

Position Based Cryptography*

Nishanth Chandran¹, Vipul Goyal^{1**}, Ryan Moriarty¹, and
Rafail Ostrovsky^{2***}

¹ Department of Computer Science, UCLA, {nishanth,vipul,ryan}@cs.ucla.edu

² Department of Computer Science and Mathematics, UCLA, rafail@cs.ucla.edu

Abstract. We consider what constitutes *identities* in cryptography. Typical examples include your name and your social-security number, or your fingerprint/iris-scan, or your address, or your (non-revoked) public-key coming from some trusted public-key infrastructure. In many situations, however, **where you are** defines your identity. For example, we know the role of a bank-teller behind a bullet-proof bank window not because she shows us her credentials but by merely knowing her location. In this paper, we initiate the study of cryptographic protocols where the identity (or other credentials and inputs) of a party are derived from its *geographic location*.

We start by considering the central task in this setting, i.e., securely verifying the position of a device. Despite much work in this area, we show that in the Vanilla (or standard) model, the above task (i.e., of secure positioning) is impossible to achieve. In light of the above impossibility result, we then turn to the Bounded Storage Model and formalize and construct information theoretically secure protocols for two fundamental tasks:

- Secure Positioning; and
- Position Based Key Exchange.

We then show that these tasks are in fact *universal* in this setting – we show how we can use them to realize Secure Multi-Party Computation. Our main contribution in this paper is threefold: to place the problem of secure positioning on a sound theoretical footing; to prove a strong impossibility result that simultaneously shows the insecurity of previous attempts at the problem; and to present positive results by showing that the bounded-storage framework is, in fact, one of the “right” frameworks (there may be others) to study the foundations of position-based cryptography.

1 Introduction

1.1 Motivation

In cryptography, typically a party will possess a set of credentials determining: its identity, what tasks it can do, which protocols it can participate in and so on.

* Research supported in part by NSF grants 0716835, 0716389, 0830803.

** Research supported in part by above NSF grants and a Microsoft Graduate Fellowship.

*** Research supported in part by an IBM Faculty Award, Xerox Innovation Group Award, NSF grants 0430254, 0716835, 0716389, 0830803 and U.C. MICRO grant.

These set of credentials will typically correspond to the party having some of the following attributes: some secret information (e.g., a secret key), authenticated information (e.g., a digitally signed certificate from a trusted entity), biometric feature and so on. In this paper, we ask the following question: can the *geographical position* of a party be one of the credentials? The geographical position of a party is valuable in a number of natural settings. We give a few examples:

- **Position based Secret Communication.** Consider communication between different military establishments. For example, the Pentagon in Washington D.C. might want to send a message (having some classified information) such that it can only be read by an individual present at the US military base in South Korea. In a traditional solution, the South Korean military base will have a secret key to decrypt the message. However, the enemy might try to break into the military base computers to capture this key. It would be desirable to add an additional layer of security that would guarantee that anyone reading the message is *physically* present at the South Korean base.
- **Position based Authentication/Signatures.** In the above example, suppose the South Korean military base wants to send some information to the Pentagon. It would be desirable for the Pentagon to have a guarantee that the message was indeed sent from the geographical position of the military base.

Indeed, the above list is not exhaustive. One could think about *position based access control* (where access to a resource needs to be restricted to certain locations, e.g., a printer or fax machine is accessible only to people inside some set of offices) and *pizza delivery* (where the pizza company first wants to verify that the person placing the order is indeed located at the delivery address he specified). To perform such “position specific” tasks, we introduce the notion of *position based cryptography*.

The first natural question that arises is: “Can you convince others about where you are?”. More precisely, we have a prover who claims to be at a geographical position P . There is a set of remote verifiers (or in other words, a positioning infrastructure) who wish to make sure that the prover is indeed at position P as claimed (for example, by executing a protocol with that prover). We call the above problem as “Secure Positioning”. The question of secure positioning is a fundamental one and deals with designing a system which enables a prover to communicate back and forth with a group of verifiers to give them an *interactive proof of its geographic position*.

The problem of secure positioning is well studied in the security community (see e.g., [20, 21, 3, 6, 5]). The de-facto method to perform secure positioning is based on the time of response technique where the messages travel with the speed of radio waves which is equal to the speed of light (this is similar in nature to how the commercial GPS systems work, see section 1.3). At a high level, the verifiers will send messages to the device and will measure the time taken to receive a response. Although there have been several proposed protocols for secure positioning, all of them are completely insecure under the so called “collusion attack”. That is, if a set of (possibly cloned) provers collude together

and work in a controlled manner during the protocol execution, the provers will be able to convince the verifiers that the verifiers are talking to a prover at position P (even though none of the adversarial provers may be at P). We in fact show that, unfortunately, such an attack is unavoidable. That is, it is impossible to have secure protocols for positioning in this Vanilla model (even if one is willing to make computational assumptions). Hence, we cannot hope to realize most of the meaningful position based tasks.

In light of the above drawbacks, in this paper we explore the intriguing possibility if secure positioning protocols exist which can resist collusion attacks. In search of an answer to this question, we turn to the bounded storage model (BSM), introduced by Maurer [18]. Quite surprisingly, this model turns out to be a right model for proving the security of position-based cryptographic tasks. We first construct a protocol for information theoretic secure positioning in this model. To our knowledge, this is the first protocol which is secure even against collusion attacks. Although secure positioning is an important step, the full power of position based cryptography can only be realized if we achieve key exchange with the device at a particular geographic position. Hence we introduce position based key exchange and present two protocols to achieve it in the BSM. Our first protocol achieves security against a computationally bounded adversary (in the BSM). In this protocol, we achieve key exchange between the verifiers and any device at position P that is enclosed within the tetrahedron formed between 4 verifiers in 3-dimensional space. Our second protocol achieves information theoretic key exchange between the verifiers and devices at positions P that lie in a specific geometric region (characterized by a condition that P must satisfy) within the tetrahedron.

