

# Oblivious Transfer from Zero-Knowledge Proofs

## Or How to Achieve Round-Optimal Quantum Oblivious Transfer and Zero-Knowledge Proofs on Quantum States\*

Léo Colisson <sup>1,3</sup>, Garazi Muguruza<sup>2,3</sup>, Florian Speelman <sup>2,3</sup>  
leo.colisson@cwil.nl, g.muguruzalasa@uva.nl, f.speelman@uva.nl

<sup>1</sup> Centrum Wiskunde & Informatica, Netherlands

<sup>2</sup> Informatics Institute, University of Amsterdam, Netherlands

<sup>3</sup> QuSoft, Netherlands

**Abstract.** We provide a generic construction to turn any classical Zero-Knowledge (ZK) protocol into a composable (quantum) oblivious transfer (OT) protocol, mostly lifting the round-complexity properties and security guarantees (plain-model/statistical security/unstructured functions. . .) of the ZK protocol to the resulting OT protocol. Such a construction is unlikely to exist classically as Cryptomania is believed to be different from Minicrypt.

In particular, by instantiating our construction using Non-Interactive ZK (NIZK), we provide the first round-optimal (2-message) quantum OT protocol secure in the random oracle model, and round-optimal extensions to string and  $k$ -out-of- $n$  OT.

At the heart of our construction lies a new method that allows us to prove properties on a received quantum state without revealing additional information on it, even in a non-interactive way, without public-key primitives, and/or with statistical guarantees when using an appropriate classical ZK protocol. We can notably prove that a state has been partially measured (with arbitrary constraints on the set of measured qubits), without revealing any additional information on this set. This notion can be seen as an analog of ZK to quantum states, and we expect it to be of independent interest as it extends complexity theory to *quantum* languages, as illustrated by the two new complexity classes we introduce, [ZKstatesQIP](#) and [ZKstatesQMA](#).

**Keywords:** Quantum Cryptography, Oblivious Transfer, Zero-Knowledge on Quantum States, Multi-Party Computing, Zero-Knowledge

---

\*© IACR 2023. This article is the full version of the work published by Springer-Verlag (ASIACRYPT 2023).

# Table of Contents

1	Introduction	3
1.1	Contributions	4
1.2	Overview of the main contributions.	5
1.3	Concurrent work	10
1.4	Open problems and ongoing works	10
2	Preliminaries	11
2.1	Notations	11
2.2	Model of security	12
2.3	Cryptographic requirements.	16
2.4	Proofs	17
3	Protocol for bit OT	18
3.1	The protocol	18
3.2	Security proof	18
4	(NI)ZKoQS and $k$ -out-of- $n$ string OT	20
4.1	ZKoQS	20
4.2	Frequently Asked Questions	25
4.3	Proof of partial measurement: a generic framework to get ZKoQS	26
4.4	Protocol to prove that a state has been semi-collapsed	28
4.5	ZKstatesQIP $_S[k]$ and ZKstatesQMA $_S$ : ZKoQS from a complexity theory point of view	31
4.6	Applications to build string and $k$ -out-of- $n$ OT protocols	32
5	Composability of [Unr15]	33
6	Acknowledgment	34
	References	35
A	Proofs of statements in preliminaries	39
B	Proofs of security of the bit OT protocol	39
C	Proof of the ZKoQS and $k$ -out-of- $n$ string OT protocols	52
D	Proof of the composability of [Unr15]	60

## 1 Introduction

Oblivious Transfer (OT) is an extremely powerful primitive, as it was shown [Kil88] to be sufficient to perform multi-party computing (MPC), allowing multiple parties to jointly compute any function while keeping the input of each party secret. Since the introduction of 2-party computing in the seminal article of Yao [Yao82], followed by the famous generalisation to arbitrary many parties of Goldreich, Micali and Wigderson [GMW87], OT and MPC received a tremendous amount of attention [Wie83, PVW08, Rab05, EGL85, CGS02, DGJ<sup>+</sup>20, KP17, LT22, YAV<sup>+</sup>22].

However, all classical OT protocols need to use some structured computational assumptions providing trapdoors. Said differently, OT (classically) lives in *Cryptomania* [Imp95], a world where public-key cryptography exists. On the other hand, it was recently shown [GLS<sup>+</sup>21, BCK<sup>+</sup>21] that quantumly, OT lives in *MiniQCrypt*, meaning that it is possible to obtain OT protocols using a much weaker assumption, based only on (unstructured) one-way functions.

There are many reasons to avoid using trapdoor functions. For instance, this additional structure can often be exploited by quantum computers, leading to attacks. As a result, many OT protocols (based on RSA, quadratic residue, elliptic curves...) are vulnerable against quantum adversaries. While some proposals [PVW08, BD18, Qua20] based on post-quantum assumptions like the Learning-With-Errors problem (LWE) still seem to resist against quantum adversaries, minimizing assumptions is an important safety-guard against potential future attacks on the computational assumptions. Understanding the minimal required assumptions is also an active field of research, with the recent introduction of the notion of pseudo-random quantum states [JLS18], which is an even weaker assumption than one-way functions.

However, while we know (even classical) 2-message OT protocols—optimal in term of round complexity—achievable using trapdoors [PVW08, BD18], there is no known round-optimal protocol requiring no structure (such protocol would necessary be quantum unless Cryptomania collapses to MiniCrypt). The original proposal [CK88] for quantum OT (studied and improved in a long line of research [BBC<sup>+</sup>92, MS94, Yao95, DFL<sup>+</sup>09, Unr10, BF10, GLS<sup>+</sup>21, BCK<sup>+</sup>21], see also this review [SMP22] for quantum OT protocols based on physical assumptions, that we will not cover here) requires 7 messages, and [ABK<sup>+</sup>22] managed to obtain a 3-message protocol (computationally secure, in the random oracle model). However, they left the following question open:

*Does there exist two-message quantum chosen-input bit OT, that allows both parties to choose inputs?*

They also raise the question of the existence of a 2-message string OT, even when the bit chosen by the receiver is random. The main bottle-neck to further reduce the communication complexity of these protocols is the use of a “cut-and-choose” approach, where the receiver sends a quantum state and some commitments on the description of this state, gets a challenge from the sender to ensure that the quantum states were honestly prepared, and opens some commitments. Classically, we can avoid cut-and-choose by using Non-Interactive Zero-Knowledge proofs (NIZK) in order to prove an NP statement on a classical string without revealing anything on that string except the fact that the statement is true. However, defining NIZK proofs on quantum states is challenging as any measurement on a quantum state will irremediably alter it. While NIZK proofs on Quantum States (NIZKoQS) have been recently introduced [CGK21] and can be used to prove really advanced properties, they rely on trapdoor functions (LWE), and

therefore live in Cryptomania, and are moreover fundamentally only computationally secure. [CGK21] actually raised two open questions:

*Is it possible to do NIZKoQS without relying on LWE? Or with statistical security?*

Article	Classical	Setup	Messages	MiniQCrypt	Composable	Statistical
[PVW08]	Yes	CRS	2	No (LWE)	Yes	Either
[BD18]	Yes	Plain M.	2	No (LWE)	Sender	Receiver
[CK88] + later works	No	Depends	7	Yes	Yes [DFL+09, Umr10]	Either
[GLS+21]	No	Plain M./ CRS	poly/ cte $\geq 7$	Yes	Yes	No
[BCK+21]	No	Plain M./ CRS	poly/ cte $\geq 7$	Yes	Yes	Sender
[ABK+22]	No	RO	3	Yes	Yes	No
[BKS23]	No	RO + Shared EPR	2	Yes	Yes	Yes
This work + [Umr15]	No	RO	2	Yes	Yes	No
This work + [HSS11]	No	Plain M.	$> 2$	No (LWE)	Yes	No
This work + S-NIZK	No	Like ZK	2	Like ZK	Yes	Sender
This work + NIZK proof	No	Like ZK	2	Like ZK	Yes	Receiver
This work + ZK	No	Like ZK	ZK +1 or $2^4$	Like ZK	Yes	Like ZK

Fig. 1: Comparison with related works. “RO” stands for Random Oracle, “Plain M.” stands for “plain model”, “Like ZK” means that the properties (mostly) inherit from the property of the underlying ZK protocol, the party in the “statistical” column represents the malicious party allowed to be unbounded to get statistical security. Note that using [WW06] we can get statistical security against the other party (of course we lose the statistical security against the first party [Lo97]), at the cost of an additional message. This list only considers standard bit or string OT (notably [BKS23] also provides a 1-message protocol in the (strong) shared-EPR model, but for a randomized-version of OT).

## 1.1 Contributions

In this work, we answer positively all these open questions. We first state our results on OT protocols (see also Fig. 1 for a table comparing existing works):

**Theorem 1.1 (informal).** *There exists a (non-black-box<sup>5</sup>) 2-message string OT (even  $k$ -out-of- $n$  string OT) quantum protocol composable secure in the random oracle model, assuming the existence of a collision-resistant hiding<sup>6</sup> function.*

<sup>4</sup>+1 in the Common Random String model, +2 in the plain model.

<sup>5</sup>Our protocol requires the use of a hash function  $h$ : since we need to prove statements on preimages of  $h$  in a ZK protocol, this makes our protocol non-black-box with respect to  $h$  since the circuit of  $h$  must be known to the verifier. Therefore, even if the assumptions on  $h$  (collision-resistant and hiding) are trivially true if  $h$  is modelled as a random oracle, we cannot directly run the ZK protocol on an oracle since the source code of  $h$  cannot efficiently be sent to the verifier. For this reason, we do not model  $h$  itself as an oracle (this assumption is required by the ZK protocol), and only assume that  $h$  is collision-resistant and hiding.

<sup>6</sup>Informally, a hiding function  $h$  is a function such that it is not possible to get any information on  $x$  given  $h(x||r)$  for sufficiently large random  $r$  (this is used for instance in commitments). Actually, we use in practice a weaker assumption called “second-bit hardcore” (the function must only hide the second bit of  $x$ ), since we believe that we could use the hardcore-bit construction of Goldreich-Levin to weaken the assumptions further by only assuming that the function is one-way.

Actually, we provide a much more generic construction that allows us to obtain a variety of quantum OT protocols, depending on whether we want to optimize the round-complexity, the security (against unbounded sender, or unbounded verifier), the setup model (plain-model, Common Reference String (CRS), Random Oracle), or the computational assumptions (one-way functions, LWE, etc.).

**Theorem 1.2 (informal).** *Assuming the existence of a collision-resistant hiding one-way function, given any  $n$ -message ZK proof (or argument) of knowledge, we can obtain a  $n + 1$ -message  $OT^7$  protocol (or  $n + 2$  in the plain model<sup>8</sup>).*

*Moreover, if the ZK protocol is secure against any unbounded verifier (resp. prover) and if the function is statistically hiding (resp. injective), the resulting OT protocol is secure against any unbounded sender (resp. receiver).*

Note that classical ZK is a widely studied primitive as it turns out to be extremely useful in many applications, including in MPC, authentication, blockchain protocols [ELE], and more. Trapdoors are not necessary to build ZK as they can be built using only hash functions, and therefore live in Minicrypt. Many candidates have been proposed to achieve various ZK flavors: statistical security against malicious prover or malicious verifier, non-interactive or constant rounds protocols, security in the plain model, CRS, or random oracle [GMR85, Lin13, Unr15, PVW08, BD18, HSS11, PS19]. . . In this paper, we notably consider the non-interactive ZK protocol of Unruh [Unr15], proven secure in the random oracle model, together with the ZK protocol of Hallgren, Smith and Song [HSS11], proven secure in the plain-model assuming the hardness of LWE, but much work has been done to study ZK under many other assumptions [Wat09, AL20, Unr12, BS20, LMS21].

At the heart of our approach lies the first creation of a (potentially statistically secure when instantiated correctly) ZK protocol on *quantum* states, that can be seen as an extension of ZK and complexity theory to *quantum* languages:

**Theorem 1.3 (informal).** *Under the same assumptions as Theorem 1.2, a receiver can obtain a quantum state while being sure that a subset  $T$  of the qubits has been measured, without getting any information on  $T$  beside the fact that it fulfils some arbitrary fixed constraints.*

*The resulting protocol is  $n$ -message ( $n + 1$  in the plain model), and can in particular be non-interactive when using a NIZK protocol. Statistical security can also be obtained under the conditions described in Theorem 1.2 (the receiver playing the role of the prover, and the sender the verifier).*

We also extend the concept of ZK on Quantum State (ZKoQS), together with the notion of *quantum* languages and we define the first two “quantum-language” based complexity classes **ZKstatesQIP** and **ZKstatesQMA**. Finally, we prove relations between ZKoQS and various ideal functionalities, we prove that we can realize them, and we show examples of quantum languages belonging to **ZKstatesQIP** and **ZKstatesQMA**.

## 1.2 Overview of the main contributions.

In this section, we provide a quick, informal, overview of our approach. The OT functionality can be described as follows: a sender, Bob, owns two bits<sup>9</sup>  $m_0$  and  $m_1$ , and Alice wants to learn

<sup>7</sup>This holds for all variations of OT: bit OT, string OT, and  $k$ -out-of- $n$  OT.

<sup>8</sup>The model of security is the same as the ZK protocol if we want a  $n + 2$ -message protocol, and if we add the Common (uniform) Reference String assumption (weaker than the Random Oracle model) to provide the hash function, we can obtain a protocol with  $n + 1$  messages.

<sup>9</sup>Our approach also works for strings or  $k$ -out-of- $n$  OT.

$m_b$  where the bit  $b$  is provided as an input. Importantly, a malicious Bob should be unable to learn the value  $b$  of Alice, and a malicious Alice should be unable to get information on both  $m_0$  and  $m_1$ .

**First attempt: a naive OT protocol.** A first remark we can make is that if we are given a state in the computational basis  $|l\rangle$  for some bit  $l$ , rotating it by applying a  $Z^m$  gate for some bit  $m$  will leave the state unchanged (up to a global phases). On the other hand, if we are given a state in the Hadamard basis  $H|r\rangle$  for some bit  $r$ , applying a  $Z^m$  gate will flip the encoded bit if  $m = 1$ , giving the state  $H|r \oplus m\rangle$ . Therefore, we can imagine a naive protocol for OT: Alice could prepare two states  $|\psi^{(b)}\rangle := H|r^{(b)}\rangle$  and  $|\psi^{(1-b)}\rangle := |l\rangle$  for some random bits  $r^{(b)}$  and  $l$ , send  $|\psi^{(0)}\rangle$  and  $|\psi^{(1)}\rangle$  to Bob, Bob could rotate the  $i$ -th qubit according to  $Z^{m_i}$ , and measure them in the Hadamard basis, getting outcomes  $z^{(i)}$  that will be sent back to Alice. In the light of the above comment, it is easy to see that  $z^{(b)} = m_b \oplus r^{(b)}$  while  $z^{(1-b)}$  is a random bit, uncorrelated with  $m_{1-b}$ . Therefore, Alice can easily recover  $m_b = z^{(b)} \oplus r^{(b)}$  while she is unable to recover  $m_{1-b}$ . Moreover, because the density matrix of  $\frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|) = \frac{1}{2}(|+\rangle\langle +| + |-\rangle\langle -|)$  is the completely mixed state, Bob cannot recover any information on  $b$ ...

Unfortunately, this protocol is not secure: Alice can easily cheat by sending two  $|+\rangle$  states to learn both  $m_0$  and  $m_1$ .

**The need for ZK on quantum state.** To avoid this trivial cheating strategy, we would like, informally, to prove to Bob that at least one of the received states is in the computational basis... without revealing the position of this qubit, and without destroying that state. So in a sense, we would like a quantum equivalent of ZK, except that the statement is on a quantum state instead of on a classical bit string.

As a first sight, this might seem to contradict laws of physics: it is impossible to learn the basis of a random state, and anyway any measurement would certainly disturb the state. However, we can change a bit the procedure to send  $|\psi^{(0)}\rangle$  and  $|\psi^{(1)}\rangle$ , by sending instead bigger, more structured states encoding the original qubit: Bob would then do some (non-destructive) tests on this large state in order to check that the encoding is valid, and that at least one state is not in superposition, before collapsing it to a 2-qubit system.

At a high level, it is handy to define the encoded state as a superposition of pre-images of multiple (publicly known) images of a given hash function  $h$ : To control the number of elements allowed in the superposition, the key idea is to prove (using this time classical ZK), that the sender knows pre-images to all the publicly known images, where some of them are tagged as *dummy*, i.e. forbidden (e.g. by making sure they start with a 0). This way, if we prove that one of the two states admits only a single non-dummy preimage (without revealing which state), this state cannot be in superposition of multiple elements, or it would be possible to extract a collision of the hash function. Of course, this assumes that the receiver performs some checks to ensure that the quantum state is a valid encoding and only contains non-dummy preimages of  $h$ : this can be done for instance by checking in superposition that all elements are non-dummy (e.g. by measuring the first bit and checking that it's one), and by computing  $h$  and checking (in superposition) that it belongs to the set of allowed images. This way, ZK is used on a classical string to verify, indirectly, properties on the quantum state.

More formally, instead of sending  $|l\rangle$ , we sample a random bit string  $w_l^{(1-b)}$  starting with a 0 (this will be important later, but informally this indicates that this is a valid, non-dummy element) and send  $|\psi^{(1-b)}\rangle := |l\rangle |w_l^{(1-b)}\rangle$ , together with the hash  $h_l^{(1-b)} := h(l||w_l^{(1-b)})$ . Similarly, we can apply this idea on states in superposition: instead of sending  $|0\rangle + (-1)^{r^{(b)}} |1\rangle$ , we sample

similarly  $w_0^{(b)}$  and  $w_1^{(b)}$ , and send  $|\psi^{(b)}\rangle := |0\rangle|w_0^{(b)}\rangle + (-1)^{r^{(b)}}|1\rangle|w_1^{(b)}\rangle$ , together with the hashes  $h_0^{(b)} := h(0||w_0^{(b)})$  and  $h_1^{(b)} := h(1||w_1^{(b)})$ . Of course, now, it is relatively easy to distinguish both qubits, as the qubit in the computational basis comes with a single classical hash, while the other comes with two hashes. To avoid this issue, we add a “dummy” hash by sampling a random  $w_{1-l}^{(1-b)}$  starting with a 1 (indicating that the hash is dummy), and defining  $h_{1-l}^{(1-b)} := h(l||w_{1-l}^{(1-b)})$ . Importantly, given a hash, it is impossible to see if it is a dummy hash, as the hash function is hiding its input. However, Alice can prove to Bob, using classical ZK, that at least one of the provided hashes is a dummy hash, without revealing its position. Therefore, to sum-up, Alice sends the hashes, proves that she knows a preimage for all of them and that one of them is a dummy hash (i.e. its preimage has a 1 in its second position), before sending the states  $|\psi^{(0)}\rangle$  and  $|\psi^{(1)}\rangle$  to Bob (if the ZK proof is non-interactive, she can send everything in a single message).

Then, after verifying the ZK proof, Bob will verify that  $|\psi^{(0)}\rangle$  and  $|\psi^{(1)}\rangle$  are in a superposition of valid, non-dummy, preimages. More precisely, for  $i \in \{0, 1\}$ , he applies the unitary  $|x\rangle|w\rangle|0\rangle \rightarrow |x\rangle|w\rangle|w[1] = 0 \wedge h(x||w) \in \{h_0^{(i)}, h_1^{(i)}\}\rangle$  on the  $i$ -th qubit (after adding an auxiliary qubit), and measures the last register to check if it is equal to 1. Note that for honestly prepared state, this measurement will not alter the state, as the last registers always contains a  $|1\rangle$  and can therefore be factored out as the state is separable. Once the check is performed, we can shrink both states to obtain a 2-qubit state by measuring the second register containing the  $w$ 's in the Hadamard basis, getting two outcomes  $s^{(i)}$ 's. One can easily check that since  $|\psi^{(1-b)}\rangle$  is already in the computational basis, it will not alter the first qubit, resulting in the  $|l\rangle$  state, i.e. a qubit in the computational basis. On the other hand, it is not hard to see that the qubit  $|\psi^{(b)}\rangle$  will be turned into  $|0\rangle|w_0^{(b)}\rangle + (-1)^{r^{(b) \oplus \langle s, w_0^{(b)} \oplus w_1^{(b)} \rangle}}|1\rangle$ , i.e. the final state will be in the Hadamard basis (the encoded bit might be flipped, but Alice can easily recover that bit flip knowing the outcomes of the measurements).

This way, we are back to the original requirement of the naive oblivious transfer described above: Bob can rotate each qubit  $i$  using  $Z^{m_i}$ , measure them in the Hadamard basis, and send the outcomes  $z^{(i)}$  to Alice, together with the measurements  $s^{(0)}$  and  $s^{(1)}$ . Alice will then be able to recover the final bit  $m_b$  by computing  $r^{(b)} \oplus \langle s, w_0^{(b)} \oplus w_1^{(b)} \rangle \oplus z^{(b)}$ .

This protocol is summarized in Protocol 1, and can easily be generalized to string OT or  $k$ -out-of- $n$  OT by sending one “hashed qubit” per bit to transmit, and proving via ZK the wanted properties on the number and position of the dummy hashes (e.g. either the first half of hashes are dummy, or the second half). This will be described in more details below.

**Sketch of security proof.** Interestingly, this method is significantly simpler to analyse than the interactive cut-and-choose approach used in previous works, as illustrated by the long line of research trying to prove the security of the original proposal [BBC<sup>+</sup>92, MS94, Yao95, DFL<sup>+</sup>09, Unr10, BF10]. Of course, part of this analysis is offloaded to the ZK protocol, but we like to see it as a feature: this allows us to have a more modular protocol (any improvement on ZK directly implies an improvement on OT), and the analysis only needs to be done once for the *classical* ZK protocol.

At a very high level, since the ZK protocol leaks no information on the witness, and because the hash is hiding<sup>10</sup>, Bob learns no information on  $b$ . Note that the quantum state does not help as one can see that for any bit string  $x_0, x_1$  the density matrix of  $|x\rangle$  where  $x \stackrel{\$}{\leftarrow} \{x_0, x_1\}$  is

<sup>10</sup>In practice, we ask for  $h$  to be “second-bit hardcore”, meaning that it is not possible to learn the second bit of  $x$  given  $h(x)$ , but we could also certainly extend the construction to work for any one-way function using the Goldreich-Levin construction and rejection sampling.

equal to the density matrix of  $|x_0\rangle \pm |x_1\rangle$ , where the sign is randomly chosen. To translate this informal argument into a composable security proof, we design our simulator by first replacing the ZK proof with a simulated proof (that does not need access to the witness), then we turn the dummy hash into a non-dummy hash (indistinguishable since  $h$  is hiding), and we sample  $|\psi^{(1-b)}\rangle$  like  $|\psi^{(b)}\rangle$  (indistinguishable by the above argument on density matrices). This way, the simulator can extract both  $m_0$  and  $m_1$ , and provide them to the ideal functionality for OT, that will be in charge of discarding  $m_{1-b}$  and outputting  $m_b$ . See Theorem 3.1 for more details.

On the other hand, to learn information about both  $m_0$  and  $m_1$ , Alice needs to produce two non-collapsed states. But the tests performed by Bob force Alice to send a superposition of non-dummy preimages (in case she does not, the test might pass with some probability, but the state will be anyway projected on a superposition of non-dummy valid preimages in that case). However, by the ZK property, at least one of the classical hashes must be a dummy hash, and therefore if the corresponding qubit contains a superposition of multiple valid preimages, one of them must either collide with the dummy hash, or with the non-dummy one. This collision can even be obtained with non-negligible probability by measuring the state in the computational basis and comparing the outcome with the preimages extracted by the simulator during the ZK protocol. More details can be found in the proof of Theorem 3.1.

Note that if all the properties hold against an unbounded Alice (resp. Bob), notably by instantiating the protocol with a ZK *proof* of knowledge and an injective function  $h$  (resp. a statistical ZK and a statistically hiding function) our OT protocol is secure against an unbounded receiver (resp. sender). Note also that since our adversaries are non-uniform, we need to find a way to distribute the function  $h$  in such a way that the non-uniform advice cannot depend on  $h$  (or it might hardcode a collision). By relying on the CRS assumption (actually a uniformly random string is enough), the hash function can be distributed non-interactively by the CRS (or heuristically replaced with a fixed hash function). If we want to stay in the plain model we can instead ask Bob to sample the function and send it to Alice at the beginning of the protocol, adding an additional message (providing a  $(n+2)$ -message OT protocol instead of  $n+1$ , where  $n$  is the number of messages of the ZK protocol).

**ZKoQS and quantum language.** The above protocol internally proves a statement on a quantum state, suggesting a quantum analogue to classical Zero-Knowledge and languages. While this notion was introduced in [CGK21] ([CGK21] actually relies on the Learning-With-Error (LWE) problem while we do not require such structure, and they are fundamentally only computationally secure), we extend their definition of ZK, notably introducing the notion of subclass needed when the protocol is composed into other protocols, and we provide a second, MPC-based point of view.

At a high level, a quantum language is, similarly to classical language  $\mathcal{L} \subseteq \{0,1\}^*$ , described by a set of quantum states  $\mathcal{L}_Q$ . Analogously to classical proof systems, where a proof should be accepted only if  $x \in L$ , quantumly we expect the proof to be accepted only if  $\rho \in \mathcal{L}_Q$ , where  $\rho$  is the obtained quantum state. Classically, we also divide  $\mathcal{L}$  into subsets  $\mathcal{L}_w$  where  $w$ 's are called witnesses: during an honest run of the protocol we expect  $x \in \mathcal{L}_w$ . Similarly, quantumly we divide  $\mathcal{L}_Q$  into subsets  $\mathcal{L}_{\omega,\omega_s}$ , where  $(\omega,\omega_s)$  are classical elements<sup>11</sup> (say bit strings, we will explain later why we need two elements): like classically<sup>12</sup>, we expect to have  $\rho \in \mathcal{L}_{\omega,\omega_s} \subseteq \mathcal{L}_\omega$  during an honest run of the protocol.  $\omega$  and  $\omega_s$  can therefore be seen as a partial classical

<sup>11</sup>For instance, you can think of  $\omega$  as the basis of  $\rho$ , and  $\omega_s$  as the bits encoded in these basis.

<sup>12</sup>Note that in the formal definitions, we actually formalize them using the more general notion of simulators for various reasons, to be compatible with simulation-based proofs, but also since quantumly it is not possible



description of  $\rho$ . Finally, classically, the ZK property states that a malicious receiver should not learn  $w$ : quantumly we expect a malicious receiver to be unable to learn  $\omega$ .

*Remark 1.4.* Despite the similarities of ZKoQS with the corresponding classical notions, there are still a few differences with the classical setting:

- First, as pictured in Fig. 3, classical ZK is typically defined in a “mono-directional” way, where the prover gets as input  $x$  and  $w$ , and where the verifier learns  $x$  and whether  $x$  belongs to  $\mathcal{L}$ . Quantumly, the prover does get  $\omega$  as input (analog of  $w$ ), but instead of receiving the classical description of  $\rho$  (the analog of  $x$ ), it *outputs*  $\omega_s$ , so that  $(\omega, \omega_s)$  (partially) describes  $\rho$ . One might wonder why  $\omega_s$  is not sent as an *input*: While this would certainly be possible, because of the fundamental non-deterministic nature of quantum mechanics, the qubit obtained by the receiver will typically *not* belong to  $\mathcal{L}_{\omega, \omega_s}$  after a single round of interaction (typically, while the basis is always the same, the encoded bit is random), so we would need another round of communication to correct the quantum state. In practice, the exact  $\omega_s$  (encoded bit) does not really matter (but we still want to know its value of course), but we do want to optimize the number of rounds of communications.
- The second question that one might ask is why we only describe *partially*  $\rho$  with  $(\omega, \omega_s)$  instead of describing the full classical description of  $\rho$  (in practice we do not reveal the bit encoded in the qubit in the computational basis). This can be explained since if we send the full description of  $\rho$ , this gives too much information to the adversary (distinguisher), to the point that we are unable to prove the security of the protocol. However, in practice this is not an issue, since the discarded information on  $\rho$  is typically a useless random value, not needed in the rest of the protocol.

**Extensions, and formalisation of ZKoQS and quantum language.** In the rest of the article, we formalize the notion of quantum language (Definition 4.3) and Zero-Knowledge on Quantum states (ZKoQS, Definition 4.5). We define the corresponding complexity classes  $\text{ZKstatesQIP}_S[k]$  and  $\text{ZKstatesQMA}_S$  (Definition 4.16). While ZKoQS is quite generic, it does not translate naturally to an ideal functionality, useful to prove the security of protocols in the simulation-based and composable quantum standalone framework [HSS11]. As a result, we define a relatively generic ideal functionality that is in charge of applying some measurement operators (Definition 4.8), and we prove that under some assumptions on the measurement operators (called postponable measurements, Definition 4.9), this functionality implies ZKoQS (Theorem 4.10). While for now we do not know a realization of this functionality for any measurement operator, we consider a particular case (Definition 4.11) where the functionality is in charge of measuring a subset  $T$  of qubits (such that  $\text{Pred}(T) = \top$  for an arbitrary predicate  $\text{Pred}$ ) and rotating randomly the other qubits. We show in Theorem 4.12 how to realize this functionality, and we prove in Corollary 4.15 that it is a ZKoQS functionality for the language  $\mathcal{L}_{\text{SemCol}}^{\text{Pred}}$  of semi-collapsed states (Definition 4.14). We provide in Corollary 4.18 the implications in term of complexity theory (e.g.  $\mathcal{L}_{\text{SemCol}}^{\text{Pred}}$  is in  $\text{ZKstatesQMA}^{\text{RO}}$ ). We also show in Theorem 4.19 that this functionality can be used to realize a very generic notion of OT protocol that we call  $\text{Pred-OT}$ , and in particular string-OT and  $k$ -out-of- $n$  OT (Corollary 4.20). Finally, since our result requires the use of (NI)ZK protocols, we prove in Section 5 that the non-interactive protocol of [Unr15] (proven secure in the RO model) can be expressed in the quantum standalone framework, and can therefore be used in our protocol ([HSS11] already provides another interactive protocol in the plain-model).

---

to physically check if a state belongs to a set, since some distributions of quantum states are different but still indistinguishable.

### 1.3 Concurrent work

A few months after releasing our article online, a related and independent article was posted on the arXiv [BKS23], but as noted in [BKS23], our contributions are orthogonal, with completely different methods. They indeed assume that adversaries share EPR pairs before starting the protocol (which is a strong assumption), but they show that in this sufficient to obtain 1-message OT assuming the hardness of (sub-exponential) LWE (requiring public-key cryptography), and a 2-message OT in the random oracle setting. See Fig. 1 for a detailed comparison.

### 1.4 Open problems and ongoing works

We expect our method used to build non-interactive OT to be of independent interest, which also raises a number of open questions:

- **Reducing entanglement:** our protocols require the preparation of states representing a superposition of bit strings, and the application of a hash function  $h$  in superposition. For practical considerations, it would be great to see if we could get 2-message quantum OT protocols and/or ZKoQS with single-qubit operations (or prove impossibility results).
- **Universal composability:** the model of security we are using allows sequential composability but not parallel composability. A priori, we expect our proof method to extend to a general composability framework like Composable Cryptography or Universal Composability, but we also need to find ZK protocols secure in this stronger model of security (note that [Unr15] already provides online extractability and is therefore certainly a good starting point).
- **Characterization of  $\text{ZKstatesQIP}$  and  $\text{ZKstatesQMA}$ :** For now we have only proven the belonging of a small class of quantum languages in  $\text{ZKstatesQMA}^{\text{RO}}$  and  $\text{ZKstatesQIP}_S^{\text{pm}}$ , but it would be thrilling to study the set of quantum languages that belong (or does not belong) to the various classes  $\text{ZKstatesQIP}$  and  $\text{ZKstatesQMA}$ . For instance it would be interesting to see if it is possible to prove that states belong to the Hadamard basis or to the computational basis (methods inspired by quantum money might be useful).
- **ZK for statistical security:** While our approach states that we can get quantum OT with statistical security assuming the existence of statistical ZK argument of knowledge (for unbounded verifier/sender) or ZK proof of knowledge (for unbounded prover/receiver), it is important to check that such protocols exist (for now the protocols we analyse only bring computational security, which results in a computationally secure OT, like [ABK<sup>+</sup>22]). There are countless classical candidates and ways to analyse them quantumly ([Wat09, AL20, BS20, LMS21], especially with the recent breakthrough of [LMS21], but each construction often uses their own slightly different definitions of ZK. Therefore, a proper analysis is needed to see which protocol fits in the quantum standalone framework. Similarly, finding a ZK in the plain model not based on trapdoors could provide a simpler proof for the results of [GLS<sup>+</sup>21, BCK<sup>+</sup>21], and even if candidates exists, we have not yet analysed them properly to see if they fit in the quantum standalone framework. Finally, the ZK construction [Unr15] that we use to get 2-message OT is in the random oracle model, and it would be great to obtain a similar ZK construction in the CRS model.
- **Even weaker assumptions:** the hash function needs to be hiding (our actual assumption is actually slightly weaker), but we don't know if we can reduce this assumption to use only one-way functions (we sketch a construction based on the Goldreich-Levin theorem, but this still need to be analysed formally). Moreover, pseudo-random states [JLS18] were introduced to provide an ever lower assumption compared to one-way functions. They are known to

imply OT [BCK<sup>+</sup>21, AQY22], but it is unclear if our approach could lead to more efficient protocols.

- **Reducing complexity:** for now, when doing string OT of size  $n$  we sample  $2n$  random  $w$ 's, and therefore we need to do  $2n$  ZK proofs on them. However, it might seem reasonable to use the same randomness for the first  $n$  bits, and a second randomness for the last  $n$  bits, leading to a much shorter ZK proof. It could also be great to see if it is possible somehow to re-use the same quantum register containing the randomness to also lower the quantum complexity for string OT.
- **Reducing communication in the plain model:** while our approach can get us to the optimal round-complexity (2 messages), such optimal complexity cannot be obtained in the plain-model, at least in a composable framework. It would be interesting to study the minimum number of rounds in the plain-model (but staying in MiniQCrypt), possibly giving up on composable proofs.
- **Applications:** While OT is definitely an important application for the ZKoQS protocol, we expect ZKoQS to find applications in other fields. Exploring the potential applications would therefore be an interesting line of research.
- **Weaker ZK protocols:** For now we assume the existence of a ZK protocol for NP, but we informally mainly want to prove that some classical languages contain few elements, which might be more efficient to realize than with a fully fledged ZK protocol for NP. Studying the links with witness indistinguishability or witness elimination [KZ09] might also be nice to see how we can weaken the assumptions. Moreover, interestingly we don't need the PoK property to extract the  $m_b$ 's, only to get the value of  $b$ . It might be interesting to see if we can get rid of the PoK assumption of the ZK protocol.
- **Comparison of quantum communication:** while our approach potentially needs non-trivial quantum operations on the server side (notably applying  $h$  in superposition, note that all ZK operations are fully classical), the quantum communication seems relatively low compared to other works like [ABK<sup>+</sup>22]. The reason is that we only send the randomness  $w$ , so if we take a randomness of size 160 (that should be enough to avoid brute-force attacks and quadratic improvement in grover-like attacks), we can transmit  $2 \times 161 = 322$  qubits to get 80 bits of security. Instead, our understanding of [ABK<sup>+</sup>22] is that we need to send  $3200\lambda = 256\,000$  qubits for a similar security guarantee. However, a proper analysis should be made.

## 2 Preliminaries

### 2.1 Notations

We assume basic familiarities with quantum computing [NC10]. For any Hermitian matrix  $A$ , we denote its trace norm as  $\|A\|_1 := \text{Tr}(\sqrt{A^\dagger A}) = \sum_i |\lambda_i|$  where  $\lambda_i$ 's are the eigen-values of  $A$  (considered with their multiplicity). We denote the trace distance between two density matrices  $\rho$  and  $\sigma$  as  $\text{TD}(\rho, \sigma) := \frac{1}{2} \|\rho - \sigma\|_1$ . A bipartite state between two registers or parties  $\mathbf{A}$  and  $\mathbf{B}$  will be denoted  $\rho^{\mathbf{A}, \mathbf{B}}$ . For any bit string  $x$  and  $x'$ ,  $x[i]$  is the  $i$ -th element of  $x$ , starting from 1, and  $\langle x, x' \rangle := \bigoplus_i x[i]x'[i]$ . For a gate  $Z$  and a quantum state  $|\psi\rangle$ ,  $Z^{\mathbf{B}, i} |\psi\rangle_{\mathbf{B}, \mathcal{E}}$  represents the state obtained after applying  $Z$  on the  $i$ -th qubit of the register  $\mathbf{B}$  of  $\psi$  (we might omit the register when it is clear from the context). We might abuse notations and consider that outputting true is the same as outputting 1, but for more complex formulas  $P$  it can be handy to define  $\delta_P \in \{0, 1\}$  such that  $\delta_P = 1$  iff  $P$  is true.

