

Let \mathbf{E}_x be the event that the statement output by \mathcal{A} is \mathbf{x} , and $\rho'_{V,x}$ be the state in \mathbf{V} right before the measurement in the modified Step 7 conditioned on \mathbf{E}_x . For any fixed P_i , the probability that P_i is consistent to $(S_V, \{W_j\}_{j \in S_V})$ and the coin tails is $\frac{1}{N'}$. Therefore, by Lemma 2.5, we have

$$\Pr[\text{Win}_5 | \mathbf{E}_x] = 1 - \frac{1}{N'} \text{Tr}(\rho'_{V,x} \mathcal{H}_x).$$

Then we have

$$\Pr[\text{Win}_5] = \sum_{\mathbf{x} \notin L} \Pr[\mathbf{E}_x] \left(1 - \frac{1}{N'} \text{Tr}(\rho'_{V,x} \mathcal{H}_x) \right) \leq \sum_{\mathbf{x} \notin L} \Pr[\mathbf{E}_x] \left(1 - \frac{\beta}{N'} \right) \leq 1 - \frac{\beta}{N'}$$

where the first inequality follows from Lemma 2.7.

By combining the above, we obtain $\Pr[\text{Win}_1] \leq 1 - \frac{\beta}{N'} + \text{negl}(\lambda)$.

This completes the proof of Lemma 4.11. \square

Proof of Lemma 4.12 (Zero-Knowledge). Let Sim_{CRS} , Sim_{sen} , and Open_{rec} be the corresponding algorithms for statistical sender's security in the hiding mode of Π_{OT} . The simulator $\text{Sim} = (\text{Sim}_0, \text{Sim}_1)$ for Π_{DM} is described below.

Sim₀(1^λ): It generates $(\text{crs}_{\text{OT}}, \text{td}_{\text{OT}}) \xleftarrow{\$} \text{Sim}_{\text{CRS}}(1^\lambda)$ and $\text{pk} \xleftarrow{\$} \text{LossyGen}_{\text{LE}}(1^\lambda)$ and outputs $\text{crs}_{\text{DM}} := (\text{crs}_{\text{OT}}, \text{pk})$ and $\text{td}_{\text{DM}} := (\text{crs}_{\text{OT}}, \text{td}_{\text{OT}}, \text{pk})$.

$\text{Sim}_1(\text{td}_{\text{DM}}, k_P, \mathbf{x})$: The simulator parses $(\text{crs}_{\text{OT}}, \text{td}_{\text{OT}}, \text{pk}) \leftarrow \text{td}_{\text{DM}}$ and $(\rho_P, \text{ot}_1) \leftarrow k_P$ and does the following.

1. Compute $J := \text{Open}_{\text{rec}}(\text{td}_{\text{OT}}, \text{ot}_1)$. Let $S_V := \{j_1, \dots, j_5\} \subseteq [N]$ where $J = (j_1, \dots, j_5)$.
2. Generate $(\hat{x}, \hat{z}) \xleftarrow{\$} \{0, 1\}^N \times \{0, 1\}^N$, $R_j \xleftarrow{\$} \mathcal{R}_{\text{LE}}$ for $j \in [N]$, $\text{ct}_j := \text{Enc}_{\text{LE}}(\text{pk}, (\hat{x}_j, \hat{z}_j); R_j)$ for all $j \in [N]$, and $\text{ot}_2 \xleftarrow{\$} \text{Sim}_{\text{sen}}(\text{crs}_{\text{OT}}, \text{ot}_1, J, \mu_J)$ where $\mu_J := (\mu_{j_1}, \dots, \mu_{j_5})$ and $\mu_{j_i} := ((\hat{x}_{j_i}, \hat{z}_{j_i}), R_{j_i})$ for $i \in [5]$.
3. Generate the classical description of the density matrix $\rho_{S_V} := \text{Sim}_{\text{hist}}(\mathbf{x}, S_V)$ where Sim_{hist} is as in Lemma 2.7.
4. Generate $\tilde{\rho}'_{\text{hist}} := \left(\prod_{j \in S_V} X_j^{\hat{x}_j} Z_j^{\hat{z}_j} \right) \rho_{S_V} \left(\prod_{j \in S_V} Z_j^{\hat{z}_j} X_j^{\hat{x}_j} \right) \otimes \frac{I_{[N] \setminus S_V}}{2^{|[N] \setminus S_V|}}$.
5. Measure j -th qubits of $\tilde{\rho}'_{\text{hist}}$ and ρ_P in the Bell basis for $j \in [N]$, and let (x_j, z_j) be the j -th measurement result.
6. Output $\pi := (x, z, \{\text{ct}_j\}_{j \in [N]}, \text{ot}_2)$ where $x := x_1 \| x_2 \| \dots \| x_N$ and $z := z_1 \| z_2 \| \dots \| z_N$.

We consider the following sequence of modified versions of Sim_1 , which take $\mathbf{w} \in R_L(\mathbf{x})$ as an additional input.

$\text{Sim}_1^{(1)}(\text{td}_{\text{DM}}, k_P, \mathbf{x}, \mathbf{w})$: This simulator works similarly to Sim_1 except that it generates the history state ρ_{hist} for $\mathcal{H}_{\mathbf{x}}$ from \mathbf{w} instead of ρ_{S_V} in Step 3, defines $\rho'_{\text{hist}} := X^{\hat{x}} Z^{\hat{z}} \rho_{\text{hist}} Z^{\hat{z}} X^{\hat{x}}$ in Step 4, and uses ρ'_{hist} instead of $\tilde{\rho}'_{\text{hist}}$ in Step 5.

$\text{Sim}_1^{(2)}(\text{td}_{\text{DM}}, k_P, \mathbf{x}, \mathbf{w})$: This simulator works similarly to $\text{Sim}_1^{(1)}$ except that in Step 2, it generates $\text{ot}_2 \xleftarrow{\$} \text{Sender}_{\text{OT}}(\text{crs}_{\text{OT}}, \text{ot}_1, (\mu_1, \dots, \mu_N))$ instead of $\text{ot}_2 \xleftarrow{\$} \text{Sim}_{\text{sen}}(\text{crs}_{\text{OT}}, \text{ot}_1, J, \mu_J)$ where $\mu_j := ((\hat{x}_j, \hat{z}_j), R_j)$ for $j \in [N]$. We note that $\text{Sim}_1^{(2)}$ needs not run Step 1 since it does not use J in later steps and thus it does not use td_{OT} .

Let $\mathcal{O}_P(\text{crs}_{\text{DM}}, \cdot, \cdot, \cdot)$ and $\mathcal{O}_S(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)$ be as in Definition 4.1 and $\mathcal{O}_S^{(i)}(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)$ be the oracle that works similarly to $\mathcal{O}_S(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)$ except that it uses $\text{Sim}_1^{(i)}$ instead of Sim_1 for $i = 1, 2$.

Then we prove the following claims.

Claim 4.13. *If Π_{LE} satisfies lossiness on lossy keys, we have*

$$\left| \Pr \left[\mathcal{D}^{\mathcal{O}_S(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)}(\text{crs}_{\text{DM}}) = 1 \right] - \Pr \left[\mathcal{D}^{\mathcal{O}_S^{(1)}(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)}(\text{crs}_{\text{DM}}) = 1 \right] \right| \leq \text{negl}(\lambda)$$

where $(\text{crs}_{\text{DM}}, \text{td}_{\text{DM}}) \xleftarrow{\$} \text{Sim}_0(1^\lambda)$ for any distinguisher \mathcal{D} that makes $\text{poly}(\lambda)$ queries of the form $(k_P = (\rho_P, \text{ot}_1), \mathbf{x}, \mathbf{w})$ for some $\mathbf{w} \in R_L(\mathbf{x})$.

Proof of Claim 4.13. Let $\tilde{\mathcal{O}}_S(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)$ and $\tilde{\mathcal{O}}_S^{(1)}(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)$ be oracles that work similarly to $\mathcal{O}_S(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)$ and $\mathcal{O}_S^{(1)}(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)$ except that they generate $\text{ct}_j := \text{Enc}_{\text{LE}}(\text{pk}, (0, 0); R_j)$ instead of $\text{ct}_j := \text{Enc}_{\text{LE}}(\text{pk}, (\hat{x}_j, \hat{z}_j); R_j)$ for $j \notin S_V$, respectively. By lossiness on lossy keys of Π_{LE} , \mathcal{D} cannot distinguish $\tilde{\mathcal{O}}_S(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)$ and $\tilde{\mathcal{O}}_S^{(1)}(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)$ from $\mathcal{O}_S(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)$ and $\mathcal{O}_S^{(1)}(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)$ with non-negligible advantage, respectively, noting that no information of $\{R_j\}_{j \notin S_V}$ is given to \mathcal{D} . When \mathcal{D} is given either of $\tilde{\mathcal{O}}_S(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)$ or $\tilde{\mathcal{O}}_S^{(1)}(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)$, it has no information on $\{\hat{x}_j, \hat{z}_j\}_{j \notin S_V}$. Therefore, by Lemma 2.3, we have

$$\rho'_{\text{hist}} = \left(\prod_{j \in S_V} X_j^{\hat{x}_j} Z_j^{\hat{z}_j} \right) \text{Tr}_{N \setminus S_V}[\rho_{\text{hist}}] \left(\prod_{j \in S_V} Z_j^{\hat{z}_j} X_j^{\hat{x}_j} \right) \otimes \frac{I_{[N] \setminus S_V}}{2^{|[N] \setminus S_V|}}$$

from the view of \mathcal{D} . By Lemma 2.7, we have $\|\rho_{S_V} - \text{Tr}_{[N]\setminus S_V} \rho_{\text{hist}}\|_{tr} \leq \text{negl}(\lambda)$. Therefore, we have $\|\tilde{\rho}'_{\text{hist}} - \rho'_{\text{hist}}\|_{tr} \leq \text{negl}(\lambda)$. This means that it cannot distinguish $\tilde{\mathcal{O}}_S(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)$ and $\tilde{\mathcal{O}}_S^{(1)}(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)$ with non-negligible advantage. By combining the above, Claim 4.13 follows. \square

Claim 4.14. *If Π_{OT} satisfies the second item of statistical sender's security in the hiding mode, we have*

$$\left| \Pr \left[\mathcal{D}^{\mathcal{O}_S^{(1)}(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)}(\text{crs}_{\text{DM}}) = 1 \right] - \Pr \left[\mathcal{D}^{\mathcal{O}_S^{(2)}(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)}(\text{crs}_{\text{DM}}) = 1 \right] \right| \leq \text{negl}(\lambda)$$

where $(\text{crs}_{\text{DM}}, \text{td}_{\text{DM}}) \stackrel{\S}{\leftarrow} \text{Sim}_0(1^\lambda)$ for any distinguisher \mathcal{D} that makes $\text{poly}(\lambda)$ queries.

Proof of Claim 4.14. Let $Q = \text{poly}(\lambda)$ be the maximum number of \mathcal{D} 's queries. For $i = 0, \dots, Q$, let $\mathcal{O}_S^{(1,i)}(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)$ be the hybrid oracle that works similarly to $\mathcal{O}_S^{(2)}(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)$ for the first i queries and works similarly to $\mathcal{O}_S^{(1)}(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)$ for the rest. By a standard hybrid argument, it suffices to prove

$$\left| \Pr \left[\mathcal{D}^{\mathcal{O}_S^{(1,i)}(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)}(\text{crs}_{\text{DM}}) = 1 \right] - \Pr \left[\mathcal{D}^{\mathcal{O}_S^{(1,(i+1))}(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)}(\text{crs}_{\text{DM}}) = 1 \right] \right| \leq \text{negl}(\lambda) \quad (1)$$

where $(\text{crs}_{\text{DM}}, \text{td}_{\text{DM}}) \stackrel{\S}{\leftarrow} \text{Sim}_0(1^\lambda)$ for all $i = 0, \dots, Q - 1$. For proving this, for any fixed $(\hat{x}, \hat{z}) \in \{0, 1\}^N \times \{0, 1\}^N$ and $\{R_j\}_{j \in [N]} \in \mathcal{R}_{\text{LE}}^N$, we consider the following adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ against the second item of statistical sender's security in the hiding mode of Π_{OT} .

$\mathcal{A}_0(\text{crs}_{\text{OT}}, \text{td}_{\text{OT}})$: It generates $\text{pk} \stackrel{\S}{\leftarrow} \text{LossyGen}(1^\lambda)$, gives $\text{crs}_{\text{DM}} := (\text{crs}_{\text{OT}}, \text{pk})$ to \mathcal{D} as input and runs it until it makes $(i + 1)$ -th query where \mathcal{A}_0 simulates responses to the first i queries similarly to $\mathcal{O}_S^{(2)}(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)$ where $\text{td}_{\text{DM}} = (\text{crs}_{\text{OT}}, \text{td}_{\text{OT}}, \text{pk})$. Let $(k_P, \mathbf{x}, \mathbf{w})$ be \mathcal{D} 's $(i + 1)$ -th query. \mathcal{A}_0 parses $(\rho_P, \text{ot}_1) \leftarrow k_P$ and computes the history state ρ_{hist} for $\mathcal{H}_{\mathbf{x}}$ from \mathbf{w} . It outputs ot_1 and $\text{st}_{\mathcal{A}} := (\rho_P, \rho_{\text{hist}})$.

$\mathcal{A}_1(\text{st}_{\mathcal{A}} = (\rho_P, \rho_{\text{hist}}), \text{ot}_2)$: It generates $\text{ct}_j := \text{Enc}_{\text{LE}}(\text{pk}, (\hat{x}_j, \hat{z}_j); R_j)$ for all $j \in [N]$ and $\rho'_{\text{hist}} := X^{\hat{x}} Z^{\hat{z}} \rho_{\text{hist}} Z^{\hat{z}} X^{\hat{x}}$, measures j -th qubits of ρ'_{hist} and ρ_P in the Bell basis for $j \in [N]$, lets (x_j, z_j) be the j -th measurement result, and returns $\pi := (x, z, \{\text{ct}_j\}_{j \in [N]}, \text{ot}_2)$ to \mathcal{D} as the response of the oracle to the $(i + 1)$ -th query where $x := x_1 \| x_2 \| \dots \| x_N$ and $z := z_1 \| z_2 \| \dots \| z_N$. \mathcal{A}_1 runs the rest of the execution of \mathcal{D} by simulating the oracle similarly to $\mathcal{O}_S^{(1)}(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)$. Finally, \mathcal{A}_1 outputs whatever \mathcal{D} outputs.

Let $\boldsymbol{\mu} := (((\hat{x}_1, \hat{z}_1), R_1), \dots, ((\hat{x}_N, \hat{z}_N), R_N))$. If ot_2 is generated as $\text{ot}_2 \stackrel{\S}{\leftarrow} \text{Sender}(\text{crs}_{\text{OT}}, \text{ot}_1, \boldsymbol{\mu})$, then \mathcal{A} perfectly simulates the execution of $\mathcal{D}^{\mathcal{O}_S^{(1,i)}(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)}(\text{crs}_{\text{DM}})$ conditioned on the fixed (\hat{x}, \hat{z}) and $\{R_j\}_{j \in [N]}$. On the other hand, if ot_2 is generated as $J := \text{Open}_{\text{rec}}(\text{td}_{\text{OT}}, \text{ot}_1)$ and $\text{ot}_2 \stackrel{\S}{\leftarrow} \text{Sim}_{\text{sen}}(\text{crs}_{\text{OT}}, \text{ot}_1, J, \boldsymbol{\mu}_J)$, then \mathcal{A} perfectly simulates the execution of $\mathcal{D}^{\mathcal{O}_S^{(1,(i+1))}(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)}(\text{crs}_{\text{DM}})$ conditioned on the fixed (\hat{x}, \hat{z}) and $\{R_j\}_{j \in [N]}$. Therefore, averaging over the random choice of (\hat{x}, \hat{z}) and $\{R_j\}_{j \in [N]}$, the l.h.s. of Equation (1) can be upper bounded by the average of the advantage of \mathcal{A} to distinguish the two cases, which is negligible by the assumption. This completes the proof of Claim 4.14. \square

Claim 4.15. *If Π_{OT} satisfies the first item of statistical sender's security in the hiding mode, We have*

$$\left| \Pr \left[\mathcal{D}^{\mathcal{O}_S^{(2)}(\text{td}_{\text{DM}}, \cdot, \cdot, \cdot)}(\text{crs}_{\text{DM}}) = 1 : (\text{crs}_{\text{DM}}, \text{td}_{\text{DM}}) \stackrel{\S}{\leftarrow} \text{Sim}_0(1^\lambda) \right] - \Pr \left[\mathcal{D}^{\mathcal{O}_P(\text{crs}_{\text{DM}}, \cdot, \cdot, \cdot)}(\text{crs}_{\text{DM}}) = 1 : \text{crs}_{\text{DM}} \stackrel{\S}{\leftarrow} \text{CRSGen}_{\text{DM}}(1^\lambda, \text{hiding}) \right] \right| \leq \text{negl}(\lambda)$$

Proof of Claim 4.15. For any $(\text{crs}_{\text{DM}}, \text{td}_{\text{DM}}) \stackrel{\$}{\leftarrow} \text{Sim}_0(1^\lambda)$, k_P , \mathbf{x} , and \mathbf{w} , we have

$$\mathcal{O}_S^{(2)}(\text{td}_{\text{DM}}, k_P, \mathbf{x}, \mathbf{w}) = \mathcal{O}_P(\text{crs}_{\text{DM}}, k_P, \mathbf{x}, \mathbf{w})$$

observing that $\text{Sim}_1^{(2)}$ works in the exactly the same way as the honest proving algorithm. Moreover, we can see that the distributions of crs_{DM} generated by $\text{Sim}_0(1^\lambda)$ and $\text{CRSGen}_{\text{DM}}(1^\lambda, \text{hiding})$ are statistically indistinguishable by the first item of statistical sender's security in the hiding mode of Π_{OT} . Therefore Claim 4.15 follows. \square

By combining Claims 4.13 to 4.15, We can complete the proof of Lemma 4.12. \square

5 CV-NIZK via Fiat-Shamir Transformation

In this section, we construct CV-NIZK in the quantum random oracle model via the Fiat-Shamir transformation.