Note that we are interested only in verifying the position claim of devices that are within the tetrahedron enclosed between the 4 verifiers. This is not a limitation, since apriori, we are restricting, by geographical bounds, the locations where an honest device can be located (such as inside a room, to get access to a printer or a hard drive). If a device makes a position claim that lies outside of this region, we reject the claim without any verification. We stress, however, that we do not make *any* assumption about the positions of adversaries in the system. In particular, this freedom for the adversarial devices guarantees that no set of adversaries (some of whom may even be outside of the tetrahedron) can falsely prove that any one of them is at position P inside the tetrahedron as long as none of them are at position P .

1.2 The Two Models Considered

The Vanilla Model. We now informally describe the Vanilla model. We have a device (also referred to as the prover) who is located at a position P (where P is a point in a d -dimensional Euclidean space). There exists a set of verifiers $\{V_1, V_2, \dots, V_m\}$ at different points in the d -dimensional space, such that P lies inside the tetrahedron enclosed by the verifiers. The verifiers are allowed to execute a protocol with the prover to achieve some task. More precisely, a verifier can send messages to the prover at different points in time (with a

speed up to the speed of radio waves) and also record the messages which are received from it (along with the time when they are received). The verifiers have a secret channel among themselves using which they can coordinate their actions by communicating before, during or after protocol execution. There could be multiple adversaries with possibly cloned devices who share a covert channel and collude together. This setting is referred to as the Vanilla model.

The Bounded Storage Model. The bounded storage model (BSM) was introduced by Maurer in [18] and has been the subject of much work [18, 4, 1, 9, 17, 22, 19, 13, 10]. Very roughly, this model assumes that there is a bound on the amount of information that parties (including an adversary) can store. It assumes the existence of random strings, having high min-entropy, available to the parties at the beginning of the protocol. An adversary is allowed to retrieve and store an arbitrary function of this random string, as long as the length of the output of the function is not longer than the adversary’s storage bound. We assume that parties can broadcast random strings having high min-entropy, but cannot store these strings. A closely related model to the BSM is the bounded retrieval model (BRM), introduced and studied in various related contexts by Di Crescenzo et al [8] and Dziembowski [11, 12]. This model assumes that parties *can store* information having high min entropy, but an adversary can only retrieve part of it. Recently, Dziembowski and Pietrzak [14] introduced *intrusion resilient secret sharing* where shares of a secret (stored on different machines) are made artificially large so that it is hard for an adversary to retrieve a share completely, even if it breaks into the storage machine. We note that in the current work, we use the work of [14] on Intrusion Resilient Secret Sharing schemes as a starting point. We build and extend these techniques by combining them with geometric arguments to prove the security of our protocol.

In the context of position based cryptography, by bounded storage model, we mean the Vanilla model setting where the verifiers can broadcast information having high entropy (or control a randomness source which can) such that the adversaries can only retrieve and store, say, a constant fraction of this information as it passes by at high speed. The assumption that the adversaries cannot retrieve (and store) all the information that goes by seems plausible in our setting since the information travels at a very high speed (particularly when, e.g., the verifiers have several sources broadcasting information at different frequencies). The reason we call our model bounded storage (as opposed to bounded retrieval) is that we do not assume that verifiers (and the honest prover) can fully store the broadcasted information themselves.

1.3 Related Work

Secure Positioning. We remark that the problem of position-based cryptography as such has not been studied before. However, secure positioning is a well-studied problem in the field of wireless security. There have been several proposed protocols ([2, 20, 23, 3, 6, 21, 24]). All these protocols are susceptible to the collusion attack outlined earlier. One can get around this problem of multiple cloned

adversaries by assuming a setup phase where the verifiers give an *unclonable* tamper-proof hardware [16, 15] (having some secret information) to all possible future provers. However in the current work, we focus on the setting where the *only* credential needed by a prover is its geographical position.

In [5], a model is considered, that makes the assumption that there can exist verifiers that are covert or hidden to provers and adversaries. Based on this, they provide solutions to secure positioning. The protocols in [5] are also susceptible to multiple colluding adversaries, although the attack required is more subtle than in other cases. We describe this attack, as well as give a detailed description of related work on secure positioning and the BSM in the full version of this paper [7].

Global Positioning System. The problem addressed by the global positioning system (GPS) is complementary to the one considered in our work. In GPS, there is device trying to determine *its own* geographic position with the aid of various satellites (in a non-adversarial setting). The GPS satellites continually broadcast information in a synchronized manner with the speed of light. The time taken by the information broadcast by various satellites to reach a GPS receiver enables the receiver to compute its position using triangulation techniques.