## 2.2 Model of security

We follow the quantum stand-alone security model defined in [HSS11] that we quickly summarize here.

**Quantum Interactive Machines (QIM).** In this model, a quantum interactive machine (QIM)  $\mathbf{A} = \{A_\lambda\}_{\lambda \in \mathbb{N}}$  is a sequence of quantum circuits  $A_\lambda$  indexed by the security parameter  $\lambda$  working on an input, output and network register. Two machines can interact by sharing their network register while they are activated alternately. A (two-party) protocol  $\Pi = (\mathbf{A}, \mathbf{B})$  is a couple of QIM. We denote by  $\mathbf{A} \rightsquigarrow \mathbf{B}$  the sequence of quantum maps (indexed by  $\lambda \in \mathbb{N}$ ) representing the interaction between  $A_\lambda$  and  $B_\lambda$ : Namely this map takes as input a quantum state on two registers  $S_A$  and  $S_B$ , provides to  $A_\lambda$  (resp.  $B_\lambda$ ) the input  $S_A$  (resp.  $S_B$ ), let  $A_\lambda$  and  $B_\lambda$  interact and outputs at the end of the interaction the two registers containing the outputs of  $A_\lambda$  and  $B_\lambda$ . We might also write  $z \leftarrow \text{OUT}_{\mathbf{B}}(\mathbf{A}_\lambda(x) \rightsquigarrow \mathbf{B}_\lambda(y))$  instead of  $(\_, z) \leftarrow (\mathbf{A}_\lambda(x) \rightsquigarrow \mathbf{B}_\lambda(y))$  to denote the output of the party  $\mathbf{B}$ . A protocol is said to be *poly-time* if all the parties run in polynomial time. The security of a protocol is expressed with respect to a *functionality*  $\mathcal{F}$  (having no input) playing the role of a trusted third party. A functionality is a QIM interacting with all parties: for two QIM  $\mathbf{A}$  and  $\mathbf{B}$ , we similarly denote as  $\mathbf{A} \overset{\mathcal{F}}{\rightsquigarrow} \mathbf{B}$  the quantum map that forwards the two input registers to  $\mathbf{A}$  and  $\mathbf{B}$  and that returns their outputs after letting both of them interact (only) with  $\mathcal{F}$ , as pictured in Fig. 2. Note that we might provide access to oracles  $H$  (QIM that answer queries to functions, e.g. a random oracle), in which case we will either denote it as  $\mathbf{A}^H \rightsquigarrow \mathbf{B}^H$  or  $\mathbf{A} \overset{H}{\rightsquigarrow} \mathbf{B}$  (in this case  $H$  is the functionality that answers queries and forwards other messages). Moreover, for two sequences of quantum maps  $\mathbf{A} = \{A_\lambda\}_{\lambda \in \mathbb{N}}$  and  $\mathbf{B} = \{B_\lambda\}_{\lambda \in \mathbb{N}}$ , we also define naturally their sequential composition as  $\mathbf{AB} := \{A_\lambda B_\lambda\}_{\lambda \in \mathbb{N}}$ .

**Adversaries.** An adversary  $\mathcal{A}$  is a QIM able to corrupt parties (i.e.  $\mathcal{A}$  will replace the corrupted parties). We consider only *static* adversaries, meaning that  $\mathcal{A} \in \{\hat{\mathbf{A}}, \hat{\mathbf{B}}\}$  chooses before the beginning of the protocol the set of corrupted party. In particular, we denote by  $\hat{\mathbf{A}}$  the adversary that corrupts (and replaces)  $\mathbf{A}$  (similarly  $\hat{\mathbf{B}}$  would corrupt  $\mathbf{B}$ ). We define  $\Pi \rightsquigarrow \mathcal{A}$  as the quantum map obtained when the protocol  $\Pi$  is run in the presence of the adversary  $\mathcal{A}$ : Notably,  $\Pi \rightsquigarrow \hat{\mathbf{A}} = \hat{\mathbf{A}} \rightsquigarrow \mathbf{B}$  and  $\Pi \rightsquigarrow \hat{\mathbf{B}} = \mathbf{A} \rightsquigarrow \hat{\mathbf{B}}$ .

**Real and ideal worlds.** The security relies on the usual *simulation* paradigm involving a real-world and an ideal-world, where the real-world represents a run of the protocol where some parties can potentially be corrupted while the ideal-world paradigm represents an idealized version of the protocol where the parties are only allowed to interact through the trusted ideal functionality. A QIM  $\mathbf{Z}$  called *environment* will be in charge of distinguishing these two worlds. Informally, if both worlds are indistinguishable, the protocol is said secure as any attack doable in the real-world would apply in the ideal-world (otherwise it would provide a way to distinguish both worlds) and therefore on the ideal functionality  $\mathcal{F}$ , which is secure by definition. In order to “fake” a transcript from the real world during an execution of the ideal world, we replace any honest party  $\mathbf{A}$  by a idealized party<sup>13</sup>  $\tilde{\mathbf{A}}$  that honestly interact with  $\mathcal{F}$  (it is typically trivially interacting with  $\mathcal{F}$  by forwarding the inputs and outputs to/from  $\mathcal{F}$  and is therefore often omitted), and we write  $\tilde{\Pi} := (\tilde{\mathbf{A}}, \tilde{\mathbf{B}})$  to denote this dummy protocol. Moreover, to deal with the corrupted parties, we introduce a special kind of adversary  $\mathbf{S}_{\mathcal{A}}$  called a *simulator*, that must corrupt the same party as the adversary  $\mathcal{A}$  and whose goal is to fake the transcript outputted by  $\mathcal{A}$  (i.e. simulate  $\mathcal{A}$ , hence its name).

<sup>13</sup>This is the analogue of filters in constructive cryptography.

We formalize now this concept:

**Definition 2.1.** Let  $\Pi = (\mathbf{A}, \mathbf{B})$  be a two-party protocol,  $\mathcal{A}$  be a static adversary as defined above,  $\mathbf{S}_{\mathcal{A}}$  be a simulator,  $\sigma = \{\sigma_\lambda \in \mathbf{S}_{\mathcal{A}}(\lambda) \otimes \mathbf{S}_B(\lambda) \otimes \mathcal{W}(\lambda)\}_{\lambda \in \mathbb{N}}$  be a sequence of quantum states and  $\mathbf{Z}$  be a QIM called environment outputting a single classical bit. We denote by  $\text{REAL}_{\Pi, \mathcal{A}, \mathbf{Z}}^\sigma := \mathbf{Z}((\Pi \rightsquigarrow \mathcal{A}) \otimes I)\sigma$  the (sequence of) binary random variables outputted by the environment  $\mathbf{Z}$  at the end of an interaction where the adversary  $\mathcal{A}$  corrupts some parties in  $\Pi$ . We define similarly  $\text{IDEAL}_{\tilde{\Pi}, \mathbf{S}_{\mathcal{A}}, \mathbf{Z}}^{\sigma, \mathcal{F}} := \mathbf{Z}((\tilde{\Pi} \xrightarrow{\mathcal{F}} \mathbf{S}_{\mathcal{A}}) \otimes I)\sigma$  as the (sequence of) binary random variables outputted by the environment  $\mathbf{Z}$  at the end of an interaction where the simulator can corrupt some dummy parties interacting with the ideal functionality  $\mathcal{F}$ .

**Definition 2.2 (Indistinguishable random variables).** Two sequences of random variables  $\mathbf{X} = \{X_\lambda\}_{\lambda \in \mathbb{N}}$  and  $\mathbf{Y} = \{Y_\lambda\}_{\lambda \in \mathbb{N}}$  are said to be  $\varepsilon$ -indistinguishable, denoted  $\mathbf{X} \approx_\varepsilon \mathbf{Y}$ , if  $|\Pr[X_n = 1] - \Pr[Y_n = 1]| \leq \varepsilon(\lambda)$ . In particular, if  $\varepsilon = \text{negl}(\lambda)$ ,  $\mathbf{X}$  and  $\mathbf{Y}$  are said to be indistinguishable, denoted  $\mathbf{X} \approx \mathbf{Y}$ .

**Definition 2.3 (Indistinguishable quantum maps).** Two sequences of quantum maps  $\mathbf{X} = \{X_\lambda\}_{\lambda \in \mathbb{N}}$  and  $\mathbf{Y} = \{Y_\lambda\}_{\lambda \in \mathbb{N}}$  are said to be computationally (resp. statistically) indistinguishable, denoted  $\mathbf{X} \approx_c \mathbf{Y}$  (resp.  $\mathbf{X} \approx_s \mathbf{Y}$ ), if for any poly-time (resp. unbounded)  $\mathbf{Z} = \{\mathbf{Z}_\lambda\}_{\lambda \in \mathbb{N}}$  and any sequence of bipartite advices  $\sigma = \{\sigma_\lambda\}_\lambda$ ,  $\mathbf{Z}(\mathbf{X} \otimes I)\sigma \approx \mathbf{Z}(\mathbf{Y} \otimes I)\sigma$ .

**Definition 2.4 (Quantum stand-alone (\*-QSA) realization of a functionality [HSS11]).**

Let  $\mathcal{F}$  be a poly-time two-party functionality and  $\Pi$  be a poly-time two-party protocol. We say that  $\Pi$  computationally quantum-stand-alone ( $\mathcal{C}$ -QSA) (resp. statistically quantum-stand-alone ( $\mathcal{S}$ -QSA)) realizes  $\mathcal{F}$  if for any poly-time (resp. unbounded) adversary  $\mathcal{A}$  there is a poly-time (in the time taken by  $\mathcal{A}$ ) simulator  $\mathbf{S}_{\mathcal{A}}$  such that for any poly-time (resp. unbounded) environment  $\mathbf{Z}$  and family of states  $\sigma = \{\sigma_\lambda\}_{\lambda \in \mathbb{N}}$ ,  $\text{REAL}_{\Pi, \mathcal{A}, \mathbf{Z}}^\sigma \approx \text{IDEAL}_{\tilde{\Pi}, \mathbf{S}_{\mathcal{A}}, \mathbf{Z}}^{\sigma, \mathcal{F}}$ .

Moreover, we extend this definition by saying that  $\Pi$   $\mathcal{CS}_S$ -QSA (where  $S$  is a set of subset of parties realizes  $\mathcal{F}$  when statistical security holds only if the adversary corrupts<sup>14</sup> a set of parties in  $S$  (i.e. if  $\mathcal{A}$  corrupts a set of party in  $S$  then  $\mathcal{A}$  and  $\mathbf{Z}$  are allowed to be unbounded, otherwise they are poly-time). In particular, if the protocol has two parties  $A$  and  $B$ ,  $\mathcal{CS}_\emptyset$ -QSA =  $\mathcal{C}$ -QSA and  $\mathcal{CS}_{\{\emptyset, \{A\}, \{B\}, \{A, B\}\}}$ -QSA =  $\mathcal{S}$ -QSA. Note that because it is always possible to turn malicious parties into honest parties,  $\mathcal{CS}_S$ -QSA implies  $\mathcal{CS}_{S \cup \{X\}}$ -QSA for any  $X \subseteq s$  such that  $s \in S$  (for instance  $\mathcal{CS}_{\{A, B\}}$ -QSA implies  $\mathcal{CS}_{\{\emptyset, \{A\}, \{B\}, \{A, B\}\}}$ -QSA). For this reason, we will consider from now only maximal sets  $S$  with respect to this augmentation procedure and we will often only write the larger set: We will notably be particularly interested in statistical security against a malicious Alice ( $\mathcal{CS}_{\{\emptyset, \{A\}\}}$ -QSA, or  $\mathcal{CS}_A$ -QSA for short) or a malicious Bob ( $\mathcal{CS}_{\{\emptyset, \{B\}\}}$ -QSA, or  $\mathcal{CS}_B$ -QSA for short).

**Some functionalities.** We present here some ideal functionalities used later, starting with the main OT functionality:

**Definition 2.5 (Functionality for bit oblivious transfer  $\mathcal{F}_{OT}$  [HSS11]).** We define the ideal functionality  $\mathcal{F}_{OT}$  for oblivious transfer as follows:

<sup>14</sup>Remember that the adversary is static, and therefore determines the set of parties to corrupt before the beginning of the protocol. Note that we will omit in the proof the case where  $\mathcal{A}$  corrupts all parties as this case is trivial (the simulator can just run the adversary and ignore the functionality).

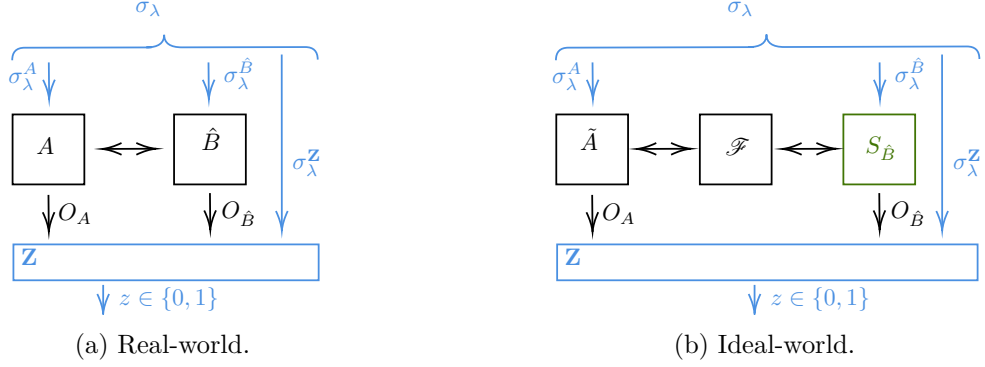


Fig. 2: Real-world and ideal-world executions when Bob is malicious.

- it receives two messages  $m_0$  and  $m_1$  from Bob’s interface, or an abort message
- it receives one bit  $b \in \{0, 1\}$  from Alice’s interface, or an abort message
- if no party decided to abort, it sends  $m_b$  to Alice.

We define trivially the dummy parties  $\tilde{\Pi} = (\tilde{\mathbf{A}}, \tilde{\mathbf{B}})$  that forward the inputs/outputs to/from  $\mathcal{F}_{OT}$ .

We will then prove that our protocol can trivially be extended to more advanced OT functionalities. First, we define a generic functionality where the statements can be proven on any predicate on the bits of the message, we will then consider particular cases like string OT (to receive strings instead of bits) or  $k$ -out-of- $m$  string OT (to receive  $k$  strings among  $n$ ):

**Definition 2.6 (Functionality for predicate oblivious transfer  $\mathcal{F}_{OT}^{\text{Pred}}$ ).** Let  $n \in \mathbb{N}$  and  $\text{Pred}: \mathcal{P}([n]) \rightarrow \{0, 1\}$  be a predicate<sup>15</sup> on any subset of bits. We define the ideal functionality  $\mathcal{F}_{OT}^{\text{Pred}}$  for predicate oblivious transfer as follows:

- It receives  $n$  bits  $(m_i)_{i \in [n]}$  from Bob’s interface, or an abort message.
- It receive a subset  $B \subseteq [n]$  from Alice’s interface (we might also encode  $B$  as a bit string, where  $B[x] = 1$  iff  $x \in B$ ), or an abort message.
- If  $B = \perp$  or  $\text{Pred}(B) = \perp$ , it sends an abort message to Bob.
- If no party decided to abort and  $\text{Pred}(B) = \top$ , it sends  $(m_i)_{i \in B}$  to Alice. Otherwise it sends  $\perp$  to all parties.

We define trivially the dummy parties  $\tilde{\Pi} = (\tilde{\mathbf{A}}, \tilde{\mathbf{B}})$  that forward the inputs/outputs to/from  $\mathcal{F}_{OT}$ .

We define particular cases of interest:

- **String OT:** If  $n = 2m$  and  $\text{Pred}(B)$  is true iff  $B \in \{1^m 0^m, 0^m 1^m\}$  then we call this functionality string OT, denoted  $\mathcal{F}_{OT}^{\text{str}}$  (to send the two messages  $m_a$  and  $m_b$ , we define  $m = m_a || m_b$ ).
- **$k$ -out-of- $m$  string OT:** If  $n = lm$  and  $\text{Pred}(B)$  is true iff  $B = B_1 || \dots || B_m$  with  $\forall i, B_i \in \{0^l, 1^l\}$ , such that the number of  $B_i$ ’s equal to  $1^l$  is equal to  $k$ , then we call this functionality  $k$ -out-of- $m$  string OT, denoted  $\mathcal{F}_{OT}^{k-m}$  (to sent the  $m$  messages  $m_a$  and  $m_b$ , we define  $m = m_a || m_b$ ).

Classical Zero-Knowledge (ZK) proofs allow a party (the *prover*) to prove a statement to another party (the *verifier*) without revealing anything beyond the fact that this statement is true. Our protocols use a ZK protocol as a blackbox. We define now the functionality corresponding to ZK.

<sup>15</sup>This predicate might depend on a secret witness  $w$  known only to the prover, in which case we always replace  $\text{Pred}(\dots)$  with  $\text{Pred}(w, \dots)$ ,  $w$  being sent to the ideal functionalities and used in the ZK proofs. For simplicity, we will omit the witness from now.

**Definition 2.7 (Functionality for zero-knowledge  $\mathcal{F}_{ZK}^{\mathcal{R}}$  [HSS11]).** We define the ideal functionality  $\mathcal{F}_{ZK}^{\mathcal{R}}$  for zero-knowledge, where  $\mathcal{R}$  is a relation describing a given language  $\mathcal{L}$  ( $x \in \mathcal{L} \Leftrightarrow \exists w, x\mathcal{R}w$ ):

- it receives  $(x, w)$  from the prover’s (a.k.a. Alice) interface or an abort message  $\perp$ ,
- if  $x\mathcal{R}w$  then the verifier (a.k.a. Bob) receives  $x$  otherwise it receives  $\perp$ .

This functionality also implies that the ZK protocol is a *proof of knowledge* protocol (PoK, quantumly it is also known as *state-preserving* as extracting the witness should not disturb the state of the adversary) as the functionality can extract the witness. But our protocol could be proven secure in different ways:

- One of them is to assume that the protocol is a state-preserving PoK (PoK is not needed to extract  $m_0$  and  $m_1$  from a malicious Bob, but is handy to extract  $b$  from a malicious Alice). That’s the approach taken in this paper since it has the advantage of applying also in the plain model.
- It should also be possible to obtain similar guarantees without state-preserving PoK, notably by assuming that the simulator can extract the queries made to the oracle (either by relying on Common Reference String (CRS) or on the random oracle model (ROM)). However, this approach is less modular and seems to rely heavily on CRS/RO and is therefore harder to generalize to the plain model. Moreover, we already know state-preserving NIZK PoK in the RO model [Unr15], so this second approach seems less interesting and will not be explored in this article.

Moreover, we often make the distinction between ZK arguments (computational soundness against malicious prover), ZK proofs (statistical soundness against malicious prover) and statistical ZK (ZK also holds against a malicious unbounded verifier). In the quantum stand-alone formalism, ZK proofs are protocols that  $\text{CS}_P\text{-QSA}$  realize  $\mathcal{F}_{ZK}^{\mathcal{R}}$  and statistical ZK are protocols that  $\text{CS}_V\text{-QSA}$  realize  $\mathcal{F}_{ZK}^{\mathcal{R}}$ .

Note that nearly all the properties of our protocol reduce to the properties of the ZK scheme. If we use a Non-Interactive ZK (NIZK) protocol secure in the Quantum Random Oracle (OT) model or in the Common Reference String (CRS) model, then our final protocols will be optimal in term of round complexity (2-message OT, or 1-message NIZKoQS) but will rely on the RO or CRS assumption. On the other hand, we may prefer to use a  $n$ -message NIZK protocol in the plain model: in that case our protocols will be secure in the plain model, and the communication complexity will be  $n$  for the NIZKoQS protocol, resulting in an  $n + 1$ -message OT protocol.

There are multiple protocols realising the  $\mathcal{F}_{ZK}^{\mathcal{R}}$  functionality, either in the plain model [HSS11] or non-interactively in the random-oracle model [Unr15] (this last work is not expressed in the quantum stand-alone model, but we prove in Section 5 that it can be reformulated in this framework).

Because we are dealing with non-uniform adversaries, we need to sample hash functions independently of the non-uniform advice, and this is usually done via a Common-Reference-String (CRS) assumption. CRS assumes that a string, honestly sampled according to a fixed procedure, can be shared among all parties (this is typically not counted in the communication as in practice we can often heuristically take a publicly known string instead, for instance by feeding the generation procedure with a known uniformly sampled string... unless the sampling needs trapdoor which is not our case here). While this adds an assumption, it can be practical sometimes to obtain more efficient protocols (in term of communication complexity), and often can be heuristically replaced by a publicly known string (e.g. if the string contains the description

of a collision resistant function like in our case, we might pick the well known SHA-256 hash function instead). Note that our protocol can also be realized without a CRS assumption at the cost of an additional message as discussed in Section 3.2 and in Lemma 2.14. We model CRS as an ideal functionality:

**Definition 2.8.** *Let  $\text{Gen}$  be a PPT sampling procedure. Then the ideal functionality  $\mathcal{F}_{CRS}^{\text{Gen}}$  samples  $x \leftarrow \text{Gen}(1^\lambda)$  and outputs  $x$  to all parties.*

**Hybrid models.** For the sake of modularity, it is often handy to express a protocol realizing a functionality  $\mathcal{F}$  assuming that there exists another (unspecified) protocol realizing a more primitive functionality  $\mathcal{G}$  ( $\mathcal{G}$  might also be considered as a setup assumption, like in the CRS model). To that end, we denote as  $\Pi^{\mathcal{G}}$  a protocol where each party can interact with a trusted party running  $\mathcal{G}$  (of course each party having only access to their respective interface), and we say that we are in the  $\mathcal{G}$ -hybrid model. The  $\mathcal{F}_{CRS}$ -hybrid model is also called the *CRS model*, and if not such assumption is made, we say that we are in the *plain model*. Importantly, if a protocol realizes  $\mathcal{G}$  and if a  $\mathcal{G}$ -hybrid protocol realizes  $\mathcal{F}$ , then combining both protocols in the natural way gives a (non-hybrid) protocol realizing  $\mathcal{G}$ .

### 2.3 Cryptographic requirements.

Before stating our security guarantees, we need to define some security definitions. A function is said to have a hardcore second-bit if it is hard to find the second bit of  $x$  given  $h(x)$  (note that this notion is weaker than the more standard notion of hiding as we only need to hide a single bit). More formally:

**Definition 2.9 (Hardcore second-bit).** *We say that a function  $h$  has a computational (resp. statistical) hardcore second-bit property if there exists two polynomials  $n$  and  $m$ , such that for any  $l \in \{0, 1\}$ , any QPT (resp. unbounded) adversary  $\mathcal{A}$  and for any advice  $\sigma = \{\sigma_\lambda\}_{\lambda \in \mathbb{N}}$ :*

$$\left| \Pr \left[ \mathcal{A}(\lambda, \sigma_\lambda, h(x)) = 1 \mid x \xleftarrow{\$} \{l\} \times \{0\} \times \{0, 1\}^{n(\lambda)} \right] - \Pr \left[ \mathcal{A}(\lambda, \sigma_\lambda, h(x)) = 1 \mid x \xleftarrow{\$} \{l\} \times \{1\} \times \{0, 1\}^{n(\lambda)} \right] \right| \leq \text{negl}(\lambda) \quad (1)$$

*We extend this definition to a family of functions  $\{h_k: \{0, 1\}^{n(\lambda)} \rightarrow \{0, 1\}^{m(\lambda)}\}_{k \in \mathcal{K}}$  if for any  $k \in \mathcal{K}$ ,  $h_k$  has a computational hardcore second-bit property, and if one can efficiently check for any  $k$  whether  $k \in \mathcal{K}$  or not.*

We note that many functions have (or are expected to have) a hardcore second-bit property, in particular since it can be seen as a special case of hiding. It is the case for random functions (e.g. in the RO model), where it is even possible to get statistical security if the function is lossy (i.e. many inputs map to the same output), and we expect it to be true for hash functions used nowadays since they are believed to be hiding. We note that people often consider a weaker assumption called hardcore bit predicate (even achievable from any one-way function thanks to the Goldreich-Levin construction [GL89]), where the unknown bit is a fixed predicate  $b(x)$  instead of the second bit of  $x$ . While we believe that our construction could be adapted to that setting (by doing a rejection sampling to find  $x$  such that  $b(x)$  has the right value), this complicates the constructions, so we leave this extension for further work. We will therefore keep this construction for future works.



**Definition 2.10 (Collision resistance).** A family of functions  $\{h_k: \{0, 1\}^{l(\lambda)} \rightarrow \{0, 1\}^{m(\lambda)}\}_{k \in \mathcal{K}}$  is said to be (computationally) collision-resistant if there exists a polynomial generation algorithm  $k \leftarrow \text{Gen}_h(1^\lambda)$  such that for any  $k \in \mathcal{K}$ ,  $h_k$  can be classically evaluated in polynomial time, and for any (potentially non-uniform) QPT adversary  $\mathcal{A}$  and advice  $\{\sigma_\lambda\}_{\lambda \in \mathbb{N}}$ :

$$\Pr \left[ x \neq x' \wedge h_k(x) = h_k(x') \mid k \leftarrow \text{Gen}_h(1^\lambda), (x, x') \leftarrow \mathcal{A}(k, \sigma_\lambda) \right] \leq \text{negl}(\lambda) \quad (2)$$

*Remark 2.11.* Note that we do not directly require the functions to be collapsable [Unr16]—which is often required when considering quantum adversaries—as we can show that any attack leads to the finding of a collision. However, we do require the existence of a ZK proof of knowledge scheme, that may, in turn, require the existence of such a function. Moreover, when considering unbounded provers, the function is expected to be statistically collision-resistant, i.e. injective, and is therefore collapsing.

Note that even if we heuristically expect the protocol to stay secure when we replace  $h_k$  with a fixed hash function like SHA-256, to prove the security we need to sample the function  $h_k$  after the beginning of the protocol. The reason is that the adversaries are non-uniform (i.e. get an arbitrary advice), and the advice could contain a collision if it was chosen after  $h_k$ . As a result, one needs to decide who is going to sample  $h_k$ , leading to various tradeoffs:

- If we let a user<sup>16</sup> sample the function, then we need to send an additional message from Bob to Alice, but on the other side we are in the plain-model.
- Otherwise, we can assume that the circuit of  $h_k$  is provided by a CRS, which requires no additional round of communication, but we are not anymore in the plain-model.

In order to keep the proof independent of this choice, we abstract the distribution of the value of  $h_k$  in an ideal functionality:

**Definition 2.12.** Let  $\{h_k: \{0, 1\}^{l(\lambda)} \rightarrow \{0, 1\}^{m(\lambda)}\}_{k \in \mathcal{K}}$  be a family of collision resistant functions generated by  $\text{Gen}$ , with a hardcore second-bit property. Then, we define the ideal functionality  $\mathcal{F}_H^{\text{Gen}}$  as follows.  $\mathcal{F}_H^{\text{Gen}}$  receives an input  $c$  from Bob’s interface, if  $c = \top$ , the functionality samples  $k \leftarrow \text{Gen}(1^\lambda)$  and sends  $k$  to both parties, otherwise if  $c \in \mathcal{K}$ , it forwards  $c$  to Alice’s interface. The ideal party  $\mathbf{A}_I$  just forwards the received  $k$ , while the ideal party  $\mathbf{B}_I$  sends  $c = \top$  to the functionality and outputs the received  $k$ .

We prove now that this functionality can be realized in the plain-model with one message or non-interactively in the CRS model.

**Lemma 2.13 ( $\mathcal{F}_H$  in the CRS model).** In the CRS model (a.k.a.  $\mathcal{F}_{\text{CRS}}^{\text{Gen}}$ -hybrid model), the trivial 0-message protocol where both Alice and Bob output the value given by  $\mathcal{F}_{\text{CRS}}^{\text{Gen}}$  realizes the functionality  $\mathcal{F}_H^{\text{Gen}}$ .

## 2.4 Proofs

See [proof](#) in [section A](#).

**Lemma 2.14 ( $\mathcal{F}_H$  in the plain model).** The 1-message protocol where Bob samples  $x \leftarrow \text{Gen}(1^\lambda)$  and sends  $x$  to Alice, and Alice outputs  $x$  only if  $x \in \mathcal{K}$  realizes the functionality  $\mathcal{F}_H^{\text{Gen}}$  in the plain model.

See [proof](#) in [section A](#).

---

<sup>16</sup>Only Bob can sample the function as collision resistance must hold against Alice and a malicious Alice could cheat when generating the function.

### 3 Protocol for bit OT

#### 3.1 The protocol

While we will define formally ZKoQS later, together with more advanced OT protocols (string-OT,  $k$ -out-of- $n$  OT...), in this section we provide a self-contained description and security proof of our bit-OT protocol. For an intuitive explanation of our protocol, we refer to the overview in Section 1.2. The bit OT protocol is described in .

#### 3.2 Security proof

We prove now our main theorem, i.e. that Protocol 1 securely realizes the OT functionality.

**Theorem 3.1 (Security and correctness).** *Let  $\{h_k\}_{k \in \mathcal{K}}$  be a family of collision resistant functions sampled by **Gen**, having the hardcore second-bit property (Definition 2.9). Let  $\Pi_h = (\mathbf{A}_h, \mathbf{B}_h)$  be a protocol<sup>17</sup>  $\mathcal{CS}_{S_h}$ -QSA realizing  $\mathcal{F}_{CRS}^{\text{Gen}}$  and  $\Pi_{z_k} = (\mathbf{A}_{z_k}, \mathbf{B}_{z_k})$  be a protocol that  $\mathcal{CS}_S$ -QSA realizes the ZK functionality  $\mathcal{F}_{ZK}^R$ , where  $(h_0^0, h_1^0, h_0^1, h_1^1) \mathcal{R}(w_0^0, w_1^0, w_0^1, w_1^1) \Leftrightarrow \forall c, d, h(d \| w_d^c) = h_d^c$  and  $\exists c, d$  such that  $w_d^c[1] = 1$ .*

*Then the Protocol 1, in which  $h$  is obtained by first running  $\Pi_h$ ,  $\mathcal{C}$ -QSA realizes the functionality  $\mathcal{F}_{OT}$ . More precisely, it  $\mathcal{CS}_{S'}$ -QSA realizes  $\mathcal{F}_{OT}$  for any set  $S'$  of unbounded parties such that:*

- $S' \subseteq S \cap S_h$ ,
- $\{\mathbf{B}\} \in S'$  only if  $h$  has the statistical hardcore second-bit property,
- $\{\mathbf{A}\} \in S'$  only if for any  $k \in \mathcal{K}$ ,  $h_k$  is injective (i.e. statistically collision resistant).

*Sketch of proof.* For a first intuitive proof of the correctness and security, we refer to the corresponding paragraph in Section 1.2. We provide here only a sketch of the proof, and we refer the reader to the [full security proof in section B](#).

**Malicious sender (Bob).** We consider the case where the adversary  $\mathcal{A} = \hat{\mathbf{B}}$  corrupts the sender Bob. Informally the goal of the simulator  $\mathbf{S}_{\hat{\mathbf{B}}}$  is to extract the two values  $m_0$  and  $m_1$  from  $\hat{\mathbf{B}}$  to provide these two values to the ideal functionality. To that end, at a high level, the simulator will interact with  $\hat{\mathbf{B}}$  by providing a transcript that an honest Alice could provide, except that  $|\psi^{(1-b)}\rangle$  is sampled like  $|\psi^{(b)}\rangle$ : since the state is now in the Hadamard basis, it can also recover  $m_{1-b}$  following the procedure used by Alice to recover  $m_b$ . However, because it is now impossible to run the ZK proof (because the statement is not even true!) the simulator will run instead the simulator of the ZK proof to convince the distinguisher that the statement is true while it is not. To prove that this simulator is valid, we write a series of hybrid games: we start from the protocol where Alice is honest, then we replace the ZK proof with the simulated proof (indistinguishable by the ZK property). In the next step we sample  $w_{1-b}$  as a non-dummy witness (i.e. starting with a 0, indistinguishable because the function  $h$  is hiding). Then we set  $|\psi^{(1-b)}\rangle = |0\rangle |w_0^{(1-b)}\rangle + (-1)^{r^{(1-b)}} |1\rangle |w_1^{(1-b)}\rangle$  where  $r^{(1-b)} \leftarrow \{0, 1\}$  is sampled uniformly at random (indistinguishable because the density matrices are equal: for any (potentially known) string  $x$  and  $y$ ,  $\frac{1}{2}(|x\rangle\langle x| + |y\rangle\langle y|) = \frac{1}{4} \sum_{r \in \{0,1\}} (|x\rangle + (-1)^r |y\rangle)(\langle x| + (-1)^r \langle y|)$ ). Note that one might be worried that the output of Alice leaks additional information on this quantum state:

<sup>17</sup>As a reminder, this protocol is sampling and distributing a function  $h$  according to **Gen**, and can either be done without communication in the CRS model (or heuristically if we replace  $h$  with a well known collision-resistant hash function), or with one message in the plain model.

---

**Protocol 1:** Protocol for 2-message chosen bit Oblivious Transfer

---

**Alice** ( $b \in \{0, 1\}$ )

**Bob** ( $(m_0, m_1) \in \{0, 1\}^2$ )

---

// Witness for  $\mathcal{L}^{(b)} = \{0, 1\}$ :

$\forall d \in \{0, 1\}, w_d^{(b)} \stackrel{\$}{\leftarrow} \{0\} \times \{0, 1\}^n$

// Witness for  $\mathcal{L}^{(1-b)} = \{l\}$ :

$l \stackrel{\$}{\leftarrow} \{0, 1\}$

$w_l^{(1-b)} \stackrel{\$}{\leftarrow} \{0\} \times \{0, 1\}^n$

$w_{1-l}^{(1-b)} \stackrel{\$}{\leftarrow} \{1\} \times \{0, 1\}^n$

// Compute the characterization

// of the languages:

$\forall (c, d) \in \{0, 1\}^2, h_d^{(c)} := h(d \| w_d^{(c)})$

// Proof that at least one language

// contains a single element

$\pi := (\text{NI})\text{ZK proof that:}$

$\exists (w_d^{(c)})_{c,d}, \forall c, d, h_d^{(c)} = h(d \| w_d^{(c)})$

and  $\exists c, d$  s.t.  $w_d^{(c)}[1] = 1$ .

// Define the quantum states:

$r^{(b)} \stackrel{\$}{\leftarrow} \{0, 1\}$

$|\psi^{(b)}\rangle := |0\rangle |w_0^{(b)}\rangle + (-1)^{r^{(b)}} |1\rangle |w_1^{(b)}\rangle$

$|\psi^{(1-b)}\rangle := |l\rangle |w_l^{(1-b)}\rangle$

$\xrightarrow{\forall (c, d) : h_d^{(c)}, \pi, |\psi^{(0)}\rangle, |\psi^{(1)}\rangle}$

// Check that one language has size  $\leq 1$ :

Check (or run if interactive proof)  $\pi$ .

// Check that the state contains a superposition

// of (valid) elements of  $\mathcal{L}^{(0)}$  and  $\mathcal{L}^{(1)}$ :

$\forall c$ , apply on  $|\psi^{(c)}\rangle |0\rangle$  the unitary:

$x, w \mapsto w[1] \neq 1 \wedge \exists d, h(x \| w) = h_d^{(c)}$ ,

measure the last (output) register

and check that the outcome is 1.

$\forall c$ , measure the second register of  $|\psi^{(c)}\rangle$

in the Hadamard basis (outcome  $s^{(c)}$ ).

At that step,  $|\psi^{(b)}\rangle = |0\rangle \pm |1\rangle$   
and  $|\psi^{(1-b)}\rangle = |l\rangle$ , but Bob does  
not know  $b$  (NIZKoQS).

..... End of NIZKoQS .....

$\forall c$ , apply  $Z^{m_c}$  on  $|\psi^{(c)}\rangle$  and measure it

in the Hadamard basis (outcome  $z^{(c)}$ ).

