Sound Proof of Proximity of Knowledge

Serge Vaudenay

EPFL
CH-1015 Lausanne, Switzerland
http://lasec.epfl.ch

Abstract. Public-key distance bounding schemes are needed to defeat relay attacks in payment systems. So far, only five such schemes exist, but fail to fully protect against malicious provers. In this paper, we solve this problem. We provide a full formalism to define the proof of proximity of knowledge (PoPoK). Protocols should succeed if and only if a prover holding a secret is within the proximity of the verifier. Like proofs of knowledge, these protocols must satisfy completeness, soundness (protection for the honest verifier), and security (protection for the honest prover). We construct ProProx, the very first sound PoPoK.

1 Introduction

Relay attacks can be a serious threat against applications such as NFC-based payment: for small payments, there is typically no action required on the creditcard or smartphone (beyond approaching to the terminal) such as typing a PIN code. So, a man-in-the-middle adversary could just relay communications between the payment device of the victim and the terminal to make payments on the behalf of the holder. The limit of the speed of communication was proposed to solve this problem [4]. Brands and Chaum [11] introduced the notion of *distance-bounding protocol* to prove that a *prover* is close enough to a *verifier*. This relies on information being local and unable to travel faster than the speed of light. So, an RFID reader can identify when participants are close enough because the round-trip communication time in challenge/response rounds have been small enough.

The literature considers several threat models.

- *Relay attack*: an adversary relay messages between a far-away honest prover and a verifier, trying to make the verifier accept. This is extended by *Mafia fraud* [15] where the adversary can also modify messages. This is further extended by *Man-in-the-Middle attack* [6,8,9] where the attack follows a learning phase where the prover could be close-by. In *Impersonation fraud* [2], the prover is absent and the adversary tries to impersonate the prover to the verifier. These threat models have in common that the prover is honest.
- *Distance fraud* [11]: a far-away malicious prover tries to pass the protocol.
- *Terrorist fraud* [15]: a far-away malicious prover, with the help of an adversary, tries to make the verifier accept, but without giving the adversary

any advantage to later pass the protocol alone. This extends to *Collusion fraud* [6,8,9] where the goal of the adversary is to run a man-in-the-middle attack. Terrorist fraud is also related to the notion of *soundness* [25]: whenever the verifier accepts, there must be an extractor who can reconstruct the secret of the prover based on the view of all close-by participants, possibly after several iterations. An hybrid model between distance fraud and terrorist fraud is the one of *Distance hijacking* [14]: A far-away prover takes advantage of some honest, active provers to make the verifier accept.

One of the first models to capture these notions was proposed by Avoine *et al.* [1]. However, it was not formal enough. Then, two parallel models were developed: the BMV model [6,8,9] and the DFKO model [16]. There exist many symmetric distance-bounding protocols but so far only the SKI protocol [5,6,7,9] (based on the BMV model), the Fischlin-Onete (FO) protocol [18] (based on the DFKO model), and DB1, DB2, and their extensions [10,24] (combining both SKI and FO in the BMV model) provide an all-encompassing proven security.

Public-key distance bounding. In interactive proofs, the prover does not share a secret key with the verifier. The verifier only knows a public key. However, so far, only the following distance-bounding protocols are in the public key model: the Brands-Chaum protocol [11], the Bussard-Bagga protocol [12], the Hermans-Peeters-Onete (HPO) protocol [22][1], and PrivDB [26]. The Bussard-Bagga protocol was broken by Bay et al. [3] and none of the others protect against terrorist fraud. Additionally, the protocol VSSDB was presented at the BalkanCryptSec'14 conference by Gambs *et al.* It is based on the random oracle model, but the instanciability is questionable, as it requires a NIZK proof on statements of form $\{x : c = f(x, H(x))\}$ where H is a random oracle. As far as we know, this does not exist. So, the problem of making a fully secure public-key distance-bounding protocol is still open.

In Table 1 we update the list from [26] with all known public-key distance bounding protocols and the proven status of their security with respect to Man-in-the-Middle (MiM), Distance Fraud (DF), Distance Hijacking (DH), Collusion Fraud (CF), Privacy, and Strong privacy.

Contribution. In clear, our contributions in this paper are as follows.

- We adapt the framework of [10] in the BMV model to provide a full formalization of public-key distance-bounding. We specify our new primitive: the *proof of proximity of knowledge (PoPoK)*.
- We change the definition of soundness from [10] and [25] to make it closer to the one of interactive proofs. So, our model is pretty natural and nicely connects recent work on distance bounding (such as the BMV model [6,8,9]) and interactive proofs.
- We construct ProProx, the very first sound PoPoK. It is based on the quadratic residuosity problem, using the Goldwasser-Micali encryption [20,21]

[1] A variant of the HPO protocol offers anonymous authentication [19].

Table 1. Existing Public-Key Distance Bounding Protocols

protocol	MiM	DF	DH	CF	Privacy	Strong privacy
Brands-Chaum [11]	secure	secure	insecure	insecure	insecure	insecure
DBPK-Log [12]		insecure		insecure	insecure	insecure
HPO [22]	secure	secure		insecure	secure	insecure
GOR [19]	secure	secure	insecure	insecure	insecure	insecure
privDB [26]	secure	secure	secure	insecure	secure	secure
ProProx (this paper)	secure	secure	secure	secure	insecure	insecure
eProProx [27]	secure	secure	secure	secure	secure	secure

as a homomorphic perfectly binding commitment $\mathsf{Com}(b;\rho)$ and the Fiat-Shamir protocol [17]. We also use a function H which is assumed to be such that $x \mapsto (\mathsf{Com}(b_1; H(x,1)), \ldots, \mathsf{Com}(b_n; H(x,n)))$ is a one-way function, where $x = (b_1, \ldots, b_n)$. (An easy instance is when H is a random oracle.)

– We provide a technique to prove security and soundness. Essentially, we construct a straightline extractor based on the "Fundamental Lemma" and prove that the protocol is zero-knowledge.

2 Model and Definitions

We refine the security definitions and other tools from the BMV model [6,8,9,25]. Constructions depend on some security parameter λ which is omitted for more readability. A *constant* does not depend on λ, while parameters defining cryptographic constructions do. Algorithms run in probabilistic polynomial-time (PPT) in terms of λ. A real function $f(\lambda)$ is negligible if for any d, we have $f(\lambda) = \mathcal{O}(\lambda^{-d})$, as $\lambda \to +\infty$. We denote $f(\lambda) = \mathsf{negl}(\lambda)$. We also define

$$\mathsf{Tail}(n, \tau, \rho) = \sum_{i=\tau}^{n} \binom{n}{i} \rho^i (1-\rho)^{n-i}$$

2.1 Computational, Communication, and Adversarial Models

In our settings, participants are interactive Turing machines running PPT algorithms. We follow the BMV model [6,8,9]: we assume that participants have a *location* which is an element of a metric space $\mathcal{S}$, with a distance function d. If a participant π_1 at a location loc_1 executes a special command $\mathsf{send}(\pi_2, m)$ at time t to send a message m to a participant π_2 at location loc_2, the message m is received by π_2 at time $t + d(\mathsf{loc}_1, \mathsf{loc}_2)$. Furthermore, any malicious participant π_3 at some location loc_3 could see this message m at time $t + d(\mathsf{loc}_1, \mathsf{loc}_3)$. We assume no authentication: π_2 does not know if the message really comes from π_1. There is however an exception preventing m from being delivered to π_2: if π_2 is honest and some (malicious) participant π_3 at some location loc_3 has sent a special signal $\mathsf{corrupt}(\pi_1, \pi_2)$ at time t', m is not delivered to π_2 if

$t+d(\mathsf{loc}_1, \mathsf{loc}_2) \geq t'+d(\mathsf{loc}_3, \mathsf{loc}_2)$. This condition is a consequence of the information traveling with a speed limit: whenever a malicious participant π_3 corrupts a $\pi_1 \to \pi_2$ channel, π_2 will only receive the messages from π_1 until his corruption signal emitted from π_3 reaches π_2.

Note that once the $\pi_1 \to \pi_2$ channel is corrupted, π_3 can still see the message m sent by π_1 and decide to send any m' to π_2, either depending on m if he waits to receive m, or not. The crux is that either m' is independent of m, or it is delivered at a later time, when $d(\mathsf{loc}_1, \mathsf{loc}_2) < d(\mathsf{loc}_1, \mathsf{loc}_3) + d(\mathsf{loc}_3, \mathsf{loc}_2)$.

The communication model is only used to prove the "Fundamental Lemma". We take here a version of it inspired from [24].

Lemma 1 (Fundamental Lemma). *Assume a multiparty protocol execution with a distinguished participant $\mathcal{V}$, the set* Far *of all participants within a distance to $\mathcal{V}$ larger than B, and the set* Close *of all other participants. At some time t in the execution, $\mathcal{V}$ broadcasts a random challenge c based on some fresh coins and waits for a response r for up to $2B$ time. (If no answer is received, we set $r = \perp$.) We let exp_c be the experiment in which the challenge is equal to c. For each instance U, we denote by View_U his initial view. For $U \neq \mathcal{V}$, this is common to all exp_c. We further denote by $\mathsf{Incoming}_U^c(E)$ the list of all incoming messages seen by U until time $t + 2B - d(\mathcal{V}, U)$ in exp_c and coming from an instance in a set E. Finally, $\mathsf{Outgoing}_\mathcal{V}$ denotes all messages sent by $\mathcal{V}$ before time t. This is common to all exp_c. There exists an algorithm* Algo *such that for all c and c_0, we have in exp_c that*

$$r = \mathsf{Algo}\left(c, (\mathsf{View}_U)_{U \in \mathsf{Close}}, (\mathsf{Incoming}_U^{c_0}(\mathsf{Far}))_{U \in \mathsf{Close} \cup \{\mathcal{V}\}}, \mathsf{Outgoing}_\mathcal{V}\right)$$

If Close *is empty, we can further write* $r = \mathsf{Algo}\left(\mathsf{Incoming}_\mathcal{V}^{c_0}(\mathsf{Far})\right)$.

To make the lemma short: r cannot depend on any message which was sent from a far away U after receiving c. So, if we simulate exp_{c_0} in a straightline way, we can compute for each c what would have been r if c was sent instead of c_0.

We provide below a detailed proof of this lemma in the BMV model.

