Position-Based Cryptography and Multiparty Communication Complexity

Joshua Brody[1], Stefan Dziembowski[2], Sebastian Faust[3], and Krzysztof Pietrzak[4*]

[1] Swarthmore College
[2] University of Warsaw
[3] Ruhr University Bochum
[4] IST Austria

Abstract. *Position based cryptography (PBC)*, proposed in the seminal work of Chandran, Goyal, Moriarty, and Ostrovsky (SIAM J. Computing, 2014), aims at constructing cryptographic schemes in which the identity of the user is his geographic position. Chandran et al. construct PBC schemes for *secure positioning* and *position-based key agreement* in the *bounded-storage model* (Maurer, J. Cryptology, 1992). Apart from bounded memory, their security proofs need a strong additional restriction on the power of the adversary: he cannot compute *joint* functions of his inputs. Removing this assumption is left as an open problem.

We first show that an answer to this question would resolve a long standing open problem in multiparty communication complexity: finding a function that is hard to compute with low communication complexity in the simultaneous message model, but easy to compute in the fully adaptive model.

On a more positive side: we also show some implications in the other direction, i.e.: we prove that lower bounds on the communication complexity of certain multiparty problems imply existence of PBC primitives. Using this result we then show two attractive ways to "bypass" our hardness result: the first uses the random oracle model, the second weakens the *locality* requirement in the bounded-storage model to *online computability.* The random oracle construction is arguably one of the simplest proposed so far in this area. Our results indicate that constructing improved provably secure protocols for PBC requires a better understanding of multiparty communication complexity. This is yet another intriguing example where *negative* results in one area (in our case: lower bounds in multiparty communication complexity) can be used to construct secure cryptographic schemes.

1 Introduction

The standard way to identify participants in cryptographic protocols is to check their knowledge of some secret data (like a password or a key), to verify some biometric information, or the possession of some hardware tokens. A new intriguing idea, known under the name of *position-based cryptography (PBC)* [15] is to construct algorithms in which the participating parties are identified by *their geographic position.* For example, consider the setting where we want to grant access to a server only to the personnel within some military base. A position-based system could be used to give access to every user that is physically located within the base, but deny it to everybody outside. There are many other examples one can think of where position-based authentication would be useful. Say, a protocol for sending confidential documents to everyone who is present in some conference room, granting WiFi access to people within some building, or checking if a food delivery was indeed ordered from some physical address. Of course, such protocols can be combined with other means of authentication, and hence they can also serve for providing an additional layer of security. See [15] for more on potential applications of this concept.

PBC protocols are typically based on the physical characteristics of wireless communication channels; concretely, they are based on the fact that electronic signals are traveling at the speed of light, denoted c (and hence traveling from point $\widehat{\mathcal{A}}$ to $\widehat{\mathcal{B}}$ takes time $\|\widehat{\mathcal{A}\mathcal{B}}\|/c$, where $\|\cdot\|$ denotes the length of the segment $\widehat{\mathcal{A}\mathcal{B}}$). Thus, if a *verifier* $\mathcal{V}$ sends a message at time T and receives a reply from a *prover* $\mathcal{P}$ within time T', the verifier can be sure that this reply was sent by a machine that is positioned at distance no more than $c \cdot (T' - T)/2$. A natural idea would be to

[*] Research supported by the European Research Council, ERC consolidator grant (682815 - TOCNeT).

exploit this fact, and use some standard trilateration techniques (like the one used in the GPS system) by having a group of verifiers $\mathcal{V}_1, \ldots, \mathcal{V}_n$ positioned in space and letting them jointly verify the distance from the prover. Unfortunately, as shown by Chandran et al. [15], the problem of designing PBC protocols is harder than it may seem at the first sight. In fact, they show that in the so-called *vanilla model* (i.e. without any additional assumptions), PBC is impossible: There exists an adversarial strategy which places devices around some point $\widehat{\mathcal{A}}$, and these devices can jointly convince the verifiers in any PBC protocol that they are at point $\widehat{\mathcal{A}}$, thus breaking the scheme. One way to get around this would be to restrict the number of adversary's devices (as the number of devices required in their attack is as large as the number of the verifiers used in the protocol). This however is not very realistic, as deploying several adversarial devices is usually easy in practice, since modern wireless devices are cheap and small.

The idea of Chandran et al. is to use Maurer's bounded-storage model [26], studied in a number of papers, e.g., [26,3,10,19,36,25,27]) (see its variant called the bounded-*retrieval* model [16,12,20]). In this model, it is assumed that the users of cryptographic protocols have short time access to a long random string X that is so large that it cannot be stored by the adversary in its entirety. The only thing that the adversary can do is to compute and store some function adv on X (where $|\mathsf{adv}(X)| \leq \xi|X|$, for a constant $0 \leq \xi \ll 1$). On the other hand, the honest parties of a protocol should be only required to access small parts of X in order to complete the protocol. The way this model is used in [15] is as follows: it is assumed that there is a group of *verifiers* $\mathcal{V}_i$ positioned in space. Suppose that a *prover* $\mathcal{P}$ claims to be at some position $\widehat{\mathcal{P}}$. Each of the verifiers broadcasts a long string X_i in such a way that all the X_i's arrive at $\widehat{\mathcal{P}}$ at the same time T. When this happens, the prover computes some function f on the X_i's, and takes some actions that depend on the computed value (e.g. sends the computed value back to the verifiers in order to prove that he is in point $\widehat{\mathcal{P}}$). The function f should be very efficiently computable. In particular, to compute it one should only have to access a small fraction of the X_i's [15].

In this model Chandran et al. construct a *positioning protocol*, where a prover convinces the verifiers that he is physically at some point $\widehat{\mathcal{P}}$. In practice a protocol like this is not very useful as a standalone primitive, since it comes with no guarantee that any future communication will be happening with the machine that is indeed in $\widehat{\mathcal{P}}$ (due to man-in-the-middle attacks). Chandran et al. also construct a more advanced primitive, called a *position-based key-agreement protocol*. Here the final output of the honest parties is a key K which is not known to a potential adversary. Both the positioning and the position-based key agreement protocols have a very simple structure (see Sect. 2.3). Namely, in case of the positioning protocol the prover just sends back $f(X_1, \ldots, X_n)$ to the verifiers (who check if this value is correct and was received at the right time). For the position-based key agreement the prover simply lets the agreed key K be equal to $f(X_1, \ldots, X_n)$. Such protocols are called *one-round*, and are very attractive because of their simplicity. They will also be the focus of this paper.

The proof in [15] requires one additional restriction on the power of the adversary, namely, it is assumed (see [15], page 1294, Sect. 1.2) that whenever an adversarial device receives strings $X_{i_1}, \ldots, X_{i_a}$ at the same time, it cannot compute an arbitrary joint function adv on $X_{i_1}, \ldots, X_{i_a}$ (with short output). Instead, it can only compute several (adaptively chosen) functions on each X_{i_j} independently (the same restriction applies to the honest parties). Removing this assumption is left as an "important open problem" in [15] and studying this open question is the main topic of this work. We show deep connections between the problem of constructing positioning and position-based key agreement protocols in the *unrestricted BRM* model (i.e. without restrictions on the adv function except of a bound on its output size), and the area of *multiparty communication complexity*. Before describing our contribution in more detail (in Sect. 1.2) let us provide a short introduction to this area (more formal definitions are given in Sect. 2.2, and for a more comprehensive introduction see [24])

1.1 Multiparty communication complexity

In a typical communication complexity problem, there are k players, denoted $\textsc{plr}_1, \ldots, \textsc{plr}_k$. There are also k inputs $x_1, \ldots, x_k \in \{0,1\}^n$, and the players must communicate to compute some function $f(x_1, \ldots, x_k)$ of the inputs. The *communication cost* of a protocol is measured as the worst-case maximal number of bits communicated, taken over all possible inputs and all choices for the random string.

In the multiplayer setting (when $k > 2$) there are two different models for how the input is shared. In the number-in-hand (NIH) model, each player $\textsc{plr}_i$ sees the ith input x_i. In the number-on-the-forehead (NOF) model, each $\textsc{plr}_i$ sees all inputs *except* x_i. One can imagine in an NOF protocol that all players meet in a room, and $\textsc{plr}_i$ has x_i written on her forehead. In this way, players can see all inputs *except* what is written on their foreheads.

When $k = 2$, the NIH and NOF models are one and the same, but for $k > 2$, they are quite different. In particular, communication in the NOF model becomes intuitively very easy, because so much information is shared. This makes proving NOF communication lower bounds *harder.* In this paper, we focus on NOF communication complexity.

It is particularly interesting is to understand what role *interaction* plays in communication complexity. In an arbitrary ("fully adaptive") protocol, players are allowed to speak back and forth, and messages are broadcast. It is also interesting to consider a more restrictive model, where each player sends a single message to a referee, who does not see the inputs, and must compute $f(x_1, \ldots, x_k)$ only from the messages sent by the players. This restricted model of communication is called the Simultaneous Messages (SM) model. Occasionally, the communication complexity of problems can be the same in the SM and interactive model, but for other problems, allowing interactive communication can even lead to an exponential decrease in the communication complexity. The NOF communication model was invented thirty years in [14], who also gave as an application lower bounds for branching programs.

Position based cryptography was partly inspired an the area called *secure positioning* [7,33,37,11]. More recently there was work towards constructing PBC protocols based on other "physical" assumptions, such as quantum channels [9,35,13][5] (see also [8] and the webpage [34]) or noisy channels [21].

1.2 Our contribution

We show that constructing a one-round positioning protocol in the unrestricted BSM gives a construction of a function π with linear SM complexity (in the NOF model). If we additionally require that the computation on the prover is *local* (i.e. he only needs to look at small parts of the input), then π has low complexity in the fully adaptive model. Finding a function with such properties is a longstanding open problem in communication complexity, and therefore this result can be viewed as a "negative" answer to the question posted in [15].

On a more positive side: we show some implications in the other direction. Namely, we prove that any protocol that has high communication complexity in the so-called "1-round almost adaptive SM model" (see Sect. 1.1 for the definition) can be transformed into a secure positioning protocol. The assumed hardness has to hold in a strong, randomized sense, i.e., the probability that any "adaptive SM" protocol computes the output correctly has to be negligible. Fortunately, we show a function that satisfies this requirement. Our function uses a hash function as a building block, and the security proof models this hash function as a random oracle (hence, our construction does not contradict the negative result mentioned above). The resulting positioning protocol is very simple: essentially, one verifier sends a long string X, the other verifiers send much shorter strings Z_i, and the output is the sub-string of X on the positions determined by the hash of the concatenated Z_i's.

[5] Note that [35] uses the random oracle model, that we use in this work (in Sect. 4.1).

We also construct positioning and position-based key agreement schemes from any function that has high complexity in the "2-round SM model" (see Sect. 1.1), which is a less standard notion (but it is weaker than the fully adaptive model). For our construction to work we need to assume even stronger hardness: the output of the function has to be "close to uniform" (in the sense of "statistical distance", see Sect. 2 for the definition). We show that the so-called "generalized inner product" function has this property. The resulting protocol does *not* have the "locality" property, i.e., the prover in the protocol needs to read its entire input. The good news is that this computation is very simple, can be performed very efficiently in an "online" fashion, and hence it may still be possible to implement it in practice.