$\xleftarrow{\forall c, s^{(c)}, z^{(c)}}$

Compute  $\alpha := r^{(b)} \oplus \langle s^{(b)}, w_0^{(b)} \oplus w_1^{(b)} \rangle$

**return**  $\alpha \oplus z^{(b)}$  // Should be  $m_b$

---

$h$  is a collision-resistant (Definition 2.10) and second-bit hardcore (Definition 2.9) function distributed using  $\mathcal{F}_H$  (Definition 2.12), either non-interactively via a CRS, heuristically using a fixed hash function, or sent by Bob, adding an additional message (Lemma 2.14)

If the ZK proof is interactive, then we actually run the ZK protocol (before sending the quantum state) instead of sending the proof (of course this adds additional rounds of communication).

however, the output of Alice is linked with the *other*, non-dummy, quantum state and any additional information regarding this dummy state are anyway discarded. Finally, we can now apply the decoding performed by Alice on both outputs and output only the one corresponding to  $m_b$ : this is exactly the role of the ideal functionality. Since nothing depends on any secret (except this very last step where the functionality discards  $m_{1-b}$  and outputs  $m_b$ ), the simulator can fully run this procedure. See the [full security proof](#) in [section B](#) for more details.

**Malicious receiver (Alice).** We consider now the case where the adversary  $\mathcal{A} = \hat{\mathbf{A}}$  corrupts the receiver Alice.

Informally the goal of the simulator  $\mathbf{S}_{\hat{\mathbf{A}}}$  is to extract the value  $b$  from Alice in order to provide this value to the ideal functionality, and to appropriately use the  $m_b$  provided by the functionality to fake measurement outcomes expected by Alice. At a high level, since the ZK protocol is a (state-preserving) proof (or argument) of knowledge (PoK), we can use this property to extract the witnesses  $(w_d^{(c)})_{c,d}$ . From this witness we can find a  $w_d^{(b)}$  that starts with a 1 in order to learn  $b$ . Then, to fake the measurement outcomes, the simulator can apply exactly the same quantum operations as the one done by the honest Bob, using the  $m_b$  given by the functionality, except that the simulator will choose  $m_{1-b} = 0$ . Note that if the malicious Alice really sent a state  $|\psi^{(1-b)}\rangle$  in the computational basis, then the  $Z^{m_{1-b}}$  rotation does nothing, irrespective of the value of  $m_{1-b}$ . Now, if Alice sent a state that is in superposition of two pre-images with non-negligible amplitude, since it must pass the test checking that it contains non-dummy preimage of  $h$ , then it means that Alice “knows” a collision for  $h$ ... or rather, we can measure the state to get a first preimage and compare it with the preimages extracted during the ZK protocol to get another preimage: with non-negligible probability (on the measurement outcome) they will be different, breaking the collision resistant property of  $h$  which contradicts our assumption. Note that some care must be taken as the probability of finding a collision differs across runs, but we can formalize this argument as shown in the full proof. In practice, we will define a few hybrid games, by first replacing the distribution of  $h$  and the ZK protocol by their simulated versions (since the ZK is a PoK, the simulator can learn  $b$  and the preimages of  $h$ ), then we remove the  $Z^{m_{1-b}}$  rotation (indistinguishable or the state is far from a state in the computational basis, in which case we can recover a collision). Finally, since this does not depend on the secret  $m_{1-b}$ , we can reorganize the elements to recover the ideal word. See the [full security proof](#) in [section B](#) for more details.  $\square$

## 4 (NI)ZKoQS and $k$ -out-of- $n$ string OT

### 4.1 ZKoQS

The main contribution in our main protocol (Protocol 1) is to provide a method to prove (potentially non-interactively) a statement on a received quantum state without revealing much information beside the fact that this statement is true: we call this property (Non-Interactive) Zero-Knowledge proofs on Quantum State ((NI)ZKoQS), by analogy with their classical analogue. While we have not yet introduced formally this definition in order to provide a self-contained OT protocol and proof, we will address this issue here.

NIZKoQS were introduced in [CGK21], but the protocol we present here is using a very different approach. While [CGK21] can be used to prove more advanced properties on the obtained quantum state, it also has multiple drawbacks that were left as open questions:

- First, while their protocol is purely classical, their approach is fundamentally *incompatible with statistical security* (like other potential approaches based on quantum multi-party

computing [DNS12, DGJ+20, KKL+23], since these protocols build upon classical MPC, which are not only impossible to do with statistical security [Lo97], but they also require OT, which is one application of ZKoQS). A malicious unbounded verifier/receiver can always fully describe the received state. On the other hand, with our approach we can get statistical security for both parties (not as the same time).

- Secondly, [CGK21] relies on lattice based cryptography (LWE), living in Cryptomania, and the protocol is really *costly* to implement in practice as the parameters used in the LWE instance lead to very large functions. On the other side, our approach only relies on hash functions, does not exploit any structure or trapdoors, and is therefore much more efficient.

Note that the definition of ZKoQS introduced in [CGK21] is slightly too restrictive for our setting as their notion of quantum language does not allow states to be  $\varepsilon$ -close to the quantum language, the states cannot be entangled with an adversary, they omit the step where the description is given back to the sender (which is important when the protocol is used in other protocols), and their adversaries are QPT. For this last reason, we introduce different notations inspired by classical ZK proofs: when the prover is unbounded (resp. bounded) we say that we have a ZK *proof* (resp. *argument*) on quantum states, denoted ZKPoQS (resp. ZKAoQS). When the verifier is unbounded, we say that we have a statistical ZKoQS (S-ZKoQS). Note that when the protocol is Non-Interactive (a single message from the prover to the verifier), we replace the “ZK” with “NIZK” in these acronyms. We formalize now these concepts.



(a) Usual setting of classical ZK protocols: we typically have  $x \in \mathcal{L}_w$ , with  $\mathcal{L}_w := \{x \mid x\mathcal{R}w\}$ .

(b) Alternative equivalent setting of classical ZK protocols.



(c) Another alternative setting of classical ZK protocols (e.g. if  $w$  contains  $x$ ). We expect  $x \in \mathcal{L}_w$ .

(d) Setting of ZK on Quantum States:  $\rho$  is the (quantum) equivalent of  $x$ , and  $(\omega, \omega_s)$  can be seen as a (partial, see (cf. Remark 4.2)) classical description of  $\rho$ : we expect  $\rho \in \mathcal{L}_{\omega, \omega_s}$ .

Fig. 3: Parallel between classical ZK and ZK on Quantum States: while classically all the above definitions are mostly equivalent, quantumly we cannot send  $\rho$  (the quantum equivalent of  $x$ ) as an input since the laws of physics forbid us from extracting any information from  $\rho$  without altering it. See also Remark 1.4 and Remark 4.2 for the justification of the choice of  $\omega$  and  $\omega_s$ .

**Quantum language.** First, we define a quantum language (we draw a parallel with classical ZK in Fig. 3, and illustrate this with an example in Example 4.1), which is informally speaking a set  $\mathcal{L}_Q$  of bipartite quantum states on two registers V and P that characterizes all states that a malicious adversary might be able to obtain (the register V being controlled by the honest

verifier, and  $\mathbf{P}$  by the malicious prover and/or the environment<sup>18</sup>). Moreover, we also provide additional information on the honest expected behavior, via sets of (bipartite<sup>19</sup>) quantum states  $\mathcal{L}_\omega \subseteq \mathcal{L}_Q$ : when the prover is given as input a *class*  $\omega$  (the quantum equivalent<sup>20</sup> of witnesses), we expect the final state to belong to  $\mathcal{L}_\omega$ . Because there might be many states in  $\mathcal{L}_\omega$ , the prover will also output a subclass  $\omega_s$  to further describe the final state, interpreted as “the verifier obtained a state belonging to  $\mathcal{L}_{\omega, \omega_s} \subseteq \mathcal{L}_\omega \subseteq \mathcal{L}_Q$ ”.

*Example 4.1.* For instance, one might be interested in  $\mathcal{L}_Q$  defined as the set of states where the registers  $\mathbf{V}$  contains exactly two qubits, where at least one of them is non-entangled with any other qubit and collapsed in the computational basis (think “even if the prover is malicious, any state obtained by the verifier belongs to  $\mathcal{L}_Q$ , i.e. contains at least one qubit collapsed in the computational basis”). For the honest behavior, we can for instance define  $\mathcal{L}_{0,0} = \{|+\rangle|0\rangle, |+\rangle|1\rangle\}$ ,  $\mathcal{L}_{0,1} = \{|-\rangle|0\rangle, |-\rangle|1\rangle\}$ ,  $\mathcal{L}_{1,0} = \{|0\rangle|+\rangle, |1\rangle|+\rangle\}$ ,  $\mathcal{L}_{1,1} = \{|0\rangle|-\rangle, |1\rangle|-\rangle\}$ ,  $\mathcal{L}_0 = \mathcal{L}_{0,0} \cup \mathcal{L}_{0,1}$  and  $\mathcal{L}_1 = \mathcal{L}_{1,0} \cup \mathcal{L}_{1,1}$ : this way, if the prover gets input 0 and outputs 1, the verifier is expected to output a state in  $\mathcal{L}_{0,1} = \{|-\rangle|0\rangle, |-\rangle|1\rangle\}$ : the class  $\omega$  represents the position of the state in the Hadamard basis, and the sub-class  $\omega_s$  represents the value encoded in this state.

*Remark 4.2 (On the choice of definition of  $\omega$  and  $\omega_s$ ).* Note that  $(\omega, \omega_s)$  only partially describes the state (in our example above, we remove the description of the state in the computational basis) as otherwise we are unable to prove the security of the scheme (but the lost information on  $\rho$  is anyway of no interest since it is discarded in the OT protocol). One might also ask why  $\omega_s$  is sent as an output and is not part of the input  $\omega$ : while in some cases it might be possible to move everything inside the input  $\omega$  and remove  $\omega_s$  (e.g. if we got a  $|+\rangle$  instead of a  $|-\rangle$  the prover could send another message “apply an additional  $Z$  gate” to flip the encoded qubit), but this comes at the cost of an additional message. In most applications, the exact value of  $\omega_s$  does not really matter as it is only a random key, while saving an additional round of communication is important.

**Definition 4.3 (Quantum Language).** Let  $E^{\mathbf{V}, \mathbf{P}} = \cup_{(n,m) \in \mathbb{N}^2} \mathcal{L}_o(\mathcal{H}_n \otimes \mathcal{H}_m)$  be the set of finite dimensional quantum states on two registers. A quantum language  $(\mathcal{L}_Q, \mathcal{C}, \mathcal{C}_s, \{\mathcal{L}_{\omega, \omega_s}\}_{\omega \in \mathcal{C}, \omega_s \in \mathcal{C}_s})$  is characterized by a set  $\mathcal{L}_Q \subseteq E^{\mathbf{V}, \mathbf{P}}$  of bipartite quantum states<sup>21</sup>, a set  $\mathcal{C} \subseteq \{0, 1\}^*$  of classes (or witnesses) motivated above, a set  $\mathcal{C}_s \subseteq \{0, 1\}^*$  of sub-classes, and for any  $\omega \in \mathcal{C}$ ,  $\omega_s \in \mathcal{C}_s$ , a set  $\mathcal{L}_{\omega, \omega_s}$  of bipartite quantum states called quantum sub-classes. We also define for any  $\omega$ ,  $\mathcal{L}_\omega = \cup_{\omega_s \in \mathcal{C}_s} \mathcal{L}_{\omega, \omega_s}$  (some of these sets might be empty in case  $\omega$  is not a valid class), and require  $\cup_\omega \mathcal{L}_\omega \subseteq \mathcal{L}_Q$ . Moreover, for any set of quantum states  $\mathcal{L}$ , we define  $\rho \in_\varepsilon \mathcal{L} \Leftrightarrow \exists \sigma \in \mathcal{L}, \text{TD}(\rho, \sigma) \leq \varepsilon$ , and  $\rho \notin_\varepsilon \mathcal{L} \Leftrightarrow \neg(\rho \in_\varepsilon \mathcal{L})$ .

<sup>18</sup>Sometimes, we will write  $(\mathbf{P}, \mathbf{Z})$  instead of  $\mathbf{P}$  to denote a more precise cut between the two sub-registers owned by the prover and the environment.

<sup>19</sup>Contrary to  $\mathcal{L}_Q$  that must represent all states potentially obtainable by a malicious party (hence the need of a second register), here  $\mathcal{L}_\omega$  are only used to denote the states obtainable by honest parties, and can therefore often be seen as a set of states on a single register owned by the verifier. The reason we define it as a bipartite state here is that we might later be interested by the generation of truly bipartite states like graph states.

<sup>20</sup>Note that classically, we can see a witness in two different ways: it can be used to efficiently verify that  $x \in \mathcal{L}$ , but more abstractly it can be seen as a way to partition  $\mathcal{L}$  into multiple  $\mathcal{L}_w$ ’s: in an honest setting, given  $w$ , we expect to have  $x \in \mathcal{L}_w$ , where  $\mathcal{L}_w = \{x \mid x\mathcal{R}w\}$ . Quantumly, we will use this second point of view, as given  $\omega$  (the quantum equivalent of  $w$ ) we expect in an honest setting to have  $\rho \in \mathcal{L}_\omega$ , even if  $\omega$  cannot be used directly to verify that property once  $\rho$  is generated because of the laws of physics.

<sup>21</sup> $\mathcal{L}_Q$  represents informally the set of states that any malicious party can generate, where the first register is the output of the verifier and the second register corresponds to registers potentially controlled by an adversary. Since only  $\mathcal{L}_Q$  is needed to characterize the security of a protocol, it is sometimes called directly the quantum language.

**ZKoQS.** We introduce now ZKoQS, that morally provides three guarantees, similar to classical ZK (cf. Fig. 3):

- **Correctness:** if the parties are honest, the prover is given a class  $\omega$  and ends up with the partial (cf. Remark 4.2) description  $(\omega, \omega_s)$  of the state  $\rho$  obtained by the verifier, i.e. such that  $\rho \in \mathcal{L}_{\omega, \omega_s} \subseteq \mathcal{L}_\omega \subseteq \mathcal{L}_Q$ .
- **Soundness:** if the sender is malicious, the honest receiver still ends up with a state  $\rho \in \mathcal{L}_Q$ .
- **Zero-Knowledge:** if the verifier is malicious, they cannot learn the value of the class/witness  $\omega$ .

*Example 4.4.* To continue our above Example 4.1, the correctness guarantees that given an input bit  $\omega \in \{0, 1\}$ , the  $\omega$ -th qubit of  $\rho$  is  $H|\omega_s\rangle$  while the other qubit is in the computational basis (we lose the information of the encoded value). The soundness mostly guarantees that even if the sender is malicious, the received quantum state contains at least one qubit collapsed in the computational basis. The ZK property guarantees that a malicious verifier cannot learn  $\omega$ , the expected position of the qubit in the Hadamard basis.

Note that the formal definition is given with respect to a “simulator”, simulating the whole protocol (and not anymore a single malicious party as usual), including in the soundness and correctness part (while usually simulators are only used in the ZK part). While we could define it without any simulator to get a more restricted definition (and during a first read, it might actually be easier to replace the simulator with the original process), simulators are helpful for multiple reasons to make the definition more useful:

- **In zero-knowledge:** the typical ZK definitions already use simulators to denote the fact that we can simulate the view of the malicious verifier without access to the witness... Therefore it should come at no surprise that we also use a simulator in the ZK property.
- **In soundness:** In a real protocol, a malicious prover might be able to produce states negligibly close (in trace distance) to the quantum language  $\mathcal{L}_Q$ , but not strictly speaking *in*  $\mathcal{L}_Q$ . One might be tempted to introduce an approximate notion  $\rho \in_\epsilon \mathcal{L}_Q$  taking into account trace distance to fix this issue, unfortunately it is not sufficient as this definition does not take into account states that are statistically speaking far from  $\mathcal{L}_Q$ , but computationally speaking “close” to  $\mathcal{L}_Q$ ... Indeed, sometimes provers might actually be able to produce states far (in trace distance) from any state in  $\mathcal{L}_Q$ , but because they are computationally bounded, they are unable to exploit that fact. This kind of false “attack” can actually be done against our protocol if the function  $h$  is not injective (explaining why we require  $h$  to be injective when considering an unbounded malicious receiver), by simply running the ZK protocol in superposition<sup>22</sup>: in that case the output state might be relatively close to a  $|+\rangle$  or  $|-\rangle$  if  $h$  is well balanced (while we expect the state to be close to  $|0\rangle$  or  $|1\rangle$ ), but a computationally bounded receiver cannot exploit this property as they need to compute all preimages of  $h$  to know if we are close to  $|+\rangle$  or  $|-\rangle$ . Simulators are therefore useful in the soundness definition to capture this “computational distance”, and discard ineffective attacks.
- **In correctness:** Perhaps surprisingly, we also use a simulator in the correctness definition. While this might not be useful when considering only a game-based security notion, we need simulator to prove for instance statements like “If a protocol  $\Pi$  realises a given functionality, then this protocol is a ZKoQS protocol” (see e.g. Theorem 4.10). Without further details on

---

<sup>22</sup>Of course by still measuring the classical transcript to send to the verifier.

$\Pi$ , the correctness of  $\Pi$  only tells us that  $\Pi$  is indistinguishable from a functionality that produces states in  $\mathcal{L}_Q$ , but it does not mean that  $\Pi$  itself produces such states, hence the need of a simulator.

We formalize the notion of ZKoQS:

**Definition 4.5 (Zero-Knowledge Proof on Quantum State (ZKoQS)).**

Let  $\mathcal{L} := (\mathcal{L}_Q, \mathcal{C}, \mathcal{C}_s, \{\mathcal{L}_{\omega, \omega_s}\}_{\omega \in \mathcal{C}, \omega_s \in \mathcal{C}_s})$  be a quantum language (Definition 4.3). We say that a protocol  $\Pi = (\mathsf{P}, \mathsf{V})$  is a ZKoQS protocol for  $\mathcal{L}$ , where  $\mathsf{P}$  takes as input a class  $\omega \in \mathcal{C}$  and outputs a sub-class  $\omega_s \in \mathcal{C}_s$  and<sup>23</sup> a quantum state  $\rho^{\mathsf{P}}$ , and  $\mathsf{V}$  takes no input and outputs a bit  $a$ , that is equal to 1 if  $\mathsf{V}$  does not abort, together with a quantum state  $\rho^{\mathsf{V}}$  (potentially entangled with  $\rho^{\mathsf{P}}$ ), if the following properties are respected:

- **Correctness:** There exists a poly-time simulator  $\mathbf{S}$  and a negligible function  $\varepsilon$  such that  $(\mathsf{P} \leftrightarrow \mathsf{V}) \approx_c \mathbf{S}$ , and for any  $\omega$  such that  $\mathcal{L}_\omega \neq \emptyset$ :

$$\Pr \left[ a = 1 \wedge \rho^{\mathsf{V}, \mathsf{P}} \in \mathcal{L}_{\omega, \omega_s} \mid ((\omega_s, \rho^{\mathsf{P}}), (a, \rho^{\mathsf{V}})) \leftarrow \mathbf{S}(\omega) \right] = 1 \quad (3)$$

- **Soundness:** For any malicious prover  $\hat{\mathsf{P}} = \{\hat{\mathsf{P}}_\lambda\}_{\lambda \in \mathbb{N}}$  (QPT for ZKAoQS, unbounded for ZKPoQS) there exists a simulator  $\mathbf{S}_{\hat{\mathsf{P}}} = \{\mathbf{S}_{\lambda, \hat{\mathsf{P}}}\}_{\lambda \in \mathbb{N}}$  (running in time polynomial in the runtime of  $\hat{\mathsf{P}}$ ) such that  $(\hat{\mathsf{P}} \leftrightarrow \mathsf{V}) \approx_c \mathbf{S}_{\hat{\mathsf{P}}}$  ( $\approx_s$  for ZKPoQS), and such that there exists a negligible function  $\varepsilon$  such that for any sequence of bipartite state  $\{\sigma_\lambda^{\mathsf{P}, \mathsf{Z}}\}_{\lambda \in \mathbb{N}}$  and  $\lambda \in \mathbb{N}$ :

$$\Pr \left[ a = 1 \wedge \rho^{\mathsf{V}, (\mathsf{P}, \mathsf{Z})} \notin \mathcal{L}_Q \mid (\rho^{\mathsf{P}}, (a, \rho^{\mathsf{V}}), \rho^{\mathsf{Z}}) \leftarrow (\mathbf{S}_{\lambda, \hat{\mathsf{P}}}^{\mathsf{P}} \otimes I^{\mathsf{Z}}) \otimes \sigma_\lambda^{\mathsf{P}, \mathsf{Z}} \right] \leq \varepsilon(\lambda) \quad (4)$$

- **Quantum Zero-Knowledge:** For any malicious verifier  $\hat{\mathsf{V}} = \{\hat{\mathsf{V}}_\lambda\}_{\lambda \in \mathbb{N}}$  (QPT for ZKoQS, unbounded for S-ZKoQS), there exists a simulator  $\mathbf{S}_{\hat{\mathsf{V}}}(b, \cdot)$  (where  $b \in \{0, 1\}$  indicates if  $\mathcal{L}_\omega$  is non-empty, and  $\cdot$  represents an additionally quantum input from the environment), and an efficiently computable map  $\xi_\omega(\cdot)$  (such that  $\forall \omega, \xi_\omega$  takes one quantum register as input and outputs a classical message in  $\mathcal{C}_s$  and a quantum state  $\rho^{\mathsf{P}}$ ), both running in polynomial time in the runtime of  $\hat{\mathsf{V}}$ , such that for any  $\omega \in \mathcal{C}$ :

$$(\mathsf{P}(\omega) \leftrightarrow \hat{\mathsf{V}}) \approx_c (\xi_\omega \otimes I)(\mathbf{S}_{\hat{\mathsf{V}}}(\mathcal{L}_\omega \neq \emptyset)) \quad (5)$$

( $\approx_s$  for ZKPoQS)

It can sometimes be handy to cut the protocol into two phases: the honest verifier will output the state  $\rho^{\mathsf{V}}$  at the end of the first **send** phase, while the output of the honest prover will be delivered in a second **describe** phase (allowing the prover to describe the state outputted earlier by the verifier). A ZKoQS protocol where each phase consists of a single message is said to be **non-interactive** (denoted **NIZKoQS**, we can similarly add the “NI” prefix to the previously seen notions, to get **NIZKPoQS**, **S-NIZKoQS**...). Finally, for a set of parties  $S$ , we write **ZKoQSS** to denote the fact that the protocol is **S-ZKoQS** if  $\mathsf{V} \in S$  and **ZKPoQS** if  $\mathsf{P} \in S$ .

Note that in ZK protocols, there is a notion of extractability, where a simulator can extract the witness  $w$  from a valid transcript (not all ZK protocols are extractable). We could define a similar notion here allowing the simulator to extract  $\omega$ , but since  $\mathcal{L}_Q$  might contain states not

<sup>23</sup>  $\rho^{\mathsf{P}}$  will actually not be necessary in our main application, but we still include it in case it turns out to be useful in future applications.



belonging to any  $\mathcal{L}_\omega$  (potentially producible by malicious provers), we need to slightly update the definition of quantum language by also introducing a special “malicious” subclass  $\perp$ , so that  $\mathcal{L}_Q = \cup_\omega(\mathcal{L}_\omega \cup \mathcal{L}_{\omega,\perp})$ , and such that the simulator in the soundness property can extract the  $\omega$  of the state produced by a malicious adversary:

**Definition 4.6 (Extractability).** *A ZKoQS protocol is said to be extractable with respect to  $(\mathcal{L}_{\omega,\perp})_{\omega \in \mathcal{C}}$  ( $\perp$  being a special subclass not belonging to  $\mathcal{C}_s$ ) such that  $\mathcal{L}_Q = \cup_\omega(\mathcal{L}_\omega \cup \mathcal{L}_{\omega,\perp})$ , and such that the soundness property is turned into:*

- **Extractability:** *For any malicious prover  $\hat{P} = \{\hat{P}_\lambda\}_{\lambda \in \mathbb{N}}$ , (QPT for ZKAoQS, unbounded for ZKPoQS) there exists a simulator  $\mathbf{S}_{\hat{P}} = \{\mathbf{S}_{\lambda,\hat{P}}\}_{\lambda \in \mathbb{N}}$  (running in time polynomial in the runtime of  $\hat{P}$ ) such that  $(\hat{P} \leftrightarrow V) \approx_c \mathbf{S}_{\hat{P}} (\approx_s \text{ for ZKPoQS})$ , and such that there exists a negligible function  $\varepsilon$  such that for any sequence of bipartite state  $\{\sigma_\lambda^{P,Z}\}_{\lambda \in \mathbb{N}}$  and  $\lambda \in \mathbb{N}$ :*

$$\Pr \left[ a = 1 \wedge \rho^{V,(P,Z)} \notin (\mathcal{L}_\omega \cup \mathcal{L}_{\omega,\perp}) \mid (\rho^P, (a, \rho^V), \rho^Z, \omega) \leftarrow (\mathbf{S}_{\lambda,\hat{P}}(\sigma_\lambda^P)) \otimes \sigma_\lambda^Z \right] \leq \varepsilon(\lambda) \quad (6)$$

## 4.2 Frequently Asked Questions

We answer here some natural questions regarding ZKoQS and quantum languages to complete the previous discussions.

- **Are quantum languages linear?** Or said differently, if  $|\psi\rangle \in \mathcal{L}_Q$  and  $|\phi\rangle \in \mathcal{L}_Q$ , are linear combinations of  $|\psi\rangle$  and  $|\phi\rangle$  part of  $\mathcal{L}_Q$ ? Not always: while it is possible to define a language stable by linear combination, this property might be undesirable. For instance, the quantum language given in Example 4.1 does not have this property (or our OT protocol would be insecure): indeed, both  $|00\rangle$  and  $|11\rangle$  belong to  $\mathcal{L}_Q$  since they are collapsed states, but  $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$  does not belong to  $\mathcal{L}_Q$  since no qubit is collapsed. Actually, if the prover could send such a Bell pair to the receiver, then it would be possible to learn  $m_0 \oplus m_1$  by simply computing the XOR of the measurement outcomes, breaking the OT protocol.
- **Can the adversary generate states negligibly close to  $\mathcal{L}_Q$  but not strictly in  $\mathcal{L}_Q$ ?** Yes. And this is not surprising: if an adversary deviates in an undetectable way (e.g. by rotating the state with a negligible angle), then this deviation is simply not detectable and would have no consequences in term of security. Note that this does not contradict the definition of soundness that states that the probability of accepting and outputting  $\rho \notin \mathcal{L}_Q$  is negligible, since the  $\rho$  is outputted by the *simulator*, not the adversary directly. So the adversary might always send a state  $\rho'$   $\varepsilon$ -close to  $\mathcal{L}_Q$  while the simulator will always generate a state inside  $\mathcal{L}_Q$  (for instance by doing an undetectable measurement on  $\rho'$  to project it back into  $\mathcal{L}_Q$ ).
- **Why do we define the above notions with respect to a simulator?** The answer to the previous question gives a first element of answer: this way we do not need to define a notion of being  $\varepsilon$ -close to a language. We also give other elements of answer in the paragraph before Definition 4.5.
- **Can we obtain ZKoQS for any quantum language?** No. For instance, if we define  $\mathcal{L}'_Q$  like  $\mathcal{L}_Q$  in the above example, but such that  $\omega$  also contains  $\omega_s$  (the encoded value), then any correct protocol would not be ZK: it is indeed always possible to learn some information on the encoded value by simply measuring the state after rotating it with an angle  $\frac{\pi}{4}$ . Similarly, if  $\omega_s$  contains the encoded value of all states (and not just those that are in the Hadamard basis), our proof method does not work since this additional information might help the

distinguisher. While we do prove that the set of quantum languages verifiable in a ZK way is non-trivial, characterizing this set precisely is an open question.

- **Why isn't ZKoQS unidirectional, like classical ZK?** Classically, the prover has no output, while quantumly they output an additional description  $\omega_s$  of the obtained state. This is actually a *choice that we made for efficiency reasons* ( $\omega_s$  could also be part of the input, just like  $x$  classically). Indeed, due to the fundamental non-deterministic nature of quantum computing, the state obtained by the verifier will be different at each run (in our case the encoded value is random): so if the prover wants a fixed encoded value, an additional correction message must be sent to the verifier, creating additional rounds of communications. But it seems like for most of the applications, we do not really need to fix the value of  $\omega_s$ , we just need to know its value: gaining unidirectionality at the cost of round efficiency was not worth it as it would complicate the construction and add rounds of communication, but there is nothing fundamental here.

### 4.3 Proof of partial measurement: a generic framework to get ZKoQS

While the notion of ZKoQS (Definition 4.5) does not explicitly mention functionalities, it is often handy to model a ZKoQS protocol inside an ideal functionality as it is easier to interpret it and use it inside other protocols. While it is not clear how to translate the ZKoQS definition into a functionality, we provide below a few ideal functionalities that “imply” ZKoQS. We will first see what is a ZKoQS ideal functionality, then we will see a class of functionalities that are ZKoQS, and we will show that our protocol realizes a particular case of these functionalities.

**Definition 4.7 (ZKoQS ideal functionality).** *Let  $(\mathcal{L}_Q, \mathcal{C}, \mathcal{C}_s, \{\mathcal{L}_{\omega, \omega_s}\}_{\omega \in \mathcal{C}, \omega_s \in \mathcal{C}_s})$  be a quantum language (Definition 4.3). We say that an ideal functionality  $\mathcal{F}$  is a ZKoQS (resp. ZKoQS<sub>S</sub>) ideal functionality for  $\mathcal{L}_Q$  iff for any protocol  $\Pi = (\mathsf{P}, \mathsf{V})$  that quantum standalone realizes  $\mathcal{F}$  (resp. CS<sub>S</sub>-QSA-realizes  $\mathcal{F}$ ),  $\Pi$  is a ZKoQS protocol (resp. ZKoQS<sub>S</sub> protocol) for  $\mathcal{L}_Q$  (Definition 4.5).*

The most natural class of ideal functionalities leading to ZKoQS are the ones in which the functionality applies an operation (a partial measurement) on an arbitrary input to enforce some structures on the output state:

**Definition 4.8 (Partial measurement  $\mathcal{F}_{\text{PartMeas}}^{M, f_0}$ ).** *Let  $M := \{M_m\}_{m \in \mathcal{M}}$  be a collection of measurement operators<sup>24</sup> (i.e. operators such that  $\sum_m M_m^\dagger M_m = I$  [NC10, Sec. 2.2.3]), implementable in quantum polynomial time, and let  $f_0: \mathcal{M} \rightarrow \mathcal{C}_s$  be an efficiently computable function<sup>25</sup>. Then, we define the proof of partial measurement functionality  $\mathcal{F}_{\text{PartMeas}}^{M, f_0}$  as follows:*

- $\mathcal{F}_{\text{PartMeas}}^{M, f_0}$  receives a state  $\rho$  from the prover's interface, together with an abort bit  $a$ .
- If  $a = \perp$ , it sends  $\perp$  to both parties and stops.
- Otherwise,  $\mathcal{F}_{\text{PartMeas}}^{M, f_0}$  measures  $\rho$  using  $M$ , obtaining an outcome  $m \in \mathcal{M}$  and a post-measured state

$$\rho' := \xi_m(\rho) := \frac{M_m \rho M_m^\dagger}{\text{Tr}(M_m^\dagger M_m \rho)} \quad (7)$$

- It sends  $\rho'$  to the verifier, and waits back for a message  $f$ , such that either  $f = \perp$  (in which case the functionality sends  $\perp$  to the prover to abort and stops),  $f = \top$  (in which case the ideal functionality redefines  $f := f_0$ ), or  $f$  is an efficiently computable function  $f: \mathcal{M} \rightarrow \{0, 1\}^*$ .

<sup>24</sup>They are the most generic way to represent a measurement.

<sup>25</sup>Informally,  $f_0$  is used to filter some information on the measurement outcome  $m$  during an honest protocol.

– Finally, it sends  $f(m)$  to the prover.

We would like to prove that this functionality is a ZKoQS functionality, but not all such functionalities are ZKoQS (in particular, if the post-measured state contains information on  $\omega$ , it has no chance of being ZK). For this reason, we expect our functionality to have an additional property, intuitively saying that we can postpone the actual measurement *after* sending the quantum state. While this might seem counter intuitive, this can actually be realized exploiting entanglement, and similar techniques were used in previous works to prove security of protocols [DFP<sup>+</sup>14].

**Definition 4.9 (Postponable measurement operator).** *A measurement operator  $M$  outputting a quantum state and a classical measurement outcome is said to be postponable with respect to a collection of sampling procedures  $\{G_\omega\}_{\omega \in A}$  outputting a quantum state if there exist a bipartite state  $\rho^{\mathbf{V},\mathbf{F}}$  and a quantum map  $M'$  taking as input a bipartite system and outputting a measurement outcome  $m'$  such that for all  $\omega \in A$ ,  $MG_\omega \approx_s (I^{\mathbf{V}} \otimes M')(\rho^{\mathbf{V},\mathbf{F}} \otimes G_\omega)$ :*

$$\begin{array}{c} \boxed{G_\omega} \text{---} \boxed{M} \\ \text{---} \end{array} = \begin{array}{c} \boxed{\rho} \\ \text{---} \\ \boxed{G_\omega} \text{---} \boxed{M'} \end{array} \quad (8)$$

We prove now that such a functionality is a ZKoQS functionality for a given quantum language and appropriately defined dummy ideal parties:

**Theorem 4.10 ( $\mathcal{F}_{\text{PartMeas}}^M$  implies ZKoQS).** *Let  $E^{\mathbf{V}_0,\mathbf{P}} = \cup_{(n,m) \in \mathbb{N}^2} \mathcal{L}_o(\mathcal{H}_n \otimes \mathcal{H}_m)$  be the set of finite dimensional quantum states on two registers  $\mathbf{V}_0$  and  $\mathbf{P}$ . Let  $\mathcal{C}$  and  $\mathcal{C}_s$  be two sets, and for any  $\omega \in \mathcal{C}$ , let  $E_\omega \subseteq E^{\mathbf{V}_0,\mathbf{P}}$  be a set of bipartite quantum states. Let  $M := \{M_m\}_{m \in \mathcal{M}}$  be a collection of measurement operators (and  $\xi_m$  as defined in Definition 4.8), and  $f_0: \mathcal{M} \rightarrow \mathcal{C}_s$  be a function. We define for any  $\omega \in \mathcal{C}$  and  $\omega_s \in \mathcal{C}_s$ :*

$$\mathcal{L}_{\omega,\omega_s} := \{\rho^{\mathbf{V},\mathbf{P}} \mid \exists \rho_0^{\mathbf{V}_0,\mathbf{P}} \in E_\omega, m \in \mathcal{M}, \text{ s.t. } \omega_s = f_0(m), \rho^{\mathbf{V},\mathbf{P}} = \xi_m(\rho_0^{\mathbf{V}_0,\mathbf{P}})\} \quad (9)$$

$$\mathcal{L}_\omega := \cup_{\omega_s} \mathcal{L}_{\omega,\omega_s} \quad (10)$$

$$\mathcal{L}_Q := \{(\xi_m \otimes \hat{\xi}_{f_0(m)})\rho^{\mathbf{V},(\mathbf{P},\mathbf{Z})} \mid \rho \in E^{\mathbf{V}_0,(\mathbf{P},\mathbf{Z})}, m \in \mathcal{M}, m \neq \perp, \hat{\xi}_{f_0(m)} \text{ being an arbitrary CPTP map depending on } f_0(m).\} \quad (11)$$

Then, let  $\tilde{\mathbf{P}}$  and  $\tilde{\mathbf{V}}$  be any poly-time ideal parties, such that:

- If  $E_\omega = \emptyset$ ,  $\tilde{\mathbf{P}}(\omega)$  sends the abort bit  $a = \perp$  to the functionality and outputs  $\perp$ . Otherwise,  $\tilde{\mathbf{P}}(\omega)$  produces a state in  $E_\omega$  according to an arbitrary sampling procedure  $G$ , sends the register  $\mathbf{V}_0$  to the ideal functionality, and outputs the  $\omega_s$  given back from the functionality together with the register  $\mathbf{P}$ .
- If  $\tilde{\mathbf{V}}$  receives  $\perp$  from the functionality, it outputs  $a = \perp$  and stop. Otherwise, it outputs the state  $\rho'$  given by the functionality together with a bit  $a = \top$  and sends back to the functionality  $f = \top$ .

