Quantum position verification in the random oracle model

Dominique Unruh
University of Tartu

February 12, 2014

Abstract. We present a quantum position verification scheme in the random oracle model. In contrast to prior work, our scheme does not require bounded storage/retrieval/entanglement assumptions. We also give an efficient position-based authentication protocol. This enables secret and authenticated communication with an entity that is only identified by its position in space.

Contents

1 Introduction

What is position verification? Consider the following setting: A device P wishes to access a location-based service. This service should only be available to devices in a certain spacial region $\mathbf{P}$, e.g., within a sports stadium. The service provider wants to be sure no malicious device outside $\mathbf{P}$ accesses the service. In other words, we need a protocol such that a prover P can prove to a verifier V that P is at certain location. Such a protocol is called a *position verification* (PV) scheme. A special case of position verification is *distance bounding*: P proves that he is within a distance δ of V. In its simplest form, this is done by V sending a random message r to P, and P has to send it back immediately. If r comes back to V in time t, P must be within

distance $tc/2$ where c is the speed of light. In general, however, it may not be practical to require a device V in the middle of a spherical region $\mathbf{P}$. (E.g., if $\mathbf{P}$ might be a rectangular room.) In general PV, thus, we assume several verifier devices $V_1, \ldots, V_n$, and a prover P somewhere in the convex hull of $V_1, \ldots, V_n$. The verifiers should then interact with P in such a way that based on the response times of P, they can make sure that P is at the claimed location (a kind of triangulation). Unfortunately, [CGMO09] showed that position verification based on *classical* cryptography cannot be secure, even when using computational assumptions, if the prover has several devices at different locations (collusion). [BCF+11] showed impossibility in the quantum setting, but only for information-theoretically secure protocols. Whether a protocol in the computational setting exists was left open.[1] In this work, we close this gap and give a simple protocol in the random oracle model.

Applications. The simplest application of PV is just for a device to prove that it is at a particular location to access a service. In a more advanced setting, location can be used for authentication: a prover can send a message which is guaranteed to have originated within a particular region (position-based authentication, PBA). Finally, when combining PBA with quantum key distribution (QKD), an encrypted message can be sent in such a way that only a recipient at a certain location can decrypt it. (E.g., think of sending a message to an embassy – you can make sure that it will be received only in the embassy, even if you do not know the embassy's public key.) More applications are position-based multi-party computation and position-based PKIs, see [CGMO09].

Our contribution. We present the first PV and PBA schemes secure against colluding provers that do not need bounded storage/retrieval/entanglement assumptions. (Cf. "related work" below.) Our protocols use quantum cryptography and are proven secure in the (quantum) random oracle model, and they work in the 3D setting. (Actually, in any number of dimensions, as well as in curved spacetime.[2]) Using [BCF+11], this also immediately implies position-based QKD. (And we even get *everlasting security*, i.e., if the adversary breaks the hash function *after* the protocol run, he cannot break the secrecy.)

We also introduce a methodology for analyzing quantum circuits in spacetime which we believe simplifies the rigorous analysis of protocols that are based on the speed of light (like, e.g., PV or relativistic commitments [Ken12, KTHW13]). And for the first time (to our knowledge), a security analysis uses adaptive programming of the quantum random oracle (in our PBA security proof).

[1] But both [CGMO09, BCF+11] give positive results assuming bounded retrieval/entanglement, see "related work" below.

[2] At the first glance, taking curvature of spacetime into account might seem like overkill. But for example GPS needs to take general relativity into account to ensure precise positioning (see, e.g., [Ash97]). There is no reason to assume that this would not be the case for long-distance PV.

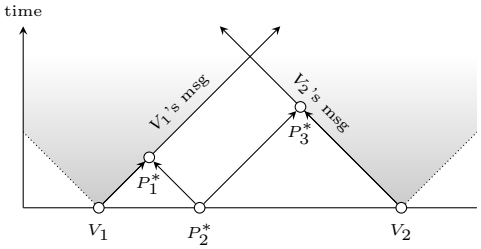


Figure 1: Message flow in [BCF+11, TFKW13]. Security is only guaranteed if no entanglement is created before the shaded region. The scheme can be attacked if P_2^* sends EPR pairs to P_1^*, P_3^* who then can execute the attack from [KMS11, Section I].

Related work. [CGMO09] showed a general impossibility of computationally secure PV in the classical setting; [BCF+11] showed the impossibility of information-theoretically secure PV in the quantum setting. [CGMO09] proposed computationally secure protocols for PV and position-based key exchange *in the bounded retrieval model*. Their model assumes that a party can only retrieve part of a large message reaching it. In particular, a party cannot forward a message ("reflection attacks" in the language of [CGMO09]); this may be difficult to ensure in practice because a mirror might be such a forwarding device. [BCF+11, TFKW13] provide a quantum protocol that is secure if the adversary can have no/limited entanglement before receiving the verifiers' messages. (I.e., in the message flow diagram Figure 1, only in the shaded areas.) In particular, using the message flow drawn in Figure 1, the attack from [KMS11, Section I] can be applied, even though no entanglement is created before the protocol start ($t = 0$) and no entanglement needs to be stored. This makes the assumption difficult to justify. Our protocol is an extension of theirs, essentially adding one hash function application. [BCF+11] also gives a generic transformation from PV to PBA; however, their construction is considerably less efficient than our specialized one and does not achieve concurrent security (see the discussion after Definition 9 below). Furthermore, the protocols from [BCF+11, TFKW13] only work in the one-dimensional setting. ([BCF+11] has a construction for the 3D case, but their proof seems incorrect, see Appendix A.)

Organization. In Section 2 we first explain our scheme in the 1D case. In Section 3.1 we explain the difficulties occurring in the 3D case which we solve in Sections 3.2 and 3.3. In Section 4 we present our PBA scheme. Section 5 discusses open problems. The appendix contains supplemental material referenced from the text.

1.1 Preliminaries

$\omega(x)$ denotes the Hamming weight of x. $h(p) = -p \log p - (1-p)\log(1-p)$ denotes the binary entropy. $|x|$ denotes the absolute value or cardinality of x. $\|x\|$ denotes the Euclidean norm. $x \xleftarrow{\$} M$ means x is uniformly random from M, and $x \leftarrow A()$ means x is chosen by algorithm A.

For a background in quantum mechanics, see [NC10]. But large parts of this paper should be comprehensible without detailed knowledge on quantum mechanics. For $x \in \{0,1\}^n$, $|x\rangle$ denotes the quantum state x encoded in the computational basis, and $|\Psi\rangle$ denotes arbitrary quantum states (not necessarily in the computational basis). $\langle\Psi|$ is the conjugate transpose of $|\Psi\rangle$. For $B \in \{0,1\}^n$, $|x\rangle_B$ denotes x encoded in the bases specified by B, more precisely $|x\rangle_B = H^{B_1}|x_1\rangle \otimes \cdots \otimes H^{B_n}|x_n\rangle$ where H is the Hadamard matrix. An *EPR pair* has state $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$. $\mathrm{TD}(\rho,\rho')$ denotes the trace distance between states ρ,ρ'. Given a (quantum) oracle algorithm A and a function H, $A^H()$ means that A has oracle access to H and can query H on different inputs in superposition. This is important for modeling the quantum random oracle correctly [BDF+11].

2 1D position verification

In this section, we consider the case of one-dimensional PV only. That is, all verifiers and the honest and malicious provers live on a line. Although this is an unrealistic setting, it allows us to introduce our construction and proof technique in a simpler setting without having to consider the additional subtleties arising from the geometry of intersecting light cones. We also suggest the content of this section for teaching.

We assume the following specific setting: There are two verifiers V_1 and V_2 at positions -1 and 1, and an honest prover P at position 0. The verifiers will send messages at time $t = 0$ to the prover P, who receives them at time $t = 1$ (i.e., we assume units in which the speed of light is $c = 1$), and his immediate response reaches the verifiers at time $t = 2$. In an attack, we assume that the malicious prover has devices P_1^* and P_2^* left and right of position 0, but no device at position 0 where the honest prover is located. See Figure 2 for a depiction of all message flows in this setting. This setting simplifies notation and is sufficient to show all techniques needed in the 1D case. The general 1D case (P not exactly in the middle, more malicious provers, not requiring P's responses to be instantaneous) will be a special case of the higher dimensional theorems in Section 3.3.

In this setting, we use the following PV scheme:

Definition 1 (1D position verification) *Let n (number of qubits) and ℓ (bit length of classical challenges) be integers, $0 \le \gamma < 1/2$ (fraction of allowed errors). Let $H : \{0,1\}^\ell \to \{0,1\}^n$ be a hash function (modeled as a quantum random oracle).*

- *Before time $t = 0$, verifier V_1 picks uniform $x_1, x_2 \in \{0,1\}^\ell$, $\hat{y} \in \{0,1\}^n$ and forwards x_2 to V_2 over a secure channel.*
- *At time $t = 0$, V_1 sends $|\Psi\rangle$ and x_1 to P. Here $B := H(x_1 \oplus x_2)$, $|\Psi\rangle := |\hat{y}\rangle_B$. And V_2 sends x_2 to P.*
- *At time $t = 1$, P receives $|\Psi\rangle, x_1, x_2$, computes $B := H(x_1 \oplus x_2)$, measures $|\Psi\rangle$ in basis B to obtain outcome y_1, and sends y_1 to V_1 and $y_2 := y_1$ to V_2. (We assume all these actions are instantaneous, so P sends y_1, y_2 at time $t = 1$.)*
- *At time $t = 2$, V_1 and V_2 receive y_1, y_2. Using secure channels, they check whether $y_1 = y_2$ and $\omega(y_1 - \hat{y}) \le \gamma n$. If so (and y_1, y_2 arrived in time), they accept.*

We can now prove security in our simplified setting.

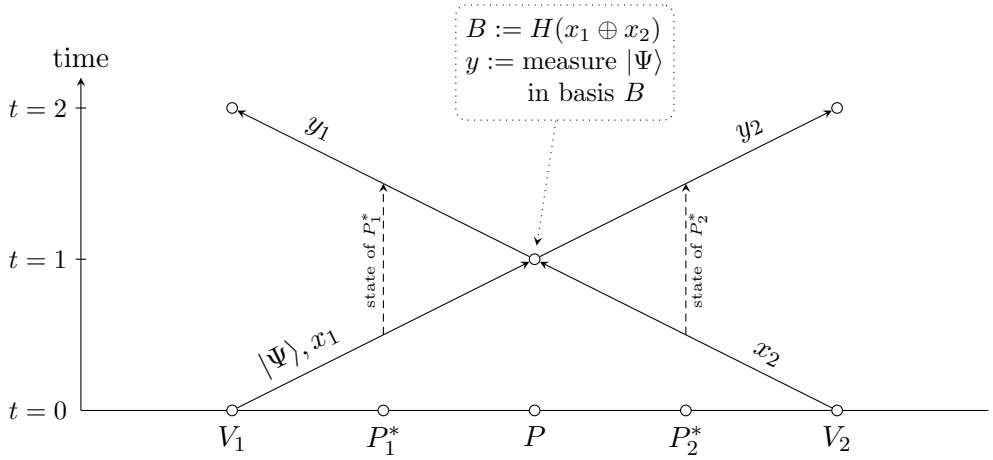


Figure 2: One-dimensional PV protocol. Dotted lines indicate additional message flows of the adversary P_1^*, P_2^*.

Theorem 2 (1D position verification) *Assume P_1^* and P_2^* perform at most q queries to H. Then in an execution of V_1, V_2, P_1^*, P_2^* with V_1, V_2 following the protocol from Definition 1, the probability that V_1, V_2 accept is at most*[3]

$$2q2^{-\ell/2} + \left(2^{h(\gamma)}\frac{1+\sqrt{1/2}}{2}\right)^n.$$

Proof. To prove this theorem, we proceed using a sequence of games. The first game is the original protocol execution, and in the last game, we will be able to show that $\Pr[\mathsf{Accept}]$ is small. Here we abbreviate the event "$y_1 = y_2$ and $\omega(y_1 - \hat{y}) \leq \gamma n$" as "$\mathsf{Accept}$".

Game 1 *An execution as described in Theorem 2.*

As a first step, we use EPR pairs to delay the choice of the basis B. This is a standard trick that has been used in QKD proofs and other settings. By choosing B sufficiently late, we will be able to argue below that B is independent of the state of P_1^* and P_2^*.

Game 2 *As in Game 1, except that V_1 prepares n EPR pairs, with their first qubits in register X and their second qubits in Y. Then V_1 sends X at time $t = 0$ instead of sending $|\Psi\rangle$. At time $t = 2$, V_1 measures Y in basis $B := H(x_1 \oplus x_2)$, the outcome is $\hat{y}$.*

Note in particular that V_1, V_2 never query H before time $t = 2$. (But P_1^*, P_2^* might, of course.)

It is easy to verify (and well-known) that for any $B \in \{0,1\}$, preparing a qubit $X := |y\rangle_B$ for random $y \in \{0,1\}$ is perfectly indistinguishable (when given X, y, B) from producing an EPR pair XY, and then measuring Y in bases B to get outcome y. Thus $\Pr[\mathsf{Accept} : \text{Game 1}] = \Pr[\mathsf{Accept} : \text{Game 2}]$.

[3]This probability is negligible if $\gamma \leq 0.037$ and n, ℓ are superlogarithmic.

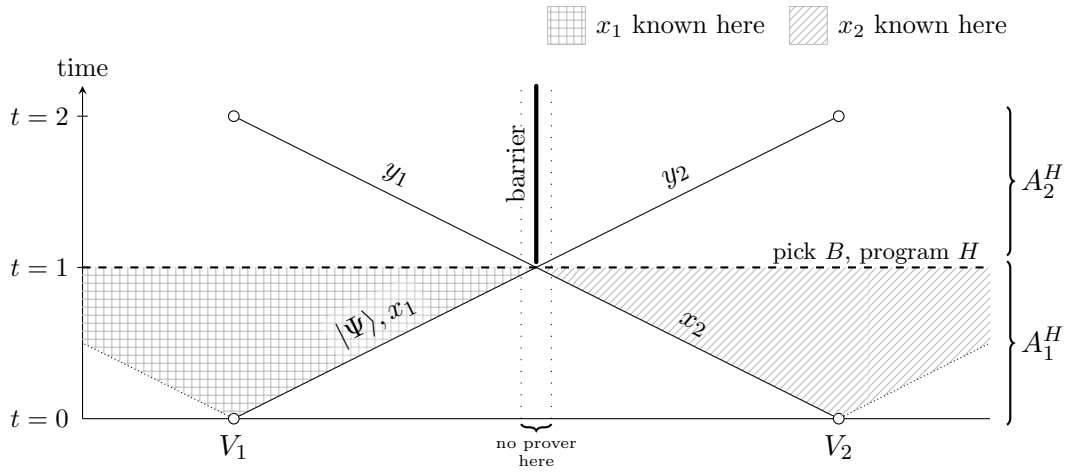


Figure 3: Spacetime diagram depicting various steps of the proof of Theorem 2.

The problem now is that, although we have delayed the time when the basis B is used, the basis is still chosen early: At time $t = 0$, the values x_1, x_2 are chosen, and those determine B via $B = H(x_1 \oplus x_2)$. We have that neither P_1^* nor P_2^* individually knows B, but that does not necessarily exclude an attack. (For example, [KMS11, Section I] gives an efficient attack for the case that H is the identity, even though in this case B would still not be known to P_1^* nor P_2^* individually before time $t = 1$.) We can only hope that H is a sufficiently complex function such that computationally, B is "as good as unknown" before time $t = 1$ (where x_1 and x_2 become known to both P_1^*, P_2^*). The next game transformation formalizes this:

Game 3 *As in Game 1, except that at time $t = 1$, the value $B \xleftarrow{\$} \{0,1\}^n$ is chosen, and the random oracle is reprogrammed to return $H(x_1 \oplus x_2) = B$ after $t = 1$.*

To clarify this, if $H_0 : \{0,1\}^\ell \to \{0,1\}^n$ denotes a random function chosen at the very beginning of the execution, then at time $t \leq 1$, $H(x) = H_0(x)$ for all $x \in \{0,1\}^\ell$, while at time $t > 1$, $H(x_0 \oplus x_1) = B$ and $H(x) = H_0(x)$ for all $x \neq x_0 \oplus x_1$.