Proof. We first show that there exists an algorithm such that for all c,

$$r = \mathsf{Algo}\left(c, (\mathsf{View}_U)_{U \in \mathsf{Close}}, (\mathsf{Incoming}_U^{c}(\mathsf{Far}))_{U \in \mathsf{Close} \cup \{\mathcal{V}\}}, \mathsf{Outgoing}_\mathcal{V}\right)$$

Indeed, we show below that for each $U \in \mathsf{Close}$ we can compute the view of U at time $t+2B-d(\mathcal{V}, U)$. Then, we can see if U sends a message r to $\mathcal{V}$. We can also see in $\mathsf{Incoming}_\mathcal{V}^c(\mathsf{Far})$ if there is a message r coming from far away. We can then compute the first of these messages which arrives to $\mathcal{V}$. We note that if r comes from some $U \in \mathsf{Close}$, then it must have been sent no later than $t+2B-d(\mathcal{V}, U)$. So, it must be among the computed messages.

Then, we show that for all $U \in \mathsf{Close} \cup \{\mathcal{V}\}$, $\mathsf{Incoming}_U^c(\mathsf{Far})$ is independent from c, so we can replace c by c_0. Indeed, every message w in $\mathsf{Incoming}_U^c(\mathsf{Far})$ is seen by some $U \in \mathsf{Close} \cup \{\mathcal{V}\}$ at time $t' \leq t+2B-d(\mathcal{V}, U)$ and comes from some $U' \in \mathsf{Far}$. So, it must have been sent at time $t' - d(U, U') \leq t + 2B - d(\mathcal{V}, U) -$

$d(U, U') \leq t + 2B - d(\mathcal{V}, U') \leq t + B$. We define exp' like in exp_c, except that each participant U'' is stopped at time $t + B - d(U', U'')$. In exp', $\mathcal{V}$ is stopped before time t, so c is not used. We show by induction that for each U'', the view of U'' at the stopping time of U'' is the same in exp_c and exp'. We deduce that w is independent from c_0.

What remains to be shown is that for each $U \in \mathsf{Close}$ we can compute the view of U at time $t + 2B - d(\mathcal{V}, U)$. This is shown by induction on the time. Indeed, this view is composed of the initial view View_U of U and of the incoming messages. These messages either come from $\mathcal{V}$, so are either c or something in $\mathsf{Outgoing}_{\mathcal{V}}$, or come from $U' \in \mathsf{Close}$, so can have been computed in the view of U', by induction, or come from $U' \in \mathsf{Far}$, so is in $\mathsf{Incoming}_U$. □

Participants can move, but not faster than communication. For simplicity, we assume that far-away participants (as defined in Def. 3) remain far away during the entire execution. Honest participants move as instructed by the adversary.

We sometimes consider that when an honest participant receives a message from another honest participant, it may be subject to noise. As for malicious participants, we could assume that they use a better equipment which eliminates noise. Also: whenever the honest-to-honest communication is not time-sensitive, we may also assume that they use error correction means so that the communication is noiseless.

2.2 PoPoK: Proofs of Proximity of Knowledge

Definition 2 (Proof of proximity of knowledge). *A proof of proximity of knowledge (PoPoK) is a tuple* $(\mathcal{K}, \mathsf{Kgen}, P, V, B)$*, consisting of: a key space* $\mathcal{K}$ *depending on a security parameter* λ*, with elements of polynomially-bounded size in terms of* λ*; a PPT algorithm* Kgen*; a two-party PPT protocol* $(P(\mathsf{sk}), V(\mathsf{pk}))$*, where* $P(\mathsf{sk})$ *is the* proving algorithm *and* $V(\mathsf{pk})$ *is the* verifying algorithm*; a distance bound* B*. The algorithm* Kgen *maps the secret* $\mathsf{sk} \in \mathcal{K}$ *to a public key* pk*.* pk *is given as input to all participants. At the end of the protocol,* $V(\mathsf{pk})$ *sends a final message* Out_V*. He accepts (*$\mathsf{Out}_V = 1$*) or rejects (*$\mathsf{Out}_V = 0$*).*

The protocol must be such that when running $P(\mathsf{sk})$ *and* $V(\mathsf{pk})$ *at locations within a distance up to* B*, in a noiseless environment, the verifier always accepts. This property is called* completeness*.*

If the protocol specifies a list of time-critical challenge/response exchanges, we say that it is complete with noise probability p_{noise} *if, in an environment in which all challenge/response rounds are independently corrupted with probability* p_{noise} *and other exchanges are not subject to noise, the probability that the verifier accepts is* $1 - \mathsf{negl}(\lambda)$*.*

In practice, if we want to have $B = 10$m, assuming that an adversary can do computation in negligible time, the timer for receiving a response r to a challenge c in Lemma 1 should be limited to 67ns. So, an honest prover at a zero distance must respond within less than 67ns. This clearly excludes any cryptographic computation. To be realistic, a PoPoK can only consider boolean (or very small) challenges and responses when it comes to use Lemma 1.

We adopt the multiparty setting from [10] and only adapt it to accommodate public-key distance bounding. We consider a setting with participants which are called either *provers*, *verifiers*, or *other actors*. In public-key settings, we assume only one verifier $\mathcal{V}$ (other verifiers can be taken as *other actors*). Similarly, we often assume that provers correspond to the same identity so share the same secret sk (provers with other secrets are considered as *other actors*). Other actors are malicious by default. The difference between malicious provers and malicious actors is in the input: they receive sk or only pk.

We assume that participants run their algorithm only once. Multiple executions are modeled by multiple instances which can be at different location or time. We only assume that instances of honest provers never run concurrently. A malicious prover may however clone himself at different locations and run many algorithms concurrently.

Definition 3 (Experiment). *Given a PoPoK* $(\mathcal{K}, \mathsf{Kgen}, P, V, B)$*, we define an* experiment exp *by several* participants *who are a* verifier $\mathcal{V}$*,* provers*, and other* actors*, and each* instance *of the participants. Instances who are within a distance of at most* B *to* $\mathcal{V}$ *are said* close-by*. Instances who are within a distance larger than* B *to* $\mathcal{V}$ *are called* far-away*. We say that the prover is* always far-away *if all its instances are far away. We adopt a* static *adversarial model: either the prover is honest, in which case all its instances run the* $P(\mathsf{sk})$ *algorithm, or the prover is malicious, in which case its instances can run any PPT algorithm.*

If the prover is honest, its instances are assumed to be non-concurrent*: at each time, it must be defined which is the current instance. An instance of a honest participant can only be active if it is the current one and if it has received a special* $\mathsf{Activate}$ *message from a malicious participant. The first current instance must be defined in the experiment. Instances store an address of the next current instance. This address can be updated by a special* $\mathsf{Destination}$ *message from a malicious participant. It can also receive a special* Halt *message making the algorithm terminate, and a special* Move *message. After receiving this message and as soon as the algorithm terminated, the instance sends a special* Moving *message to the instance specified in his destination address. Only current instances can send this message to an instance of the same participant.*[2]

At the beginning of the experiment, for malicious provers, $(\mathsf{sk}, \mathsf{pk})$ *is set arbitrarily. If the provers are honest,* $\mathsf{sk} \in \mathcal{K}$ *is randomly selected and* $\mathsf{pk} = \mathsf{Kgen}(\mathsf{sk})$ *is computed. Then,* sk *is given as input to all prover instances, while* pk *is given as input to all participants.* $\mathcal{V}$ *runs* $V(\mathsf{pk})$*. All participants are then activated and run concurrently. (If the prover is honest, only one is activated.) The experiment terminates when* $\mathcal{V}$ *produces its final output* Out_V*.*

We formalize security following [10].

Definition 4 (Honest Prover Security of PoPoK). *We say that a PoPoK* $(\mathcal{K}, \mathsf{Kgen}, P, V, B)$ *is* HP-secure *if* $\Pr[\mathsf{Out}_V = 1] = \mathsf{negl}(\lambda)$ *for any experiment with a single prover, where the prover is honest and always far-away from* $\mathcal{V}$*.*

[2] All these special messages are defined in order to avoid participants moving faster than messages and to allow arbitrary movements influenced by the adversary.

This definition clearly captures relay attacks, Mafia fraud [15], man-in-the-middle attacks in general, and even models (like in [6,8,9]) which distinguish a learning phase (with provers which could be close-by) and an attack phase (with far-away provers).

We now formalize the protection for the honest verifier. Intuitively, we want that if the proof is accepted, it must be because the information about the secret sk is in the close-by neighborhood.

Definition 5 (Soundness of PoPoK). *Given a function $p(\lambda)$, we say that a PoPoK $(\mathcal{K}, \mathsf{Kgen}, P, V, B)$ is* $p(\lambda)$-sound *if for any experiment* exp *in which* $\Pr[\mathsf{Out}_V = 1] > p(\lambda)$*, there exists an algorithm $\mathcal{E}$ called* extractor*, with the following property.* exp *defines an oracle which simulates an execution of* exp *and returns the views of all participants which are close-by (excluding $\mathcal{V}$) and the transcript of the protocol seen by $\mathcal{V}$. $\mathcal{E}$ can invoke the oracle many times. Then, $\mathcal{E}$ finally outputs* sk' *such that* $\mathsf{Kgen}(\mathsf{sk}') = \mathsf{pk}$*, using an expected time complexity of* $\frac{\mathsf{Poly}(\lambda)}{\Pr[\mathsf{Out}_V=1]-p(\lambda)}$.

This is trivial for experiments with a close-by prover as sk is in the view of the prover. For experiments with no close-by participant at all, the transcript as seen by $\mathcal{V}$ would leak. Otherwise, close-by actors would extract the prover's credential. So, a far away malicious prover is bound to leak.

Compared to the soundness of interactive proofs, our notion uses a straight-line extractor: we extract the secret from close-by participants without rewinding them and after several independent executions. This makes the treatment of multiparty settings much easier. As we will see, our extractor essentially uses Lemma 1. Interestingly, the extractor is also used to prove HP-security: if the protocol is zero-knowledge, the oracle extractor can be transformed into a stand-alone extractor which contradicts the one-wayness of Kgen.

Clearly, our definition nicely connects the infamous terrorist-fraud resistance to the soundness of interactive proofs. To compare with the literature, we could see that terrorist frauds in our model make the secret leak instead of only making man-in-the-middle attack feasible as in the notion of collusion fraud proposed in [6,8,9], and on which the SKI protocol is based, or only making impersonation attack feasible as in [10]. Our soundness is thus stronger.

Our notion and the one of [10] are close to soundness as defined in [25], except that we no longer require $1/\Pr[\mathsf{Out}_V = 1]$ to be polynomial. Also, compared to [10], we no longer need the condition on the success of the experiment to extract and we call an oracle $\mathcal{O}$ many times instead of using m views.

Just like other notions of TF-resistance, soundness is incomparable with SimTF-security [16] or GameTF-security [18] in the DFKO model.