5.1 Definition

We give a formal definition of CV-NIZK in the QRO + $(V \rightarrow P)$ model.

Definition 5.1 (CV-NIZK in the QRO + $(V \rightarrow P)$ Model). *A CV-NIZK for a QMA promise problem $L = (L_{\text{yes}}, L_{\text{no}})$ in the QRO + $(V \rightarrow P)$ model w.r.t. a random oracle distribution ROdist consists of algorithms $\Pi = (\text{Preprocess}, \text{Prove}, \text{Verify})$ with the following syntax:*

Preprocess (1^λ) : *This is a QPT algorithm that takes the security parameter 1^λ as input, and outputs a quantum proving key k_P and a classical verification key k_V . We note that this algorithm is supposed to be run by the verifier as preprocessing, and k_P is supposed to be sent to the prover while k_V is supposed to be kept on verifier's side in secret. We also note that they can be used only once and cannot be reused.*

Prove $^H(k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$: *This is a QPT algorithm that is given quantum oracle access to the random oracle H . It takes the proving key k_P , a statement \mathbf{x} , and $k = \text{poly}(\lambda)$ copies $\mathbf{w}^{\otimes k}$ of a witness $\mathbf{w} \in R_L(\mathbf{x})$ as input, and outputs a classical proof π .*

Verify $^H(k_V, \mathbf{x}, \pi)$: *This is a PPT algorithm that is given classical oracle access to the random oracle H . It takes the verification key k_V , a statement \mathbf{x} , and a proof π as input, and outputs \top indicating acceptance or \perp indicating rejection.*

We require Π to satisfy the following properties.

Completeness. *For all $\mathbf{x} \in L_{\text{yes}} \cap \{0, 1\}^\lambda$, and $\mathbf{w} \in R_L(\mathbf{x})$, we have*

$$\Pr \left[\begin{array}{c} H \stackrel{\$}{\leftarrow} \text{ROdist} \\ \text{Verify}^H(k_V, \mathbf{x}, \pi) = \top : (k_P, k_V) \stackrel{\$}{\leftarrow} \text{Preprocess}(1^\lambda) \\ \pi \stackrel{\$}{\leftarrow} \text{Prove}^H(k_P, \mathbf{x}, \mathbf{w}^{\otimes k}) \end{array} \right] \geq 1 - \text{negl}(\lambda).$$

Adaptive Statistical Soundness. *For all adversaries \mathcal{A} that make at most $\text{poly}(\lambda)$ quantum random oracle queries, we have*

$$\Pr \left[\begin{array}{c} H \stackrel{\$}{\leftarrow} \text{ROdist} \\ \mathbf{x} \in L_{\text{no}} \wedge \text{Verify}^H(k_V, \mathbf{x}, \pi) = \top : (k_P, k_V) \stackrel{\$}{\leftarrow} \text{Preprocess}(1^\lambda) \\ (\mathbf{x}, \pi) \stackrel{\$}{\leftarrow} \mathcal{A}^H(k_P) \end{array} \right] \leq \text{negl}(\lambda).$$

Adaptive Multi-Theorem Zero-Knowledge. For defining the zero-knowledge property in the QROM, we define the syntax of a simulator in the QROM following [Unr15]. A simulator is given quantum access to the random oracle H and classical access to reprogramming oracle **Reprogram**. When the simulator queries (x, y) to **Reprogram**, the random oracle H is reprogrammed so that $H(x) := y$ while keeping the values on other inputs unchanged. Then the adaptive multi-theorem zero-knowledge property is defined as follows:

There exists a QPT simulator **Sim** with the above syntax such that for any QPT distinguisher \mathcal{D} , we have

$$\left| \Pr \left[\mathcal{D}^{H, \mathcal{O}_P^H(\cdot, \cdot)}(1^\lambda) = 1 : H \stackrel{\$}{\leftarrow} \text{ROdist} \right] - \Pr \left[\mathcal{D}^{H, \mathcal{O}_S^{H, \text{Reprogram}}(\cdot, \cdot)}(1^\lambda) = 1 : H \stackrel{\$}{\leftarrow} \text{ROdist} \right] \right| \leq \text{negl}(\lambda)$$

where \mathcal{D} 's queries to the second oracle should be of the form $(k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$ where $\mathbf{w} \in R_L(\mathbf{x})$ and $\mathbf{w}^{\otimes k}$ is unentangled with \mathcal{D} 's internal registers,¹² $\mathcal{O}_P^H(k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$ returns $\text{Prove}^H(k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$, and $\mathcal{O}_S^{H, \text{Reprogram}}(k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$ returns $\text{Sim}^{H, \text{Reprogram}}(k_P, \mathbf{x})$.

Remark 5. Remark that the “multi-theorem” zero-knowledge does not mean that a preprocessing can be reused many times. It rather means that a single random oracle can be reused as long as a fresh preprocessing is run every time. This is consistent to the definition in the $\text{CRS} + (V \rightarrow P)$ model (Definition 4.1) if we think of the random oracle as replacement of CRS.

5.2 Building Blocks

We use the two cryptographic primitives, a non-interactive commitment scheme and a Σ -protocol with quantum preprocessing, for our construction.

Definition 5.2 (Non-interactive commitment scheme). A non-interactive commitment scheme with the message space \mathcal{M} is a tuple of PPT algorithms (**Commit**, **Verify**) with the following syntax:

Commit $(1^\lambda, m)$: It takes the security parameter 1^λ and a message $m \in \mathcal{M}$ as input, and outputs a commitment **com** and a decommitment d .

Verify $(1^\lambda, m, \text{com}, d)$: It takes the security parameter 1^λ , a message $m \in \mathcal{M}$, commitment **com**, and decommitment d as input, and outputs \top indicating acceptance or \perp indicating rejection.

We require a non-interactive commitment scheme to satisfy the following properties:

Perfect Correctness. For any $\lambda \in \mathbb{N}$ and $m \in \mathcal{M}$, we have

$$\Pr[\text{Verify}(1^\lambda, m, \text{com}, d) = \top : (\text{com}, d) \stackrel{\$}{\leftarrow} \text{Commit}(1^\lambda, m)] = 1.$$

Perfect Binding. For all $\lambda \in \mathbb{N}$, there do not exist m, m', com, d, d' such that $m \neq m'$ and $\text{Verify}(1^\lambda, m, \text{com}, d) = \text{Verify}(1^\lambda, m', \text{com}, d') = \top$.

Computational Hiding. For any QPT adversary \mathcal{A} and messages m_0, m_1 , we have

$$\left| \Pr[\mathcal{A}(\text{com}) = 1 : (\text{com}, d) \stackrel{\$}{\leftarrow} \text{Commit}(1^\lambda, m_0)] - \Pr[\mathcal{A}(\text{com}) = 1 : (\text{com}, d) \stackrel{\$}{\leftarrow} \text{Commit}(1^\lambda, m_1)] \right| = \text{negl}(\lambda).$$

¹²We remark that k_P is allowed to be entangled with \mathcal{D} 's internal registers unlike $\mathbf{w}^{\otimes k}$. See also footnote 6.

It is known that a non-interactive commitment scheme exists assuming the existence of injective one-way functions or perfectly correct public key encryption (or more generally key exchange protocols) [LS19]. In the QROM, a non-interactive commitment scheme exists without any assumption since a random oracle with a sufficiently large range is injective with overwhelming probability over the choice of the random oracle and hard to invert even with quantum access to the oracle [BBBV97]. In our constructions and security proofs, we use a non-interactive commitment scheme in the standard model. This is for notational simplicity and also for clarifying that the full power of random oracles is not needed for this component. We stress that this does not mean that we assume an additional assumption for our construction of NIZK since a non-interactive commitment scheme unconditionally exists in the QROM as mentioned above and all security proofs work similarly with a non-interactive commitment scheme in the QROM.

Definition 5.3 (Σ -protocol with Quantum Preprocessing). *A Σ -protocol with quantum preprocessing for a QMA promise problem $L = (L_{\text{yes}}, L_{\text{no}})$ consists of algorithms $\Pi = (\text{Preprocess}, \text{Prove}_1, \text{Verify}_1, \text{Prove}_2, \text{Verify}_2)$ with the following syntax:*

$\text{Preprocess}(1^\lambda)$: *This is a QPT algorithm that takes the security parameter 1^λ as input, and outputs a quantum proving key k_P and a classical verification key k_V . We note that this algorithm is supposed to be run by the verifier as preprocessing, and k_P is supposed to be sent to the prover while k_V is supposed to be kept on verifier's side in secret. We also note that they can be used only once and cannot be reused.*

$\text{Prove}_1(k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$: *This is a QPT algorithm that takes the proving key k_P , a statement \mathbf{x} , and $k = \text{poly}(\lambda)$ copies $\mathbf{w}^{\otimes k}$ of a witness $\mathbf{w} \in R_L(\mathbf{x})$ as input, and outputs a classical message msg_1 and a state st .*

$\text{Verify}_1(1^\lambda)$: *This is a PPT algorithm that takes the security parameter 1^λ , and outputs a classical message msg_2 , which is uniformly sampled from a certain set.*

$\text{Prove}_2(\text{st}, \text{msg}_2)$: *This is a QPT algorithm that takes the state st and the message msg_2 as input, and outputs a classical message msg_3 .*

$\text{Verify}_2(k_V, \mathbf{x}, \text{msg}_1, \text{msg}_2, \text{msg}_3)$: *This is a PPT algorithm that takes the verification key k_V , the statement \mathbf{x} , and classical messages $\text{msg}_1, \text{msg}_2, \text{msg}_3$ as input, and outputs \top indicating acceptance or \perp indicating rejection.*

We require Π to satisfy the following properties.

c-Completeness. *For all $\mathbf{x} \in L_{\text{yes}} \cap \{0, 1\}^\lambda$, and $\mathbf{w} \in R_L(\mathbf{x})$, we have*

$$\Pr \left[\text{Verify}_2(k_V, \mathbf{x}, \text{msg}_1, \text{msg}_2, \text{msg}_3) = \top : \begin{array}{l} (k_P, k_V) \stackrel{\$}{\leftarrow} \text{Preprocess}(1^\lambda) \\ (\text{msg}_1, \text{st}) \stackrel{\$}{\leftarrow} \text{Prove}_1(k_P, \mathbf{x}, \mathbf{w}^{\otimes k}) \\ \text{msg}_2 \stackrel{\$}{\leftarrow} \text{Verify}_1(1^\lambda) \\ \text{msg}_3 \stackrel{\$}{\leftarrow} \text{Prove}_2(\text{st}, \text{msg}_2) \end{array} \right] \geq c.$$

(Adaptive Statistical) s-soundness. *For all adversary $(\mathcal{A}_1, \mathcal{A}_2)$, we have*

$$\Pr \left[\mathbf{x} \in L_{\text{no}} \wedge \Sigma.\text{Verify}_2(k_V, \mathbf{x}, \text{msg}_1, \text{msg}_2, \text{msg}_3) = \top : \begin{array}{l} (k_P, k_V) \stackrel{\$}{\leftarrow} \text{Preprocess}(1^\lambda) \\ (\mathbf{x}, \text{st}, \text{msg}_1) \stackrel{\$}{\leftarrow} \mathcal{A}_1(k_P) \\ \text{msg}_2 \stackrel{\$}{\leftarrow} \text{Verify}_1(1^\lambda) \\ \text{msg}_3 \stackrel{\$}{\leftarrow} \mathcal{A}_2(\text{st}, \text{msg}_2) \end{array} \right] \leq s.$$

Special Zero-Knowledge. *There exists a QPT algorithm Sim such that for any $\mathbf{x} \in L_{\text{yes}}$, $\mathbf{w} \in R_L(\mathbf{x})$, msg_2 , and QPT adversary $(\mathcal{A}_1, \mathcal{A}_2)$, we have*

$$\left| \begin{array}{l} \Pr \left[\begin{array}{l} (k_P, \text{st}_{\mathcal{A}}) \xleftarrow{\$} \mathcal{A}_1(1^\lambda) \\ (\text{msg}_1, \text{st}) \xleftarrow{\$} \text{Prove}_1(k_P, \mathbf{x}, \mathbf{w}^{\otimes k}) \\ \text{msg}_3 \xleftarrow{\$} \text{Prove}_2(\text{st}, \text{msg}_2) \end{array} \right] \\ - \Pr \left[\begin{array}{l} (k_P, \text{st}_{\mathcal{A}}) \xleftarrow{\$} \mathcal{A}_1(1^\lambda) \\ (\text{msg}_1, \text{msg}_3) \xleftarrow{\$} \text{Sim}(k_P, \mathbf{x}, \text{msg}_2) \end{array} \right] \end{array} \right| \leq \text{negl}(\lambda).$$

High Min-Entropy. *Prove₁ can be divided into the “quantum part” and “classical part” as follows:*

Prove₁^Q($k_P, \mathbf{x}, \mathbf{w}^{\otimes k}$): *This is a QPT algorithm that outputs a classical string st’.*

Prove₁^C(st’): *This is a PPT algorithm that outputs msg₁ and st.*

Moreover, for any st’ generated by Prove₁^Q, we have

$$\max_{\text{msg}_1^*} \Pr[\text{Prove}_1^C(\text{st}') = \text{msg}_1^*] = \text{negl}(\lambda).$$

Remark 6 (On Soundness). *Some existing works require a Σ -protocol to satisfy special soundness, which means that one can extract a witness from two accepting transcripts whose first messages are identical and the second messages are different. This property is often useful for achieving proof of knowledge. We do not require special soundness since we do not consider proof of knowledge in this paper and our construction does not seem to satisfy special soundness.*

Remark 7 (On Zero-Knowledge). *Our definition of the zero-knowledge property is based on the special honest-verifier zero-knowledge often required for classical Σ -protocol without preprocessing. However, our definition considers a partially malicious verifier that maliciously runs the preprocessing, which is a crucial difference from the classical case. This is why we call this property as special zero-knowledge rather than special honest-verifier zero-knowledge. Note that special zero-knowledge property is weaker than the standard zero-knowledge property for general interactive protocols since the standard zero-knowledge considers malicious verifiers that adaptively choose msg₂ rather than fixing it.*

Remark 8 (On High Min-Entropy). *We require the high min-entropy property because this property is needed in the proof of adaptive multi-theorem zero-knowledge property of the NIZK obtained by the Fiat-Shamir transform in Section 5.3. The property requires two requirements: the first is about the structure of Prove₁ and the second is that msg₁ has a high min-entropy. The latter is needed even for Fiat-Shamir transform for Σ -protocols for NP (e.g., see [Unr15]). On the other hand, the former is unique to our work, and we do not know if this is inherent. However, since this requirement makes the security proof of our NIZK easier and our construction of Σ -protocol with quantum preprocessing satisfies this property, we include this as a default requirement.*

Lemma 5.4 (Gap Amplification for Σ -protocol with quantum preprocessing). *If there exists a Σ -protocol with quantum preprocessing for a promise problem L that satisfies c-completeness, s-soundness, special zero-knowledge, and high min-entropy for some $0 < s < c < 1$ such that $c - s > 1/\text{poly}(\lambda)$, then there exists a Σ -protocol with quantum preprocessing for L with $(1 - \text{negl}(\lambda))$ -completeness, $\text{negl}(\lambda)$ -soundness, special zero-knowledge, and high min-entropy.*

Proof. It is clear that the parallel repetition can amplify the completeness-soundness gap, and that the high min-entropy is preserved under the parallel repetition. We can also show that parallel repetition preserves the special zero-knowledge property by a standard hybrid argument. \square

Theorem 5.5. *If a non-interactive commitment scheme exists, then there exists a Σ -protocol with quantum preprocessing for QMA.*

As mentioned in Section 5.1, a non-interactive commitment scheme unconditionally exists in the QROM. Therefore, the above theorem implies the following corollary.