1.4 Our Contributions

In this paper, we give the following results towards developing a theory of position based cryptography:

- We begin with a lower bound for the Vanilla model in Section 3. We show that there does not exist a protocol in the Vanilla model using which a group of verifiers can securely verify the location claim of a prover. The impossibility is obtained via an explicit attack which does not depend on the computational power of the parties. To begin with, the lower bound holds if all the parties (i.e., the verifiers, the honest prover and the adversaries) are given unbounded computational power. Further, it holds even if the verifiers are given unbounded computational power but the adversaries (and thus obviously the honest prover) are restricted to being probabilistic polynomial time (PPT) machines (i.e., one may make cryptographic hardness assumptions). Finally, the protocols in [5] additionally assume the existence of hidden and mobile base stations and present protocols for secure positioning. In the full version of this paper [7], we describe explicit attacks breaking the security of these protocols in common settings (where a prover learns the response of the success or failure of the protocol). With the impossibility of this most fundamental task, we cannot hope to perform most other meaningful position based tasks (including position based key exchange) in the Vanilla model.
- Given the above severe lower bound, the task of now choosing a model in which protocols for secure positioning exist becomes a tricky one. One of the main technical contributions of this paper is to connect the bounded storage model

to position based cryptography. Remarkably, bringing these seemingly unrelated ideas together enables us to achieve meaningful and elegant protocols and proofs of security for position based cryptography.

- In the BSM, we give a protocol for secure positioning (in Section 5) which is provably secure against any number of (possibly computationally unbounded) adversaries colluding together, as long as the total amount of information they can retrieve and store is bounded. To our knowledge, this is the first protocol for positioning which does not fail against collusion attacks. We also describe, in Section 6, how our protocol for secure positioning can be compiled with any unauthenticated computationally secure key exchange protocol (like Diffie-Hellman) to achieve computationally secure position based key exchange in the BSM. That is, only a prover who is at a designated position P will receive the key to be shared (even under the presence of any number of PPT adversaries with bounded storage).
- We then present a protocol (in Section 7) that does information theoretically secure key exchange between the verifiers and a device at P . The construction of such a protocol turns out to be surprisingly intricate. While our secure positioning (and computationally secure position based key exchange) can handle claims of all positions P that lie within the tetrahedron formed between 4 verifiers in 3-dimensional space, our information theoretic key exchange protocol can handle positions P that lie in a specific region (which we characterize, using a geometric argument, by a condition that P must satisfy) within the tetrahedron. In the full version, we show (for a few example cases) that this region is a large fraction of the enclosed tetrahedron and also provide some figures containing what this region looks like (for various placements of the 4 verifiers). In order to show the security of our protocol, we need to prove delicate timing arguments (based on geometric properties) as well as prove that the protocol of [14] is secure even in the case when multiple parallel adversaries can gain access to the machines and may collude after they have finished accessing the machines.
- Using the above two fundamental protocols as building blocks, we demonstrate that the protocols for more complex tasks can be readily constructed. We consider the problem of establishing a secure channel between two devices (such that each device has a guarantee on the geographic position of the device at the other end). After establishing pairwise secure channels, a group of devices can perform “position based” multi-party computation, where associated with each input, there is a guarantee about the position of the device giving that input. We also discuss the setup of a position based public key infrastructure, where a certificate provides a guarantee about the position (as opposed to the identity) of the owner of the public key in question (We discuss these applications in further detail in the full version.). We remark that the above is obviously not intended to be an exhaustive list of applications and one can construct protocols for several other tasks.

Our results do not require any pre-sharing of data (cryptographic keys and so on) between the prover and the verifiers. The only required credential of a

prover is its *real geographic position*. We present the high level ideas of the model and constructions and defer the full formal proofs to the full version [7].

Open Problem: Other Models for Position Based Cryptography. By turning to the bounded storage model, we are able to provide the first provably secure constructions of cryptographic tasks that use position as an identity. Given our strong impossibility results in the Vanilla model, an important open question is: do there exist other natural models that allow us to obtain positive results of similar nature?

2 The Model

In this section, we briefly discuss our model. More details can be found in the full version. There are three types of parties in our model: Prover, Verifier and Adversary. We treat time and space as “continuous” (rather than discrete). We assume that messages travel at a speed equal to that of radio waves (which is the same as the speed of light). In the beginning, each party (prover, verifiers and adversaries) is given as input, party’s own position (as a point in the d -dimensional space), the position of all verifiers and the security parameter κ . The verifiers and the adversaries are given the claimed position of the prover.

The parties can send out the following two types of messages : (a) Broadcast messages: A broadcast message originating at a position P travels in concentric hyperspheres centered at P in all directions, (b) Directional messages: A directional message, instead of traveling in all directions, travels only in a specific direction specified by a sector. Such messages can be sent using directional antennas. Additionally, verifiers have a private channel among themselves which allows them to talk to each other secretly. Adversaries also have a private (and covert) channel among themselves which allows them to talk to each other secretly such that no verifier suspects any adversarial activity. More details about these messages (along with formal definitions of secure positioning and key exchange) can be found in the full version.

The above is our so called Vanilla Model where we prove the impossibility of realizing the most basic position based task (i.e., secure positioning). We assume that parties can send directional messages in the Vanilla model in order to prove a strong lower bound. As noted earlier, all our positive results are in the BSM. Our bounded storage model is the same as the Vanilla model except for the following changes:

- Verifiers “possess” a reverse block entropy source (defined formally in the full version) capable of generating strings with high min-entropy, say $(\delta + \beta)n$, where n is the length of the string (and $0 < \delta + \beta < 1$; it is also called min-entropy rate). By possessing a reverse block entropy source, we mean that either the verifier itself is capable of generating such a stream of high min-entropy, or it has a randomness source (located at the same point in space as itself) which generates and broadcasts such a stream. We do not assume that

the verifiers can retrieve and store the broadcasted stream of data themselves. Generating a lot of entropy is easy; one can think of an “explosion” which generates a lot of noise that can be measured but not stored.