Definition 6 (Distance-fraud security). *A PoPoK $(\mathcal{K}, \mathsf{Kgen}, P, V, B)$ resists to* distance fraud *if for any experiment* exp *where all participants are far away from $\mathcal{V}$, we have that* $\Pr[\mathsf{Out}_V = 1] = \mathsf{negl}(\lambda)$.

We adapt the definition of distance-hijacking security from [26].

Definition 7 (Resistance to Distance Hijacking [26]). *We say that the PoPoK* $(\mathcal{K}, \mathsf{Kgen}, P, V, B)$ *is* DH-secure *if for all PPT algorithms* K *and* $\mathcal{A}$*, the following game makes* $\mathcal{V}$ *accept with negligible probability:*

1: *pick* $\mathsf{sk}' \in \mathcal{K}$, $\mathsf{pk}' = \mathsf{Kgen}(\mathsf{sk})$, $K(\mathsf{pk}') \to (\mathsf{sk}, \mathsf{pk})$*; if* $\mathsf{pk} = \mathsf{pk}'$*, the game aborts*
2: *let* $\mathcal{A}$ *run* $\mathcal{A}(\mathsf{sk}, \mathsf{pk}, \mathsf{pk}')$*, let* $\mathcal{V}$ *runs* $V(\mathsf{pk})$*, let* P'*,* $P'_1, P'_2, \ldots$ *run* $P(\mathsf{sk}')$
3: *let* $\mathcal{A}$ *interact with* $P', P'_1, P'_2 \ldots$ *and* $\mathcal{V}$ *concurrently until the initialization phase ends for* $\mathcal{V}$
4: *let* P' *and* $\mathcal{V}$ *continue interacting with each other until the challenge phase ends for* $\mathcal{V}$*;* $\mathcal{A}$ *receives the exchanged messages but remains passive*
5: *let* $\mathcal{A}$ *continue interacting with* $P', P'_1, P'_2 \ldots$ *and* $\mathcal{V}$ *concurrently during the verification phase*

3 ProProx: a PoPoK Scheme

3.1 Building Blocks

Perfectly binding bit commitment. Depending on the security parameter λ, we use a (multiplicative) group structure with two Abelian groups L and G and an element θ such that G is generated by L and θ, $\theta \notin L$, and L is the set of all squares of G. We further assume that it is easy to do group operations and comparisons in G and to sample elements in G uniformly.[3] Finally, we assume it is computationally hard to distinguish elements from L and from G.

We define $\mathsf{Com}(b; \rho) = \theta^b \rho^2$ for a bit b and a random $\rho \in G$, like in the Goldwasser-Micali cryptosystem [20,21]. So, Com is computationally hiding as defined by Def. 8. We will not require any secret key to extract b, although there *exists* a function Com^{-1} such that $\mathsf{Com}^{-1}(\mathsf{Com}(b; \rho)) = b$ for all $b \in \{0, 1\}$ and $\rho \in G$. We will rather use the homomorphic properties of the commitment and prove the correct commitment in a zero-knowledge way.

Definition 8 (Bit commitment). *A bit commitment consists of a PPT algorithm* Com *taking as input* λ*, a bit* $b \in \mathbf{Z}_2$*, and some random* $\rho \in G$*. It computes* $\mathsf{Com}(b; \rho) \in G$*. We define the following properties: 1. homomorphic: for all* $b, b' \in \mathbf{Z}_2$ *and* $\rho, \rho' \in G$*,* $\mathsf{Com}(b; \rho)\mathsf{Com}(b'; \rho') = \mathsf{Com}(b + b'; \rho\rho')$*; 2. perfectly binding: for all* $b, b' \in \mathbf{Z}_2$ *and* $\rho, \rho' \in G$*,* $\mathsf{Com}(b; \rho) = \mathsf{Com}(b'; \rho')$ *implies* $b = b'$*; 3. computationally hiding: for* ρ *random, the distributions of* $\mathsf{Com}(0; \rho)$ *and* $\mathsf{Com}(1; \rho)$ *are computationally indistinguishable.*

For instance, we can take a Blum integer N, i.e., $N = PQ$ for two distinct primes P and Q which are congruent to 3 modulo 4. We set L to the set of quadratic residues modulo N and $\theta = -1$: a residue modulo N such that $\left(\frac{\theta}{P}\right) = \left(\frac{\theta}{Q}\right) = -1$. The algorithm Com is given N and θ. We sample $r \in G$ by $r = \theta^b \rho^2 \bmod N$, for $b \in \mathbf{Z}_2$ and $\rho \in \mathbf{Z}_N^*$. Distinguishing G from L is the (supposedly hard) quadratic residuosity problem. In this case, N is assumed to come from a Common Reference String (CRS).

[3] So, we can sample an element of L uniformly by taking r^2 with r uniform in G.

A zero-knowledge proof for z being a square. We use the Fiat-Shamir protocol [17]. Namely, we show that z is a commitment to zero with a witness ζ (i.e., $z = \zeta^2$) with the protocol from Fig. 1, based on a perfectly hiding trapdoor commitment. Concretely, we use Def. 9 and Def. 10 with the $\mathcal{NP}$ language L of all squares. If $z = \zeta^2$, we say that z is a member of L with witness ζ.

Verifier public: z	$z = \zeta^2$	**Prover** secret: ζ
	$\xleftarrow{\mathsf{pk}}$	generate sk, $\mathsf{pk} = \mathsf{Gen}(\mathsf{sk})$
pick $e \in \mathbf{Z}_2$, pick r	$\xrightarrow{\mathsf{Commit}_{\mathsf{pk}}(e;r)}$	
	$\xleftarrow{h}$	pick $g \in G$, $h = g^2$
	$\xrightarrow{e,r}$	open commitment
check $z^e = \ell^2 h^{-1}$, $\mathsf{pk} = \mathsf{Gen}(\mathsf{sk})$	$\xleftarrow{\ell,\mathsf{sk}}$	$\ell = g\zeta^e$
	$\xrightarrow{\mathsf{Out}_V}$	

Fig. 1. $\mathsf{ZKP}(z : \zeta)$: a Sound and Zero-Knowledge Proof for z Being a Square.

Definition 9 (Sound proof of membership). *An interactive proof for a language L is a pair $(P(\zeta), V(z))$ of PPT algorithms such that 1.* completeness*: for any $z \in L$ with witness ζ, $\Pr[\mathsf{Out}_V = 1 : P(\zeta) \leftrightarrow V(z)] = 1$; 2. κ-*soundness*: for any $z \notin L$ and any algorithm P^* then $\Pr[\mathsf{Out}_V = 1 : P^* \leftrightarrow V(z)] \leq \kappa$.*

Definition 10 (Zero-knowledge protocol). *A protocol $(P(\zeta), V(z))$ for a language L is computationally zero-knowledge for $P(\zeta)$ if for any PPT interactive machine $V^*(z, \mathsf{aux})$ there exists a PPT algorithm $S(z, \mathsf{aux})$ and a negligible ε such that for any PPT distinguisher, any $(z : \zeta) \in L$, and any aux, the advantage for distinguishing the final view of $V^*(z, \mathsf{aux})$ in $P(\zeta) \leftrightarrow V^*(z, \mathsf{aux})$ and the output of $S(z, \mathsf{aux})$ is bounded by ε.*

The protocol of Fig. 1 is $\frac{1}{2}$-sound and zero-knowledge. It must be run k times in parallel to achieve a soundness level $\kappa = 2^{-k}$. We denote it by $\mathsf{ZKP}_\kappa(z : \zeta)$.

By using parallel composition, we extend the protocol to prove that $z_1, \ldots, z_k$ are some commitments to zero with witness $\zeta_1, \ldots, \zeta_k$ respectively, and denote it by $\mathsf{ZKP}_\kappa(z_1, \ldots, z_k : \zeta_1, \ldots, \zeta_k)$. I.e., it succeeds with probability up to κ if there exists i such that $z_i \notin L$.

(Perfectly binding) deterministic commitment. Given a hash function H making coins for Com, we define a deterministic commitment by

$$\mathsf{Com}_H(\mathsf{sk}) = (\mathsf{Com}(\mathsf{sk}_1; H(\mathsf{sk}, 1)), \ldots, \mathsf{Com}(\mathsf{sk}_s; H(\mathsf{sk}, s)))$$

for $\mathsf{sk} \in \mathbf{Z}_2^s$. We assume that Com_H is a one-way function (as defined by Def. 11). We assume the existence of Com and H such that Com_H is one-way as independent primitives. This is the case in particular when H is a random oracle, but H

is not necessarily assumed to be a random oracle. Constructions without using a random oracle are left to future work.

Definition 11 (One-way function). *We consider a function* Com *taking as input* λ *and a message* $\mathsf{sk} \in \mathbf{Z}_2^s$ *which is computable in deterministic polynomial time. The function is one-way if for any algorithm receiving* $\mathsf{Com}(\mathsf{sk})$*, for* $\mathsf{sk} \in \mathbf{Z}_2^s$ *random, the probability that it outputs* sk *is negligible.*

3.2 The ProProx Protocol

We define the ProProx protocol, as depicted on Fig. 2 (there, double arrows indicate messages which can be subject to noise). We consider s (the size of the secret), n (the number of rounds per iteration), τ (the minimal number of correct rounds per iteration for acceptance) as functions in terms of the security parameter λ. We assume s and n are asymptotically linear. We use a matrix $b \in \mathbf{Z}_2^{sn}$. The use of b will only appear in Theorem 18 to treat distance fraud. There, we will consider two variants.

Variant I: b is constant and the columns must have a Hamming weight of $\left\lfloor \frac{n}{2} \right\rfloor$ to make sure that $b_{i,j} + x_j \neq 0$ in half of the rounds. This requires $n \geq 2$.

Variant II: b is randomly selected by V and sent to P during the initialization phase. This requires $n \geq 1$.

Verifier public: pk	$\mathsf{pk} = \mathsf{Com}_H(\mathsf{sk})$	**Prover** secret: sk
	initialization phase for $i = 1$ to n and $j = 1$ to s	
		pick $a_{i,j} \in \mathbf{Z}_2$, $\rho_{i,j}$
	$\xleftarrow{A_{i,j}}$	$A_{i,j} = \mathsf{Com}(a_{i,j}; \rho_{i,j})$
	challenge phase for $i = 1$ to n and $j = 1$ to s	
pick $c_{i,j} \in \mathbf{Z}_2$		
start $\mathsf{timer}_{i,j}$	$\xrightarrow{c_{i,j}}$	receive $c'_{i,j}$
receive $r_{i,j}$, stop $\mathsf{timer}_{i,j}$	$\xleftarrow{r'_{i,j}}$	$r'_{i,j} = a_{i,j} + c'_{i,j} b_{i,j} + c'_{i,j} \mathsf{sk}_j$
	verification phase	
check $\#I_j = \tau$, $\mathsf{timer}_{i,j} \leq 2B$	$\xleftrightarrow{\text{agree on } I_1,\ldots,I_s}$	
for $i \in I_j$, $j = 1, \ldots, s$		
$z_{i,j} = A_{i,j} \left(\theta^{b_{i,j}} \mathsf{pk}_j\right)^{c_{i,j}} \theta^{-r_{i,j}}$	$\xleftrightarrow{ZKP_\kappa(z_{i,j}:\zeta_{i,j}:i \in I_j, j=1,\ldots,s)}$	$\zeta_{i,j} = \rho_{i,j} H(\mathsf{sk}, j)^{c'_{i,j}}$
	$\xrightarrow{\mathsf{Out}_V}$	

Fig. 2. ProProx: a Sound and Secure PoPoK.