Corollary 5.6. *There exists a Σ -protocol with quantum preprocessing for QMA in the QROM.*

Proof of Theorem 5.5. Let $L = (L_{\text{yes}}, L_{\text{no}})$ be a QMA promise problem, and $\mathcal{H}_x, N, M, p_i, s_i, P_i, \alpha, \beta$, and ρ_{hist} be as in Lemma 2.7 for the promise problem L . We let $N' := 3^5 \sum_{i=1}^5 \binom{N}{i}$ similarly to Lemma 3.2. Let $\Pi_{\text{comm}} = (\text{Commit}_{\text{comm}}, \text{Verify}_{\text{comm}})$ be a non-interactive commitment scheme as defined in Definition 5.2. Then our Σ -protocol with quantum preprocessing $\Pi_{\Sigma} = (\Sigma.\text{Preprocess}, \Sigma.\text{Prove}_1, \Sigma.\text{Verify}_1, \Sigma.\text{Prove}_2, \Sigma.\text{Verify}_2)$ for L is described in Figure 5.

Lemma 5.7. Π_{Σ} satisfies $(1 - \frac{\alpha}{N'})$ -completeness and $(1 - \frac{\beta}{N'} + \text{negl}(\lambda))$ -soundness.

Proof. Let us consider the virtual protocol Π'_{Σ} given in Fig. 6. Due to Lemma 2.2 and the fact that the measurements by the prover and the verifier are commute with each other, the acceptance probability in Π'_{Σ} is equal to that in Π_{Σ} . We therefore have only to show the $(1 - \frac{\alpha}{N'})$ -completeness and $(1 - \frac{\beta}{N'} + \text{negl}(\lambda))$ -soundness for the virtual protocol Π'_{Σ} .

First let us show the completeness. If the prover is honest, it is clear that the history state with Pauli byproducts, $(\prod_{j \in [N]} X_j^{x_j} Z_j^{z_j}) \rho_{\text{hist}} (\prod_{j \in [N]} X_j^{x_j} Z_j^{z_j})$, is teleported to the verifier, and the verifier can correct the byproducts on S (with probability one from the perfect completeness of the commitment scheme). From Lemma 2.5 and Lemma 2.7, and the fact that the probability that P_i is consistent to $(S, \{W_j\}_{j \in S})$ and the coin tails is $1/N'$, we obtain the acceptance probability in Π'_{Σ} when $\mathbf{x} \in L_{\text{yes}}$ to be

$$\left(1 - \frac{1}{N'}\right) + \frac{1}{N'} \left[1 - \text{Tr}(\mathcal{H}_{\mathbf{x}} \rho_{\text{hist}})\right] \geq 1 - \frac{\alpha}{N'}.$$

We have therefore shown the $(1 - \frac{\alpha}{N'})$ -completeness.

Next let us show the soundness. The malicious prover first does any POVM measurement on ρ_P to get $\text{msg}_1 = \{\text{com}_j\}_{j \in [N]}$, and sends it to the verifier. After receiving S from the verifier, the prover does another POVM measurement on the remaining state st to get msg_3 , and sends it to the verifier. The verifier therefore measures all qubits of the N -qubit state $(\prod_{j \in S} X_j^{x_j} Z_j^{z_j}) \rho (\prod_{j \in S} X_j^{x_j} Z_j^{z_j})$, where ρ is the state of the register V after the prover does the first POVM measurement, and $\{x_j, z_j\}_{j \in S}$ is that in msg_3 . Note that ρ is independent of S , because the first POVM measurement is done before S is given to the prover. Due to the binding of the commitment scheme, each com_j can be opened to a unique value (\hat{x}_j, \hat{z}_j) or rejected by $\text{Verify}_{\text{comm}}$. We can assume that the prover always sends correct msg_3 so that all $\{\text{com}_j\}_{j \in [S]}$ are accepted by $\text{Verify}_{\text{comm}}$, because otherwise the prover is rejected. Therefore, it is equivalent that the verifier measures the energy of $\mathcal{H}_{\mathbf{x}}$ on the N -qubit state $\hat{\rho} := (\prod_{j \in [N]} X_j^{\hat{x}_j} Z_j^{\hat{z}_j}) \rho (\prod_{j \in [N]} X_j^{\hat{x}_j} Z_j^{\hat{z}_j})$. Because $\{\hat{x}_j, \hat{z}_j\}_{j \in [N]}$ is fixed before S is chosen, $\hat{\rho}$ is independent of S . Then due to Lemma 2.5 and Lemma 2.7, the acceptance probability in Π'_{Σ} when $\mathbf{x} \in L_{\text{no}}$ is at most

$$\left(1 - \frac{1}{N'}\right) + \frac{1}{N'} \left[1 - \text{Tr}(\mathcal{H}_{\mathbf{x}} \hat{\rho})\right] + \text{negl}(\lambda) \leq 1 - \frac{\beta}{N'} + \text{negl}(\lambda).$$

Σ .Preprocess(1^λ): It chooses $(W_1, \dots, W_N) \xleftarrow{\$} \{X, Y, Z\}^N$ and $(m_1, \dots, m_N) \xleftarrow{\$} \{0, 1\}^N$, and outputs a proving key $k_P := \rho_P := \bigotimes_{j=1}^N (U(W_j)|m_j\rangle)$ and a verification key $k_V := (W_1, \dots, W_N, m_1, \dots, m_N)$.

Σ .Prove₁($k_P, \mathbf{x}, \mathbf{w}$): It parses $\rho_P \leftarrow k_P$, and generates the history state ρ_{hist} for $\mathcal{H}_{\mathbf{x}}$ from \mathbf{w} . It measures j -th qubits of ρ_{hist} and ρ_P in the Bell basis for $j \in [N]$. Let $x := x_1 \| x_2 \| \dots \| x_N$, and $z := z_1 \| z_2 \| \dots \| z_N$ where $(x_j, z_j) \in \{0, 1\}^2$ denotes the outcome of j -th measurement. It computes $\text{Commit}_{\text{comm}}(1^\lambda, (x_j, z_j)) \rightarrow (\text{com}_j, d_j)$ for each $j \in [N]$. It outputs a classical message $\text{msg}_1 := \{\text{com}_j\}_{j \in [N]}$ and the state st , which is its entire final state.

Σ .Verify₁(1^λ): It chooses a subset $S \subset [N]$ such that $1 \leq |S| \leq 5$ uniformly at random, and outputs $\text{msg}_2 := S$.

Σ .Prove₂(st, msg_2): It parses st as the final entire state of Σ .Prove₁ and $\text{msg}_2 \leftarrow S$. It outputs $\text{msg}_3 := (\{d_j\}_{j \in S}, \{x_j, z_j\}_{j \in S})$.

Σ .Verify₂($k_V, \mathbf{x}, \text{msg}_1, \text{msg}_2, \text{msg}_3$): It parses $(W_1, \dots, W_N, m_1, \dots, m_N) \leftarrow k_V$, $\{\text{com}_j\}_{j \in [N]} \leftarrow \text{msg}_1$, $S \leftarrow \text{msg}_2$, and $(\{d_j\}_{j \in S}, \{x_j, z_j\}_{j \in S}) \leftarrow \text{msg}_3$. It computes $\text{Verify}_{\text{comm}}(1^\lambda, (x_j, z_j), \text{com}_j, d_j)$ for all $j \in S$. If not all outputs are \top , it outputs \perp and aborts. It chooses $i \in [M]$ according to the probability distribution defined by $\{p_i\}_{i \in [M]}$ (i.e., chooses i with probability p_i). Let

$$S_i := \{j \in [N] \mid j\text{th Pauli operator of } P_i \text{ is not } I\}.$$

We note that we have $1 \leq |S_i| \leq 5$ by the 5-locality of $\mathcal{H}_{\mathbf{x}}$. We say that P_i is consistent to $(S, \{W_j\}_{j \in S})$ if and only if $S_i = S$ and the j th Pauli operator of P_i is W_j for all $j \in S_i$. If P_i is not consistent to $(S, \{W_j\}_{j \in S})$, it outputs \perp . If P_i is consistent to $(S, \{W_j\}_{j \in S})$, it flips a biased coin that heads with probability $1 - 3^{|S_i| - 5}$. If heads, it outputs \top . If tails, it defines

$$m'_j := \begin{cases} m_j \oplus x_j & (W_j = Z), \\ m_j \oplus z_j & (W_j = X), \\ m_j \oplus x_j \oplus z_j & (W_j = Y) \end{cases}$$

for $j \in S_i$, and outputs \top if $(-1)^{\bigoplus_{j \in S_i} m'_j} = -s_i$ and \perp otherwise.

Figure 5: Σ -protocol with quantum preprocessing Π_Σ .

Σ .Preprocess(1^λ): It generates N Bell-pairs between registers P and V . Let ρ_P and ρ_V be quantum states in registers P and V , respectively. It outputs a proving key $k_P := \rho_P$ and a verification key $k_V := \rho_V$.

Σ .Prove $_1(k_P, \mathbf{x}, \mathbf{w})$: The same as that of Π_Σ .

Σ .Verify $_1(1^\lambda)$: The same as that of Π_Σ .

Σ .Prove $_2(\text{st}, \text{msg}_2)$: The same as that of Π_Σ .

Σ .Verify $_2(k_V, \mathbf{x}, \text{msg}_1, \text{msg}_2, \text{msg}_3)$: It parses $\rho_V \leftarrow k_V$, $\{\text{com}_j\}_{j \in [N]} \leftarrow \text{msg}_1$, $S \leftarrow \text{msg}_2$, and $(\{d_j\}_{j \in S}, \{x_j, z_j\}_{j \in S}) \leftarrow \text{msg}_3$. It computes $\text{Verify}_{\text{comm}}(1^\lambda, (x_j, z_j), \text{com}_j, d_j)$ for all $j \in S$. If not all outputs are \top , it outputs \perp and aborts. It chooses $(W_1, \dots, W_N) \stackrel{\$}{\leftarrow} \{X, Y, Z\}^N$. It generates $(\prod_{j \in S} X_j^{x_j} Z_j^{z_j}) \rho_V (\prod_{j \in S} X_j^{x_j} Z_j^{z_j})$, and measures its j th qubit in the W_j -basis for every $j \in [N]$. Let $m_j \in \{0, 1\}$ be the measurement result for the j th qubit. It chooses $i \in [M]$ according to the probability distribution defined by $\{p_i\}_{i \in [M]}$ (i.e., chooses i with probability p_i). Let

$$S_i := \{j \in [N] \mid j\text{th Pauli operator of } P_i \text{ is not } I\}.$$

We note that we have $1 \leq |S_i| \leq 5$ by the 5-locality of \mathcal{H}_x . We say that P_i is consistent to $(S, \{W_j\}_{j \in S})$ if and only if $S_i = S$ and the j th Pauli operator of P_i is W_j for all $j \in S_i$. If P_i is not consistent to $(S, \{W_j\}_{j \in S})$, it outputs \top . If P_i is consistent to $(S, \{W_j\}_{j \in S})$, it flips a biased coin that heads with probability $1 - 3^{|S_i| - 5}$. If heads, it outputs \top and aborts. If tails, it outputs \top if $(-1)^{\bigoplus_{j \in S_i} m_j} = -s_i$ and \perp otherwise.

Figure 6: The virtual protocol Π'_Σ for Σ -protocol with quantum preprocessing Π_Σ .

For any \mathbf{x} , let \mathbf{E}_x be the event that the statement output by \mathcal{A}_1 is \mathbf{x} . Then,

$$\Pr[\mathbf{x} \in L_{\text{no}} \wedge \text{verifier outputs } \top] \leq \sum_{\mathbf{x} \in L_{\text{no}}} \Pr[\mathbf{E}_x] \left(1 - \frac{\beta}{N'} + \text{negl}(\lambda)\right) \leq \left(1 - \frac{\beta}{N'} + \text{negl}(\lambda)\right).$$

We have therefore shown the $\left(1 - \frac{\beta}{N'} + \text{negl}(\lambda)\right)$ -soundness. \square

Lemma 5.8. Π_Σ satisfies special zero-knowledge property.

Proof. We construct the simulator Sim_Σ as follows.

$\text{Sim}_\Sigma(k_P, \mathbf{x}, \text{msg}_2)$: It parses $\text{msg}_2 = S$ and generates a quantum state $\rho_S := \text{Sim}_{\text{hist}}(\mathbf{x}, S)$ using Sim_{hist} in Lemma 2.7. Then it measures the corresponding qubits of ρ_S and ρ_P in the Bell basis. Let $\{x_j, z_j\}_{j \in S}$ be the measurement outcomes. It computes $\text{Commit}_{\text{comm}}(1^\lambda, (x_j, z_j)) \rightarrow (\text{com}_j, d_j)$ for each $j \in S$ and $\text{Commit}_{\text{comm}}(1^\lambda, (0, 0)) \rightarrow (\text{com}_j, d_j)$ for each $j \in [N] \setminus S$. It outputs $\text{msg}_1 := \{\text{com}_j\}_{j \in [N]}$ and $\text{msg}_3 = (\{d_j\}_{j \in S}, \{x_j, z_j\}_{j \in S})$.

In the following, we prove that the above simulator satisfies the requirement of the special zero-knowledge. For proving this, we consider the following sequence of modified versions of Sim_Σ , which take $\mathbf{w} \in R_L(\mathbf{x})$ as an additional input.

$\text{Sim}_\Sigma^{(1)}(k_P, \mathbf{x}, \mathbf{w}, \text{msg}_2)$: This simulator works similarly to Sim_Σ except that it first generates the history state ρ_{hist} and then uses the corresponding part of ρ_{hist} instead of ρ_S . Note that this simulator can generate the history state since it takes \mathbf{w} as input.

$\text{Sim}_\Sigma^{(2)}(k_P, \mathbf{x}, \mathbf{w}, \text{msg}_2)$: This simulator works similarly to $\text{Sim}_\Sigma^{(1)}$ except that it measures j -th qubits of ρ_{hist} and ρ_P for all $j \in [N]$ (rather than only for $j \in S$) and gets the measurement outcomes $\{x_j, z_j\}_{j \in [N]}$. Note that this simulator generates the commitments in the same way as $\text{Sim}_\Sigma^{(1)}$.

$\text{Sim}_\Sigma^{(3)}(k_P, \mathbf{x}, \mathbf{w}, \text{msg}_2)$: This simulator works similarly to $\text{Sim}_\Sigma^{(2)}$ except that it generates $\text{Commit}_{\text{comm}}(1^\lambda, (x_j, z_j)) \rightarrow (\text{com}_j, d_j)$ for all $j \in [N]$.

Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a QPT adversary. For notational simplicity, we let $\text{Sim}_\Sigma^{(0)} := \text{Sim}_\Sigma$,

$$p_0 := \Pr \left[\mathcal{A}_2(\text{st}_{\mathcal{A}}, \mathbf{x}, \text{msg}_1, \text{msg}_2, \text{msg}_3) = 1 : \begin{array}{l} (k_P, \text{st}_{\mathcal{A}}) \xleftarrow{\$} \mathcal{A}_1(1^\lambda) \\ (\text{msg}_1, \text{msg}_3) \xleftarrow{\$} \text{Sim}_\Sigma(k_P, \mathbf{x}, \text{msg}_2) \end{array} \right],$$

$$p_i := \Pr \left[\mathcal{A}_2(\text{st}_{\mathcal{A}}, \mathbf{x}, \text{msg}_1, \text{msg}_2, \text{msg}_3) = 1 : \begin{array}{l} (k_P, \text{st}_{\mathcal{A}}) \xleftarrow{\$} \mathcal{A}_1(1^\lambda) \\ (\text{msg}_1, \text{msg}_3) \xleftarrow{\$} \text{Sim}_\Sigma^{(i)}(k_P, \mathbf{x}, \mathbf{w}, \text{msg}_2) \end{array} \right]$$

for $i = 1, 2, 3$, and

$$p_{\text{real}} := \Pr \left[\mathcal{A}_2(\text{st}_{\mathcal{A}}, \mathbf{x}, \text{msg}_1, \text{msg}_2, \text{msg}_3) = 1 : \begin{array}{l} (k_P, \text{st}_{\mathcal{A}}) \xleftarrow{\$} \mathcal{A}_1(1^\lambda) \\ (\text{msg}_1, \text{st}) \xleftarrow{\$} \text{Prove}_1(k_P, \mathbf{x}, \mathbf{w}) \\ \text{msg}_3 \xleftarrow{\$} \text{Prove}_2(\text{st}, \text{msg}_2) \end{array} \right].$$

What we have to prove is $|p_{\text{real}} - p_0| = \text{negl}(\lambda)$. We prove this by the following claims.