Intuitively, the change between Games 2 and 3 cannot be noticed because before time $t = 1$, the verifiers never query $H(x_1 \oplus x_2)$, and the provers cannot query $H(x_1 \oplus x_2)$ either: before time t, in no spacial location the prover will have access to both x_1 and x_2.

This is illustrated in Figure 3: The hatched areas represent where x_1 and x_2 are known respectively. Note that they do not overlap. The dashed horizontal line represents where the random oracle is programmed ($t = 1$).

Purists may object that choosing B and programming the random oracle to return B at all locations in a single instant in time needs superluminal communication which in turn is know to violate causality and might thus lead to inconsistent reasoning. Readers worried about this aspect should wait until we prove the general case of the PV protocol in Section 3.3, there this issue will not arise because we first transform the whole protocol

execution into a non-relativistic quantum circuit and perform the programming of the random oracle in that circuit.

To prove that Games 2 and 3 are indistinguishable, we use the following lemma which is a special case of Lemma 15 (Appendix B).

Lemma 3 *Let $H : \{0,1\}^\ell \to \{0,1\}^n$ be a random oracle. Let (A_1, A_2) be oracle algorithms sharing state between invocations that perform at most q queries to H. Let C_1 be an oracle algorithm that on input (j, x) does the following: Run $A_1^H(x)$ till the j-th query to H, then measure the argument of that query in the computational basis, and output the measurement outcome. (Or $\perp$ if no j-th query occurs.) Let*

$$P_A^1 := \Pr[b' = 1 : H \xleftarrow{\$} (\{0,1\}^\ell \to \{0,1\}^n), x \leftarrow \{0,1\}^\ell, A_1^H(x), b' \leftarrow A_2^H(x, H(x))]$$

$$P_A^2 := \Pr[b' = 1 : H \xleftarrow{\$} (\{0,1\}^\ell \to \{0,1\}^n), x \leftarrow \{0,1\}^\ell, B \xleftarrow{\$} \{0,1\}^n, A_1^H(x), H(x) := B, b' \leftarrow A_2^H(x, B)]$$

$$P_C := \Pr[x = x' : H \xleftarrow{\$} (\{0,1\}^\ell \to \{0,1\}^n), x \leftarrow \{0,1\}^\ell, j \xleftarrow{\$} \{1, \dots, q\}, x' \leftarrow C_1^H(j, x)]$$

Then $|P_A^1 - P_A^2| \leq 2q\sqrt{P_C}$.

In other words, an adversary can only notice that the random oracle is reprogrammed at position x if he can guess x before the reprogramming takes place.

To apply Lemma 3 to Games 2 and 3, let $A_1^H(x)$ be the machine that executes verifiers and provers from Game 2 until time $t = 1$ (inclusive). When V_1 chooses x_1, x_2, $A_1^H(x)$ chooses $x_1 \xleftarrow{\$} \{0,1\}^\ell$ and $x_2 := x \oplus x_1$. And let $A_2^H(x, B)$ be the machine that executes verifiers and provers after time $t = 1$. When V_1 queries $H(x_1 \oplus x_2)$, A_2^H uses the value B instead. In the end, A_2^H returns 1 iff $y_1 = y_2$ and $\omega(\hat{y} - y_1) \leq \gamma n$. (See Figure 3 for the time intervals handled by A_1^H,A_2^H.) Since V_1, V_2 make no oracle queries except for $H(x_1 \oplus x_2)$, and since P_1^*, P_2^* make at most q oracle queries, we have that A_1^H, A_2^H perform at most q queries.

By construction, $P_A^1 = \Pr[\mathsf{Accept} : \text{Game 2}]$. And $P_A^2 = \Pr[\mathsf{Accept} : \text{Game 3}]$. And $P_C = \Pr[x' = x_1 \oplus x_2 : \text{Game 4}]$ for the following game:

Game 4 *Pick $j \xleftarrow{\$} \{1, \dots, q\}$. Then execute Game 2 till time $t = 1$ (inclusive), but stop at the j-th query and measure the query register. Call the outcome x'.*

Since Game 4 executes only till time $t = 1$, and since till time $t = 1$, no gate can be reached by both x_1, x_2 (note: at time $t = 1$, at position 0 both x_1, x_2 could be known, but no malicious prover may be at that location), the probability that $x_1 \oplus x_2$ will be guessed is bounded by $2^{-\ell}$. Hence $\Pr[x' = x_1 \oplus x_2 : \text{Game 3}] \leq 2^{-\ell}$. (This argument was a bit nonrigorous; we will be more precise in the proof of the generic case, in the proof of Theorem 6.)

Thus by Lemma 3, we have

$$\begin{aligned}\big|\Pr[\mathsf{Accept} : \text{Game 2}] - \Pr[\mathsf{Accept} : \text{Game 3}]\big| &= |P_A^1 - P_A^2| \leq 2q\sqrt{P_C} \\ &= 2q\sqrt{\Pr[x' = x_1 \oplus x_2 : \text{Game 4}]} \leq 2q2^{-\ell/2}. \quad (1)\end{aligned}$$

We continue to modify Game 3.

Game 5 *Like Game 3, except that for time $t > 1$, we install a barrier at position 0 (i.e., where the honest prover P would be) that lets no information through.*

The barrier is illustrated in Figure 3 with a thick vertical line.

Time $t = 1$ is latest time at which information from position 0 could reach the verifiers V_1, V_2 at time $t \leq 2$. Since we install the barrier only for time $t > 1$, whether the barrier is there or not cannot influence the measurements of V_1, V_2 at time $t = 2$. And Accept only depends on these measurements. Thus $\Pr[\mathsf{Accept} : \textit{Game } 3] = \Pr[\mathsf{Accept} : \textit{Game } 5]$.

Let ρ be the state of the execution of Game 5 directly after time $t = 1$ (i.e., after the gates at times $t \leq 1$ have been executed). Then ρ is a threepartite state consisting of registers Y, L, R where Y is the register containing the EPR qubits which will be measured to give $\hat{y}$ (cf. Game 2), and L and R are the quantum state left and right of the barrier respectively. Then $\hat{y}$ is the result of measuring Y in basis B, and y_1 is the result of applying some measurement M_1 to L (consisting of all the gates left of the barrier), and y_2 is the result of applying some measurement M_2 to R. Notice that due to the barrier, M_1 and M_2 operate only on L and R, respectively, without interaction between those two.

We have thus:

$$\begin{aligned}\Pr[\mathsf{Accept} : \text{Game } 5] = \Pr[y_1 = y_2 \text{ and } \omega(\hat{y} - y_1) \leq \gamma n : B \xleftarrow{\$} \{0,1\}^n, YLR \leftarrow \rho,\\ \hat{y} \leftarrow M^B(Y), y_1 \leftarrow M_1(L), y_2 \leftarrow M_2(R)]\end{aligned}$$

where $YLR \leftarrow \rho$ means initializing YLR with state ρ. And M^B is a measurement in bases B. And $\hat{y} \leftarrow M^B(Y)$ means measuring register Y using measurement M^B and assigning the result to $\hat{y}$. And $y_1 \leftarrow M_1(L), y_2 \leftarrow M_2(R)$ analogously.

The rhs of this equation is a so-called monogamy of entanglement game, and [TFKW13] shows that the rhs is bounded by $\left(2^{h(\gamma)} \frac{1+\sqrt{1/2}}{2}\right)^n$. Thus $\Pr[\mathsf{Accept} : \text{Game } 5] \leq \left(2^{h(\gamma)} \frac{1+\sqrt{1/2}}{2}\right)^n$. And from (1) and the equalities between games, we have $\left|\Pr[\mathsf{Accept} : \text{Game } 1] - \Pr[\mathsf{Accept} : \text{Game } 5]\right| \leq 2q2^{-\ell/2}$.

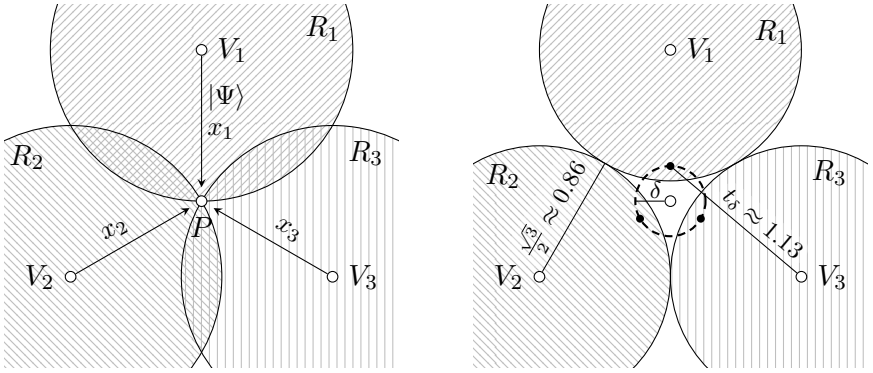
Thus altogether $\Pr[\mathsf{Accept} : \text{Game } 1] \leq 2q2^{-\ell/2} + \left(2^{h(\gamma)} \frac{1+\sqrt{1/2}}{2}\right)^n$. □

3 Position verification in higher dimensions

3.1 Difficulties

Excepting special cases where the honest prover happens to lie on a line between two verifiers, one-dimensional PV with two verifiers is not very useful. We therefore need to generalize the approach to three dimensions. It turns out that some non-trivialities occur here. (See also Appendix A for 3D-problems in prior work.) For n-dimensional PV we need at least $n + 1$ verifiers.[4] To illustrate the problems occurring in the higher

[4]PV (in Euclidean space) can only work if the prover P is in the convex hull C of the verifiers. Otherwise, if we project P onto the hypersurface H separating C from P, we get a point P' that is closer



(a) (b)

Figure 4: The geometry of space at time t_δ (i.e., when B first becomes known). Left for $\delta = 0$, right for $\delta = \sqrt{4 - \sqrt{12}} - \frac{1}{2} \approx 0.23$.

dimensional case, we sketch what happens if we try to generalize the protocol and proof from Section 2 to the 2D case.

In the 2D case we need at least three verifiers V_1, V_2, V_3. Let's assume that they are arranged in a equilateral triangle, each at distance 1 from an honest prover P in the center. (Cf. Figure 4 (a).) V_1 sends a quantum state $|\Psi\rangle$, and all V_i send a random x_i. At time $t = 1$, all x_i are received by P who computes $B := H(x_1 \oplus x_2 \oplus x_3)$ and measures $|\Psi\rangle$ in basis B, yielding the value y to be sent to V_1, V_2, V_3.

Now as in Section 2 we can argue that before time $t = 1$, there is no point in space where all x_1, x_2, x_3 are known. Hence $B := H(x_1 \oplus x_2 \oplus x_3)$ will not be queried before $t = 1$. Hence by programming the random oracle (using Lemma 3) we can assume that the basis B is chosen randomly only at time $t = 1$. In Section 2 we then observed that space is partitioned into two disjoint regions: Region L from which light can reach V_1 by time $t = 2$, and region R from which light can reach V_2 by time $t = 2$. The results from [TFKW13] then imply that the correct y cannot be obtained from two independent (but possibly entangled) quantum registers L and R simultaneously. What happens if we apply this reasoning in the 2D case? Figure 4 (a) depicts the three regions R_1, R_2, R_3 of points that can reach V_1, V_2, V_3 until time $t = 2$. These regions are not disjoint! We cannot argue that measuring y in each of these regions violates the monogamy of entanglement, y does not result from measuring separate quantum registers.

Can we fix this? The most obvious consequence would be to weaken the security claim: "A malicious prover which has devices anywhere except at point P *or distance* δ *from* P cannot make the verifiers accept." Then the time t_δ when the random oracle is programmed is the earliest time at which some point at distance δ from P has access to all x_1, x_2, x_3. We can see that this time is $t_\delta = \sqrt{3/4 + (1/2 + \delta)^2}$.[5] Then R_1, R_2, R_3

to any point of C than P. Since the convex hull of n provers can at most be $n - 1$ dimensional, we need at least $n + 1$ provers to get an n dimensional convex hull.

[5]To see this, first observe that a point X that gets x_1, x_2, x_3 first must lie on the circle C around P

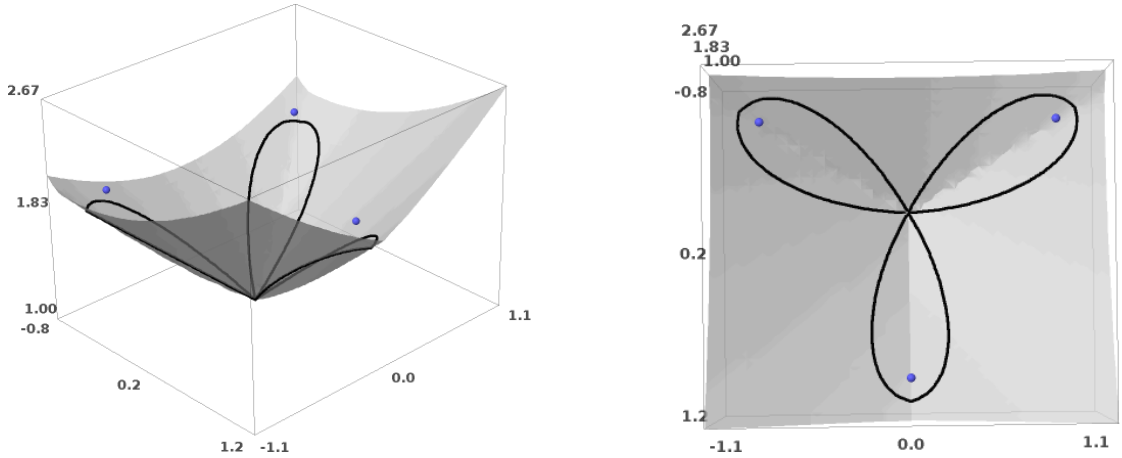


Figure 5: The surface S in spacetime at which B is sampled. The dots floating over S denote when the verifiers need to receive y (i.e., the dots are at time 2 and space V_1, V_2, V_3). The thick black lines enclose the areas R_1, R_2, R_3 on S from which the verifiers can be reached in time. (Right: top view. In PDF: click figures for interaction.)

are the regions from which light can travel to V_1, V_2, V_3 within time $2 - t_\delta$. In order for them to be disjoint, we thus need $2 - t_\delta < a/2$ where $a = \sqrt{3}$ is the distance between two verifiers. This is achieved iff $\delta > \sqrt{4 - \sqrt{12}} - \frac{1}{2} \approx 0.23$. (Cf. Figure 4 (b).) This means that the malicious prover is only guaranteed to be within a circle of diameter 2δ, which is about 46% of the distance between prover and verifier. In the 3D case, using a numerical calculation, we even get $\delta \approx 0.38$.