- There exists a bound βn on the total amount of information the adversaries can store as the information passes at a high speed. The storage bound βn could be any constant fraction of the min-entropy $(\delta + \beta)n$. The honest parties (including the verifiers) are required to have a storage capacity of only $O(\kappa \cdot \log(n))$.
- Verifiers and provers cannot send directional messages. We however do not restrict the adversary from sending directional messages.
- Let X be a string having large min-entropy as before. The sender (which is a verifier) generates X and sends it out. Any receiver gets to retrieve and store $f(X)$ (for any arbitrary f) in a way such that the total amount of information which it has retrieved does not exceed the storage bounds. In case a party receives multiple strings simultaneously, it can retrieve information from these strings, in any order, any number of times (i.e., we do not restrict the adversaries to retrieve information from a string only once) as long as the total memory bound is not violated on the amount retrieved.

Observe that the last step above also enforces that any information about a string X (having large min-entropy) that is sent from one adversary to the other is also bounded (since an adversary gets to retrieve and resend only $f(X)$). This rules out simple “reflection attacks” to create a huge storage (where a pair of adversaries close to each other just keep reflecting the string X to each other hence essentially storing X thus violating the bounded storage assumption).

Relaxing Assumptions. For clarity of exposition during our positive results, we make the assumption that the devices can read bits from the stream and perform computations instantaneously. We refer the reader to the full version for details on how to remove this assumption.

3 Lower Bound on Secure Positioning in the Vanilla Model

We now show a lower bound for the Vanilla model. We show that there does not exist a protocol in the Vanilla model using which a group of verifiers can securely verify the location claim of a prover. The impossibility is obtained via an explicit attack which does not depend on the computational power of the parties. To begin with, the lower bound holds if all the parties (i.e., the verifiers, the honest prover and the adversaries) are given unbounded computational power. Further, it holds even if the verifiers are given unbounded computational power but the adversaries (and thus obviously the honest party) are restricted to being PPT machines (i.e., one may make cryptographic hardness assumptions). Finally, we present a few extensions of our lower bound in the full version.

Theorem 1 *There does not exist a protocol to achieve secure positioning in the Vanilla model.*

Proof. Let there be n verifiers $\{V_1, V_2, \dots, V_n\}$ that take part in a protocol to verify that a prover is at a position P . We show that for any protocol, there exists a set of l adversaries (l to be defined later) who can interact with the verifiers in such a manner that it is impossible to distinguish if the verifiers are interacting with an adversary or the actual prover.

Consider the hypersphere of radius r around position P such that the distance between V_i and P for all i be strictly greater than r . In other words, we require that r is such that no verifier is present within the hypersphere of radius r centered at position P . For all i , let the straight line joining V_i and P intersect the hypersphere at position I_i . Let there exist $l \leq n$ such intersection points. We note that l could be less than n because, two (or more) verifiers $V_i, V_j, i \neq j$ may be such that P, V_i and V_j lie on the same straight line in d -dimensional space. We place adversaries A_1, A_2, \dots, A_l at points I_1, I_2, \dots, I_l . The verifiers may run an interactive protocol with the prover in order to verify that the prover is at position P . We show that these l adversaries together can simulate the execution of the protocol in such a way that the verifiers cannot distinguish between an execution in which they are interacting with the prover at P and an execution in which they are interacting with these set of adversaries.

Any verification protocol is a series of messages (along with corresponding times), each being from one of the n verifiers to the prover or vice-versa. The verifiers can then verify the position of the prover by analyzing the message they sent and the response they got (along with corresponding times). We give a strategy for every A_m such that the adversaries together can prove that they are at position P .

Let the time taken for any message to travel between V_i and P be T_i . Note that the distance between A_m , for all m , and P is fixed (equal to r). Hence, let the time taken for a message to travel between A_m (for all m) and P be α . Let the set of verifiers that lie on the same straight line that connects A_m and P be \mathcal{V}_m . Let the distance between two adversaries A_m and $A_{m'}$ be $dist(m, m')$ (note that $dist(m, m) = 0$).

Now during the protocol execution, every A_m does the following. A_m only listens to messages sent by all $V_i \in \mathcal{V}_m$ and ignores messages sent by other verifiers. A_m is at a location such that all the messages sent by V_i (s.t., $V_i \in \mathcal{V}_m$) to the point P would be received by it (irrespective of whether V_i sends a broadcast message or a directional message). Lets say that a message M is received from a verifier V_i . For every adversary $A_{m'}$ (including itself, i.e., $1 \leq m' \leq l$), A_m internally delays M by the duration of time $delay(m, m') = 2\alpha - dist(m, m')$, and then sends it to $A_{m'}$ over the covert channel. Hence, every single adversary (including A_m itself) would receive the message at time 2α (over the covert channel) after the time when A_m receives it from V_i (over the broadcast or directional channel).

For every adversary A_m , now assume that the protocol requires the honest prover to send a reply message, at time t , in directions such that verifiers in

set \mathcal{V}_m would receive it (note that since all of them are in a straight line in the same direction of point P , either all of them would receive it or none would). In that case, A_m computes the response message using its view over the covert channel so far and sends it at time $t + \alpha$ using a directional message (such that only verifiers in \mathcal{V}_m receive it). However, A_m does not send any messages to V_i for $V_i \notin \mathcal{V}_m$ (if the verifiers in other sets are required to receive this message as well, they will be “taken care of” by other adversaries near them).