Claim 5.9. $|p_0 - p_1| \leq \text{negl}(\lambda)$.

Proof. By Lemma 2.7, $\|\rho_S - \text{Tr}_{[N] \setminus S} \rho_{\text{hist}}\|_{\text{tr}} = \text{negl}(\lambda)$. The claim immediately follows from this. \square

Claim 5.10. $p_1 = p_2$.

Proof. This immediately follows from the fact that the measurement results corresponding to $j \in [N] \setminus S$ are not used, which is equivalent to tracing out all qubits of ρ_{hist} in $[N] \setminus S$. \square

Claim 5.11. $|p_2 - p_3| \leq \text{negl}(\lambda)$.

Proof. This follows from a straightforward reduction to the computational hiding property of the commitment scheme. \square

Claim 5.12. $p_3 = p_{\text{real}}$.

Proof. This claim clearly holds since $\text{Sim}_\Sigma^{(3)}$ generates msg_1 and msg_3 in exactly the same way as by the real proving algorithm. \square

By combining the above claims, we have $|p_{\text{real}} - p_0| \leq \text{negl}(\lambda)$. This completes the proof of Lemma 5.8. \square

Lemma 5.13. Π_Σ satisfies high min-entropy property.

Preprocess_{QRO}(1^λ): It runs $\Sigma.\text{Preprocess}(1^\lambda) \rightarrow (\Sigma.k_V, \Sigma.k_P)$, and outputs $k_V := \Sigma.k_V$ and $k_P := \Sigma.k_P$.

Prove_{QRO}^H($k_P, \mathbf{x}, \mathbf{w}^{\otimes k}$): It parses $\Sigma.k_P \leftarrow k_P$, and runs $\Sigma.\text{Prove}_1(k_P, \mathbf{x}, \mathbf{w}^{\otimes k}) \rightarrow (\text{msg}_1, \text{st})$. It computes $\text{msg}_2 := H(\mathbf{x}, \text{msg}_1)$. It runs $\Sigma.\text{Prove}_2(\text{st}, \text{msg}_2) \rightarrow \text{msg}_3$. It outputs $\pi := (\text{msg}_1, \text{msg}_3)$.

Verify_{QRO}^H(k_V, \mathbf{x}, π): It parses $\Sigma.k_V \leftarrow k_V$ and $(\text{msg}_1, \text{msg}_3) \leftarrow \pi$. It computes $\Sigma.\text{Verify}_2(k_V, \mathbf{x}, \text{msg}_1, H(\mathbf{x}, \text{msg}_1), \text{msg}_3)$. If the output is \perp , it outputs \perp . If the output is \top , it outputs \top .

Figure 7: CV-NIZK in the QRO + ($V \rightarrow P$) model Π_{QRO} .

Proof. We define $\Sigma.\text{Prove}_1^Q$ to be the part of $\Sigma.\text{Prove}_1$ that generates $\{x_j, z_j\}_{j \in [N]}$ by the Bell basis measurements and $\Sigma.\text{Prove}_1^C$ to be the rest of $\Sigma.\text{Prove}_1$. By the computational hiding property of the commitment, a commitment does not take a fixed value with non-negligible probability. Then it is clear that $\text{msg}_1 = \{\text{com}_j\}_{j \in [N]}$ does not take a fixed value with non-negligible probability. \square

\square

5.3 Construction

In this section, we construct a CV-NIZK in the QRO + ($V \rightarrow P$) model. As a result, we obtain the following theorem.

Theorem 5.14. *There exists a CV-NIZK for QMA in the QRO + ($V \rightarrow P$) model.*

Let $L = (L_{\text{yes}}, L_{\text{no}})$ be a QMA promise problem, H be a random oracle, and $\Pi_\Sigma = (\Sigma.\text{Preprocess}, \Sigma.\text{Prove}_1, \Sigma.\text{Verify}_1, \Sigma.\text{Prove}_2, \Sigma.\text{Verify}_2)$ be a Σ -protocol with quantum preprocessing (with $(1 - \text{negl}(\lambda))$ -completeness and $\text{negl}(\lambda)$ -soundness). Then our CV-NIZK in the QRO + ($V \rightarrow P$) model $\Pi_{\text{QRO}} = (\text{Preprocess}_{\text{QRO}}, \text{Prove}_{\text{QRO}}, \text{Verify}_{\text{QRO}})$ for L is described in Figure 7.

Lemma 5.15. Π_{QRO} satisfies $(1 - \text{negl}(\lambda))$ -completeness and adaptive $\text{negl}(\lambda)$ -soundness.

Proof of Lemma 5.15. The completeness is clear. For proving soundness, we rely on the following lemma shown in [DFM20].

Lemma 5.16 ([DFM20, Theorem 2]). *Let X and Y be non-empty sets and \mathcal{A} be an arbitrary oracle quantum algorithm that takes as input a quantum state ρ , makes q queries to a uniformly random $H : X \rightarrow Y$, and outputs some $x \in X$ and a (possibly quantum) output z . There exist black-box quantum algorithms \mathcal{S}_1^A and \mathcal{S}_2^A such that for any quantum input ρ , $x^* \in X$, and any predicate V :*

$$\begin{aligned} & \Pr_H [x = x^* \wedge V(x, H(x), z) : (x, z) \leftarrow \mathcal{A}^H(\rho)] \\ & \leq (2q + 1)^2 \Pr_y \left[x = x^* \wedge V(x, y, z) : \begin{array}{l} (x, \text{st}) \leftarrow \mathcal{S}_1^A(\rho) \\ z \leftarrow \mathcal{S}_2^A(\text{st}, y) \end{array} \right] \end{aligned}$$

Furthermore, \mathcal{S}_1^A and \mathcal{S}_2^A run in time polynomial in q , $\log |X|$, and $\log |Y|$.

Based on the above lemma, we prove the soundness of Π_Σ as follows:

$$\begin{aligned}
& \Pr_{H, (k_P, k_V)} \left[\mathbf{x} \in L_{\text{no}} \wedge \text{Verify}_{\text{QRO}}^H(k_V, \mathbf{x}, \pi) = \top : (\mathbf{x}, \pi) \leftarrow \mathcal{A}^H(k_P) \right] \\
&= \Pr_{H, (k_P, k_V)} \left[\begin{array}{c} \mathbf{x} \in L_{\text{no}} \\ \wedge \\ \Sigma.\text{Verify}_2(k_V, \mathbf{x}, \text{msg}_1, H(\text{msg}_1), \text{msg}_3) = \top \end{array} : (\mathbf{x}, (\text{msg}_1, \text{msg}_3)) \leftarrow \mathcal{A}^H(k_P) \right] \\
&= \mathbb{E}_{(k_P^*, k_V^*)} \Pr_H \left[\begin{array}{c} \mathbf{x} \in L_{\text{no}} \\ \wedge \\ \Sigma.\text{Verify}_2(k_V^*, \mathbf{x}, \text{msg}_1, H(\text{msg}_1), \text{msg}_3) = \top \end{array} : (\mathbf{x}, (\text{msg}_1, \text{msg}_3)) \leftarrow \mathcal{A}^H(k_P^*) \right] \\
&= \mathbb{E}_{(k_P^*, k_V^*)} \sum_{\mathbf{x}^* \in L_{\text{no}}, \text{msg}_1^*} \\
& \Pr_H \left[\begin{array}{c} (\mathbf{x}, \text{msg}_1) = (\mathbf{x}^*, \text{msg}_1^*) \\ \wedge \\ \Sigma.\text{Verify}_2(k_V^*, \mathbf{x}, \text{msg}_1, H(\text{msg}_1), \text{msg}_3) = \top \end{array} : (\mathbf{x}, (\text{msg}_1, \text{msg}_3)) \leftarrow \mathcal{A}^H(k_P^*) \right] \\
&\leq (2q+1)^2 \mathbb{E}_{(k_P^*, k_V^*)} \sum_{\mathbf{x}^* \in L_{\text{no}}, \text{msg}_1^*} \\
& \Pr_{\text{msg}_2} \left[\begin{array}{c} (\mathbf{x}, \text{msg}_1) = (\mathbf{x}^*, \text{msg}_1^*) \\ \wedge \\ \Sigma.\text{Verify}_2(k_V^*, \mathbf{x}, \text{msg}_1, \text{msg}_2, \text{msg}_3) = \top \end{array} : \begin{array}{l} (\mathbf{x}, \text{msg}_1, \text{st}) \stackrel{\$}{\leftarrow} \mathcal{S}_1^A(k_P^*) \\ \text{msg}_3 \stackrel{\$}{\leftarrow} \mathcal{S}_2^A(\text{st}, \text{msg}_2) \end{array} \right] \\
&= (2q+1)^2 \Pr_{\text{msg}_2, (k_P, k_V)} \left[\begin{array}{c} \mathbf{x} \in L_{\text{no}} \\ \wedge \\ \Sigma.\text{Verify}_2(k_V, \mathbf{x}, \text{msg}_1, \text{msg}_2, \text{msg}_3) = \top \end{array} : \begin{array}{l} (\mathbf{x}, \text{msg}_1, \text{st}) \stackrel{\$}{\leftarrow} \mathcal{S}_1^A(k_P) \\ \text{msg}_3 \stackrel{\$}{\leftarrow} \mathcal{S}_2^A(\text{st}, \text{msg}_2) \end{array} \right] \\
&\leq (2q+1)^2 \text{negl}(\lambda) \\
&= \text{negl}(\lambda)
\end{aligned}$$

where the first inequality is obtained by applying Lemma 5.16 for each fixed (k_P^*, k_V^*) with $\rho := k_P^*$, $x := (\mathbf{x}, \text{msg}_1)$, $y := \text{msg}_2$, $z := \text{msg}_3$, and $V((\mathbf{x}, \text{msg}_1), \text{msg}_2, \text{msg}_3) := (\Sigma.\text{Verify}_2(k_V^*, \mathbf{x}, \text{msg}_1, \text{msg}_2, \text{msg}_3) \stackrel{?}{=} \top)$ and the second inequality follows from the soundness of Π_Σ . \square

Lemma 5.17. Π_{QRO} satisfies adaptive multi-theorem zero-knowledge property.

Proof of Lemma 5.17. For proving the zero-knowledge property, we use the following lemma.

Lemma 5.18 (Adaptive Reprogramming [GHHM20]). *Let X_1, X_2, X', Y be some finite sets. For an algorithm \mathcal{A} , we consider the following experiment for $b \in \{0, 1\}$:*

$\text{Exp}_b^{\mathcal{A}}$: *The experiment first uniformly chooses a function $H : X_1 \times X_2 \rightarrow Y$, which may be updated during the execution of the experiment. \mathcal{A} can make the following two types of queries:*

Random Oracle Query: *When \mathcal{A} queries $(x_1, x_2) \in X_1 \times X_2$, the oracle returns $H(x)$. This oracle can be accessed quantumly (i.e., upon a query $\sum_{x_1, x_2, y} |x_1, x_2\rangle |y\rangle$, the oracle returns $\sum_{x_1, x_2, y} |x_1, x_2\rangle |y \oplus H(x_1, x_2)\rangle$).*

Reprogramming Query: *A reprogramming query should consist of $x_1 \in X_1$ and a description of a probabilistic distribution D over $X_2 \times X'$. On input (x_1, D) , the oracle works as follows. First, the oracle takes $(x_2, x') \stackrel{\$}{\leftarrow} D$ and $y \stackrel{\$}{\leftarrow} Y$. Then it does either of the following depending of the value of b .*

1. If $b = 0$, it does nothing.
2. If $b = 1$, it reprograms H so that $H(x_1, x_2) = y$. Note that the reprogrammed H is used for answering random queries hereafter.

Finally, it returns (x_2, x') to \mathcal{A} . Note that this algorithm is only classically accessed.

After making an arbitrary number of queries to the above oracles, \mathcal{A} finally outputs a bit b' , which is treated as the output of the experiment.

Suppose that \mathcal{A} makes at most q_H random oracle queries and at most q_R reprogramming queries and let $p_{max} := \max_{D, x_2^*} \Pr[x_2 = x_2^* : (x_2, x') \stackrel{\$}{\leftarrow} D]$ where the maximum is taken over all D queried by \mathcal{A} as part of a reprogramming query and $x_2^* \in X_2$. Then we have

$$|\Pr[\text{Exp}_0^{\mathcal{A}} = 1] - \Pr[\text{Exp}_1^{\mathcal{A}} = 1]| \leq \frac{3q_R}{2} \sqrt{q_H p_{max}}$$

Remark 9. The above lemma is a special case of [GHHM20, Theorem 1]. We note that the roles of X_1 and X_2 are swapped from the original one, but this is just for convenience in later use and does not make any essential difference. We also note that a similar special case is stated in [GHHM20, Proposition 2], but the above lemma is slightly more general than that since their proposition assumes that \mathcal{A} uses the same D for all reprogramming queries.

Proof. Let Sim_Σ be the simulator for Π_Σ . We construct a simulator Sim_{QRO} for Π_{QRO} as follows where \mathcal{C} is the set from which msg_2 is uniformly chosen.

$\text{Sim}_{\text{QRO}}^{H, \text{Reprogram}}(k_P, \mathbf{x})$: It randomly chooses $\text{msg}_2 \stackrel{\$}{\leftarrow} \mathcal{C}$, generates $(\text{msg}_1, \text{msg}_3) \stackrel{\$}{\leftarrow} \text{Sim}_\Sigma(k_P, \mathbf{x}, \text{msg}_2)$, queries $((\mathbf{x}, \text{msg}_1), \text{msg}_2)$ to Reprogram , which reprograms H so that $H(\mathbf{x}, \text{msg}_1) = \text{msg}_2$, and outputs $(\text{msg}_1, \text{msg}_3)$.

In the following, we prove that the above simulator satisfies the requirement for adaptive multi-theorem zero-knowledge. For proving this, we consider the following sequence of modified versions of Sim_{QRO} , which take k copies of a witness $\mathbf{w} \in R_L(\mathbf{x})$ as an additional input.

$\text{Sim}_{\text{QRO}}^{(1) H, \text{Reprogram}}(k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$: This simulator uses the real proving algorithm instead of the simulator to generate msg_1 and msg_3 . That is, it generates $(\text{msg}_1, \text{st}) \stackrel{\$}{\leftarrow} \Sigma.\text{Prove}_1(k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$, randomly chooses $\text{msg}_2 \stackrel{\$}{\leftarrow} \mathcal{C}$, generates $\text{msg}_3 \stackrel{\$}{\leftarrow} \Sigma.\text{Prove}_2(\text{st}, \text{msg}_2)$, queries $((\mathbf{x}, \text{msg}_1), \text{msg}_2)$ to Reprogram , which reprograms H so that $H(\mathbf{x}, \text{msg}_1) = \text{msg}_2$, and outputs $(\text{msg}_1, \text{msg}_3)$.

$\text{Sim}_{\text{QRO}}^{(2) H, \text{Reprogram}}(k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$: This simulator derives msg_2 by querying to the random oracle instead of randomly choosing msg_2 and then reprogramming the random oracle to be consistent. That is, it generates $(\text{msg}_1, \text{st}) \stackrel{\$}{\leftarrow} \Sigma.\text{Prove}_1(k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$, sets $\text{msg}_2 := H(\mathbf{x}, \text{msg}_1)$, generates $\text{msg}_3 \stackrel{\$}{\leftarrow} \Sigma.\text{Prove}_2(\text{st}, \text{msg}_2)$, and outputs $(\text{msg}_1, \text{msg}_3)$. Note that this simulator no longer makes a query to Reprogram .

Let \mathcal{D} be a QPT distinguisher. For notational simplicity, let $\mathcal{O}_{S(0)} := \mathcal{O}_S$, $\mathcal{O}_{S(i)}$ be the oracle that works similarly to \mathcal{O}_S except that $\text{Sim}_{\text{QRO}}^{(i)}$ is used instead of Sim_{QRO} for $i = 1, 2$,

$$p_i := \Pr \left[\mathcal{D}^{H, \mathcal{O}_{S(i)}^{H, \text{Reprogram}}(\cdot, \cdot)}(1^\lambda) = 1 : H \stackrel{\$}{\leftarrow} \text{ROdist} \right]$$

for $i = 0, 1, 2$, and

$$p_{\text{real}} := \Pr \left[\mathcal{D}^{H, \mathcal{O}_P^H(\cdot, \cdot)}(1^\lambda) = 1 : H \stackrel{\$}{\leftarrow} \text{ROdist} \right].$$

What we have to prove is $|p_{\text{real}} - p_0| = \text{negl}(\lambda)$. We prove this by the following claims.

Claim 5.19. $|p_0 - p_1| \leq \text{poly}(\lambda)$.

Proof. This claim can be proven by a straightforward reduction to the special zero-knowledge property of Π_Σ and a standard hybrid argument. \square

Claim 5.20. $|p_1 - p_2| \leq \text{poly}(\lambda)$.