Then, if  $M$  are postponable measurement operators with respect to  $\{G_\omega\}_{\omega, \mathcal{L}_\omega \neq \emptyset}$  (Definition 4.9),  $\mathcal{F}_{\text{PartMeas}}^{M,f_0}$  is a ZKoQS protocol (actually ZKoQS<sub>S</sub> for any set  $S$ , see Definition 4.7) for the language  $\mathcal{L}_Q$  previously defined.

*Sketch of proof.* The proof mostly derives from the definitions, and from the fact that having postponable operators allows us to push the part of the ideal functionality that depends on the secret after the interaction with the adversary, preserving the ZK property. We refer to the **full security proof** in [section C](#) for more details.  $\square$

While the above results show that we can obtain a ZKoQS protocol from any protocol realizing the functionality  $\mathcal{F}_{\text{PartMeas}}^{M, f_0}$  (where  $M$  must be postponable), we show in the next section how we can realize such a functionality to prove that a state was partially collapsed (measured in the computational basis) without revealing the position of the collapsed qubit. We will then see that, as a corollary, there exists a ZKoQS protocol for the quantum language of “semi-collapsed” states.

#### 4.4 Protocol to prove that a state has been semi-collapsed

We prove now that we can realize the functionality below, that informally measures a set  $T$  of qubits (the measured qubits, chosen by the prover, being constraint to respect  $\text{Pred}(T) = \top$ , for an arbitrary predicate  $\text{Pred}$ ), randomly rotates the other one, and provides the resulting state to the verifier.

**Definition 4.11 (Semi-collapsing functionality  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$ ).** *Let  $n \in \mathbb{N}$ , and  $\text{Pred}: \mathcal{P}([n]) \rightarrow \{\top, \perp\}$  be an efficiently computable predicate on the subsets of  $[n]$ . We define the semi-collapsing functionality  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$  as  $\mathcal{F}_{\text{PartMeas}}^{M, f_0}$  (Definition 4.8), where:*

- $M$  is the measurement operator that receives a quantum state on two registers, measures (destructively) the first register<sup>26</sup> in the computational basis to get (an encoding of)  $T \subseteq [n]$  and a sequence of bits  $(r^{(i)})_{i \in [n] \setminus T}$ , checks if  $\text{Pred}(T) = \top$ : if not it outputs  $m = \perp$  and a dummy quantum state  $|\perp\rangle$ . Otherwise, it measures (non-destructively) in the computational basis all qubits in the second register whose index belongs to the set of “target” qubits  $T$ , getting outcomes  $\{m^{(j)}\}_{j \in T}$ , and for any  $i \in [n] \setminus T$ , it applies  $Z^{r^{(i)}}$  on the  $i$ -th qubit. Finally it outputs  $m = (T, (m^{(j)})_{j \in T}, (r^{(i)})_{i \in [n] \setminus T})$  and the post-measured state.
- If  $m = \perp$ ,  $f_0(m) = \perp$ , otherwise if  $m = (T, (m^{(j)})_{j \in T}, (r^{(i)})_{i \in [n] \setminus T})$ ,  $f_0(m) = (r^{(i)})_{i \in [n] \setminus T}$ .

We also consider the following dummy ideal parties:

- $\tilde{P}(T, \rho)$  samples<sup>27</sup> uniformly at random a sequence of bits  $(r^{(i)})_{i \in [n] \setminus T}$ , sends  $a = \text{Pred}(T)$  and  $|T, (r^{(i)})_{i \in [n] \setminus T}\rangle \langle T, (r^{(i)})_{i \in [n] \setminus T}| \otimes \rho$  to the ideal functionality  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$ , and forwards the received message from the functionality.
- $\tilde{V}$  checks if it received  $a = \perp$  from the functionality, or if the received quantum state is  $|\perp\rangle$ . If so it sends back  $f = \perp$  to the functionality and aborts, and otherwise it sets  $f = \top$  for the functionality and outputs the quantum state to the environment.

We prove now that we can realize the functionality  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$ :

<sup>26</sup>Informally this register contains the subset of qubits in the second register to measure and a (typically random) sequence of  $Z$  rotations to apply on the remaining qubits. Since the first operation of  $M$  is to measure them, we can (and will) also consider them as classical inputs.

<sup>27</sup>Note that this sequence of rotations is only needed for correctness as in the real protocol the non-measured qubits will be arbitrarily rotated.

**Theorem 4.12 (Realization of  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$ ).** Let  $\{h_k\}_{k \in \mathcal{K}}$  be a family of collision resistant functions sampled by **Gen**, having the hardcore second-bit property (Definition 2.9). Let  $\Pi_h = (P_h, V_h)$  be a protocol<sup>28</sup>  $\text{CS}_{S_h}$ -QSA realizing  $\mathcal{F}_{\text{CRS}}^{\text{Gen}}$  and  $\Pi_{zk} = (\mathbf{A}_{zk}, \mathbf{B}_{zk})$  be a protocol that  $\text{CS}_S$ -QSA realizes the ZK functionality  $\mathcal{F}_{\text{ZK}}^{\text{R}}$ , where  $(h_d^{(c)})_{c \in [n], d \in \{0,1\}} \mathcal{R}(T, (w_d^{(c)})_{c \in [n], d \in \{0,1\}}) \Leftrightarrow \text{Pred}(T) = \top \wedge \forall c, d, h(d \| w_d^{(c)}) = h_d^{(c)}$  and  $\forall c \in T, \exists c$  such that  $w_d^{(c)}[1] = 1$ .

Then, the protocol  $\Pi_{\text{SemCol}}$  (Protocol 2)  $\text{CS}_{S'}$ -QSA-realizes  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$  for any  $S'$  such that:

- $S' \subseteq S \cap S_h$ ,
- $\{P\} \in S'$  only if  $h$  has the statistical hardcore second-bit property,
- $\{V\} \in S'$  only if for any  $k \in \mathcal{K}$ ,  $h_k$  is injective (i.e. statistically collision resistant).

*Sketch of proof.* Part of the proofs of this theorem are generalizations of Theorem 3.1. Some care must be taken to show that the distributions in the honest case (ideal world versus real world) are really indistinguishable, we do so by computing the appropriate density matrices. There is also a slight difference as here we measure the state instead of applying a rotation, but it turns out that measuring is indistinguishable from rotating a state and discarding the rotation angle. We refer to the **full security proof** in section C for more details.  $\square$

We will see that the  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$  functionality can be used to trivially get more advanced OT protocols, notably string OT and  $k$ -out-of- $n$  OT for any  $k$  and  $n$ . But first, we prove that it is a ZKoQS functionality for the quantum language of “semi-collapsed” states with respect to a predicate  $\text{Pred}$ . Informally, we define the quantum language of *semi-collapsed* states as the set of states such that there exists a subset  $T$  of qubits such that  $\text{Pred}(T) = \top$ , and such that all qubits in  $T$  are collapsed, i.e. measured in the computational basis and equal to  $|0\rangle$  or  $|1\rangle$  (therefore not entangled with any other system). Moreover, the identity of the set  $T$  of collapsed qubits stays hidden to a malicious verifier, and in an honest protocol the non-collapsed qubits are either a  $|+\rangle$  or a  $|-\rangle$ , this description being known to the prover.

*Remark 4.13.* Note that the predicate  $\text{Pred}$  might (implicitly<sup>29</sup>) depend on an additional secret classical witness (like a password, a signature provided by some trusted parties, or any NP statement) only known by the prover. This can allow the prover to prove even more advanced statements, like “Either all states are collapsed, or I am the owner of this bitcoin wallet and a single state is collapsed”, which can for instance be useful to obtain “anonymous authorized OT” (i.e. an OT protocol where only parties knowing the witness can participate, while the sender never knows if the receiver knows the witness or not).

**Definition 4.14 (Semi-collapsed states  $\mathcal{L}_{\text{SemCol}}^{\text{Pred}}$ ).** The quantum language  $\mathcal{L}_{\text{SemCol}}^{\text{Pred}}$  of semi-collapsed states relative to a predicate  $\text{Pred}: \mathcal{P}([n]) \rightarrow \{\top, \perp\}$  on the subsets of qubits is composed of the classes  $\mathcal{C} = \mathcal{P}([n])$  (denoting the set of collapsed qubits), the sub-classes  $\mathcal{C}_s = \{s \in \{0,1\}^* \mid |s| \leq n\}$  (denoting the description of the non-collapsed qubits), and the quantum (sub-)classes defined as follows, for any  $T \in \mathcal{C}$  and  $\omega_s \in \mathcal{C}_s$ :

- $\mathcal{L}_{T, \omega_s}$  is the empty set if  $\text{Pred}(T) = \text{false}$  or if  $|\omega_s| \neq |T|$ , and otherwise is the set of all  $n$ -qubits states where qubits in  $T$  are either  $|0\rangle$  or  $|1\rangle$ , and other qubits  $i$  ( $i \in \{1, \dots, |T|\}$  is the index of the qubits in  $[n] \setminus T$ ) are equal to  $|+\rangle$  if  $\omega_s[i] = 0$  and  $|-\rangle$  otherwise.

<sup>28</sup>As a reminder, this protocol is sampling and distributing a function  $h$  according to **Gen**, and can either be done without communication in the CRS model (or heuristically if we replace  $h$  with a well known collision-resistant hash function), or with one message in the plain model.

<sup>29</sup>In which case, the witness must be used to generate the ZK proof.

---

**Protocol 2:** ZKoQS protocol to realize  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$ 


---

 $P(\mathbf{T} \subseteq [n], \rho^{(1), \dots, (n)})$ 
 $V$ 

//  $T$  is a subset of qubits to measure.

Run  $P_h$  to obtain  $h$ .

 $\longleftrightarrow$ 

Run  $V_h$  to obtain  $h$ .

If  $\text{Pred}(T) = \perp$  abort and send  $\perp$  to  $V$ .

$\forall d \in \{0, 1\}, i \in [n] \setminus T, w_d^{(i)} \stackrel{\$}{\leftarrow} \{0\} \times \{0, 1\}^n$

$\forall j \in T, l \stackrel{\$}{\leftarrow}$  Measure non destructively  $\rho^{(j)}$

$w_l^{(j)} \stackrel{\$}{\leftarrow} \{0\} \times \{0, 1\}^n$

$w_{1-l}^{(j)} \stackrel{\$}{\leftarrow} \{1\} \times \{0, 1\}^n$

// Compute the characterization

// of the languages:

$\forall (c, d) \in T \times \{0, 1\}, h_d^{(c)} := h(d \| w_d^{(c)})$

$\pi := (\text{NI})\text{ZK}$  proof that:  $\leftarrow$

$\exists T \subseteq [n], (w_d^{(c)})_{c \in T, d \in \{0, 1\}},$

$\forall c, d, h_d^{(c)} = h(d \| w_d^{(c)})$

and  $\forall j \in T, \exists d$  s.t.  $w_d^{(j)}[1] = 1,$

and  $\text{Pred}(T) = \top.$

$\forall i \in [n] \setminus T, r^{(i)} \stackrel{\$}{\leftarrow} \{0, 1\}$

$\forall i \in [n] \setminus T,$  applies  $Z^{r^{(i)}}$   $\rho^{(i)}$

$\forall c \in [n],$  Apply on  $\rho^{(c)}$ :  $|x\rangle \mapsto |x\rangle |w_x^{(c)}\rangle$

(call  $\rho_1^{(1), \dots, (n)}$  the resulting state)

$\forall (c \in [n], d \in \{0, 1\}) : h_d^{(c)}, \pi, \rho_1^{(1), \dots, (n)}$

$\leftarrow \forall c, s^{(c)}$

$h$  is a collision-resistant (Definition 2.10) and second-bit hardcore (Definition 2.9) function distributed using  $\mathcal{F}_H$  (Definition 2.12), either non-interactively via a CRS, heuristically using a fixed hash function, or sent by the verifier, adding an additional message (Lemma 2.14)

If the ZK proof is interactive, then we actually run the ZK protocol (before sending the quantum state) instead of sending the proof (of course this adds additional rounds of communication).

If Alice sent an abort message  $\perp$ , **return**  $\perp$

Check (or run if interactive proof)  $\pi$ .

$\forall c,$  apply on  $\rho_1^{(c)} \otimes |0\rangle\langle 0|$  the unitary implementing:

$x, w \mapsto w[1] \neq 1 \wedge \exists d, h(x \| w) = h_d^{(c)},$

measure the last (output) register

and check that the outcome is 1.

$\forall c,$  measure the second register of  $\rho_1^{(c)}$

in the Hadamard basis (outcome  $s^{(c)}$ ).

..... End of the send procedure .....

**return** Remaining (first) qubit of each  $\rho_1^{(i)}$ .

Compute:

$\omega_s := (r^{(i)} \oplus \langle s^{(i)}, w_0^{(i)} \oplus w_1^{(i)} \rangle)_{i \in [n] \setminus T}$

**return**  $\omega_s$

---

- $\mathcal{L}_{\text{SemCol}}^{\text{Pred}}$  is the set of bipartite states on registers  $\mathsf{P}$  and  $\mathsf{V}$  such that  $\mathsf{V}$  contains  $n$  qubits, and such that there exists  $T \subseteq [n]$  such that  $\text{Pred}(T) = \perp$  and for any  $i \in T$ ,  $i$ -th qubit of register  $\mathsf{V}$  is not entangled with any other qubit and either  $|0\rangle$  or  $|1\rangle$ .

**Corollary 4.15 (ZKoQS for semi-collapsed states).** *Let  $G'(T)$  be the procedure that samples  $(r^{(i)})_{i \in [n]} \stackrel{\$}{\leftarrow} \{0, 1\}$  and outputs the quantum state  $\bigotimes_i H^{\delta_{i \notin T}} |a^{(i)}\rangle$  (i.e. all qubits in  $T$  are in the computational basis, others are in the Hadamard basis).*

*The functionality  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$  (where the ideal party  $\tilde{\mathsf{P}}$  is slightly updated<sup>30</sup>: instead of receiving  $T$  and  $\rho$ , it receives  $T \subseteq [n]$ , and samples  $\rho \leftarrow G'(T)$ , before continuing as usual) is a ZKoQS ideal functionality (Definition 4.7) for the quantum language  $\mathcal{L}_{\text{SemCol}}^{\text{Pred}}$  (Definition 4.14).*

*In particular, if we consider the protocol where the honest prover gets as input  $T$ , picks  $\rho \leftarrow G'(T)$ , and runs Protocol 2, this protocol is a ZKoQS protocol for the quantum language  $\mathcal{L}_{\text{SemCol}}^{\text{Pred}}$ .*

This is mostly a corollary of Theorem 4.10. The only non-trivial statement is to prove that the measurement is postponable: this can be done by teleporting the state without applying any correction. See proof in section C.

#### 4.5 $\text{ZKstatesQIP}_S[k]$ and $\text{ZKstatesQMA}_S$ : ZKoQS from a complexity theory point of view

While we defined ZKoQS using a “cryptographic” definition, we can also consider them from the point of view of complexity theory. While classically, complexity classes involve a verifier taking an input  $x$  potentially belonging to a given classical language  $\mathcal{L}$ , and outputting a single accept bit (this is not an issue as the input  $x$  can anyway be copied by the verifier if it needs to be used later), for quantum languages this definition turns out to be hard (or even impossible) to use as the verification procedure will alter the input state. ([KA04] does something along that line, but needs to send many copies of the input state, which is of little interest in cryptography as it leads to polynomial security.) To overcome this issue, it is therefore natural to say that the quantum state belonging to the quantum language must be an *output* of the verifier. This is the successful point of view that we took above, and a similar approach has also been used before in [RY22] to quantify the complexity to produce a given state by defining a complexity class  $\text{stateQIP}$ . However, the class  $\text{stateQIP}$  only captures how hard it is to generate a given state, but it does not capture any notion of privacy against a malicious verifier. The following definition addresses this issue:

**Definition 4.16 ( $\text{ZKstatesQIP}_S[k]$  and  $\text{ZKstatesQMA}_S$ ).** *Let  $\mathcal{L}$  be a quantum language (Definition 4.3),  $k \in \mathbb{N}$  be a number of exchanged messages,  $S \in \{\emptyset, \mathsf{P}, \mathsf{V}\}$  be a subset of parties allowed to be unbounded, and  $\text{setup}$  be a given setup assumption (e.g. CRS, Random Oracle, or plain-model). We say that  $\mathcal{L}$  belongs to the complexity class  $\text{ZKstatesQIP}_S^{\text{setup}}[k]$  if there exists a  $\text{ZKoQS}_S$  protocol for  $\mathcal{L}$ , secure assuming the setup assumption  $\text{setup}$ , whose send phase consists of  $k$  exchanged messages (note that we might omit  $S$ ,  $\text{setup}$ , or  $k$  if we do not want to constraint this parameter).*

*Similarly, we define  $\text{ZKstatesQMA}_S^{\text{setup}} = \text{ZKstatesQIP}_S^{\text{setup}}[1]$  to capture non-interactive protocols.*

<sup>30</sup>  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$  can be used for any input quantum state, but for the ZKoQS we need to consider a particular case where the initial state is picked by the party instead of by the environment. The reason is that in ZKoQS protocols, an honest prover is only given as input a class.

*Remark 4.17.* Note that [RY22] defines multiple complexity classes like `stateQIP` (update: similarly, [DGL<sup>+</sup>23], that was uploaded online a day after our own work, defines `stateQMA`): it is therefore natural to want to compare `stateQIP` and `ZKstatesQIP`, similarly to the result `stateQIP = statePSPACE` presented in [MY23]. However, note that since [RY22] and [DGL<sup>+</sup>23] mostly care about the complexity required to create quantum states, there is no notion of hiding or witnesses<sup>31</sup>: expressed with our terminology, their quantum languages have a single element (for a fixed, public,  $n$ )  $|\psi_n\rangle$ . Said differently, the verifier knows in advance the state  $|\psi_n\rangle$  that will be generated with the help of the prover. On the other side, we have no reasons to introduce in `ZKstatesQIP` an asymptotic parameter  $n$  denoting the size of the quantum state obtained by the verifier, as `ZKoQS` already makes sense for a fixed  $n = 2$  (but of course, nothing prevents  $\mathcal{L}_{\mathcal{Q}}$  from containing states of various sizes). However, it might be possible to generalize the definition of [RY22] by replacing  $(|\psi_n\rangle)_{n \in \mathbb{N}}$  with  $\mathcal{L}_{\mathcal{Q}}$ , where the parameter  $n$  could represent the size of the state obtained by the verifier, or to consider a sequence of quantum languages  $(\mathcal{L}_{\mathcal{Q},n})_{n \in \mathbb{N}}$ , in order to define `statesQIP` (the additional `s` being used to denote the fact that the verifier might produce a state among multiple, valid, candidates). However, properly generalizing [RY22] and defining `statesQIP`/`statesQMA` is out of the scope of this paper.

We prove now that  $\mathcal{L}_{\text{SemCol}}^{\text{Pred}}$  belongs to these classes:

**Corollary 4.18** ( $\mathcal{L}_{\text{SemCol}}^{\text{Pred}}$  is in `ZKstatesQMARO`). *For any predicate  $\text{Pred}$ , the quantum language  $\mathcal{L}_{\text{SemCol}}^{\text{Pred}}$  belong to `ZKstatesQMARO` (where `RO` stands for *Random Oracle model*). Moreover, assuming the hardness of `LWE` (see [HSS11] for the exact assumptions),  $\mathcal{L}_{\text{SemCol}}^{\text{Pred}}$  belongs to `ZKstatesQIPpm` (where `pm` stands for *plain-model*).*

*More generally, assuming the existence of a  $k$ -message `ZK` protocol `CSS-QSA` realizing  $\mathcal{F}_{\text{ZK}}$  for any `NP` statement assuming a setup  $\text{setup}$ ,  $\mathcal{L}_{\text{SemCol}}^{\text{Pred}}$  belong to `ZKstatesQIPSsetup[k]`.*

These statements can be proven using Corollary 4.15, together with the constructions of [Unr15] and [HSS11]. See proof in section C.

## 4.6 Applications to build string and $k$ -out-of- $n$ OT protocols

We prove in this section that the above functionality  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$  actually allows us to have string OT or  $k$ -out-of- $n$  OT. But first, we show that we can realize this functionality:

**Theorem 4.19.** *Let  $\text{Pred}$  be a predicate on subsets of  $[n]$ . Assuming the existence of a protocol  $\Pi_{\text{SemCol}} = (\mathbf{A}_{\text{SemCol}}, \mathbf{B}_{\text{SemCol}})$  that `CSS-QSA`-realises  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$ , there Protocol 3 `CSS-QSA`-realises  $\mathcal{F}_{\text{OT}}^{\text{Pred}}$ .*

This is a generalisation of the last part of the proof of Theorem 3.1. See proof in section C.

**Corollary 4.20.** *By choosing appropriate values for  $\text{Pred}$  like in Definition 2.6, the protocol Protocol 3 realizes the string OT functionality  $\mathcal{F}_{\text{OT}}^{\text{str}}$  and the  $k$ -out-of- $n$  OT functionality  $\mathcal{F}_{\text{OT}}^{k-m}$ .*

*Proof.* This is a direct consequence of Theorem 4.19 and of the definition of  $\mathcal{F}_{\text{OT}}^{k-m}$  and  $\mathcal{F}_{\text{OT}}^{\text{str}}$ .  $\square$

<sup>31</sup>Actually, in [DGL<sup>+</sup>23], they do define witnesses but in a different way, as they send the witness directly to the verifier: thus, their notion of witness corresponds rather to the transcript of the proof in our case, and should not be understood as an information that must be hidden to the verifier like our own notion of witness (a.k.a. class).



---

**Protocol 3:** Protocol to compile a ZKoQS protocol  $(\mathbf{A}_{\text{SemCol}}, \mathbf{B}_{\text{SemCol}})$  for the quantum language  $\mathcal{L}_{\text{SemCol}}^{\text{Pred}}$  into a predicate OT protocol.

---

<b>Alice</b> ( $B \subseteq \{0, 1\}^n$ )	<b>Bob</b> ( $(m_1, \dots, m_n) \in \{0, 1\}^n$ )
If $\text{Pred}(B) = \perp$ , abort.	
$\forall i \in [n], r^{(i)} \leftarrow \{0, 1\}$	
$\rho := \otimes_{i \in [n]} H^{\delta_{i \in B}}  r^{(i)}\rangle$	
$(s^{(i)})_{i \in B} \leftarrow \mathbf{A}_{\text{SemCol}}(B, \rho)$	$\longleftrightarrow$
Abort if the previous step aborted.	$\rho \leftarrow \mathbf{B}_{\text{SemCol}}$
	If the previous step aborted, abort.
	$\forall c$ , apply $Z^{m_c}$ on $\rho^{(c)}$ and measure it
	in the Hadamard basis (outcome $z^{(c)}$ ).
<b>return</b> $(r^{(i)} \oplus s^{(i)} \oplus z^{(i)})_{i \in B}$	$\longleftarrow \forall c, z^{(c)}$

---

## 5 Composability of [Unr15]

We show now that the online extractable NIZK protocol from [Unr15] quantum stand-alone realizes the  $\mathcal{F}_{\text{ZK}}^{\mathcal{R}}$  functionality in Definition 2.7, when the RO assumption is made. This is needed to instantiate Corollary 4.20 with a concrete ZK protocol.

The polynomial-time QIM prover  $\mathbf{P}$  and verifier  $\mathbf{V}$  from [Unr15, Fig. 1] have access to two random oracles,  $G$  and  $H$ , which can be queried in superposition by both parties (for simplicity we will just refer to a single oracle  $H$ ). We will denote the polynomial-time two-party protocol by  $\Pi_{\text{zk}}^H = (\mathbf{P}, \mathbf{V})$  to stress the interaction between two machines and the trusted random oracles  $\mathbf{P} \overset{H}{\rightsquigarrow} \mathbf{V}$ . Note that a single message is sent from the prover  $\mathbf{P}$  to the verifier  $\mathbf{V}$ , leading to a so-called non-interactive protocol.

The protocol  $\Pi_{\text{zk}}^H$  is proven to be complete, zero-knowledge and (even simulation-sound) online-extractable. We recall the definitions in Definition D.1 for clarity (note that we assume they also hold against non-uniform adversaries).

In the following we will prove that the protocol  $\Pi_{\text{zk}}^H$  quantum stand-alone realizes the functionality  $\mathcal{F}_{\text{ZK}}^{\mathcal{R}}$ .

**Theorem 5.1.** *Let  $H$  be a random oracle. The non-interactive protocol  $\Pi_{\text{zk}}^H = (\mathbf{P}, \mathbf{V})$  from [Unr15] quantum stand-alone realizes the classical zero-knowledge functionality  $\mathcal{F}_{\text{ZK}}^{\mathcal{R}}$ , where  $x \in \mathcal{L} \Leftrightarrow \exists w, x \mathcal{R} w$ .*

See [proof](#) in [section D](#).

**Corollary 5.2.** *In the random oracle model, assuming the existence of a collision-resistant and second-bit hardcore hash function (which holds if  $h$  is modeled as a random oracle model, see discussion in Theorem 1.1), there exists a protocol realizing the string OT functionality  $\mathcal{F}_{\text{OT}}^{\text{str}}$  and the  $k$ -out-of- $n$  OT functionality  $\mathcal{F}_{\text{OT}}^{k-m}$ .*

*Proof.* This is a direct consequence of Corollary 4.20 and Theorem 5.1, where [Unr15] is used to instantiate the ZK protocol.  $\square$

## 6 Acknowledgment

The authors deeply thank Christian Schaffner for many insightful exchanges, together with Stacey Jeffery, Alex Grilo, Geoffroy Couteau and James Bartusek for precious discussions, and anonymous reviewers for many helpful comments and for pointing a mistake (now corrected) in a proof that generalizes our first result. This work is co-funded by the European Union (ERC, ASC-Q, 101040624) and supported by the Dutch National Growth Fund (NGF), as part of the Quantum Delta NL programme. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

## References

- [ABK<sup>+</sup>22] A. Agarwal, J. Bartusek, D. Khurana, and N. Kumar. A New Framework for Quantum Oblivious Transfer, 2022.
- [AL20] P. Ananth and R. L. La Placa. Secure Quantum Extraction Protocols. In R. Pass and K. Pietrzak, editors, *Theory of Cryptography*, Lecture Notes in Computer Science, pages 123–152, Cham. Springer International Publishing, 2020.
- [AQY22] P. Ananth, L. Qian, and H. Yuen. Cryptography from Pseudorandom Quantum States. In Y. Dodis and T. Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022*, Lecture Notes in Computer Science, pages 208–236, Cham. Springer Nature Switzerland, 2022.
- [BBC<sup>+</sup>92] C. H. Bennett, G. Brassard, C. Crépeau, and M.-H. Skubiszewska. Practical Quantum Oblivious Transfer. In J. Feigenbaum, editor, *Advances in Cryptology – CRYPTO ’91*, Lecture Notes in Computer Science, pages 351–366, Berlin, Heidelberg. Springer, 1992.
- [BCK<sup>+</sup>21] J. Bartusek, A. Coladangelo, D. Khurana, and F. Ma. One-Way Functions Imply Secure Computation in a Quantum World. In T. Malkin and C. Peikert, editors, *Advances in Cryptology – CRYPTO 2021*, Lecture Notes in Computer Science, pages 467–496, Cham. Springer International Publishing, 2021.
- [BD18] Z. Brakerski and N. Döttling. Two-Message Statistically Sender-Private OT from LWE. In A. Beimel and S. Dziembowski, editors, *Theory of Cryptography*. Volume 11240, Lecture Notes in Computer Science, pages 370–390. Springer International Publishing, Cham, 2018.
- [BF10] N. J. Bouman and S. Fehr. Sampling in a Quantum Population, and Applications. In T. Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, Lecture Notes in Computer Science, pages 724–741, Berlin, Heidelberg. Springer, 2010.
- [BKS23] J. Bartusek, D. Khurana, and A. Srinivasan. Secure Computation with Shared EPR Pairs (Or: How to Teleport in Zero-Knowledge), 2023.
- [BS20] N. Bitansky and O. Shmueli. Post-quantum zero knowledge in constant rounds. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, pages 269–279, New York, NY, USA. Association for Computing Machinery, June 22, 2020.
- [CGK21] L. Colisson, F. Grosshans, and E. Kashefi. Non-Destructive Zero-Knowledge Proofs on Quantum States, and Multi-Party Generation of Authorized Hidden GHZ States. April 10, 2021.
- [CGS02] C. Crépeau, D. Gottesman, and A. Smith. Secure multi-party quantum computation. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, STOC ’02, pages 643–652, New York, NY, USA. Association for Computing Machinery, May 19, 2002.
- [CJP<sup>+</sup>21] T. Crette, E. Jeandel, S. Perdrix, and R. Vilmart. Completeness of Graphical Languages for Mixed-State Quantum Mechanics. *ACM Transactions on Quantum Computing*, 2(4):17:1–17:28, December 21, 2021.
- [CK17] B. Coecke and A. Kissinger. *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, Cambridge, 2017.
- [CK88] C. Crépeau and J. Kilian. Achieving oblivious transfer using weakened security assumptions. In *[Proceedings 1988] 29th Annual Symposium on Foundations of Computer Science*. [Proceedings 1988] 29th Annual Symposium on Foundations of Computer Science, pages 42–52, October 1988.
- [DFL<sup>+</sup>09] I. Damgård, S. Fehr, C. Lunemann, L. Salvail, and C. Schaffner. Improving the Security of Quantum Protocols via Commit-and-Open. In S. Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, Lecture Notes in Computer Science, pages 408–427, Berlin, Heidelberg. Springer, 2009.
- [DFP<sup>+</sup>14] V. Dunjko, J. F. Fitzsimons, C. Portmann, and R. Renner. Composable Security of Delegated Quantum Computation. In P. Sarkar and T. Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014*, Lecture Notes in Computer Science, pages 406–425, Berlin, Heidelberg. Springer, 2014.

- [DGJ<sup>+</sup>20] Y. Dulek, A. B. Grilo, S. Jeffery, C. Majenz, and C. Schaffner. Secure Multi-party Quantum Computation with a Dishonest Majority. In A. Canteaut and Y. Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020*, Lecture Notes in Computer Science, pages 729–758, Cham. Springer International Publishing, 2020.
- [DGL<sup>+</sup>23] H. Delavenne, F. L. Gall, Y. Liu, and M. Miyamoto. Quantum Merlin-Arthur proof systems for synthesizing quantum states, March 3, 2023.
- [DNS12] F. Dupuis, J. B. Nielsen, and L. Salvail. Actively Secure Two-Party Evaluation of Any Quantum Operation. In R. Safavi-Naini and R. Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, Lecture Notes in Computer Science, pages 794–811, Berlin, Heidelberg. Springer, 2012.
- [EGL85] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, June 1, 1985.
- [ELE] ELECTRIC COIN COMPANY. Zcash: Privacy-protecting digital currency. Zcash. URL: <https://z.cash/> (visited on 02/10/2023).
- [GL89] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, STOC '89, pages 25–32, New York, NY, USA. Association for Computing Machinery, February 1, 1989.
- [GLS<sup>+</sup>21] A. B. Grilo, H. Lin, F. Song, and V. Vaikuntanathan. Oblivious Transfer Is in MiniQCrypt. In A. Canteaut and F.-X. Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021*, Lecture Notes in Computer Science, pages 531–561, Cham. Springer International Publishing, 2021.
- [GMR85] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, pages 291–304, New York, NY, USA. Association for Computing Machinery, December 1, 1985.
- [GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play ANY mental game. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, pages 218–229, New York, NY, USA. Association for Computing Machinery, January 1, 1987.
- [HSS11] S. Hallgren, A. Smith, and F. Song. Classical Cryptographic Protocols in a Quantum World. In P. Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, Lecture Notes in Computer Science, pages 411–428, Berlin, Heidelberg. Springer, 2011.
- [Imp95] R. Impagliazzo. A personal view of average-case complexity. In *Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference*. Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference, pages 134–147, June 1995.
- [JLS18] Z. Ji, Y.-K. Liu, and F. Song. Pseudorandom Quantum States. In H. Shacham and A. Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, Lecture Notes in Computer Science, pages 126–152, Cham. Springer International Publishing, 2018.
- [KA04] E. Kashefi and C. M. Alves. On the Complexity of Quantum Languages, April 12, 2004.
- [Kil88] J. Kilian. Founding cryptography on oblivious transfer. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 20–31, New York, NY, USA. Association for Computing Machinery, January 1, 1988.
- [KKL<sup>+</sup>23] T. Kapourniotis, E. Kashefi, D. Leichtle, L. Music, and H. Ollivier. Asymmetric Quantum Secure Multi-Party Computation With Weak Clients Against Dishonest Majority, March 15, 2023.
- [KP17] E. Kashefi and A. Pappa. Multiparty Delegated Quantum Computing. *Cryptography*, 1(2):12, 2, September 2017.
- [KZ09] A. Kiayias and H.-S. Zhou. Zero-Knowledge Proofs with Witness Elimination. In S. Jarecki and G. Tsudik, editors, *Public Key Cryptography – PKC 2009*, Lecture Notes in Computer Science, pages 124–138, Berlin, Heidelberg. Springer, 2009.
- [Lin13] Y. Lindell. A Note on Constant-Round Zero-Knowledge Proofs of Knowledge. *Journal of Cryptology*, 26(4):638–654, October 1, 2013.
- [LMS21] A. Lombardi, F. Ma, and N. Spooner. Post-Quantum Zero Knowledge, Revisited (or: How to Do Quantum Rewinding Undetectably), November 23, 2021.

- [Lo97] H.-K. Lo. Insecurity of quantum secure computations. *Physical Review A*, 56(2):1154–1162, August 1, 1997.
- [LT22] P. Laud and R. Talviste. Review of the state of the art in secure multiparty computation. In *Cybernetica As*, 2022.
- [MS94] D. Mayers and L. Salvail. Quantum oblivious transfer is secure against all individual measurements. In *Proceedings Workshop on Physics and Computation. PhysComp '94*. Proceedings Workshop on Physics and Computation. PhysComp '94, pages 69–77, November 1994.
- [MY23] T. Metger and H. Yuen. stateQIP = statePSPACE, January 18, 2023.
- [NC10] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*, December 2010.
- [PS19] C. Peikert and S. Shiehian. Noninteractive Zero Knowledge for NP from (Plain) Learning with Errors. In A. Boldyreva and D. Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, Lecture Notes in Computer Science, pages 89–114, Cham. Springer International Publishing, 2019.
- [PVW08] C. Peikert, V. Vaikuntanathan, and B. Waters. A Framework for Efficient and Composable Oblivious Transfer. In D. Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, Lecture Notes in Computer Science, pages 554–571, Berlin, Heidelberg. Springer, 2008.
- [Qua20] W. Quach. UC-Secure OT from LWE, Revisited. In C. Galdi and V. Kolesnikov, editors, *Security and Cryptography for Networks*, Lecture Notes in Computer Science, pages 192–211, Cham. Springer International Publishing, 2020.
- [Rab05] M. O. Rabin. How To Exchange Secrets with Oblivious Transfer, 2005.
- [RY22] G. Rosenthal and H. Yuen. Interactive proofs for synthesizing quantum states and unitaries. In M. Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPICs*, 112:1–112:4. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
- [SMP22] M. B. Santos, P. Mateus, and A. N. Pinto. Quantum oblivious transfer: a short review. *Entropy*, 24(7):945, July 7, 2022.
- [Unr10] D. Unruh. Universally Composable Quantum Multi-party Computation. In H. Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, Lecture Notes in Computer Science, pages 486–505, Berlin, Heidelberg. Springer, 2010.
- [Unr12] D. Unruh. Quantum Proofs of Knowledge. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, Lecture Notes in Computer Science, pages 135–152, Berlin, Heidelberg. Springer, 2012.
- [Unr15] D. Unruh. Non-Interactive Zero-Knowledge Proofs in the Quantum Random Oracle Model. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, Lecture Notes in Computer Science, pages 755–784, Berlin, Heidelberg. Springer, 2015.
- [Unr16] D. Unruh. Computationally Binding Quantum Commitments. In *Advances in Cryptology – EUROCRYPT 2016*. Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 497–527. Springer, Berlin, Heidelberg, 2016.
- [vdWet20] J. van de Wetering. ZX-calculus for the working quantum computer scientist. December 27, 2020.
- [Wat09] J. Watrous. Zero-Knowledge against Quantum Attacks. *SIAM Journal on Computing*, 39(1):25–58, January 1, 2009.
- [Wie83] S. Wiesner. Conjugate coding. *ACM SIGACT News*, 15(1):78–88, January 1, 1983.
- [Wil17] M. M. Wilde. *From Classical to Quantum Shannon Theory*. 2017.
- [WW06] S. Wolf and J. Wullschleger. Oblivious Transfer Is Symmetric. In S. Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, Lecture Notes in Computer Science, pages 222–232, Berlin, Heidelberg. Springer, 2006.
- [Yao82] A. C. Yao. Protocols for secure computations. In *23rd Annual Symposium on Foundations of Computer Science (Sfcs 1982)*. 23rd Annual Symposium on Foundations of Computer Science (Sfcs 1982), pages 160–164, November 1982.