The following simple argument shows that every adversary A_m runs exactly a copy of the execution of the prover, only at a time α later. Once this is shown, since it takes time T_i for a prover to send a response to V_i when $V_i \in \mathcal{V}_m$, and it takes A_m only time $T_i - \alpha$, the exact same response will reach V_i at exactly the same instance of time (irrespective of whether it originated at P or at the location of A_m).

We show that the following two statements are true. $\text{delay}(m, m')$ is a non-negative value for all m, m' and every message which reaches the prover at P will reach all the adversaries after exactly time α . This will prove that all adversaries run exactly the same copy of the prover, but at a later instance of time.

The first statement follows trivially from triangle inequality. For the second statement, assume that verifier V_i sends a message to the prover at time t . Let m be such that $V_i \in \mathcal{V}_m$ and t' be the time taken for a message to travel between V_i and A_m . The honest prover clearly receives the message at time $t + t' + \alpha$. The adversary A_m receives the message at time $t + t'$ and hence all the adversaries receive it at time $t + t' + 2\alpha$ over the covert channel.

This proves that all adversaries run exactly the same copy of the prover, but at an instance α later. Hence, any execution of a protocol run between the n verifiers and the prover can be simulated by l adversaries running the protocol with the n verifiers. \square

We remark here that the above impossibility result holds even in a stronger model where there is a fixed bound on the number of adversaries, as long as this bound can depend on the number of verifiers in the system (but not on the secure positioning protocol itself). This motivates our search for alternative models, where we do not restrict the number of adversaries and still achieve positive results.

4 Preliminaries

Vadhan [22], introduced BSM pseudorandom generators (PRG). Informally, for string X sampled from a distribution having high min-entropy and for a uniformly random seed K , the distribution of the output of the BSM PRG (denoted by $\text{PRG}(X, K)$), is statistically close to the uniform distribution of appropriate length even when given K and $A(X)$ where A is any arbitrary function with bounded output length. We introduce a relaxation of BSM PRGs, which we call BSM entropy generators (EG). The difference between a BSM EG and a BSM PRG is that the output distribution of a BSM EG is only guaranteed to

have high min-entropy, and not necessarily be close to the uniform distribution. We refer the reader to the full version for formal details about the definitions, constructions and instantiations.

5 Secure Positioning in the Bounded Storage Model

In this section, we propose protocols for secure positioning in the BSM. We shall build upon the primitives described in Section 4. To make the intuition clear, we first give a secure positioning protocol for 1-dimension.

5.1 Secure Positioning in 1-dimension

For 1-dimension, we employ two verifiers, denoted by V_1 and V_2 (which send messages with the speed of radio waves). We assume that the position P being claimed by the prover is located between V_1 and V_2 . Our protocol is secure against an arbitrary number of adversaries colluding together to prove a position P , as long as the total information that these adversaries can store during the protocol is bounded. We let βn denote the aforementioned storage bound. Verifier V_1 is assumed to possess a random source X_1, X_2, \dots which is a reverse block entropy source of minimum entropy rate $\delta + \beta$, where $X_i \in \{0, 1\}^n$.

We shall use a (ε, ψ) -secure BSM entropy generator $\text{EG}: \{0, 1\}^n \times \{0, 1\}^\ell \rightarrow \{0, 1\}^m$ as discussed in the previous section. We choose the input size ℓ such that $\varepsilon + 2^{-\psi}$ is negligible in the security parameter κ . An example of a fast BSM EG, which is just a random sampler requiring no computations at all, is presented in the full version.

Before the protocol starts, the verifier V_1 selects a key $K \xleftarrow{R} \{0, 1\}^\ell$ and sends it to verifier V_2 over the private channel (using a private multicast message). Let t and t' be the time taken for radio waves to reach P from V_1 and V_2 respectively. Verifier V_1 sends out X from the reverse block entropy source such that X has min-entropy $(\delta + \beta)n$. At the same time, V_1 computes $\text{EG}(X, K)$ and stores it on its output tape. Let T be the time at which X reaches P . Verifier V_2 sends the key K out at a time such that it *meets* X at time T at the position P . More precisely, X and K are sent at times $(T - t)$ and $(T - t')$ by V_1 and V_2 respectively.

At time T , the prover claiming to be at position P evaluates $y = \text{EG}(X, K)$ and sends it back to the verifier V_1 . Verifier V_1 verifies that the string y is received at time $(T + t)$ and that it equals $\text{EG}(X, K)$. If these verifications succeed, the position claim of the prover is accepted and it is assumed to be indeed at position P . Otherwise, the position claim is rejected.

The protocol clearly satisfies the completeness property since an honest prover at position P will have both X and K available at time T and hence it can compute y (by asking the hypothetical ITM P_{env} to compute the function $\text{EG}(\cdot, K)$.) and report it back to V_1 by time $(T + t)$. We discuss the security below:

Theorem 2 *The 1-dimension secure positioning protocol is secure against an arbitrary number of adversaries colluding together, with the total adversary information storage bounded by βn .*

Proof. Suppose there exists an adversarial strategy with which a set of adversaries, none of which is at position P , are able to report back the correct y to the verifier V_1 at time $(T + t)$ with a non-negligible probability in the security parameter. We show that the above contradicts the properties of the EG.