Proof. This claim can be proven by a straightforward reduction to Lemma 5.18 where \mathbf{x} , msg_1 , st , and msg_2 play the roles of x_1 , x_2 , x' , and y , respectively, and the output distribution of $\Sigma.\text{Prove}_1^C(\text{st}')$ where $\text{st}' \stackrel{\$}{\leftarrow} \Sigma.\text{Prove}_1^Q(k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$ plays the role of the distribution D . (See Definition 5.3 for the definitions of $\Sigma.\text{Prove}_1^C$ and $\Sigma.\text{Prove}_1^Q$). Since the number of \mathcal{D} 's queries is $\text{poly}(\lambda)$ and msg_1 sampled by $\Sigma.\text{Prove}_1^C(\text{st}')$ does not take any fixed value with non-negligible probability as required by the high min-entropy property of Π_Σ , p_{max} in Lemma 5.18 is negligible. Then Lemma 5.18 directly gives the above claim. \square

Claim 5.21. $p_2 = p_{\text{real}}$.

Proof. This is clear since $\text{Sim}_{\text{QRO}}^{(2)}$ works similarly to the real proving algorithm $\text{Prove}_{\text{QRO}}$. \square

By combining the above claims, we obtain $|p_{\text{real}} - p_0| \leq \text{negl}(\lambda)$. This completes the proof of Lemma 5.17. \square

\square

Shared Bell-pair model. Remark that the verifier of Π_{QRO} just sends a state $\rho_P := \bigotimes_{j=1}^N (U(W_j)|m_j\rangle)$ for $(W_1, \dots, W_N) \stackrel{\$}{\leftarrow} \{X, Y, Z\}^N$ and $(m_1, \dots, m_N) \stackrel{\$}{\leftarrow} \{0, 1\}^N$ while keeping $(W_1, \dots, W_N, m_1, \dots, m_N)$ as a verification key. This step can be done in a non-interactive way if N Bell-pairs are a priori shared between the prover and verifier. That is, the verifier can measure his halves of Bell pairs in a randomly chosen bases (W_1, \dots, W_N) to get measurement outcomes (m_1, \dots, m_N) . Apparently, this does not harm either of soundness or zero-knowledge since the protocol is the same as Π_{QRO} from the view of the prover and the malicious verifier's power is just weaker than that in Π_{QRO} in the sense that it cannot control the quantum state to be sent to the prover. Thus, we obtain the following theorem.

Theorem 5.22. *There exists a CV-NIZK for QMA in the QRO + shared Bell pair model.*

See Appendix F for the formal definition of CV-NIZK for QMA in the QRO + shared Bell pair model.

References

- [AB20] V. Arte and M. Bellare. Dual-Mode NIZKs: Possibility and Impossibility Results for Property Transfer. In *INDOCRYPT 2020*, pages 859–881. 2020.
- [ACGH20] G. Alagic, A. M. Childs, A. B. Grilo, and S.-H. Hung. Non-interactive Classical Verification of Quantum Computation. In *TCC 2020, Part III*, pages 153–180. 2020.
- [BBBV97] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and Weaknesses of Quantum Computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997.
- [BCKM21] J. Bartusek, A. Coladangelo, D. Khurana, and F. Ma. On the Round Complexity of Secure Quantum Computation. In *CRYPTO 2021, Part I*, pages 406–435, Virtual Event, 2021.
- [BCR86] G. Brassard, C. Crépeau, and J.-M. Robert. Information Theoretic Reductions among Disclosure Problems. In *27th FOCS*, pages 168–173. 1986.
- [BCR87] G. Brassard, C. Crépeau, and J.-M. Robert. All-or-Nothing Disclosure of Secrets. In *CRYPTO’86*, pages 234–238. 1987.
- [BD18] Z. Brakerski and N. Döttling. Two-Message Statistically Sender-Private OT from LWE. In *TCC 2018, Part II*, pages 370–390. 2018.
- [BFM88] M. Blum, P. Feldman, and S. Micali. Non-Interactive Zero-Knowledge and Its Applications (Extended Abstract). In *20th ACM STOC*, pages 103–112. 1988.
- [BG20] A. Broadbent and A. B. Grilo. QMA-hardness of Consistency of Local Density Matrices with Applications to Quantum Zero-Knowledge. In *61st FOCS*, pages 196–205. 2020.
- [BHY09] M. Bellare, D. Hofheinz, and S. Yilek. Possibility and Impossibility Results for Encryption and Commitment Secure under Selective Opening. In *EUROCRYPT 2009*, pages 1–35. 2009.
- [BJSW20] A. Broadbent, Z. Ji, F. Song, and J. Watrous. Zero-Knowledge Proof Systems for QMA. *SIAM J. Comput.*, 49(2):245–283, 2020.
- [BM21] J. Bartusek and G. Malavolta. Candidate Obfuscation of Null Quantum Circuits and Witness Encryption for QMA. *IACR Cryptology ePrint Archive*, 2021:421, 2021.
- [BS20] N. Bitansky and O. Shmueli. Post-quantum zero knowledge in constant rounds. In *52nd ACM STOC*, pages 269–279. 2020.
- [BY20] Z. Brakerski and H. Yuen. Quantum Garbled Circuits, arXiv:2006.01085, 2020.
- [Can20] R. Canetti. Universally Composable Security. *J. ACM*, 67(5):28:1–28:94, 2020.
- [CCKV08] A. Chailloux, D. F. Ciocan, I. Kerenidis, and S. P. Vadhan. Interactive and Noninteractive Zero Knowledge are Equivalent in the Help Model. In *TCC 2008*, pages 501–534. 2008.
- [CM16] T. S. Cubitt and A. Montanaro. Complexity Classification of Local Hamiltonian Problems. *SIAM J. Comput.*, 45(2):268–316, 2016.

- [CNs07] J. Camenisch, G. Neven, and a. shelat. Simulatable Adaptive Oblivious Transfer. In *EUROCRYPT 2007*, pages 573–590. 2007.
- [CVZ20] A. Coladangelo, T. Vidick, and T. Zhang. Non-interactive Zero-Knowledge Arguments for QMA, with Preprocessing. In *CRYPTO 2020, Part III*, pages 799–828. 2020.
- [DFM20] J. Don, S. Fehr, and C. Majenz. The Measure-and-Reprogram Technique 2.0: Multi-round Fiat-Shamir and More. In *CRYPTO 2020, Part III*, pages 602–631. 2020.
- [DFMS19] J. Don, S. Fehr, C. Majenz, and C. Schaffner. Security of the Fiat-Shamir Transformation in the Quantum Random-Oracle Model. In *CRYPTO 2019, Part II*, pages 356–383. 2019.
- [DMP90] A. De Santis, S. Micali, and G. Persiano. Non-Interactive Zero-Knowledge with Preprocessing. In *CRYPTO’88*, pages 269–282. 1990.
- [FHM18] J. F. Fitzsimons, M. Hajdušek, and T. Morimae. Post hoc verification with a single prover. *Phys. Rev. Lett.*, 120:040501, 2018.
- [FLS99] U. Feige, D. Lapidot, and A. Shamir. Multiple NonInteractive Zero Knowledge Proofs Under General Assumptions. *SIAM J. Comput.*, 29(1):1–28, 1999.
- [FS87] A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *CRYPTO’86*, pages 186–194. 1987.
- [GHHM20] A. B. Grilo, K. Hövelmanns, A. Hülsing, and C. Majenz. Tight adaptive reprogramming in the QROM, arXiv:2010.15103, 2020.
- [GMR89] S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM J. Comput.*, 18(1):186–208, 1989.
- [GOS12] J. Groth, R. Ostrovsky, and A. Sahai. New Techniques for Noninteractive Zero-Knowledge. *J. ACM*, 59(3):11:1–11:35, 2012.
- [GS12] J. Groth and A. Sahai. Efficient Noninteractive Proof Systems for Bilinear Groups. *SIAM J. Comput.*, 41(5):1193–1232, 2012.
- [GSY19] A. B. Grilo, W. Slofstra, and H. Yuen. Perfect Zero Knowledge for Quantum Multi-prover Interactive Proofs. In *60th FOCS*, pages 611–635. 2019.
- [IKLP06] Y. Ishai, E. Kushilevitz, Y. Lindell, and E. Petrank. Black-box constructions for secure computation. In *38th ACM STOC*, pages 99–108. 2006.
- [Kob03] H. Kobayashi. Non-interactive Quantum Perfect and Statistical Zero-Knowledge. In *Algorithms and Computation, 14th International Symposium, ISAAC 2003, Kyoto, Japan, December 15-17, 2003, Proceedings*, pages 178–188. 2003.
- [Lin08] A. Y. Lindell. Efficient Fully-Simulatable Oblivious Transfer. In *CT-RSA 2008*, pages 52–70. 2008.
- [LS19] A. Lombardi and L. Schaeffer. A Note on Key Agreement and Non-Interactive Commitments. Cryptology ePrint Archive, Report 2019/279, 2019. <https://eprint.iacr.org/2019/279>.

- [LZ19] Q. Liu and M. Zhandry. Revisiting Post-quantum Fiat-Shamir. In *CRYPTO 2019, Part II*, pages 326–355. 2019.
- [Mah18] U. Mahadev. Classical Homomorphic Encryption for Quantum Circuits. In *59th FOCS*, pages 332–338. 2018.
- [MNS18] T. Morimae, D. Nagaj, and N. Schuch. Quantum proofs can be verified using only single-qubit measurements. *Phys. Rev. A*, 93:022326, 2018.
- [MW18] S. Menda and J. Watrous. Oracle Separations for Quantum Statistical Zero-Knowledge, arXiv:1801.08967, 2018.
- [NC00] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
- [NP01] M. Naor and B. Pinkas. Efficient oblivious transfer protocols. In *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA*, pages 448–457. 2001.
- [Pas13] R. Pass. Unprovable Security of Perfect NIZK and Non-interactive Non-malleable Commitments. In *TCC 2013*, pages 334–354. 2013.
- [Ps05] R. Pass and A. shelat. Unconditional Characterizations of Non-interactive Zero-Knowledge. In *CRYPTO 2005*, pages 118–134. 2005.
- [PS19] C. Peikert and S. Shiehian. Noninteractive Zero Knowledge for NP from (Plain) Learning with Errors. In *CRYPTO 2019, Part I*, pages 89–114. 2019.
- [PVW08] C. Peikert, V. Vaikuntanathan, and B. Waters. A Framework for Efficient and Composable Oblivious Transfer. In *CRYPTO 2008*, pages 554–571. 2008.
- [Qua20] W. Quach. UC-Secure OT from LWE, Revisited. In *SCN 20*, pages 192–211. 2020.
- [Reg09] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.
- [RT19] R. Raz and A. Tal. Oracle separation of BQP and PH. In *51st ACM STOC*, pages 13–23. 2019.
- [Shm21] O. Shmueli. Multi-theorem Designated-Verifier NIZK for QMA. In *CRYPTO 2021, Part I*, pages 375–405, Virtual Event, 2021.
- [Unr15] D. Unruh. Non-Interactive Zero-Knowledge Proofs in the Quantum Random Oracle Model. In *EUROCRYPT 2015, Part II*, pages 755–784. 2015.

A More Explanation on Lemma 2.7

Here, we explain how to obtain Lemma 2.7 based on [BG20]. Let $L = (L_{\text{yes}}, L_{\text{no}})$ be any **QMA** promise problem, and $V = U_T \dots U_1$ be its verification circuit, where each U_i is an elementary gate taken from a universal gate set. For $\mathbf{x} \in L_{\text{yes}}$, there exists a witness state $|\psi\rangle$ such that V accepts with probability exponentially close to 1, whereas for $\mathbf{x} \in L_{\text{no}}$, any state makes V accept with probability exponentially small.

As is explained in [BG20], we consider the encoded version of the verification circuit V' with a certain quantum error correcting code. The circuit V' consists of gates from the universal gate set $\{CNOT, T, H, X, Z\}$. From the standard circuit-to-Hamiltonian construction technique, we can construct a local Hamiltonian $H_{\mathbf{x}} := \sum_i H_i$ corresponding to V' . If there is a witness state $|\psi\rangle$ that makes V' accept with probability $1 - \text{negl}(|\mathbf{x}|)$, then the history state

$$\frac{1}{\sqrt{T+1}} \sum_{t \in [T+1]} |0^{T-t} 1^t\rangle \otimes U_t \dots U_1 (\text{Enc}(|\psi\rangle) \otimes |0^A\rangle)$$

has exponentially small energy. Due to the local simulatability, there is an efficient deterministic algorithm that outputs the classical description of a state that is close to the reduced density matrix of the history state on at most five qubits [BG20, GSY19]. If every quantum state $|\psi\rangle$ makes V' reject with probability at least ϵ , then the groundenergy of H is at least $\Omega(\frac{\epsilon}{T^3})$.

Let $\mathcal{H}_{\mathbf{x}} = \sum_{i=1}^M c_i P_i$ be the local Hamiltonian, where $M = \text{poly}(|\mathbf{x}|)$, c_i is real, and P_i is a tensor product of Pauli operators (I, X, Y, Z) . In the standard circuit-to-Hamiltonian construction, each P_i is a tensor product of at most five non-trivial Pauli operators (X, Y, Z) . As is shown in [MNS18], this Hamiltonian can be changed to the form of $\sum_{i=1}^M p_i \frac{I + s_i P_i}{2}$ with $M = \text{poly}(|\mathbf{x}|)$, $s_i \in \{+1, -1\}$, $p_i > 0$, $\sum_{i=1}^M p_i = 1$, and P_i is a tensor product of Pauli operators (I, X, Y, Z) with at most five non-trivial Pauli operators (X, Y, Z) . In fact, define the normalized Hamiltonian

$$\mathcal{H}'_{\mathbf{x}} := \frac{1}{2} \left(I + \frac{\mathcal{H}_{\mathbf{x}}}{\sum_{i=1}^M |c_i|} \right) = \sum_{i=1}^M \frac{|c_i|}{\sum_{i=1}^M |c_i|} \frac{I + \text{sign}(c_i) P_i}{2},$$

and we have only to take $p_i := \frac{|c_i|}{\sum_{i=1}^M |c_i|}$ and $s_i := \text{sign}(c_i)$.

B More details for the proof of Lemma 3.2

Here we give more details of the completeness and the soundness of the virtual protocol 2. In the virtual protocol 2, $i \in [M]$ is chosen after S_V and (W_1, \dots, W_N) are chosen, but we can assume that i is chosen before S_V and (W_1, \dots, W_N) are chosen, because they are independent. When P_i is not consistent to $(S_V, \{W_j\}_{j \in S_V})$ or the coin heads, the measurement result on ρ'_V is not used. The probability that such cases happen is

$$\begin{aligned} & \sum_{i=1}^M p_i \left(\Pr[\text{not consistent}|i] + \Pr[\text{consistent}|i](1 - 3^{|S_i|-5}) \right) \\ = & \sum_{i=1}^M p_i \left(\frac{3^N \sum_{j=1}^5 \binom{N}{j} - 3^{N-|S_i|}}{3^N \sum_{j=1}^5 \binom{N}{j}} + \frac{3^{N-|S_i|}}{3^N \sum_{j=1}^5 \binom{N}{j}} (1 - 3^{|S_i|-5}) \right) \\ = & \sum_{i=1}^M p_i \left(1 - \frac{1}{3^5 \sum_{j=1}^5 \binom{N}{j}} \right) \\ = & 1 - \frac{1}{3^5 \sum_{j=1}^5 \binom{N}{j}} \\ = & 1 - \frac{1}{N^5}. \end{aligned}$$

The probability that it is consistent and the coin tails is therefore $\frac{1}{N^5}$. In this case, the measurement result on ρ'_V is used. The probability that the measurement result satisfies $(-1)^{\bigoplus_{j \in S_i} m'_j} = -s_i$ is

from Lemma 2.5,

$$\sum_{i=1}^M p_i \text{Tr} \left[\left(I - \frac{I + s_i P_i}{2} \right) \rho'_V \right] = 1 - \text{Tr}(\mathcal{H}_x \rho'_V).$$

The total acceptance probability is therefore

$$1 - \frac{1}{N'} + \frac{1}{N'} \left[1 - \text{Tr}(\mathcal{H}_x \rho'_V) \right] = 1 - \frac{\text{Tr}(\mathcal{H}_x \rho'_V)}{N'}.$$

C Alternative Simpler Construction of CV-NIZK in the QSP Model.

Here, we give an alternative construction of a CV-NIZK in the QSP model, which is slightly simpler than the construction given in Section 3.

Our construction of a CV-NIZK for a **QMA** promise problem L is given in Figure 8 where \mathcal{H}_x , N , M , p_i , s_i , P_i , α , β , and ρ_{hist} are as in Lemma 2.7 for L .