- [Yao95] A. C.-C. Yao. Security of quantum protocols against coherent measurements. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '95, pages 67–75, New York, NY, USA. Association for Computing Machinery, May 29, 1995.
- [YAV<sup>+</sup>22] V. K. Yadav, N. Andola, S. Verma, and S. Venkatesan. A Survey of Oblivious Transfer Protocol. *ACM Computing Surveys*, 54:211:1–211:37, 10s, September 13, 2022.

# SUPPLEMENTARY MATERIAL

## A Proofs of statements in preliminaries

*Proof of Lemma 2.13.* The proof is trivial: we can split the proof in multiple cases. If no party is corrupted (correctness), then the outputs of both parties are always distributed according to  $\mathcal{F}_{CRS}^{\text{Gen}}$ , so no environment can (even statistically) distinguish between the ideal and real world. If Alice gets corrupted, we define the simulator exactly as the malicious Alice  $\hat{\mathbf{A}}$ , and both worlds are identical (up to the name we give to the different parts of the world) and therefore indistinguishable. If Bob ( $\hat{\mathbf{B}}$ ) is malicious, then we define the simulator as outputting  $c = \top$  to the functionality, and then feeding the received advice and the received  $k$  to  $\hat{\mathbf{B}}$ . Again, the ideal and real worlds are equal, which concludes the proof.  $\square$

*Proof of Lemma 2.14.* The proof is trivial and very similar to Lemma 2.13. The only case where it differs is when Bob is malicious. Then the simulator runs  $\hat{\mathbf{B}}$  and forwards the output to the ideal functionality: the test performed by the functionality is exactly the test performed by Alice in the real world, so both worlds are equal.  $\square$

## B Proofs of security of the bit OT protocol

*Proof of Theorem 3.1.* Let  $\mathcal{A}$  be a static adversary. To prove that the above protocol realizes the above functionality, we will split the proof depending on the parties that the static adversary  $\mathcal{A}$  can corrupt.

**Case 1: correctness (no corrupted party).** If no party is corrupted, we are proving that the protocol is correct. First, we can cut  $\mathbf{A}$  and  $\mathbf{B}$  in four parts: the part that runs  $\mathbf{A}_h$ , the part that generates  $h_c^{(c)}$ , the part that runs the ZK proof, and the rest. Because of the correctness of the protocol distributing  $h$ , and because of the completeness of the ZK protocol, we can indistinguishably replace the first and third parts of  $\mathbf{A}$  and  $\mathbf{B}$  with the ideal dummy parties and the corresponding functionalities. Now, because  $w_{1-l}^{(1-b)}[1] = 1$ , the statement that we prove in the NIZK proof is true, so by the completeness of the NIZK protocol the check succeeds. Then, during the second step we apply the unitary that maps  $|x\rangle |w\rangle |0\rangle$  to  $|x\rangle |w\rangle |1\rangle$  only if  $w[1] \neq 1$  (the witness is valid) and if  $h(x||w)$  appears in the list of hashes, which is true for all terms appearing in  $|\psi^{(b)}\rangle$  and  $|\psi^{(1-b)}\rangle$  by construction, so after this step the two states become:

$$|\psi^{(b)}\rangle \rightsquigarrow |0\rangle |w_0^{(b)}\rangle |1\rangle + (-1)^{r^{(b)}} |1\rangle |w_1^{(b)}\rangle |1\rangle = |\psi^{(b)}\rangle |1\rangle \quad (12)$$

$$|\psi^{(1-b)}\rangle \rightsquigarrow |l\rangle |w_l^{(b)}\rangle |1\rangle = |\psi^{(1-b)}\rangle |1\rangle \quad (13)$$

Because the last register (let us call them  $\mathcal{T}^{(b)}$  and  $\mathcal{T}^{(1-b)}$ ) of each state is not entangled with their respective first two registers, measuring  $\mathcal{T}^{(b)}$  and  $\mathcal{T}^{(1-b)}$  will output 1 in both cases and will not disturb the states on the first two registers. Then, we measure the second register of each state in the Hadamard basis, i.e. we first apply Hadamard gates on all qubits and then we measure in the computational basis. After the Hadamard gates, the state  $|\psi^{(b)}\rangle$  is turned into (omitting the constants):

$$(I \otimes H^{n+1}) |\psi^{(b)}\rangle \quad (14)$$

$$= |0\rangle H^{n+1} |w_0^{(b)}\rangle + (-1)^{r^{(b)}} |1\rangle H^{n+1} |w_1^{(b)}\rangle \quad (15)$$

$$= |0\rangle \sum_{s^{(b)} \in \{0,1\}^{n+1}} (-1)^{\langle s^{(b)}, w_0^{(b)} \rangle} |s^{(b)}\rangle + (-1)^{r^{(b)}} |1\rangle \sum_{s^{(b)} \in \{0,1\}^{n+1}} (-1)^{\langle s^{(b)}, w_1^{(b)} \rangle} |s^{(b)}\rangle \quad (16)$$

$$= |0\rangle + |1\rangle \left( \sum_{s^{(b)} \in \{0,1\}^{n+1}} (-1)^{r^{(b)} \oplus \langle s^{(b)}, w_0^{(b)} \oplus w_1^{(b)} \rangle} |s^{(b)}\rangle \right) \quad (17)$$

where  $\langle a, b \rangle := \oplus_i a[i]b[i]$  is the standard inner product of bit strings. If we measure then an outcome  $s^{(b)}$  and define  $\alpha := r^{(b)} \oplus \langle s^{(b)}, w_0^{(b)} \oplus w_1^{(b)} \rangle$  the above state collapses to:

$$|\psi'^{(b)}\rangle := |0\rangle + (-1)^\alpha |1\rangle \quad (18)$$

(note that the state is in the Hadamard basis.)

For the second state  $|\psi^{(1-b)}\rangle = |l\rangle |w_l^{(1-b)}\rangle$ , the first register is not entangled with the second, so measuring the second register does not disturb the first one. So we end up with the state

$$|\psi'^{(1-b)}\rangle := |l\rangle \quad (19)$$

(i.e. the state is in the computational basis).

In the final step, Bob rotates the two states:

$$Z^{m_b} |\psi'^{(b)}\rangle = |0\rangle + (-1)^{\alpha \oplus m_b} |1\rangle \quad Z^{m_{1-b}} |\psi'^{(1-b)}\rangle = Z^{m_{1-b}} |l\rangle = |l\rangle \quad (20)$$

(note that the rotation does nothing in the second case, hence all information about  $m_{1-b}$  is lost). Then, Bob measures both states in the Hadamard basis: therefore we have  $z^{(b)} = \alpha \oplus m_b$  and  $z^{(1-b)}$  is just a random bit since we measure a qubit in the computational basis. So at the end, Alice outputs  $\alpha \oplus z^{(b)} = m_b$  which ends the proof of correctness.

**Case 2: malicious sender (Bob).** We consider now the case where the adversary  $\mathcal{A} = \hat{\mathbf{B}}$  corrupts the sender Bob. For an intuitive overview of the proof, see Section 1.2, together with the sketch of proof after Theorem 3.1.

First, we notice that since the functionality  $\mathcal{F}_{ZK}$  sends the word  $x$  (in our case the hashes  $h_d^{(c)}$ ) to the verifier, we do not need to send  $x$  another time before (if needed we can assume that the ZK protocol starts by sending  $x$ ). Let  $\hat{\mathbf{B}}$  be an adversary. Without loss of generality<sup>32</sup>, we can decompose  $\hat{\mathbf{B}}$  into  $\hat{\mathbf{B}}_0$  and  $\hat{\mathbf{B}}_1$ , where  $\hat{\mathbf{B}}_0$  is the QIM running during the ZK protocol (so it receives an arbitrary advice  $\sigma$  and interacts with  $\mathbf{A}$  during the ZK protocol), forwarding its final internal state to  $\hat{\mathbf{B}}_1$  that runs the rest of the protocol (in particular  $\hat{\mathbf{B}}_1$  receives the quantum state and is supposed to output some measurement outcomes). For any  $\mathbf{Z}$  and family of states  $\sigma = \{\sigma_\lambda\}_{\lambda \in \mathbb{N}}$ , we define below the following hybrids:

- $\text{World}_0 := \text{REAL}_{\Pi, \hat{\mathbf{B}}, \mathbf{Z}}^\sigma$  is the real world (where Alice runs internally the ZK protocol  $\mathbf{A}_{zk}$  with  $\hat{\mathbf{B}}$ ) as pictured in Fig. 4.
- $\text{World}_1$  is like  $\text{World}_0$  except that the  $\Pi_h$  protocol (in charge of sharing  $h$ ) is replaced by the simulated version as pictured in Fig. 5.
- $\text{World}_2$  is like  $\text{World}_0$  except that the ZK protocol is replaced by the simulated version as pictured in Fig. 6.

<sup>32</sup>If the ZK is non-interactive,  $\hat{\mathbf{B}}_0$  would just store the proof  $\pi$ , the hashes  $(h_{i,j})$  and advice  $\sigma$  and forward it to  $\hat{\mathbf{B}}_1$ .



- World<sub>3</sub> is like World<sub>2</sub> except that we remove  $\mathcal{F}_{ZK}$  and always forward  $(h_d^c)_{c,d}$  to  $\mathbf{S}_{\hat{\mathbf{B}}_0}$  as pictured in Fig. 7.
- World<sub>4</sub> is like World<sub>3</sub> except that we sample  $w_{1-l}^{(1-b)} \stackrel{\$}{\leftarrow} \{0\} \times \{0,1\}^n$  as pictured in Fig. 8.
- World<sub>5</sub> is like World<sub>4</sub> except that we define instead  $|\psi^{(1-b)}\rangle := |0\rangle |w_0^{(1-b)}\rangle + (-1)^{r^{(1-b)}} |1\rangle |w_1^{(1-b)}\rangle$  where  $r^{(1-b)} \stackrel{\$}{\leftarrow} \{0,1\}$  is a random bit as pictured in Fig. 9.
- World<sub>6</sub>, pictured in Fig. 10, is like World<sub>5</sub> except that we reorder some operations and we cut Alice in three parts: a simulator  $\mathbf{S}_{\hat{\mathbf{B}}}$  (the simulator will also absorb  $\mathbf{S}_{\hat{\mathbf{B}}_0}$  and  $\hat{\mathbf{B}}_1$  by simply forwarding the input  $\sigma$  to  $\mathbf{S}_{\hat{\mathbf{B}}_0}$  and the output of  $\hat{\mathbf{B}}_1$  to  $\mathbf{Z}$ ), the ideal functionality  $\mathcal{F}_{OT}$ , and the dummy party  $\tilde{\mathbf{A}}$  that forwards  $b$  to  $\mathcal{F}_{OT}$  and outputs the answer  $m_b$  of  $\mathcal{F}_{OT}$  (in Fig. 10  $\mathcal{F}_{OT}$  and  $\tilde{\mathbf{A}}$  are drawn together for to save space). More precisely, we see that all the messages sent to  $\hat{\mathbf{B}}$  are sampled exactly in the same way, irrespective of the value of  $b$ , so we can push that outside of Alice into the simulator. The only part that still depends on  $b$  is the output message. To avoid this dependency, the simulator will compute the two outputs (when  $b = 0$  and when  $b = 1$ ) and send them to  $\mathcal{F}_{OT}$  that will be in charge of outputting the appropriate value. This way, we see that  $\text{World}_6 = \text{IDEAL}_{\tilde{\mathbf{A}}, \mathbf{S}_{\hat{\mathbf{B}}}, \mathbf{Z}}^{\sigma, \mathcal{F}_{OT}}$ .

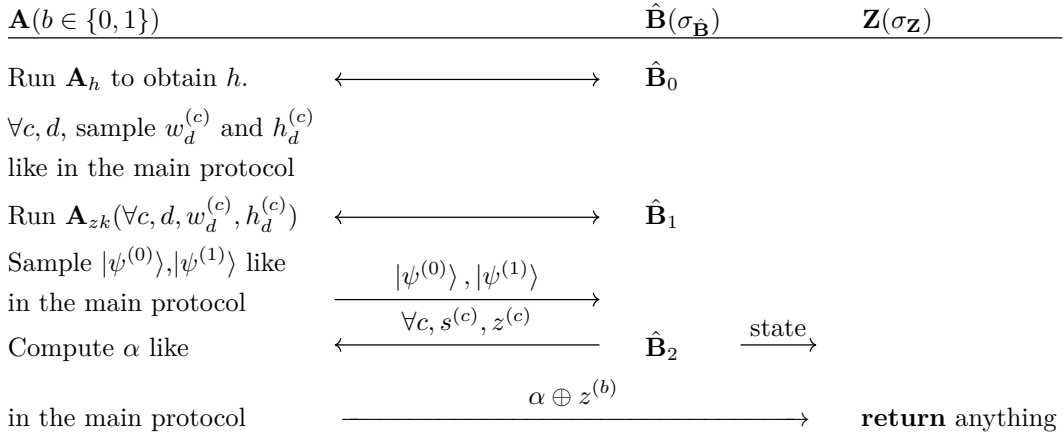


Fig. 4: World<sub>0</sub>

First, we see that  $\text{World}_0 \approx \text{World}_2$  because we assumed that the underlying protocol  $\text{CS}_S\text{-QSA}$  realizes  $\mathcal{F}_{ZK}$ : If it were not the case, then we could easily break the  $\text{CS}_{S'}\text{-QSA}$  property of  $\mathcal{F}_{ZK}$  (and therefore the  $\text{CS}_S\text{-QSA}$  property of  $\mathcal{F}_{ZK}$  since  $S' \subseteq S$ ) by merging the classical sampling procedure inside  $\sigma$  to get a new  $\sigma'$  (the value of the witness being kept as a side-information for  $\mathbf{Z}$  in  $\sigma'$ ) and the rest of the procedure (preparation of the quantum state and running  $\hat{\mathbf{B}}_1$ ) inside  $\mathbf{Z}$  to produce a new  $\mathbf{Z}'$  able to attack  $\mathcal{F}_{ZK}$  with exactly the same probability.

Then,  $\text{World}_2 = \text{World}_3$ : by construction, there always exists a witness starting with a 0, so  $\mathcal{F}_{ZK}$  will always forward  $(h_d^c)_{c,d}$ .

We also have  $\text{World}_3 \approx \text{World}_4$  because of the hardcore second-bit property (see Definition 2.9). Otherwise, we can easily break the hardcore second-bit property by defining  $\mathcal{A}$  as World<sub>3</sub> except that the  $w_{1-l}^{(1-b)}$  is sampled externally by the “challenger” that only provides  $h_{1-l}^{(1-b)}$  to  $\mathcal{A}$  (note that  $w_{1-l}^{(1-b)}$  is not needed here except to compute  $h_{1-l}^{(1-b)}$ ). Note that this attacker is only valid if the computational power of  $\hat{\mathbf{B}}$  is lower than the computational power defined in the hardcore second-bit property, but this is fine if  $\{\mathbf{B}\} \in S'$  only if  $h$  has the statistically hardcore second-bit property as assumed.

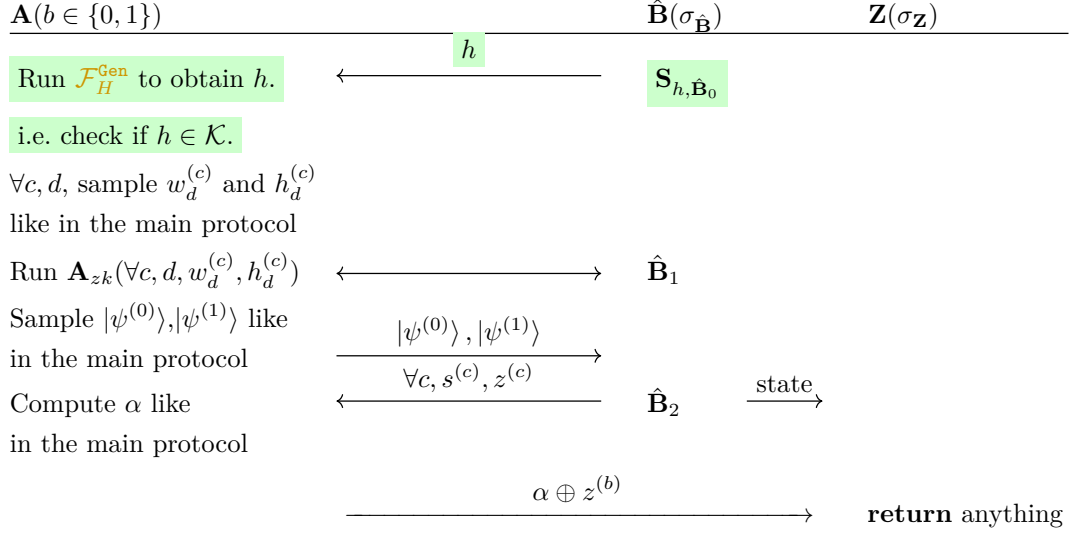


Fig. 5: World<sub>1</sub>

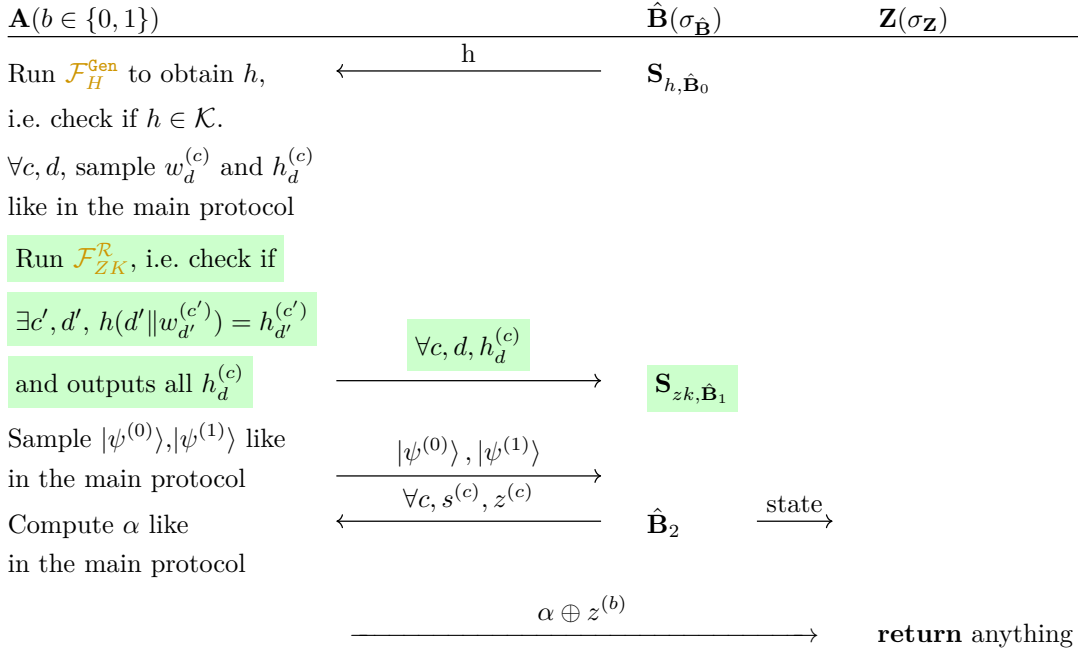


Fig. 6: World<sub>2</sub>

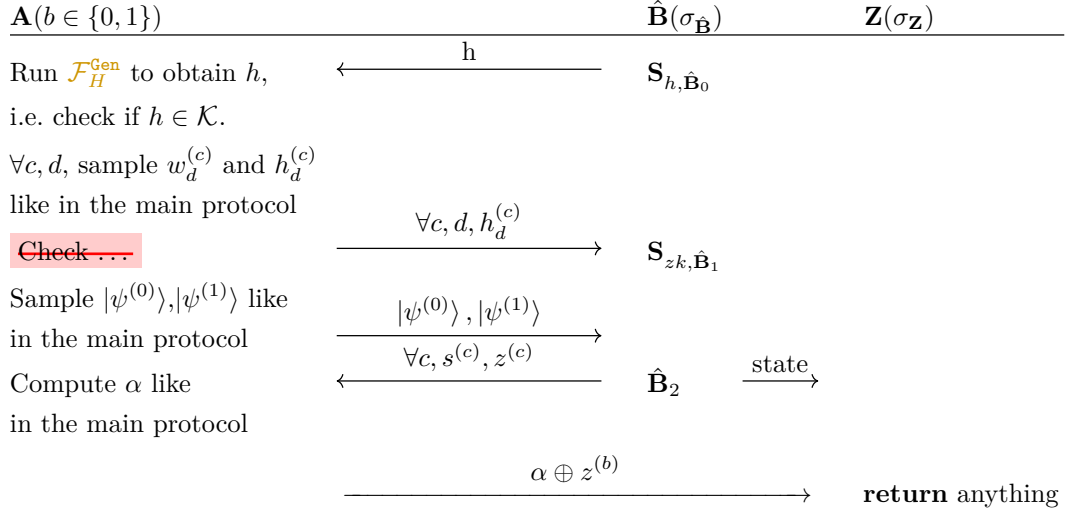


Fig. 7: World<sub>3</sub>

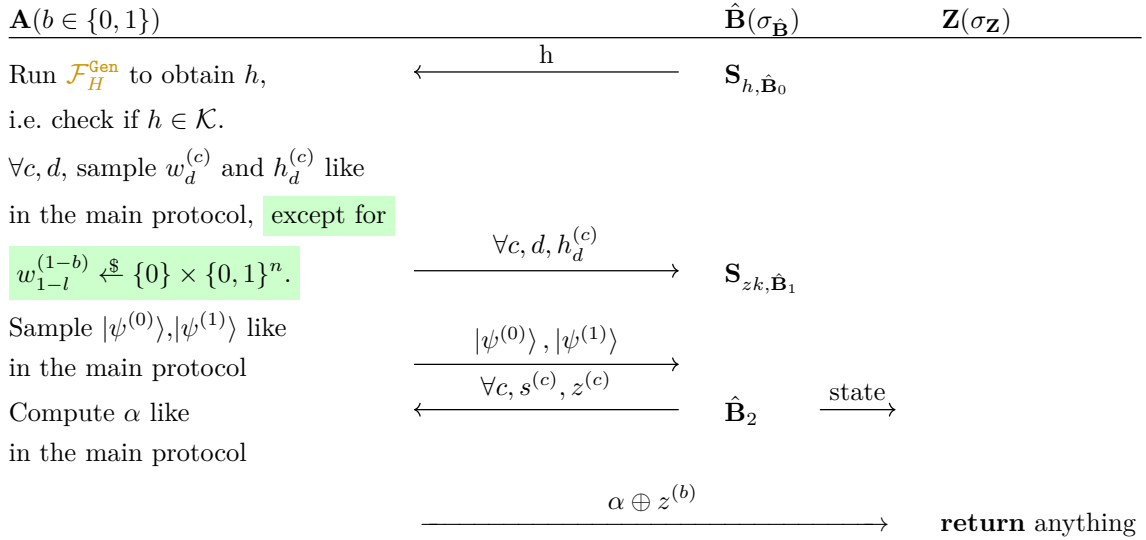


Fig. 8: World<sub>4</sub>

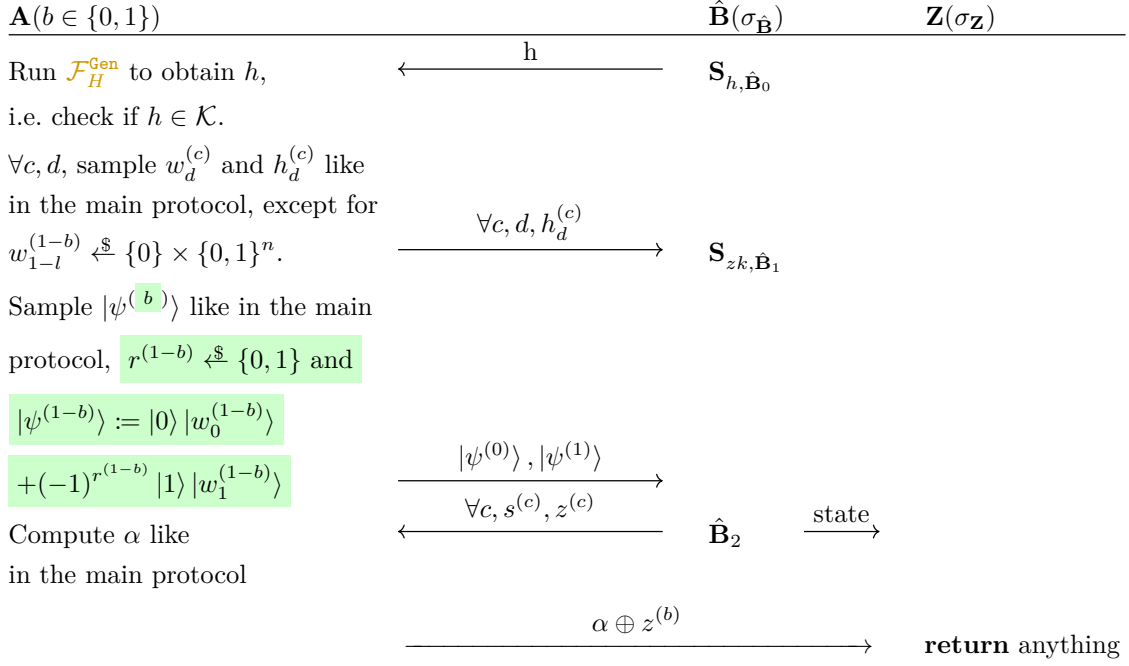


Fig. 9: World<sub>5</sub>

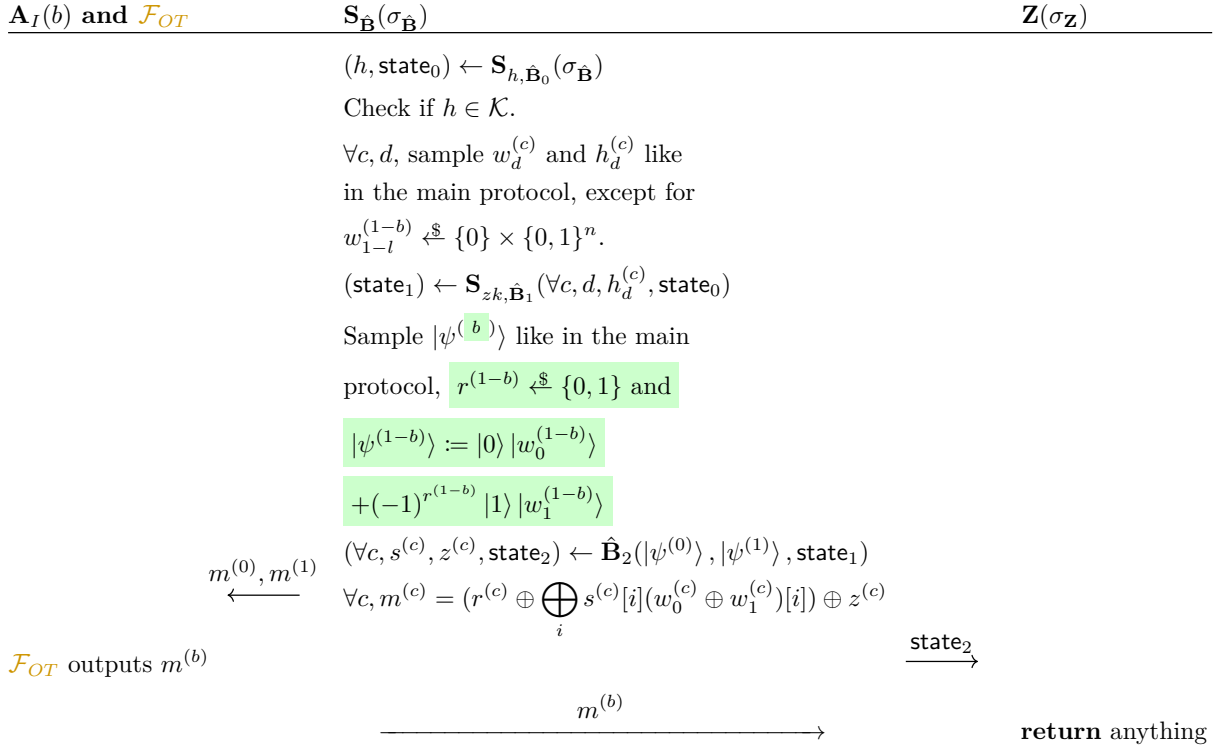


Fig. 10: World<sub>6</sub>

$\text{World}_4 = \text{World}_5$ , because for any  $x$  and  $x'$ , it is statistically impossible to distinguish  $|x\rangle + (-1)^{r^{(1-b)}} |x'\rangle$  from  $|x''\rangle$  where  $x''$  equals  $x$  with probability  $\frac{1}{2}$  and  $x'$  otherwise, even given  $x$  and  $x'$  (the random sign  $r^{(1-b)}$  must however stay hidden<sup>33</sup>). Indeed, we can simply study the density matrices, first of  $|x\rangle \pm |x'\rangle$  (the first sum is on the random choice  $r^{(1-b)}$  of the sign):

$$\frac{1}{2} \left( \frac{|x\rangle + |x'\rangle}{\sqrt{2}} \frac{\langle x| + \langle x'|}{\sqrt{2}} + \frac{|x\rangle - |x'\rangle}{\sqrt{2}} \frac{\langle x| - \langle x'|}{\sqrt{2}} \right) \quad (21)$$

$$= \frac{1}{4} (|x\rangle\langle x| + |x\rangle\langle x'| + |x'\rangle\langle x| + |x'\rangle\langle x'| + |x\rangle\langle x| - |x\rangle\langle x'| - |x'\rangle\langle x| + |x'\rangle\langle x'|) \quad (22)$$

$$= \frac{1}{2} (|x\rangle\langle x| + |x'\rangle\langle x'|) \quad (23)$$

But this last expression is exactly the density matrix of  $|x''\rangle$  where  $x''$  equals  $x$  with probability  $1/2$  and  $x'$  otherwise.

In our case, we have  $x = 0 \| w_0^{(1-b)}$ ,  $x' = 1 \| w_1^{(1-b)}$  and  $x'' = l \| w_l^{(1-b)}$  since  $l$  plays the role of the random coin determining the element to send in  $\text{World}_4$  (you can see that  $l$  plays no role in  $\text{World}_5$  since we can re-order the steps in the sampling procedure). Since  $l$  and  $r^{(1-b)}$  are only used to determine  $|\psi^{(1-b)}\rangle$  and since both worlds are equal beside the choice of  $|\psi^{(1-b)}\rangle$ , we can conclude that  $\text{World}_4 = \text{World}_5$  as otherwise we could “factor out”  $|\psi^{(1-b)}\rangle$  from the worlds and use the remaining part to distinguish between  $|x\rangle + (-1)^{r^{(1-b)}} |x'\rangle$  from  $|x''\rangle$  which is physically impossible as we just saw.

Finally,  $\text{World}_5 = \text{World}_6$  since we just reordered the sampling procedure, delocalized their computation to  $\mathbf{S}_{\hat{\mathbf{B}}}$  and used  $\mathcal{F}_{OT}$  to discard the message corresponding to  $m_{1-b}$  in order to keep only  $m_b$  as in  $\text{World}_5$ .

Therefore, by transitivity,  $\text{World}_0 \approx \text{World}_6$ , i.e.  $\text{REAL}_{\Pi, \hat{\mathbf{B}}, \mathbf{Z}}^\sigma \approx \text{IDEAL}_{\hat{\Pi}, \mathbf{S}_{\hat{\mathbf{B}}}, \mathbf{Z}}^{\sigma, \mathcal{F}_{OT}}$  which concludes this part of the proof.

**Case 3: malicious receiver (Alice).** We consider now the case where the adversary  $\mathcal{A} = \hat{\mathbf{A}}$  corrupts the receiver Alice. For an intuitive overview of the proof, see Section 1.2, together with the sketch of proof after Theorem 3.1.

Like in case 2, without loss of generality we can decompose  $\hat{\mathbf{A}}$  into three parts:  $\hat{\mathbf{A}}_0$  will be the part running the  $\Pi_h$  protocol,  $\hat{\mathbf{A}}_1$  will be the part running the ZK protocol, forwarding its internal state to  $\hat{\mathbf{A}}_2$  that will be in charge of the rest of the protocol as pictured in Fig. 11. In the following, we define, for any bit  $b$ ,  $\mathcal{X}^{(b)}$  as the register containing the state  $|\psi^{(b)}\rangle$ ,  $\mathcal{W}^{(1-b)}$  as the sub-register of  $\mathcal{X}^{(b)}$  containing the witness  $w$  (so all but the first qubit) and  $\mathcal{W}^{(1-b)}[1]$  as the first qubit in the register  $\mathcal{W}^{(1-b)}$ .

We formalize this reasoning by defining the following hybrid worlds:

- $\text{World}_0 := \text{REAL}_{\Pi, \hat{\mathbf{A}}, \mathbf{Z}}^\sigma$  is the real world (where Bob runs internally the ZK protocol  $\mathbf{B}_{zk}$  with  $\hat{\mathbf{A}}$ ) as pictured in Fig. 11.
- $\text{World}_1$  is like  $\text{World}_0$  except that we replace  $\Pi_h$  with the ideal world ( $\hat{\mathbf{A}}_0$  is replaced with  $\mathbf{S}_{\hat{\mathbf{A}}_0}$ , and  $\mathbf{B}_h$  is now replaced with the ideal resource  $\mathcal{F}_H^{\text{Gen}}$  and the idealized party  $\mathbf{B}_{h,I}$  sampling honestly  $h \leftarrow \text{Gen}(1^\lambda)$ ), as pictured in Fig. 12.
- $\text{World}_2$  is like  $\text{World}_1$  except that we replace  $\hat{\mathbf{A}}_1$  with the simulator of the ZK protocol interacting with  $\mathcal{F}_{ZK}$ , and integrate the ideal resource checking if there exists  $w^{1-b}$  starting with a 1 in  $\mathbf{B}$  as pictured in Fig. 13.

<sup>33</sup>It is also possible to remove the sign, but then the proof is harder and only applies statistically if  $h$  is lossy.

- World<sub>3</sub> is like World<sub>2</sub> except that it does not perform the  $Z^{m_1-b}$  rotation, as pictured in Fig. 14.
- For World<sub>4</sub>, we can realize that in World<sub>3</sub>, the code does not depend on  $m_1-b$  anymore. So we can reorganise the elements of World<sub>3</sub> to let the functionality provide  $m_b$  and discard  $m_1-b$ : we split  $\mathbf{B}$  into three parts: the functionality  $\mathcal{F}_{OT}$ , a dummy Bob that just forwards  $m_0$  and  $m_1$  to  $\mathcal{F}_{OT}$ , and a simulator  $\mathbf{S}_{\hat{\mathbf{A}}}$  that runs the code of  $\mathbf{B}$  in the previous World (extracting  $b$  in the same way), except that it sends  $b$  to  $\mathcal{F}_{OT}$  to get  $m_b$ . We also let  $\mathbf{S}_{\hat{\mathbf{A}}}$  absorb  $\hat{\mathbf{A}}, \mathbf{B}_{h,I}, \mathbf{S}_{\hat{\mathbf{A}}_0}$  and  $\mathbf{S}_{\hat{\mathbf{A}}_1}$  (appropriately forwarding their input/outputs): this way  $\text{World}_4 = \text{IDEAL}_{\hat{\mathbf{I}}, \mathbf{S}_{\hat{\mathbf{A}}}, \mathbf{Z}}^{\sigma, \mathcal{F}_{OT}}$ . This is pictured Fig. 15.

