

Quantum position verification in the random oracle model

Dominique Unruh

University of Tartu

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.

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., \mathbf{P} might be a rectangular room.) In general PV, thus, we assume several verifier devices V_1, \dots, V_n , and a prover P somewhere in the convex hull of V_1, \dots, 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, [5] 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). [4] 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).

¹ But both [5,4] give positive results assuming bounded retrieval/entanglement, see “related work” below.

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 [5].

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 [4], 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 of the protocol.)

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 (such as, e.g., PV or relativistic commitments [7,6]). And for the first time (to our knowledge), a security analysis uses adaptive programming of the quantum random oracle (in our PBA security proof).³

Related work. [5] showed a general impossibility of computationally secure PV in the classical setting; [4] showed the impossibility of information-theoretically secure PV in the quantum setting. [5] 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 [5]); this may be difficult to ensure in practice because a mirror might be such a forwarding device. [4,11] 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

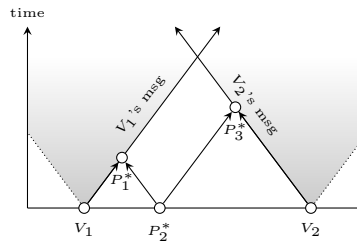


Fig. 1: Message flow in [4,11]. Security is only guaranteed if no entanglement is created before the shaded region. The scheme can be attacked if P_2^* sends EPR pairs to P_1^*, P_3^* who then can execute the attack from [8, Section I].

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., [1]). There is no reason to assume that this would not be the case for long-distance PV.

³ The semi-constant distribution technique from [13] programs the random oracle *before* the first adversary invocation, i.e., only *non-adaptive* programming is possible.

[8, 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. [4] 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 7 below). Furthermore, the protocols from [4,11] only work in the one-dimensional setting. ([4] has a construction for the 3D case, but their proof seems incorrect, see the full version [12] for a discussion.)

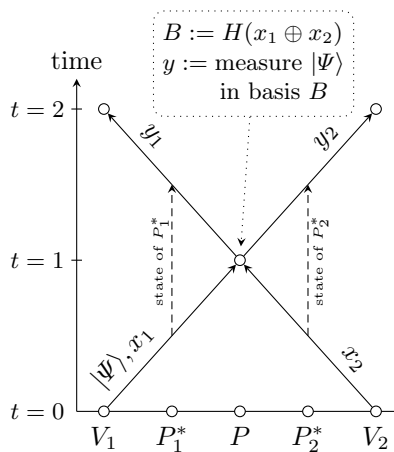


Fig. 2: One-dimensional PV protocol. Dotted lines indicate additional message flows of the adversary P_1^*, P_2^* .

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. Full proofs and further discussion are deferred to the full version [12].

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. $x \stackrel{\$}{\leftarrow} M$ 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 [9]. 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 $|x\rangle_B = H^{B_1}|x_1\rangle \otimes \dots \otimes H^{B_n}|x_n\rangle$ where H is the Hadamard matrix. An *EPR pair* has state $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$. $\text{TD}(\rho, \rho')$ 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 [3].

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, $0 \leq \gamma < 1/2$ (fraction of allowed errors). Let $H : \{0, 1\}^\ell \rightarrow \{0, 1\}^n$ 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 $\omega(y_1 - \hat{y}) \leq \gamma n$. If so (and y_1, y_2 arrived in time), they accept.

We can now prove security in our simplified setting.

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 most⁴*

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

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 $\Pr[\text{Accept}]$ is small. Here we abbreviate the event “ $y_1 = y_2$ and $\omega(y_1 - \hat{y}) \leq \gamma n$ ” as “Accept”.

⁴ This probability is negligible if $\gamma \leq 0.037$ and n, ℓ are superlogarithmic.