Setup(1^λ): The setup algorithm chooses $(W_1, \dots, W_N, m_1, \dots, m_N) \stackrel{\$}{\leftarrow} \{X, Y, Z\}^N \times \{0, 1\}^N$ and a uniformly random subset $S_V \subseteq [N]$ such that $1 \leq |S_V| \leq 5$, and outputs a proving key $k_P := \bigotimes_{j=1}^N (U(W_j) | m_j \rangle)$ and a verification key $k_V := (S_V, \{W_j, m_j\}_{j \in S_V})$.

Prove($k_P, \mathbf{x}, \mathbf{w}$): The proving algorithm generates the history state ρ_{hist} for \mathcal{H}_x from \mathbf{w} and measures j -th qubits of ρ_{hist} and k_P in the Bell basis for $j \in [N]$. Let $x := x_1 \| x_2 \| \dots \| x_N$, and $z := z_1 \| z_2 \| \dots \| z_N$ where (x_j, z_j) denotes the outcome of j -th measurement. It outputs a proof $\pi := (x, z)$.

Verify(k_V, \mathbf{x}, π): The verification algorithm parses $(S_V, \{W_j, m_j\}_{j \in S_V}) \leftarrow k_V$ and $(x, z) \leftarrow \pi$, chooses $i \in [M]$ according to the probability distribution defined by $\{p_i\}_{i \in [M]}$ (i.e., chooses i with probability p_i). Let

$$S_i := \{j \in [N] \mid j\text{th Pauli operator of } P_i \text{ is not } I\}.$$

We note that we have $1 \leq |S_i| \leq 5$ by the 5-locality of \mathcal{H}_x . We say that P_i is consistent to $(S_V, \{W_j\}_{j \in S_V})$ if and only if $S_i = S_V$ and the j th Pauli operator of P_i is W_j for all $j \in S_i$. If P_i is not consistent to $(S_V, \{W_j\}_{j \in S_V})$, it outputs \perp . If P_i is consistent to $(S_V, \{W_j\}_{j \in S_V})$, it flips a biased coin that heads with probability $1 - 3^{|S_i| - 5}$. If heads, it outputs \perp . If tails, it defines

$$m'_j := \begin{cases} m_j \oplus x_j & (W_j = Z), \\ m_j \oplus z_j & (W_j = X), \\ m_j \oplus x_j \oplus z_j & (W_j = Y) \end{cases}$$

for $j \in S_i$, and outputs \perp if $(-1)^{\bigoplus_{j \in S_i} m'_j} = -s_i$ and \perp otherwise.

Figure 8: CV-NIZK in the QSP model Π'_{NIZK} .

We have the following lemmas.

Lemma C.1 (Completeness and Soundness). Π'_{NIZK} satisfies $(1 - \frac{\alpha}{N'})$ -completeness and $(1 - \frac{\beta}{N'})$ -soundness where $N' := 3^5 \sum_{i=1}^5 \binom{N}{i}$.

Lemma C.2 (Zero-Knowledge). Π'_{NIZK} satisfies the zero-knowledge property.

They can be proven similarly to Lemmas 3.2 and 3.3, respectively.

D CV-NIP in the QSP model

We call a CV-NIZK in the QSP model a CV-NIP (classically-verifiable non-interactive proof) in the QSP model if the zero-knowledge is not satisfied. Here we give a construction of an information-theoretically sound CV-NIP for **QMA** in the QSP model. Specifically, we prove the following theorem.

Theorem D.1. *There exists a CV-NIP for **QMA** in the QSP model (without any computational assumption).*

We note that this theorem is subsumed by Theorem 3.1. Nonetheless, we give a proof of the theorem because the CV-NIP given here is much simpler.

Its proof is based on the fact that the 2-local $\{ZZ, XX\}$ -local Hamiltonian problem is **QMA**-complete. That is, we have the following lemma.

Lemma D.2 (**QMA**-completeness of 2-local $\{ZZ, XX\}$ -Hamiltonian problem [CM16]). *For any **QMA** promise problem $L = (L_{\text{yes}}, L_{\text{no}})$, there is a classical polynomial-time computable deterministic function that maps $\mathbf{x} \in \{0, 1\}^*$ to an N -qubit Hamiltonian $\mathcal{H}_{\mathbf{x}}$ of the form*

$$\mathcal{H}_{\mathbf{x}} = \sum_{j_1 < j_2} \frac{p_{j_1, j_2}}{2} \left(\frac{I + s_{j_1, j_2} X_{j_1} X_{j_2}}{2} + \frac{I + s_{j_1, j_2} Z_{j_1} Z_{j_2}}{2} \right)$$

where $N = \text{poly}(|\mathbf{x}|)$, $p_{j_1, j_2} > 0$, $\sum_{j_1 < j_2} p_{j_1, j_2} = 1$, and $s_{j_1, j_2} \in \{+1, -1\}$, and satisfies the following: There are $0 < \alpha < \beta < 1$ such that $\beta - \alpha = 1/\text{poly}(|\mathbf{x}|)$ and

- if $\mathbf{x} \in L_{\text{yes}}$, then there exists an N -qubit state ρ such that $\text{Tr}(\rho \mathcal{H}_{\mathbf{x}}) \leq \alpha$, and
- if $\mathbf{x} \in L_{\text{no}}$, then for any N -qubit state ρ , we have $\text{Tr}(\rho \mathcal{H}_{\mathbf{x}}) \geq \beta$.

Moreover, for any $\mathbf{x} \in L_{\text{yes}}$, we can convert any witness $\mathbf{w} \in R_L(\mathbf{x})$ into a state ρ_{hist} , called the history state, such that $\text{Tr}(\rho_{\text{hist}} \mathcal{H}_{\mathbf{x}}) \leq \alpha$ in quantum polynomial time.

Remark 10. *It might be possible to prove **QMA**-completeness of 2-local $\{ZZ, XX\}$ -Hamiltonian problem with local simulatability by combining the techniques of [BG20, GSY19] and [CM16]. However, this is not clear, and indeed, this is mentioned as an open problem in [BG20]. Therefore we consider the 5-local Hamiltonian problem whenever we need local simulatability.*

Our construction of a CV-NIP for a **QMA** promise problem L is given in Figure 9 where $\mathcal{H}_{\mathbf{x}}$, N , p_{j_1, j_2} , s_{j_1, j_2} , α , β , and ρ_{hist} are as in Lemma D.2 for L . We remark that the proving algorithm uses only one witness, and thus we have $k = 1$ in Definition 2.8 for this protocol. Multiple copies of the witness are needed only when we do the gap amplification (Lemma 2.9). A similar remark applies to all protocols proposed in this paper.

We prove the following lemma.

Setup(1^λ): The setup algorithm chooses $(h, m_1, \dots, m_N) \xleftarrow{\$} \{0, 1\}^{N+1}$, and outputs a proving key $k_P := \bigotimes_{j=1}^N (H^h |m_j\rangle)$ and a verification key $k_V := (h, m_1, \dots, m_N)$.

Prove($k_P, \mathbf{x}, \mathbf{w}$): The proving algorithm generates the history state ρ_{hist} for $\mathcal{H}_{\mathbf{x}}$ from \mathbf{w} and measures j -th qubits of ρ_{hist} and k_P in the Bell basis for $j \in [N]$. Let $x := x_1 \| x_2 \| \dots \| x_N$, and $z := z_1 \| z_2 \| \dots \| z_N$ where $(x_j, z_j) \in \{0, 1\}^2$ denotes the outcome of j -th measurement. It outputs a proof $\pi := (x, z)$.

Verify(k_V, \mathbf{x}, π): The verification algorithm parses $(h, m_1, \dots, m_N) \leftarrow k_V$ and $(x, z) \leftarrow \pi$, chooses $(j_1, j_2) \in [N]^2$ according to the probability distribution defined by $\{p_{j_1, j_2}\}_{j_1 < j_2}$ (i.e., chooses (j_1, j_2) with probability p_{j_1, j_2}), defines $m'_{j_b} := m_{j_b} \oplus (hz_{j_b} \oplus (1-h)x_{j_b})$ for $b \in \{1, 2\}$, and outputs \top if $(-1)^{m'_{j_1} \oplus m'_{j_2}} = -s_{j_1, j_2}$ and \perp otherwise.

Figure 9: CV-NIP Π_{NIP} .

Setup_{vir-1}(1^λ): The setup algorithm generates N Bell-pairs between registers \mathbf{P} and \mathbf{V} and lets k_P and k_V be quantum states in registers \mathbf{P} and \mathbf{V} , respectively. Then it outputs (k_P, k_V) .

Prove_{vir-1}($k_P, \mathbf{x}, \mathbf{w}$): This is the same as **Prove**($k_P, \mathbf{x}, \mathbf{w}$) in Figure 9.

Verify_{vir-1}(k_V, \mathbf{x}, π): The verification algorithm chooses $h \xleftarrow{\$} \{0, 1\}$, and measures each qubit of k_V in basis $\{H^h |0\rangle, H^h |1\rangle\}$, and lets $(m_1, \dots, m_N) \in \{0, 1\}^N$ be the measurement outcomes. The rest of this algorithm is the same as **Verify**(k_V, \mathbf{x}, π) given in Figure 9.

Figure 10: The virtual protocol 1 for Π_{NIP}

Lemma D.3 (Completeness and Soundness). Π_{NIP} satisfies $(1 - \alpha)$ -completeness and $(1 - \beta)$ -soundness.

Since $(1 - \alpha) - (1 - \beta) = \beta - \alpha \geq 1/\text{poly}(\lambda)$, by combining Lemma 2.9 and Lemma D.3, Theorem D.1 follows.

In the following, we give a proof of Lemma D.3.

Proof of Lemma D.3. We prove this lemma by considering virtual protocols that do not change completeness and soundness. An alternative direct proof is given later. First, we consider the virtual protocol 1 described in Figure 10. The difference from the original protocol is that the setup algorithm generates N Bell pairs and gives each halves to the prover and verifier, and the verifier obtains (m_1, \dots, m_n) by measuring his halves in either standard or Hadamard basis.

Because verifier's measurement and the prover's measurement commute with each other, in the virtual protocol 1, verifier's acceptance probability does not change even if the verifier chooses h and measures k_V (i.e., the \mathbf{V} register of the N Bell-pairs) in the corresponding basis to obtain outcomes (m_1, \dots, m_N) before k_P (i.e, the \mathbf{P} register of the N Bell-pairs) is given to the prover. Moreover, conditioned on the above measurement outcomes, the state in \mathbf{P} collapses to $\bigotimes_{j=1}^N (H^h |m_j\rangle)$. (See Lemma 2.1.) Therefore, the virtual protocol 1 is exactly the same as the original protocol from the

Setup_{vir-2}(1^λ): This is the same as Setup_{vir-1}(1^λ) in Figure 10.

Prove_{vir-2}(k_P, **x, **w**):** This is the same as Prove(k_P, **x**, **w**) in Figure 9.

Verify_{vir-2}(k_V, **x, π):** The verification algorithm parses (x, z) ← π, computes k'_V := X^xZ^zk_VZ^zX^x, chooses h $\stackrel{\$}{\leftarrow}$ {0, 1}, measures each qubit of k'_V in basis {H^h|0⟩, H^h|1⟩}, and lets (m'₁, ..., m'_N) be the measurement outcomes. It chooses (j₁, j₂) ∈ [N]² according to the probability distribution defined by {p_{j₁, j₂}}_{j₁ < j₂} (i.e., chooses (j₁, j₂) with probability p_{j₁, j₂}) and outputs ⊤ if (−1)^{m'_{j₁} ⊕ m'_{j₂}} = −s_{j₁, j₂} and ⊥ otherwise.

Figure 11: The virtual protocol 2 for Π_{NIP}

prover's view, and verifier's acceptance probability of the virtual protocol 1 is the same as that of the original protocol Π_{NIP} for any possibly malicious prover.

Next, we further modify the protocol to define the virtual protocol 2 described in Figure 11. The difference from the virtual protocol 1 is that instead of setting m'_j := m_j ⊕ (hz_j + (1 − h)x_j), the verification algorithm applies a corresponding Pauli operator to (x, z) on k_V, and then measures it to obtain m'_j. Since X and Z before the measurement has the effect of flipping the measurement outcome for Z and X basis measurements, respectively, this does not change the distribution of (m'₁, ..., m'_N). (See Lemma 2.2.) Therefore, verifier's acceptance probability of the virtual protocol 2 is the same as that of the virtual protocol 1 for any possibly malicious prover.

Therefore, it suffices to prove (1 − α)-completeness and (1 − β)-soundness for the virtual protocol 2. When **x** ∈ L_{yes} and π is honestly generated, then k'_V is the history state ρ_{hist}, which satisfies Tr(ρ_{hist}ℋ_{**x**}) ≤ α, by the correctness of quantum teleportation (Lemma 2.4). Therefore, by Lemma 2.5 and Lemma D.2, verifier's acceptance probability is 1 − Tr(ρ_{hist}ℋ_{**x**}) ≥ 1 − α.

Let \mathcal{A} be an adaptive adversary against soundness of virtual protocol 2. That is, \mathcal{A} is given k_P and outputs (**x**, π). We say that \mathcal{A} wins if **x** ∈ L_{no} and Verify(k_V, **x**, π) = ⊤. For any **x**, let E_{**x**} be the event that the statement output by \mathcal{A} is **x**, and k'_{V, **x**} be the state in **V** right before the measurement by Verify conditioned on E_{**x**}. Similarly to the analysis for the completeness, by Lemma 2.5 and Lemma D.2, we have

$$\Pr[\mathcal{A} \text{ wins}] = \sum_{\mathbf{x} \in L_{\text{no}}} \Pr[E_{\mathbf{x}}] (1 - \text{Tr}(k'_{V, \mathbf{x}} \mathcal{H}_{\mathbf{x}})) \leq \sum_{\mathbf{x} \in L_{\text{no}}} \Pr[E_{\mathbf{x}}] (1 - \beta) \leq 1 - \beta.$$

□

Another proof of Lemma D.3. We first show the soundness. Let us define $H^h := \prod_{j=1}^N H_j^h$ and

$|m\rangle := \bigotimes_{j=1}^N |m_j\rangle$. Let $\{\Lambda_{x,z,\mathbf{x}}\}_{x,z,\mathbf{x}}$ be the POVM that the adversary \mathcal{A} does on k_P . Then,

$$\begin{aligned}
& \Pr \left[\mathbf{x} \in L_{\text{no}} \wedge \text{Verify}(k_V, \mathbf{x}, \pi) = \top : (k_P, k_V) \stackrel{\$}{\leftarrow} \text{Setup}(1^\lambda), (\mathbf{x}, \pi) \stackrel{\$}{\leftarrow} \mathcal{A}(k_P) \right] \\
&= \frac{1}{2} \sum_{h \in \{0,1\}} \frac{1}{2^N} \sum_{m \in \{0,1\}^N} \sum_{x,z} \sum_{\mathbf{x} \notin L} \langle m | H^h \Lambda_{x,z,\mathbf{x}} H^h | m \rangle \sum_{j_1, j_2} p_{j_1, j_2}^{\mathbf{x}} \frac{1 - s_{j_1, j_2}^{\mathbf{x}} (-1)^{m'_{j_1} \oplus m'_{j_2}}}{2} \\
&= \frac{1}{2} \sum_{h \in \{0,1\}} \frac{1}{2^N} \sum_{m \in \{0,1\}^N} \sum_{x,z} \sum_{\mathbf{x} \notin L} \sum_{j_1, j_2} p_{j_1, j_2}^{\mathbf{x}} \langle m | H^h \Lambda_{x,z,\mathbf{x}} H^h H^h X^x Z^z H^h \frac{I - s_{j_1, j_2}^{\mathbf{x}} Z_{j_1} Z_{j_2}}{2} H^h Z^z X^x H^h | m \rangle \\
&= \frac{1}{2} \sum_{h \in \{0,1\}} \frac{1}{2^N} \sum_{x,z} \sum_{\mathbf{x} \notin L} \sum_{j_1, j_2} p_{j_1, j_2}^{\mathbf{x}} \text{Tr} \left[H^h \Lambda_{x,z,\mathbf{x}} H^h H^h X^x Z^z H^h \frac{I - s_{j_1, j_2}^{\mathbf{x}} Z_{j_1} Z_{j_2}}{2} H^h Z^z X^x H^h \right] \\
&= \frac{1}{2^N} \sum_{x,z} \sum_{\mathbf{x} \notin L} \text{Tr} \left[Z^z X^x \Lambda_{x,z,\mathbf{x}} X^x Z^z (I - \mathcal{H}_{\mathbf{x}}) \right] \\
&= \text{Tr}[\sigma(I - \mathcal{H}_{\mathbf{x}})] \\
&\leq \text{Tr} \left[\frac{\sigma}{\text{Tr} \sigma} (I - \mathcal{H}_{\mathbf{x}}) \right] \\
&= 1 - \text{Tr} \left[\frac{\sigma}{\text{Tr} \sigma} \mathcal{H}_{\mathbf{x}} \right] \\
&\leq 1 - \beta,
\end{aligned}$$

where $\sigma := \frac{1}{2^N} \sum_{x,z} \sum_{\mathbf{x} \notin L} Z^z X^x \Lambda_{x,z,\mathbf{x}} X^x Z^z$. Note that $\frac{\sigma}{\text{Tr} \sigma}$ is a quantum state for any POVM $\{\Lambda_{x,z,\mathbf{x}}\}_{x,z,\mathbf{x}}$.