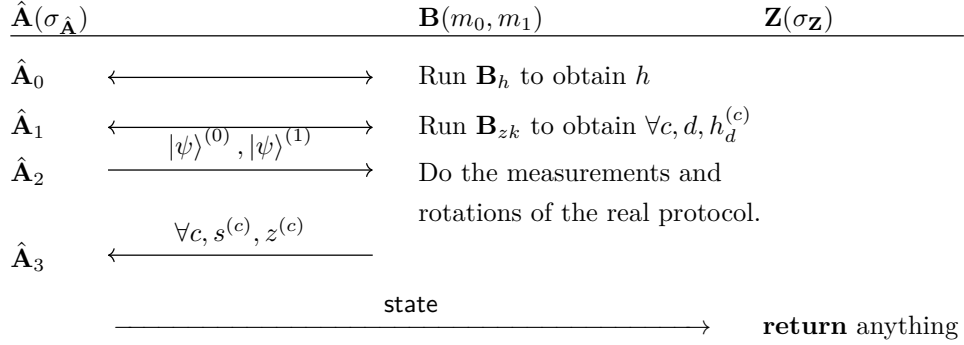


Fig. 11: Case 3 (malicious Alice): World<sub>0</sub>

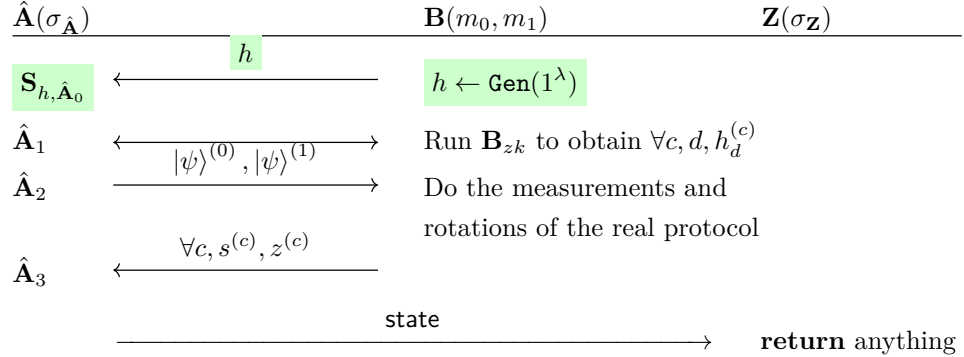


Fig. 12: Case 3 (malicious Alice): World<sub>1</sub>

First, we see that  $\text{World}_0 \approx \text{World}_1$  because we assumed that the  $\Pi_h$  protocol  $\text{CS}_{S_h}$ -QSA realizes  $\mathcal{F}_{ZK}$ : If it were not the case, then we could easily break the  $\text{CS}_{S'}$ -QSA property of  $\Pi_h$  (and therefore the  $\text{CS}_{S_h}$ -QSA property since  $S' \subseteq S_h$ ) by merging basically all the procedure after the  $\Pi_h$  protocol inside  $\mathbf{Z}$  to produce a new  $\mathbf{Z}'$  able to attack  $\mathcal{F}_{ZK}$  with exactly the same probability.

We have also that  $\text{World}_1 \approx \text{World}_2$  because we assumed that the underlying protocol  $\text{CS}_S$ -QSA realizes  $\mathcal{F}_{ZK}$ : If it were not the case, then we could easily break the  $\text{CS}_{S'}$ -QSA property of this underlying protocol (and therefore its  $\text{CS}_S$ -QSA property since  $S' \subseteq S$ ) by merging basically all the procedure after the ZK protocol inside  $\mathbf{Z}$  to produce a new  $\mathbf{Z}'$  able to attack  $\mathcal{F}_{ZK}$  with exactly the same probability.

Then, we prove that  $\text{World}_2 \approx \text{World}_3$ .

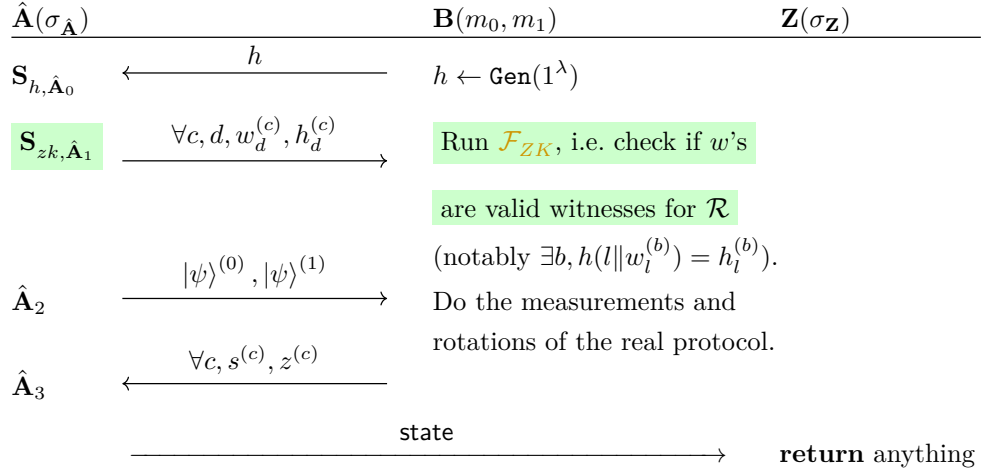


Fig. 13: Case 3 (malicious Alice): World<sub>2</sub>

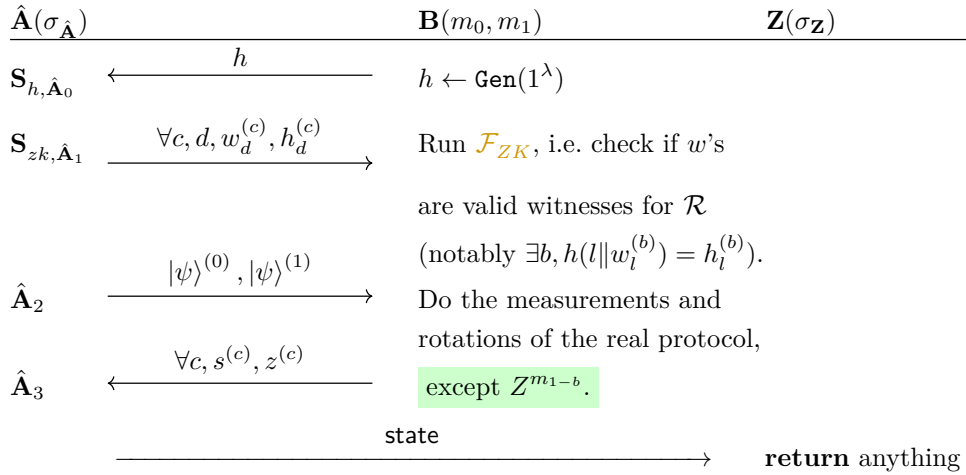


Fig. 14: Case 3 (malicious Alice): World<sub>3</sub>

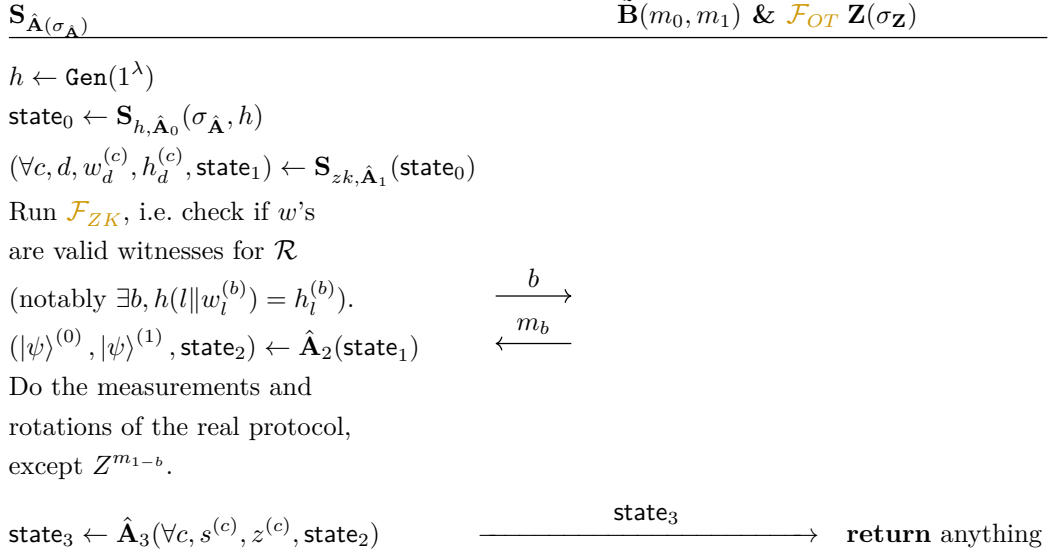


Fig. 15: Case 3 (malicious Alice):  $\text{World}_4$

This part is slightly more technical, but intuitively the goal is to show that the state sent by Alice is close in trace distance to a state in the computational basis (otherwise we can break the collision resistance property of  $h$ ), and therefore a  $Z$  rotation does not significantly disturb the state.

To formalize this intuition, we will first associate a quantity  $\beta$  to each run, show that this  $\beta$  is linked to the probability of finding a collision and to the trace distance to the measured state, and finally we show that the average<sup>34</sup> value of  $\beta$  must be negligible, like that average trace distance between the two worlds. To that aim, it is handy to consider, for a fixed run, a (normalized) purification  $|\psi\rangle_{\mathcal{T}, \mathbf{B}, \mathcal{E}} = |t\rangle \sum_{x,y} \beta_{x,y}^{(1-b)} |x\rangle |y\rangle$  of the states  $|\psi^{(0)}\rangle$  and  $|\psi^{(1)}\rangle$  sent by  $\hat{\mathbf{A}}$  and partially measured<sup>35</sup> by  $\mathbf{B}$  with the result of the tests in the register  $\mathcal{T}$ , including any potential entanglement with the adversary or environment by adding a third register  $\mathcal{E}$  (we also put in this register the internal memory of  $\hat{\mathbf{A}}$  right after she sent the state to  $\mathbf{B}$ ). Moreover, we can assume without loss of generality that  $\hat{\mathbf{A}}$  does nothing before receiving the measurement outcomes sent by Bob, by simply postponing in time its actions.

Then, we can first study the simplest case: if  $h$  is injective (a.k.a. statistically collision resistant), then  $\text{World}_2 = \text{World}_3$ :

First, if  $t = 0$  (invalid test), then the remaining actions in  $\text{World}_2$  and  $\text{World}_3$  are identical. Now, if  $t = 1$ , because  $h$  is injective there exists at most one  $x^{(1-b)}$  such  $x^{(1-b)}$  has a 0 at the second position and  $h(x^{(1-b)}) \in \{h_0^{(1-b)}, h_1^{(1-b)}\}$  (we already extracted above one pre-image of  $h_0^{(1-b)}$  or  $h_1^{(1-b)}$  with a 1 at the second position in the ZK protocol, and by injectivity of  $h$  there are at most two pre-images to this set, so a single image can have a 0 at the second position). Because Bob measured exactly that the first register is a pre-image

<sup>34</sup>We could also, equivalently, prove that the probability of having a non-negligible  $\beta$  is negligible, but this introduces two polynomials, leading to a less elegant proof.

<sup>35</sup>We are referring to the operations where Bob extracts the witness, where it checks that the quantum register containing the witnesses in superposition starts with a 0 and that they are valid pre-images. The result of these tests ( $t = 1$  iff all tests pass) is put in a new register  $\mathcal{T}$ .



of  $\{h_0^{(1-b)}, h_1^{(1-b)}\}$  (since  $t = 1$ ), the state  $|\psi\rangle_{\mathbf{B},E}$  will collapse into  $|x^{(1-b)}\rangle (\sum_y \beta_{x,y}^{(1-b)} |y\rangle)$  (up to a renormalisation factor). Therefore, since  $Z^m |x\rangle = |x\rangle$ , applying  $Z_2^{m_{1-b}}$  or not does not change the state at all (this is true for any run), leading to  $\text{World}_2 = \text{World}_3$ .

Now, we consider the case where  $h$  is only computationally collision-resistant. During a valid extraction of the witness, we found an element  $x_{1-l}^{(1-b)}$  whose witness part starts with a 1 ( $x_{1-l}^{(1-b)}[2] = 1$ ) such that  $h(x_{1-l}^{(1-b)}) \in \{h_0^{(1-b)}, h_1^{(1-b)}\}$ : let  $x^* := x_l^{(1-b)}$  be the other part of the witness such that  $h(x^*) \in \{h_0^{(1-b)}, h_1^{(1-b)}\}$  (if both of them start with a 1, we can choose arbitrarily). Then, there exists  $\beta \in [0, 1]$  (possibly equal to 1 if  $x^*$  starts with a 1), a normalized pure state  $|\phi^*\rangle$ , and a normalized pure state  $|\phi\rangle$  such that  $\text{Tr}(|x^*\rangle\langle x^*| \otimes I) |\phi\rangle\langle\phi| = 0$  such that:

$$|\psi\rangle_{\mathcal{T},\mathbf{B},E} = |t\rangle (\sqrt{1-\beta} |x^*\rangle |\phi^*\rangle + \sqrt{\beta} |\phi\rangle) \quad (24)$$

This can easily be seen by rewriting  $|\psi\rangle_{\mathbf{B},E}$  appropriately:

$$|\psi\rangle_{\mathbf{B},E} = \sum_{x,y} \beta_{x,y}^{(1-b)} |x\rangle |y\rangle \quad (25)$$

$$= |x^*\rangle (\sum_y \beta_{x^*,y}^{(1-b)} |y\rangle) + \sum_{x \neq x^*, y} \beta_{x,y}^{(1-b)} |x\rangle |y\rangle \quad (26)$$

$$= (\sum_y |\beta_{x^*,y}^{(1-b)}|^2) |x^*\rangle \left( \frac{1}{\sum_y |\beta_{x^*,y}^{(1-b)}|^2} \sum_y \beta_{x^*,y}^{(1-b)} |y\rangle \right) + \sum_{x \neq x^*, y} \beta_{x,y}^{(1-b)} |x\rangle |y\rangle \quad (27)$$

Then, because  $|\psi\rangle_{\mathbf{B},E}$  is normalized, we have  $0 \leq \sum_y |\beta_{x^*,y}^{(1-b)}|^2 \leq 1$ , so by defining

$$\beta := 1 - (\sum_y |\beta_{x^*,y}^{(1-b)}|^2)^2 \quad |\phi^*\rangle := \frac{1}{\sum_y |\beta_{x^*,y}^{(1-b)}|^2} \sum_y \beta_{x^*,y}^{(1-b)} |y\rangle$$

$$|\phi\rangle := \frac{1}{\sqrt{\beta}} (|\psi\rangle_{\mathbf{B},E} - \sqrt{1-\beta} |\phi^*\rangle)$$

(if  $\beta = 0$ ,  $|\phi\rangle = 0$ ) we have that  $\beta \in [0, 1]$ ,  $|\phi^*\rangle$  is normalized,  $|\psi\rangle_{\mathbf{B},E} = \sqrt{1-\beta} |x^*\rangle |\phi^*\rangle + \sqrt{\beta} |\phi\rangle$ ,  $\text{Tr}(|x^*\rangle\langle x^*| \otimes I) |\phi\rangle\langle\phi| = 0$  (since  $|\phi\rangle \propto |\psi\rangle_{\mathbf{B},E} - \sqrt{1-\beta} |\phi^*\rangle = \sum_{x \neq x^*, y} \beta_{x,y}^{(1-b)} |x\rangle |y\rangle$ , i.e. a sum on terms  $x \neq x^*$ ), and  $|\phi\rangle$  is also normalized since  $|\psi\rangle_{\mathbf{B},E}$  is also normalized.

We observe now that the probability of finding a collision is greater than  $t\beta$ :

First, if  $t = 0$  (the test fails), then  $t\beta = 0$  so this is obviously true. Now, if  $t = 1$ , this can be seen by first remarking that if we measure the register  $\mathbf{B}$  of  $|\psi\rangle_{\mathbf{B},E}$ , we get an outcome  $x$ : since  $\text{Tr}(|x^*\rangle\langle x^*| \otimes I) |\phi\rangle\langle\phi| = 0$ ,  $x$  equals  $x^*$  with probability  $\sqrt{1-\beta}^2 = 1-\beta$ , i.e.  $x \neq x^*$  with probability  $\beta$ . Moreover, because  $t = 1$ , we know that  $x[2] = 0$  and  $h(x) \in \{h_0^{(1-b)}, h_1^{(1-b)}\}$ . In the first case, if  $h(x) = h_{1-l}^{(1-b)}$ , then  $x \neq x_{1-l}^{(1-b)}$  since  $x[2] = 1$  and  $x_{1-l}^{(1-b)}[2] = 0$  so  $(x, x_{1-l}^{(1-b)})$  is a collision (reminder:  $x_{1-l}^{(1-b)}$  was extracted by the simulator during the ZK protocol). In the second case, if  $h(x) = h_l^{(1-b)}$ , because with probability  $\beta$  we have  $x \neq x^*$ ,  $(x, x^*)$  forms a collision with probability  $\beta$ . Therefore we can find a collision with probability greater than  $\beta = t\beta$ .

Moreover, we observe that the trace distance between  $|\psi\rangle_{\mathcal{T},\mathbf{B},E}$  and  $Z_{\mathbf{B},2}^{tm_{1-b}} |\psi\rangle_{\mathbf{B},E}$  is smaller than  $2t\sqrt{\beta}$ :

First, if  $t = 0$ , both states are strictly equal, so their trace distance is 0. If  $t = 1$ , this can be seen using the triangle inequality (first inequality), the fact that on states in the computational basis,  $Z|x\rangle = |x\rangle$ , Lemma B.1, and the well known fact that the trace distance is preserved under unitary transform ( $\text{TD}(U|\psi\rangle, U|\phi\rangle) = \text{TD}(|\psi\rangle, |\phi\rangle)$ ):

$$\text{TD}(|\psi\rangle_{\mathcal{T}, \mathbf{B}, E}, (Z_{\mathbf{B}, 2}^{tm_1-b} |\psi\rangle_{\mathcal{T}, \mathbf{B}, E})) = \text{TD}(|t\rangle |\psi\rangle_{\mathbf{B}, E}, |t\rangle Z_{\mathbf{B}, 2}^{tm_1-b} |\psi\rangle_{\mathbf{B}, E}) \quad (28)$$

$$\leq \text{TD}(|\psi\rangle_{\mathbf{B}, E}, |x^*\rangle |\phi^*\rangle) + \text{TD}(|x^*\rangle |\phi^*\rangle, Z_2^{m_1-b} |x^*\rangle |\phi^*\rangle) \quad (29)$$

$$+ \text{TD}(Z_2^{m_1-b} |x^*\rangle |\phi^*\rangle, Z_2^{m_1-b} |\psi\rangle_{\mathbf{B}, E}) \quad (30)$$

$$\leq \sqrt{\beta} + 0 + \text{TD}(|x^*\rangle |\phi^*\rangle, |\psi\rangle_{\mathbf{B}, E}) \quad (30)$$

$$\leq 2\sqrt{\beta} = 2t\sqrt{\beta} \quad (31)$$

We prove now that  $\text{World}_2 \approx_{\alpha} \text{World}_3$ , where  $\alpha := \mathbb{E}_{|t\rangle |\psi\rangle_{\mathbf{B}, E} \leftarrow \xi_0(\sigma)} [2t\sqrt{\beta_{\psi}}]$ .

First, we remark that by stopping the worlds right before the  $Z$  rotations, we can define two binary POVM<sup>36</sup>  $\xi_0$  (taking as input (a purification of)  $\sigma_{\lambda}$  and outputting the state  $|\psi\rangle_{\mathcal{T}, \mathbf{B}, E} = |t\rangle |\psi_{\mathbf{B}, E}\rangle$  defined above) and  $\xi_1$  (performing the rest  $Z^{m_b}$  rotation, the measurement in the  $H$  basis, the adversary  $\hat{\mathbf{A}}_2$  and  $\mathbf{Z}$ ) such that  $\text{World}_3$  is the sequential composition of  $\xi_0$  and  $\xi_1$ , and  $\text{World}_2$  is the sequential composition of  $\xi_0$ ,  $Z_2^{tm_1-b}$  and  $\xi_1$ . This way, we can write:

$$|\Pr[\text{World}_3 = 1] - \Pr[\text{World}_2 = 1]| \quad (32)$$

$$= \left| \Pr_{|t\rangle |\psi\rangle_{\mathbf{B}, E} \leftarrow \xi_0(\sigma)} [\xi_1(|t\rangle |\psi\rangle_{\mathbf{B}, E}) = 1] \right. \quad (33)$$

$$\left. - \Pr_{|t\rangle |\psi\rangle_{\mathbf{B}, E} \leftarrow \xi_0(\sigma)} [\xi_1(|t\rangle (Z_{\mathbf{B}, 2}^{tm_1-b} |\psi\rangle_{\mathbf{B}, E})) = 1] \right|$$

$$= \left| \int_{|t\rangle |\psi\rangle_{\mathbf{B}, E}} \Pr[\xi_0(\sigma) = |t\rangle |\psi\rangle_{\mathbf{B}, E}] \left( \Pr[\xi_1(|t\rangle |\psi\rangle_{\mathbf{B}, E}) = 1] \right. \quad (34)$$

$$\left. - \Pr[\xi_1(|t\rangle (Z_{\mathbf{B}, 2}^{tm_1-b} |\psi\rangle_{\mathbf{B}, E})) = 1] \right) \Big|$$

$$\leq \int_{|t\rangle |\psi\rangle_{\mathbf{B}, E}} \Pr[\xi_0(\sigma) = |t\rangle |\psi\rangle_{\mathbf{B}, E}] \left| \Pr[\xi_1(|t\rangle |\psi\rangle_{\mathbf{B}, E}) = 1] \right. \quad (35)$$

$$\left. - \Pr[\xi_1(|t\rangle (Z_{\mathbf{B}, 2}^{tm_1-b} |\psi\rangle_{\mathbf{B}, E})) = 1] \right|$$

$$= \mathbb{E}_{\xi_0(\sigma) = |t\rangle |\psi\rangle_{\mathbf{B}, E}} \left[ \left| \Pr[\xi_1(|t\rangle |\psi\rangle_{\mathbf{B}, E}) = 1] \right. \quad (36)$$

$$\left. - \Pr[\xi_1(|t\rangle (Z_{\mathbf{B}, 2}^{tm_1-b} |\psi\rangle_{\mathbf{B}, E})) = 1] \right| \Big]$$

Moreover, it is well known that for any state  $\rho$  and  $\sigma$ ,  $\text{TD}(\rho, \sigma) = \max_P \text{Tr} P(\rho - \sigma) = \max_P \Pr[P(\rho) = 1] - \Pr[P(\sigma) = 1]$  (see e.g. [NC10, eq. 9.22]), where the maximum is taken over any POVM. Since  $\xi_1$  is a POVM, we have therefore

$$|\Pr[\xi_1(|t\rangle |\psi\rangle_{\mathbf{B}, E}) = 1] - \Pr[\xi_1(|t\rangle (Z_{\mathbf{B}, 2}^{tm_1-b} |\psi\rangle_{\mathbf{B}, E})) = 1]| \quad (37)$$

<sup>36</sup>We slightly abuse notations, as technically a POVM is not a map but a set of projectors (one for each outcome), so we define  $\Pr[\xi_0(\rho) = 1] = \text{Tr}(\xi_0\rho)$ .

$$\leq \text{TD}(|\psi\rangle_{\mathcal{T},\mathbf{B},E}, Z_{\mathbf{B},2}^{tm_1-b} |\psi\rangle_{\mathcal{T},\mathbf{B},E}) \quad (38)$$

$$\stackrel{(31)}{\leq} 2t\sqrt{\beta_\psi} \quad (39)$$

(Note that  $\beta$  is different for any value of  $|\psi\rangle_{\mathcal{T},\mathbf{B},E}$ , hence the notation  $\beta_\psi$ )

By injecting that into Eq. (36), we get

$$|\Pr[\text{World}_3 = 1] - \Pr[\text{World}_2 = 1]| \leq \mathbb{E}_{|t\rangle|\psi\rangle_{\mathbf{B},E} \leftarrow \xi_0(\sigma)} [2t\sqrt{\beta_\psi}] = \alpha \quad (40)$$

Finally, the last step is to prove that  $\alpha$  is negligible, by reducing it to the probability of finding a collision.

We already shown above that given  $|t\rangle|\psi\rangle_{\mathbf{B},E}$ , there is a procedure  $P_h$  to find a collision with probability greater than  $t\beta_\psi$  (but the initial state might depend on  $h$ ). From that, we can define the algorithm that first runs  $\xi_0(\sigma_\lambda)$  ( $\sigma_\lambda$  is now independent of  $h$ , and  $h$  is sampled according to  $\text{Gen}(1^\lambda)$  in  $\xi_0$  as expected), then  $P_h$ . The probability success of this procedure is therefore:

$$\alpha' := \mathbb{E}_{|t\rangle|\psi\rangle_{\mathbf{B},E} \leftarrow \xi_0(\sigma)} [t\beta_\psi] \quad (41)$$

Because  $h$  is collision resistant (Definition 2.10), we have  $\alpha' = \text{negl}(\lambda)$ . Moreover, by defining  $f(x) = \sqrt{2x}$ ,  $f$  is concave, and therefore using Jensen's inequality we get:

$$\alpha = \mathbb{E}_{|t\rangle|\psi\rangle_{\mathbf{B},E} \leftarrow \xi_0(\sigma)} [2t\sqrt{\beta_\psi}] \quad (42)$$

$$= \mathbb{E}_{|t\rangle|\psi\rangle_{\mathbf{B},E} \leftarrow \xi_0(\sigma)} [2\sqrt{t\beta_\psi}] \quad (43)$$

$$\leq 2\sqrt{\mathbb{E}_{|t\rangle|\psi\rangle_{\mathbf{B},E} \leftarrow \xi_0(\sigma)} [t\beta_\psi]} \quad (44)$$

$$= 2\sqrt{\alpha} \quad (45)$$

$$= \text{negl}(\lambda) \quad (46)$$

Therefore, since we know that  $\text{World}_2 \approx_\alpha \text{World}_3$  and we just proved that  $\alpha$  is negligible, we get that  $\text{World}_2 \approx \text{World}_3$ .

Finally, it is easy to see that  $\text{World}_3 = \text{World}_4$  since they are actually exactly the same quantum map, except that we attribute the operations to different parties. By defining  $\mathbf{S}_{\hat{\mathbf{A}},\mathbf{Z}}$  as the block composed of all elements on the left of the ideal resource, we have  $\text{World}_5 = \text{IDEAL}_{\Pi,\mathbf{S}_{\hat{\mathbf{A}}},\mathbf{Z}}^{\sigma,\mathcal{F}_{OT}}$ .

By transitivity, we have  $\text{World}_0 \approx \text{World}_4$ , i.e.  $\text{REAL}_{\Pi,\hat{\mathbf{A}},\mathbf{Z}}^\sigma \approx \text{IDEAL}_{\Pi,\mathbf{S}_{\hat{\mathbf{A}}},\mathbf{Z}}^{\sigma,\mathcal{F}_{OT}}$  which concludes the proof.  $\square$

This small lemma is useful to prove Theorem 3.1:

**Lemma B.1.** *Let  $|\phi\rangle$  and  $|\phi'\rangle$  be two normalized orthogonal pure states and  $\beta \in [0, 1]$ . We consider the normalized state  $|\psi\rangle = \sqrt{1-\beta}|\phi\rangle + \sqrt{\beta}|\phi'\rangle$ . Then,  $\text{TD}(|\phi\rangle, |\psi\rangle) = \sqrt{\beta}$ .*

*Proof of Lemma B.1.* This is a direct characterization of the fact that for any pure states  $|\phi\rangle$  and  $|\psi\rangle$ ,  $\text{TD}(|\phi\rangle, |\phi\rangle) = \sqrt{1 - |\langle\psi|\phi\rangle|^2}$  (see e.g. [Wil17, eq. (9.173)]):

$$\text{TD}(|\phi\rangle, |\psi\rangle) = \sqrt{1 - |\langle\phi|\psi\rangle|^2} = \sqrt{1 - |\langle\phi|(\sqrt{1-\beta}|\phi\rangle + \sqrt{\beta}|\phi'\rangle)|^2} \quad (47)$$

$$= \sqrt{1 - |\sqrt{1-\beta}|^2} = \sqrt{\beta} \quad (48)$$

□

## C Proof of the ZKoQS and $k$ -out-of- $n$ string OT protocols

*Proof of Theorem 4.10.* Let  $S$  be any subset of parties, and let us assume the above assumptions. Let  $\Pi = (\text{P}, \text{V})$  be a protocol that  $\text{CS}_S$ -QSA-realizes  $\mathcal{F}_{\text{PartMeas}}^{M, f_0}$  with the above dummy ideal parties. We prove that  $\Pi$  is a ZKoQS $_S$  protocol, by proving first the completeness.

First, since  $\Pi$   $\text{CS}_S$ -QSA-realizes  $\mathcal{F}_{\text{PartMeas}}^{M, f_0}$ , by defining  $\mathbf{S}(\omega) := \tilde{\text{P}}(\omega) \overset{\mathcal{F}_{\text{PartMeas}}^{M, f_0}}{\rightsquigarrow} \tilde{\text{V}}$ , we have in particular  $(\text{P} \rightsquigarrow \text{V}) \approx_c \mathbf{S}$  and  $\mathbf{S}$  runs in poly-time since  $M$  is efficiently implementable. Let  $\omega$  such that  $\mathcal{L}_\omega \neq \emptyset$ . Because  $\mathcal{L}_\omega \neq \emptyset$ ,  $\tilde{\text{P}}(\omega)$  produces, by definition, a state  $\rho_0^{\text{V}_0, \text{P}}$  in  $E_\omega$ , and since  $\tilde{\text{V}}$  sends  $f = \top$ ,  $\tilde{\text{P}}$  outputs  $\omega_s = f_0(m)$ , where  $m$  is the measurement outcome of the ideal functionality. Also, by definition of  $\mathcal{F}_{\text{PartMeas}}^{M, f_0}$ , the post-measured state is  $\rho^{\text{P}, \text{V}} = \xi_m(\rho_0^{\text{V}_0, \text{P}})$ . Therefore, by definition of  $\mathcal{L}_{\omega, \omega_s}$ , the output state  $\rho^{\text{P}, \text{V}}$  belongs to  $\mathcal{L}_{\omega, \omega_s}$  with probability 1, and  $\tilde{\text{V}}$  always outputs  $a = 1$ . Therefore, Eq. (3) is true, completing the proof of completeness.

We proceed now with the soundness:

Let  $\hat{\text{P}} = \{\hat{\text{P}}_\lambda\}_{\lambda \in \mathbb{N}}$  be a malicious prover (unbounded if  $\text{P} \in S$ , in which case all symbols  $\approx_c$  should be replaced with  $\approx_s$  in the rest of this proof). Because  $\Pi$   $\text{CS}_S$ -QSA-realizes  $\mathcal{F}_{\text{PartMeas}}^{M, f_0}$ , there exists  $\mathbf{S}'_{\hat{\text{P}}}$  such that  $(\hat{\text{P}} \rightsquigarrow \text{V}) \approx_c \mathbf{S}'_{\hat{\text{P}}} \overset{\mathcal{F}_{\text{PartMeas}}^{M, f_0}}{\rightsquigarrow} \tilde{\text{V}}$ . So let us define  $\mathbf{S}_{\hat{\text{P}}}(\sigma_\lambda^{\text{P}}) = (\mathbf{S}'_{\hat{\text{P}}}(\sigma_\lambda^{\text{P}}) \overset{\mathcal{F}_{\text{PartMeas}}^{M, f_0}}{\rightsquigarrow} \tilde{\text{V}})$ : we then have  $(\hat{\text{P}} \rightsquigarrow \text{V}) \approx_c \mathbf{S}_{\hat{\text{P}}}$ . Moreover, since  $\mathcal{L}_Q$  corresponds to all the bipartite states that can be obtained when applying  $\xi_m$  for some  $m$  on the first register, and an arbitrary deviation depending on  $f_0(m)$  on the other register (remember that  $\mathcal{F}_{\text{PartMeas}}^{M, f_0}$  gives back  $f_0(m)$  to the adversary, so  $\hat{\xi}_{f_0(m)}$  would be defined as the second part of  $\mathbf{S}_{\hat{\text{P}}}$ , after waiting for the answer of the functionality), Eq. (4) is true (even with probability 1), completing the proof.

We finish now with the quantum ZK property:

Let  $\hat{\text{V}} = \{\hat{\text{V}}_\lambda\}_{\lambda \in \mathbb{N}}$  be a malicious verifier (unbounded if  $\text{V} \in S$ , in which case all symbols  $\approx_c$  should be replaced with  $\approx_s$  in the rest of this proof). Because  $\Pi$   $\text{CS}_S$ -QSA-realizes  $\mathcal{F}_{\text{PartMeas}}^{M, f_0}$ , there exists  $\mathbf{S}'_{\hat{\text{V}}}$  such that  $(\text{P} \rightsquigarrow \hat{\text{V}}) \approx_c (\tilde{\text{P}} \overset{\mathcal{F}_{\text{PartMeas}}^{M, f_0}}{\rightsquigarrow} \mathbf{S}'_{\hat{\text{V}}})$ . Then, we define  $\mathbf{S}_{\hat{\text{V}}}(b)$  as follows. First, if  $b = \perp$ , it forwards the input of the environment to  $\mathbf{S}'_{\hat{\text{V}}}$  together with  $\perp$  (pretending to be the functionality  $\mathcal{F}_{\text{PartMeas}}^{M, f_0}$ ). Since  $\tilde{\text{P}}(\omega)$  aborts iff  $\mathcal{L}_\omega = \emptyset$ , we actually have for any  $\omega$  such that  $\mathcal{L}_\omega = \emptyset$ ,  $\tilde{\text{P}}(\omega) \overset{\mathcal{F}_{\text{PartMeas}}^{M, f_0}}{\rightsquigarrow} \mathbf{S}'_{\hat{\text{V}}} = (\xi_\omega \otimes I)(\mathbf{S}_{\hat{\text{V}}}(\mathcal{L}_\omega \neq \emptyset))$  where  $\xi_\omega$  is given  $b$  as input and outputs  $\perp$  if  $b = \perp$ : by transitivity, Eq. (5) is true in that case. Otherwise, if  $b = \top$ , since  $M$  are postponable measurement operators with respect to  $G$ , there exists by definition  $\rho^{\text{V}, F}$  and a quantum map  $M'$  such that  $MG \approx_s (I^{\text{V}} \otimes M')(\rho^{\text{V}, F} \otimes G)$ . Therefore,  $\tilde{\text{P}} \overset{\mathcal{F}_{\text{PartMeas}}^{M, f_0}}{\rightsquigarrow} \mathbf{S}'_{\hat{\text{V}}}$  is equivalent to:

1. Generate  $\rho^{V,F}$  and send  $\rho^V$  to  $\mathbf{S}'_{\hat{V}}$  (pretending to be  $\mathcal{F}_{\text{PartMeas}}^{M,f_0}$ ) to get a function  $f$  and an output state  $\rho'$ .
2. Run  $M'(\rho^F \otimes G(\omega))$  to get outcome  $m'$ , send back  $f_\omega(m')$  to  $\tilde{\mathbf{P}}(\omega)$  and outputs the final state of  $\tilde{\mathbf{P}}(\omega)$ .

However, if we define  $\mathbf{S}_{\hat{V}}$  to do the first step, and  $\xi_\omega$  to be the second and third step (when  $b = 1$ ), we have  $(\tilde{\mathbf{P}} \xrightarrow{\mathcal{F}_{\text{PartMeas}}^{M,f_0}} \mathbf{S}'_{\hat{V}}) = (\xi_\omega \otimes I)\mathbf{S}_{\hat{V}}$ , therefore Eq. (5) is respected, concluding the proof. □

*Proof of Theorem 4.12. Case 1: correctness (no corrupted party).*

We first prove the correctness (when all parties are honest):

First, if  $\text{Pred}(T) = \perp$ , the parties abort in both the ideal and real worlds. Otherwise, like in Theorem 3.1, the ZK proof in the real world succeeds by the completeness of the ZK protocol, and the first measurement of Bob will not disturb the state, so we can remove indistinguishably the ZK proof and the first measurement of Bob (we should technically define new worlds as we did before, but we omit them for conciseness as we already applied similar arguments earlier). Similarly, since for any  $j \in T$ , the first register of the state  $\rho^{(j)}$  is already measured, the Hadamard basis measurement on its second register does not alter the state, so we can remove indistinguishably all operations involving  $\rho^{(j)}$ 's except for the measurements in the computational basis by  $\mathbf{P}$ . Therefore, we can concentrate now on  $\rho^{(i)}$  for  $i \in [n] \setminus T$ , and show that the random rotation  $Z^{r^{(i)}}$  followed by the map  $x \rightarrow x, w_x^{(i)}$ , the Hadamard measurement on the second register, and the update of  $r^{(i)}$  is statistically indistinguishable from a single random  $Z^{r^{(i)}}$ .