The prover holds a secret $\mathsf{sk} \in \mathbf{Z}_2^s$ and the public key is $\mathsf{pk} = \mathsf{Com}_H(\mathsf{sk})$. We iterate s times and in parallel a protocol which we call an iteration and which

corresponds to an index j. First, the prover selects n bits $a_{1,j}, \ldots, a_{n,j} \in \mathbf{Z}_2$ and commits to them using some fresh coins $\rho_{1,j}, \ldots, \rho_{n,j}$, respectively. So, $A_{i,j} = \mathsf{Com}(a_{i,j}; \rho_{i,j})$, $i = 1, \ldots, n$. The $A_{i,j}$'s are sent to the verifier.

In the challenge phase, we have n time-critical rounds (in each iteration). These rounds may be subject to noise. The verifier picks a challenge $c_{i,j} \in \mathbf{Z}_2$ at random and sends it to the prover. The prover receives $c'_{i,j}$ (which may be different, due to noise). He computes his response $r'_{i,j} = a_{i,j} + c'_{i,j} b_{i,j} + c'_{i,j} \mathsf{sk}_j$ and sends it back to the verifier at once. The verifier receives $r_{i,j}$. The verifier measures the elapsed time $\mathsf{timer}_{i,j}$ taken to receive $r_{i,j}$ after $c_{i,j}$ was sent. Below, p_{noise} is the probability that some noise corrupts a challenge/response round. We assume that the noise corrupts each round independently.

Thus, the $c'_{i,j} \mapsto r'_{i,j}$ function maps one bit to one bit.

In the verification phase, the prover and the verifier determine a set I_j of τ round indices which they believe are correct. The way this agreement is done is not important (as long as the prover does not leak). Then, the verifier checks whether I_j has cardinality τ and the corresponding timers are small enough. If this fails, the verifier rejects. As a concrete instance for I_j agreement, we suggest that the prover sends (through the lazy noiseless channel) the $c'_{i,j}$ and $r'_{i,j}$ to the verifier. The verifier then takes the first τ rounds for which $c_{i,j} = c'_{i,j}$, $r_{i,j} = r'_{i,j}$, and $\mathsf{timer}_{i,j} \leq 2B$ to define I_j and sends I_j to the prover. If there are not enough correct rounds, the protocol aborts.

Next, the prover and the verifier run the interactive proof ZKP_κ to show that the responses $r_{i,j}$'s are consistent with the $A_{i,j}$'s and pk_j's. Namely, for all j and $i \in I_j$, they compute

$$z_{i,j} = A_{i,j} \left(\theta^{b_{i,j}} \mathsf{pk}_j\right)^{c_{i,j}} \theta^{-r_{i,j}} \quad , \quad \zeta_{i,j} = \rho_{i,j} H(\mathsf{sk}, j)^{c'_{i,j}}$$

Since $A_{i,j} = \theta^{a_{i,j}} \rho_{i,j}^2$ and $\mathsf{pk}_j = \theta^{\mathsf{sk}_j} H(\mathsf{sk}, j)^2$, it is easy to verify that $r_{i,j} = a_{i,j} + c_{i,j} b_{i,j} + c_{i,j} \mathsf{sk}_j$ is equivalent to the existence of $\zeta_{i,j}$ such that $z_{i,j} = \zeta_{i,j}^2$. That is, $z_{i,j} \in L$. If this fails, the protocol aborts. When the protocol aborts, the verifier sends $\mathsf{Out}_V = 0$. Otherwise, he sends $\mathsf{Out}_V = 1$.

3.3 Analysis

Theorem 12 (Completeness). *Let $\varepsilon > 0$ be a constant. We assume either that $n = \Omega(\lambda)$ and $\frac{\tau}{n} < 1 - p_{\mathsf{noise}} - \varepsilon$ or that $p_{\mathsf{noise}} = 0$. We assume that Com is a homomorphic bit commitment [Def. 8] and that ZKP_κ is complete [Def. 9]. The ProProx protocol is a PoPoK which fails with probability bounded by*

$$p_{\mathsf{Comp}} = 1 - \mathsf{Tail}\left(n, \tau, 1 - p_{\mathsf{noise}}\right)^s \tag{1}$$

when the challenge/response rounds are subject to a noise level of p_{noise} [Def. 2].

Proof. Completeness for $p_{\mathsf{noise}} = 0$ is trivial. Proving completeness when $\frac{\tau}{n} < 1 - p_{\mathsf{noise}} - \varepsilon$ is straightforward: in an iteration, we have less than τ noiseless rounds with probability $1 - \mathsf{Tail}(n, \tau, 1 - p_{\mathsf{noise}}) < e^{-2\varepsilon^2 n}$ due to the Chernoff-Hoeffding bound (Lemma 13), which is negligible since $n = \Omega(\lambda)$. Then, the completeness failure is bounded by p_{Comp} which is also negligible. □

We recall here some useful bound on the tail of the binomial distribution.

Lemma 13 (Chernoff-Hoeffding bound [13,23]). *For any ε, n, τ, q we have $\frac{\tau}{n} < q - \varepsilon \Longrightarrow \mathsf{Tail}(n, \tau, q) > 1 - e^{-2\varepsilon^2 n}$ and $\frac{\tau}{n} > q + \varepsilon \Longrightarrow \mathsf{Tail}(n, \tau, q) < e^{-2\varepsilon^2 n}$.*

We construct an extractor giving an output which is close to the secret.

Lemma 14 (Straightline extractor). *Under the assumption that Com is a perfectly binding homomorphic bit commitment, and that ZKP_κ is a κ-sound proof of membership, for any experiment, there is a PPT algorithm $\mathsf{Extract}$ which takes the views of all close-by participants and the transcript of the protocol seen by $\mathcal{V}$ and which aborts if $\mathcal{V}$ rejects, otherwise produces a vector $\mathsf{sk}' \in \{0,1\}^s$. For any w, the probability that $\mathcal{V}$ accepts and the Hamming distance between sk and sk' is at least $w+1$ is lower than*

$$p_{\mathsf{Sound}} = \mathsf{Tail}\left(\left\lceil \frac{n}{2} \right\rceil, \tau - \left\lfloor \frac{n}{2} \right\rfloor, \frac{1}{2}\right)^{w+1} + \kappa \tag{2}$$

We will often define $p_B = \mathsf{Tail}(\lceil \frac{n}{2} \rceil, \tau - \lfloor \frac{n}{2} \rfloor, \frac{1}{2})$. We note that if we assume that $s = \Omega(\lambda)$ and $\tau \geq n - (\frac{1}{2} - 2\varepsilon)\lceil \frac{n}{2} \rceil$ with a constant ε, we have $\frac{\tau - n + \lceil \frac{n}{2} \rceil}{\lceil \frac{n}{2} \rceil} \geq \frac{1}{2} + 2\varepsilon$. So, $p_B \leq e^{-8\varepsilon^2 n}$ due to the Chernoff-Hoeffding bound (Lemmma 13), which is negligible. This case will be use subsequently.

Proof. We assume that we have an experiment making $\mathcal{V}$ accept with probability p. We define $p_B = \mathsf{Tail}(\lceil \frac{n}{2} \rceil, \tau - \lfloor \frac{n}{2} \rfloor, \frac{1}{2})$.

We take the viewpoint of $\mathcal{V}$. Since we have a perfectly binding commitment, the value pk_j uniquely defines $\mathsf{sk}_j = \mathsf{Com}^{-1}(\mathsf{pk}_j)$, and the value of $A_{i,j}$ uniquely defines $a_{i,j} = \mathsf{Com}^{-1}(A_{i,j})$. (We stress that we need not compute these values, we just mathematically define them given the view of the verifier.) The purpose of the proof is to show that we can extract a good approximation of sk (i.e., at a distance lower than w), except with some negligible probability p_{Sound}.

Let $p = \Pr[\mathsf{Out}_V = 1]$. Let S be the event that for all j and for at least τ values of i (for each j), we have $r_{i,j} = a_{i,j} + c_{i,j}(b_{i,j} + \mathsf{sk}_j)$ (where the values are those seen by $\mathcal{V}$). In the case where the statement proven by ZKP_κ is true, for all j and $i \in I_j$, $z_{i,j}$ is clearly a commitment to zero. Due to the homomorphic property of Com, we know that $z_{i,j}$ is the commitment to $a_{i,j} + c_{i,j}(b_{i,j} + \mathsf{sk}_j) - r_{i,j}$. So, we deduce that S occurs. By using the κ-soundness of ZKP_κ (Def. 9), we deduce $\Pr[\mathsf{Out}_V = 1 | \neg S] \leq \kappa$. So, $\Pr[\neg S, \mathsf{Out}_V = 1] \leq \kappa$.

Since the $c_{i,j}$ challenges are sent in sequence, in what follows we denote by $c_q = c_{i_q, j_q}$ the qth challenge sent. We further denote by ρ all random coins of the experiment except those defining the challenges. So, we compute probabilities over the independent distributions of ρ and all $c_{i,j}$.