Next we show the completeness. The POVM corresponding to Prove is $\{\Lambda_{x,z} = \frac{1}{2^N} Z^z X^x \rho_{\text{hist}} X^x Z^z\}_{x,z}$. Note that this is a POVM, because $\Lambda_{x,z} \geq 0$, and

$$\sum_{x,z} \Lambda_{x,z} = 2^N \times \frac{1}{2^{2N}} \sum_{x,z} Z^z X^x \rho_{\text{hist}} X^x Z^z = 2^N \frac{I}{2^N} = I.$$

The reason why such $\{\Lambda_{x,z}\}_{x,z}$ is the POVM done by Prove algorithm is as follows. The Prove algorithm first prepares $\rho_{\text{hist}} \otimes H^h |m\rangle \langle m| H^h$, and then measures j th qubit of the history state and the j th qubit of $H^h |m\rangle$ in the Bell basis for all $j = 1, 2, \dots, N$. Then,

$$\begin{aligned}
& \left(\bigotimes_{j=1}^N \langle \phi_{x_j, z_j} | \right) \left(\rho_{\text{hist}} \otimes H^h |m\rangle \langle m| H^h \right) \left(\bigotimes_{j=1}^N | \phi_{x_j, z_j} \rangle \right) \\
&= \text{Tr} \left[\frac{1}{2^N} Z^z X^x \rho_{\text{hist}} X^x Z^z \times H^h |m\rangle \langle m| H^h \right].
\end{aligned}$$

Hence

$$\begin{aligned}
& \Pr \left[\text{Verify}(k_V, \mathbf{x}, \pi) = \top : (k_P, k_V) \stackrel{\$}{\leftarrow} \text{Setup}(1^\lambda), \pi \stackrel{\$}{\leftarrow} \text{Prove}(k_P, \mathbf{x}, \mathbf{w}^{\otimes k}) \right] \\
&= \frac{1}{2^N} \sum_{x,z} \text{Tr} \left[Z^z X^x \left(\frac{1}{2^N} Z^z X^x \rho_{\text{hist}} X^x Z^z \right) X^x Z^z (I - \mathcal{H}_{\mathbf{x}}) \right] \\
&= \text{Tr} \left[\rho_{\text{hist}} (I - \mathcal{H}_{\mathbf{x}}) \right] \\
&= 1 - \text{Tr} \left[\rho_{\text{hist}} \mathcal{H}_{\mathbf{x}} \right] \\
&\geq 1 - \alpha.
\end{aligned}$$

□

Impossibility of classical setup. In our protocol, the setup algorithm sends a quantum proving key to the prover. Can it be classical? It is easy to see that such a protocol can exist only for languages in **AM**.¹³ In fact, assume that we have a CV-NIP for L in the SP model where the proving key is classical. Then, we can construct a 2-round interactive proof for L where the verifier runs the setup by itself and sends the proving key to the prover, and then the prover replies as in the original protocol. Since $\mathbf{IP}(2) = \mathbf{AM}$, the above implies $L \in \mathbf{AM}$. Since it is believed that **BQP** is not contained in **AM** [RT19], it is highly unlikely that there is a CV-NIP even for **BQP** in the SP model with classical setup.

E Construction of Dual-Mode k -out-of- n Oblivious Transfer

In this section, we prove Lemma 4.7. That is, we give a construction of a dual-mode k -out-of- n oblivious transfer defined in Definition 4.6 based on the LWE assumption.

E.1 Building Block

We introduce dual-mode encryption that is used as a building block for our construction. We refer to [PVW08] for the intuition of this primitive.

Definition E.1 (Dual-Mode Encryption [PVW08, Qua20]¹⁴). *A dual-mode encryption scheme over the message space \mathcal{M} consists of PPT algorithms $\Pi_{\text{DEnc}} = (\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec}, \text{FindMessy}, \text{TrapKeyGen})$ with the following syntax.*

Setup($1^\lambda, \text{mode}$): *The setup algorithm takes the security parameter 1^λ and a mode $\text{mode} \in \{\text{messy}, \text{dec}\}$ as input, and outputs a common reference string crs and a trapdoor td_{mode} .*

KeyGen(crs, σ): *The key generation algorithm takes the common reference string crs and a branch value $\sigma \in \{0, 1\}$ as input, and outputs a public key pk and a secret key sk .*

Enc($\text{crs}, \text{pk}, b, \mu$): *The encryption algorithm takes the common reference string crs , a public key pk , a branch value $b \in \{0, 1\}$, and a message $\mu \in \mathcal{M}$ as input, and outputs a ciphertext ct .*

Dec($\text{crs}, \text{sk}, \text{ct}$): *The decryption algorithm takes the common reference string crs , a secret key sk , and a ciphertext ct as input, and outputs a message $\mu \in \mathcal{M}$.*

FindMessy($\text{crs}, \text{td}_{\text{messy}}, \text{pk}$): *The messy branch finding algorithm takes the common reference string crs , trapdoor td_{messy} in the messy mode, and a public key pk as input, and outputs a branch value $b \in \{0, 1\}$.*

TrapKeyGen($\text{crs}, \text{td}_{\text{dec}}$): *The trapdoor key generation algorithm takes the common reference string crs and a trapdoor td_{dec} in the decryption mode as input, and outputs a public key pk_0 and two secret keys sk_0 and sk_1 that correspond to branches 0 and 1, respectively.*

We require Π_{DEnc} to satisfy the following properties.

Correctness for Decryptable Branch For all $\text{mode} \in \{\text{messy}, \text{dec}\}$, $\sigma \in \{0, 1\}$, and $\mu \in \mathcal{M}$, we have

$$\Pr \left[\begin{array}{l} (\text{crs}, \text{td}_{\text{mode}}) \stackrel{\$}{\leftarrow} \text{Setup}(1^\lambda, \text{mode}) \\ (\text{pk}, \text{sk}_\sigma) \stackrel{\$}{\leftarrow} \text{KeyGen}(\text{crs}, \sigma) \\ \text{ct} \stackrel{\$}{\leftarrow} \text{Enc}(\text{crs}, \text{pk}, \sigma, \mu) \end{array} \right] \geq 1 - \text{negl}(\lambda).$$

¹³A similar observation is also made in [Ps05].

¹⁴This definition is based on the definition in [Qua20], which has several minor differences from that in [PVW08].

Statistical Security in the Messy Mode *With overwhelming probability over $(\text{crs}, \text{td}_{\text{messy}}) \xleftarrow{\$} \text{Setup}(1^\lambda, \text{messy})$, for all possibly malformed pk , all messages $\mu_0, \mu_1 \in \{0, 1\}^\ell$, and all unbounded-time distinguisher \mathcal{D} , we have*

$$\left| \Pr \left[\mathcal{D}(\text{ct}) = 1 : \begin{array}{l} b \xleftarrow{\$} \text{FindMessy}(\text{crs}, \text{td}_{\text{messy}}, \text{pk}) \\ \text{ct} \xleftarrow{\$} \text{Enc}(\text{crs}, \text{pk}, b, \mu_0) \end{array} \right] - \Pr \left[\mathcal{D}(\text{ct}) = 1 : \begin{array}{l} b \xleftarrow{\$} \text{FindMessy}(\text{crs}, \text{td}_{\text{messy}}, \text{pk}) \\ \text{ct} \xleftarrow{\$} \text{Enc}(\text{crs}, \text{pk}, b, \mu_1) \end{array} \right] \right| \leq \text{negl}(\lambda).$$

Statistical Security in the Decryption Mode *With overwhelming probability over $(\text{crs}, \text{td}_{\text{dec}}) \xleftarrow{\$} \text{Setup}(1^\lambda, \text{dec})$, for all $\sigma \in \{0, 1\}$ and all unbounded-time distinguisher \mathcal{D} , we have*

$$\left| \Pr \left[\mathcal{D}(\text{pk}, \text{sk}_\sigma) = 1 : (\text{pk}, \text{sk}_\sigma) \xleftarrow{\$} \text{KeyGen}(\text{crs}, \sigma) \right] - \Pr \left[\mathcal{D}(\text{pk}, \text{sk}_\sigma) = 1 : (\text{pk}, \text{sk}_0, \text{sk}_1) \xleftarrow{\$} \text{TrapKeyGen}(\text{crs}, \text{td}_{\text{dec}}) \right] \right| \leq \text{negl}(\lambda).$$

Computational Mode Indistinguishability *For any non-uniform QPT distinguisher \mathcal{D} , we have*

$$\left| \Pr \left[\mathcal{D}(\text{crs}) = 1 : (\text{crs}, \text{td}_{\text{messy}}) \xleftarrow{\$} \text{CRSGen}(1^\lambda, \text{messy}) \right] - \Pr \left[\mathcal{D}(\text{crs}) = 1 : (\text{crs}, \text{td}_{\text{dec}}) \xleftarrow{\$} \text{CRSGen}(1^\lambda, \text{dec}) \right] \right| \leq \text{negl}(\lambda).$$

Quach [Qua20] gave a construction of a dual-mode encryption scheme based on the LWE assumption.

Lemma E.2 ([Qua20]). *If the LWE assumption holds, then there exists a dual-mode encryption scheme.*

Remark 11. *Peikert, Vaikuntanathan, and Waters [PVW08] gave a construction of a relaxed variant of dual-mode encryption scheme based on the LWE assumption. Their construction is more efficient than that of Quach [Qua20] since they only rely on LWE with polynomial size modulus whereas Quach's construction relies on LWE with super-polynomial modulus. However, their scheme does not suffice for our purpose due to the following two reasons.*

1. *The security in the decryption mode holds only against computationally bounded adversaries.*
2. *crs can be reused only for bounded number of times.*

E.2 1-out-of- n Oblivious Transfer

In this section, we construct a dual-mode 1-out-of- n oblivious transfer based on dual-mode encryption. That is, we prove the following lemma.

Lemma E.3. *If there exists a dual-mode encryption scheme, then there exists a dual-mode 1-out-of- n oblivious transfer.*

CRSGen_{1-n}(1^λ, mode): Let mode' := dec if mode = binding and mode' := messy if mode = hiding. Then it generates (crs, td_{mode'}) $\stackrel{\$}{\leftarrow}$ Setup(1^λ, mode') and outputs crs.

Receiver_{1-n}(crs, j): It generates (pk_i, sk_{i,σ_i}) $\stackrel{\$}{\leftarrow}$ KeyGen(crs, σ_i) for all i ∈ [N] where σ_j := 1 and σ_i := 0 for all i ∈ [n] \ {j}. It outputs ot₁ := {pk_i}_{i∈[n]} and st := (j, {σ_i, sk_{i,σ_i}}_{i∈[n]}).

Sender_{1-n}(crs, ot₁, μ): It parses {pk_i}_{i∈[n]} ← ot₁ and (μ₁, ..., μ_n) ← μ, generates (r₁, ..., r_{N-1}) $\stackrel{\$}{\leftarrow}$ {0, 1}^{ℓ×(N-1)}, sets μ'_{i,0} := μ_i ⊕ r_{i-1} and μ'_{i,1} := r_i ⊕ r_{i-1} for all i ∈ [n] where r₀ is defined to be 0^ℓ. Then it generates ct_{i,b} $\stackrel{\$}{\leftarrow}$ Enc(pk_i, b, μ'_{i,b}) for all i ∈ [n] and b ∈ {0, 1}, and outputs ot₂ := {ct_{i,b}}_{i∈[n], b∈{0,1}}.

Derive_{1-n}(st, ot₂): It parses (j, {σ_i, sk_{i,σ_i}}_{i∈[n]}) ← st and {ct_{i,b}}_{i∈[n], b∈{0,1}} ← ot₂, computes μ'_{i,σ_i} $\stackrel{\$}{\leftarrow}$ Dec(sk_{i,σ_i}, ct_{i,σ_i}) for all i ∈ [j] and outputs μ_j := $\bigoplus_{i=1}^j \mu'_{i,\sigma_i}$.

Figure 12: Our 1-out-of-*n* oblivious transfer Π_{1-n}

Let Π_{DEnc} = (Setup, KeyGen, Enc, Dec, FindMessy, TrapKeyGen) be a dual-mode encryption scheme over the message space $\mathcal{M} = \{0, 1\}^\ell$. Then our construction of a dual-mode 1-out-of-*n* oblivious transfer OT_{1-n} = (CRSGen_{1-n}, Receiver_{1-n}, Sender_{1-n}, Derive_{1-n}) over the message space \mathcal{M} is given in Figure 12. This can be seen as a protocol obtained by applying the conversion of [BCR86] to the dual-mode 1-out-of-2 oblivious transfer of [Qua20].

Then we prove the following lemmas.

Lemma E.4. Π_{1-n} satisfies correctness

Proof. This easily follows from correctness of Π_{DEnc}. □

Lemma E.5. Π_{1-n} satisfies the computational mode indistinguishability.

Proof. This can be reduced to the computational mode indistinguishability of Π_{DEnc} in a straightforward manner. □

Lemma E.6. Π_{1-n} satisfies statistical receiver's security in the binding mode.

Proof. We construct Sim_{rec} as follows.

Sim_{rec}(crs): It generates (pk_i, sk_{i,0}) $\stackrel{\$}{\leftarrow}$ KeyGen(crs, 0) for all i ∈ [n], and outputs ot₁ := {pk_i}_{i∈[n]}.

By statistical security in the decryption mode of Π_{DEnc}, with overwhelming probability over (crs, td_{dec}) $\stackrel{\$}{\leftarrow}$ Setup(1^λ, dec), the distribution of pk generated as (pk, sk_σ) $\stackrel{\$}{\leftarrow}$ KeyGen(crs, σ) for any fixed σ ∈ {0, 1} is statistically close to that generated as (pk, sk₀, sk₁) $\stackrel{\$}{\leftarrow}$ TrapKeyGen(crs, td_{dec}), which does not depend on σ. Therefore, the distributions of pk_i generated by Sim_{rec}(crs) and Receiver(crs, j) are statistically close for any j ∈ [n]. Then statistical receiver's security in the binding mode of Π_{1-n} follows by a standard hybrid argument. □

Lemma E.7. Π_{1-n} satisfies the statistical sender's security in the hiding mode.

Proof. We construct Sim_{CRS}, Open_{rec}, and Sim_{sen} as follows.

$\text{Sim}_{\text{CRS}}(1^\lambda)$: It generates $(\text{crs}, \text{td}_{\text{messy}}) \xleftarrow{\$} \text{Setup}(1^\lambda, \text{messy})$ and outputs crs and $\text{td} := \text{td}_{\text{messy}}$.

$\text{Open}_{\text{rec}}(\text{td}, \text{ot}_1)$: It parses $\text{td}_{\text{messy}} \leftarrow \text{td}$ and $\{\text{pk}_i\}_{i \in [n]} \leftarrow \text{ot}_1$, computes $\sigma_i \xleftarrow{\$} \text{FindMessy}(\text{td}_{\text{messy}}, \text{pk}_i)$ for all $i \in [n]$, and outputs the minimal $j \in [n]$ such that $\sigma_j = 1$.

$\text{Sim}_{\text{sen}}(\text{crs}, \text{ot}_1, j, \mu_j)$: It generates $\mu_i \xleftarrow{\$} \mathcal{M}$ for $i \in [n] \setminus \{j\}$, and outputs $\text{ot}_2 \xleftarrow{\$} \text{Sender}_{1-n}(\text{crs}, \text{ot}_1, (\mu_1, \dots, \mu_n))$.

The first item of statistical sender's security in the hiding mode is clear because $\text{Sim}_{\text{CRS}}(1^\lambda)$ generates crs in exactly the same manner as $\text{CRSGen}(1^\lambda, \text{hiding})$. In the following, we prove the second item is also satisfied. For any unbounded-time adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ and fixed $\boldsymbol{\mu} = (\mu_1, \dots, \mu_n)$, we consider the following sequence of games between \mathcal{A} and the challenger. We denote by E_i the event that \mathcal{A}_1 returns 1 in Game_i .

Game₁: This game works as follows.