Can we improve on this bound? Indeed, when we said that the B is sampled at time $t = 1$, this was not a tight analysis. At time $t = 1$, the query $B = H(x_1 \oplus x_2 \oplus x_3)$ can only occur at point P. The farther away from P we get, the later we get all of x_1, x_2, x_3. Thus, if we plot the earliest time of querying B as a function of space, we get a surface S in 3D spacetime (Figure 5) which is not a plane. Now, instead of considering the state of the provers at time $t = 1$, we consider the state of the prover on S. (I.e., the state of all devices of the prover at points in spacetime in S.) We ask the reader to take it on trust for the moment this is actually a well-defined state. And now we can again ask whether S decomposes into distinct regions R_1, R_2, R_3 if we consider regions that can reach the verifiers V_1, V_2, V_3 by time $t = 2$. (See Figure 5.) This approach has the potential of giving a much tighter security analysis. However, it is quite complicated to reason about the geometry of S and R_1, R_2, R_3, and in the 3D case things will get even more complicated. Therefore in the following section we will take an approach that abstracts away from the precise geometry of spacetime and uses a more generic reasoning. This has the twofold advantage that we do not need to analyze what S actually looks

with radius δ. And X must have equal distance to at least two of the verifiers. Thus X is a point on the altitude of the triangle. There are six intersections between C and the altitudes. Those that minimize the distance to the farthest vertex are the ones closest to a vertex and have distance $\sqrt{(a/2)^2 + (r + \delta)^2}$ from the farthest vertex ($a = \sqrt{3}$ is the side length of the triangle, and $r = 1/2$ is the inradius).

like (although S implicitly occurs in the proof), and that our result will be much more general: it holds in any number of dimensions, and it even holds if we consider curved spacetime (general relativity theory). To state and prove our results, we first need to introduce some (simple) notation from general relativity theory.

3.2 Circuits in spacetime

Spacetime is the set of all locations in space and time. That is, intuitively spacetime consists of all tuples $(t, x_1, \ldots, x_n)$ where t is the time and $x_1, \ldots, x_n$ is the position in space. Such a location in spacetime is called an *event*. Relativity theory predicts that there is no natural distinction between the time coordinate t and the space coordinates $x_1, \ldots, x_n$. (In a similar way as in "normal" space there is no reason why three particular directions in space are coordinates.) As it turns out, for analyzing our PV protocol, we do not need to know the structure of spacetime, so in the following spacetime will just be some set of events, with no particular structure.[6] However, the reader may of course assume throughout the paper that spacetime consists of events $(t, x_1, \ldots, x_n)$ with $t, x_1, \ldots, x_n \in \mathbb{R}$. This is called *flat spacetime*.

The geometry of spacetime (to the extent needed here) is described by a partial order on the events: We say x *causally precedes* y ($x \prec y$) iff information originating from event x can reach event y. Or in other words, if you can get from x to y traveling at most the speed of light. In flat spacetime, this relation is familiar: $(t_x, x_1, \ldots, x_n) \prec (t_y, y_1, \ldots, y_n)$ iff $t_x \leq t_y$ and $\|(x_1, \ldots, x_n) - (y_1, \ldots, y_n)\| \leq t_y - t_x$.

Given this relation, we can define the *causal future* $C^+(x)$ of an event x as the set of all events reachable from x, $C^+(x) := \{y : x \prec y\}$. Similarly, we define the *causal past* $C^-(x) := \{y : y \prec x\}$.

In the case of flat spacetime, the causal future of $x = (t, x_1, \ldots, x_n)$ is an infinite cone with its point at x and extending towards the future. Thus it is also called a future *light cone*. Similarly the causal past of x is an infinite cone with its point at x extending into the past.

This language allows us to express quantum computations in space that do not transfer information faster than light. A *spacetime circuit* is a quantum circuit where every gate is at a particular event. There can only be a wire from a gate at event x to a gate at event y if x causally precedes y ($x \prec y$). Note that since $\prec$ is a partial order and thus antisymmetric, this ensures that a circuit cannot be cyclic. Note further that there is no limit to how much computation can be performed in an instant since $\prec$ is reflexive. We can model malicious provers that are not at the location of an honest prover by considering circuits with no gates in $\mathbf{P}$, where $\mathbf{P}$ is a region in spacetime. (This allows for more finegrained specifications than, e.g., just saying that the malicious prover is not within δ distance of the honest prover. For example, $\mathbf{P}$ might only consist of events within a certain time interval; this means that the malicious prover is allowed to be at

[6] For readers knowledgeable in general relativity: We do assume that spacetime is a Lorentzian manifold which is time-orientable (otherwise the notions of causal future/past would not make send) without closed causal curves (at least in the spacetime region where the protocol is executed; otherwise quantum circuits may end up having loops).

any space location outside that time interval.) Notice that a spacetime circuit is also just a normal quantum circuit if we forget where in spacetime gates are located. Thus transformations on quantum circuits (such as changing the execution order of commuting gates) can also be applied to spacetime circuits, the result will be a valid circuit, though possibly not a spacetime circuit any more.

3.3 Achieving higher-dimensional position verification

We can now formulate the definition of secure PV in higher dimensions using the language from the previous section.

Definition 4 (Sound position verification) *Let* $\mathbf{P}$ *be a region in spacetime. A position verification protocol is* sound for $\mathbf{P}$ *iff for any non-uniform polynomial-time*[7] *spacetime circuit* P^* *that has no gates in* $\mathbf{P}$*, the following holds: In an interaction between the verifiers and* P^**, the probability that the verifiers accept (the* soundness error*) is negligible.*

The smaller the region $\mathbf{P}$ is, the better the protocol localizes the prover. Informally, we say the protocol has higher *precision* if $\mathbf{P}$ is smaller.

Next, we describe the generalization of the protocol in Section 2. In this generalization, only two of the verifiers check whether the answers of the prover are correct. Although we believe that we get higher precision if more verifiers check the answers, it is an open problem to prove that.

Definition 5 (Position verification protocol) *Let* P *be a prover, and* P° *an event in spacetime (*P° *specifies where and when the honest prover performs its computation). Let* $V_1, \dots, V_r$ *be verifiers. Let* $V_1^+, \dots, V_r^+$ *be events in spacetime that causally precede* P°*. (*V_i^+ *specifies where and when the verifier* V_i *sends its challenge.) Let* V_1^-, V_2^- *be events in spacetime such that* P° *causally precedes* V_1^-, V_2^-*. (*V_i^- *specifies where and when* V_i *expects the prover's response.)*

Let n *(number of qubits) and* ℓ *(bit length of classical challenges) be integers, and* $0 \leq \gamma < \frac{1}{2}$ *(fraction of allowed errors). Let* $H : \{0,1\}^\ell \to \{0,1\}^n$ *be a hash function (modeled as a quantum random oracle).*

- *The verifiers choose uniform* $x_1, \dots, x_r \in \{0,1\}^\ell$*,* $\hat{y} \in \{0,1\}^n$*. (By communicating over secure channels.)*
- *At some event that causally precedes* P°*,* V_0 *sends* $|\Psi\rangle$ *to* P*. Here* $B := H(x_1 \oplus x_2)$*,* $|\Psi\rangle := |\hat{y}\rangle_B$*.*
- *For* $i = 1, \dots, r$*:* V_r *sends* x_r *to* P *at event* V_r^+*.*
- *At event* P°*,* P *will have* $|\Psi\rangle, x_1, \dots, x_r$*. Then* P *computes* $B := H(x_1 \oplus \cdots \oplus x_r)$*, measures* $|\Psi\rangle$ *in basis* B *to obtain outcome* y_1*, and sends* y_1 *to* V_1 *and* $y_2 := y_1$ *to* V_2*.*

[7] Non-uniform polynomial-time means that we are actually considering a family of circuits of polynomial size in the security parameter, consisting only of standard gates (from some fixed universal set) and oracle query gates. In addition, we assume that the circuit is given an (arbitrary) initial quantum state that does not need to be efficiently computable.

- *At events V_1^-, V_2^-, V_1 and V_2 receive y_1, y_2. Using secure channels, the verifiers check whether $y_1 = y_2$ and $\omega(y_1 - \hat{y}) \leq \gamma n$. If so (and y_1, y_2 indeed arrived at V_1^-, V_2^-), the verifiers accept.*

In the protocol description, for simplicity we assume that V_1, V_2 are the receiving verifiers. However, there is no reason not to choose other two verifiers, or even additional verifiers not used for sending. Similarly, $|\Psi\rangle$ could be sent by any verifier, or by an additional verifier. In the analysis, we only use the events at which different messages are sent/received, not which verifier device sends which message.

Note that this protocol also allows for realistic provers that cannot perform instantaneous computations: In this case, one chooses the events V_1^-, V_2^- such that the prover's messages can still reach them even if the prover sends y_1, y_2 with some delay.

We can now state the main security result:

Theorem 6 *Assume that $\gamma \leq 0.037$ and n, ℓ are superlogarithmic.*

Then the PV protocol from Definition 5 is sound for $\mathbf{P} := \bigcap_{i=1}^r C^+(V_i^+) \cap C^-(V_1^-) \cap C^-(V_2^-)$. (In words: There is no event in spacetime outside of $\mathbf{P}$ at which one can receive the messages x_i from all V_i, and send messages that will be received in time by V_1, V_2.)

Concretely, if the malicious prover performs at most q oracle queries,[8] *then the soundness error is at most $\nu := \left(2^{h(\gamma)} \frac{1+\sqrt{1/2}}{2}\right)^n + 2q2^{-\ell/2}$.*

Notice that the condition on the locations of the provers is tight: If $E \in \bigcap_{i=1}^r C^+(V_i^+) \cap C^-(V_1^-) \cap C^-(V_2^-) \setminus \mathbf{P} \neq \varnothing$, then the protocol could even be broken by a malicious prover with a single device: P^* could be at event E, receive $x_1, \ldots, x_r$, compute y_1, y_2 honestly, and send them to V_1, V_2 in time. The same reasoning applies to any protocol where only two verifiers receive. Our protocol is thus optimal in terms of precision under all such protocols.

Proof of Theorem 6. In the following, we write short C_i^+ for $C^+(V_i^+)$ and C_i^- for $C^-(V_i^-)$. We also write $\bigcap$ instead of $\bigcap_{i=1}^r$. The precondition of the theorem then becomes: $\bigcap C_i^+ \cap C_1^- \cap C_2^- \subseteq \mathbf{P}$. Let Ω denote all of spacetime.

We now partition the gates in the spacetime circuit P^* into several disjoint sets of gates (subcircuits), depending on where they are located in spacetime. For each subcircuit, we also give an rough intuitive meaning; those meanings are not precisely what the subcircuits do but help to guide the intuition in the proof.

Subcircuit	Region in spacetime	Intuition
P^*_{pre}	$(C_1^- \cup C_2^-) \setminus \bigcap C_i^+$	Precomputation
$P^*_{\mathbf{P}}$	$\bigcap C_i^+ \cap C_1^- \cap C_2^-$	Gates in $\mathbf{P}$ (empty)
P^*_1	$\bigcap C_i^+ \cap C_1^- \setminus C_2^-$	Computing y_1
P^*_2	$\bigcap C_i^+ \cap C_2^- \setminus C_1^-$	Computing y_2
P^*_{post}	$\Omega \setminus C_1^- \setminus C_2^-$	After protocol end

[8] Actually, it is sufficient if the number of queries performed by gates inside $C_1^- \cup C_2^-$ is bounded by q. In particular, oracle queries after both verifiers have received y_1, y_2 do not count (as expected).

Note that all those subcircuits are disjoint, and their union is all of Ω. The subcircuits have analogues in the proof in the one-dimensional case. P^*_{pre} corresponds to the gates below the dashed line in Figure 3; P^*_1 to the gates above the dashed line and left of the barrier; P^*_2 above the dashed line and right of the barrier; P^*_{post} to everything that is above the picture. This correspondance is not exact, because as discussed in Section 3.1, the dashed line needs to be replaced by a surface S (Figure 5) which is not flat. In our present notation, S is the border between P^*_{pre} and the other subcircuits.

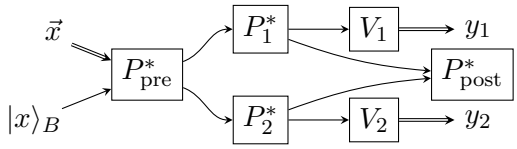
In addition, in some abuse of notation, by V_1 we denote the circuit at V_1^- that receives y_1. Similar for V_2.

By definition of spacetime circuits, there can only be a wire from gate G_1 to gate G_2 if G_1, G_2 are at events E_1, E_2 with $E_1 \prec E_2$ (E_1 causally precedes E_2). Thus, by definition of causal futures and the transitivity of $\prec$, there can be no wire leaving C_i^+. Similarly, there can be no wire entering C_i^-. These two facts are sufficient to check the following facts:

$$P^*_1, P^*_2, P^*_{\text{post}} \not\twoheadrightarrow P^*_{\text{pre}}, \quad P^*_1 \not\twoheadrightarrow P^*_2, \quad P^*_2 \not\twoheadrightarrow P^*_1, \quad P^*_1 \not\twoheadrightarrow V_2, \quad P^*_2 \not\twoheadrightarrow V_1, \quad P^*_{\text{post}} \not\twoheadrightarrow P^*_1, P^*_2, V_1, V_2. \tag{2}$$

Here $A \not\twoheadrightarrow B$ means that there is no wire from subcircuit A to subcircuit B.

Given these subcircuits, we can write the execution of the protocol as the following quantum circuit:



(3)

Here $\vec{x}$ is short for $x_1, \ldots, x_r$. And we have omitted wires between subcircuits that are in the transitive hull of the wires drawn. (E.g., there can be a wire from P^*_{pre} to V_1, but we did not draw it because we drew wires from P^*_{pre} to P^*_1 to V_1.) Note that $P^*_{\mathbf{P}}$ does not occur in this circuit, because it contains no gates (it consists of gates in $\bigcap C_i^+ \cap C_1^- \cap C_2^- = \mathbf{P}$ which by assumption contains no gates).

From (2) it follows that no wires are missing in (3). In particular, (2) implies that the quantum circuit is well-defined. If we did not have, e.g., $P^*_1 \not\twoheadrightarrow P^*_{\text{pre}}$, there might be wires between P^*_1 and P^*_{pre} in both directions; the result would not be a quantum circuit. We added arrow heads in (2), these are only to stress that the wires indeed go in the right directions, below we will follow the usual left-to-right convention in quantum circuits and omit the arrow heads.

The circuit (3) now encodes all information dependencies that we will need, we can forget that (3) is a spacetime circuit and treat it as a normal quantum circuit.

We now proceed to analyze the protocol execution using a sequence of games. The original execution can be written as follows:

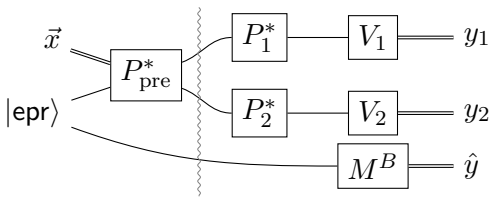
Game 1 (Protocol execution) *Pick* $x_1, \ldots, x_r \xleftarrow{\$} \{0,1\}^\ell$, $\hat{y} \xleftarrow{\$} \{0,1\}^n$, $H \xleftarrow{\$}$ *Fun where Fun is the set of functions* $\{0,1\}^\ell \to \{0,1\}^n$. *Let* $B := H(x_1 \oplus \cdots \oplus x_r)$. *Execute circuit* (3) *resulting in* y_1, y_2. *Let* $\mathsf{accept} := 1$ *iff* $y_1 = y_2$ *and* $\omega(y_1 - \hat{y}) \leq \gamma n$.

To prove the theorem, we need to show that $\Pr[\mathsf{accept} = 1 : \text{Game 1}] \leq \nu$.

As in the proof of the 1D case, we now delay the choice of $\vec{x}$ by using EPR pairs. And we remove the subcircuit P^*_{post} which clearly has no effect on the outputs y_1, y_2.

Game 2 (Using EPR pairs) *Pick* $x_1, \dots, x_r \xleftarrow{\$} \{0,1\}^\ell$, $H \xleftarrow{\$}$ *Fun. Let* $B := H(x_1 \oplus \cdots \oplus x_r)$. *Execute circuit* (4) *resulting in* y_1, y_2.

Let $\mathsf{accept} := 1$ *iff* $y_1 = y_2$ *and* $\omega(y_1 - \hat{y}) \leq \gamma n$.



(4)

Here $|\mathsf{epr}\rangle$ is the state consisting of n EPR pairs, i.e., $|\mathsf{epr}\rangle = 2^{-n/2} \sum_{x \in \{0,1\}^n} |x\rangle \otimes |x\rangle$. The top and bottom wire originating from $|\mathsf{epr}\rangle$ represent the first and last n qubits, respectively. And M^B is the gate that measures n qubits in bases $B \in \{0,1\}^n$. The wiggly line can be ignored for now.