2 Preliminaries

Let A and B be random variables distributed over set $\mathcal{A}$. The *statistical distance between A and B* is defined as $\Delta(A; B) := \frac{1}{2}\sum_{a\in\mathcal{A}} |\mathbb{P}(A = a) - \mathbb{P}(B = a)|$. The *statistical distance of A from uniformity* is defined as $d(A) := \Delta(A; U_{\mathcal{A}})$, where $U_{\mathcal{A}}$ has uniform distribution over $\mathcal{A}$. The statistical distance of A from uniformity *conditioned on* B is defined as $d(A \mid B) = \Delta((A, B); (U_{\mathcal{A}}, B)$ (where $U_{\mathcal{A}}$ is uniform and independent from B). The *min-entropy* of a random variable W is defined as $\mathbb{H}_\infty(W) := -\log_2(\max_w \mathbb{P}[W = w])$. In other words, it is a negative binary logarithm of the maximal probability of guessing W. We will use the following fact that can be viewed as a chain-rule for the statistical distance from uniformity (see, e.g., [19], Lemma 3).

Lemma 1. *For any random variables $X_1, \ldots, X_n$, and Y we have that*

$$d(X_1, \ldots, X_n|Y) \leq \sum_{i=1}^{n} d(X_i|X_1, \ldots, X_{i-1}, Y).$$

We also have the following (see, e.g., [19], Lemma 1).

Lemma 2. *For every random variables X and Y taking values from $\mathcal{X}$ and $\mathcal{Y}$ (respectively) we have that* $\max_{\alpha:\mathcal{Y}\to\mathcal{X}} (\mathbb{P}(X = \alpha(Y))) \leq d(X \mid Y) + 1/|\mathcal{X}|$. *Moreover, if $\mathcal{X} = \{0, 1\}$, then* $2\max_{\alpha:\mathcal{Y}\to\mathcal{X}} (\mathbb{P}(X = \alpha(Y))) - 1 = d(X \mid Y)$.

We also have the following lemma whose proof appears in Appx. A.

Lemma 3. *For any variables X, Y, Z, and V we have that $d(X, Y \mid Z, V) \geq d(X \mid Z)$.*

We will sometimes state our results in asymptotic terms. In this case, our constructions will be parametrized by some *security parameter* t. We say that a function $\mu : \mathbb{N} \to \mathbb{R}$ is *negligible in* t (and write $\mu(t) \leq \mathsf{negl}(t)$) if $|\mu|$ approaches 0 faster than the inverse of any polynomial, i.e., for every $c \geq 1$ there exists t_0 such that for every $t > t_0$ we have $|\mu(t)| \leq t^{-c}$. If $\mathcal{A}$ is a finite set, then $A \leftarrow \mathcal{A}$ denotes the fact that A is sampled uniformly at random from $\mathcal{A}$. For a natural q the symbol $\mathsf{GF}(q)$ denotes the Galois field of order q. The "$||$" symbol denotes the concatenation of stings, and for $X = (X_1, \ldots, X_n) \in \mathcal{X}^n$ (for some set $\mathcal{X}$) and $i, j \in \{1, \ldots, n\}$ (such that $i \leq j$) by writing $X[i]$ we mean X_i, and by $X[i, \ldots, j]$ we mean $(X_i, \ldots, X_j)$.

We will use the *random oracle model* (ROM). In this model all the participants of the protocol (in our case: the prover, the verifiers, and the adversaries), have access to an interactive machine Ω (called a *random oracle*) that contains a function $H : \mathcal{X} \to \mathcal{Y}$ (for some sets $\mathcal{X}$ and $\mathcal{Y}$, where $\mathcal{Y}$ has to be finite). We assume that H is chosen uniformly at random from the space of all functions of type $H : \mathcal{X} \to \mathcal{Y}$. Every participant of the protocol can query Ω on any $x \in \mathcal{X}$. Each such a query is answered by Ω with $H(x)$.

If W takes values from $\{0, 1\}^n$ (for some n), then the *min-entropy rate of* W is defined as $\mathbb{H}_\infty(W)/n$. It is easy to see that the maximal min-entropy rate is 1 (attained when W is uniform), and the minimal min-entropy rate is 0 (when W is constant).

2.1 Guessing bits from "compressed" information

The following machinery will be needed in Sect. 4.1. Consider the following natural question. Suppose $X \leftarrow \{0,1\}^n$ is chosen uniformly at random. Let $\mathsf{compress} : \{0,1\}^n \to \{0,1\}^{\beta n}$ be any function that "compresses" X, i.e., such that $\beta < 1$. Let us ask what is the maximal probability that given $\mathsf{compress}(X)$ one can compute the substring consisting of t random positions in X? More precisely, let $\mathsf{guess} : \{1,\dots,n\}^t \times \{0,1\}^{\beta s} \to \{0,1\}^t$ be any function that tries to "predict" these bits. We ask what is the maximal (over $\mathsf{compress}$ and guess) probability that $\mathsf{guess}(R, \mathsf{compress}(X)) = (X[R_1],\dots,X[R_t])$, where $R = (R_1,\dots,R_a) \leftarrow \{1,\dots,n\}^t$ is random. This question was first answered by Nisan and Zuckerman [29]. In what follows, we use the presentation from [15] (which, in turn, is partly based on [36]). The following lemma can be derived (we do in Appx. B) from the discussion in Sect. 4.3 (page 1306) of [15].

Lemma 4 ([29,36,15]). *Let $\beta < 1$ be some fixed parameter. For every t take n such that $n > t$. Then for every $\mathsf{compress} : \{0,1\}^n \to \{0,1\}^{\beta n}$ and $\mathsf{guess} : \{1,\dots,n\}^t \times \{0,1\}^{\beta n} \to \{0,1\}^t$ and a uniformly random $X \leftarrow \{0,1\}^n$ and $R = (R_1,\dots,R_t) \leftarrow \{1,\dots,n\}^t$ we have that*

$$\mathbb{P}\left(\mathsf{guess}(R, \mathsf{compress}(X)) = (X[R_1],\dots,X[R_t])\right) \leq \mathsf{negl}(t). \tag{1}$$

2.2 Multiparty communication complexity

A brief introduction to the multiparty complexity was already given in Sect. 1.1. We now introduce more formally the concrete computation models that are used later in this paper. A *protocol* is a tuple $\textsc{prot} := (\textsc{plr}_1,\dots,\textsc{plr}_k,\textsc{ref})$ of players (modeled as Turing machines) that interact with each other. We assume that the protocol is the the *public coins* model, i.e., the players have access to some common source of randomness. The *input of the protocol* is a tuple $(x_1,\dots,x_k) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ (where $\mathcal{X}_i$'s are some sets). Informally speaking, the goal of the players is to jointly compute some function $f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \to \mathcal{Y}$ (where $\mathcal{Y}$ is some set). The models that are considered in the literature differ in terms of what access the players have to the input, and how can they communicate. The player $\textsc{ref}$ is called the *referee* and typically takes no input. In the number-on-the-forehead (NOF) model each $\textsc{plr}_i$ sees all inputs *except x_i*. We also impose some restrictions on the communication between the parties. We say that *the protocol* $\textsc{prot}$ *operates in t-round simultaneous message (SM) model* if the parties communicate as follows.

1. Every player $\textsc{plr}_i$ (for $i = 1$ to k) receives input $x_1,\dots,x_{i-1},x_{i+1},\dots,x_k$ (where each $x_i \in \mathcal{X}_i$), and the referee $\textsc{ref}$ receives no input.
2. The computation is structured in t rounds. In the jth round (for $j = 1$ to t) the following happens:
   For $i = 1,\dots,k$ every player $\textsc{plr}_i$ (for $i = 1,\dots,k$) broadcasts some value w_i^j, which is a function of his input variables and the messages broadcast by other players in the previous rounds.
3. Finally, $\textsc{ref}$ computes the output of the protocol, denoted $\textsc{prot}(x_1,\dots,x_k)$, that is a function of the values w_i^j that were broadcast by the $\textsc{plr}_i$'s during the computation.

We will only consider the cases when $t = 1, 2$, or is unbounded. We note that for the case of $t = 1$ our notion is equivalent to the classic SM model, while the case of $t = 2$ can be viewed as its generalization (which we need in Sect. 3.2). We comment about the case of t being unbounded in a moment. The *t-round almost adaptive SM model* [31] is the same as the t-round SM model, except that one of the players, $\textsc{plr}_k$, say, is the referee (and hence there is no need to specify $\textsc{ref}$ separately, and we can write $\textsc{prot} = (\textsc{plr}_1,\dots,\textsc{plr}_k)$). Compared to the t-round SM model the only difference is in Step 3, that in case of the t-round almost adaptive SM model becomes:

3'. $\textsc{plr}_k$ computes the output of the protocol, denoted $\textsc{prot}(x_1,\dots,x_k)$, that is a function of his own input $(x_1,\dots,x_{k-1})$ and the values w_i^j that were broadcast by the $\textsc{plr}_i$'s during the computation.

Observe that in case of the t-round almost adaptive SM model we can assume that the the message v_k^t (sent by PLR_k in the last round) is empty, since the only receiver of this message is PLR_k himself. If there is no restriction on the number t of rounds then we will call a t-round SM model a *fully adaptive model*. Note that in this case there is no difference between the SM and the almost adaptive SM version, and in particular we can assume that there is no referee, and the output of the protocol is produced by PLR_k.

For a protocol PROT the maximal *total* length of the v_i^j's (where the maximum is taken over all $(x_1, \dots, x_k) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$) is called the *communication complexity* of PROT. As explained above, we are mostly interested in the average-case complexity of the multiparty protocols.

Definition 1. *We say that a function $f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \to \mathcal{Y}$ is (s, ε)-*hard in the t-round SM model (or t-round almost adaptive SM model, or fully adaptive model) *if for every protocol* PROT *whose communication complexity is at most s, and that operates in the t-round SM model (or t-round almost adaptive SM model, or fully adaptive model, respectively), the probability that* PROT *computes f correctly is at most ε, i.e.,*

$$\mathbb{P}\left(\text{PROT}(X_1, \dots, X_k) = f(X_1, \dots, X_k)\right) \leq \epsilon, \tag{2}$$

where the probability is taken over $(X_1, \dots, X_k) \leftarrow \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ and the public randomness available to the players (the probability in Eq. (2) is called the correctness probability*).*

Observe that the adversary can always achieve $\epsilon = 1/|\mathcal{Y}|$. As we will be interested in protocols where ϵ is negligible, we will usually use $\mathcal{Y}$'s that are of size exponential in the security parameter t (e.g., $\mathcal{Y}$ can consist of bit strings of length t). We will also use a stronger notion of hardness that informally speaking requires that the information about $f(X_1, \dots, X_k)$ obtained by a referee in a multiparty protocol with communication complexity s is small.