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} .*

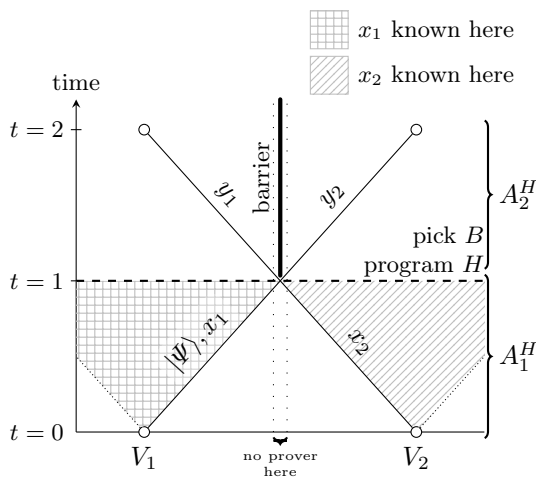


Fig. 3: Spacetime diagram depicting various steps of the proof of Theorem 2.

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 $\Pr[\text{Accept} : \text{Game 1}] = \Pr[\text{Accept} : \text{Game 2}]$.

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, [8, 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 $B \leftarrow \{0, 1\}^n$ is chosen, and the random oracle is reprogrammed to return $H(x_1 \oplus x_2) = B$ after $t = 1$.*

To clarify this, if $H_0 : \{0, 1\}^\ell \rightarrow \{0, 1\}^n$ 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 shown in the full version.

Lemma 3. *Let $H : \{0, 1\}^\ell \rightarrow \{0, 1\}^n$ 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*

$$\begin{aligned} P_A^1 &:= \Pr[b' = 1 : H \xleftarrow{\$}(\{0, 1\}^\ell \rightarrow \{0, 1\}^n), x \leftarrow \{0, 1\}^\ell, A_1^H(x), b' \leftarrow A_2^H(x, H(x))] \\ P_A^2 &:= \Pr[b' = 1 : H \xleftarrow{\$}(\{0, 1\}^\ell \rightarrow \{0, 1\}^n), x \leftarrow \{0, 1\}^\ell, B \xleftarrow{\$}\{0, 1\}^n, \\ &\quad A_1^H(x), H(x) := B, b' \leftarrow A_2^H(x, B)] \\ P_C &:= \Pr[x = x' : H \xleftarrow{\$}(\{0, 1\}^\ell \rightarrow \{0, 1\}^n), x \leftarrow \{0, 1\}^\ell, j \xleftarrow{\$}\{1, \dots, q\}, x' \leftarrow C_1^H(j, x)] \end{aligned}$$

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 re-programmed 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, $P_A^1 = \Pr[\text{Accept} : \text{Game 2}]$. And $P_A^2 = \Pr[\text{Accept} : \text{Game 3}]$. 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

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

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 $\Pr[\text{Accept} : \text{Game 3}] = \Pr[\text{Accept} : \text{Game 5}]$.

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 tripartite 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:

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

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 [11] shows that the rhs is bounded by $\left(2^{h(\gamma)} \frac{1+\sqrt{1/2}}{2}\right)^n$. Thus $\Pr[\text{Accept} : \text{Game 5}] \leq \left(2^{h(\gamma)} \frac{1+\sqrt{1/2}}{2}\right)^n$. And from (1) and the equalities between games, we have $|\Pr[\text{Accept} : \text{Game 1}] - \Pr[\text{Accept} : \text{Game 5}]| \leq 2q2^{-\ell/2}$.