As in the 1D case, we use that preparing a qubit $X := |y\rangle_B$ for random $y \in \{0,1\}$ is perfectly indistinguishable (when given X, y, B) from producing an EPR pair XY, and then measuring Y to get outcome y. Thus $\Pr[\mathsf{accept} = 1 : \text{Game 1}] = \Pr[\mathsf{accept} = 1 : \text{Game 2}]$.

Again like in the 1D case, we will now reprogram the random oracle. That is, instead of computing $B := H(x_1 \oplus \cdots \oplus x_r)$, we pick $B \xleftarrow{\$} \{0,1\}^n$ at some point in the execution and then program the random oracle via $H(x_1 \oplus \cdots \oplus x_r) := B$. The question is: at which point shall we program the random oracle? In the 1D case, we used the fact that before time $t = 1$ (dashed line in Figure 3), there is no event at which both x_1 and x_2 are known. An analogous reasoning can be done in the present setting: since P^*_{pre} consists only of gates outside $\bigcap C_i^+$, it means that any gate in P^*_{pre} is outside some C_i^+ and thus does not have access to x_i. (We will formally prove this later.) So we expect that left of the wiggly line in (4), $H(x_1 \oplus \cdots \oplus x_r)$ occurs with negligible probability only. In other words, the wiggly line corresponds to the surface S discussed in Section 3.1. In fact, if we draw the border between P^*_{pre} and the remaining gates, we get exactly Figure 5 (in the 2D case at least). However, the approach of decomposing spacetime into subcircuits removes the necessity of dealing with the exact geometry of S.

Formally, we will need to apply Lemma 3. Given a function H and values x, B, let $H_{x \mapsto B}$ denote the function identical to H, except that $H_{x \mapsto B}(x) = B$. Let $A_1^H(x)$ denote the oracle machine that picks $x_1, \dots, x_{r-1} \xleftarrow{\$} \{0,1\}^\ell$ and sets $x_r := x \oplus x_1 \oplus \cdots \oplus x_{r-1}$ and prepares the state $|\mathsf{epr}\rangle$ and then executes P^*_{pre}. Let $A_2^H(x, B)$ denote the oracle machine that, given the state from A_1^H, executes $P_1^*, P_2^*, V_1, V_2, M^B$ with oracle access to $H_{x \mapsto B}$ instead of H, sets $\mathsf{accept} := 1$ iff $y_1 = y_2$ and $\omega(y_1 - \hat{y}) \leq \gamma n$, and returns accept. Let C_1, P_A^1, P_A^2, P_C be defined as in Lemma 3. Then by construction, $P_A^1 = \Pr[\mathsf{accept} = 1 : \text{Game 2}]$ (using the fact that $H = H_{x \mapsto H(x)}$). And $P_A^2 = \Pr[\mathsf{accept} = 1 : \text{Game 3}]$ for the following game:

Game 3 (Reprogramming H) *Pick $x_1, \dots, x_r \xleftarrow{\$} \{0,1\}^\ell$, $H \xleftarrow{\$}$ Fun. Execute circuit (4) until the wiggly line (with oracle access to H). Pick $B \xleftarrow{\$} \{0,1\}^n$. Execute circuit (4) after the wiggly line (with oracle access to $H_{x \mapsto B}$) resulting in $y_1, y_2, \hat{y}$. Let* $\mathsf{accept} := 1$ *iff $y_1 = y_2$ and $\omega(y_1 - \hat{y}) \leq \gamma n$.*

And finally $P_C = \Pr[x' = x_1 \oplus \cdots \oplus x_r : \text{Game 4}]$ for the following game:

Game 4 (Guessing $x_1 \oplus \cdots \oplus x_r$) *Pick $x_1, \dots, x_r \xleftarrow{\$} \{0,1\}^\ell$, $H \xleftarrow{\$}$ Fun, $j \xleftarrow{\$} \{1, \dots, q\}$. Prepare $|\mathsf{epr}\rangle$ and execute circuit P^*_{pre} until the j-th query to H. Measure the argument x' of that query.*

By Lemma 3, we have $|P_A^1 - P_A^2| \leq 2q\sqrt{P_C}$. Thus, abbreviating $x = x_1 \oplus \cdots \oplus x_r$ as guessX, we have

$$\big|\Pr[\mathsf{accept} = 1 : \text{Game 2}] - \Pr[\mathsf{accept} = 1 : \text{Game 3}]\big| \leq 2q\sqrt{\Pr[\mathsf{guessX} : \text{Game 4}]}. \quad (5)$$

We now focus on Game 3. Let ρ_{YLR} denote the state in circuit (4) at the wiggly line (for random $x_1, \dots, x_r, H$). Let L refer to the part of ρ_{YLR} that is on the wires entering P_1^*, and R refer to the part of ρ_{LR} on the wires entering P_2^*. Let Y refer to the lowest wire (containing EPR qubits). Notice that we have now reproduced the situation from the 1D case where space is split into two separate registers R and L, and the computation of y_1, y_2 is performed solely on R, L, respectively. In fact, we have now also identified the regions R_1, R_2 from the discussion in Section 3.1 (Figure 5): R_1 is the boundary between P^*_{pre} and P_1^*; analogously R_2. (R_3 from Figure 5 has no analogue here because V_3 does not receive here.) For given B, let $M_L(B)$ be the POVM operating on L consisting of P_1^* and V_1. (M_L can be modeled as a POVM because P_1^* and V_1 together return only a classical value and thus constitute a measurement.) Let $M_R(B)$ be the POVM operating on R consisting of P_2^* and V_2. Then we can rewrite Game 3 as:

Game 5 (Monogamy game) *Prepare ρ_{YLR}. Pick $B \xleftarrow{\$} \{0,1\}^n$. Apply measurement $M_L(B)$ to L, resulting in y_1. Apply measurement $M_R(B)$ to R, resulting in y_2. Measure Y in basis B, resulting in $\hat{y}$. Let* $\mathsf{accept} := 1$ *iff $y_1 = y_2$ and $\omega(y_1 - \hat{y}) \leq \gamma n$.*

Then $\Pr[\mathsf{accept} = 1 : \text{Game 3}] = \Pr[\mathsf{accept} = 1 : \text{Game 5}]$. Furthermore, Game 5 is again a monogamy of entanglement game, and [TFKW13] shows that $\Pr[\mathsf{accept} = 1 : \text{Game 5}] \leq \left(2^{h(\gamma)} \frac{1+\sqrt{1/2}}{2}\right)^n$. Combining this with the equalities between games derived so far, and with (5), we get

$$\Pr[\mathsf{accept} = 1 : \text{Game 1}] \leq \left(2^{h(\gamma)} \frac{1+\sqrt{1/2}}{2}\right)^n + 2q\sqrt{\Pr[\mathsf{guessX} : \text{Game 4}]}. \quad (6)$$

It remains to analyze Game 4. The intuition is that each oracle query performed by P^*_{pre} will be out of reach of one of the x_i, and thus unable to query $H(x_1 \oplus \cdots \oplus x_r)$. To formalize this, consider the j-th oracle query gate in P^*_{pre}, and denote with E_j the event at which that gate is located. Since P^*_{pre} is contained in the complement of the spacetime

region $\bigcap C_i^+$, for any $j \in \{1,\dots,q\}$, there is an i such that $E_j \notin C_i^+$. (This formalizes the fact that E_j cannot be reached by x_i.) Thus we can partition $\{1,\dots,q\}$ into disjoint sets J_i ($i = 1,\dots,r$) such that for all $j \in J_i$ we have $E_j \notin C_i^+$. Let Game 4_i denote Game 4 with the only difference that we pick $j \leftarrow J_i$ instead of $j \leftarrow \{1,\dots,q\}$. Then

$$\Pr[\mathsf{guessX} : \text{Game } 4] = \sum_{i=1}^{r} \frac{|J_i|}{q} \Pr[\mathsf{guessX} : \text{Game } 4_i]. \tag{7}$$

Let P^i_{low} be the subcircuit of P^*_{pre} not contained in C_i^+, and P^i_{high} the subcircuit of P^*_{pre} contained in C_i^+. Intuitively, P^i_{low} has no access to x_i, but P^i_{high} has. Since no wire can leave C_i^+, there is no wire from P^i_{high} to P^i_{low}. That is, executing P^*_{pre} is equivalent to first executing P^i_{low} and then P^i_{high}. Furthermore, for any $j \in J_i$, the j-th query gate is outside C_i^+ and thus in P^i_{low}. Hence, executing P^*_{pre} until the j-th query (for $j \in J_i$) is equivalent to executing P^i_{low} until the j-th query, P^i_{high} will never be executed. Thus we can rewrite Game 4_i as:

Game 6_i (Executing P^i_{low} only) *Pick $x_1,\dots,x_r \xleftarrow{\$} \{0,1\}^\ell$, $H \xleftarrow{\$}$ Fun, $j \xleftarrow{\$} J_i$. Prepare $|\mathsf{epr}\rangle$ and execute circuit P^i_{low} until the j-th query to H. Measure the argument x' of that query.*

Then $\Pr[\mathsf{guessX} : \text{Game } 4_i] = \Pr[\mathsf{guessX} : \text{Game } 6_i]$. Finally, note that x_i is sent by V_i at event $V_i^+ \in C_i^+$. So x_i may be accessed in P^i_{high}, but not in P^i_{low}. Thus in Game 6_i, x_i is chosen uniformly random from $\{0,1\}^\ell$ but never accessed. Thus $\Pr[\mathsf{guessX} : \text{Game } 6_i] = \Pr[x' = x_1 \oplus \cdots \oplus x_r : \text{Game } 6_i] \leq 2^{-\ell}$. Hence

$$\Pr[\mathsf{guessX} : \text{Game } 4] \overset{(7)}{=} \sum_{i=1}^{r} \tfrac{|E_i|}{q} \Pr[\mathsf{guessX} : \text{Game } 4_i] \leq 2^{-\ell} \sum_{i=1}^{r} \tfrac{|E_i|}{q} = 2^{-\ell}.$$

With (6) we get

$$\Pr[\mathsf{accept} = 1 : \text{Game } 1] \leq \left(2^{h(\gamma)} \frac{1+\sqrt{1/2}}{2}\right)^n + 2q2^{-\ell/2} = \nu.$$

Numerically, we can verify that for $\gamma \leq 0.037$, we have $2^{h(\gamma)} \frac{1+\sqrt{1/2}}{2} < 1$ and thus ν is negligible (for superlogarithmic n, ℓ and polynomially bounded q). □

In flat spacetime. Theorem 6 tells us where in spacetime a prover can be that passes verification. (Region **P**.) However, the theorem is quite general; it is not immediate what this means in the concrete setting of flat spacetime. In Section 3.4 we derive specialized criteria for flat spacetime and show that Theorem 6 implies that a prover can be precisely localized by verifiers arranged as a tetrahedron.

3.4 Position verification in flat spacetime

By Theorem 6, our PV scheme guarantees that the prover is within spacetime region $\mathbf{P} := \bigcap_{i=1}^r C^+(V_i^+) \cap C^-(V_1^-) \cap C^-(V_2^-)$. Or in words: the prover is at a point where he

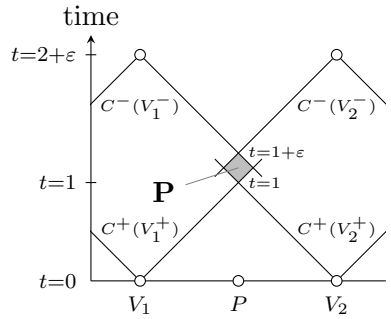


Figure 6: Precision in the 1D case

can receive $x_1, \ldots, x_r$ and send y_1, y_2 in time to be received by V_1, V_2. (In other words, even with several devices, the prover cannot do better than with a single device.) But what does this mean concretely? What is the precision (i.e., the size of $\mathbf{P}$) in concrete use cases?

For the 1D case discussed in Section 2, it is easy to compute $\mathbf{P}$. V_1, V_2 are located at positions $-1, 1$ and send x_1, x_2 at time $t = 0$, and expect y_1, y_2 at time $t = 2$. We assume flat spacetime, thus $C^+((x_0, t_0)) = \{(x, t) : |x - x_0| \leq t - t_0\}$ and $C^-((x_0, t_0)) = \{(x, t) : |x - x_0| \leq t_0 - t\}$. Hence $\mathbf{P} = C^+((-1, 0)) \cap C^+((1, 0)) \cap C^-((-1, 2)) \cap C^-((1, 2)) = \{(0, 1)\}$. I.e., in the 1D case we guarantee that the prover is at position 0 (at time $t = 1$), this gives an alternate proof of Theorem 2. We can also easily consider the case where the verifiers give the honest prover additional response time ε to account for the fact that a real prover cannot respond instantaneously. In this case the verifiers expect the answer at time $2 + \varepsilon$, and $\mathbf{P} = \{(x, t) : |x| \leq (t - 1) \wedge |x| \leq (1 + \varepsilon - t)\}$, thus the protocol verifiers that the prover is in the space interval $[-\varepsilon, \varepsilon]$. (Cf. Figure 6.)

In the 3D case, computing $\mathbf{P}$ is more complicated. The following corollary gives a characterization in flat spacetime.

Corollary 7 (Soundness in flat spacetime) *Assume flat spacetime (any dimension). Assume that $\gamma \leq 0.037$ and n, ℓ are superlogarithmic.*

Given two points x, y in space and a real number d, let $E(x, y, d) := \{z : \|z - x\| + \|z - y\| \leq d\}$. (I.e., E denotes an ellipsoid with foci x, y.).

Assume that V_i sends at time t_i^+ and space x_i^+, and expects the response at time t_i^- and space x_i^- in the PV scheme from Definition 5. Then that PV scheme is sound for $\mathbf{P} := \mathbf{P}_{\mathsf{space}} \times \mathbb{R}$ with $\mathbf{P}_{\mathsf{space}} = \bigcap_{\substack{i=1,\ldots,r \\ j=1,2}} E(x_i^+, x_j^-, t_j^- - t_j^+)$. (That is, the scheme proves that the prover is in the spacial region $\mathbf{P}_{\mathsf{space}}$.)

This corollary simplifies the computation of the location of the prover. For example, in 3D-space, with $r = 4$, we just need to compute the intersection of 8 ellipsoids (e.g., numerically).

Proof of Corollary 7. In the following, let i always range over $1, \ldots, r$ and j over $1, 2$. Let $\mathbf{P}_{\mathsf{space}} := \{x : (x, t) \in \mathbf{P}\}$ where $\mathbf{P}$ is as in Theorem 6. That is, $\mathbf{P}_{\mathsf{space}}$ is the projection

of $\mathbf{P}$ from spacetime into space. To show the corollary, by Theorem 6 it is then sufficient to show that $\mathbf{P}_{\mathsf{space}} = \bigcap E(x_i^+, x_j^-, t_j^- - t_j^+)$. For any z we have:

$$
\begin{aligned}
& z \in \mathbf{P}_{\mathsf{space}} \\
\text{iff} \quad & \exists t : (z,t) \in \mathbf{P} = \textstyle\bigcap_i C^+\big((x_i^+, t_i^+)\big) \cap \bigcap_j C^-\big((x_j^-, t_j^-)\big) \\
\text{iff} \quad & \exists t : \big(\forall i : \|x_i^+ - z\| \le t - t_i^+\big) \wedge \big(\forall j : \|x_j^- - z\| \le t_j^- - t\big) \\
\text{iff} \quad & \exists t : \big(\max_i \|x_i^+ - z\| + t_i^+ \le t\big) \wedge \big(\max_j \|x_j^- - z\| - t_j^- \le -t\big) \\
\text{iff} \quad & \big(\max_i \|x_i^+ - z\| + t_i^+\big) + \big(\max_j \|x_j^- - z\| - t_j^-\big) \le 0 \\
\text{iff} \quad & \max_{i,j} \|x_i^+ - z\| + t_i^+ + \|x_j^- - z\| - t_j^- \le 0 \\
\text{iff} \quad & \forall i,j : \|x_i^+ - z\| + \|x_j^- - z\| \le t_j^- - t_i^+ \\
\text{iff} \quad & z \in \textstyle\bigcap_{i,j} E(x_i^+, x_j^-, t_j^- - t_i^+).
\end{aligned}
$$

□

This corollary allows us to analyze particular settings, but it does not give any immediate insight as to whether we get nontrivial $\mathbf{P}_{\mathsf{space}}$ when doing so. For example, it might be that $\mathbf{P}_{\mathsf{space}}$ is no smaller than the ball of radius $\delta \approx 0.38$ from the naive approach in Section 3.1. The following lemma shows that, at least for a specific setup and for provers who answer instantaneously, this is not the case: the prover is localized perfectly.