Definition 2. *We say that a function $f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \to \mathcal{Y}$ is (s, ε)-*strongly-hard in the t-round SM model (or fully adaptive model) *if for every protocol* PROT *whose communication complexity is at most s, and that operates in the t-round SM model (or fully adaptive model, respectively) we have that*

$$d\left(f(X_1, \dots, X_k) \mid \{W_1^j, \dots, W_k^j\}_{j=1}^t\right) \leq \epsilon, \tag{3}$$

where the experiment in (3) consists of sampling $(X_1, \dots, X_k) \leftarrow \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ and the public randomness of the players, and each W_i^j is the message broadcast by PLR_i in the jth round.

It is natural to ask what is the relative strength of these models. Firstly, it is easy to see that every protocol in the t-round SM model is also a protocol in the t-round almost adaptive SM model. Moreover, obviously, every t-round protocol can also be viewed as a t'-round model for $t' \geq t$. To see why the notion defined in Def. 1 is at least as strong as the one from Def. 2, observe that, by Lemma 2, Eq. (3) implies that $\mathbb{P}\left(\text{PROT}(X_1, \dots, X_k) = f(X_1, \dots, X_k)\right) \leq 1/|\mathcal{Y}| + \epsilon$ (see Eq. (2)), which is small for large $\mathcal{Y}$ (and small ϵ).

2.3 Secure Positioning and the Position-Based Key Agreement

In this section we describe in details the model that was already informally discussed in Sect. 1 (for the full formal definition see [15]). A *secure positioning protocol in D dimensions* is a tuple $\Pi = (\mathcal{V}_1, \dots, \mathcal{V}_{D+1}, \mathcal{P})$, where the $\mathcal{V}_i$'s are the *verifiers* positioned in a D-dimensional space (and not lying on one $(D-1)$-dimensional hyperspace) and a $\mathcal{P}$ is a *prover*, positioned within the polytope determined by the verifiers. The protocol will be attacked be a set of adversaries $\{\mathcal{A}_1, \dots, \mathcal{A}_t\}$, each $\mathcal{A}_i$ positioned in place $\widehat{\mathcal{A}_i}$. The $\mathcal{V}_i$'s, $\mathcal{A}_i$'s, and $\mathcal{P}$ are modeled as randomized Turing machines. We also assume that the $\mathcal{A}_i$'s have access to the common public randomness.[6]

[6] This assumption is made in order to keep this model consistent with the model from Sect. 2.2. Clearly it can be done without loss of generality, as such common randomness can be easily sampled by one of the $\mathcal{A}_i$'s and sent to the other ones via a private channel.

We assume that all the machines are equipped with perfect clocks and that their computation takes no time. Each machine is aware of its own position in space (more formally: it gets it as an auxiliary input). The position of each verifier $\mathcal{V}_i$ is denoted by $\widehat{\mathcal{V}}_i$. The verifiers also get as input a position $\widehat{\mathcal{P}}$ where the prover "claims to be". Their goal is to check if he indeed is in this position. The decision (yes/no) of the verifiers is communicated at the end of the protocol by one of them ($\mathcal{V}_1$, say).

The only messages that are sent are of a broadcast type (i.e. there are no directional antennas). A message sent by a machine positioned in point U arrives to a machine in point U' in time $\|UU'\|/c$, where c is the speed of light. We assume that the adversary cannot block or delay the messages sent between the honest participants. It is clear that such an assumption is unavoidable, as, by blocking all the messages, the adversary can always prevent any protocol from succeeding. The communication links between the verifiers are secure (secret and authenticated). Obviously, this can be achieved by standard cryptographic techniques.

As already highlighted in Sect. 1, the important difference between our model and the one of [15] is that we assume that if in some moment T several messages $X_{i_1}, \ldots, X_{i_\ell}$ meet at point $\widehat{\mathcal{A}}_i$, then $\mathcal{A}_i$ can compute any joint function adv_i^T of $(X_{i_1}, \ldots, X_{i_\ell})$. Let A_i^T be the result of this computation, and let A be the random variable denoting the concatenation of all the A_i^T. We require that $|A| \leq s$, where s is called the *retrieval bound*. Informally speaking, the adversary can either broadcast A_i^T or store it in his memory, but to keep the model as simple as possible we will make no distinction between these two cases. Namely, we assume that (1) each adversary always broadcasts every value immediately after he computed it, (2) each adversary stores all the messages that he sent, and (3) each adversary stores every message broadcast by any verifier.[7] Hence function adv_i^T can depend on all the adversarial messages received by $\mathcal{A}_i$ at time T the latest (including the messages sent by $\mathcal{A}_i$ himself in time T). We say that *Π is an (s, ρ)-secure positioning protocol* if the following two conditions hold:

correctness: If the prover $\mathcal{P}$ is placed in the claimed position $\widehat{\mathcal{P}} \in \mathcal{G}$ then $\mathcal{V}_1$ produces as output "yes",

security: Suppose the prover is not in position $\widehat{\mathcal{P}}$. Then any set of adversaries $\{\mathcal{A}_1, \ldots, \mathcal{A}_t\}$ with retrieval bound s the verifier $\mathcal{V}_1$ produces as output "yes" with probability at most ρ. (If $\mathcal{V}_1$ produced "yes" then we say that the adversaries *broke the scheme*.)

Following [15], we also consider a stronger type of protocols called the *position-based key agreement*. In such a protocol the goal of the prover and the verifiers is to agree on a key $K \in \{0,1\}^m$. More formally, at the end of the execution the prover produces as output $K_{\mathcal{P}}$, and one of the verifiers, $\mathcal{V}_1$ (say) produces $K_{\mathcal{V}}$. We say that *Π is an (s, ρ)-secure position-based key agreement protocol in D dimensions* if the following two conditions hold (assuming the prover $\mathcal{P}$ is placed in the claimed position $\widehat{\mathcal{P}} \in \mathcal{G}$):

correctness: The agreed keys are identical, i.e., $K_{\mathcal{P}} = K_{\mathcal{V}}$.

security: For any set of adversaries $\{\mathcal{A}_1, \ldots, \mathcal{A}_t\}$ with retrieval bound s (such that no adversary is in position $\widehat{\mathcal{P}}$) we have that $d(K_{\mathcal{P}} \mid A) \leq \rho$ (recall that A is the random variable denoting all the information computed by the adversaries).[8]

For reasons explained in the introduction we are interested in protocols that have the following simple structure (let T be some moment in time):

1. Each $\mathcal{V}_i$ sends a message $X_i \leftarrow \mathcal{X}_i$ (where $\mathcal{X}_i$ is some set) to $\mathcal{P}$ in time $T - \|\widehat{\mathcal{V}}_i\widehat{\mathcal{P}}\|/c$ (in this way all X_i's arrive to $\mathcal{P}$ in time T).

[7] Observe that the assumptions (1)–(3) can be made without loss of generality, as storing the computed values does not affect the retrieval bound.

[8] In [15] the security of a key agreement is defined using the "indistinguishability" paradigm (cf. Def. 2.2 in [15]): no adversary, after learning A, should be able to distinguish $K_{\mathcal{P}}$ from a uniformly random key, with advantage larger than ρ. It is easy to see that these definitions are equivalent.

2. $\mathcal{P}$ computes $Y = \pi(X_1, \ldots, X_{D+1})$ (for some function $\pi : \mathcal{X}_1 \times \cdots \times \mathcal{X}_{D+1} \to \mathcal{Y}$) and
   - **in case of the positioning protocols:** $\mathcal{P}$ broadcasts Y,
   - **in case of the key-agreement protocols:** $\mathcal{P}$ sets $K_{\mathcal{P}} = Y$.
3. In the last step the verifiers compute $\pi(X_1, \ldots, X_{D+1})$ in some way (e.g., they may simply send to one of the all the inputs and let him compute the output). The details of this computation depend on the function that they compute. In many cases there also exist techniques that allow to save on the communication and space complexities of this procedure, e.g., each X_i can be generated pseudorandomly from some seed S_i, in which case it is enough that the verifiers store and send to each other only the S_i's. We write more about it when we consider the concrete implementations in Sect. 4.
   - **in case of the positioning protocols:** each $\mathcal{V}_i$ accepts the proof only if y that he received is indeed equal to $\pi(X_1, \ldots, X_{D+1})$ and it arrived to him in time $T + \|\widehat{\mathcal{V}}_i\widehat{\mathcal{P}}\|/c$,
   - **in case of the key-agreement protocols:** the verifier $\mathcal{V}_1$ produces $K_{\mathcal{V}} = \pi(X_1, \ldots, X_{D+1})$ as the agreed key.

A protocol of this type will be called a *one-round protocol parametrized by* π. We say that a protocol is *for positions in the set* $\mathcal{W} \subseteq \mathbb{R}^D$ if it works only if $\widehat{\mathcal{P}} \in \mathcal{W}$ (note, however, that we do *not* restrict the set of positions where the adversary can be placed). Let us also comment on the assumption that X_i is sampled uniformly from some set. This is done mostly for the sake of simplicity, and to keep our model consistent with the one in Sect. 2.2. We could also have more general definition where the X_i's would come from some more general class of distributions, e.g., the distributions with high min-entropy (as it is done in [15]). For the equivalence results shown in Sect. 3 to hold, we would need to extend the hardness definitions in Sect. 2.2 to cover also the case when the X_i's are not uniform, but this can be done in a straightforward way. Also our constructions can be easily generalized to cover the case when the inputs come from a high-min entropy source (this generalization will be described in the full version of this paper).

It is natural to ask how do these two primitives relate to each other. Obviously, every (s, ρ)-secure position-based key agreement protocol can be converted into an (s, ρ')-secure positioning protocol with $\rho' = 2^{-|K|} + \rho$ in the following way: let the prover send $K_{\mathcal{P}}$ to $\mathcal{V}_1$, and let $\mathcal{V}_1$ output "yes" only if $K_{\mathcal{P}} = K_{\mathcal{V}}$. It easily follows from Lemma 2 that if $\mathcal{P}$ is not in the position $\widehat{\mathcal{P}}$ then the probability that he can guess $K_{\mathcal{P}}$ is at most ρ'.

On the other hand, it is also possible to convert every secure (s, ρ)-secure positioning protocol (for some negligible ρ) into an (s, ρ')-secure position-based key agreement protocol (for negligible ρ'), at a cost of introducing computational assumptions. We refer to [15] (Section 6, page 1311) for further details.

2.4 Prover's efficiency

The function π needs to be computed also by the prover $\mathcal{P}$, and it is important to choose π such that this computation can be done efficiently. Note that the advantage of $\mathcal{P}$ over the adversaries is that he has simultaneous access to all the π's inputs $X_1, \ldots, X_{D+1}$. Since the X_i's are very long, we would ideally like to be able to compute π by looking only on some small parts of the inputs (polylogaritmic in $|X|$, say). This property, called *locality*, was stated as an explicit requirement in [15]. It is also common in the previous papers on the bounded-storage model [26,3,20,18]. One of our constructions in this paper (see Sec. 4.2) does not have this property (the one in Sect. 4.1 has it). Instead it has the property of being *online computable* which means that π reads its input by just processing its input online in small memory. We remark that in some cases such algorithms may actually be easier to implement than some of the locally computable ones (think of a locally computable algorithm that is required to access many bits on its input that are located far away).