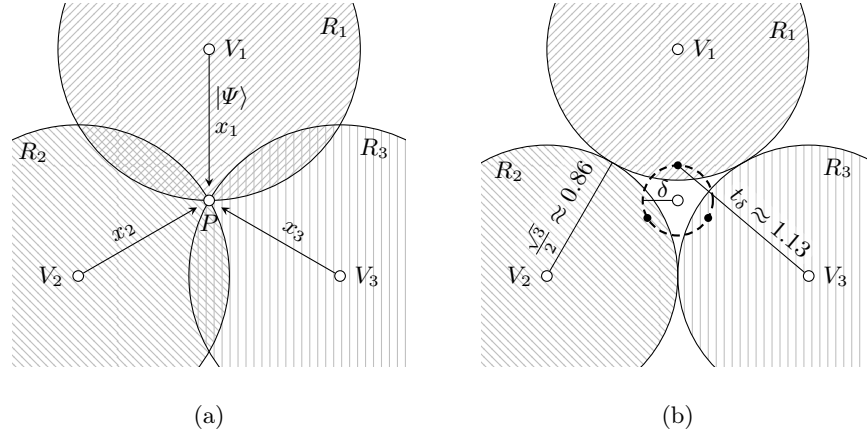


Fig. 4: The geometry of space at time t_δ (i.e., when B first becomes known). Left for $\delta = 0$, right for $\delta = \sqrt{4 - \sqrt{12}} - \frac{1}{2} \approx 0.23$.

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

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. For n -dimensional PV we need at least $n + 1$ verifiers.⁵ To illustrate the problems occurring in the higher 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 an 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)

⁵ 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 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.

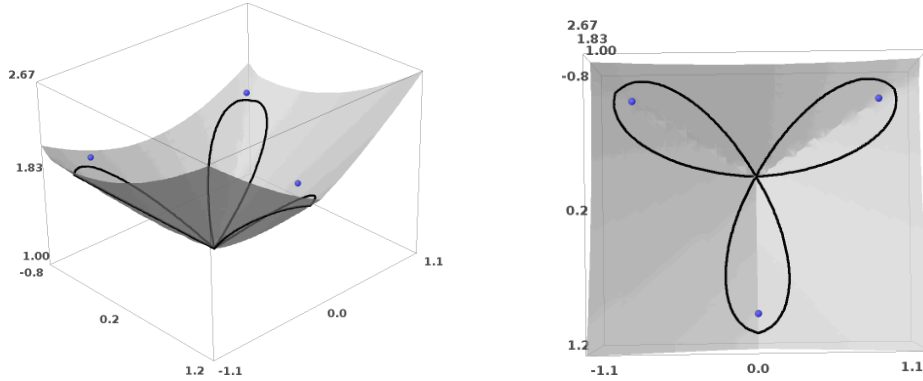


Fig. 5: The surface S in spacetime at which B is sampled. The dots floating over S denote when the verifiers need to receive y (i.e., the dots are at time 2 and space V_1, V_2, V_3). The thick black lines enclose the areas R_1, R_2, R_3 on S from which the verifiers can be reached in time. (Right: top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

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 [11] 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 . Then R_1, R_2, R_3 are the regions from which light can travel to V_1, V_2, V_3 within time $2 - t_\delta$. We can compute that they are disjoint 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 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, \dots, x_n) where t is the time and x_1, \dots, 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, \dots, 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, \dots, x_n) with $t, x_1, \dots, x_n \in \mathbb{R}$. 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: $(t_x, x_1, \dots, x_n) \prec (t_y, y_1, \dots, y_n)$ iff $t_x \leq t_y$ and $\|(x_1, \dots, x_n) - (y_1, \dots, y_n)\| \leq t_y - t_x$.

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 $x = (t, x_1, \dots, x_n)$ 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.

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 sense) without closed causal curves (at least in the spacetime region where the protocol is executed; otherwise quantum circuits may end up having loops).

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 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-time⁷ 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.)*

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.

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 $H : \{0, 1\}^\ell \rightarrow \{0, 1\}^n$ 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 $B := H(x_1 \oplus \dots \oplus x_r)$, $|\Psi\rangle := |\hat{y}\rangle_B$.
- 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 $B := H(x_1 \oplus \dots \oplus x_r)$, 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 .
- 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, 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 $E \in \bigcap_{i=1}^r C^+(V_i^+) \cap C^-(V_1^-) \cap C^-(V_2^-) \setminus \mathbf{P} \neq \emptyset$, then the protocol could even be broken by a malicious prover with a single device: P^* could be at event E , receive x_1, \dots, 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

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.