Corollary 8 *Assume flat spacetime. Assume that the verifiers $V_1, \dots, V_4$ are on the vertices of a regular tetrahedron, and that the honest prover P is in the center of that tetrahedron. $V_1, \dots, V_4$ send $x_1, \dots, x_4$ at the same time t^+, and V_1, V_2 expect the answers at time $t^- := t^+ + 2R$ where R is the distance between V_i and P. (That is, V_1, V_2 expect instantaneous responses.)*

Then the PV scheme from Definition 5 is sound for $\mathbf{P} := \{P\} \times \mathbb{R}$, i.e., the prover is indeed guaranteed to be at location P.

Proof. Without loss of generality, we can assume that $P = 0$ and $R = 1$. Let $V_1, \dots, V_4$ be the vertices of the tetrahedron (and therefore the locations of the verifiers). By Corollary 7, all we need to show is then $\mathbf{P}_{\mathsf{space}} := \bigcap_{\substack{i=1,\dots,4 \\ j=1,2}} E(V_i, V_j, 2) = \{0\}$. Assume this is not the case. Let $P^* \in \mathbf{P}_{\mathsf{space}} \setminus \{0\}$. Since $\mathbf{P}_{\mathsf{space}}$ is convex, $\varepsilon P^* \in \mathbf{P}_{\mathsf{space}}$ for all $\varepsilon \in [0,1]$.

Let $f_{ij}(z) := \|z - V_i\| + \|z - V_j\| - 2$. Then $z \in E(V_i, V_j, 2)$ iff $f_{ij}(z) \le 0$. We have $f_{ij}(0) = 0$ for all i, j.

Since f_{ij} is differentiable at 0, we have $\frac{\partial f_{ij}(\varepsilon P^*)}{\partial \varepsilon}|_{\varepsilon=0} = \nabla f_{ij} \cdot P^*$ where $\cdot$ is the inner product, and ∇f_{ij} the gradient of f_{ij}. Furthermore, by symmetry (or direct calculation), we have that $\nabla f_{ij} \propto -(V_i + V_j)/2 =: -V_{ij}$. (Here $x \propto y$ means $x = \alpha y$ for some $\alpha > 0$.) For all $i = 1, \dots, 4$, $j = 1, 2$, and $\varepsilon \in [0,1]$, we have $\varepsilon P^* \in E(V_i, V_j, 2)$ and thus $f_{ij}(\varepsilon P^*) \le 0$. Together with $f_{ij}(0) = 0$ this implies that $0 \ge \frac{\partial f_{ij}(\varepsilon P^*)}{\partial \varepsilon}|_{\varepsilon=0} = \nabla f_{ij} \cdot P^* \propto -V_{ij} \cdot P^*$. Thus $V_{ij} \cdot P^* \ge 0$ for all $i = 1, \dots, 4$, $j = 1, 2$. $V_{31}, V_{41}, V_{32}, V_{42}$ form a square (they are

the midpoints of four edges of the tetrahedron). And $(V_{31}+V_{41}+V_{32}+V_{42})/4 = 0$, hence 0 is in the center of this square. Since $V_{ij} \cdot P^* \geq 0$ for $ij = 31, 41, 32, 42$ and $V_{31} = -V_{42}$ and $V_{32} = -V_{41}$, we have $V_{31} \cdot P^* = V_{32} \cdot P^* = 0$. Thus P^* is orthogonal to the plane containing that square, i.e., $P^* \propto \pm V_{12}$. $P^* \propto -V_{12}$ is excluded because $P^* \cdot V_{12} \geq 0$. Thus $P^* \propto V_{12}$.

We have $f_{31}(0) = 0$. And by symbolically computing the differentials, we verify that $\frac{\partial f_{31}(\varepsilon P^*)}{\partial \varepsilon}|_{\varepsilon=0} \propto \frac{\partial f_{31}(\varepsilon V_{12})}{\partial \varepsilon}|_{\varepsilon=0} = 0$ and $\frac{\partial^2 f_{31}(\varepsilon P^*)}{\partial \varepsilon^2}|_{\varepsilon=0} \propto \frac{\partial^2 f_{31}(\varepsilon V_{12})}{\partial \varepsilon^2}|_{\varepsilon=0} = \frac{4}{9} > 0$. This implies that $f_{31}(\varepsilon P^*) > 0$ for sufficiently small $\varepsilon \in [0,1]$. Hence $\varepsilon P^* \notin E(V_3, V_1, 2)$, in contradiction to $\varepsilon P^* \in \mathbf{P}_{\mathsf{space}}$ for all $\varepsilon \in [0,1]$. Thus our assumption that $\mathbf{P}_{\mathsf{space}} \neq \{0\}$ was wrong. By Corollary 7, the PV scheme is sound for $\mathbf{P} = \mathbf{P}_{\mathsf{space}} \times \mathbb{R} = \{0\} \times \mathbb{R}$. □

4 Position-based authentication

Position verification is, in itself, a primitive of somewhat limited use. It guarantees that no prover outside the region $\mathbf{P}$ can pass the verification. Yet nothing forbids a prover to just wait until some other honest party has successfully passed position verification, and then to impersonate that honest party. To realize the applications described in the introduction, we need a stronger primitive that not only proves that a prover is at a specific location, but also allows him to bind this proof to specific data. (The difference is a bit like that between identification schemes and message authentication schemes.) Such a primitive is be *position-based authentication*. This guarantees that the malicious prover cannot authenticate a message m unless he is in region $\mathbf{P}$ (or some honest party at location m wishes to authenticate that message).

Definition 9 (Secure position-based authentication) *A position-based authentication (PBA) scheme is a PV scheme where provers and verifiers get an additional argument m, a message to be authenticated.*

Let $\mathbf{P}$ be a region in spacetime. A position-based authentication (PBA) protocol is sound for $\mathbf{P}$ *iff for any non-uniform polynomial-time spacetime circuit P^* that has no gates in $\mathbf{P}$, the probability that the challenge verifiers (*soundness error*) accept is negligible in the following execution:*

P^* picks a message m^* and then interacts with honest verifiers (called the challenge verifiers) on input m^*. Before, during, and after that interaction, P^* may spawn instances of the honest prover and honest verifiers, running on inputs $m \neq m^*$. These instances run concurrently with P^* and the challenge verifiers and P^* may arbitrarily interact with them. Note that the honest prover/honest verifier instances may have gates in $\mathbf{P}$.

PBA was already studied in [BCF+11]. They give a generic transformation to convert a PV protocol into a PBA. The generic solution has two drawbacks, though:

- It needs $\Omega(\ell\mu)$ invocations of the PV protocol for *ell*-bit messages and $2^{-\mu}$ security level. (Our protocol below will need only one invocation.)

- It is only secure if a single instance of the honest prover runs concurrently. If the malicious prover can suitably interleave several instances of the honest prover, he can authenticate arbitrary messages.

(We do not know whether their solution gives adaptive security, i.e., whether the adversary can choose m^* and the honest provers' inputs m depending on communication he has seen before.) Although we do not have a generic transformation from PV to PBA that solves these issues, a small modification of our PV protocol leads to an efficient PBA secure against concurrent executions of the honest prover:

Definition 10 (Position-based authentication protocol) *The protocol is the same as in Definition 5, with the following modification only: Whenever in Definition 5, the verifier or prover queries $B := H(x_1 \oplus \cdots \oplus x_r)$, here he queries $B := H(x_1 \oplus \cdots \oplus x_r \| m)$ instead. (Where m is the message to be authenticated.) We also require that the verifiers do not start sending the messages x_i or expect y_1, y_2 before all V_i got m, and that $V_1^+ \neq V_2^+$ (i.e., V_1, V_2 do not send x_1, x_2 from the same location in space at the same time, a natural assumption).*

Theorem 11 *Assume that $\gamma \leq 0.037$ and n, ℓ are superlogarithmic.*

Then the PBA protocol from Definition 10 is sound for $\mathbf{P} := \bigcap_{i=1}^r C^+(V_i^+) \cap C^-(V_1^-) \cap C^-(V_2^-)$. (In words: There is no event in spacetime outside of $\mathbf{P}$ at which one can receive the messages x_i from all V_i, and send messages that will be received in time by V_1, V_2.)

Concretely, if the malicious prover performs at most q oracle queries,[9] then the soundness error is at most $\left(2^{h(\gamma)} \frac{1+\sqrt{1/2}}{2}\right)^n + 6q2^{-\ell/2}$.

The main difference to Theorem 6 is that now oracle queries are performed even within $\mathbf{P}$ (by the honest provers). We thus need to show that these queries do not help the adversary. The main technical challenge is that the message m^* is chosen adaptively by the adversary.

The specialized criteria for flat spacetime from Section 3.4 apply also for the PBA protocol, with identical proofs.

Proof of Theorem 11. We prove a stronger statement, namely that Theorem 11 holds even if the malicious prover P^* may have gates in $\mathbf{P}$, as long as he does not perform any queries $H(x\|m^*)$ for any $x \in \{0,1\}^\ell$ where m^* is the message picked by the malicious prover (see Definition 9).[10] Since the concurrently running honest verifiers and provers do not perform such queries, we can subsume them into P^* and assume that no honest verifiers or provers run, except for the challenge verifiers for m^*.

The proof now is similar to that of Theorem 6. We will heavily rely on the notation from that proof, and we will not reiterate the intuitive explanations behind the individual proof steps, unless new ideas are used.

[9] Actually, it is sufficient if the number of queries performed by gates inside $C_1^- \cup C_2^-$ is bounded by q. In particular, oracle queries after both verifiers have received y_1, y_2 do not count (as expected).

[10] For this to be well-defined we need that at any oracle query in $\mathbf{P}$, m^* is already defined. This is the case because by assumption m^* must be picked before the verifiers send x_i, i.e., $\forall i. V_i \in C^+(E)$ where E is the event where m^* is picked. Then $\mathbf{P} \subseteq C^+(E)$ by transitivity of $\prec$.

Since the adversary's circuit is finite (though arbitrary large), there is a finite upper bound on the length of the inputs to the random oracle. (That bound may depend on the security parameter, of course.) Let dom_H denote the set of all bitstrings of at most that length. Then we can assume that $H : \mathsf{dom}_H \to \{0,1\}^n$. (This ensures that the set of possible H does not get uncountable, else we would need to work with non-separable Hilbert spaces.) The subcircuits $P^*_{\mathbf{P}}, P^*_1, P^*_2$ are defined as in Theorem 6, but we use a different definition of P^*_{pre} and add another subcircuit P^*_{pickm}.

Subcircuit	Region in spacetime	Intuition
P^*_{pickm}	$\left((C_1^- \cup C_2^-) \setminus \bigcap C_i^+\right) \cap \left(\bigcap C^-(V_i^+) \cap C_1^- \cap C_2^-\right)$	Picking m^*
P^*_{pre}	$\left((C_1^- \cup C_2^-) \setminus \bigcap C_i^+\right) \setminus \left(\bigcap C^-(V_i^+) \cap C_1^- \cap C_2^-\right)$	Precomputation
$P^*_{\mathbf{P}}$	$\bigcap C_i^+ \cap C_1^- \cap C_2^-$	Gates in $\mathbf{P}$
P^*_1	$\bigcap C_i^+ \cap C_1^- \setminus C_2^-$	Computing y_1
P^*_2	$\bigcap C_i^+ \cap C_2^- \setminus C_1^-$	Computing y_2
P^*_{post}	$\Omega \setminus C_1^- \setminus C_2^-$	After protocol end

That is, we have split what was P^*_{pre} in Theorem 6 into two subcircuits P^*_{pre} and P^*_{pickm}. Again, those subcircuits partition the circuit P^*.

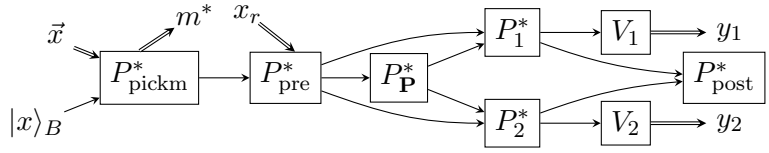
Using the analogous reasoning as in Theorem 6, we get the following facts:

$$P^*_{\text{pre}}, P^*_{\mathbf{P}}, P^*_1, P^*_2, P^*_{\text{post}} \nrightarrow P^*_{\text{pickm}}, \qquad P^*_{\mathbf{P}}, P^*_1, P^*_2 \nrightarrow P^*_{\text{pre}}, \qquad P^*_1, P^*_2 \nrightarrow P^*_{\mathbf{P}},$$
$$P^*_1 \nrightarrow P^*_2, \qquad P^*_2 \nrightarrow P^*_1, \qquad P^*_1 \nrightarrow V_2, \qquad P^*_2 \nrightarrow V_1, \qquad P^*_{\text{post}} \nrightarrow P^*_{\text{pre}}, P^*_{\mathbf{P}}, P^*_1, P^*_2, V_1, V_2. \tag{8}$$

We will now justify the name P^*_{pickm}, namely we show that the gate G which chooses m^* is in P^*_{pickm}. Let G be at event E. Definition 10 explicitly requires that all verifiers have the input m^* before they start the protocol, thus m^* must be chosen before $V_1, \ldots, V_r$ send their values x_i and before V_1, V_2 expect the answers y_1, y_2. Thus $E \in \bigcap C^-(V_i^+) \cap C_1^- \cap C_2^-$. Thus we immediately have $G \notin P^*_{\text{pre}}, P^*_{\text{post}}$. Assume $E \in \bigcap C_i^+$. Then $E \in C_1^+ \cap C^-(V_1^+)$. By antisymmetry of $\prec$, we have $C_1^+ \cap C^-(V_1^+) = \{V_1^+\}$. Thus $E = V_1^+$. Analogously $E = V_2^+$. Since $V_1^+ \neq V_2^+$, this cannot be, thus the assumption $E \in \bigcap C_i^+$ was false. Hence $E \notin \bigcap C_i^+$ and thus $G \notin P^*_{\mathbf{P}}, P^*_1, P^*_2$. Therefore $G \in P^*_{\text{pickm}}$.

Furthermore, we have that at least one of the x_i is not accessed in P^*_{pickm}. Assume all x_i are accessed in P^*_{pickm}. In particular, x_1 is accessed, thus there is a gate G in P^*_{pickm} in C_1^+. By definition of P^*_{pickm}, G is in $C^-(V_2^+)$, too. Thus $C^+(V_1^+) \cap C^-(V_2^+) \neq \varnothing$, thus $V_1^+ \prec V_2^+$. Analogously $V_2^+ \prec V_1^+$. By antisymmetry of $\prec$, $V_1^+ = V_2^+$ in contradiction to $V_1^+ \neq V_2^+$. Hence not all x_i are accessed in P^*_{pickm}. For simplicity, assume that it is x_r which is not accessed in P^*_{pickm}.