We consider the state of the system at time T . X and K are at position P . Let there be g adversaries between V_1 and P and the information they have retrieved about X be S_1, S_2, \dots, S_g respectively. Let S denote the combined information $S_1 \cup S_2 \cup \dots \cup S_g$. Clearly since K has not yet crossed P , S is an arbitrary function of X alone. Further, $|S| \leq \beta n$ since βn is the total storage bound. Now we have the following:

Lemma 1. *The string y to be sent to the verifier V_i at time $(t + T)$, can be an arbitrary function of S and K alone. More formally, given an adversarial strategy to compute y in our setting, there exists a simulator that outputs y only given S and K (and not the stream X).*

The above lemma holds because (a) S is the only information stored by the adversaries between V_1 and P , (b) there is no adversary at P , and, (c) any information about X between P and V_2 at time T cannot reach V_1 by time $(t + T)$.

Hence we have $y = A(S, K)$, where $A(., .)$ is any arbitrary adversarial algorithm. However, given S and K , using properties of the BSM EG, the probability of an adversary correctly guessing y is upper bounded by $\varepsilon + 2^{-\psi}$. But $\varepsilon + 2^{-\psi}$ is negligible in the security parameter by our choice of ℓ . Thus we have reached a contradiction. \square

5.2 Secure Positioning in 3-dimensions

We generalize the above protocol to obtain a protocol for secure positioning in 3-dimensional space. βn is the total adversary information storage bound. We use 4 verifiers denoted by V_1, \dots, V_4 possessing reverse block sources of minimum entropy $(\delta + \beta)n$ that output strings X_i . Position P being claimed by the prover is enclosed in the tetrahedron defined by these 4 verifiers. t_i is the time taken for radio waves to reach the point P from verifier V_i . $\text{PRG}:\{0, 1\}^n \times \{0, 1\}^m \rightarrow \{0, 1\}^m$ is an ε -secure BSM pseudorandom generator. We choose the parameters such that $\varepsilon + 2^{-m}$ is negligible in the security parameter. In order for the verifiers to themselves compute the response expected from the prover, we first assume that verifiers can store the X_i values. We later show how this assumption can be removed. The protocol is illustrated in Figure 1. For more details, refer the full version.

The completeness follows from the fact that verifiers can compute K_4 from the stored X_i values and the prover can also compute K_4 since all the information

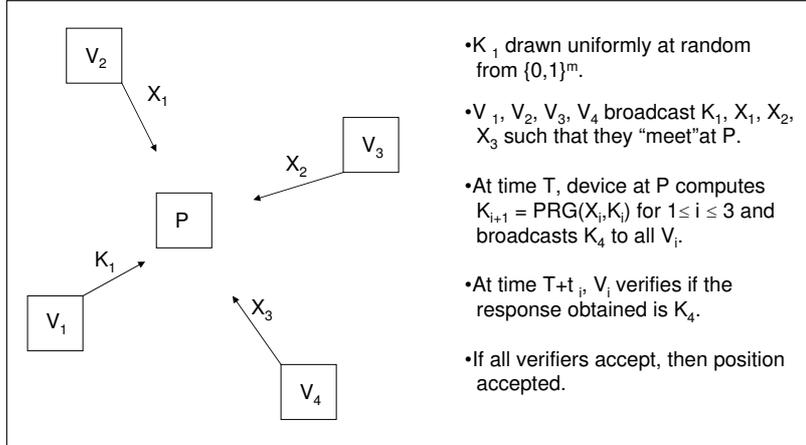


Fig. 1. Secure positioning protocol in 3-Dimensions

required is present jointly at P at time T . The security of this protocol is proven using techniques from the proof of security of the protocol for 3-dimensional position based key exchange that is discussed in Section 7 (note that position based key exchange implies a protocol for secure positioning).

We now describe, how to remove the assumption that verifiers can store strings drawn from their respective reverse block entropy sources. Note that the problem we face when verifiers cannot store the large strings is that verifiers have no way of verifying the response of the prover. This is because, when for example, V_3 broadcasts string X_2 , it does not know the key K_2 used to compute K_3 from X_2 . We get around this problem as follows. The verifiers pre-determine the keys K_1, K_2, K_3, K_4 that are to be used at every iteration of the application of the PRG. Now, the expected response of the prover, K_4 is known before protocol execution to all verifiers. The protocol is as follows:

1. V_1, V_2, V_3 and V_4 pick keys $K_1, K_2, K_3, K_4 \xleftarrow{R} \{0,1\}^m$ and broadcast them over their private channel.
2. V_1 broadcasts key K_1 at time $T - t_1$. V_2 broadcasts X_1 at time $T - t_2$ and simultaneously also broadcasts $K'_2 = \text{PRG}(X_1, K_1) \oplus K_2$. Similarly, V_3 broadcasts $(X_2, K'_3 = \text{PRG}(X_2, K_2) \oplus K_3)$ at time $T - t_3$ and V_4 broadcasts $(X_3, K'_4 = \text{PRG}(X_3, K_3) \oplus K_4)$ at time $T - t_4$.
3. At time T , the prover at position P computes messages $K_{i+1} = \text{PRG}(X_i, K_i) \oplus K'_{i+1}$ for $1 \leq i \leq 3$. The prover returns K_4 to all verifiers.
4. All verifiers check that the string K_4 is received at time $(T + t_i)$ and that it equals the K_4 that they pre-picked. If these verifications succeed, the position claim of the prover is accepted and it is assumed to be indeed at position P . Otherwise, the position claim is rejected.

The completeness of this protocol is as follows. Note that since the verifiers picked K_4 before the execution of the protocol, they can verify the response of a prover without storing any of the large random strings. To informally argue security of the protocol, note that in this protocol, instead of using the output of the PRG as an input key in the next round, one treats the output as one secret share of the key to be used. The other share of this key is broadcast in the clear. Now, if one of the shares of an additive secret sharing scheme is random, then the secret is hidden. Hence, by the security of the protocol in which verifiers could store the large random strings, it follows that this protocol is also secure.