3 The reductions

In this section we show strong connections between the two areas described in Sect. 2. We start (Sect. 3.1) with showing that a construction of a positioning protocol immediately gives a construction of a function with a high 1-round SM complexity. Note that this means that a similar implication holds for position-based key agreement (since, as explained in Sect. 2.3, position-based key agreement is a stronger primitive than secure positioning). Then, in Sect. 3.2, we show an implication in the opposite direction, namely, we prove that every function with high 1-round almost adaptive SM complexity gives rise to a secure positioning protocol, and every function with high 2-round SM complexity gives rise to a secure position-based key agreement protocol.

From the point of view of the position-based cryptography applications, the results in Sect. 3.1 can be viewed as "negative", because they show that in order to construct secure positioning protocols (and they position-based key agreement protocols) we need to show multiparty functions that have high communication complexity, which seems to be non-trivial, especially if the locality is required (see end of Sect. 3.1 for a discussion on this). On the other hand, the results from Sect. 3.2 can be viewed as "positive" ones, since they provide a way to construct secure positioning (and position-based key agreement) protocols. Notice that these positive results yield a constructive use of lower bounds in communication complexity. We instantiate these constructions with concrete protocols is Sect. 4.

3.1 Secure positioning implies lower bounds for SM complexity

We now show that existence of a one-round protocol for secure positioning implies lower bounds for the multiparty communication complexity. Note that, as described in Sect. 2.3, the secure positioning protocols are a weaker primitive than the position-based key agreement protocols, and a similar implication also holds for the position-based key agreement. To keep the exposition simple we address only the case when the verifiers are placed on vertices of a regular D-dimensional simplex, but it should be clear that our argument can be easily extended to more general cases. The statement of the lemma assumes that $D = 2$ or $D = 3$. This is because, obviously, the case of $D > 3$ has no practical relevance, and for $D = 1$ the function π has only two arguments, so, as described in the introduction, it makes little sense to talk about the NOF complexity. Recall that a regular 2-dimensional simplex is an equilateral triangle, and a regular 3-dimensional simplex is a regular tetrahedron. We now have the following lemma.

Lemma 5. *Suppose Π is an (s, ρ)-secure one-round protocol in D dimensions (for $D = 2$ or $D = 3$) parametrized by $\pi : \mathcal{X}_1 \times \cdots \times \mathcal{X}_{D+1} \to \mathcal{Y}$ with verifiers positioned on vertices of a regular D-dimensional simplex. Then π is $(s/2, \rho)$-hard in the 1-round SM model.*

Proof. Let a denote the length of the edge of the simplex, or, in other words, the distance between any pair of verifiers. For the sake of contradiction suppose π can be computed in a 1-round SM model by a protocol $\textsc{prot} = (\textsc{plr}_1, \ldots, \textsc{plr}_{D+1}, \textsc{ref})$ with communication complexity $s' = s/2$ and correctness probability $\rho' > \rho$. For every $\textsc{plr}_j \in \{\textsc{plr}_1, \ldots, \textsc{plr}_{D+1}\}$ let $\mathrm{Msg}_j(X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_{D+1})$ denote the message computed by $\textsc{plr}_i$, and let $\mathrm{Ref}(y_1, \ldots, y_{D+1})$ be the value computed by the referee $\textsc{ref}$ (equal to $\pi(X_1, \ldots, X_{D+1})$ with probability ρ'). We now show a set of adversaries $\mathcal{A}_1, \ldots, \mathcal{A}_{D+1}, \mathcal{B}_1, \ldots, \mathcal{B}_{D+1}$ with retrieval bound s that break Π with probability ρ' (and none of them is positioned in position $\widehat{\mathcal{P}}$).

We assume that position $\widehat{\mathcal{P}}$ is the center of mass of the simplex determined by the verifiers. Hence, $\widehat{\mathcal{P}}$ is in the same distance to all the verifiers, and therefore all the messages X_i are sent in the same moment $U = T - \|\widehat{\mathcal{P}}\widehat{\mathcal{V}}_1\|/c$, where (as it can be easily verified using basic geometric arguments) $\|\widehat{\mathcal{P}}\widehat{\mathcal{V}}_1\|$ is equal to $a\sqrt{3}/3$ (if $D = 2$) and is equal to $a\sqrt{6}/4$ (if $D = 3$). This situation is depicted on Fig. 1 for the case $D = 2$.

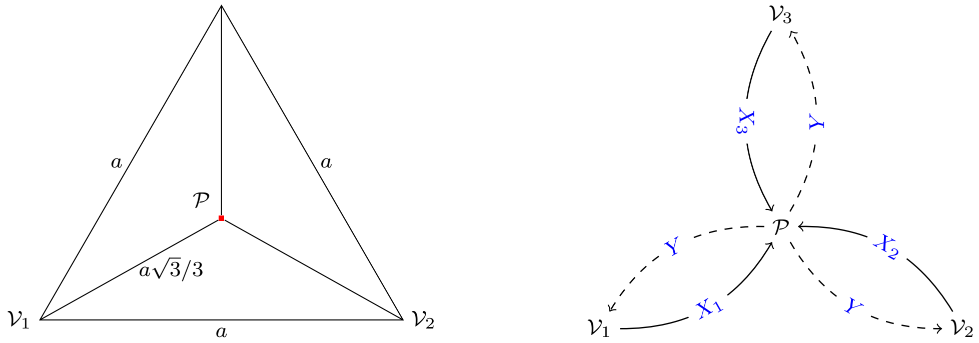

Fig. 1. On the left: the configuration of the prover and the verifiers for in the two-dimensional case. On the right: the execution of the positioning protocol in this configuration. The dashed lines indicate the messages sent back by the prover. Note that the X_i's and Y are broadcast (there are no directional antennas in our model), and the lines are only indicating the communication that matters for the protocol.

Obviously, all the verifiers expect to receive the answer from the prover in time $T + \|\widehat{\mathcal{P}}\widehat{\mathcal{V}}_1\|/c = U + 2\|\widehat{\mathcal{P}}\widehat{\mathcal{V}}_1\|/c$. The adversaries $\mathcal{A}_1, \dots, \mathcal{A}_{D+1}, \mathcal{B}_1, \dots, \mathcal{B}_{D+1}$ behave in the following way (see Fig. 2).

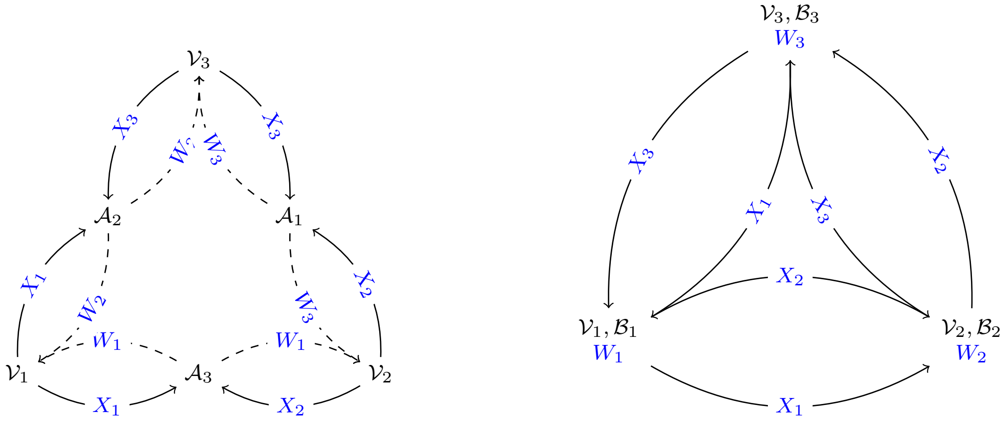

Fig. 2. On the left: the actions of the $\mathcal{A}_i$'s, on the right: the actions of the $\mathcal{B}_i$'s (recall that each W_i is a function of all the X_j's except of X_i).

- Each $\mathcal{A}_j$ is positioned in point $\widehat{\mathcal{A}}_j$ defined as follows: $\widehat{\mathcal{A}}_j$ is the center of mass of the facet determined by the points $\widehat{\mathcal{V}}_1, \dots, \widehat{\mathcal{V}}_{j-1}, \widehat{\mathcal{V}}_{j+1}, \dots, \widehat{\mathcal{V}}_{D+1}$. This facet is either a line segment — in case $D = 2$, or an equilateral triangle — in case $D = 3$. From the regularity of this facet we get that the messages $X_1, \dots, X_{j-1}, X_{j+1}, \dots, X_{D+1}$ (send by the verifiers $\mathcal{V}_1, \dots, \mathcal{V}_{j-1}, \mathcal{V}_{j+1}, \dots, \mathcal{V}_{D+1}$) arrive to point $\widehat{\mathcal{A}}_j$ in the same moment. In the moment when they arrive there, the adversary $\mathcal{A}_j$ computes $W_j = \mathrm{Msg}_j(X_1, \dots, X_{j-1}, X_{j+1}, \dots, X_{D+1})$ and broadcasts the result. This happens in time $U + \|\widehat{\mathcal{A}}_j\widehat{\mathcal{V}}_i\|/c$.
- Each $\mathcal{B}_i$ is positioned in point $\widehat{\mathcal{V}}_i$.[9] He does the following:

[9] The reader may object that it is not realistic to assume that an adversary is positioned at zero distance from a verifier. At the end of the proof we argue that $\mathcal{B}_i$ can actually be put at some place far from any verifier. We decided to assume that $\mathcal{B}_i$ is positioned exactly in point $\widehat{\mathcal{V}}_i$ to keep the exposition simple.