We can therefore write the execution of the protocol as the following quantum circuit:



(9)

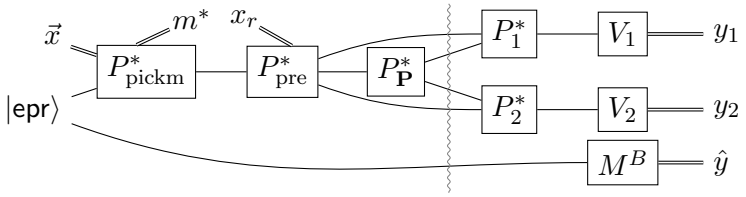
As before, we omit wires in the transitive hull. $\vec{x}$ denotes $x_1, \ldots, x_{r-1}$. Note that in contrast to Theorem 6, we cannot omit $P^*_{\mathbf{P}}$ here, since it is not empty.

The original protocol execution can be written as follows:

Game 1 (Protocol execution) *Pick* $x_1, \ldots, x_r \xleftarrow{\$} \{0,1\}^\ell$, $\hat{y} \xleftarrow{\$} \{0,1\}^n$, $H \xleftarrow{\$}$ *Fun where Fun is the set of functions* $\mathsf{dom}_H \to \{0,1\}^n$. *Let* $B := H(x_1 \oplus \cdots \oplus x_r)$. *Execute circuit* (9) *until* P^*_{pickm} *resulting in* m^*, y_1, y_2. *Let* $\mathsf{accept} := 1$ *iff* $y_1 = y_2$ *and* $\omega(y_1 - \hat{y}) \leq \gamma n$.

Game 2 (Using EPR pairs) *Pick* $x_1, \ldots, x_r \xleftarrow{\$} \{0,1\}^\ell$, $H \xleftarrow{\$}$ *Fun. Execute circuit* (10) *resulting in* $m^*, \hat{y}, y_1, y_2$, *where* M^B *uses basis* $B := H(x_1 \oplus \cdots \oplus x_r \| m^*)$.

Let $\mathsf{accept} := 1$ *iff* $y_1 = y_2$ *and* $\omega(y_1 - \hat{y}) \leq \gamma n$.



(10)

Exactly as in Theorem 6, $\Pr[\mathsf{accept} = 1 : \text{Game 1}] = \Pr[\mathsf{accept} = 1 : \text{Game 2}]$.

In Theorem 6, we used Lemma 3. In the present setting, Lemma 3 is not sufficient because it does not handle the fact that the adversary adaptively (i.e., depending on the random oracle itself) picks m^* which again determines where the random oracle is reprogrammed. Instead, we use the stronger Lemma 15 from Appendix B.

Given a function H and values x, m, B, let $H_{xm \mapsto B}$ denote the function identical to H, except that $H_{xm \mapsto B}(x\|m) = B$. Let $A_0^H()$ denote the oracle machine that prepares the state $|\mathsf{epr}\rangle$, picks $x_1, \ldots, x_{r-1} \xleftarrow{\$} \{0,1\}^\ell$, and then executes P^*_{pickm} and returns m^*. Let $A_1^H(x, B)$ denote the oracle machine that, given the state from A_0^H, sets $x_r := x \oplus x_1 \oplus \cdots \oplus x_{r-1}$ and then executes P^*_{pre} and $P^*_{\mathbf{P}}$. Let $A_2^H(x, B)$ denote the oracle machine that, given the state from A_1^H, executes $P_1^*, P_2^*, V_1, V_2, M^B$ with oracle access to $H_{xm^* \mapsto B}$ instead of H, sets $\mathsf{accept} := 1$ iff $y_1 = y_2$ and $\omega(y_1 - \hat{y}) \leq \gamma n$, and returns accept. Let C_1, P_A^1, P_A^2, P_C be defined as in Lemma 15. Then by construction, $P_A^1 = \Pr[\mathsf{accept} = 1 : \text{Game 2}]$ (using the fact that $H = H_{xm \mapsto H(x\|m)}$). And $P_A^2 = \Pr[\mathsf{accept} = 1 : \text{Game 3}]$ for the following game:

Game 3 (Reprogramming H) *Pick* $x_1, \ldots, x_r \xleftarrow{\$} \{0,1\}^\ell$, $H \xleftarrow{\$}$ *Fun. Execute circuit* (10) *until the wiggly line (with oracle access to* H*). Pick* $B \xleftarrow{\$} \{0,1\}^n$. *Execute circuit* (10) *after the wiggly line (with oracle access to* $H_{xm^* \mapsto B}$*) resulting in* $y_1, y_2, \hat{y}$. *Let* $\mathsf{accept} := 1$ *iff* $y_1 = y_2$ *and* $\omega(y_1 - \hat{y}) \leq \gamma n$.

And finally $P_C = \Pr[x' = x_1 \oplus \cdots \oplus x_r \text{ and } m' = m^* : \text{Game 4}]$ for the following game:

Game 4 (Guessing $x_1 \oplus \cdots \oplus x_r$) *Pick* $x_1, \ldots, x_r \xleftarrow{\$} \{0,1\}^\ell$, $H \xleftarrow{\$}$ *Fun*, $j \xleftarrow{\$} \{1, \ldots, q\}$. *Prepare* $|\mathsf{epr}\rangle$ *and execute circuit* $m^* \leftarrow P^*_{\text{pickm}}$. *Execute circuit* $P^*_{\text{pre}} \cup P^*_{\mathbf{P}}$ *until the* j*-th query to* H. *Measure the argument* $x'\|m'$ *of that query.*

By Lemma 15, we have $|P_A^1 - P_A^2| \leq 2q\sqrt{P_C} + q2^{-\ell/2+2}$. Thus, abbreviating "$x = x_1 \oplus \cdots \oplus x_r$ and $m' = m^*$" as guessX, we have

$$\big|\Pr[\mathsf{accept} = 1 : \text{Game 2}] - \Pr[\mathsf{accept} = 1 : \text{Game 3}]\big| \leq 2q\sqrt{\Pr[\mathsf{guessX} : \text{Game 4}]} + q2^{-\ell/2+2}. \tag{11}$$

We now focus on Game 3. Let ρ_{YLR} denote the state in circuit (10) at the wiggly line (for random $x_1, \ldots, x_r, H$). Let L refer to the part of ρ_{YLR} that is on the wires entering P_1^*, and R refer to the part of ρ_{LR} on the wires entering P_2^*. Let Y refer to the lowest wire (containing EPR qubits). For given B, let $M_L(B)$ be the POVM operating on L consisting of P_1^* and V_1. Let $M_R(B)$ be the POVM operating on R consisting of P_2^* and V_2. Then we can rewrite Game 3 as:

Game 5 (Monogamy game) *Prepare ρ_{YLR}. Pick $B \xleftarrow{\$} \{0,1\}^n$. Apply measurement $M_L(B)$ to L, resulting in y_1. Apply measurement $M_R(B)$ to R, resulting in y_2. Measure Y in basis B, resulting in $\hat{y}$. Let $\mathsf{accept} := 1$ iff $y_1 = y_2$ and $\omega(y_1 - \hat{y}) \leq \gamma n$.*

Exactly as in Theorem 6, we then derive $\Pr[\mathsf{accept} = 1 : \text{Game 3}] = \Pr[\mathsf{accept} = 1 : \text{Game 5}]$ using (11) and [TFKW13]:

$$\Pr[\mathsf{accept} = 1 : \text{Game 1}] \leq \left(2^{h(\gamma)}\frac{1+\sqrt{1/2}}{2}\right)^n + 2q\sqrt{\Pr[\mathsf{guessX} : \text{Game 4}]} + q2^{-\ell/2+2}. \tag{12}$$

By assumption, no gate in $\mathbf{P}$ queries $H(x\|m^*)$ for any x. Since $P_{\mathbf{P}}^*$ is contained in $\mathbf{P}$, this means no gate in $P_{\mathbf{P}}^*$ queries $H(x\|m^*)$. Thus in Game 4, guessX (which implies $m' = m^*$) can only occur if the j-th gate is not in $P_{\mathbf{P}}^*$. Thus $\Pr[\mathsf{guessX} : \text{Game 4}] = \Pr[\mathsf{guessX} : \text{Game 6}]$ where in Game 6 we remove $P_{\mathbf{P}}^*$:

Game 6 (Guessing $x_1 \oplus \cdots \oplus x_r$ without $P_{\mathbf{P}}^*$) *Pick $x_1, \ldots, x_r \xleftarrow{\$} \{0,1\}^\ell$, $H \xleftarrow{\$}$ Fun, $j \xleftarrow{\$} \{1, \ldots, q\}$. Prepare $|\mathsf{epr}\rangle$ and execute circuit $m^* \leftarrow P^*_{\text{pickm}}$. Execute circuit P^*_{pre} until the j-th query to H. Measure the argument $x'\|m'$ of that query.*

$P^*_{\text{pickm}} \cup P^*_{\text{pre}}$ are contained in $(C_1^- \cup C_2^-) \setminus \bigcap C_i^+$ (like P^*_{pre} was in Theorem 6). Thus, using the same proof as in the analysis of Game 4 in Theorem 6, we can show that $\Pr[x' = x_1 \oplus \cdots \oplus x_r : \text{Game 6}] \leq 2^{-\ell}$. Since $\Pr[\mathsf{guessX} : \text{Game 6}] \leq \Pr[x' = x_1 \oplus \cdots \oplus x_r : \text{Game 6}]$, with (12) we get

$$\Pr[\mathsf{accept} = 1 : \text{Game 1}] \leq \left(2^{h(\gamma)}\frac{1+\sqrt{1/2}}{2}\right)^n + 2q2^{-\ell/2} + q2^{-\ell/2+2} = \nu.$$

As in Theorem 6, ν is negligible under the assumptions of the theorem. □

Position-based quantum key distribution. Once we have PBA, we immediately get position-based quantum key distribution, and thus we can send messages that can only be decrypted by someone within region $\mathbf{P}$. We refer to [BCF+11] who describe how to do this, their construction applies to arbitrary PBA schemes. (As long as it has adaptive security, since in the QKD protocol, the adversary can influence the messages to be authenticated.)

5 Open problems

We list a number of open problems in the area of PV which, in our opinion, constitute interesting future work:

- We prove security if the verifiers allow the prover's answers to have an error rate up to 3.7% (Theorems 6 and 11), which may be challenging for implementations. For higher error rates, the results from [TFKW13] about monogamy of entanglement games do not give any guarantees. However, the best known attack is to measure each qubit in the Breidbart basis [BCF+11], leading to a much higher error rate of $1 - \cos(\pi/8)^2 \approx 14.6\%$. Can we improve the bound of 3.7%?

- We analyzed the case that the prover sends his measurement result y to two verifiers. We expect to get a much higher precision (especially when the prover's computation is not instantaneous) if more than two verifiers check his answers. But our proof does not cover that case. (We would at least need some generalization of the monogamy of entanglement games that handles more than two malicious parties.)

- Our analysis is in the random oracle model. Can we base the security of this or another protocol on computational assumptions in the standard model, e.g., the existence of quantum one-way functions?

- [BCF+11] gives general impossibility for information-theoretical PV, if the adversary has doubly exponential entanglement. [BK11] improves this to adversaries using only exponential entanglement. This leaves open whether there are PV schemes secure against all (even computationally unlimited) adversaries that have a polynomial amount of entanglement. Finding such a protocol would be highly interesting even beyond PV, because it would be (to our knowledge) the first protocol that has security against polynomial-time adversaries but not against unlimited adversaries, yet without using any computational hardness assumptions. Note that such a protocol would also circumvent our criticism concerning the protocols from [BCF+11, TFKW13] (Figure 1) because a polynomial-time prover P_2^* cannot produce the required amount of entanglement.

Acknowledgements. We thank Serge Fehr and Andris Ambainis for valuable discussions. Dominique Unruh was supported by the Estonian ICT program 2011-2015 (3.2.1201.13-0022), the European Union through the European Regional Development Fund through the sub-measure "Supporting the development of R&D of info and communication technology", by the European Social Fund's Doctoral Studies and Internationalisation Programme DoRa, by the Estonian Centre of Excellence in Computer Science, EXCS. We also used Sage [S+14] and PPL [BHZ08] for calculations and experiments, and the Sage Cluster funded by National Science Foundation Grant No. DMS-0821725.

A Buhrmann et al. and the 3D case

In the introduction, we claimed that the position verification protocol from [BCF+11] is not known to be secure in the 3D case. In this section, we explain why we believe that their security proof is incorrect. [BCF+11, full version] shows security of the 3D protocol by reduction to the security of the 1D protocol. On a very high level, the steps of their argument are as follows (for details, see [BCF+11, full version]):

1. In their 3D protocol, one verifier V_0 sends a quantum state $|\Psi\rangle$, and the other verifiers $V_1, \dots, V_3$ send values $\theta_1, \dots, \theta_3$ such that $\theta := \theta_1 \oplus \theta_2 \oplus \theta_3$ indicates in which basis $|\Psi\rangle$ is to be measured.
2. They identify the malicious prover P_0^* which is closest to V_0, as well as a verifier (w.l.o.g. V_1) who is far away from P_0^*.
3. P_0^* cannot keep the state $|\Psi\rangle$ until he gets θ_1 from V_1 because then he could not send his response in time.
4. Thus P_0^* needs to apply a transformation on $|\Psi\rangle$, leading to two quantum registers E_0, E_1 which are kept at P_0^* and forwarded to another prover P_1^*, respectively. This needs to happen before P_0^* knows θ_1 (and thus θ).
5. Thus now the responses sent to V_0 and V_1 need to be computed from two separate quantum registers E_0, E_1 when θ becomes known. This is the same situation as in the 1D case and shown to succeed only with negligible probability.

There are several issues with this proof: In step 2 we choose the prover P_0^* closest to V_0. However, this prover does not need to play a relevant role in the protocol. For example, P_0^* might not be involved in the attack at all, or P_0^* could just forward $|\Psi\rangle$ immediately to some other prover P_2^* who does the real attacking, but who is not close to V_0^*. Thus being close to V_0 is not something that in any way singles out a prover in a special way. (You could assume, e.g., that there is always one prover at the same location as V_0 who just forwards $|\Psi\rangle$ to wherever V_0 would have sent it anyway.)

In step 3, it is argued that P_0^* cannot keep the state $|\Psi\rangle$. That is true, strictly speaking. But stating it like this seems to suggest that P_0^* actually is the one who gets $|\Psi\rangle$. This is not necessarily true. (Or P_0^* might get $|\Psi\rangle$ but forward it immediately to another prover who then does something nontrivial with $|\Psi\rangle$.)

In step 4, even if we assume that P_0^* is indeed the prover that operates non-trivially on $|\Psi\rangle$, there is no reason to assume that P_0^* splits $|\Psi\rangle$ into *two* parts E_0, E_1. He could send parts of $|\Psi\rangle$ to two or more other provers. The situation becomes particularly challenging if some of these other provers are located close to two verifiers simultaneously (e.g., in the intersection of two R_i in Figure 4 (a)).

Solving these problems does not seem trivial. Although we do not have an attack on the protocol in higher dimensions, it seems that a proof would need to be considerably more involved than the 1D case. We believe that a proof will at least have to deal with difficulties similar to those described in Section 3.1. It seems that the protocol from [BCF+11] is secure in the 2D case (using a different proof) [Feh14], but it is unclear how to generalize that to the 3D case.

[TFKW13] (which analyses a very similar protocol) only addresses the 1D case.

In summary, we believe that the security of the 3D-protocol from [BCF+11] is an open problem.

B Random oracles

In this section, we derive a number of results for working with quantum random oracles. We first restate an auxiliary lemma from [Unr14, full version, Lemma 7]:

Lemma 12 *Let $|\Psi_1\rangle, |\Psi_2\rangle$ be quantum states that can be written as $|\Psi_i\rangle = |\Psi_i^*\rangle + |\Phi^*\rangle$ where both $|\Psi_i^*\rangle$ are orthogonal to $|\Phi^*\rangle$. Then $\mathrm{TD}(|\Psi_1\rangle, |\Psi_2\rangle) \leq 2\||\Psi_2^*\rangle\|$.*

Our first lemma is a slight generalization of the hardness of unstructured search [BBHT98] to an indistinguishability result. It states that it is hard to even decide whether a search problem (given as an oracle) has a solution, even if the claimed solution is provided after the last query to the oracle. We do not claim that the lemma is novel, but we are not aware of any writeup in the literature.