Let  $i \in [n] \setminus T$ . We can show this property in two ways: either by doing the computation directly on density matrices (which is a bit long, but we do it below for completeness), or remark that the map and Hadamard measurement commute with the  $Z^{r^{(i)}}$ , and that the map and Hadamard measurement performs a  $Z^{\langle s^{(b)}, w_0^{(b)} \rangle}$  rotation: therefore we can sample  $r^{(i)}$  after knowing the outcome of the measurement, and since the distribution  $r^{(i)}$  is indistinguishable from the distribution  $c \oplus r^{(i)}$  for any constant  $c$ , we can take  $c = \langle s^{(b)}, w_0^{(b)} \rangle$  to cancel the rotation applied earlier by the Hadamard, making it virtually equal to a single rotation  $Z^{r^{(i)}}$ .

We provide now an **alternative proof**, more verbose but certainly more formal, by directly computing the appropriate density matrices. If we consider a purification<sup>37</sup> of  $\rho^{(i)}$ , there exists two vectors  $|\psi_0\rangle$  and  $|\psi_1\rangle$  such that  $\rho^{(i)} = \alpha_0 |0\rangle |\psi_0\rangle + \alpha_1 |1\rangle |\psi_1\rangle$ . Moreover, if we consider the density operator of this whole process, we have after the sampling of  $a$  (put on the first register) and  $Z^a$  rotation:

$$\begin{aligned} & \frac{1}{2}(|0\rangle (\alpha_0 |0\rangle |\psi_0\rangle + \alpha_1 |1\rangle |\psi_1\rangle))((\alpha_0^* \langle \psi_0| \langle 0| + \alpha_1^* \langle \psi_1| \langle 1|) \langle 0|) \\ & + \frac{1}{2}(|1\rangle (\alpha_0 |0\rangle |\psi_0\rangle - \alpha_1 |1\rangle |\psi_1\rangle))((\alpha_0^* \langle \psi_0| \langle 0| - \alpha_1^* \langle \psi_1| \langle 1|) \langle 1|) \end{aligned} \quad (49)$$

<sup>37</sup>Technically  $\rho^{(i)}$  could be entangled with other  $\rho^{(i')}$ , so this purification might contain elements in  $\rho^{(i')}$ . This is not an issue as soon as we apply this transformation sequentially, on a single  $i$  at a time.

Then, after the  $x \mapsto x, w_x^{(c)}$  operation, and the Hadamard (omitting the normalisation factor), we get:

$$\begin{aligned}
& \sum_{s^b} \sum_{s^{(i)}} \left( \frac{1}{2} (|0\rangle (\alpha_0 (-1)^{\langle s^{(i)}, w_0^{(i)} \rangle} |0\rangle |\psi_0\rangle |s^{(i)}\rangle + \alpha_1 (-1)^{\langle s^{(i)}, w_1^{(i)} \rangle} |1\rangle |\psi_1\rangle |s^{(i)}\rangle) \right) \\
& ((\alpha_0^* (-1)^{\langle s'^{(i)}, w_0^{(i)} \rangle} \langle s'^{(i)} | \langle \psi_0 | \langle 0 | + \alpha_1^* (-1)^{\langle s'^{(i)}, w_1^{(i)} \rangle} \langle s'^{(i)} | \langle \psi_1 | \langle 1 | \rangle \langle 0 |) \\
& + \frac{1}{2} (|1\rangle (\alpha_0 (-1)^{\langle s^{(i)}, w_0^{(i)} \rangle} |0\rangle |\psi_0\rangle |s^{(i)}\rangle - \alpha_1 (-1)^{\langle s^{(i)}, w_1^{(i)} \rangle} |1\rangle |\psi_1\rangle |s^{(i)}\rangle)) \\
& ((\alpha_0^* (-1)^{\langle s'^{(i)}, w_0^{(i)} \rangle} \langle s'^{(i)} | \langle \psi_0 | \langle 0 | - \alpha_1^* (-1)^{\langle s'^{(i)}, w_1^{(i)} \rangle} \langle s'^{(i)} | \langle \psi_1 | \langle 1 | \rangle \langle 1 |))
\end{aligned} \tag{50}$$

However, since the output of P XOR to  $b$  the value  $\langle s^{(i)}, w_0^{(i)} \oplus w_1^{(i)} \rangle$ , the final density matrix representing this process is as follows (to obtain this, we factored out the  $(-1)^{\langle s^{(i)}, w_0^{(i)} \rangle}$  that gets canceled as a global phase, we rename  $\alpha_{s^{(i)}} := \langle s^{(i)}, w_0^{(i)} \oplus w_1^{(i)} \rangle$ , and we XOR the first register with  $\alpha_{s^{(i)}}$ ):

$$\begin{aligned}
& \frac{1}{2} \sum_{s^b} \sum_{s^{(i)}} ((|\alpha_{s^{(i)}}\rangle (\alpha_0 |0\rangle |\psi_0\rangle |s^{(i)}\rangle + \alpha_1 (-1)^{\alpha_{s^{(i)}}} |1\rangle |\psi_1\rangle |s^{(i)}\rangle)) \\
& ((\alpha_0^* \langle s'^{(i)} | \langle \psi_0 | \langle 0 | + \alpha_1^* (-1)^{\alpha_{s^{(i)}}} \langle s'^{(i)} | \langle \psi_1 | \langle 1 | \rangle \langle \alpha_{s^{(i)}} |) \\
& + (|1 \oplus \alpha_{s^{(i)}}\rangle (\alpha_0 |0\rangle |\psi_0\rangle |s^{(i)}\rangle - \alpha_1 (-1)^{\alpha_{s^{(i)}}} |1\rangle |\psi_1\rangle |s^{(i)}\rangle)) \\
& ((\alpha_0^* \langle s'^{(i)} | \langle \psi_0 | \langle 0 | - \alpha_1^* (-1)^{\alpha_{s^{(i)}}} \langle s'^{(i)} | \langle \psi_1 | \langle 1 | \rangle \langle 1 \oplus \alpha_{s^{(i)}} |))
\end{aligned} \tag{51}$$

Moreover, the last register  $s^{(i)}$  is discarded, so we can trace is out:

$$\begin{aligned}
& \frac{1}{2} \sum_{s^b} ((|\alpha_{s^{(i)}}\rangle (\alpha_0 |0\rangle |\psi_0\rangle + \alpha_1 (-1)^{\alpha_{s^{(i)}}} |1\rangle |\psi_1\rangle)) \\
& ((\alpha_0^* \langle \psi_0 | \langle 0 | + \alpha_1^* (-1)^{\alpha_{s^{(i)}}} \langle \psi_1 | \langle 1 | \rangle \langle \alpha_{s^{(i)}} |) \\
& + (|1 \oplus \alpha_{s^{(i)}}\rangle (\alpha_0 |0\rangle |\psi_0\rangle - \alpha_1 (-1)^{\alpha_{s^{(i)}}} |1\rangle |\psi_1\rangle)) \\
& ((\alpha_0^* \langle \psi_0 | \langle 0 | - \alpha_1^* (-1)^{\alpha_{s^{(i)}}} \langle \psi_1 | \langle 1 | \rangle \langle 1 \oplus \alpha_{s^{(i)}} |))
\end{aligned} \tag{52}$$

We have now two cases: if  $w_0^{(i)} \oplus w_1^{(i)} = 0 \dots 0$ , then  $\alpha_{s^{(i)}} = 0$  and we actually see that the second register is not even entangled with the first one, so the Hadamard measurement does not change the first qubit, therefore we only apply a random  $Z^a$  on it and output  $a$ , like in the ideal world. Now, if  $w_0^{(i)} \oplus w_1^{(i)} \neq 0$ , then the the number of  $s^{(i)}$  such that  $\alpha_{s^{(i)}} = 0$  is exactly equal to the number of cases where  $\alpha_{s^{(i)}} = 1$ :

To see that, since  $w_0^{(i)} \oplus w_1^{(i)} \neq 1$ , there exists one position where they differ: then, just flipping the bit of  $s^{(i)}$  at that position will also flip the value of  $\alpha_{s^{(i)}}$ , providing a simple way to partition  $s^{(i)}$ 's in two sets of equal size, each set having the same value of  $\alpha_{s^{(i)}}$ .

Therefore, we can sum over  $\alpha_{s^{(i)}}$  instead of  $s^b$  (this adds a fixed constant (thanks to the argument we just mentioned) factor that we ignore for simplicity):

$$\begin{aligned}
& \frac{1}{2} \sum_{\alpha_{s^{(i)}} \in \{0,1\}} ((|\alpha_{s^{(i)}}\rangle (\alpha_0 |0\rangle |\psi_0\rangle + \alpha_1 (-1)^{\alpha_{s^{(i)}}} |1\rangle |\psi_1\rangle)) \\
& ((\alpha_0^* \langle \psi_0 | \langle 0 | + \alpha_1^* (-1)^{\alpha_{s^{(i)}}} \langle \psi_1 | \langle 1 | \rangle \langle \alpha_{s^{(i)}} |) \\
& + (|1 \oplus \alpha_{s^{(i)}}\rangle (\alpha_0 |0\rangle |\psi_0\rangle - \alpha_1 (-1)^{\alpha_{s^{(i)}}} |1\rangle |\psi_1\rangle)) \\
& ((\alpha_0^* \langle \psi_0 | \langle 0 | - \alpha_1^* (-1)^{\alpha_{s^{(i)}}} \langle \psi_1 | \langle 1 | \rangle \langle 1 \oplus \alpha_{s^{(i)}} |))
\end{aligned} \tag{53}$$

$$\begin{aligned}
&= \frac{1}{2}((|0\rangle (\alpha_0 |0\rangle |\psi_0\rangle + \alpha_1 |1\rangle |\psi_1\rangle)) \\
&\quad ((\alpha_0^* \langle \psi_0 | \langle 0 | + \alpha_1^* \langle \psi_1 | \langle 1 |) \langle 0 | \\
&\quad + (|1\rangle (\alpha_0 |0\rangle |\psi_0\rangle - \alpha_1 |1\rangle |\psi_1\rangle)) \\
&\quad ((\alpha_0^* \langle \psi_0 | \langle 0 | - \alpha_1^* \langle \psi_1 | \langle 1 |) \langle 1 | \\
&\quad + (|1\rangle (\alpha_0 |0\rangle |\psi_0\rangle - \alpha_1 |1\rangle |\psi_1\rangle)) \\
&\quad ((\alpha_0^* \langle \psi_0 | \langle 0 | - \alpha_1^* \langle \psi_1 | \langle 1 |) \langle 1 | \\
&\quad + (|0\rangle (\alpha_0 |0\rangle |\psi_0\rangle + \alpha_1 |1\rangle |\psi_1\rangle)) \\
&\quad ((\alpha_0^* \langle \psi_0 | \langle 0 | + \alpha_1^* \langle \psi_1 | \langle 1 |) \langle 0 |))
\end{aligned} \tag{54}$$

$$\begin{aligned}
&= \frac{1}{2}((|0\rangle (\alpha_0 |0\rangle |\psi_0\rangle + \alpha_1 |1\rangle |\psi_1\rangle)) \\
&\quad ((\alpha_0^* \langle \psi_0 | \langle 0 | + \alpha_1^* \langle \psi_1 | \langle 1 |) \langle 0 | \\
&\quad + (|1\rangle (\alpha_0 |0\rangle |\psi_0\rangle - \alpha_1 |1\rangle |\psi_1\rangle)) \\
&\quad ((\alpha_0^* \langle \psi_0 | \langle 0 | - \alpha_1^* \langle \psi_1 | \langle 1 |) \langle 1 | \\
&\quad + (|1\rangle (\alpha_0 |0\rangle |\psi_0\rangle - \alpha_1 |1\rangle |\psi_1\rangle)) \\
&\quad ((\alpha_0^* \langle \psi_0 | \langle 0 | - \alpha_1^* \langle \psi_1 | \langle 1 |) \langle 1 | \\
&\quad + (|0\rangle (\alpha_0 |0\rangle |\psi_0\rangle + \alpha_1 |1\rangle |\psi_1\rangle)) \\
&\quad ((\alpha_0^* \langle \psi_0 | \langle 0 | + \alpha_1^* \langle \psi_1 | \langle 1 |) \langle 0 |))
\end{aligned} \tag{55}$$

$$\begin{aligned}
&= (|0\rangle (\alpha_0 |0\rangle |\psi_0\rangle + \alpha_1 |1\rangle |\psi_1\rangle)) \\
&\quad ((\alpha_0^* \langle \psi_0 | \langle 0 | + \alpha_1^* \langle \psi_1 | \langle 1 |) \langle 0 | \\
&\quad + (|1\rangle (\alpha_0 |0\rangle |\psi_0\rangle - \alpha_1 |1\rangle |\psi_1\rangle)) \\
&\quad ((\alpha_0^* \langle \psi_0 | \langle 0 | - \alpha_1^* \langle \psi_1 | \langle 1 |) \langle 1 |)
\end{aligned} \tag{56}$$

We see (once we renormalize this state) that this is exactly the density matrix of the ideal world that applies a random  $Z^a$  operation on the qubit and outputs  $a$  in the first register.

**Case 2: malicious verifier.** We prove now the equivalence of the ideal and real worlds if the adversary corrupts the verifier.

We consider now the case where the adversary  $\mathcal{A} = \hat{V}$  corrupts the verifier. The proof of this section is quite close to the second case of the proof of Theorem 3.1, so we will be quicker here. First, as before we cut  $\hat{V}$  into multiple parts ( $\hat{V}_0$  running against  $A_h$  to generate  $h$ ,  $\hat{V}_1$  will be the circuit run when receiving  $\perp$  (since the interaction will stop there in that case,  $\hat{V}_1$  only outputs a state for the environment), otherwise  $\hat{V}_2$  will be the circuit playing the ZK proof, and  $\hat{V}_3$  receiving the quantum state and outputting a final state and the measurements back to  $A$ ). Then, similarly to what we did before we define the simulator  $\mathbf{S}_{\hat{V}}$  as follows: first, the simulator will simulate the protocol  $\Pi_h$  by running  $\mathbf{S}_{h, \hat{B}_0}$  to get  $h$ . Then, if the output of the functionality  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$  is  $\perp$ , it will run  $\hat{V}_1$  and forward the output of  $\hat{V}_1$  to the environment. Otherwise, it samples all  $w_c^{(d)}$  starting with a 0, computes the hashes  $\{h_d^{(c)}\}_{c \in [n], d \in \{0,1\}}$ , and runs the ZK simulator  $\mathbf{S}_{zk, \hat{B}_2}((h_d^{(c)})_{c,d})$ . Then, for all  $c \in [n]$ , it will sample  $r^{(c)} \xleftarrow{\$} \{0,1\}$  and send the states  $\rho^{(c)} = |0\rangle |w_0^{(c)}\rangle + (-1)^{r^{(c)}} |1\rangle |w_1^{(c)}\rangle$ . After receiving the  $\{s^{(c)}\}_{c \in \{0,1\}}$ , it defines  $f(T, (m^{(j)})_{j \in T}, (r^{(i)})_{i \in [n] \setminus T}) := (r^{(i)} \oplus \bigoplus_k s^{(i)}[k](w_0^{(i)} \oplus w_1^{(i)})[k])_{i \in [n] \setminus T}$  and sends  $f$  to the functionality.

To prove that we have  $P \rightsquigarrow \hat{V} \approx_c \tilde{P} \xrightarrow{\mathcal{F}_{\text{SemCol}}^{\text{Pred}}} \mathbf{S}_{\hat{V}}$  ( $\approx_c$  are replaced with  $\approx_s$  if  $\{V\} \in S$ ), we can first study the case where the classical message  $T$  sent to the prover is such that  $\text{Pred}(T) = \perp$ . In that case, by definition of  $P$ ,  $P$  will abort, and send  $\perp$  to  $\hat{V}$ , which is also exactly what  $\mathbf{S}_{\hat{V}}$  is doing once  $\tilde{P}$  sent the abort bit forwarded by the functionality to the simulator. Now, we focus on  $T$  such that  $\text{Pred}(T) = \top$ , and therefore  $\hat{V}_1$  is never called. We can design as before a series of games, where we also cut  $P$  in a part that runs  $\Pi_h$ , a part  $P_0$  that checks  $\text{Pred}$ , a part  $P_1$  that measures the state and samples  $w$ 's, a part  $P_2$  that runs the ZK proof, and the last part  $P_3$  that runs the rest of the protocol (we refer to Theorem 3.1 for more details on the steps that are almost identical).

- We start from  $P \rightsquigarrow \hat{V}$ . Then, we replace  $P_0$  interacting with  $\hat{B}_0$  with  $\mathbf{S}_{h, \hat{B}_0}$  to get  $h$ . Both worlds are indistinguishable since  $\Pi_h \text{CS}_{S_h}$ -QSA realizes  $\mathcal{F}_{CRS}^{\text{Gen}}$ .
- Then, since the statement proven is true, we can replace  $P_2$  interacting with  $V_2$  with  $\mathbf{S}_{z_k, V_2}$  that only takes as input the hashes. This is indistinguishable since  $\Pi_{z_k} \text{CS}_S$ -QSA realizes  $\mathcal{F}_{ZK}$ .
- Then, we can now sample all  $w$ 's such that they start with a 0, which is indistinguishable thanks to the hardcore second-bit property (Definition 2.9) of  $h$ .
- Then, we sample  $r^{(i)}$  for all  $i \in [n]$ , and instead of measuring the qubits for  $i \in T$ , we rotate them according to  $Z^{r^{(i)}}$ . This is statistically indistinguishable since neither  $l$  nor  $r^{(i)}$  are reused anywhere else, and are therefore discarded. But one can easily see that measuring and discarding the outcome is statistically equivalent to applying  $Z^{r^{(i)}}$  and discarding  $r^{(i)}$ . This can easily be seen diagrammatically (using the doubling formalism), or via simple computations:

If we purify a state as  $\alpha_0 |0\rangle |\psi_0\rangle + \alpha_1 |1\rangle |\psi_1\rangle$ , then applying a random  $Z$  phase on the first qubit gives the density matrix:

$$\frac{1}{2}((\alpha_0 |0\rangle |\psi_0\rangle + \alpha_1 |1\rangle |\psi_1\rangle)(\alpha_0^* \langle\psi_0| \langle 0| + \alpha_1^* \langle\psi_1| \langle 1|) \quad (57)$$

$$+ (\alpha_0 |0\rangle |\psi_0\rangle - \alpha_1 |1\rangle |\psi_1\rangle)(\alpha_0^* \langle\psi_0| \langle 0| - \alpha_1^* \langle\psi_1| \langle 1|) \\ = |\alpha_0|^2 |0\rangle |\psi_0\rangle \langle\psi_0| \langle 0| + |\alpha_1|^2 |1\rangle |\psi_1\rangle \langle\psi_1| \langle 1| \quad (58)$$

and this second line corresponds to the density matrix of a (non-destructive) measurement in the computational basis.

- Finally, by reorganizing the elements and using the functionality with the appropriately chosen function  $f$  defined above to pick the appropriate  $r^{(i)} \oplus \langle s^{(i)}, w_0^{(i)} \oplus w_1^{(i)} \rangle$  only for  $i \in [n] \setminus T$ , we obtain the ideal world concluding the proof.

**Case 3: malicious prover.** We prove now the equivalence of the ideal and real worlds if the adversary corrupts the prover.

We consider now the case where the adversary  $\mathcal{A} = \hat{P}$  corrupts the prover. The proof of this section is quite close to the last case of the proof of Theorem 3.1, so we will be quicker here. First, we can divide  $\hat{P}$  and  $V$  into multiple parts:  $\hat{P}_0$  will interact with  $V_0$  that will play the protocol  $V_h$  to obtain  $h$ , then  $\hat{P}_1$  will interact with  $V_1$  to run the ZK protocol, and finally  $\hat{P}_2$  will interact with  $V_2$  for the remaining part of the protocol.

Now, we can, as before, define a series of indistinguishable worlds.

- First, we start from the ideal world, and we replace  $\hat{P}_0$  and  $V_0$  with the simulator  $\mathbf{S}_{h, \hat{P}_0}$  interacting with  $\mathcal{F}_H^{\text{Gen}}$ . This is indistinguishable since  $\Pi_h$  realises  $\mathcal{F}_H^{\text{Gen}}$ .



- Similarly, we replace  $\hat{P}_1$  and  $V_1$  with the simulator  $S_{z^k, \hat{P}_1}$  interacting
- Then, since the simulator has now access to the set  $T$ , it can measure all the states in  $T$  once the quantum test passes. To show that it is indistinguishable, we use the exact same argument as the one made in case 3 of Theorem 3.1 to show that  $\text{World}_2 \approx \text{World}_3$ . The only difference is that now we measure more states (but we can apply sequentially the same argument for one state at a time, and since the number of states is polynomial the distinguishing probability is still negligible: note that in the current proof the ZK contains more statements, but in particular the statements needed in Theorem 3.1 are fulfilled). The second difference is that we show that the trace distance between  $\rho$  and  $Z_1 \rho Z_1^\dagger$  is smaller than  $2t\sqrt{\beta}$  (Eq. (31)), while here we want to show that the trace distance between  $\rho$  and the measured  $\rho$  (non-destructively and in the computational basis) denoted  $\rho'$  is negligible. However, we showed in Eq. (58) that rotating by a random  $Z^a$  (and discarding the  $a$ ) is strictly equivalent to measuring (non-destructively) and discarding the outcome. Therefore:

$$\text{TD}(\rho, M\rho) = \text{TD}\left(\rho, \frac{1}{2}\rho + \frac{1}{2}Z_1\rho Z_1^\dagger\right) \quad (59)$$

$$\leq \frac{1}{2}(\text{TD}(\rho, \rho) + \text{TD}(\rho, Z_1\rho Z_1^\dagger)) \quad (60)$$

$$\leq \frac{1}{2}2t\sqrt{\beta} \leq 2t\sqrt{\beta} \quad (61)$$

which allows us to conclude.

- Then, because the state is collapsed before applying the Hadamard basis measurement on its second register, it is still collapsed after applying this measurement: so we can indistinguishably apply a second measurement in the computational basis after.
- Then, we can just attribute the operations to the appropriate parties and  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$  to obtain the ideal world: the simulator will run all the tasks, except that it sends  $\perp$  to  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$  if the ZK protocol aborted, and otherwise forwards  $T$  and sets  $\forall i \in [n] \setminus T, r_i = 0$ , while  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$  performs the second measurement in the computational basis on qubits in  $T$  (since  $r_i$ 's are all equal to 0,  $Z^{r_i}$  is the identity), as expected by definition. Since all these attribution does not change the global map performed by all parties, this is statistically indistinguishable, concluding the proof of security. □

*Proof of Corollary 4.15.* The first statement is a quite direct application of Theorem 4.10, where we define:

- $M$  like in Definition 4.11,
- for any  $T \in [n] = \mathcal{C}$ , if  $\text{Pred}(T) = \perp$  then  $E_T = \emptyset$ , otherwise:

$$E_T := \{|T\rangle \otimes |r\rangle \otimes (|l\rangle^{(T)} |+\rangle^{[n]\setminus T})\}_{l \in \{0,1\}^{|T|}, r \in \{0,1\}^{[n]\setminus T}} \quad (62)$$

- and for any  $T \subseteq [n]$ ,  $G_T$  is defined as the procedure that runs  $\rho \leftarrow G'(T)$ , samples  $(r^i)_{i \in [n]\setminus T} \stackrel{\$}{\leftarrow} \{0, 1\}^{n-|T|}$ , and outputs  $|T\rangle | (r^i)_{i \in [n]\setminus T} \rangle \otimes \rho$  (note that it is basically the operation performed by the ideal dummy party).

The only non-trivial check is to show that  $M$  are postponable measurement operators (Definition 4.9) with respect to  $\{G_T\}_{T, \mathcal{L}_T \neq \emptyset} = \{G_T\}_{T, \text{Pred}(T) = \top}$ . The idea is to do teleportation without

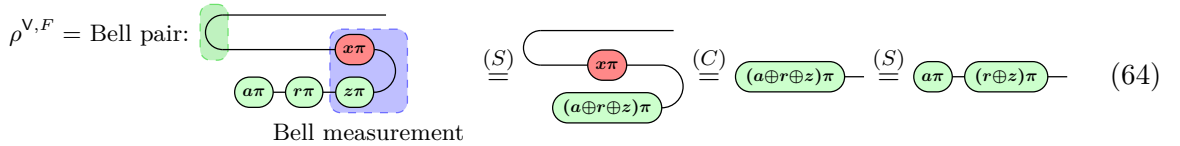
corrections, exploiting the fact that the uncorrected corrections just flip the encoded bit without changing its basis. (Note that to avoid dirty matrix computations, we give a more intuitive proof in the ZX calculus (but it is of course possible to derive the same proof with the usual matrix algebra). We refer curious readers unfamiliar with the ZX calculus to [vdWet20].) More precisely, we define  $\rho^{\mathbf{V},F}$  as the system containing  $n$  Bell pairs  $|00\rangle + |11\rangle$  shared between registers  $\mathbf{V}$  and  $F$  (the Bell pairs being between the  $\mathbf{V}^{(i)}$  and  $F^{(i)}$  for all  $i$ ). Then, we define  $M'$  as follows:

- $M'$  takes as input the register  $F$  containing Bell pairs, and a state  $\rho_0$  (sampled by  $G_T$ ).
- Then, it runs  $M\rho_0$ , to get a post-measured state  $\rho_1$  and a measurement outcome

$$(T, (m^{(j)})_{j \in T}, (r^{(i)})_{i \in [n] \setminus T}) \quad (63)$$

- Then, for all  $i \in [n]$ , it performs a Bell measurement (i.e. it does a projection on  $\{|0x\rangle + (-1)^z |1\bar{x}\rangle\}_{x \in \{0,1\}, z \in \{0,1\}}$ ) between the  $i$ -th qubit of  $F$  and the  $i$ -th qubit of  $\mathbf{V}$  to get outcomes  $(x^{(i)}, z^{(i)})$ .
- Finally, it outputs the outcome measurement  $(T, (x^{(j)} \oplus m^{(j)})_{j \in T}, (z^{(i)} \oplus r^{(i)})_{i \in [n] \setminus T})$

We prove now that  $MG_T \approx_s (I^{\mathbf{V}} \otimes M')(\rho^{\mathbf{V},F} \otimes G_T)$ . Note that both  $M$  and  $M'$  work separately on each system composed of the  $i$ -th qubit of each register, we can therefore just consider each system  $i \in [n]$  separately. Let  $T \in \mathcal{P}([n])$  such that  $\text{Pred}(T) = \top$ , and  $i \in [n]$ . Then  $G_T$  outputs on qubit  $i$  the state  $H^{\delta_{i \notin T}} |0\rangle$  with probability  $1/2$  and  $H^{\delta_{i \notin T}} |1\rangle$  otherwise. We do two cases: if  $i \notin T$ , then  $G_T$  generated a state  $H|a\rangle$  (we omit the index,  $H|a\rangle$  being represented as  $\textcircled{a\pi}$ — in the ZX calculus), and  $M$  (and therefore  $M'$ ) will perform a random  $Z^r$  flip on it (represented as  $\textcircled{r\pi}$ —), we can easily see (manually or with the ZX calculus) that the final state at the end of the procedure  $(I^{\mathbf{V}} \otimes M')(\rho^{\mathbf{V},F} \otimes G_T)$  is  $Z^{r \oplus z} H|a\rangle$ :



(the first equality comes from the spider fusion rule that allows to merge spiders of the same color, adding the angles modulo  $2\pi$ , the second rule being a particular case of the copy rule and the “only topology matters” principle that states that one can deform a graph without changing its interpretation). Note that since  $r$  is sampled uniformly at random, and since the outcome of the measurement  $z$  is independent of the value of  $r\pi$  and uniformly distributed (this is easy to see as the norm of this state (computable by computing the trace using the discarding ground operation that trivially absorbs<sup>38</sup> all terms:  $\| \textcircled{(a \oplus r \oplus z)\pi} \| = \sqrt{\frac{1}{2}}$ ) is independent of the value of all the variables). Therefore, by defining  $r' := r \oplus z$ ,  $r'$  is sampled uniformly at random, and the final state corresponds to a random rotation  $Z^{r'}$  of the original qubit, exactly like in the original  $M$  operation, concluding the proof<sup>39</sup>.

Now, if  $i \in T$ , then  $M'$  will measure the state  $|a\rangle$  provided by  $G_T$ , but since it is anyway already in the computational basis this measurement has no effect (except revealing the value of

<sup>38</sup>Note that here we removed all scalars and global phases as they are just constant re-normalisation factors.

<sup>39</sup>If wants to be even more formal, we can actually compute the exact density matrix of the whole process, including the outcome of the measurement, using either standard linear algebra, or diagrammatically using the discard construction [CJP<sup>+</sup>21] or using the doubling formalism [CK17] to represent density matrices, but this lead to the same result.

a). Then, the Bell measurement similarly gives:

$$\text{Diagram showing a Bell measurement process. On the left, two qubits are shown: a red circle labeled 'aπ' and a green circle labeled 'zπ'. They are connected by a line that loops around to a red circle labeled 'xπ'. This is labeled '(C)'. On the right, a red circle labeled 'aπ' is connected by a line that loops around to a red circle labeled 'xπ'. This is labeled '(S)'. Both are connected to a final red circle labeled '(a⊕x)π'.$$
(65)

where  $(a \oplus x)\pi = |a \oplus x\rangle$ . We can similarly see that the outcome  $x$  is independent of  $a$ . Therefore, since  $a$  and  $x$  are uniformly at random, the final state is a state sampled uniformly at random in the computational basis, exactly like in  $MG_T$ , concluding this part of the proof.

Finally, the second statement is trivial to check: since we just shown that the functionality  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$  is a ZKoQS, all protocols realizing it are ZKoQS protocols. However, we already know from Theorem 4.12 that Protocol 2 realizes  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$ , with a slightly different dummy ideal party, say  $\tilde{P}_0$ . However, since we obtain the new ideal dummy party and new protocol by doing exactly the same pre-processing ( $\tilde{P} = \tilde{P}_0(G_T)$  and  $P = P_0(G_T)$ , where  $P_0$  is the original party in Protocol 2), the distinguishing probability between the new real world and ideal world is lower, otherwise a distinguisher could apply  $G_T$  to attack the original protocol known to be secure.  $\square$

*Proof of Corollary 4.18.* The last statement is a direct consequence of Corollary 4.15 and of the definition of  $\text{ZKstatesQMA}_S^{\text{setup}k}$ . The first statement is obtained by instantiating the ZK protocol using the NIZK construction of [Unr15] secure in the Random Oracle model (we prove in Section 5 that their definition can be translated in a the quantum standalone framework). The second statement is obtained by instantiating the ZK protocol using the construction of [HSS11], proven secure in the plain-model assuming the hardness of LWE (this construction is already proven secure on the quantum standalone model, so no additional work is required).  $\square$

*Proof of Theorem 4.19.* This is a straightforward generalisation of the last part of the proof of Theorem 3.1. We start with correctness.

**Case 1: correctness (no corrupted party).** Since  $\Pi_{\text{SemCol}} \text{CS}_S\text{-QSA}$ -realises  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$ , we can indistinguishably replace  $\mathbf{A}_{\text{SemCol}}$  and  $\mathbf{B}_{\text{SemCol}}$  with dummy ideal adversaries interacting with  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$ . Then, we show that this world is indistinguishable from the real world: if the input  $B$  is such that  $\text{Pred}(T) = \perp$ , all parties would abort like in the real world. Otherwise, if  $\text{Pred}(T) = \top$ , then the functionality measures states not in  $B \dots$  but since these states are already measured, it left them unchanged. It also rotates the qubits in  $B$  (in the Hadamard basis) them by  $Z^{z^{(i)}}$ . Therefore, for all  $i \in B$ , the  $i$ -th qubit becomes  $\rho^{(i)} = Z^{r^{(i)} \oplus s^{(i)}} |+\rangle$ . After the rotation performed by  $\mathbf{B}$ , we get  $Z^{r^{(i)} \oplus s^{(i)} \oplus m_i} |+\rangle$ , and therefore the measurement in the Hadamard basis gives  $z^{(i)} = r^{(i)} \oplus s^{(i)} \oplus m_i$ . Since  $\mathbf{A}$  outputs for each  $i \in B$ ,  $r^{(i)} \oplus s^{(i)} \oplus z^{(i)} = m_i$ , the output of  $\mathbf{A}$  is exactly the same as in the ideal world, making both worlds indistinguishable, concluding the correctness proof.

**Case 2: malicious receiver Alice.** If  $\hat{\mathbf{A}}$  is malicious, we can cut  $\hat{\mathbf{A}}$  in two parts,  $\hat{\mathbf{A}}_0$  interacting with  $\mathbf{B}_{\text{SemCol}}$  and  $\hat{\mathbf{A}}_1$  doing the rest of the computation. Since  $\Pi_{\text{SemCol}} \text{CS}_S\text{-QSA}$ -realises  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$ , there exists a simulator  $\mathbf{S}_{\hat{\mathbf{A}}_0}$  such that  $\hat{\mathbf{A}}_0 \rightsquigarrow \mathbf{B}_{\text{SemCol}} \approx_c \mathbf{S}_{\hat{\mathbf{A}}_0} \overset{\mathcal{F}_{\text{SemCol}}^{\text{Pred}}}{\rightsquigarrow} \tilde{\mathbf{B}}_{\text{SemCol}}$  (or  $\approx_s$  if  $P \in S$ ). We can therefore indistinguishably replace the first one with the last one. Since  $\tilde{\mathbf{B}}_{\text{SemCol}}$  measures qubits not in  $B$ , the rotation on these qubits performed by Bob has no effect: as a result, we can remove them indistinguishably. Now, neither  $\mathbf{A}$  nor  $\mathbf{B}$  depends on  $m_i$  for  $i \notin B$ , we can therefore move them into the final simulator, using the  $(m_i)_{i \in B}$  provided by  $\mathcal{F}_{\text{OT}}^{\text{Pred}}$  otherwise (note that if  $\text{Pred}(B) = \perp$ ,  $\mathcal{F}_{\text{OT}}^{\text{Pred}}$  would not provide these values and abort, but this is not an

issue since anyway  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$  also aborts in that case). This new world is therefore equal to the ideal world, and indistinguishable from the previous world, concluding the proof of security.