- When the messages $X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_{D+1}$ arrive to him (observe that, from the regularity of the simplex, they all arrive in the same moment $T' = U + a/c$) he computes $W_i := \mathrm{Msg}_j(X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_{D+1})$ and stores the result.
- He also stores each message W_j broadcast by $\mathcal{A}_j$ (for $j \in \{1, \ldots, i-1, i+1, \ldots, D+1\}$) when it arrives to him. This happens in time $T'' = U + \|\widehat{\mathcal{A}}_j \widehat{\mathcal{V}}_i\|/c + \|\widehat{\mathcal{A}}_j \widehat{\mathcal{B}}_i\|/c$ (where $\widehat{\mathcal{B}}_i$ is the position of $\mathcal{B}_i$, which is equal to $\widehat{\mathcal{V}}_i$). Hence T'' is equal to $U + 2\|\widehat{\mathcal{A}}_j \widehat{\mathcal{V}}_i\|/c$. Observe also that, by the triangle inequality $\|\widehat{\mathcal{A}}_j \widehat{\mathcal{V}}_i\| + \|\widehat{\mathcal{A}}_j \widehat{\mathcal{B}}_i\| \geq \|\widehat{\mathcal{A}}_j \widehat{\mathcal{V}}_i\| = a$, and therefore $T'' \geq T'$.
- After the two steps above are completed (which happens in time $\max(T', T'') = T''$) the adversary $\mathcal{B}_i$ knows all $W_1, \ldots, W_{D+1}$ and he can simply compute the output Y as $\mathrm{Ref}(W_1, \ldots, W_{D+1})$, and pass it $\mathcal{V}_i$ (which takes zero time, since $\mathcal{B}_i$ is positioned exactly in $\widehat{\mathcal{V}}_i$). Moreover, he can do it exactly in time $U + 2\|\widehat{\mathcal{P}}\widehat{\mathcal{V}}_1\|/c$ when $\mathcal{V}_i$ expects to receive y. This is possible, because (as we show below)

$$T'' < U + 2\|\widehat{\mathcal{P}}\widehat{\mathcal{V}}_1\|/c. \tag{4}$$

We now show (4). Let us start with case $D = 2$. Since in this case each facet of the simplex is a line segment of length a, hence $\|\widehat{\mathcal{A}}_j \widehat{\mathcal{V}}_i\| = a/2$. Therefore (4) becomes

$$U + a/c < U + 2a\sqrt{3}/(3c), \tag{5}$$

which holds because $1 < 2\sqrt{3}/3$. In case $D = 3$ each facet is a regular triangle with edge of length a. Thus $\|\widehat{\mathcal{A}}_j \widehat{\mathcal{V}}_i\| = \sqrt{3}/3$, and therefore (4) becomes

$$U + 2a\sqrt{3}/(3c) < U + 2a\sqrt{6}/(4c), \tag{6}$$

which holds because $2\sqrt{3}/3 < 2\sqrt{6}/4$. Clearly the adversaries constructed this way compute function π correctly with exactly the same probability σ' as the SM protocol computes it. It remains to calculate how much information was retrieved by the adversaries. Observe that each Msg_j was computed twice (by $\mathcal{A}_j$ and $\mathcal{B}_j$). Hence, the total amount of retrieved information if $2s' = s$. This finishes the proof.

Finally, note that both inequalities (5) and (6) are sharp, and the differences between the left hand sides and the right hand sides are non-negligible. This means that $\mathcal{B}_i$ has to wait some noticeable amount of time before he sends y to the verifier $\mathcal{V}_i$. Hence, it is also ok to place $\mathcal{B}_i$ in some position $\widehat{\mathcal{B}}_i$ further away from $\mathcal{V}_i$ (as long as the $\widehat{\mathcal{B}}_i$ is in equal distance to the remaining verifiers). □

Recall that according to the standard definitions (see Sect. 2.4) we want function π to be locally computable, which means that it should be possible to compute it by looking only at a polylogarithmic number of bits of its input $(X_1, \ldots, X_{D+1})$. It is easy to see that such an algorithm is trivial to implement by a multiparty protocol that has polylogarithmic communication complexity in the fully adaptive settings. On the other hand, function π, by Lemma 5, needs to have a linear complexity in the 1-round SM model. Since finding such functions is an open problem we view this result as an indication why showing 1-round positioning protocols in the unrestricted BRM model is hard. The reader may object that typically the communication complexity literature is more focused on deterministic functions that compute one bit, while here we consider randomized functions (with small correctness probability) with multi-bit output. This is not a problem for the following reasons: (1) it is easy to see that a lower bound on the communication complexity of our multi-bit output randomized function also implies a lower bound on a single-bit output functions (since there has to be at least one bit of output that is hard to guess with good probability), and (2) randomized lower bounds imply the deterministic ones.

3.2 Lower bounds for SM complexity imply results for PBC

In this section we show implications in the other direction than in Sect. 3.1, i.e., we show how to build positioning and position-based key agreement protocols from functions that have high communication complexity. Unlike in case of Sect. 3.1 we consider these two cases separately (the first one in Lemma 6), and the second one in Lemma 71. Although in principle the second construction would suffice for showing the general implication (as the key agreement is a stronger primitive than the positioning), such a separation makes sense, since the requirements for the communication complexity that we need in Lemma 6 are weaker (and hence Lemma 6 does not directly follow from Lemma 7). Also the conditions on the position of the prover $\mathcal{P}$ are more restrictive in Lemma 7.

Lemma 6. *Suppose $D \in \{2,3\}$. Let $\pi : \mathcal{X}_1 \times \cdots \times \mathcal{X}_{D+1} \to \mathcal{Y}$ be an (s,ρ)-hard function in the 1-round almost adaptive SM model. Let Π be a 1-round positioning protocol parametrized by π. Then Π is (s,ρ)-secure for positions within the D-dimensional simplex whose vertices are the positions of the verifiers $\widehat{\mathcal{V}}_1,\ldots,\widehat{\mathcal{V}}_{D+1}$.*

Proof. Set $k := D+1$. Let $\mathcal{P}$ be the prover, and $\mathcal{V}_1,\ldots,\mathcal{V}_{D+1}$ be the verifiers. Assume the position $\widehat{\mathcal{P}}$ of $\mathcal{P}$ is within the D-dimensional simplex whose vertices are the positions of the verifiers $\widehat{\mathcal{V}}_1,\ldots,\widehat{\mathcal{V}}_{D+1}$. For the sake of contradiction assume that Π can be broken by adversaries with retrieval bound s with probability $\rho' > \rho$. This means that one of the adversaries, $\mathcal{A}_1$, say, was able to send to the verifiers a message Y equal to $\pi(X_1,\ldots,X_k)$ with probability ρ' (assuming $(X_1,\ldots,X_k) \leftarrow \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$), and this message arrived to $\mathcal{V}_k$ in time $T + \|\widehat{\mathcal{V}}_k\widehat{\mathcal{P}}\|/c$. The value of Y is a result of computation that depended on some subset of the variables $X_1,\ldots,X_k$ and possibly some messages sent by the other adversaries (that, in turn, may depend on some other subset of variables, other messages, and so on). Hence we can view this message exchange as a multiparty protocol. From the assumption that π is (s,ρ)-hard in the 1-round almost adaptive SM model, it follows that Y could not be computed by a 1-round almost adaptive protocol with communication complexity s. Therefore at some point of computing Y one of the following had to happen:

1. a joint function α of $(X_1,\ldots,X_k)$ was computed, or
2. a function β was computed on some subset of variables $\mathcal{S} \subseteq \{X_1,\ldots,X_k\}$ and β was chosen adaptively after learning some function β' of another subset of variables $\mathcal{S}' \subseteq \{X_1,\ldots,X_k\}$, and $\mathcal{S}$ and $\mathcal{S}'$ are such that (a) $\mathcal{S} \cup \mathcal{S}' = \{X_1,\ldots,X_k\}$ and (b) $X_k \in \mathcal{S} \cap \mathcal{S}'$.

We consider only the second case, as it is clearly more general than the first one. Let $\widehat{\mathcal{B}}$ be the point in which β is computed. Since $X_k \in \mathcal{S}$ thus this computation happens exactly in $\|\widehat{\mathcal{B}}\widehat{\mathcal{V}}_k\|/c$ time after X_k was sent by $\mathcal{V}_k$, i.e. in time $T - \|\widehat{\mathcal{P}}\widehat{\mathcal{V}}_k\|/c + \|\widehat{\mathcal{B}}\widehat{\mathcal{V}}_k\|/c$. Communicating the result of this computation back to $\mathcal{V}_k$ takes time at least $\|\widehat{\mathcal{B}}\widehat{\mathcal{V}}_k\|/c$. Hence the reply arrives to $\mathcal{V}_k$ at earliest at time $T - \|\widehat{\mathcal{P}}\widehat{\mathcal{V}}_k\|/c + 2\|\widehat{\mathcal{B}}\widehat{\mathcal{V}}_k\|/c$. On the other hand, if $\mathcal{V}_k$ accepts this reply then it has to arrive to it exactly at time $T + \|\widehat{\mathcal{P}}\widehat{\mathcal{V}}_k\|/c$. We thus get the following inequality

$$T - \|\widehat{\mathcal{P}}\widehat{\mathcal{V}}_k\|/c + 2\|\widehat{\mathcal{B}}\widehat{\mathcal{V}}_k\|/c \leq T + \|\widehat{\mathcal{P}}\widehat{\mathcal{V}}_k\|/c,$$

which implies that $\|\widehat{\mathcal{B}}\widehat{\mathcal{V}}_k\|/c \leq \|\widehat{\mathcal{P}}\widehat{\mathcal{V}}_k\|/c$. This, in turn, means that β was computed in time T at the latest. On the other hand, at the moment when it was computed, the adversary B located in point $\widehat{\mathcal{B}}$ has already learned all the variables in $\{X_1,\ldots,X_k\}$ — this is because he learns the variables in $\mathcal{S}$ exactly in this moment and he learns the variables in $\mathcal{S}'$ even earlier (from the triangle inequality). Therefore for every $i \in \{1,\ldots,k\}$ we have that $T - \|\widehat{\mathcal{P}}\widehat{\mathcal{V}}_i\| + \|\widehat{\mathcal{B}}\widehat{\mathcal{V}}_i\| \leq T$, which implies that $\|\widehat{\mathcal{B}}\widehat{\mathcal{V}}_i\| \leq \|\widehat{\mathcal{P}}\widehat{\mathcal{V}}_i\|$. From a geometric argument used in [15] (see page 1310) we get that there is only one point $\widehat{\mathcal{B}}$ that satisfies it, namely $\widehat{\mathcal{B}} = \widehat{\mathcal{P}}$ (provided $\widehat{\mathcal{P}}$ lies within the simplex whose vertices are the positions of the verifiers $\widehat{\mathcal{V}}_1,\ldots,\widehat{\mathcal{V}}_{D+1}$). This completes the proof. □

We now show Lemma 7 that is similar to Lemma 6, but it holds for position-based key agreement. Observe that for the lemma to hold we need a slightly stronger assumption than in Lemma 6, namely that π is hard in the 2-round SM model. Also, unlike in Lemma 6, we do not specify explicitly what geometric configurations of the verifiers and the prover are allowed. Instead, we simply say that they need to be such that the messages sent by the verifier (see Sect. 2.3) never "meet" at any place other than the position $\widehat{\mathcal{P}}$ of the prover. More precisely, we require that there does not exist time U and place $\widehat{\mathcal{Z}} \neq \widehat{\mathcal{P}}$ such that at time U all the X_i's are in $\widehat{\mathcal{Z}}$. For the detailed analysis on what are the possible configurations of the parties in order for this assumption to holds we refer the reader to [15], Sect. 7.3.1.