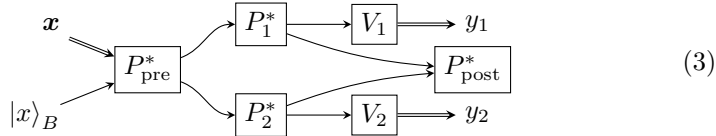
In addition, in some abuse of notation, by V_1 we denote the circuit at V_1^- that receives y_1 . Similar for V_2 .

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:

$$\begin{aligned}
P_1^*, P_2^*, P_{\text{post}}^* &\not\rightarrow P_{\text{pre}}^*, & P_1^* &\not\rightarrow P_2^*, & P_2^* &\not\rightarrow P_1^*, \\
P_1^* &\not\rightarrow V_2, & P_2^* &\not\rightarrow V_1, & P_{\text{post}}^* &\not\rightarrow P_1^*, P_2^*, V_1, V_2.
\end{aligned} \tag{2}$$

Here $A \not\rightarrow B$ 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:



Here \mathbf{x} is short for x_1, \dots, 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., $P_1^* \not\rightarrow P_{\text{pre}}^*$, 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:

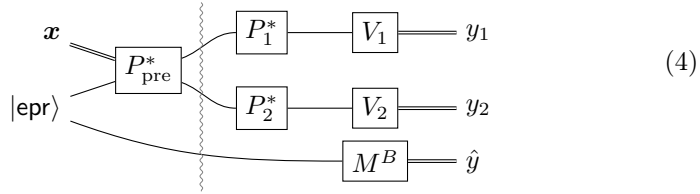
Game 1 (Protocol execution) Pick $x_1, \dots, x_r \xleftarrow{\$} \{0, 1\}^\ell$, $\hat{y} \xleftarrow{\$} \{0, 1\}^n$, $H \xleftarrow{\$} \text{Fun}$ where Fun is the set of functions $\{0, 1\}^\ell \rightarrow \{0, 1\}^n$. Let $B := H(x_1 \oplus \dots \oplus x_r)$. Execute circuit (3) resulting in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

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

As in the proof of the 1D case, we now delay the choice of \mathbf{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 $x_1, \dots, x_r \xleftarrow{\$} \{0, 1\}^\ell$, $H \xleftarrow{\$} \text{Fun}$. Let $B := H(x_1 \oplus \dots \oplus x_r)$. Execute circuit (4) resulting in y_1, y_2 .

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



Here $|\text{epr}\rangle$ is the state consisting of n EPR pairs, i.e., $|\text{epr}\rangle = 2^{-n/2} \sum_{x \in \{0, 1\}^n} |x\rangle \otimes |x\rangle$. The top and bottom wire originating from $|\text{epr}\rangle$ 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 $\Pr[\text{accept} = 1 : \text{Game 1}] = \Pr[\text{accept} = 1 : \text{Game 2}]$.

Again like in the 1D case, we will now reprogram the random oracle. That is, instead of computing $B := H(x_1 \oplus \dots \oplus x_r)$, we pick $B \xleftarrow{\$} \{0, 1\}^n$ at some point in the execution and then program the random oracle via $H(x_1 \oplus \dots \oplus x_r) := B$. 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), $H(x_1 \oplus \dots \oplus x_r)$ 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 $H_{x \rightarrow B}$ denote the function identical to H , except that $H_{x \rightarrow B}(x) = B$. Let $A_1^H(x)$ denote the oracle machine that picks $x_1, \dots, x_{r-1} \stackrel{\$}{\leftarrow} \{0, 1\}^\ell$ and sets $x_r := x \oplus x_1 \oplus \dots \oplus x_{r-1}$ and prepares the state $|\text{epr}\rangle$ 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 $H_{x \rightarrow B}$ instead of H , sets $\text{accept} := 1$ 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, $P_A^1 = \Pr[\text{accept} = 1 : \text{Game 2}]$ (using the fact that $H = H_{x \rightarrow H(x)}$). And $P_A^2 = \Pr[\text{accept} = 1 : \text{Game 3}]$ for the following game:

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

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

Game 4 (Guessing $x_1 \oplus \dots \oplus x_r$) Pick $x_1, \dots, x_r \stackrel{\$}{\leftarrow} \{0, 1\}^\ell$, $H \stackrel{\$}{\leftarrow} \text{Fun}$, and $j \stackrel{\$}{\leftarrow} \{1, \dots, q\}$. Prepare $|\text{epr}\rangle$ 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 $x = x_1 \oplus \dots \oplus x_r$ as guessX , we have

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

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_{YLR} 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 $\text{accept} := 1$ iff $y_1 = y_2$ and $\omega(y_1 - \hat{y}) \leq \gamma n$.