Thanks to Lemma 1, we can write $r_{i,j} = \mathsf{Algo}_{i,j}(c_{i,j}, \mathsf{Data}_{i,j})$ with $\mathsf{Data}_{i,j} = (\mathsf{Views}, \mathsf{Incoming}_{i,j}, \mathsf{Outgoing}_{i,j})$, where Views lists the initial view of close-by participants, $\mathsf{Incoming}_{i,j}$ gives the list of incoming messages from far away that they can see until the sender can see $c_{i,j}$, and $\mathsf{Outgoing}_{i,j}$ includes the list of outgoing messages from $\mathcal{V}$ before $c_{i,j}$. Note that $\mathsf{Data}_{i,j}$ can be computed

from the final views of the close-by participants but depends on the selected challenges before $c_{i,j}$. So, thanks to Lemma 1, we can compute in this case both $\mathsf{resp}_{i,j}(0) = \mathsf{Algo}_{i,j}(0, \mathsf{Data}_{i,j})$ and $\mathsf{resp}_{i,j}(1) = \mathsf{Algo}_{i,j}(1, \mathsf{Data}_{i,j})$ without rewinding (i.e., from the final view only). Since $r_{i,j}$ is supposed to be $a_{i,j} + c_{i,j}(b_{i,j} + \mathsf{sk}_j)$, we can compute the guess $\xi_{i,j} = \mathsf{resp}_{i,j}(1) - \mathsf{resp}_{i,j}(0) - b_{i,j}$ for sk_j. (Note that if the answer $r_{i,j}$ comes to $\mathcal{V}$ from far-away, we can still apply Lemma 1 and deduce that the answer is the same for $c_{i,j} = 0$ and $c_{i,j} = 1$, so $\xi_{i,j} = -b_{i,j}$.) In all cases, we can always compute the vectors $\xi_j = (\xi_{1,j}, \ldots, \xi_{n,j})$ of guesses for sk_j. The extractor is taking all $\mathsf{Algo}_{i,j}(., \mathsf{Data}_{i,j})$ to compute ξ_j then $\mathsf{sk}'_j = \mathsf{majority}(\xi_j)$ for all j.

Given c, if $a_{i,j} + c(b_{i,j} + \mathsf{sk}_j) = \mathsf{resp}_{i,j}(c)$, we say that the answer to $c_{i,j} = c$ is correct relative to the previous challenges (we recall that $\mathsf{Data}_{i,j}$ depends on all challenges which are sent before $c_{i,j}$). Based on ρ, we construct a binary tree T of depth ns in which a node at depth q corresponds to the selection of c_q. We denote by $G(c|c_1, \ldots, c_{q-1})$ the predicate that c is correct relative to $c_1, \ldots, c_{q-1}$. Let $S_T^{c_1,\ldots,c_q}$ be an s-tuple of integers such that $(S_T^{c_1,\ldots,c_q})_j = \#\{q' \leq q; j_{q'} = j, G(c_{q'}|c_1, \ldots, c_{q'-1})\}$. This counts how many good answers we had until step q for the $c_{.,j}$ challenges which are based on sk_j. We let S_ρ denote the event that $(S_T^{C_1,\ldots,C_{ns}})_j \geq \tau$ for all j where $C_1, \ldots, C_{ns}$ are the random challenges from the experiment. $\mathcal{V}$ only accepts when S_ρ holds. Let $R_T^{c_1,\ldots,c_q}$ be an s-tuple of integers such that $(R_T^{c_1,\ldots,c_q})_j = \#\{q' \leq q; j_{q'} = j, G(0|c_1, \ldots, c_{q'-1}), G(1|c_1, \ldots, c_{q'-1})\}$. This counts how many times both values lead to good answers for the $c_{.,j}$ challenges. If $G(0|c_1, \ldots, c_{q'-1})$ and $G(1|c_1, \ldots, c_{q'-1})$ hold, then $\xi_{i_q,j_q} = \mathsf{sk}_j$. So, if $(R_T^{c_1,\ldots,c_{ns-1}})_j \geq \lfloor \frac{n}{2} \rfloor + 1$, we have $\mathsf{sk}'_j = \mathsf{sk}_j$. We let W_ρ be the number of j such that $(R_T^{C_1,\ldots,C_{ns}})_j \leq \lfloor \frac{n}{2} \rfloor$. We show below that for all ρ, $\Pr[S_\rho, W_\rho > w] \leq p_{\mathsf{Sound}} - \kappa$ over the distribution of the $c_{i,j}$. By averaging over ρ, we have $\Pr[S, W > w] \leq p_{\mathsf{Sound}} - \kappa$. Thus, by splitting with the S and $\neg S$ events,

$$\Pr[W > w, \mathsf{Out}_V = 1] \leq \Pr[\neg S, \mathsf{Out}_V = 1] + \Pr[S, W > w] \leq p_{\mathsf{Sound}}$$

So, having that $\mathcal{V}$ accepts and the extractor gives at least $w + 1$ errors occurs with probability bounded by p_{Sound}, which is what we wanted to prove.

To show that $\Pr[S_\rho, W_\rho > w] \leq p_{\mathsf{Sound}} - \kappa$ in the fixed tree T, we first modify the tree in a way which only make this probability increase. Namely, we add more $G(c_q|c_1, \ldots, c_{q-1})$ so that for all j, $(R_T^{c_1,\ldots,c_{ns}})_j$ is either n or $\lfloor \frac{n}{2} \rfloor$. Then, we show a more general property. We consider a balanced binary tree of depth q with some indexing $q \leftrightarrow (i_q, j_q)$. We denote q_j the number of $k \in \{1, \ldots, q\}$ such that $j_k = j$. So, $q = q_1 + \cdots + q_s$. We let $W_T(J, w)$ be the event that for at least $w + 1$ values of $j \in J$ we have $(R_T^{c_1,\ldots,c_q})_j \leq q_j - \lceil \frac{n}{2} \rceil$, for other values of j we have $(R_T^{c_1,\ldots,c_q})_j = q_j$, and for all $j \in J$ we have $(S_T^{C_1,\ldots,C_q})_j \geq \tau - (n - q_j)$.

We show that for all J, τ_j's, and μ_j's, we have

$$\begin{aligned}&\Pr\left[W_T(J,w), \bigwedge_{j\notin J}(S_T^{C_1,\ldots,C_q})_j \geq \tau_j, (R_T^{C_1,\ldots,C_q})_j \leq \mu_j\right]\\ &\leq p_B^{w+1} \times \prod_{j\notin J} \mathsf{tail}\left(q_j-\mu_j, \tau_j-\mu_j, \frac{1}{2}\right) \qquad (3)\end{aligned}$$

Then, we apply it with $J = \{1,\ldots,s\}$. We obtain $\Pr[S_\rho, W_\rho > w] \leq p_B^{w+1} = p_{\mathsf{Sound}} - \kappa$.

The (3) property is proven by induction on q. It is trivial for $q = 0$. Assuming it holds for $q-1$, we prove it for q by looking at the two subtrees T_0 and T_1 of T. We have

$$\begin{array}{lll}(S_T^{c_1,\ldots,c_q})_j{=}(S_{T_{c_1}}^{c_2,\ldots,c_q})_j & (R_T^{c_1,\ldots,c_q})_j{=}(R_{T_{c_1}}^{c_2,\ldots,c_q})_j & \text{if } j\neq j_1\\ (S_T^{c_1,\ldots,c_q})_j{=}(S_{T_{c_1}}^{c_2,\ldots,c_q})_j + 1_{G(c_1)} & (R_T^{c_1,\ldots,c_q})_j{=}(R_{T_{c_1}}^{c_2,\ldots,c_q})_j + 1_{G(0),G(1)} & \text{if } j=j_1\end{array}$$

If $j_1 \notin J$ or $G(0)\wedge G(1)$ holds, $W_T(J,w)$ is equivalent to $W_{T_{C_1}}(J',w')$ for $J'=J$ and $w'=w$. If now $j_1\in J$ and $\neg G(0)\vee\neg G(1)$ holds, we define $\tau_{j_1} = \tau-(n-q_{j_1})$, $\mu_{j_1} = q_{j_1} - \left\lceil\frac{n}{2}\right\rceil$, $J' = J-\{j_1\}$, and $w'=w-1$. Then, $W_T(J,w)$ is equivalent to, $(S_T^{C_1,\ldots,C_q})_{j_1} \geq \tau_{j_1}$, $(R_T^{c_1,\ldots,c_q})_{j_1} = \mu_{j_1}$, and $W_{T_{C_1}}(J',w')$. So,

$$\begin{aligned}&\Pr\left[W_T(J,w), \bigwedge_{j\notin J}(S_T^{C_1,\ldots,C_q})_j \geq \tau_j, (R_T^{C_1,\ldots,C_q})_j \leq \mu_j\right]\\ &= \sum_{c_1=0}^{1}\Pr\left[W_T(J,w), \bigwedge_{j\notin J}(S_T^{c_1,C_2,\ldots,C_q})_j \geq \tau_j, (R_T^{c_1,C_2,\ldots,C_q})_j \leq \mu_j, C_1=c_1\right]\\ &\leq \sum_{c_1=0}^{1}\Pr\left[W_{T_{c_1}}(J',w'), \bigwedge_{j\notin J'}(S_T^{c_1,C_2,\ldots,C_q})_j \geq \tau_j, (R_T^{c_1,C_2,\ldots,C_q})_j \leq \mu_j, C_1=c_1\right]\\ &= \sum_{c_1=0}^{1}\frac{1}{2}\Pr\left[W_{T_{c_1}}(J',w'), \bigwedge_{j\notin J'}(S_{T_{c_1}}^{C_2,\ldots,C_q})_j \geq \tau_j - 1_{j=j_1,G(c_1)}, (R_{T_{c_1}}^{C_2,\ldots,C_q})_j \leq \mu_j - 1_{j=j_1,G(0),G(1)}\right]\\ &\leq \sum_{c_1=0}^{1}\frac{1}{2}p_B^{w'+1}\prod_{j\notin J'}\mathsf{tail}\left(q_j - 1_{j=j_1} - \mu_j + 1_{j=j_1,G(0),G(1)}, \tau_j - 1_{j=j_1,G(c_1)} - \mu_j + 1_{j=j_1,G(0),G(1)}, \tfrac{1}{2}\right)\end{aligned}$$

When $j_1\in J'$, this proves (3). For $j_1\notin J'$, we obtain

$$\begin{aligned}&p_B^{w'+1}\left(\sum_{c_1=0}^{1}\frac{1}{2}\mathsf{tail}\left(q_{j_1}-1-\mu_{j_1}+1_{G(0),G(1)}, \tau_{j_1}-1_{G(c_1)}-\mu_{j_1}+1_{G(0),G(1)}, \frac{1}{2}\right)\right)\\ &\times\prod_{\substack{j\notin J'\\ j\neq j_1}}\mathsf{tail}\left(q_j-\mu_j, \tau_j-\mu_j, \frac{1}{2}\right)\end{aligned}$$