Lemma 7. *Suppose $D \in \{2, 3\}$. Let $\pi : \mathcal{X}_1 \times \cdots \times \mathcal{X}_{D+1} \rightarrow \mathcal{Y}$ be an (s, ρ)-strongly-hard function in the 2-round SM model. Let Π be a one-round key-agreement protocol in D dimensions parametrized by π. Then Π is a (s, ρ)-secure key-agreement protocol assuming all the messages sent by the verifiers never meet at any other place than the position $\widehat{\mathcal{P}}$ of the prover.*

Proof. For the sake of contradiction suppose Π is not (s, ρ)-secure, i.e. there exists adversaries $\mathcal{A}_1, \ldots, \mathcal{A}_t$, each positioned in $\widehat{\mathcal{A}}_i, \ldots, \widehat{\mathcal{A}}_t$ (resp.), such that

$$d(\pi(X_1, \ldots, X_{D+1}) \mid A) = \rho' > \rho, \tag{7}$$

where $X_1, \ldots, X_{D+1} \leftarrow \mathcal{X}_1 \times \cdots \times \mathcal{X}_{D+1}$ are the input variables, and A is a concatenation of the outputs A_i^T of the adv_i^T functions computed by the adversaries when the protocol Π is executed on input $(X_1, \ldots, X_{D+1})$. To finish the proof we show an NOF protocol with communication complexity s such that

$$d(\pi(X_1, \ldots, X_{D+1}) \mid W) = \rho', \tag{8}$$

where W is a concatenation of the messages $W_1^1, \ldots, W_{D+1}^1, W_1^2, \ldots, W_{D+1}^2$ sent by the players when the NOF protocol is executed on variables $(X_1, \ldots, X_{D+1})$. Clearly, showing (8) will contradict the assumption that π is (s, ρ)-strongly-hard in the NOF model. Call the A_i^T the *intermediate variables* (as opposed to the *input variables* X_i).

We now recursively define the *dependence* relation among the intermediate variables. We say that (a) every A_i^T depends on itself, and (b) A_i^T *depends on* A_j^U if the adversary A_i at time T the latest received some other variable A_k^S that depends on A_j^U. Clearly the dependence relation is a partial order. We also say that A_i *depends on* an input variable X_j if either A_i is a function of an input variable X_j or it depends on some intermediate A_k that is a function of X_j.

The NOF protocol that we construct consists of the following phases:

- **Computation Phase 1:** Compute every A_i^T such that the set $\hat{\mathcal{X}}_i^T$ of input variables that it depends on is a *proper* subset of $\mathcal{X}$. Obviously, since this subset is proper hence there exists a player that knows all the variables in this subset, and hence each such A_i^T can be computed in this phase by some player.
- **Computation Phase 2:** Compute all the remaining A_i^T's.
- **Output Phase:** Output all the A_i^T's computed in the previous phases.

What remains to show is that all the variables that remained after Phase 1 can be computed in Phase 2. Suppose it is not the case. Then there exists an intermediate variable, call it B_3, that was not computed in the first two phases, but depends only on the variables that were computed in these two phases. Let $\mathcal{Q}_3$ be the set of variables that B_3 is a function of. Clearly B_3 has to depend on some other variable B_2 that was computed in Phase 2. Let $\mathcal{Q}_2$ be the set of input variables that B_2 is a function of. Of course B_3 cannot be a subset of $\mathcal{Q}_2$, as otherwise B_3 would be computed in Phase 2. Let x be any element of $\mathcal{Q}_3 \setminus \mathcal{Q}_2$. Since B_2 was computed in Phase 2 thus it has to depend on all input variables $\mathcal{Q}$ (otherwise it would have been computed in Phase 1), and hence it has to depend on some variable B_1 that is a function of x.

To summarize: B_3 and B_1 are both functions of x. Moreover B_3 depends on B_2 which, in turn, depends on B_1. The only way that could happen is that B_2 would lie on a line connecting B_3 and B_1, and B_2 would also be a function of x. Since $x \notin \mathcal{Q}_2$ we obtain contradiction.

□

4 Concrete constructions

In this section we provide two concrete constructions of positioning and position-based key agreement protocols. This is done using the theory developed in Sect. 3.2, i.e., we first prove that some function π has high communication complexity, and then use this function to construct a position-based protocol. We start with a construction of a positioning protocol that has the "locality" property (see Sect. 2.4), and works in the random oracle model. Note, that using the techniques from [15], this positioning protocol can be transformed into a position-based key agreement, using the computational assumptions (see Sect. 2.3). Then, in Sect. 4.2, we show a construction of a position-based key agreement in the plain model (i.e. without a random oracle assumption). This second construction comes without the locality property, i.e., the prover has to read the entire random strings X_i that are sent to him by the verifiers. On the other hand, it has the *on-line-computability* property, i.e., the X_i's need to be read only once in an on-line fashion, by an a machine with very small memory (see Sect. 2.4).

4.1 Protocols in the Random Oracle Model

As proven in Sect. 3.2 (see Lemma 6), to construct such a protocol it is enough to show a function $\pi : \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \to \mathcal{Y}$ whose 1-round almost adaptive SM complexity is high. Let t be a security parameter. We assume that the parties have access to t random oracles containing functions $\{H_j : \{0,1\}^* \to \{1,\ldots,m\}\}_{i=1}^t$ (let $\mathcal{H}$ denote this family of functions). The function π will depend on the functions in $\mathcal{H}$. Also every party will have access to the functions in $\mathcal{H}$. More concretely, let $\pi_{k,n}^{\mathcal{H},t} : (\{0,1\}^t)^{k-1} \times \{0,1\}^n \to \{0,1\}^t$ be a function defined as: $\pi_{k,n}^{\mathcal{H},t}(Z_1,\ldots,Z_{k-1},X) := (X[H_1(Z)],\ldots,X[H_t(Z)])\,,$ where $Z = (Z_1||\cdots||Z_{k-1})$.

Our positioning protocol $\mathit{\Pi}_{D,n}^{\mathcal{H},t}$ in D dimensions (for $D \in \{2,3\}$) is simply the one-round positioning protocol parametrized by $\pi_{D+1,n}^{\mathcal{H},t}$ (see Sect. 2.3). More concretely: it consists of $D+1$ verifiers $\mathcal{V}_1,\ldots,\mathcal{V}_{D+1}$ (positioned in $\widehat{\mathcal{V}}_1,\ldots,\widehat{\mathcal{V}}_{D+1}$, resp.). Each $\mathcal{V}_i$ (for $i \leq D$) sends a random $Z_i \leftarrow \{0,1\}^t$ in time $T - \|\widehat{\mathcal{V}}_i\widehat{\mathcal{P}}\|$ (where $\widehat{\mathcal{P}}$ is the claimed position of the prover), and $\mathcal{V}_{D+1}$ sends a random $X \leftarrow \{0,1\}^n$ (in time $T - \|\widehat{\mathcal{V}}_i\widehat{\mathcal{P}}\|$). All the messages arrive to $\mathcal{P}$ in time T. Then, $\mathcal{P}$ computes $X[H_1(Z)],\ldots,X[H_t(Z)]$, and sends the result back to $\mathcal{V}_{D+1}$, who checks if the result is correct (at the end of this section we discuss how this check can be done very efficiently). The security of $\mathit{\Pi}_{D,n}^{\mathcal{H},t}$ follows directly from Lemma 7, and the following fact.

Lemma 8. *Consider an almost adaptive 1-round SM protocol* $\text{PLR}_1,\ldots,\text{PLR}_k$ *with* PLR_k *being the referee, and every player having random oracle access to the functions in* $\mathcal{H}$*. Let* βn *denote the total communication complexity of this protocol (where* $n \in \mathbb{N}$ *and* $\beta < 1$ *is some constant) and let* q *be the number of times the parties query the random oracles. Assume* q *is polynomial in* t *and* $n > t$*. Let* Y *denote the output of* PLR_k*. Then we have*

$$\mathbb{P}\left(Y = \pi_{k,n}^{\mathcal{H},t}(Z_1,\ldots,Z_{k-1},X)\right) \leq \mathsf{negl}(t), \tag{9}$$

where the probability in (9) is taken over random $X \leftarrow \{0,1\}^m$*,* $(Z_1,\ldots,Z_{k-1}) \leftarrow (\{0,1\}^t)^{k-1}$*, and the random choice of the functions on* $\mathcal{H}$*.*

Proof. Suppose we have a almost adaptive 1-round SM protocol $\text{PLR}_1,\ldots,$ PLR_k (with PLR_k being the referee) such that the probability in (9) is non-negligible. Recall the guessing game from

Sect. 2.1. We now show how to use $\textsc{plr}_1, \dots, \textsc{plr}_k$ to construct a pair of functions $\mathsf{compress} : \{0,1\}^n \to \{0,1\}^{\beta n}$ and $\mathsf{guess} : \{1,\dots,n\}^t \times \{0,1\}^{\beta n} \to \{0,1\}^t$ such that the probability that $\mathsf{guess}(R, \mathsf{compress}(X)) = (X[R_1], \dots, X[R_t])$ is non-negligible in t, where $X \leftarrow \{0,1\}^n$ and $R = (R_1, \dots, R_t) \leftarrow \{1,\dots,n\}^t$. Since by Lemma 4 we know that this is impossible, we will obtain that the probability in (9) has to be negligible.

The functions $\mathsf{compress}$ and guess that we construct are randomized, i.e., they depend on some external fresh randomness. In particular, we will assume that the hash functions $\mathcal{H}$ that the players have access to (via the random oracle) were sampled in advance. Of course, such sampling cannot be done efficiently (since the set of all such functions is of exponential size), but this is ok, since our construction is anyway information-theoretic (note that Lemma 4 does not involve any complexity-theoretic assumptions). We will later argue why the assumption about the availability of external randomness can be done without loss of generality. First, however, let us present the definitions of the functions $\mathsf{compress}$ and guess.

The function $\mathsf{compress}$ is defined as follows. First it samples $(Z_1, \dots, Z_{k-1}) \leftarrow (\{0,1\}^t)^{k-1}$. Then, on input $(X, R_1, \dots, R_{k-1})$ it produces as output a tuple $(V_1, \dots, V_{k-1})$, where each V_i is equal to the output of player $\textsc{plr}_i$ on input $(R_1, \dots, R_{i-1}, R_{i+1}, \dots, R_{k-1}, X)$ (recall that in this model the referee $\textsc{plr}_k$ does not produce any output in the first phase). Note that simulating the $\textsc{plr}_i$'s may require replying to their random oracle queries. We reply to each such a query using the hash functions $\mathcal{H}$ that were sampled beforehand. Observe that $|(V_1, \dots, V_{k-1})| \leq \beta n$, and therefore $\mathsf{compress}$ can fit this output in the set $\{0,1\}^{\beta n}$.

On input $(R_1, \dots, R_t)$ and X the function guess does the following. It simulates the referee $\textsc{plr}_k$ on input $(Z_1, \dots, Z_{k-1})$ (which are the values that were already sampled by $\mathsf{compress}$). It answers all the random oracle queries using $\mathcal{H}$, with one important exception. Namely, every query of a form $(Z_1 || \cdots || Z_{k-1})$ to an oracle containing a hash function H_j (for $j = 1, \dots, t$) is answered with R_j.