6 Computational Position based Key Exchange

Informally, position based key exchange should have the property that if there is a prover at the claimed position P , then at the end of the protocol, the verifiers should share a uniform key K with it while for a group of colluding adversaries (none of whom is at P) K should look indistinguishable from a key drawn uniformly at random. This also implies that in the absence of a prover at position P , such a group of adversaries should be unable to execute the key exchange protocol on their own to obtain a shared key with the verifiers. In the full version [7], we show how to compile any 1-round *information theoretically* secure positioning protocol SP in our bounded storage model along with any unauthenticated key-exchange protocol KE to obtain an authenticated computational position based key exchange protocol CKE in the BSM.

7 Information theoretic Position based Key-Exchange

In this section, we present an information theoretic protocol to achieve position based key exchange. The overview of our protocol can be found in Figure 2. We start with some intuition behind our protocol and the techniques required to prove its security. Let us first consider the case of one dimension. We extend the protocol for secure positioning in one dimension presented earlier for the case of key exchange as follows. Instead of only one verifier V_2 sending a “large” string (drawn from a reverse block entropy source), both the verifiers send one large string each. More precisely, the verifier V_1 sends a key K_1 and a large string X_2 while the verifier V_2 sends a large string X_1 such that all of them meet at the claimed position P at the same instance of time T . The computation of the final key K_3 is done by the prover as follows: set $K_2 = \text{PRG}(X_1, K_1)$, $K_3 = \text{PRG}(X_2, K_2)$.

To see the intuition behind why this protocol is a secure one dimensional information theoretic position based key exchange, let us consider the state of the system at time T . Adversaries between V_1 and P (say, adversaries of type I) have stored $(K_1, A(X_2, K_1))$ while adversaries between P and V_2 (say, adversaries of type II) have stored $A(X_1)$. After time T , the adversaries of type I can compute K_2 thus transitioning their state to $(K_2, A(X_2, K_1))$ while adversaries of type II can only transition their state to $A(X_1), K_1, A(X_2, K_1)$. Thus it seems that

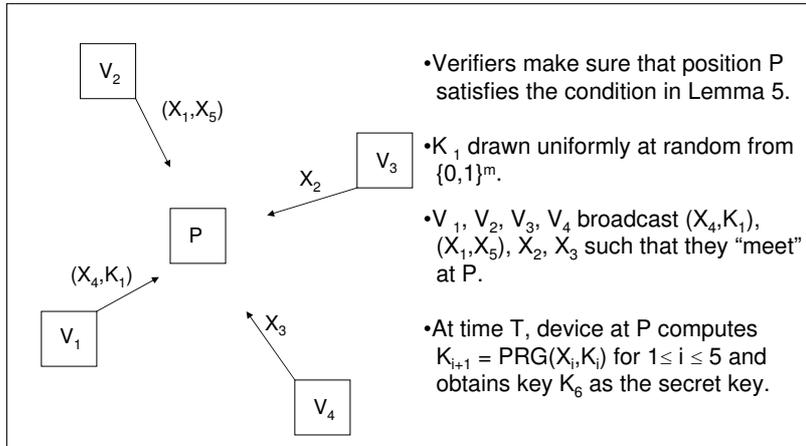


Fig. 2. Position based key exchange in 3-Dimensions

to both these types of adversaries together, the final key K_3 remains uniform. Indeed it turns out that this intuition is sound and the above is a secure one dimensional information theoretic position based key exchange protocol.

For three dimensions, we have the prover to be inside the tetrahedron defined by the four verifiers. Now, one can similarly try to extend the three-dimensional information theoretic secure positioning protocol presented earlier to achieve three-dimensional information theoretic position based key exchange. Simply add a fourth long string X_4 to be sent by V_1 in the natural way. However, it turns out that the above idea is not sufficient because of the fact that there might be adversaries (far) outside this tetrahedron trying to compute the key exchanged between the verifiers and an honest prover. In the case of secure positioning, such adversaries would be too late in sending their response to the verifiers (there is no honest prover to aid these adversaries). However, the key exchange scenario requires that once the verifiers and the honest prover get a shared key after running the protocol, this key should be uniform to the adversaries even at a much later point in time.

In contrast to what the intuition might suggest, the first problem we face is that there are certain regions in the tetrahedron defined by the verifiers such that if the claimed position P lies within one of these regions, there exists points, other than the point P , in the three dimensional space (but outside the tetrahedron) where the messages broadcast by the four verifiers all meet simultaneously. Thus, if there is an adversary located at such a point, it can compute the final key shared between the verifiers and the honest prover simply by following the algorithm of the honest prover. To overcome this problem, we characterize such regions of the tetrahedron (we further show that the remaining region is a still a large fraction of the tetrahedron) and exclude them from the area from which position claims are accepted (refer the full version for the Lemma characterizing

such regions and for further details). That is, given an area from which position claims need to be accepted, our lemma depicts the acceptable positioning of the verifiers so that they can verify the position claims from that area.