Then $\Pr[\text{accept} = 1 : \text{Game 3}] = \Pr[\text{accept} = 1 : \text{Game 5}]$. Furthermore, Game 5 is again a monogamy of entanglement game, and [11] shows that $\Pr[\text{accept} = 1 : \text{Game 5}] \leq \left(2^{h(\gamma)} \frac{1 + \sqrt{1/2}}{2}\right)^n$. We can furthermore show (see the full version [12]) that $\Pr[\text{guessX} : \text{Game 4}] \leq 2^{-\ell}$. With (5) we get

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

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). \square

In flat spacetime. Theorem 6 tells us where in spacetime a prover can be that passes verification. (Region \mathbf{P} .) However, the theorem is quite general; it is not immediate what this means in the concrete setting of flat spacetime. In the full version [12] 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.

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 called *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 7 (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 [4]. 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 $\ell\mu$ -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 8 (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 $B := H(x_1 \oplus \dots \oplus x_r)$, here he queries $B := H(x_1 \oplus \dots \oplus x_r \| m)$ 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 9. *Assume that $\gamma \leq 0.037$ and n, ℓ are superlogarithmic.*

Then the PBA protocol from Definition 8 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, 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 proof is given in the full version [12].

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 [4] 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.)

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 [10] and PPL [2] for calculations and experiments, and the Sage Cluster funded by National Science Foundation Grant No. DMS-0821725.

References

1. N. 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 WebCite at <http://www.webcitation.org/6ND19QXJ3>).
2. 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/>.
3. D. Boneh, O. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry. Random oracles in a quantum world. In *ASIACRYPT 2011*, pages 41–69. Springer, 2011.
4. H. Buhrman, N. Chandran, S. Fehr, R. Gelles, V. Goyal, R. Ostrovsky, and C. Schaffner. Position-based quantum cryptography: Impossibility and constructions. In *CRYPTO 2011*, volume 6841 of *LNCS*, pages 429–446. Springer, 2011.
5. N. Chandran, V. Goyal, R. Moriarty, and R. Ostrovsky. Position based cryptography. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 391–407. Springer, 2009.
6. J. Kaniewski, M. Tomamichel, E. Hänggi, and S. Wehner. Secure bit commitment from relativistic constraints. *IEEE Trans. on Inf. Theory*, 59(7):4687–4699, 2013.
7. A. Kent. Unconditionally secure bit commitment by transmitting measurement outcomes. *Phys. Rev. Lett.*, 109(13):130501, Sept. 2012.
8. A. Kent, W. J. Munro, and T. P. Spiller. Quantum tagging: Authenticating location via quantum information and relativistic signaling constraints. *Phys. Rev. A*, 84:012326, 2011.
9. M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 10th anniversary edition, 2010.
10. W. Stein et al. *Sage Mathematics Software (Version 5.12)*. The Sage Development Team, 2014. <http://www.sagemath.org>.
11. M. Tomamichel, S. Fehr, J. Kaniewski, and S. Wehner. One-sided device-independent QKD and position-based cryptography from monogamy games. In *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 609–625. Springer, 2013.
12. D. Unruh. Quantum position verification in the random oracle model. IACR ePrint 2014/118, February 2014. Full version of this paper.
13. M. Zhandry. Secure identity-based encryption in the quantum random oracle model. In *Crypto 2012*, volume 7417 of *LNCS*, pages 758–775. Springer, 2012.