Now, let $\mathcal{E}$ denote the event that it never happened that any of the $\textsc{plr}_1, \dots, \textsc{plr}_{k-1}$ queried any of the random oracles on $(Z_1 || \cdots || Z_{k-1})$. It is easy to see that we have the following:

$$\begin{aligned} &\mathbb{P}\left(Y = \pi^{\mathcal{H},t}_{k,n}(Z_1, \dots, Z_{k-1}, X) \mid \mathcal{E}\right) \\ &= \mathbb{P}\left(\mathsf{guess}(R, \mathsf{compress}(X)) = (X[R_1], \dots, X[R_t]) \mid \mathcal{E}\right). \end{aligned} \tag{10}$$

This is because if $\mathcal{E}$ occurred then the functions $\mathsf{compress}$ and guess perfectly "emulated" the execution of $\textsc{plr}_1, \dots, \textsc{plr}_k$. Observe that here we use the assumption that the R_j's are uniform, which implies that our answers to the "$(Z_1 || \cdots || Z_{k-1})$" queries are indistinguishable from the answers of the "real" random oracle. Of course, this would not be true if such a query was earlier asked by one of $\textsc{plr}_1, \dots, \textsc{plr}_{k-1}$, but this did not happen, since in (10) we condition on the event $\mathcal{E}$.

On the other hand, it is clear that $\mathbb{P}(\neg\mathcal{E}) \leq q/2^t$. This is because querying the oracle on "$(Z_1 || \cdots || Z_{k-1})$" requires the knowledge of all the Z_i's, and every $\textsc{plr}_i$ (for $i = 1, \dots, k-1$) does not know one of them. Hence the probability that any $\textsc{plr}_i$ guesses "$(Z_1 || \cdots || Z_{k-1})$" in one query is 2^{-t} (remember that each of them is uniformly random on $\{0,1\}^t$). Consequently, the probability that it guesses it in *at least one* of its q queries is at most $q/2^t$. Since we assumed that q is polynomial in t, thus we get that $\mathbb{P}(\neg\mathcal{E}) \leq \mathsf{negl}(t)$. Combining it with (10) we obtain

$$\mathbb{P}\left(Y = \pi^{\mathcal{H},t}_{k,n}(Z_1, \dots, Z_{k-1}, X)\right) \tag{11}$$

$$\leq \mathbb{P}\left(\mathsf{guess}(R, \mathsf{compress}(X)) = (X[R_1], \dots, X[R_t])\right) + \mathsf{negl}(t). \tag{12}$$

Thus, since we assumed that (11) is non-negligible, we obtain the the probability in (12) is non-negligible.

What remains is to describe how to "derandomize" the $\mathsf{compress}$ and guess functions that we constructed. This can be done via a very standard argument. Since the inequality (12) holds when the probability is computed *including* the internal randomness ρ of $\mathsf{compress}$ and guess thus there has to exist a concrete value ρ_0 such that (12) holds if we fix ρ to ρ_0. We can therefore derandomize these functions by simply "hardwiring" these randomness into them. This finishes the proof. □

Let us also discuss the nature of the $\pi_{k,n}^{\mathcal{H},t}$ function, focusing on the (simplest) case when $t = 1$, i.e., only on bit is produced as output. The reader familiar with the communication complexity literature may observe that this function is similar to so-called *shift function* [28], and more general notion of called the *general addressing function (GAF)* [31,4]. The shift function is defined very similarly to $\pi_{k,n}^{\mathcal{H},1}$, except that the Z_i's take values in the $\mathbb{Z}_n$ group, and H_1 is defined as $H(Z_1, \dots, Z_n) := Z_1 + \cdots Z_{k-1}$ (in case of GAF we can also have groups other than $\mathbb{Z}_n$). Somewhat surprisingly it appears very hard to prove the lower bounds for the SM complexity in this model. The only known non-trivial lower bound in the shift function is $\Omega(n^{1/k})$ [28,31]. Moreover, sublinear upper bounds on this complexity are known [31,30,1,2]. The hardness of this problem can in some sense serve as a justification for the use of the random oracles in our construction. Lemma 7 and 8 together imply the following.

Corollary 1. *For any $\beta < 1$ and for $n > t$ the protocol $\Pi_{D,n}^{\mathcal{H},t}$ is a $(\beta n, \mathsf{negl}(t))$-secure positioning protocol for positions within the the D-dimensional simplex whose vertices are the positions of the protocol's verifiers.*

Let us also now mention that in a practical implementation one can let the verifiers choose the Z_i's in advance. Therefore $\mathcal{V}_{D+1}$ can compute $H_i(Z_i)$'s and store only the $X[H_i(Z_i)]$'s. Thus, the storage requirements of this protocol are very low.

4.2 Protocols in the Plain Model

In this section we propose an alternative construction of positioning and key agreement protocols. The protocols presented in this section are online computable (see Sect. 2.4), and do not require the random oracle assumption. Let us first recall the definition of the generalized inner product function [5]. Let $\mathbb{F} = \mathsf{GF}(2^m)$ be a finite field (for simplicity we restrict ourselves to the Galois fields of order 2^m, but our results can be generalized to arbitrary finite fields). For some natural parameters ℓ and k (such that $k \geq 2$) define the *generalized inner product (GIP)* function as $\mathsf{GIP}_{\ell,k} : (\mathbb{F}^\ell)^k \to \mathbb{F}$ as $\mathsf{GIP}_{\ell,k}^{\mathbb{F}}((x_1^1, \dots, x_\ell^1), \dots, (x_1^k, \dots, x_\ell^k)) = \sum_{i=1}^{\ell} \prod_{j=1}^{k} x_i^j$.

The positioning and the position-based key agreement protocols (in $D \in \{2, 3\}$ dimensions), denoted $\Gamma_{\ell,D,t}^{\mathsf{pos}}$ and $\Gamma_{\ell,D,t}^{\mathsf{ka}}$ (resp.), are simply the one-round protocols parameterized by $\mathsf{GIP}_{\ell,D+1}^{\mathbb{F}}$ (see Sect. 2.3), i.e., the verifiers $\mathcal{V}_1, \dots, \mathcal{V}_{D+1}$ broadcast random strings $X_i \leftarrow \mathbb{F}^\ell$, and the prover computes $\mathsf{GIP}(X_1, \dots, X_{D+1})$, which he either keeps as the agreed key, or broadcasts back to the verifiers (depending on whether the protocol is for key agreement or for positioning). The verifiers compute $\mathsf{GIP}(X_1, \dots, X_{D+1})$ and keep it as the agreed key (in the first case), or simply check if it is identical to what they got from the prover (in the second case). We now have the following lemma that states that GIP is hard in the fully adaptive model. Note that this lemma implies hardness also in the 2-round SM model and the almost adaptive 1 round SM model (since these models are more restrictive), and hence, together with Lemmas 6 and 7, implies security of the $\Gamma_{\ell,D,t}^{\mathsf{pos}}$ and $\Gamma_{\ell,D,t}^{\mathsf{ka}}$ protocols. The communication complexity of the GIP function has been studied in multiple papers [5,6,32,23,22], but up to our knowledge, not in the strong randomized settings that we need in this work. Our proof is a rather straightforward adaptation of the techniques form this prior work.

Lemma 9. *Suppose $\mathbb{F} = \mathsf{GF}(2^m)$ (for any m such that $2^m \geq k^{1+\xi}$ for some $\xi > 0$). Then for every ℓ, k, the $\mathsf{GIP}_{\ell,k}^{\mathbb{F}}$ function is (s, δ)-strongly hard in the fully adaptive model, for some $s = \Omega(m\ell/2^k)$ and $\delta = \mathsf{negl}(\ell)$.*

Proof. Consider an arbitrary fully adaptive protocol ($\textsc{plr}_1, \ldots, \textsc{plr}_k$). Let s denote its communication complexity. Suppose that $\overrightarrow{X}^1, \ldots, \overrightarrow{X}^k$ are sampled uniformly and independently, each from $\mathbb{F}^\ell$. Let V denote the sequence of all the messages that were broadcast by the parties during the execution of the protocol on input $(\overrightarrow{X}^1, \ldots, \overrightarrow{X}^k)$. Let $Y := \mathsf{GIP}^{\mathbb{F}}_{\ell,k}(\overrightarrow{X}^1, \ldots, \overrightarrow{X}^k)$. We will now treat $Y \in \mathsf{GF}(2^m)$ as a bit-strings of length m. We start with the following.

Claim. For any $i \in 1, \ldots, m$ and $s = \Omega(m\ell/2^k)$ we have that

$$d(Y[i] \mid Y[1, \ldots, i-1], V)) \leq \mathsf{negl}(\ell). \tag{13}$$

Proof (Proof of the Claim). We use the results of [5] which introduced the so-called *multiparty communication complexity with help*. More precisely, in [5] the authors consider protocols where the players can obtain an extra "help" from an external entity in a form of a function H that gets as input all the inputs of all the players, the only restriction being that the output of H has to be one bit shorter than the output of the computed function. Hence, in our case H is any function of a type $H : (\mathbb{F}^\ell)^k \to \{0,1\}^{m-1}$. What they prove in their Lemma 3.3 can be translated to our notation as follows:

For any protocol whose communication complexity is at most

$$\log\left(\frac{1/2 - \epsilon}{\Gamma(f, \mathcal{C})}\right) \tag{14}$$

(we will comment on the "$\Gamma(f, \mathcal{C})$" term in a moment) and for any $H : (\mathbb{F}^\ell)^k \to \{0,1\}^{m-1}$ and any function α we have that

$$\mathbb{P}\left(\mathsf{GIP}^{\mathbb{F}}_{\ell,k}(\overrightarrow{X}^1, \ldots, \overrightarrow{X}^k) = \alpha(H(\overrightarrow{X}^1, \ldots, \overrightarrow{X}^k), V)\right) \leq 1 - \epsilon. \tag{15}$$

(provided $2^m \geq k^{1+\xi}$ for some $\xi > 0$).

Above $\Gamma(f, \mathcal{C})$ is a value called *the strong discrepancy of f in $\mathcal{C}$* (for this discussion it is irrelevant what $\mathcal{C}$ is). Moreover, as inspection of the proof of Corollary 4.12 [5] shows we have that

$$\log(1/\Gamma(f, \mathcal{C})) \geq \Omega(m\ell/2^k). \tag{16}$$

Now, set $\epsilon := 1/2 - \sqrt{\Gamma(f, \mathcal{C})}$. It is easy to see that (14) now becomes equal to

$$\log(1/\sqrt{\Gamma(f, \mathcal{C})}) \geq \Omega(m\ell/2^k).$$

This also implies that $\epsilon - 1/2$ is negligible in ℓ. Moreover, by Lemma 2, we have that

$$d(\mathsf{GIP}^{\mathbb{F}}_{\ell,k}(\overrightarrow{X}^1, \ldots, \overrightarrow{X}^k) \mid H(\overrightarrow{X}^1, \ldots, \overrightarrow{X}^k), V) \leq 2(1-\epsilon) - 1 \leq \mathsf{negl}(\ell), \tag{17}$$

Now set $H(\overrightarrow{X}^1, \ldots, \overrightarrow{X}^k) := (Y[1, \ldots, i-1], Y[i+1], \ldots, Y[m])$. Then, (17) becomes

$$\begin{aligned} \mathsf{negl}(\ell) &\geq d(Y \mid Y[1, \ldots, i-1], Y[i+1], \ldots, Y[m], V) \\ &\geq d(Y[i] \mid Y[1, \ldots, i-1], V), \end{aligned} \tag{18}$$

where (18) follows from Lemma 2. Hence (13) is proven.