Lemma 13 *Let A be an oracle machine making at most q queries. Let $\delta_x(x) := 1$ and $\delta_x(y) := 0$ for $x \neq y$. Let 0 denote the all-zero function ($0(y) = 0$ for all y). Let ρ_0 denote the final state of A together with x in the following experiment: Pick $x \xleftarrow{\$} \{0,1\}^\ell$. Run $A^{\delta_x}()$. Let ρ_1 denote the final state state of A together with x in the following experiment: Pick $x \xleftarrow{\$} \{0,1\}^n$. Run $A^0()$.[11] Then $\mathrm{TD}(\rho_0, \rho_1) \leq q2^{-\ell/2+1}$.*

Proof. We can assume that A uses three quantum registers A, K, V for its state, oracle inputs, and oracle outputs. For a function f, let $O_f|a,k,v\rangle = |a,k,v \oplus f(k)\rangle$. Then the final state of $A^f()$ is $(UO_f)^q|\Psi_0\rangle$ for some unitary U and some initial state $|\Psi_0\rangle$.

Let $|\Psi_x^i\rangle := (UO_{\delta_x})^i|\Psi_0\rangle$ and $|\Psi^i\rangle := (UO_0)^i|\Psi_0\rangle = U^i|\Psi_0\rangle$. Then $\rho_0 = \sum_x 2^{-\ell}|x\rangle\langle x| \otimes |\Psi_x^q\rangle\langle\Psi_x^q|$ and $\rho_1 = \sum_x 2^{-\ell}|x\rangle\langle x| \otimes |\Psi^q\rangle\langle\Psi^q|$. Abbreviating $\mathrm{TD}(|\Psi\rangle\langle\Psi|, |\Phi\rangle\langle\Phi|)$ with $\mathrm{TD}(|\Psi\rangle, |\Phi\rangle)$, we compute:

$$\begin{aligned} D_i^x &:= \mathrm{TD}(|\Psi_x^i\rangle, |\Psi^i\rangle) = \mathrm{TD}(O_{\delta_x}|\Psi_x^{i-1}\rangle, |\Psi^{i-1}\rangle) \\ &\leq \mathrm{TD}(O_{\delta_x}|\Psi_x^{i-1}\rangle, O_{\delta_x}|\Psi^{i-1}\rangle) + \mathrm{TD}(O_{\delta_x}|\Psi^{i-1}\rangle, |\Psi^{i-1}\rangle) \\ &= D_{i-1}^x + \mathrm{TD}(O_{\delta_x}|\Psi^{i-1}\rangle, |\Psi^{i-1}\rangle). \end{aligned}$$

[11] Formally, $\rho_0 = \sum_{x\in\{0,1\}^\ell} 2^{-\ell}|x\rangle\langle x| \otimes \rho^x$ where ρ^x is the final state of $A^{\delta_x}()$. And $\rho_1 = \sum_{x\in\{0,1\}^\ell} 2^{-\ell}|x\rangle\langle x| \otimes \rho'$ where ρ' is the final state of $A^0()$.

Furthermore $D_0^x = \mathrm{TD}(|\Psi_0\rangle, |\Psi_0\rangle) = 0$, thus $D_q^x \leq \sum_{i=0}^{q-1} \mathrm{TD}(O_{\delta_x}|\Psi^i\rangle, |\Psi^i\rangle)$. We then have

$$\begin{aligned}
&\sum_{x\in\{0,1\}^\ell} 2^{-\ell}\mathrm{TD}(|\Psi_x^q\rangle, |\Psi^q\rangle) \leq \sum_{x,i} 2^{-\ell}\mathrm{TD}(O_{\delta_x}|\Psi^i\rangle, |\Psi^i\rangle)\\
&\leq \sum_{x,i} 2^{-\ell}\mathrm{TD}\big(O_{\delta_x}Q_x|\Psi^i\rangle + (1-Q_x)|\Psi^i\rangle,\ Q_x|\Psi^i\rangle + (1-Q_x)|\Psi^i\rangle\big)\\
&\overset{(*)}{\leq} \sum_{x,i} 2^{-\ell} 2\|Q_x|\Psi^i\rangle\| \overset{(**)}{\leq} 2\sum_i \sqrt{\sum_x 2^{-\ell}\|Q_x|\Psi^i\rangle\|^2}\\
&= 2\sum_i \sqrt{2^{-\ell}\cdot 1} = q2^{-\ell/2+1}
\end{aligned}$$

where Q_x is a projector projecting K onto $|x\rangle$ (i.e., $Q_x = I \otimes |x\rangle\langle x| \otimes I$). And $(*)$ uses Lemma 12. And $(**)$ uses Jensen's inequality. Finally,

$$\begin{aligned}
\mathrm{TD}(\rho_0, \rho_1) &= \mathrm{TD}\Big(\sum_x 2^{-\ell}|x\rangle\langle x| \otimes |\Psi_x^q\rangle\langle\Psi_x^q|, \sum_x 2^{-\ell}|x\rangle\langle x| \otimes |\Psi^q\rangle\langle\Psi^q|\Big)\\
&= \sum_x 2^{-\ell}\mathrm{TD}\big(|\Psi_x^q\rangle, |\Psi^q\rangle\big) \leq q2^{-\ell/2+1}.
\end{aligned}$$

□

The following lemma is a generalization of a lemma from [Unr14]. That lemma states that distinguishing a value $H(x)$ from random is as hard as finding x, for uniform x. In our generalization, instead of $H(x)$, we consider $H(x\|m)$, where m is chosen adaptively based on earlier random oracle queries.

Lemma 14 (One-way to hiding, adaptive) *Let $H : \{0,1\}^* \to \{0,1\}^n$ be a random oracle. Consider an oracle algorithm A_0 that makes at most q_0 queries to H. Consider an oracle algorithm A_1 that uses the final state of A_0 and makes at most q_1 queries to H. Let C_1 be an oracle algorithm that on input (j, B, x) does the following: run $A_1^H(x, B)$ until (just before) the j-th query, measure the argument of the query in the computational basis, output the measurement outcome. (When A makes less than j queries, C_1 outputs $\perp \notin \{0,1\}^\ell$.)*

Let

$$\begin{aligned}
P_A^1 &:= \Pr[b' = 1 : H \xleftarrow{\$} (\{0,1\}^* \to \{0,1\}^n), m \xleftarrow{\$} A_0^H(), x \leftarrow \{0,1\}^\ell, b' \leftarrow A_1^H(x, H(x\|m))]\\
P_A^2 &:= \Pr[b' = 1 : H \xleftarrow{\$} (\{0,1\}^* \to \{0,1\}^n), m \xleftarrow{\$} A_0^H(), x \leftarrow \{0,1\}^\ell, B \xleftarrow{\$} \{0,1\}^n, b' \leftarrow A_1^H(x, B)]\\
P_C &:= \Pr[x = x' \wedge m = m' : H \xleftarrow{\$} (\{0,1\}^* \to \{0,1\}^n), m \xleftarrow{\$} A_0^H(), x \leftarrow \{0,1\}^\ell, B \xleftarrow{\$} \{0,1\}^n,\\
&\qquad\qquad j \xleftarrow{\$} \{1,\ldots,q_1\}, x'\|m' \leftarrow C_1^H(j, B, x)]
\end{aligned}$$

Then $|P_A^1 - P_A^2| \leq 2q_1\sqrt{P_C} + q_0 2^{-\ell/2+2}$.

Proof. The proof follows the lines of [Unr14], but with many changes due to the additional adaptive choice of m.

Like in the proof of Theorem 11, we can assume the domain of H to be a finite (but large) set dom_H to avoid dealing with non-separable Hilbert spaces.

We first rewrite the probability P_A^1:

$$\begin{aligned}
P_A^1 &= \Pr[b'=1 : H \xleftarrow{\$} (\mathsf{dom}_H \to \{0,1\}^n), x \leftarrow \{0,1\}^\ell, m \xleftarrow{\$} A_0^H(), B \xleftarrow{\$} \{0,1\}^n, b' \leftarrow A_1^H(x, H(x\|m))] \\
&\overset{\varepsilon}{\approx} \Pr[b'=1 : H \xleftarrow{\$} (\mathsf{dom}_H \to \{0,1\}^n), x \leftarrow \{0,1\}^\ell, m \xleftarrow{\$} A_0^{H_{\setminus x}}(), B \xleftarrow{\$} \{0,1\}^n, b' \leftarrow A_1^H(x, H(x\|m))] \\
&\overset{(*)}{=} \Pr[b'=1 : H \xleftarrow{\$} (\mathsf{dom}_H \to \{0,1\}^n), x \leftarrow \{0,1\}^\ell, m \xleftarrow{\$} A_0^{H_{\setminus x}}(), B \xleftarrow{\$} \{0,1\}^n, b' \leftarrow A_1^{H_{xm\mapsto B}}(x, B)] \\
&\overset{\varepsilon}{\approx} \Pr[b'=1 : H \xleftarrow{\$} (\mathsf{dom}_H \to \{0,1\}^n), x \leftarrow \{0,1\}^\ell, m \xleftarrow{\$} A_0^H(), B \xleftarrow{\$} \{0,1\}^n, b' \leftarrow A_1^{H_{xm\mapsto B}}(x, B)] =: \hat{P}_A^1
\end{aligned}$$

Here $H_{\setminus x}$ denotes the function identical to H, except that $H_{\setminus x}(x\|\hat{m}) := 0$ for all $\hat{m}$. And $a \overset{\varepsilon}{\approx} b$ means that $|a-b| \leq \varepsilon := q_0 2^{-\ell/2+1}$. Here $(*)$ uses the fact that $H_{xm\mapsto B}$ and H are identically distributed for an adversary that did not query $H(x\|\hat{m})$ for any $\hat{m}$. And the first $\overset{\varepsilon}{\approx}$ is shown by reduction to Lemma 13: Let $A^f()$ be the algorithm that picks $H \xleftarrow{\$} (\mathsf{dom}_H \to \{0,1\}^n)$ and then runs $m \xleftarrow{\$} A_0^{H'}()$ where $H'(\hat{x}\|\hat{m}) := H(\hat{x}\|\hat{m})$ if $f(x)=0$ and $H'(\hat{x}\|\hat{m}) := 0$ otherwise. A^f performs at most q_0 queries to f. Then for uniform $x \in \{0,1\}^\ell$, the final state of $A^0()$ together with x is the state of the game on the lhs of $\overset{\varepsilon}{\approx}$ before choosing B, and the final state of $A^{\delta_x}()$ together with x is the state of the game in the rhs before choosing B. By Lemma 13, those two states have trace distance at most $\varepsilon = q_0 2^{-\ell/2+1}$. Thus the probabilities on the lhs and rhs of the first $\overset{\varepsilon}{\approx}$ have distance at most ε, as claimed. The second $\overset{\varepsilon}{\approx}$ is shown in the same way. Thus

$$|P_A^1 - \hat{P}_A^1| \leq 2\varepsilon = q_0 2^{-\ell/2+2}. \tag{13}$$

We proceed to bound $|\hat{P}_A^1 - P_A^2|$. We assume without loss of generality that A_0, A_1 perform only unitary operations (except for a final projective measurement that returns their classical output). For a given B, H, let $|\Phi_{HBm}\rangle$ denote the final state of $A_0^H(B)$, conditioned on classical output m. Let α_{HBm} be the probability that $A_0^H(B)$ outputs m.

We can assume that A_1 uses three quantum registers A, K, V for its state, oracle inputs, and oracle outputs. The initial state of A_1 is the final state $|\Phi_{HBm}\rangle$ of $A_0^H(B)$. For an oracle H, let $O_H|a,k,v\rangle = |a,k,v\oplus H(k)\rangle$. Let U denote the unitary transformation applied by A_1^H between queries to H. Let U_{xB} be an initial unitary transformation that depends on the inputs (x,B) of A_1^H. Then the final state of $A_1^H(x,B)$ running after $m \leftarrow A_0^H(B)$ is $|\Psi_{HBmx}^{q_1}\rangle$ with $|\Psi_{HBmx}^i\rangle := (UO_H)^i U_{xB}|\Phi_{HBm}\rangle$. And the final state of $A_1^{H_{xm\mapsto B}}(x,B)$ running after $m \leftarrow A_0^H(B)$ is $|\tilde{\Psi}_{HBmx}^{q_1}\rangle$ with $|\tilde{\Psi}_{HBmx}^i\rangle := (UO_{H_{xm\mapsto B}})^i U_{xB}|\Phi_{HBm}\rangle$. (Note: in the last sentence, we use A_0^H, not $A_0^{H_{xm\mapsto B}}$.)

Thus

$$|\hat{P}_A^1 - P_A^2| \leq \mathrm{TD}\Big(\sum_{H,B,x,m} \beta\,\alpha_{HBm} |\Psi_{HBmx}^{q_1}\rangle\langle\Psi_{HBmx}^{q_1}|, \sum_{H,B,x,m} \beta\,\alpha_{HBm} |\tilde{\Psi}_{HBmx}^{q_1}\rangle\langle\tilde{\Psi}_{HBmx}^{q_1}|\Big) \tag{14}$$

where $\beta := 2^{-n\,|\mathsf{dom}_H|} \cdot 2^{-n} \cdot 2^{-\ell}$ is the probability of each tuple (H,B,x).

In order to bound the rhs of (14), we will first bound $\mathrm{TD}(|\Psi_{HBmx}^{q_1}\rangle, |\tilde{\Psi}_{HBmx}^{q_1}\rangle)$. (Here $\mathrm{TD}(|\Psi\rangle,|\Phi\rangle)$ abbreviates $\mathrm{TD}(|\Psi\rangle\langle\Psi|, |\Phi\rangle\langle\Phi|)$.) For this, we fix H, B, m, x and omit

those values from the indices until further notice. Let $D_j := \mathrm{TD}(|\Psi^j\rangle, |\tilde{\Psi}^j\rangle)$. Then $D_0 = \mathrm{TD}(U_{xB}|\Phi_{HBm}\rangle, U_{xB}|\Phi_{HBm}\rangle) = 0$.

Let $V|a,k,v\rangle := |a,k,v \oplus B\rangle$, and let Q_{xm} project K onto $\big|x\|m\big\rangle$. (Formally, $Q_{xm} = I \otimes \big|x\|m\big\rangle\big\langle x\|m\big| \otimes I$.) Then $O_{H_{xm\mapsto B}} = VQ_{xm} + O_H(1-Q_{xm})$. (This is easily verified on basis vectors $|a,k,v\rangle$.) Then for $j \geq 1$,

$$\begin{aligned}
D_j &= \mathrm{TD}\big(UO_H|\Psi^{j-1}\rangle, UO_{H_{xm\mapsto B}}|\tilde{\Psi}^{j-1}\rangle\big) = \mathrm{TD}\big(O_H|\Psi^{j-1}\rangle, O_{H_{xm\mapsto B}}|\tilde{\Psi}^{j-1}\rangle\big) \\
&\leq \mathrm{TD}\big(O_H|\Psi^{j-1}\rangle, O_{H_{xm\mapsto B}}|\Psi^{j-1}\rangle\big) + \mathrm{TD}\big(O_{H_{xm\mapsto B}}|\Psi^{j-1}\rangle, O_{H_{xm\mapsto B}}|\tilde{\Psi}^{j-1}\rangle\big) \\
&= \mathrm{TD}\big(O_H|\Psi^{j-1}\rangle, O_{H_{xm\mapsto B}}|\Psi^{j-1}\rangle\big) + D_{j-1} \\
&= \mathrm{TD}\big(O_HQ_{xm}|\Psi^{j-1}\rangle + O_H(1-Q_{xm})|\Psi^{j-1}\rangle, VQ_{xm}|\Psi^{j-1}\rangle + O_H(1-Q_{xm})|\Psi^{j-1}\rangle\big) + D_{j-1} \\
&\overset{(*)}{\leq} 2\|VQ_{xm}|\Psi^{j-1}\rangle\| + D_{j-1} \leq 2\|Q_{xm}|\Psi^{j-1}\rangle\| + D_{j-1}.
\end{aligned}$$

Here $(*)$ uses Lemma 12. (Lemma 12 can be applied because $O_H(1-Q_{xm})|\Psi^{j-1}\rangle$ is in the image of $(1-Q_{xm})$ while $O_HQ_{xm}|\Psi^{j-1}\rangle$ and $VQ_{xm}|\Psi^{j-1}\rangle$ are in the image of Q_{xm} which is orthogonal to that of $(1-Q_{xm})$.)