If both $G(0)$ and $G(1)$ are true the sum in parentheses is clearly equal to $\mathsf{tail}(q_{j_1} - \mu_{j_1}, \tau_{j_1} - \mu_{j_1}, \frac{1}{2})$. If either $G(0)$ or $G(1)$ is true but not both, the sum is

$$\frac{1}{2}\mathsf{tail}\left(q_{j_1} - 1 - \mu_{j_1}, \tau_{j_1} - \mu_{j_1}, \frac{1}{2}\right) + \frac{1}{2}\mathsf{tail}\left(q_{j_1} - 1 - \mu_{j_1}, \tau_{j_1} - 1 - \mu_{j_1}, \frac{1}{2}\right)$$

which is also equal to $\mathsf{tail}(q_{j_1} - \mu_{j_1}, \tau_{j_1} - \mu_{j_1}, \frac{1}{2})$. Finally, if neither $G(0)$ nor $G(1)$ hold, the sum is $\mathsf{tail}(q_{j_1} - 1 - \mu_{j_1}, \tau_{j_1} - \mu_{j_1}, \frac{1}{2})$ which is bounded by $\mathsf{tail}(q_{j_1} - \mu_{j_1}, \tau_{j_1} - \mu_{j_1}, \frac{1}{2})$. So, in all cases this proves (3). □

Theorem 15 (Soundness). *We assume that* Com *is a perfectly binding homomorphic bit commitment, and that* ZKP_κ *is a* κ*-sound proof of membership. ProProx is a* p_{Sound}*-sound proof of proximity, where* p_{Sound} *is defined by (2).*

More precisely, for all constant w*, if the experiment succeeds with probability* $p > p_{\mathsf{Sound}}$ *there exists an extractor following Def. 5 with complexity*

$$T_{\mathsf{exp}}.\mathcal{O}\left(\frac{1}{p - p_{\mathsf{Sound}}}\right) + T_{\mathsf{Com}_H}.\mathcal{O}\left(B_w^s\right)$$

where T_{exp} *is the complexity of the experiment,* T_{Com_H} *is the complexity to compute* Com_H*, and* $B_w^s = \sum_{i=0}^{w} \binom{s}{i}$*. The second term is actually the complexity of an exhaustive search with* B_w^s *iterations on* sk *until* $\mathsf{pk} = \mathsf{Com}_H(\mathsf{sk})$*.*

To use (2) with concrete parameters, w is chosen as the maximal value such that an adversary could afford an exhaustive search of B_w^s trials.

Proof. We can use the extractor of Lemma 14 on views taken from an experiment run. If $\mathcal{V}$ rejects, the extraction produces nothing. We iterate this extraction $\mathcal{O}(\frac{1}{p})$ times until one experiment succeeds. So, we obtain for sure a guess sk' for sk (with possible errors). The probability that at least w errors occurs in the extracted pairs is bounded by $\frac{p_{\mathsf{Sound}}}{p}$. When there are less errors, we can correct them by exhaustive search in time $T_{\mathsf{Com}_H}.\mathcal{O}(B_w^s)$ (which is polynomial). If this fails (i.e., if it gives no preimage of pk by Com_H) as some extracted pairs may have too many errors, we can just iterate. With a number of iterations of $\mathcal{O}\left(\left(1 - \frac{p_{\mathsf{Sound}}}{p}\right)^{-1}\right)$, we finally extract sk. The overall expected complexity is thus $\mathsf{Poly}(\lambda)/(p - p_{\mathsf{Sound}})$. More precisely, it is $T_{\mathsf{exp}}.\mathcal{O}\left(\frac{1}{p - p_{\mathsf{Sound}}}\right) + T_{\mathsf{Com}_H}.\mathcal{O}\left(B_w^s\right)$. □

Our technique to prove HP-security relies on Lemma 14 and zero-knowledge.

Lemma 16 (Zero-knowledge). *Under the assumption that* Com *is a computationally hiding bit commitment and that* ZKP_κ *is a computationally zero-knowledge proof of membership, The ProProx protocol is zero-knowledge following Def. 10. More precisely, for any malicious verifier, given a simulator for* ZKP_κ *of complexity* T_{Sim} *producing views which are* p_{ZKP}*-indistinguishable to the real ones, we construct a simulator for ProProx of complexity* $T_{\mathsf{Sim}} + ns.T_{\mathsf{Com}}$ *and producing views which are* p_{ZK}*-indistinguishable to the real ones, where*

$$p_{\mathsf{ZK}} = p_{\mathsf{ZKP}} + ns.p_{\mathsf{Com}} \tag{4}$$

where p_{Com} is the bound on the hiding property of Com.

Proof. We have to prove that, given two participants $P(\mathsf{sk})$ and $V^*(\mathsf{pk}, \mathsf{aux})$, there exists a simulator $S(\mathsf{pk}, \mathsf{aux})$ such that $V^*(\mathsf{pk}, \mathsf{aux}) \leftrightarrow P(\mathsf{sk})$ produces a view of $V^*(\mathsf{pk}, \mathsf{aux})$ which is computationally indistinguishable from the output of $S(\mathsf{pk}, \mathsf{aux})$. We actually construct a sequence of simulations. We define an interactive $V'(\mathsf{pk}, \mathsf{aux})$ to replace $V^*(\mathsf{pk}, \mathsf{aux})$, and some interactive $P'(\mathsf{sk})$ and P'' to replace $P(\mathsf{sk})$.

We denote $\bar{z}$ the vector of all $z_{i,j}$ for $j = 1, \ldots, s$ and $i \in I_j$, and $\bar{\zeta}$ the vector of all $\zeta_{i,j}$. We split $V^*(\mathsf{pk}, \mathsf{aux})$ into two protocols $V_1(\mathsf{pk}, \mathsf{aux})$ and $V_2(\bar{z}, \mathsf{aux}')$, where V_1 mimics V^* until the $\mathsf{ZKP}_\kappa(\bar{z} : \bar{\zeta})$ protocol must start. V_2 executes only $\mathsf{ZKP}_\kappa(\bar{z} : \bar{\zeta})$ where aux' is the final view of $V_1(\mathsf{pk}, \mathsf{aux})$. The final view of $V_2(\bar{z}, \mathsf{aux}')$ is of form $v = (\bar{z}, \mathsf{aux}', t)$. We write $g(v) = (\mathsf{aux}', t)$, which is the final view of $V^*(\mathsf{pk}, \mathsf{aux})$. Similarly, we split $P(\mathsf{sk})$ into $P_1(\mathsf{sk})$ and $P_2(\mathsf{sk}, u)$ where (sk, u) is the view of $P_1(\mathsf{sk})$. Running either $V^*(\mathsf{pk}, \mathsf{aux}) \leftrightarrow P(\mathsf{sk})$ and taking the final view of V^*, or $V_1(\mathsf{pk}, \mathsf{aux}) \leftrightarrow P_1(\mathsf{sk})$, $V_2(\bar{z}, \mathsf{aux}') \leftrightarrow P_2(\mathsf{sk}, u)$, then taking $g(v)$ is the same. This simulation is illustrated on the left-hand side of Fig. 3.

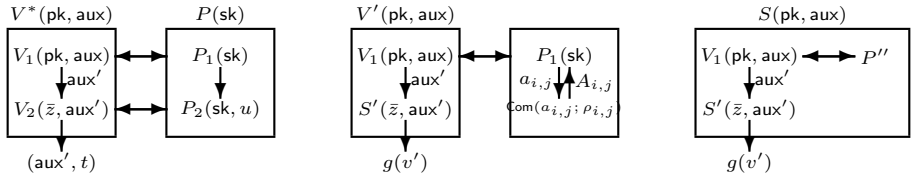

Fig. 3. Applying a ZK Reduction.

First, $V'(\mathsf{pk}, \mathsf{aux})$ runs a simulation of $V_1(\mathsf{pk}, \mathsf{aux})$ interacting with $P_1(\mathsf{sk})$. Then, $V'(\mathsf{pk}, \mathsf{aux})$ runs the simulator $S'(\bar{z}, \mathsf{aux}')$ of the $\mathsf{ZKP}_\kappa(\bar{z} : \bar{\zeta})$ protocol associated to the verifier $V_2(\bar{z}, \mathsf{aux}')$ with complexity T_{Sim}. Let v' be the output of $S'(\bar{z}, \mathsf{aux}')$. Finally, $V'(\mathsf{pk}, \mathsf{aux})$ produces $g(v')$ as an output. This simulation is illustrated on the middle of Fig. 3. Due to the zero-knowledge property of $\mathsf{ZKP}_\kappa(\bar{z} : \bar{\zeta})$, v' is p_{ZKP}-indistinguishable from the final view of $V_2(\bar{z}, \mathsf{aux}')$. So, the final view of $V'(\mathsf{pk}, \mathsf{aux})$ in $V'(\mathsf{pk}, \mathsf{aux}) \leftrightarrow P_1(\mathsf{sk})$ and the final view of $V^*(\mathsf{pk}, \mathsf{aux})$ in $V^*(\mathsf{pk}, \mathsf{aux}) \leftrightarrow P(\mathsf{sk})$ are p_{ZKP}-indistinguishable.

Note that $P_1(\mathsf{sk})$ makes no longer extra use of the coins ρ_i's (as $P_2(\mathsf{sk}, u)$ does in ZKP_κ). So, the commitment can be outsourced to a challenger playing the real-or-random hiding game for Com. We modify $P_1(\mathsf{sk})$ into an algorithm $P'(\mathsf{sk})$ who sets $A_{i,j}$ to the commitment to some random bit instead of $a_{i,j}$. Thanks to the hiding property of Com applied ns times, the output of $V'(\mathsf{pk}, \mathsf{aux}) \leftrightarrow P_1(\mathsf{sk})$ and of $V'(\mathsf{pk}, \mathsf{aux}) \leftrightarrow P'(\mathsf{sk})$ are $ns.p_{\mathsf{Com}}$-indistinguishable.