To finish the proof of Lemma 9 we just apply the chain-rule for the statistical distance (Lemma 1), obtaining

$$d(Y \mid V) \leq m \cdot \mathsf{negl}(\ell) = \mathsf{negl}(\ell).$$

We therefore obtain that for any protocol with the communication complexity $\Omega(m\ell/2^k)$ we have $d(\mathsf{GIP}^{\mathbb{F}}_{\ell,k}(\overrightarrow{X}^1, \ldots, \overrightarrow{X}^k) \mid V) \leq \mathsf{negl}(\ell)$, and the lemma is proven. □

Now, combining Lemma 9 with Lemmas 6 and 7 we obtain the following.

Corollary 2. *For $D \in \{2,3\}$ and for $k, m,$ and ℓ as in Lemma 9, we have that $\Gamma^{\mathsf{pos}}_{\ell,D,t}$ is one-round $(\Omega(m\ell), \mathsf{negl}(\ell))$-secure positioning protocol in D dimensions for positions inside of a simplex determined by the verifiers, and $\Gamma^{\mathsf{ka}}_{\ell,D,t}$ is a one-round $(\Omega(m\ell), \mathsf{negl}(\ell))$-secure key agreement protocol in D dimensions for positions such that the messages sent by the verifiers never meet at any other position than the one claimed by the prover (see [15], Sect. 7.3.1).*

4.3 Practical considerations for the GIP-based protocol

Note that, unlike in the case of protocol $\Pi^{\mathcal{H},t}_{D,n}$ (see remark after Corollary 1), there is no simple trick to avoid the need for the verifiers to store large amounts of data (the X_i's), as long as we want the protocols to be information-theoretically secure. However, if we move to the "computational world" we can simply let the X_i's be generated pseudorandomly: for $i = 1, \ldots, D + 1$ sample a short random seed S_i, and let $X_i := \mathsf{prg}(S_i)$, where prg is a pseudorandom generator. In this case, the verifiers need to store only the S_i's. Also, instead of sending the X_i's (via a private channel) to each other, they can just send the S_i.

References

1. Andris Ambainis. Upper bounds on multiparty communication complexity of shifts. In Claude Puech and Rüdiger Reischuk, editors, *STACS 96, 13th Annual Symposium on Theoretical Aspects of Computer Science, Grenoble, France, February 22-24, 1996, Proceedings*, volume 1046 of *Lecture Notes in Computer Science*, pages 631–642. Springer, 1996.
2. Andris Ambainis and Satyanarayana V. Lokam. Imroved upper bounds on the simultaneous messages complexity of the generalized addressing function. In Gaston H. Gonnet, Daniel Panario, and Alfredo Viola, editors, *LATIN 2000: Theoretical Informatics, 4th Latin American Symposium, Punta del Este, Uruguay, April 10-14, 2000, Proceedings*, volume 1776 of *Lecture Notes in Computer Science*, pages 207–216. Springer, 2000.
3. Yonatan Aumann and Michael O. Rabin. Information theoretically secure communication in the limited storage space model. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 65–79, Santa Barbara, CA, USA, August 15–19, 1999. Springer, Heidelberg, Germany.
4. László Babai, Anna Gál, Peter G. Kimmel, and Satyanarayana V. Lokam. Communication complexity of simultaneous messages. *SIAM J. Comput.*, 33(1):137–166, January 2004.
5. László Babai, Thomas P. Hayes, and Peter G. Kimmel. The cost of the missing bit: Communication complexity with help. *Combinatorica*, 21(4):455–488, 2001.
6. László Babai, Noam Nisan, and Mario Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *J. Comput. Syst. Sci.*, 45(2):204–232, 1992.
7. Stefan Brands and David Chaum. Distance-bounding protocols (extended abstract). In Tor Helleseth, editor, *EUROCRYPT'93*, volume 765 of *LNCS*, pages 344–359, Lofthus, Norway, May 23–27, 1994. Springer, Heidelberg, Germany.
8. G. Brassard. Quantum information: The conundrum of secure positioning. *Nature*, 479:307–308, November 2011.
9. Harry Buhrman, Nishanth Chandran, Serge Fehr, Ran Gelles, Vipul Goyal, Rafail Ostrovsky, and Christian Schaffner. Position-based quantum cryptography: Impossibility and constructions. *SIAM Journal on Computing*, 43(1):150–178, 2014.
10. Christian Cachin and Ueli M. Maurer. Unconditional security against memory-bounded adversaries. In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 292–306, Santa Barbara, CA, USA, August 17–21, 1997. Springer, Heidelberg, Germany.
11. S. Capkun and J.-P. Hubaux. Secure positioning of wireless devices with application to sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 1917–1928 vol. 3, March 2005.
12. David Cash, Yan Zong Ding, Yevgeniy Dodis, Wenke Lee, Richard J. Lipton, and Shabsi Walfish. Intrusion-resilient key exchange in the bounded retrieval model. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 479–498, Amsterdam, The Netherlands, February 21–24, 2007. Springer, Heidelberg, Germany.
13. Kaushik Chakraborty and Anthony Leverrier. Practical position-based quantum cryptography. *Phys. Rev. A*, 92:052304, Nov 2015.
14. Ashok K. Chandra, Merrick L. Furst, and Richard J. Lipton. Multi-party protocols. In *Proc. 15th Annual ACM Symposium on the Theory of Computing*, pages 94–99, 1983.
15. Nishanth Chandran, Vipul Goyal, Ryan Moriarty, and Rafail Ostrovsky. Position-based cryptography. *SIAM Journal on Computing*, 43(4):1291–1341, 2014.
16. Stefan Dziembowski. Intrusion-resilience via the bounded-storage model. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 207–224, New York, NY, USA, March 4–7, 2006. Springer, Heidelberg, Germany.
17. Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. Non-malleable codes from two-source extractors. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 239–257, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany.
18. Stefan Dziembowski and Ueli M. Maurer. Tight security proofs for the bounded-storage model. In *34th ACM STOC*, pages 341–350, Montréal, Québec, Canada, May 19–21, 2002. ACM Press.

19. Stefan Dziembowski and Ueli M. Maurer. Optimal randomizer efficiency in the bounded-storage model. *Journal of Cryptology*, 17(1):5–26, January 2004.
20. Stefan Dziembowski and Krzysztof Pietrzak. Intrusion-resilient secret sharing. In *48th FOCS*, pages 227–237, Providence, USA, October 20–23, 2007. IEEE Computer Society Press.
21. Stefan Dziembowski and Maciej Zdanowicz. Position-based cryptography from noisy channels. In David Pointcheval and Damien Vergnaud, editors, *AFRICACRYPT 14*, volume 8469 of *LNCS*, pages 300–317, Marrakesh, Morocco, May 28–30, 2014. Springer, Heidelberg, Germany.
22. Jeff Ford and Anna Gál. Hadamard tensors and lower bounds on multiparty communication complexity. *Computational Complexity*, 22(3):595–622, 2013.
23. Fan Chung Graham. Quasi-random hypergraphs revisited. *Random Struct. Algorithms*, 40(1):39–48, 2012.
24. Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, Cambridge, 1997.
25. Chi-Jen Lu. Encryption against storage-bounded adversaries from on-line strong extractors. *Journal of Cryptology*, 17(1):27–42, January 2004.
26. Ueli M. Maurer. Conditionally-perfect secrecy and a provably-secure randomized cipher. *Journal of Cryptology*, 5(1):53–66, 1992.
27. Tal Moran, Ronen Shaltiel, and Amnon Ta-Shma. Non-interactive timestamping in the bounded-storage model. *Journal of Cryptology*, 22(2):189–226, April 2009.
28. Noam Nisan and Avi Wigderson. Rounds in communication complexity revisited. In *23rd ACM STOC*, pages 419–429, New Orleans, Louisiana, USA, May 6–8, 1991. ACM Press.
29. Noam Nisan and David Zuckerman. Randomness is linear in space. *J. Comput. Syst. Sci.*, 52(1):43–52, February 1996.
30. Pavel Pudlák. Unexpected upper bounds on the complexity of some communication games. In Serge Abiteboul and Eli Shamir, editors, *Automata, Languages and Programming, 21st International Colloquium, ICALP94, Jerusalem, Israel, July 11-14, 1994, Proceedings*, volume 820 of *Lecture Notes in Computer Science*, pages 1–10. Springer, 1994.
31. Pavel Pudlk, Vojtech Rdl, and Jir Sgall. Boolean circuits, tensor ranks, and communication complexity. *SIAM Journal on Computing*, 26(3):605–633, 1997.
32. Ran Raz. The bns-chung criterion for multi-party communication complexity. *Computational Complexity*, 9(2):113–122, 2000.
33. Naveen Sastry, Umesh Shankar, and David Wagner. Secure verification of location claims. In *Proceedings of the 2Nd ACM Workshop on Wireless Security*, WiSe '03, pages 1–10, New York, NY, USA, 2003. ACM.
34. Christian Schaffner. Position-based quantum cryptography. webpage `http://homepages.cwi.nl/~schaffne/positionbasedqcrypto.php`, accessed on Feb 17, 2016.
35. Dominique Unruh. Quantum position verification in the random oracle model. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 1–18, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Heidelberg, Germany.
36. Salil P. Vadhan. Constructing locally computable extractors and cryptosystems in the bounded-storage model. *Journal of Cryptology*, 17(1):43–77, January 2004.
37. Adnan Vora and Mikhail Nesterenko. Secure location verification using radio broadcast. In *Proceedings of the 8th International Conference on Principles of Distributed Systems*, OPODIS'04, pages 369–383, Berlin, Heidelberg, 2005. Springer-Verlag.

A Proof of Lemma 3

Let $U_{\mathcal{X}}$ and $U_{\mathcal{Y}}$ be uniform random variables distributed over the same sets as X and Y, respectively. We have

$$\begin{aligned} d(X, Y \mid Z, V) &= \Delta((X, Y, Z, V); (U_{\mathcal{X}}, Y_{\mathcal{Y}}, Z, V)) \\ &\geq \Delta((X, Z); (U_{\mathcal{X}}, Z)) \qquad (19) \\ &= d(X \mid Z) \end{aligned}$$

where (19) comes from the fact that form any random variables and any function φ we have $\Delta(A; B) \geq \Delta(\varphi(A); \varphi(B))$ (see, e.g., Lemma 4 of the extended version of [17]). □

B Proof of Lemma 4

Simple inspection of the argument in Sect. 4.3 of [15]. Observe that EG in [15] is defined as $\mathsf{EG}(X, R) := (X[Z_1], \ldots, X[Z_t])$. The argument in [15] uses parameters β and δ in, where β is defined as the “adversarial storage rate“ (and is the same parameter as in our notation), and the δ is such that the min-entropy rate of X is $\beta + \delta$. Since in our case X is uniform, thus we can simply set $\delta := (1 - \beta)$. Observe that $\delta > 0$. In [15] the authors use a security parameter κ and require that $t \geq (2/\delta)\kappa$. We can however also treat t as the security parameter, and then set $\kappa := t\delta/2$. In [15] it is shown that the probability p of guessing $\mathsf{EG}(X, R)$ correctly is negligible in κ. Therefore it is also negligible in t (as δ is a positive constant). $\square$