Thus

$$\mathrm{TD}(|\Psi^{q_1}_{HBmx}\rangle, |\tilde{\Psi}^{q_1}_{HBmx}\rangle) = D_{q_1} \leq \sum_{j=1}^{q_1} 2\big\|Q_{xm}|\Psi^{j-1}\rangle\big\|. \tag{15}$$

From now on, we again write the indices H, B, m, x. We then have

$$\begin{aligned}
|\hat{P}^1_A - P^2_A| &\overset{(14),(*)}{\leq} \sum_{H,B,x,m} \beta\,\alpha_{HBm}\,\mathrm{TD}\big(|\Psi^{q_1}_{HBmx}\rangle, |\tilde{\Psi}^{q_1}_{HBmx}\rangle\big) \\
&\overset{(15)}{\leq} 2q_1 \sum_{H,B,x,m} \sum_{j=1}^{q_1} \frac{\beta\,\alpha_{HBm}}{q_1} \big\|Q_{xm}|\Psi^{j-1}_{HBmx}\rangle\big\| \\
&\overset{(**)}{\leq} 2q_1 \sqrt{\sum_{H,B,x,m} \sum_{j=1}^{q_1} \frac{\beta\,\alpha_{HBm}}{q_1} \big\|Q_{xm}|\Psi^{j-1}_{HBmx}\rangle\big\|^2}
\end{aligned} \tag{16}$$

Here $(*)$ uses the convexity of the trace distance (e.g., [NC10, eq. (9.50)]). And $(**)$ uses Jensen's inequality.

When starting with the final state $|\Phi_{HBm}\rangle$ from $m \leftarrow A^H_0(B)$, the final state of $C^H_1(j,B,x)$ is $|\Psi^{j-1}_{HBmx}\rangle$ by definition of C_1. Thus the probability that $C_1(j,B,x)$ outputs a given value $x'\|m'$ is $\big\|Q_{x'm'}|\Psi^{j-1}_{HBmx}\rangle\big\|$. Thus

$$P_C = \sum_{H,B,x,m} \sum_{j=1}^{q_1} \frac{\beta\,\alpha_{HBm}}{q_1} \big\|Q_{xm}|\Psi^{j-1}_{HBmx}\rangle\big\|^2.$$

With (16) we get $|\hat{P}^1_A - P^2_A| \leq 2q_1\sqrt{P_C}$ and hence

$$|P^1_A - P^2_A| \leq |\hat{P}^1_A - P^2_A| + |P^1_A - \hat{P}^1_A| \overset{(13)}{\leq} 2q_1\sqrt{P_C} + q_0 2^{-\ell/2+2}. \qquad \square$$

The following lemma is a general case of the random oracle programming lemma from Section 2 (Lemma 3). The lemma here additionally allows the adversary to adaptively select the part of the domain of the random oracle in which the reprogramming will take place. We only consider the specific case where this part is of the form $\cdot\|m$ for some m chosen by the adversary, but we believe that our proof may also give guidance for proofs for similar settings with adaptive programming.

Lemma 15 (Random oracle programming, adaptive) *Let $H : \{0,1\}^* \to \{0,1\}^n$ be a random oracle. Let (A_0, A_1, A_2) be oracle algorithms such that A_0 performs at most q_0 queries, and A_1, A_2 together perform at most q_{12} queries to H. Let C_1 be an oracle algorithm that on input (j, B, x) does the following: Run $A_1^H(x, B)$ till the j-th query to H, then measure the argument of that query in the computational basis, and output the measurement outcome. (Or $\perp$ if no j-th query occurs.) Let*

$$P_A^1 := \Pr[b' = 1 : H \xleftarrow{\$} (\{0,1\}^* \to \{0,1\}^n), m \xleftarrow{\$} A_0^H(), x \xleftarrow{\$} \{0,1\}^\ell, A_1^H(x, H(x\|m)), b' \leftarrow A_2^H(x, H(x\|m))]$$

$$P_A^2 := \Pr[b' = 1 : H \xleftarrow{\$} (\{0,1\}^* \to \{0,1\}^n), m \xleftarrow{\$} A_0^H(), x \xleftarrow{\$} \{0,1\}^\ell, B \xleftarrow{\$} \{0,1\}^n, A_1^H(x, B), H(x\|m) := B, b' \leftarrow A_2^H(x, B)]$$

$$P_C := \Pr[x = x' \wedge m = m' : H \xleftarrow{\$} (\{0,1\}^* \to \{0,1\}^n), m \xleftarrow{\$} A_0^H(), x \xleftarrow{\$} \{0,1\}^\ell, B \xleftarrow{\$} \{0,1\}^n, j \xleftarrow{\$} \{1, \ldots, q_{12}\}, x'\|m' \leftarrow C_1^H(j, B, x)]$$

Then $|P_A^1 - P_A^2| \leq 2q_{12}\sqrt{P_C} + q_0 2^{-\ell/2+2}$.

Proof. We will use Lemma 14 to prove Lemma 15. To keep the variable names apart, we decorate all variables from Lemma 14 with an overline. E.g., $\bar{A}_1^H, \bar{q}_0, \bar{P}_A^1$ etc. instead of A_1^H, q_0, P_A^1 etc.

Let $\bar{A}_0^H() := A_0^H()$, $\bar{q}_0 := q_0$, $\bar{q}_1 := q_{12}$. The algorithm $\bar{A}_1^H$, upon input (x, B), runs $A_1^H(x, B)$, then runs $b' \leftarrow A_2^{H_{xm\mapsto B}}(x, B)$, and returns b'. Here $H_{xm\mapsto B}$ denotes the function identical to H, except that $H_{xm\mapsto B}(x\|m) = B$.

We have

$$\begin{aligned}
P_A^1 &= \Pr[b' = 1 : H \xleftarrow{\$} (\{0,1\}^* \to \{0,1\}^n), m \xleftarrow{\$} A_0^H(), x \xleftarrow{\$} \{0,1\}^\ell, \\
&\qquad A_1^H(x, H(x\|m)), b' \leftarrow A_2^H(x, H(x\|m))] \\
&\overset{(*)}{=} \Pr[b' = 1 : H \xleftarrow{\$} (\{0,1\}^* \to \{0,1\}^n), m \xleftarrow{\$} A_0^H(), x \xleftarrow{\$} \{0,1\}^\ell, \\
&\qquad A_1^H(x, H(x\|m)), b' \leftarrow A_2^{H_{xm\mapsto H(x\|m)}}(x, H(x\|m))] \\
&= \Pr[b' = 1 : H \xleftarrow{\$} (\{0,1\}^* \to \{0,1\}^n), m \xleftarrow{\$} \bar{A}_0^H(), x \xleftarrow{\$} \{0,1\}^\ell, \\
&\qquad b' \leftarrow \bar{A}_1^H(x, H(x\|m))] = \bar{P}_A^1.
\end{aligned}$$

Here $(*)$ follows from the fact that H and $H_{xm\mapsto H(x\|m)}$ are identical functions. Similarly, we get $P_A^2 = \bar{P}_A^2$.

Furthermore,

$$\begin{aligned}
P_C &= \Pr[x = x' \wedge m = m' : H \xleftarrow{\$} (\{0,1\}^* \to \{0,1\}^n), m \xleftarrow{\$} A_0^H(), x \xleftarrow{\$} \{0,1\}^n, \\
&\qquad B \xleftarrow{\$} \{0,1\}^n, j \xleftarrow{\$} \{1,\ldots,q_{12}\}, x'\|m' \leftarrow C_1^H(j,B,x)] \\
&\overset{(*)}{\leq} \Pr[x = x' \wedge m = m' : H \xleftarrow{\$} (\{0,1\}^* \to \{0,1\}^n), m \xleftarrow{\$} A_0^H(), x \xleftarrow{\$} \{0,1\}^n, \\
&\qquad B \xleftarrow{\$} \{0,1\}^n, j \xleftarrow{\$} \{1,\ldots,q_{12}\}, x'\|m' \leftarrow \bar{C}_1^H(j,B,x)] \\
&= \Pr[x = x' \wedge m = m' : H \xleftarrow{\$} (\{0,1\}^* \to \{0,1\}^n), m \xleftarrow{\$} \bar{A}_0^H(), x \xleftarrow{\$} \{0,1\}^n, \\
&\qquad B \xleftarrow{\$} \{0,1\}^n, j \xleftarrow{\$} \{1,\ldots,\bar{q}_1\}, x'\|m' \leftarrow \bar{C}_1^H(j,B,x)] = \bar{P}_C.
\end{aligned}$$

For $(*)$, note that C_1^H by definition simulates A_1^H, aborting at the j-th query. While $\bar{C}_1^H$ simulates A_1^H followed by $A_2^{H_{xm\mapsto B}}$, aborting at the j-th query. Thus $\bar{C}_1^H$ is at least as likely to return the correct x', m' as C_1^H. The latter will return $x' = \perp$ when the j-th query would be made by $A_2^{H_{xm\mapsto B}}$.

Hence

$$|P_A^1 - P_A^2| = |\bar{P}_A^1 - \bar{P}_A^2| \overset{(*)}{\leq} 2\bar{q}_1\sqrt{\bar{P}_C} + \bar{q}_0 2^{-\ell/2+2} \leq 2q_{12}\sqrt{P_C} + q_0 2^{-\ell/2+2}.$$

Here $(*)$ uses Lemma 14. □

References

[Ash97] Neil Ashby. General relativity in the global positioning system. *Matters of gravity (newsletter of the Topical Group in Gravitation of the APS)*, 9, 1997. `http://www.phys.lsu.edu/mog/mog9/node9.html`, accessed: 2014-02-07. (Archived by WebCiteï¿½ at `http://www.webcitation.org/6ND19QXJ3`).

[BBHT98] Michel Boyer, Gilles Brassard, Peter Hï¿½yer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4-5):493–505, June 1998. Full version is arXiv:quant-ph/9605034.

[BCF+11] Harry Buhrman, Nishanth Chandran, Serge Fehr, Ran Gelles, Vipul Goyal, Rafail Ostrovsky, and Christian Schaffner. Position-based quantum cryptography: Impossibility and constructions. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 429–446. Springer, 2011. Full version is arXiv:1009.2490v4 [quant-ph].

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In *ASIACRYPT 2011*, pages 41–69, Berlin, Heidelberg, 2011. Springer-Verlag.

[BHZ08] R. Bagnara, P. M. Hill, and E. Zaffanella. The Parma Polyhedra Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. *Science of Computer Programming*, 72(1–2):3–21, 2008. `http://bugseng.com/products/ppl/`.

[BK11] Salman Beigi and Robert Koenig. Simplified instantaneous non-local quantum computation with applications to position-based cryptography. *New Journal of Physics*, 13(9):093036, 2011. Full version on arXiv:1101.1065 [quant-ph].

[CGMO09] Nishanth Chandran, Vipul Goyal, Ryan Moriarty, and Rafail Ostrovsky. Position based cryptography. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 391–407. Springer, 2009. Full version on `http://eprint.iacr.org/2009/364`.

[Feh14] Serge Fehr. Personal communication., January 2014. (Co-author of [BCF+11]).

[Ken12] Adrian Kent. Unconditionally secure bit commitment by transmitting measurement outcomes. *Phys. Rev. Lett.*, 109(13):130501, September 2012. Full version arXiv:1108.2879 [quant-ph].

[KMS11] Adrian Kent, William J. Munro, and Timothy P. Spiller. Quantum tagging: Authenticating location via quantum information and relativistic signaling constraints. *Phys. Rev. A*, 84:012326, Jul 2011. Full version on arXiv:1008.2147v6 [quant-ph].

[KTHW13] Jędrzej Kaniewski, Marco Tomamichel, Esther Hï¿½nggi, and Stephanie Wehner. Secure bit commitment from relativistic constraints. *IEEE Trans. on Inf. Theory*, 59(7):4687–4699, July 2013. Full version arXiv:1206.1740 [quant-ph].

[NC10] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 10th anniversary edition, 2010.

[S+14] W. A. Stein et al. *Sage Mathematics Software (Version 5.12)*. The Sage Development Team, 2014. `http://www.sagemath.org`.

[TFKW13] Marco Tomamichel, Serge Fehr, Jędrzej Kaniewski, and Stephanie Wehner. One-sided device-independent qkd and position-based cryptography from monogamy games. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 609–625. Springer, 2013. Full version at `http://arxiv.org/abs/1210.4359`.

[Unr14] Dominique Unruh. Revocable quantum timed-release encryption. In *Eurocrypt 2014*, LNCS. Springer, 2014. To appear, full version is `http://eprint.iacr.org/2013/606/20130923:033730`.

Symbol index

P_1^*	Subcircuit of P^* computing y_1	13
P_2^*	Subcircuit of P^* computing y_2	13
$P_{\mathbf{P}}^*$	Subcircuit of P^* in forbidden region $\mathbf{P}$	13
V_i^-	Event when V_i expects to receive	12
Ω	All of spacetime	13
$\mathbf{P}$	Region in spacetime in which the prover is verified to be	12
$\mathbf{P}_{\mathsf{space}}$	Region in space in which the prover is verified to be	18
P_{post}^*	Postcomputation done by P^*	13
$A \nrightarrow B$	No wire from subcircuit A to subcircuit B	14
M^B	Measurement in bases B	8
$x \leftarrow A$	x is assigned output of algorithm A	3
$h(\gamma)$	Binary entropy	3
$x \xleftarrow{\$} S$	x chosen uniformly from set S/according to distribution S	3
$\{0,1\}^n$	Bitstrings of length n	
$\omega(x)$	Hamming weight of x	3
$\langle\Psi\vert$	Conjugate transpose of $\vert\Psi\rangle$	4
$\vert\Psi\rangle$	Vector in a Hilbert space (usually a quantum state)	4
$\mathrm{TD}(\rho,\rho')$	Trace distance between ρ and ρ'	4
$\vert x\rangle_B$	Bitstring x encoded in basis B	4
dom_H	Domain of the random oracle	22
$H_{\setminus x}$	Like function H, but returns 0 given $x\Vert\cdot$	29
$\Vert x\Vert$	Euclidean norm of x	3
$\vert x\vert$	Absolute value / cardinality of x	3
$\mathbb{R}$	Real numbers	
Accept	Event that the verifiers accept	5
$C^+(x)$	Causal future of event x	11
$x \prec y$	x causally precedes y	11
C_i^+	Short for $C^+(V_i^+)$	13
$C^-(x)$	Causal past of event x	11
V_i^+	Event when V_i sends	12
C_i^-	Short for $C^-(V_i^-)$	13
accept	$\mathsf{accept} = 1$ iff verifiers accept	14, 23
Fun	Functions $\{0,1\}^\ell \to \{0,1\}^n$	14, 23
P_{pickm}^*	Subcircuit of P^* that picks m^*	22
P_{high}^i	Part of P_{pre}^* in C_i^+	17
P_{low}^i	Part of P_{pre}^* not in C_i^+	17
guessX	Abbreviation for $x = x_1 \oplus \cdots \oplus x_r$	16, 24
$H_{x\mapsto B}$	Like function H, but returns B given x	15
$\vert\mathsf{epr}\rangle$	n EPR pairs	15

Keyword index