Finally, r'_i in $P'(\mathsf{sk})$ is now uniformly distributed and independent from all the rest, so we change $P'(\mathsf{sk})$ into an algorithm P'' which sends a random r'_i instead. Note that P'' no longer needs sk. So, the view of V^* in $V^*(\mathsf{pk}, \mathsf{aux}) \leftrightarrow P(\mathsf{sk})$ and

the output of $V'(\mathsf{pk}, \mathsf{aux}) \leftrightarrow P''$ are indistinguishable. This defines a simulator $S(\mathsf{pk}, \mathsf{aux})$, as illustrated on the right-hand-side of Fig. 3. □

Theorem 17 (HP-Security). *We assume that* Com *is a perfectly binding, and computationally hiding homomorphic bit commitment, that* Com_H *is one-way, and that* ZKP_κ *is a* κ*-sound computationally zero-knowledge proof of membership for* $\kappa = \mathsf{negl}(\lambda)$*. For all* w*, we take an experiment with* r *instances of the honest prover and we split it into* r *successive experiments, with one honest prover per splitted experiment. Each of them is associated to a simulator* Sim_i *for the* ZKP_κ *protocol and we denote by* T_{Sim_i} *the complexity of the simulator. Assuming that the experiment succeeds with probability at least*

$$p_{\mathsf{Sec}} = p_{\mathsf{Sound}} + r.p_{\mathsf{ZK}} + p_{\mathsf{Com}} \tag{5}$$

(where p_{Sound} *is defined by (2)) we construct an inversion algorithm for* Com_H *with complexity*

$$\sum_{i=1}^{r} T_{\mathsf{Sim}_i} + T_{\mathsf{Com}_H}.\mathcal{O}\left(B_w^s\right) + rns.T_{\mathsf{Com}}$$

where p_{ZK}*,* p_{Com}*, and* T_{Com} *are defined as in Lemma 16,* T_{Com_H} *is the complexity of* Com_H*, and* B_w^s *is defined in Th. 15. For* $s = \Omega(\lambda)$ *and that* $\tau \geq n - (\frac{1}{2} - 2\varepsilon)\lceil \frac{n}{2} \rceil$ *with a constant* ε*,* p_{Sec} *is negligible. So, ProProx is HP-secure.*

Proof. We consider an experiment exp with an honest always far-away prover. Let p be the probability that $\mathcal{V}$ accepts. We want to show that $p = \mathsf{negl}(\lambda)$.

We define $p_B = \mathsf{Tail}(\lceil \frac{n}{2} \rceil, \tau - \lfloor \frac{n}{2} \rfloor, \frac{1}{2})$. We use Lemma 14 to extract the vector sk' when $\mathcal{V}$ accepts, with at least w errors to sk with probability bounded by p_{Sound}. Then, by a $T_{\mathsf{Com}_H}.\mathcal{O}(B_w^s)$-time exhaustive search on the errors, we correct sk' and check if we obtain a preimage of Com_H like in Th. 15. This gives sk in polynomial time and a probability of success of at least $p - p_{\mathsf{Sound}}$, by playing with some non-concurrent instances of $P(\mathsf{sk})$. For each of the non-concurrent instances of $P(\mathsf{sk})$, we then use the ZK property of $P(\mathsf{sk})$ to construct an algorithm inverting Com_H with probability of success of at least $p - p_{\mathsf{Sound}} - r.p_{\mathsf{ZK}}$, where r is the number of $P(\mathsf{sk})$ instances in one experiment. By assumption on Com_H, this must be bounded by some negligible p_{Com}. So, we have $p \leq p_{\mathsf{Sec}}$ with p_{Sec} defined by Eq. (5). The values κ, p_{ZK}, and p_{Com} are negligible, while r is polynomial and w is constant. So, p_{Sound} and p_{Sec} are negligible. □

Note that a malicious prover can run a distance fraud in each round such that $b_{i,j} = \mathsf{sk}_j$, as $r_{i,j}$ no longer depends on $c_{i,j}$. For $\mathsf{sk} = 0$ (as allowed in the malicious prover model) and $b = 0$, this can be done in all rounds, so we can have a distance fraud. There is no contradiction with soundness: an observer seeing that the verifier accepts can deduce that sk_j is likely to be zero, for all j. So, the malicious prover leaks.

To have distance fraud resistance, we adopt a trick from DB2 [10]: we select a vector b_j with Hamming weight $\lfloor \frac{n}{2} \rfloor$ so that half of the rounds will really use $c_{i,j}$. Actually, b_j has a maximal distance to the repetition code.

Theorem 18 (DF-Resistance). *We assume that* Com *is a perfectly binding bit commitment and that* ZKP_κ *is a* κ*-sound proof of membership for* $\kappa = \mathsf{negl}(\lambda)$*. Every distance fraud in ProProx succeeds with a probability bounded by*

$$p_{\mathsf{DF}} = \begin{cases} \kappa + \mathsf{Tail}\left(\left\lfloor \frac{n}{2} \right\rfloor, \tau - \left\lceil \frac{n}{2} \right\rceil, \frac{1}{2}\right)^s & \textit{in variant I} \\ \kappa + \left(\frac{3}{4}\right)^{ns} & \textit{in variant II} \end{cases} \tag{6}$$

For $n = \Omega(\lambda)$ *and* $\tau \geq n - (\frac{1}{2} - 2\varepsilon)\lfloor \frac{n}{2} \rfloor$ *with a constant* ε*,* p_{DF} *is negligible. So, ProProx is DF-resistant.*

Proof. We concentrate on the jth iteration. Let w_j be the weight of the vector $b_j \oplus (\mathsf{sk}_j, \ldots, \mathsf{sk}_j)$. Due to the perfectly binding property, the view of $\mathcal{V}$ uniquely defines sk_j and $a_{i,j}$. Thanks to Lemma 1, $r_{i,j}$ is obtained from $\mathsf{Incoming}_{\mathcal{V}}(\mathsf{Far})$, so independent from $c_{i,j}$. So, for $b_{i,j} \neq \mathsf{sk}_j$ (which happens for w_j rounds), we have that $\Pr[r_{i,j} = a_{i,j} + c_{i,j} b_{i,j} + c_{i,j}\mathsf{sk}_j] = \frac{1}{2}$. So, the probability that the statement in ZKP_κ holds is bounded by $\prod_{j=1}^{s} \mathsf{Tail}(w_j, \tau - n + w_j, \frac{1}{2})$ which is negligible for $\frac{\tau - n + w_j}{w_j} \geq \frac{1}{2} + 2\varepsilon$, due to the Chernoff-Hoeffding bound (Lemma 13) for $n = \Omega(\lambda)$. Due to the fact that ZKP_κ is sound, the verifier accepts with probability bounded by $\kappa + \mathsf{Tail}(\lfloor \frac{n}{2} \rfloor, \tau - \lceil \frac{n}{2} \rceil, \frac{1}{2})^s$ in the first variant of the protocol. In the second variant, we first note that $E(\mathsf{Tail}(w_j, \tau - n + w_j, \frac{1}{2})) = E(2^{-w_j}) = \left(\frac{3}{4}\right)^n$ since $n = \tau$. So, the verifier accepts with probability bounded by $\kappa + \left(\frac{3}{4}\right)^{ns}$. □

We also treat distance hijacking [14] specifically.

Theorem 19 (DH-Resistance). *We assume that* Com *is a perfectly binding bit commitment, that* Com_H *is one-way, and that* ZKP_κ *is a* κ*-sound proof of membership for* $\kappa = \mathsf{negl}(\lambda)$*. For any constant* w*, given a DH attack succeeding with probability at least*

$$p_{\mathsf{DH}} = \begin{cases} \kappa + \mathsf{Tail}\left(n, \tau, \frac{1}{2}\right)^w & \textit{in variant I} \\ \kappa + \left(\frac{1}{2}\right)^{w\lceil \frac{n}{2} \rceil} & \textit{in variant II} \end{cases} \tag{7}$$

we can construct an inversion algorithm for Com_H *with complexity* $T_{\mathsf{Com}_H}.\mathcal{O}(s^w)$ *where* T_{Com_H} *is the complexity of* Com_H*. For* $n = \Omega(\lambda)$ *and* $\tau \geq n - (\frac{1}{2} - 2\varepsilon)\lceil \frac{n}{2} \rceil$ *with a constant* ε*,* p_{DH} *is negligible. So, ProProx is DH-resistant.*

Proof. We consider a DH attack with a malicious prover $P^* = P(\mathsf{sk})$ with a public key pk and an honest prover $P' = P(\mathsf{sk}')$ with a public key pk'. During the initialization, P' chooses some $a'_{i,j}$ bits which are committed in some $A'_{i,j}$. He also receives some $b'_{i,j}$ bits while V has some $b_{i,j}$ bits. The malicious prover P^* sends some $A_{i,j}$ to V and we denote $a_{i,j} = \mathsf{Com}^{-1}(A_{i,j})$. During the challenge phase, V and P' interact in a noisy channel. We write $r'_{i,j} = e_{i,j} + a'_{i,j} + c_{i,j}(b'_{i,j} + \mathsf{sk}'_i)$ the response by P', where $\Pr[e_{i,j} = 1] = p_{\mathsf{noise}}$ and all $e_{i,j}$ are independent. As the verifier expects $r_{i,j} = a_{i,j} + c_{i,j}(b_{i,j} + \mathsf{sk}_j)$, this holds if and only if $a_{i,j} + a'_{i,j} + e_{i,j} = c_{i,j}(b_{i,j} + b'_{i,j} + \mathsf{sk}_j + \mathsf{sk}'_j)$. This can only hold with probability $\frac{1}{2}$ when the content of the parenthesis is equal to 1.

Let w_j be the number of i such that $b_{i,j} + b'_{i,j} \neq \mathsf{sk}_j + \mathsf{sk}'_j$. Clearly, the jth iteration has τ correct responses with probability bounded by $\mathsf{Tail}(w_j, \tau - n + w_j, \frac{1}{2})$. If $w_j \geq \lceil \frac{n}{2} \rceil$, this is bounded by $\mathsf{Tail}(\lceil \frac{n}{2} \rceil, \tau - \lfloor \frac{n}{2} \rfloor, \frac{1}{2})$. Otherwise, the probability is bounded by 1, but the majority of $b_{i,j} + b'_{i,j}$ matches $\mathsf{sk}_j + \mathsf{sk}'_j$ so the adversary deduces sk'_j. Let w be the number of j such that $w_j \geq \lceil \frac{n}{2} \rceil$. Clearly, the responses are overall acceptable with a probability bounded by $\mathsf{Tail}(\lceil \frac{n}{2} \rceil, \tau - \lfloor \frac{n}{2} \rfloor, \frac{1}{2})^w$. Due to the soundness of ZKP_κ, the probability of success is bounded by $\kappa + \mathsf{Tail}(\lceil \frac{n}{2} \rceil, \tau - \lfloor \frac{n}{2} \rfloor, \frac{1}{2})^w$. Furthermore, by the majority decoding, we have an inversion algorithm for Com_H with complexity $\mathcal{O}\left(s^w . T_{\mathsf{Com}_H}\right)$.

We note that when b is fixed in the protocol, w_j is equal to either 0 or n. So, in the first variant of the protocol, the same analysis as above concludes to a probability of success bounded by $\kappa + \mathsf{Tail}(n, \tau, \frac{1}{2})^w$. In the second variant, we have $n = \tau$ and the probability simplifies to $2^{-w\lceil \frac{n}{2} \rceil}$. □

3.4 Simplification in the Noiseless Communications Case

The protocol could be simplified in noiseless environment. For this, we would take $n = \tau$. There is clearly no need to agree on I_j which is always the full set $I_j = \{1, \ldots, n\}$. The protocol is much simpler. Variant I and Variant II use in (6) the bounds $\left(\frac{1}{2}\right)^{\lfloor \frac{n}{2} \rfloor s}$ and $\left(\frac{3}{4}\right)^{ns}$, respectively. For n even, the Variant I is better, but if we want to lower n down to $n = 1$, we must use Variant II.