The second main problem that arises is that even if the messages broadcast by the verifiers do not all meet at a single point (other than P), there of course could be multiple colluding adversaries which utilize different information available at multiple different points at different time instances to try to compute the final key. Indeed, it can be shown that there is in fact an explicit attack on the protocol discussed earlier (that is, the protocol resulting from a natural extension of our three-dimensional secure positioning protocol where the verifiers broadcast four long strings) which allows multiple colluding adversaries to completely recover the key exchanged between the verifiers and an honest prover. To solve the above problem, we introduce a fifth long string in a similar way as before. Introducing this fifth long string allows us to construct a geometric argument, along with a reduction argument relying on techniques from [14], that multiple colluding adversaries do not have sufficient information, and our security proofs go through. Our final protocol is given in Figure 2. Our security proofs are a careful combination of the following two components:

- A geometric argument which rules out a “nice” way for adversaries to recover the final key exchanged. In other words, very roughly, there does not exist a strategy for multiple colluding adversaries to perform the operation $K_{i+1} = \text{PRG}_i(X_i, K_i)$ in sequence for each $i \in [5]$ to recover the final key K_6 .
- A reduction argument relying on the techniques from [14] to prove the final security of our protocol. In more detail, given the above geometric argument, if there exists an adversarial strategy that can distinguish the final key K_6 from uniform in our protocol, then we can construct an adversarial strategy to contradict the security guarantees of an intrusion resilient secret sharing scheme (as defined and constructed in [14]).

All details of our protocol and the security proofs are given in the full version [7] of this paper. The completeness of the above protocol described relies on the assumption that the verifiers can store the long strings they generated to be able to compute the final key K_6 themselves. In the full version, we show that, as with the case of secure positioning, this assumption can be relaxed by using the same secret sharing technique introduced in Section 5.

Acknowledgments. We thank Yevgeniy Dodis for interesting discussions.

References

1. Yonatan Aumann and Michael O. Rabin. Information theoretically secure communication in the limited storage space model. In *CRYPTO*, pages 65–79, 1999.
2. Stefan Brands and David Chaum. Distance-bounding protocols. In *EUROCRYPT '93: Advances in cryptology*, pages 344–359. Springer-Verlag New York, Inc., 1994.
3. Laurent Bussard. *Trust Establishment Protocols for Communicating Devices*. PhD thesis, Eurecom-ENST, 2004.

4. Christian Cachin and Ueli M. Maurer. Unconditional security against memory-bounded adversaries. In *CRYPTO '97: 17th Annual International Cryptology Conference, Advances in Cryptology*, pages 292–306, 1997.
5. Srdjan Capkun, Mario Cagalj, and Mani Srivastava. Secure localization with hidden and mobile base stations. In *IEEE INFOCOM*, 2006.
6. Srdjan Capkun and Jean-Pierre Hubaux. Secure positioning of wireless devices with application to sensor networks. In *IEEE INFOCOM*, pages 1917–1928, 2005.
7. Nishanth Chandran, Vipul Goyal, Ryan Moriarty, and Rafail Ostrovsky. Position based cryptography. Cryptology ePrint Archive, 2009. <http://eprint.iacr.org/2009/>.
8. Giovanni Di Crescenzo, Richard J. Lipton, and Shabsi Walfish. Perfectly secure password protocols in the bounded retrieval model. In *TCC*, pages 225–244, 2006.
9. Yan Zong Ding. Oblivious transfer in the bounded storage model. In *CRYPTO*, pages 155–170, 2001.
10. Yan Zong Ding. Error correction in the bounded storage model. In *TCC*, pages 578–599, 2005.
11. Stefan Dziembowski. Intrusion-resilience via the bounded-storage model. In *TCC*, pages 207–224, 2006.
12. Stefan Dziembowski. On forward-secure storage. In Cynthia Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 251–270. Springer, 2006.
13. Stefan Dziembowski and Ueli M. Maurer. On generating the initial key in the bounded-storage model. In *EUROCRYPT*, pages 126–137, 2004.
14. Stefan Dziembowski and Krzysztof Pietrzak. Intrusion-resilient secret sharing. In *FOCS '07: Proceedings of the 48th Annual IEEE Foundations of Computer Science*, 2007.
15. Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. One-time programs. In *CRYPTO 2008*, pages 39–56, 2008.
16. Jonathan Katz. Universally composable multi-party computation using tamper-proof hardware. In *EUROCRYPT 2007*, Lecture Notes in Computer Science, pages 115–128. Springer, 2007.
17. Chi-Jen Lu. Encryption against storage-bounded adversaries from on-line strong extractors. *J. Cryptology*, 17(1):27–42, 2004.
18. Ueli M. Maurer. Conditionally-perfect secrecy and a provably-secure randomized cipher. *J. Cryptology*, 5(1):53–66, 1992.
19. Tal Moran, Ronen Shaltiel, and Amnon Ta-Shma. Non-interactive timestamping in the bounded storage model. In *CRYPTO*, pages 460–476, 2004.
20. Naveen Sastry, Umesh Shankar, and David Wagner. Secure verification of location claims. In *WiSe '03: Proceedings of the 2003 ACM workshop on Wireless security*, pages 1–10, 2003.
21. Dave Singelee and Bart Preneel. Location verification using secure distance bounding protocols. In *IEEE Conference on Mobile Adhoc and Sensor Systems Conference*, 2005.
22. Salil P. Vadhan. Constructing locally computable extractors and cryptosystems in the bounded-storage model. *J. Cryptology*, 17(1):43–77, 2004.
23. Adnan Vora and Mikhail Nesterenko. Secure location verification using radio broadcast. In *OPDIS '04: 8th International Conference on Principles of Distributed Systems*, pages 369–383, 2004.
24. Yanchao Zhang, Wei Liu, Yuguang Fang, and Dapeng Wu. Secure localization and authentication in ultra-wideband sensor networks. *IEEE Journal on Selected Areas in Communications*, 24:829–835, 2006.