1. The challenger generates $(\text{crs}, \text{td}_{\text{messy}}) \xleftarrow{\$} \text{Setup}(1^\lambda, \text{messy})$ and sets $\text{td} := \text{td}_{\text{messy}}$.
2. \mathcal{A}_0 takes (crs, td) as input and outputs $\text{ot}_1 = \{\text{pk}_i\}_{i \in [n]}$ and $\text{st}_{\mathcal{A}}$.
3. The challenger computes $j := \text{Open}_{\text{rec}}(\text{td}, \text{ot}_1)$. That is, it computes $\sigma_i \xleftarrow{\$} \text{FindMessy}(\text{td}_{\text{messy}}, \text{pk}_i)$ for all $i \in [n]$ and let j be the minimal value such that $\sigma_j = 1$.
4. The challenger sets $\tilde{\mu}_j := \mu_j$, generates $\tilde{\mu}_i \xleftarrow{\$} \mathcal{M}$ for $i \in [n] \setminus \{j\}$ and $(r_1, \dots, r_{N-1}) \xleftarrow{\$} \{0, 1\}^{\ell \times (N-1)}$, and sets $\mu'_{i,0} := \tilde{\mu}_i \oplus r_{i-1}$ and $\mu'_{i,1} := r_i \oplus r_{i-1}$ for all $i \in [n]$ where r_0 is defined to be 0^ℓ . Then it generates $\text{ct}_{i,b} := \text{Enc}(\text{pk}_i, b, \mu'_{i,b})$ for all $i \in [n]$ and $b \in \{0, 1\}$ and sets $\text{ot}_2 := \{\text{ct}_{i,b}\}_{i \in [n], b \in \{0,1\}}$.
5. \mathcal{A}_1 takes $\text{st}_{\mathcal{A}}$ and ot_2 as input and outputs a bit β .

Game₂: This game is identical to the previous game except that μ'_{i,σ_i} is replaced with 0^ℓ for all $i \in [n]$.

By the statistical security in the messy mode of Π_{DEnc} , it is easy to see that we have $|\Pr[\text{E}_2] - \Pr[\text{E}_1]| \leq \text{negl}(\lambda)$.

Game₃: This game is identical to the previous game except that $\mu'_{i,0}$ is replaced with an independently and uniformly random element of \mathcal{M} for all $i > j$. We note that this game does not use $\{\tilde{\mu}_i\}_{i \neq j}$ at all.

By an easy information theoretical argument, we can see that the distribution of $\{\mu'_{i,b}\}_{i \in [n], b \in \{0,1\}}$ does not change from the previous game, and thus we have $\Pr[\text{E}_3] = \Pr[\text{E}_2]$.

Game₄: This game is identical to the **Game₁** except that the challenger uses $\boldsymbol{\mu}$ instead of $\tilde{\boldsymbol{\mu}}$.

By considering similar game hops to those from **Game₁** to **Game₃** in the reversed order, by the statistical security in the messy mode of Π_{DEnc} , we have $|\Pr[\text{E}_4] - \Pr[\text{E}_3]| \leq \text{negl}(\lambda)$.

Combining the above, we have $|\Pr[\text{E}_4] - \Pr[\text{E}_1]| \leq \text{negl}(\lambda)$. This is exactly the second item of statistical sender's security in the hiding mode. \square

By combining Lemmas [E.4](#) to [E.7](#), we obtain Lemma [E.3](#).

CRSGen_{k-n}(1^λ, mode): It generates $\text{crs} \stackrel{\$}{\leftarrow} \text{CRSGen}_{1-n}(1^\lambda, \text{mode})$ and outputs crs .

Receiver_{k-n}(crs, J): It parses $(j_1, \dots, j_k) \leftarrow J$, generates $(\text{ot}_{1,i}, \text{st}_i) \stackrel{\$}{\leftarrow} \text{Receiver}_{1-n}(\text{crs}, j_i)$ for all $i \in [k]$, and outputs $\text{ot}_1 := \{\text{ot}_{1,i}\}_{i \in [k]}$ and $\text{st} := \{\text{st}_i\}_{i \in [k]}$.

Sender_{k-n}(crs, ot₁, μ): It parses $\{\text{ot}_{1,i}\}_{i \in [k]} \leftarrow \text{ot}_1$, generates $\text{ot}_{2,i} \stackrel{\$}{\leftarrow} \text{Sender}_{1-n}(\text{crs}, \text{ot}_{1,i}, \mu)$ for all $i \in [k]$, and outputs $\text{ot}_2 := \{\text{ot}_{2,i}\}_{i \in [k]}$.

Derive_{k-n}(crs, st, ot₂): It parses $\{\text{st}_i\}_{i \in [k]} \leftarrow \text{st}$, computes $\mu_{j_i} \stackrel{\$}{\leftarrow} \text{Derive}_{1-n}(\text{crs}, \text{st}_i, \text{ot}_{2,i})$ for $i \in [k]$, and outputs $(\mu_{j_1}, \dots, \mu_{j_k})$.

Figure 13: Our k -out-of- n oblivious transfer Π_{k-n}

E.3 k -out-of- n Oblivious Transfer

In this section, we construct a dual-mode k -out-of- n oblivious transfer based on dual-mode 1-out-of- n oblivious transfer by k parallel repetitions. That is, we prove the following lemma.

Lemma E.8. *If there exists a dual-mode 1-out-of- n oblivious transfer, then there exists a dual-mode k -out-of- n oblivious transfer.*

By combining Lemmas E.2, E.3 and E.8, we obtain Lemma 4.7.

What is left is to prove Lemma E.8. Let $\Pi_{1-n} = (\text{CRSGen}_{1-n}, \text{Receiver}_{1-n}, \text{Sender}_{1-n}, \text{Derive}_{1-n})$ be a dual-mode 1-out-of- n oblivious transfer over the message space \mathcal{M} . Then our dual-mode k -out-of- n oblivious transfer $\Pi_{k-n} = (\text{CRSGen}_{k-n}, \text{Receiver}_{k-n}, \text{Sender}_{k-n}, \text{Derive}_{k-n})$ is described in Figure 13.

Then we prove the following lemmas.

Lemma E.9. Π_{k-n} satisfies correctness.

Proof. This can be reduced to correctness of Π_{1-n} in a straightforward manner. \square

Lemma E.10. Π_{k-n} satisfies the computational mode indistinguishability.

Proof. This can be reduced to the computational mode indistinguishability of Π_{1-n} in a straightforward manner. \square

Lemma E.11. Π_{k-n} satisfies statistical receiver's security in the binding mode.

Proof. Let $\text{Sim}_{\text{rec},1-n}$ be the corresponding algorithm for statistical receiver's security in the binding mode of Π_{1-n} . Then We construct $\text{Sim}_{\text{rec},k-n}$ for Π_{k-n} as follows.

Sim_{rec,k-n}(crs): It parses $(\text{crs}, \text{pk}) \leftarrow \text{crs}$, computes $\text{ot}_{1,i} \stackrel{\$}{\leftarrow} \text{Sim}_{\text{rec},1-n}(\text{crs})$ for all $i \in [k]$, and outputs $\text{ot}_1 := \{\text{ot}_{1,i}\}_{i \in [k]}$.

Statistical receiver's security in the binding mode of Π_{k-n} follows from that of Π_{1-n} by a straightforward hybrid argument. \square

Lemma E.12. Π_{k-n} satisfies the statistical sender's security in the hiding mode.

Proof. Let $\text{Sim}_{\text{CRS},1-n}$, $\text{Open}_{\text{rec},1-n}$, and $\text{Sim}_{\text{sen},1-n}$ be the corresponding algorithms for statistical sender's security in the hiding mode of Π_{1-n} . Then We construct $\text{Sim}_{\text{CRS},k-n}$, $\text{Open}_{\text{rec},k-n}$, and $\text{Sim}_{\text{sen},k-n}$ for Π_{k-n} as follows.

$\text{Sim}_{\text{CRS},k-n}(1^\lambda)$: This is exactly the same as $\text{Sim}_{\text{CRS},1-n}(1^\lambda)$.

$\text{Open}_{\text{rec}}(\text{td}, \text{ot}_1)$: It parses $\{\text{ot}_{1,i}\}_{i \in [k]} \leftarrow \text{ot}_1$, computes $j_i := \text{Open}(\text{td}, \text{ot}_{1,i})$ for all $i \in [k]$, and outputs $J = (j_1, \dots, j_k)$.

$\text{Sim}_{\text{sen}}(\text{crs}, \text{ot}_1, J, \boldsymbol{\mu}_J)$: It parses $(j_1, \dots, j_k) \leftarrow J$ and $(\mu_{j_1}, \dots, \mu_{j_k}) \leftarrow \boldsymbol{\mu}_J$, generates $\mu_i \xleftarrow{\$} \mathcal{M}$ for $i \in [n] \setminus \{j_1, \dots, j_k\}$, and outputs $\text{ot}_2 \xleftarrow{\$} \text{Sender}_{k-n}(\text{crs}, \text{ot}_1, (\mu_1, \dots, \mu_n))$.

The first item of statistical sender's security in the hiding mode of Π_{k-n} immediately follows from that of Π_{1-n} . In the following, we prove the second item. For any unbounded-time adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ and fixed $\mu = (\mu_1, \dots, \mu_n)$, we consider the following sequence of games between \mathcal{A} and the challenger. We denote by E_i the event that \mathcal{A}_1 returns 1 in Game_i .

Game_1 : This game works as follows.

1. The challenger generates $(\text{crs}, \text{td}) \xleftarrow{\$} \text{Sim}_{\text{CRS},k-n}(1^\lambda)$.
2. \mathcal{A}_0 takes (crs, td) as input and outputs $\text{ot}_1 = \{\text{ot}_{1,i}\}_{i \in [k]}$ and $\text{st}_{\mathcal{A}}$.
3. The challenger computes $J := \text{Open}_{\text{rec},k-n}(\text{td}, \text{ot}_1)$. That is, it computes $j_i := \text{Open}_{\text{rec},1-n}(\text{td}, \text{ot}_{1,i})$ for all $i \in [k]$ and lets $J := (j_1, \dots, j_k)$.
4. The challenger generates $\text{ot}_{2,i} \xleftarrow{\$} \text{Sim}_{\text{sen},1-n}(\text{crs}, \text{ot}_{1,i}, j_i, \mu_{j_i})$ for $i \in [k]$ and sets $\text{ot}_2 := \{\text{ot}_{2,i}\}_{i \in [k]}$.
5. \mathcal{A}_1 takes $\text{st}_{\mathcal{A}}$ and ot_2 as input and outputs a bit β .

Game_2 : This game is identical to the previous game except that $\text{ot}_{2,i}$ is generated as $\text{ot}_{2,i} \xleftarrow{\$} \text{Sender}(\text{crs}, \text{ot}_{1,i}, \boldsymbol{\mu})$ for $i \in [k]$.

By the second item of statistical sender's security in the hiding mode of Π_{1-n} , we have $|\Pr[\text{E}_2] - \Pr[\text{E}_1]| \leq \text{negl}(\lambda)$ by a standard hybrid argument. This is exactly the second item of statistical sender's security in the hiding mode.

□

By combining Lemmas E.9 to E.12, we obtain Lemma E.8.

F QRO + Shared Bell pair model

Definition F.1 (CV-NIZK in the QRO + Shared Bell pair Model). *A CV-NIZK for a QMA promise problem $L = (L_{\text{yes}}, L_{\text{no}})$ in the QRO + shared Bell pair model w.r.t. a random oracle distribution ROdist consists of algorithms $\Pi = (\text{Setup}, \text{Prove}, \text{Verify})$ with the following syntax:*

$\text{Setup}(1^\lambda)$: *This algorithm generates $\text{poly}(\lambda)$ Bell pairs (a state $\frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle)$) and sends the first and second halves to the prover and verifier as proving key k_P and verification key k_V , respectively.*

$\text{Prove}^H(k_P, \mathbf{x}, \mathbf{w}^{\otimes k})$: *This is a QPT algorithm that is given quantum oracle access to the random oracle H . It takes the proving key k_P , a statement \mathbf{x} , and $k = \text{poly}(\lambda)$ copies $\mathbf{w}^{\otimes k}$ of a witness $\mathbf{w} \in R_L(\mathbf{x})$ as input, and outputs a classical proof π .*

$\text{Verify}^H(k_V, \mathbf{x}, \pi)$: This is a QPT algorithm that is given quantum oracle access to the random oracle H . It takes the verification key k_V , a statement \mathbf{x} , and a proof π as input, and outputs \top indicating acceptance or \perp indicating rejection.

We require Π to satisfy the following properties.

Completeness. For all $\mathbf{x} \in L_{\text{yes}} \cap \{0, 1\}^\lambda$, and $\mathbf{w} \in R_L(\mathbf{x})$, we have

$$\Pr \left[\begin{array}{l} H \stackrel{\$}{\leftarrow} \text{ROdist} \\ \text{Verify}^H(k_V, \mathbf{x}, \pi) = \top : (k_P, k_V) \stackrel{\$}{\leftarrow} \text{Setup}(1^\lambda) \\ \pi \stackrel{\$}{\leftarrow} \text{Prove}^H(k_P, \mathbf{x}, \mathbf{w}^{\otimes k}) \end{array} \right] \geq 1 - \text{negl}(\lambda).$$

Adaptive Statistical Soundness. For all adversaries \mathcal{A} that make at most $\text{poly}(\lambda)$ quantum random oracle queries, we have

$$\Pr \left[\begin{array}{l} H \stackrel{\$}{\leftarrow} \text{ROdist} \\ \mathbf{x} \in L_{\text{no}} \wedge \text{Verify}^H(k_V, \mathbf{x}, \pi) = \top : (k_P, k_V) \stackrel{\$}{\leftarrow} \text{Setup}(1^\lambda) \\ (\mathbf{x}, \pi) \stackrel{\$}{\leftarrow} \mathcal{A}^H(k_P) \end{array} \right] \leq \text{negl}(\lambda).$$

Adaptive Multi-Theorem Zero-Knowledge. For defining the zero-knowledge property in the QROM, we define the syntax of a simulator in the QROM following [Unr15]. A simulator is given quantum access to the random oracle H and classical access to reprogramming oracle Reprogram . When the simulator queries (x, y) to Reprogram , the random oracle H is reprogrammed so that $H(x) := y$ while keeping the values on other inputs unchanged. Then the adaptive multi-theorem zero-knowledge property is defined as follows:

There exists a QPT simulator Sim with the above syntax such that for any QPT distinguisher \mathcal{D} , we have

$$\left| \Pr \left[\mathcal{D}^{H, \mathcal{O}_P^H(\cdot, \cdot)}(1^\lambda) = 1 : H \stackrel{\$}{\leftarrow} \text{ROdist} \right] - \Pr \left[\mathcal{D}^{H, \mathcal{O}_S^{H, \text{Reprogram}}(\cdot, \cdot)}(1^\lambda) = 1 : H \stackrel{\$}{\leftarrow} \text{ROdist} \right] \right| \leq \text{negl}(\lambda)$$

where \mathcal{D} 's queries to the second oracle should be of the form $(\mathbf{x}, \mathbf{w}^{\otimes k})$ where $\mathbf{w} \in R_L(\mathbf{x})$ and $\mathbf{w}^{\otimes k}$ is unentangled with \mathcal{D} 's internal registers, $\mathcal{O}_P^H(\mathbf{x}, \mathbf{w}^{\otimes k})$ generates $(k_P, k_V) \stackrel{\$}{\leftarrow} \text{Setup}(1^\lambda)$ and returns $(k_V, \text{Prove}^H(k_P, \mathbf{x}, \mathbf{w}^{\otimes k}))$, and $\mathcal{O}_S^{H, \text{Reprogram}}(\mathbf{x}, \mathbf{w}^{\otimes k})$ generates $(k_P, k_V) \stackrel{\$}{\leftarrow} \text{Setup}(1^\lambda)$ and returns $\text{Sim}^{H, \text{Reprogram}}(k_P, \mathbf{x})$.

Remark 12. The difference from the zero-knowledge property in the QRO + $(V \rightarrow P)$ model is that the malicious verifier is not allowed to maliciously generate k_P . This is because the setup is supposed to be run by a trusted third party in this model.

Contents

1	Introduction	1
1.1	Background	1
1.2	Our Results	2
1.3	Technical Overview	5
1.4	Related Work	8
2	Preliminaries	9
2.1	Quantum Computation Preliminaries	9
2.2	QMA and Local Hamiltonian Problem	11
2.3	Classically-Verifiable Non-Interactive Zero-knowledge Proofs	12
3	CV-NIZK in the QSP model	14
4	Dual-Mode CV-NIZK with Preprocessing	18
4.1	Definition	18
4.2	Building Blocks	21
4.3	Construction	24
5	CV-NIZK via Fiat-Shamir Transformation	30
5.1	Definition	30
5.2	Building Blocks	31
5.3	Construction	38
A	More Explanation on Lemma 2.7	44
B	More details for the proof of Lemma 3.2	45
C	Alternative Simpler Construction of CV-NIZK in the QSP Model.	46
D	CV-NIP in the QSP model	47
E	Construction of Dual-Mode k-out-of-n Oblivious Transfer	51
E.1	Building Block	51
E.2	1-out-of- n Oblivious Transfer	52
E.3	k -out-of- n Oblivious Transfer	55
F	QRO + Shared Bell pair model	56