**Case 3: malicious sender Bob.** If  $\hat{\mathbf{B}}$  is malicious, we can cut  $\hat{\mathbf{B}}$  in two parts,  $\hat{\mathbf{B}}_0$  interacting with  $\mathbf{A}_{\text{SemCol}}$  and  $\hat{\mathbf{B}}_1$  doing the rest of the protocol. Since  $\Pi_{\text{SemCol}} \text{CS}_S\text{-QSA}$ -realises  $\mathcal{F}_{\text{SemCol}}^{\text{Pred}}$ , there exists a simulator  $\mathbf{S}_{\hat{\mathbf{B}}_0}$  such that  $\mathbf{A}_{\text{SemCol}} \rightsquigarrow \hat{\mathbf{B}}_0 \approx_c \mathbf{B}_{\text{SemCol}} \xrightarrow{\mathcal{F}_{\text{SemCol}}^{\text{Pred}}} \mathbf{S}_{\hat{\mathbf{B}}_0}$  (or  $\approx_s$  if  $\mathbf{P} \in \mathcal{S}$ ). We can therefore indistinguishably replace the first one with the last one. Then, since the qubits in  $B$  are already in the computational basis, the functionality can skip the measurements of these qubits without being detected. Similarly, as we already saw it earlier, since sampling a qubit in  $|0\rangle$  or  $|1\rangle$  is indistinguishable from sampling a qubit in  $|+\rangle$  or  $|-\rangle$  (the encoded value being discarded), we can indistinguishably apply an  $H$  gate on these qubits to turn them back into qubits in the Hadamard basis, and also compute  $r^{(i)} \oplus s^{(i)} \oplus z^{(i)}$  for these qubits (it will be discarded anyway). Then, except for the return procedure of  $\mathbf{P}$  that discards some terms, all the steps are independent of  $m_i$ 's: we can therefore move everything into our final simulator, let it send the  $r^{(i)} \oplus s^{(i)} \oplus z^{(i)}$  to the ideal functionality  $\mathcal{F}_{\text{OT}}^{\text{Pred}}$  that will be in charge of discarded elements not it  $T$ . This last step is indistinguishable as we only moved some operations, ending this proof of security.  $\square$

## D Proof of the composability of [Unr15]

**Definition D.1.** 1. **Completeness** ([Unr15, Def. 1]):  $\Pi_{zk}^H$  is complete iff for any quantum-polynomial-time oracle algorithm  $A$  and advice  $(\sigma_\lambda)_{\lambda \in \mathbb{N}}$ ,

$$\Pr \left[ (x, w) \in \mathcal{R} \wedge y = 0 \mid H \leftarrow \text{ROdist}, (x, w) \leftarrow A^H(\sigma_\lambda), \right. \\ \left. y \leftarrow \text{OUT}_V \langle \mathbf{P}(x, w) \xrightarrow{H} \mathbf{V} \rangle \right] \leq \text{negl}(\lambda). \quad (66)$$

2. **Zero-knowledge** ([Unr15, Def. 2]):  $\Pi_{zk}^H$  is zero-knowledge iff there exists a polynomial-time simulator  $\mathbf{S} = (S_{\text{init}}, \mathbf{S}_P)$  such that for every quantum-polynomial-time oracle algorithm  $A$  and advice  $(\sigma_\lambda)_{\lambda \in \mathbb{N}}$ ,

$$\left| \Pr \left[ z = 1 \mid H \leftarrow \text{ROdist}, z \leftarrow A^{H, \mathbf{P}}(\sigma_\lambda) \right] \right. \\ \left. - \Pr \left[ z = 1 \mid H \leftarrow S_{\text{init}}, z \leftarrow A^{H, \mathbf{S}_P}(\sigma_\lambda) \right] \right| \leq \text{negl}(\lambda). \quad (67)$$

Since in the quantum setting we cannot allow the simulator to learn the input for each query, because this can be done in superposition, here the simulator  $S_{\text{init}}$  outputs a circuit describing a classical function representing the initial random oracle instead. We assume that both  $S_{\text{init}}$  and  $\mathbf{S}_P$  have access to the polynomial upper bound on the runtime of  $A$ .

3. **Online-extractability** ([Unr15, Def. 3]):  $\Pi_{zk}^H$  is online extractable with respect to  $S_{\text{init}}$  iff there exists a polynomial-time extractor  $E$  such that for any quantum-polynomial-time oracle algorithm  $A$  and advice  $(\sigma_\lambda)_{\lambda \in \mathbb{N}}$ ,

$$\Pr \left[ y = 1 \wedge (x, w) \notin \mathcal{R} \mid H \leftarrow S_{\text{init}}(), (x, \pi) \leftarrow A^H(\sigma_\lambda), y \leftarrow \mathbf{V}^H(x, \pi), \right. \\ \left. w \leftarrow E(H, x, \pi) \right] \leq \text{negl}(\lambda). \quad (68)$$

We assume that both  $S_{\text{init}}$  and  $E$  have access to the polynomial upper bound on the runtime of  $A$ .

Note that Unruh's original definition he considers only uniform adversaries, here we assume that the above conditions hold for the protocol when the adversary receives advice.

*Proof of Theorem 5.1.* We define trivially the dummy parties  $\tilde{\Pi}_{zk} = (\tilde{P}, \tilde{V})$  that forward the inputs/outputs to/from  $\mathcal{F}_{ZK}^R$ . We want to show that for any (poly-)time adversary  $\mathcal{A}$  there exists a (poly-)time simulator  $\mathbf{S}$  such that, for any poly-time distinguisher  $\mathbf{Z}$  and input state  $\sigma_\lambda$ , we have  $\text{REAL}_{\Pi_{zk}, \mathcal{A}, \mathbf{Z}}^\sigma \approx \text{IDEAL}_{\tilde{\Pi}_{zk}, \mathbf{S}, \mathbf{Z}}^{\sigma, \mathcal{F}_{ZK}^R}$ . We will split the proof depending on the parties that the static adversary  $\mathcal{A}$  corrupts (nobody, the prover  $P$  or the verifier  $V$ ).

Note that in the non-interactive protocol  $\Pi_{zk}^H$  the prover rejects if it receives a non-valid witness  $(x, w) \notin \mathcal{R}$ , which is essential.

**Case 1: correctness (no corrupted party).** For any bipartite input state  $\sigma_\lambda^{P, \mathbf{Z}} \in D(S_\lambda \otimes R_\lambda)$ , with  $R_\lambda$  an arbitrary reference system, we want to show that for any environment  $\mathbf{Z}$  the probability of distinguishing  $\Pi_{zk}^H$  from the ideal functionality  $\tilde{\Pi}_{zk}$  is negligible

$$\left| \Pr \left[ \mathbf{Z}(y, \sigma_\lambda^{\mathbf{Z}}) = 1 \mid H \leftarrow \text{ROdist}, y \leftarrow \text{OUT}_V \langle P(\sigma_\lambda^P) \overset{H}{\longleftrightarrow} V \rangle \right] - \Pr \left[ \mathbf{Z}(y, \sigma_\lambda^{\mathbf{Z}}) = 1 \mid H \leftarrow \text{ROdist}, y \leftarrow \text{OUT}_V \langle \tilde{P}(\sigma_\lambda^P) \overset{\mathcal{F}}{\longleftrightarrow} \tilde{V} \rangle \right] \right| \leq \text{negl}(\lambda). \quad (69)$$

In order to prove the above inequality we are interested in rewriting the probability of the distinguisher outputting 1 in terms of the input/output of the interaction. Since the prover expects a classical message, we can model  $\sigma_\lambda^{P, \mathbf{Z}}$  as a quantum instrument:

$$\sigma_\lambda^{P, \mathbf{Z}} = \sum_{x, w} p_{x, w} |x, w\rangle \langle x, w| \otimes \sigma_{\lambda, x, w}. \quad (70)$$

This allows to write the LHS of Eq. (69) as

$$\sum_{x, w} p_{x, w} \Pr \left[ \mathbf{Z}(y, \sigma_{\lambda, x, w}) = 1 \mid H \leftarrow \text{ROdist}, y \leftarrow \text{OUT}_V \langle P(x, w) \overset{H}{\longleftrightarrow} V \rangle \right] \quad (71)$$

$$= \sum_{b \in \{0, 1\}} \sum_{x, w} \left( p_{x, w} \Pr [\mathbf{Z}(b, \sigma_{\lambda, x, w}) = 1] \cdot \Pr \left[ y = b \mid H \leftarrow \text{ROdist}, y \leftarrow \text{OUT}_V \langle P(x, w) \overset{H}{\longleftrightarrow} V \rangle \right] \right) \quad (72)$$

For the dummy protocol  $\tilde{\Pi}_{zk}$  the above equation has a very simple form since if  $(x, w) \in \mathcal{R}$  (resp.  $(x, w) \notin \mathcal{R}$ ), then  $\tilde{\Pi}_{zk}$  will output 1 (resp. 0) with probability 1. Therefore, for any input state  $\sigma_\lambda^{P, \mathbf{Z}}$  we can simplify the RHS of Eq. (69) to

$$\sum_{(x, w) \in \mathcal{R}} p_{x, w} \Pr [\mathbf{Z}(1, \sigma_{\lambda, x, w}) = 1] + \sum_{(x, w) \notin \mathcal{R}} p_{x, w} \Pr [\mathbf{Z}(0, \sigma_{\lambda, x, w}) = 1] \quad (73)$$

For the honest protocol  $\Pi_{zk}^H$ , note that for invalid witnesses  $(x, w) \notin \mathcal{R}$  the honest prover  $P^H$  will also always reject, therefore for  $(x, w) \notin \mathcal{R}$ :

$$\Pr \left[ y = 1 \mid H \leftarrow \text{ROdist}, y \leftarrow \text{OUT}_V \langle P(x, w) \overset{H}{\longleftrightarrow} V \rangle \right] = 0, \quad (74)$$

thus for an arbitrary mixture of invalid witnesses  $\sum_{(x, w) \notin \mathcal{R}} q_{x, w} |x, w\rangle \langle x, w|$  we will have that

$$\sum_{(x, w) \notin \mathcal{R}} q_{x, w} \Pr \left[ y = 1 \mid H \leftarrow \text{ROdist}, y \leftarrow \text{OUT}_V \langle P(x, w) \overset{H}{\longleftrightarrow} V \rangle \right] = 0, \quad (75)$$

and consequently

$$\sum_{(x,w) \notin \mathcal{R}} q_{x,w} \Pr \left[ y = 0 \mid H \leftarrow \text{ROdist}, y \leftarrow \text{OUT}_V \langle P(x,w) \xleftrightarrow{H} V \rangle \right] = 1. \quad (76)$$

The case of valid witnesses is also easy as we know from completeness Eq. (66) that given any input state  $(x,w) \in \mathcal{R}$ , if we pick the constant algorithm  $A : \sigma_\lambda^P \mapsto (x,w)$ , then

$$\Pr \left[ y = 0 \mid H \leftarrow \text{ROdist}, y \leftarrow \text{OUT}_V \langle P(x,w) \xleftrightarrow{H} V \rangle \right] \leq \text{negl}(\lambda), \quad (77)$$

thus for an arbitrary mixture of valid witnesses  $\sum_{(x,w) \in \mathcal{R}} q_{x,w} |x,w\rangle\langle x,w|$  we will have that

$$\sum_{(x,w) \in \mathcal{R}} q_{x,w} \Pr \left[ y = 0 \mid H \leftarrow \text{ROdist}, y \leftarrow \text{OUT}_V \langle P(x,w) \xleftrightarrow{H} V \rangle \right] \leq \text{negl}(\lambda), \quad (78)$$

and consequently

$$\sum_{(x,w) \in \mathcal{R}} q_{x,w} \Pr \left[ y = 1 \mid H \leftarrow \text{ROdist}, y \leftarrow \text{OUT}_V \langle P(x,w) \xleftrightarrow{H} V \rangle \right] \geq 1 - \text{negl}(\lambda). \quad (79)$$

We can combine the above Eqs. (75), (76), (78) and (79) to obtain the desired inequality Eq. (69) for any input  $\sigma_\lambda^{P,Z}$  and any distinguisher  $\mathbf{Z}$  by noting that for any received input  $b \in \{0,1\}$  and advice  $\sigma_{\lambda,x,w}$ , the probability of the distinguisher outputting 1 is bounded

$$\Pr [\mathbf{Z}(b, \sigma_{\lambda,x,w}) = 1] \leq 1, \quad (80)$$

and therefore by developing Eq. (69) in terms of the advice as in Eq. (72) we can bound the difference by

$$\begin{aligned} & \leq \left| \sum_{(x,w) \in \mathcal{R}} p_{x,w} \Pr \left[ y = 0 \mid H \leftarrow \text{ROdist}, y \leftarrow \text{OUT}_V \langle P(x,w) \xleftrightarrow{H} V \rangle \right] \right| \\ & \quad + \left| \sum_{(x,w) \in \mathcal{R}} p_{x,w} \left( \Pr \left[ y = 1 \mid H \leftarrow \text{ROdist}, y \leftarrow \text{OUT}_V \langle P(x,w) \xleftrightarrow{H} V \rangle \right] - 1 \right) \right| \\ & \quad + \left| \sum_{(x,w) \notin \mathcal{R}} p_{x,w} \left( \Pr \left[ y = 0 \mid H \leftarrow \text{ROdist}, y \leftarrow \text{OUT}_V \langle P(x,w) \xleftrightarrow{H} V \rangle \right] - 1 \right) \right| \\ & \quad + \left| \sum_{(x,w) \notin \mathcal{R}} p_{x,w} \Pr \left[ y = 1 \mid H \leftarrow \text{ROdist}, y \leftarrow \text{OUT}_V \langle P(x,w) \xleftrightarrow{H} V \rangle \right] \right| \\ & \leq \text{negl}(\lambda), \end{aligned} \quad (81)$$

**Case 2: malicious Alice.** If the adversary corrupts the prover  $\mathcal{A} = \hat{P}$ , we will use online-extractability to construct the desired simulator.

**Simulator  $\mathbf{S}_A$**  :=  $(S, E, S_{\text{init}})$ : Prover is corrupted.

1.  $S$  initializes  $\hat{P}$  with whatever input state it receives.

2.  $S$  obtains  $(x, \pi)$  from  $\hat{P}$ .
3.  $S$  initializes  $E$  with  $(x, \pi)$  and the description of the oracle  $H$  given by the simulator  $S_{\text{init}}$ .
4.  $E(H, x, \pi)$  extracts a witness  $w$  or an abort message  $\perp$  and sends it to  $S$ .
5.  $S$  sends  $(x, w)$  or  $\perp$  to  $\mathcal{F}_{ZK}^R$ .

We will now prove that no distinguisher can differentiate between the real protocol with corrupted Alice and the ideal functionality with the above simulator. This proof relies on the closeness of the following hybrid worlds (see Appendix D for a graphical depiction):

- $\text{World}_0 := \text{IDEAL}_{\tilde{\Pi}_{zk}, \mathbf{S}_{A, \mathbf{Z}}}^{\sigma, \mathcal{F}_{ZK}^R}$  is the ideal world. Consists of one output by the functionality (forwarded by the dummy verifier) which is accepting if the witness obtained by the extractor in Item 4 is valid, i.e.,  $(x, w) \in \mathcal{R}$ .
- $\text{World}_1$  is like  $\text{World}_0$  except that we substitute the dummy verifier by a merge of Unruh's verifier  $V$  and the simulator. In particular, we replace  $\tilde{V}$  by a verifier  $V_1$  that forwards the proof from the simulator to the extractor  $E$ . If the extractor provides a witness  $w$ , then  $V_1$  accepts  $y = x$  and else aborts  $y = \perp$ .
- $\text{World}_2$  is like  $\text{World}_1$  except that we drop the extractor (as it is only being used to check the proof) and the dummy verifier  $V_1$  which is only forwarding information, and we use Unruh's verifier  $V$  to perform the check of the proof received by the simulator  $S$  instead.
- $\text{World}_3$  is like  $\text{World}_2$  except that we drop the simulator  $S$  as it is only forwarding the information to the verifier.
- $\text{World}_4$  differs from  $\text{World}_3$  in that we replace the simulator  $S_{\text{init}}$  by the oracle  $H$  that is simulating. Note that now  $\text{World}_4 := \text{REAL}_{\tilde{\Pi}_{zk}, \mathbf{A}, \mathbf{Z}}^{\sigma}$ .

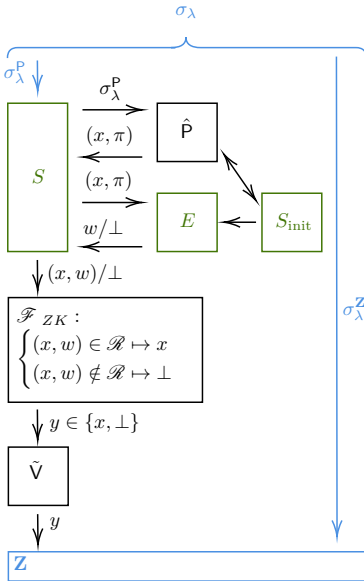


Fig. 16: World<sub>0</sub>

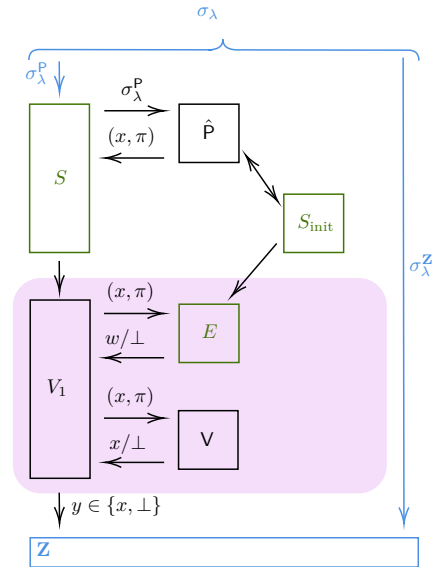


Fig. 17: World<sub>1</sub>

The similarity of the first two worlds,  $\text{World}_0 \approx \text{World}_1$ , is a consequence of online-extractability. More precisely, if we could distinguish these two worlds, there would exist an input  $\sigma_\lambda^P$  such that

$$\left| \Pr \left[ \mathbf{Z}(y, \sigma_\lambda^Z) = 1 \mid H \leftarrow S_{\text{init}}(), y \leftarrow \text{OUT}_{\tilde{V}} \langle \mathbf{S}_{\mathcal{A}}(\sigma_\lambda^P) \stackrel{\mathcal{F}}{\rightsquigarrow} \tilde{V} \rangle \right] - \Pr \left[ \mathbf{Z}(y, \sigma_\lambda^Z) = 1 \mid H \leftarrow S_{\text{init}}(), y \leftarrow \text{OUT}_{V_1} \langle \mathbf{S}_{\mathcal{A}}(\sigma_\lambda^P) \rightsquigarrow V_1 \rangle \right] \right| > \text{negl}(\lambda). \quad (82)$$

In order to work with them jointly we expand them in terms of the verifier accepting/rejecting  $y \in \{x, \perp\}$ :

$$\left| \Pr \left[ \mathbf{Z}(y, \sigma_\lambda^Z) = 1 \mid H \leftarrow S_{\text{init}}(), y \leftarrow \text{OUT}_{\tilde{V}} \langle \mathbf{S}_{\mathcal{A}}(\sigma_\lambda^P) \stackrel{\mathcal{F}}{\rightsquigarrow} \tilde{V} \rangle \right] - \Pr \left[ \mathbf{Z}(y, \sigma_\lambda^Z) = 1 \mid H \leftarrow S_{\text{init}}(), y \leftarrow \text{OUT}_{V_1} \langle \mathbf{S}_{\mathcal{A}}(\sigma_\lambda^P) \rightsquigarrow V_1 \rangle \right] \right| \quad (83)$$

$$= \left| \sum_{b \in \{x, \perp\}} \Pr \left[ \mathbf{Z}(b, \sigma_\lambda^Z) = 1 \right] \Pr \left[ y = b \mid H \leftarrow S_{\text{init}}(), y \leftarrow \text{OUT}_{\tilde{V}} \langle \mathbf{S}_{\mathcal{A}}(\sigma_\lambda^P) \stackrel{\mathcal{F}}{\rightsquigarrow} \tilde{V} \rangle \right] - \sum_{b \in \{x, \perp\}} \Pr \left[ \mathbf{Z}(b, \sigma_\lambda^Z) = 1 \right] \Pr \left[ y = b \mid H \leftarrow S_{\text{init}}(), y \leftarrow \text{OUT}_{V_1} \langle \mathbf{S}_{\mathcal{A}}(\sigma_\lambda^P) \rightsquigarrow V_1 \rangle \right] \right| \quad (84)$$

$$= \left| \Pr \left[ \mathbf{Z}(x, \sigma_\lambda^Z) = 1 \right] \cdot \sum_{M \in \{\tilde{V}, V_1\}} (-1)^{\delta_V} \Pr \left[ y = x \mid H \leftarrow S_{\text{init}}(), y \leftarrow \text{OUT}_M \langle \mathbf{S}_{\mathcal{A}}(\sigma_\lambda^P) \stackrel{\mathcal{F}}{\rightsquigarrow} M \rangle \right] + \Pr \left[ \mathbf{Z}(\perp, \sigma_\lambda^Z) = 1 \right] \cdot \sum_{M \in \{\tilde{V}, V_1\}} (-1)^{\delta_V} \Pr \left[ y = \perp \mid H \leftarrow S_{\text{init}}(), y \leftarrow \text{OUT}_M \langle \mathbf{S}_{\mathcal{A}}(\sigma_\lambda^P) \rightsquigarrow M \rangle \right] \right| \quad (85)$$

$$\leq \left| \sum_{M \in \{\tilde{V}, V_1\}} (-1)^{\delta(V)} \Pr \left[ y = x \mid H \leftarrow S_{\text{init}}(), y \leftarrow \text{OUT}_M \langle \mathbf{S}_{\mathcal{A}}(\sigma_\lambda^P) \stackrel{\mathcal{F}}{\rightsquigarrow} M \rangle \right] \right| + \left| \sum_{M \in \{\tilde{V}, V_1\}} (-1)^{\delta(V)} \Pr \left[ y = \perp \mid H \leftarrow S_{\text{init}}(), y \leftarrow \text{OUT}_M \langle \mathbf{S}_{\mathcal{A}}(\sigma_\lambda^P) \rightsquigarrow M \rangle \right] \right|. \quad (86)$$

We write down the probabilities of each protocol accepting,  $y = x$ , to better visualize:

$$\Pr \left[ y = x \mid H \leftarrow S_{\text{init}}(), y \leftarrow \text{OUT}_{\tilde{V}} \langle \mathbf{S}_{\mathcal{A}}(\sigma_\lambda^P) \stackrel{\mathcal{F}}{\rightsquigarrow} \tilde{V} \rangle \right] \quad (87)$$

$$= \Pr \left[ y = x \mid H \leftarrow S_{\text{init}}(), (x, \pi) \leftarrow \hat{\mathbf{P}}^H(\sigma_\lambda^P), w' \leftarrow E(x, \pi, H), \right. \quad (88)$$

$$\left. y \leftarrow \mathcal{F}_{ZK}(x, w') \right]$$

$$= \Pr \left[ w' \neq \perp \wedge (x, w') \in \mathcal{R} \mid H \leftarrow S_{\text{init}}(), (x, \pi) \leftarrow \hat{\mathbf{P}}^H(\sigma_\lambda^P), \right. \quad (89)$$

$$\left. w' \leftarrow E(x, \pi, H) \right],$$

$$\Pr \left[ y = x \mid H \leftarrow S_{\text{init}}(), y \leftarrow \text{OUT}_{V_1} \langle \mathbf{S}_{\mathcal{A}}(\sigma_\lambda^P) \rightsquigarrow V_1 \rangle \right] \quad (90)$$



$$= \Pr \left[ y = x \wedge w' \neq \perp \mid H \leftarrow S_{\text{init}}(), (x, \pi) \leftarrow \hat{\mathbf{P}}^H(\sigma_\lambda^{\mathbf{P}}), w' \leftarrow E(x, \pi, H), \right. \\ \left. y \leftarrow V(x, \pi) \right] \quad (91)$$

$$= \Pr \left[ y = x \wedge w' \neq \perp \wedge (x, w') \in \mathcal{R} \mid H \leftarrow S_{\text{init}}(), (x, \pi) \leftarrow \hat{\mathbf{P}}^H(\sigma_\lambda^{\mathbf{P}}), \right. \\ \left. w' \leftarrow E(x, \pi, H), y \leftarrow V(x, \pi) \right] \quad (92)$$

$$+ \Pr \left[ y = x \wedge w' \neq \perp \wedge (x, w') \notin \mathcal{R} \mid H \leftarrow S_{\text{init}}(), (x, \pi) \leftarrow \hat{\mathbf{P}}^H(\sigma_\lambda^{\mathbf{P}}), \right. \\ \left. w' \leftarrow E(x, \pi, H), y \leftarrow V(x, \pi) \right], \quad (93)$$

where in the last equality we just used the marginal probability expansion. Moreover, by online-extractability Eq. (68), we know that for all  $\sigma_\lambda^{\mathbf{P}}$ :

$$\Pr \left[ y = x \wedge w' \neq \perp \wedge (x, w') \notin \mathcal{R} \mid H \leftarrow S_{\text{init}}(), (x, \pi) \leftarrow \hat{\mathbf{P}}^H(\sigma_\lambda^{\mathbf{P}}), \right. \\ \left. y \leftarrow V(x, \pi), w' \leftarrow E(x, \pi, H) \right] < \text{negl}(\lambda), \quad (94)$$

$$\Pr \left[ y = x \wedge w' \neq \perp \wedge (x, w') \in \mathcal{R} \mid H \leftarrow S_{\text{init}}(), (x, \pi) \leftarrow \hat{\mathbf{P}}^H(\sigma_\lambda^{\mathbf{P}}), \right. \\ \left. y \leftarrow V(x, \pi), w' \leftarrow E(x, \pi, H) \right] > 1 - \text{negl}(\lambda), \quad (95)$$

thus bounding Eq. (93). Note that we can bound Eq. (92) by Eq. (89), as the former adds one more restriction to the output. Therefore, the probabilities of each protocol accepting is nearly the same as

$$\left| \Pr \left[ y = x \mid H \leftarrow S_{\text{init}}(), y \leftarrow \text{OUT}_{\tilde{V}} \langle \mathbf{S}_{\mathcal{A}}(\sigma_\lambda^{\mathbf{P}}) \overset{\mathcal{F}}{\rightsquigarrow} \tilde{V} \rangle \right] \right. \\ \left. - \Pr \left[ y = x \mid H \leftarrow S_{\text{init}}(), y \leftarrow \text{OUT}_{V_1} \langle \mathbf{S}_{\mathcal{A}}(\sigma_\lambda^{\mathbf{P}}) \overset{\mathcal{F}}{\rightsquigarrow} V_1 \rangle \right] \right| \quad (96)$$

$$\leq \Pr \left[ w' \neq \perp \wedge (x, w') \in \mathcal{R} \mid H \leftarrow S_{\text{init}}(), (x, \pi) \leftarrow \hat{\mathbf{P}}^H(\sigma_\lambda^{\mathbf{P}}), \right. \\ \left. w' \leftarrow E(x, \pi, H) \right] \quad (97)$$

$$- \Pr \left[ y = x \wedge w' \neq \perp \wedge (x, w') \in \mathcal{R} \mid H \leftarrow S_{\text{init}}(), (x, \pi) \leftarrow \hat{\mathbf{P}}^H(\sigma_\lambda^{\mathbf{P}}), \right. \\ \left. w' \leftarrow E(x, \pi, H), y \leftarrow V(x, \pi) \right] + \text{negl}(\lambda) \quad (98)$$

$$\leq \text{negl}(\lambda). \quad (99)$$

This bound is enough to show the similarity of the worlds as

$$\Pr \left[ y = \perp \mid H \leftarrow S_{\text{init}}(), y \leftarrow \text{OUT}_M \langle \mathbf{S}_{\mathcal{A}}(\sigma_\lambda^{\mathbf{P}}) \overset{\mathcal{F}}{\rightsquigarrow} M \rangle \right] \quad (100) \\ = 1 - \Pr \left[ y = x \mid H \leftarrow S_{\text{init}}(), y \leftarrow \text{OUT}_M \langle \mathbf{S}_{\mathcal{A}}(\sigma_\lambda^{\mathbf{P}}) \overset{\mathcal{F}}{\rightsquigarrow} M \rangle \right],$$

for both  $M = \tilde{V}$  and  $M = V_1$ .

We can also prove  $\text{World}_1 \approx \text{World}_2$  using online extractability, as the verifier  $V_1$  is only using the extractor to see if it does not abort. More precisely, following the same argument as before, and expanding the probability of accepting for the protocol from  $\text{World}_2$ :

$$\Pr \left[ y = x \mid H \leftarrow S_{\text{init}}(), y \leftarrow \text{OUT}_{\tilde{V}} \langle \mathbf{S}_{\mathcal{A}}(\sigma_\lambda^{\mathbf{P}}) \overset{\mathcal{F}}{\rightsquigarrow} \tilde{V} \rangle \right] \quad (101)$$

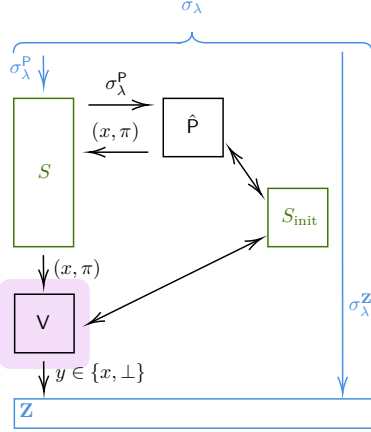


Fig. 18: World<sub>2</sub>

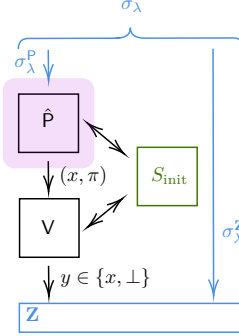


Fig. 19: World<sub>3</sub>

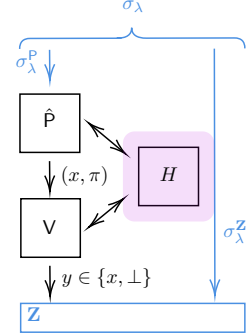


Fig. 20: World<sub>4</sub>

$$= \Pr \left[ y = x \mid H \leftarrow S_{\text{init}}(), (x, \pi) \leftarrow \hat{\mathbf{P}}^H(\sigma_\lambda^{\mathbf{P}}), y \leftarrow \mathbf{V}(x, \pi) \right] \quad (102)$$

$$= \Pr \left[ y = x \wedge (x, w') \in \mid H \leftarrow S_{\text{init}}(), (x, \pi) \leftarrow \hat{\mathbf{P}}^H(\sigma_\lambda^{\mathbf{P}}), y \leftarrow \mathbf{V}(x, \pi) \right] \\ + \Pr \left[ y = x \mid H \leftarrow S_{\text{init}}(), (x, \pi) \leftarrow \hat{\mathbf{P}}^H(\sigma_\lambda^{\mathbf{P}}), y \leftarrow \mathbf{V}(x, \pi) \right]. \quad (103)$$

In order to proof the equivalence it is enough to note that

$$\left| \Pr \left[ y = x \mid H \leftarrow S_{\text{init}}(), y \leftarrow \text{OUT}_{V_1}(\mathbf{S}_{\mathcal{A}}(\sigma_\lambda^{\mathbf{P}}) \rightsquigarrow V_1) \right] \right. \\ \left. - \Pr \left[ y = x \mid H \leftarrow S_{\text{init}}(), y \leftarrow \text{OUT}_{V_1}(\mathbf{S}_{\mathcal{A}}(\sigma_\lambda^{\mathbf{P}}) \rightsquigarrow V) \right] \right| \quad (104)$$

$$= \left| \Pr \left[ y = x \wedge w' \neq \perp \mid H \leftarrow S_{\text{init}}(), (x, \pi) \leftarrow \hat{\mathbf{P}}^H(\sigma_\lambda^{\mathbf{P}}), w' \leftarrow E(x, \pi, H), \right. \right. \\ \left. \left. y \leftarrow V(x, \pi) \right] \right. \\ \left. - \Pr \left[ y = x \mid H \leftarrow S_{\text{init}}(), (x, \pi) \leftarrow \hat{\mathbf{P}}^H(\sigma_\lambda^{\mathbf{P}}), y \leftarrow V(x, \pi) \right] \right|. \quad (105)$$

$$= \Pr \left[ y = x \wedge w' = \perp \mid H \leftarrow S_{\text{init}}(), (x, \pi) \leftarrow \hat{\mathbf{P}}^H(\sigma_\lambda^{\mathbf{P}}), w' \leftarrow E(x, \pi, H), \right. \\ \left. y \leftarrow V(x, \pi) \right] \quad (106)$$

$$= \Pr \left[ y = x \wedge w' = \perp \wedge (x, w') \notin \mathcal{R} \mid H \leftarrow S_{\text{init}}(), (x, \pi) \leftarrow \hat{\mathbf{P}}^H(\sigma_\lambda^{\mathbf{P}}), \right. \\ \left. w' \leftarrow E(x, \pi, H), y \leftarrow V(x, \pi) \right] \quad (107)$$

$$< \text{negl}(\lambda), \quad (108)$$

where in Eq. (107) we used that  $\{w' = \perp\} \subseteq \{(x, w') \notin \mathcal{R}\}$ .

The rest of the relations are obvious since World<sub>3</sub> is just World<sub>2</sub> with a different routing of the messages – the simulator  $S$  is only redirecting the information. World<sub>3</sub>  $\approx$  World<sub>4</sub> is changing the  $S_{\text{init}}$  by the oracle that it is simulating  $H$ .

**Case 3: malicious Bob.** If the adversary corrupts the verifier  $\mathcal{A} = V$ , the simulator from [Unr15] from the zero-knowledge property will be enough.

Recall that in his description of the adversary it also encompasses the distinguisher, but by allowing adversaries that receive advice, it is equivalent to assuming an adversary that outputs

a proof for the verifier. This is, Unruh's simulator  $\mathbf{S} = (S_{\text{init}}, \mathbf{S}'_P)$  fulfills

$$\begin{aligned} & \left| \Pr \left[ \mathbf{Z}(y, \sigma_\lambda^{\mathbf{Z}}) = 1 \mid H \leftarrow \text{ROdist}, (x, \pi) \leftarrow \mathbf{P}^H(\sigma_\lambda^{\mathbf{P}}), y \leftarrow \hat{\mathbf{V}}(x, \pi, \sigma_\lambda^{\mathbf{V}}) \right] \right. \\ & \quad \left. - \Pr \left[ \mathbf{Z}(y, \sigma_\lambda^{\mathbf{Z}}) = 1 \mid H \leftarrow S_{\text{init}}(), (x, \pi) \leftarrow \mathbf{S}'_P(\sigma_\lambda^{\mathbf{P}}), y \leftarrow \hat{\mathbf{V}}(x, \pi, \sigma_\lambda^{\mathbf{V}}) \right] \right| \\ & \quad < \text{negl}(\lambda). \end{aligned} \quad (109)$$

Note that the simulator  $\mathbf{S}'_P$  is replacing both the ideal functionality  $\mathcal{F}_{\text{ZK}}^{\mathcal{R}}$  and the dummy verifier  $\tilde{\mathbf{P}}$  in the ideal world, i.e. aborts whenever  $(x, w) \notin \mathcal{R}$  and runs  $\mathbf{S}_P(x, \sigma_\lambda^{\mathbf{V}})$  otherwise. However, we can easily modify this simulator to obtain the desired one in terms of the subsimulator  $\mathbf{S}_P$ .

**Simulator  $\mathbf{S}_{\mathcal{A}} := (S, S_{\text{init}})$ :** Verifier is corrupted.

1. If  $S$  does not receive an abort  $\perp$  message from the ideal functionality  $\mathcal{F}_{\text{ZK}}^{\mathcal{R}}$ , redirects the input  $x$  to the simulator  $\mathbf{S}_P$ . Else, it aborts.
2.  $S$  receives  $(x, \pi)$  from the simulator  $\mathbf{S}_P(x)$ .
3.  $S$  sends  $(x, \pi, \sigma_\lambda)$  to the verifier  $\mathbf{V}$ .
4.  $S$  redirects the output of the verifier  $\mathbf{V}$  to the distinguisher  $\mathbf{Z}$ .

It is clear that  $\mathbf{Z}$  cannot distinguish between the proofs provided by the verifier and the simulator, as our simulator is just a rewiring of Unruh's simulator, see Fig. 23.  $\square$

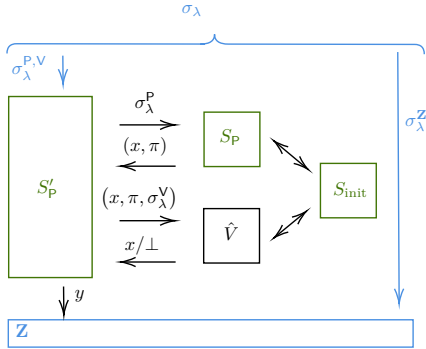


Fig. 21: Protocol for  $\mathbf{S}'_P$ .

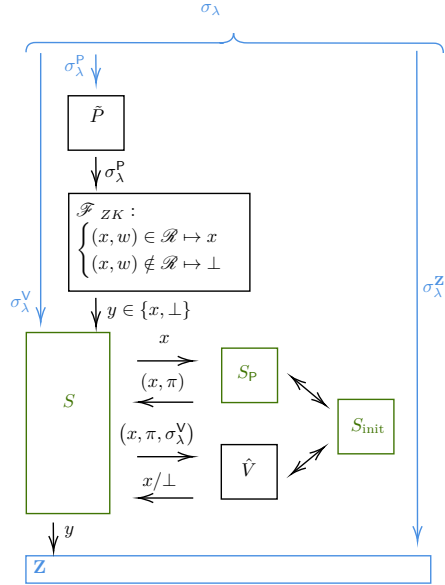


Fig. 22: Ideal functionality with  $\mathbf{S}_{\mathcal{A}}$ .

Fig. 23: Construction of the simulator for adversarial verifier.