3.5 Concrete Parameters

To see if the proven bounds Eq. (2), Eq. (5), Eq. (6), and Eq. (7) are tight or not, we look at the best known attacks. They correspond to the following probabilities of success:

$$p^{\mathrm{I}}_{\mathsf{DF}} = \mathsf{Tail}\left(\left\lfloor \frac{n}{2} \right\rfloor, \tau - \left\lceil \frac{n}{2} \right\rceil, \frac{1}{2}\right)^s \qquad p^{\mathrm{II}}_{\mathsf{DF}} = \left(\frac{3}{4}\right)^{ns}$$

$$p_{\mathsf{Sec}} = p_{\mathsf{DH}} = \mathsf{Tail}\left(n, \tau, \frac{1}{2}\right)^s \qquad p_{\mathsf{Sound}} = \mathsf{Tail}\left(\left\lceil \frac{n}{2} \right\rceil, \tau - \left\lfloor \frac{n}{2} \right\rfloor, \frac{1}{2}\right)^s$$

where p_{DF} depends on Variant I or Variant II. The DF attack with success probability p_{DF} consists of guessing c_i in half of the rounds for which $b_{i,j} \neq \mathsf{sk}_j$. So, the proven bound Eq. (6) is pretty tight.

The MF attack with success probability p_{Sec} follows the post-ask strategy: the adversary first guesses the answers to all challenges then plays with the prover with the same challenges. Clearly, there is a gap between p_{Sec} and the proven bound of Eq. (5). The DH case is similar: the malicious prover commits to some random $a_{i,j}$ which will make the correspondence between $c_{i,j}$ and $r_{i,j}$ between P' correct for P with probability $\frac{1}{2}$.

The TF attack with success probability p_{Sound} consists of giving a table of all $c'_{i,j} \mapsto r'_{i,j}$ which is corrupted in half of the rounds (selected at random) in each iteration, so that it gives no information about sk_j. Having the table $c'_{i,j} \mapsto r'_{i,j}$

corrupted means that one of the two entries (selected at random) is flipped. There is also a gap with the proven bound Eq. (2).

So, it may be the case that either the bounds Eq. (2), Eq. (5), and Eq. (7) can be improved, or that there exist better attacks. To select the parameters, we could either use the *proven* bounds or the above equations based on the best known attacks that we call the *empirical* bounds.

As concrete parameters, we could suggest $\lambda = 80$ bits as the security parameter and a modulus N of 1 024 bits. Then, we look for n and τ which minimize the total number of rounds n while keeping $p_{\mathsf{Comp}} \approx 1 - 2^{-7}$ and different objectives: we propose several vectors of parameters to reach the online security of either $\sigma = 2^{-20}$ (*high*) or $\sigma = 2^{-10}$ (*low*), with *proven* bounds or *empirical* bound, and with either $p_{\mathsf{noise}} = 1\%$ or the noiseless variant ($p_{\mathsf{noise}} = 0$) from Section 3.4. In the computation of Eq. (2) and Eq. (5), we took $\kappa = \frac{\sigma}{4}$ and w such that the exhaustive search is not more for a random s-bit string, i.e., $B_w^s \leq 2^\lambda$. For that, we took $s = \lambda + 1$ and $w = \lceil \frac{s}{2} \rceil$.

The total number of rounds is ns.

security	bounds	p_{noise}	ns	s	n	w	τ	Variant	p_{Comp}	p_{DF}	p_{Sec}	p_{Sound}	p_{DH}
high	proven	1%	648	81	8	41	6	I	$1-2^{-8}$	2^{-22}	2^{-21}	2^{-21}	2^{-22}
high	empirical	1%	640	80	8	–	6	I	$1-2^{-8}$	2^{-43}	2^{-223}	2^{-43}	2^{-223}
low	proven	1%	567	81	7	41	5	I	$1-2^{-9}$	2^{-12}	2^{-12}	2^{-12}	2^{-12}
low	empirical	1%	560	80	7	–	5	I	$1-2^{-9}$	2^{-15}	2^{-171}	2^{-43}	2^{-171}
high	proven	0	162	81	2	41	2	I	1	2^{-22}	2^{-22}	2^{-22}	2^{-22}
high	empirical	0	160	80	2	–	2	I	1	2^{-80}	2^{-160}	2^{-80}	2^{-160}
low	proven	0	162	81	2	41	2	I	1	2^{-12}	2^{-12}	2^{-12}	2^{-12}
low	empirical	0	160	80	2	–	2	I	1	2^{-80}	2^{-160}	2^{-80}	2^{-160}
high	proven	0	81	81	1	41	1	II	1	2^{-22}	2^{-22}	2^{-22}	2^{-22}
high	empirical	0	80	80	1	–	1	II	1	2^{-33}	2^{-80}	2^{-80}	2^{-80}

Clearly, there is a big gap between proven and empirical parameters in the high security values. We can observe that the noise has a huge impact on the complexity. Sometimes, the obtained parameters with low and high security are the same. This comes from p_{Sec} and p_{Sound} being basically equal to κ. As we can see, the noiseless case with $n = 1$ and $s = 80$ offers pretty efficient parameters.

For other parameters, ns may look high. However, we shall keep in mind that distance bounding rounds are exchanging bits very quickly. A challenge/response round shall take much less than 100ns. So, even by "wasting" 10μs in between rounds, $ns = 648$ takes less than 7ms. So, the round-complexity is not so important. What matters more is the impact on other cryptographic operations. Indeed, the prover needs to compute ns commitments, so $\frac{3}{2}ns$ multiplications, and $-\tau s \log_2 \kappa$ parallel rounds of ZKP, so $-\frac{3}{2}\tau s \log_2 \kappa$ multiplications. So, $\frac{3}{2}(n - \tau \log_2 \kappa)s$ multiplications in total. Hence, we shall consider the regular tricks to perform batch ZKP proofs to reduce the complexity.

4 Conclusion

We proposed ProProx, the very first PoPoK addressing soundness. It is provably secure. A remaining challenge is to construct a more efficient PoPoK. Another open question would be to have a tight security proof for ProProx.

Acknowledgements. This work was partly sponsored by the ICT COST Action IC1403 Cryptacus in the EU Framework Horizon 2020.

References

1. G. Avoine, M. Bingöl, S. Kardas, C. Lauradoux, B. Martin. A Framework for Analyzing RFID Distance Bounding Protocols. *Journal of Computer Security*, vol. 19(2), pp. 289–317, 2011.
2. G. Avoine, A. Tchamkerten. An Efficient Distance Bounding RFID Authentication Protocol: Balancing False-Acceptance Rate and Memory Requirement. In *Information Security ISC'09*, Pisa, Italy, Lecture Notes in Computer Science 5735, pp. 250–261, Springer-Verlag, 2009.
3. A. Bay, I. Boureanu, A. Mitrokotsa, I. Spulber, S. Vaudenay. The Bussard-Bagga and Other Distance-Bounding Protocols under Attacks. In *INSCRYPT'12*, Beijing, China, Lecture Notes in Computer Science 7763, pp. 371–391, Springer-Verlag, 2012.
4. T. Beth, Y. Desmedt. Identification Tokens or: Solving The Chess Grandmaster Problem. In *Advances in Cryptology CRYPTO'90*, Santa Barbara, California, U.S.A., Lecture Notes in Computer Science 537, pp. 169–176, Springer-Verlag, 1991.
5. I. Boureanu, A. Mitrokotsa, S. Vaudenay. Secure & Lightweight Distance-Bounding. In *Lightweight Cryptography for Security and Privacy LightSec'13*, Gebze, Turkey, Lecture Notes in Computer Science 8162, pp. 97–113, Springer-Verlag, 2013.
6. I. Boureanu, A. Mitrokotsa, S. Vaudenay. Practical & Provably Secure Distance-Bounding. IACR Eprint 2013/465 report, 2013.
7. I. Boureanu, A. Mitrokotsa, S. Vaudenay. Towards Secure Distance Bounding. In *Fast Software Encryption'13*, Singapore, Lecture Notes in Computer Science 8424, pp. 55–67, Springer-Verlag, 2013.
8. I. Boureanu, A. Mitrokotsa, S. Vaudenay. Practical & Provably Secure Distance-Bounding. To appear in the proceedings of ISC'13.
9. I. Boureanu, K. Mitrokotsa, S. Vaudenay. Practical and Provably Secure Distance-Bounding. *Journal of Computer Security (JCS)*, vol. 23 (2), pp. 229–257, 2015.
10. I. Boureanu, S. Vaudenay. Optimal Proximity Proofs. In *Information Security and Cryptology Inscrypt'14*, Beijing, China, Lecture Notes in Computer Science 8957, pp. 170–190, Springer-Verlag, 2014. Eprint 2014/693.
11. S. Brands, D. Chaum. Distance-Bounding Protocols (Extended Abstract). In *Advances in Cryptology EUROCRYPT'93*, Lofthus, Norway, Lecture Notes in Computer Science 765, pp. 344–359, Springer-Verlag, 1994.
12. L. Bussard, W. Bagga. Distance-Bounding Proof of Knowledge to Avoid Real-Time Attacks. In *IFIP TC11 International Conference on Information Security SEC'05*, Chiba, Japan, pp. 223–238, Springer, 2005.

13. H. Chernoff. A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the sum of Observations. *Annals of Mathematical Statistics*, vol. 23 (4), pp. 493-507, 1952.
14. C.J. F. Cremers, K.B. Rasmussen, B. Schmidt, S. Čapkun. Distance Hijacking Attacks on Distance Bounding Protocols. In *IEEE Symposium on Security and Privacy S&P'12*, San Francisco, California, USA, pp. 113–127, IEEE Computer Society, 2012.
15. Y. Desmedt. Major Security Problems with the "Unforgeable" (Feige-)Fiat-Shamir Proofs of Identity and How to Overcome Them. In *Congress on Computer and Communication Security and Protection Securicom'88*, Paris, France, pp. 147–159, SEDEP Paris France, 1988.
16. U. Dürholz, M. Fischlin, M. Kasper, C. Onete. A Formal Approach to Distance-Bounding RFID Protocols. In *Information Security ISC'11*, Xi'an, China, Lecture Notes in Computer Science 7001, pp. 47–62, Springer-Verlag, 2011.
17. A. Fiat, A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *Advances in Cryptology CRYPTO'86*, Santa Barbara, California, U.S.A., Lecture Notes in Computer Science 263, pp. 186–194, Springer-Verlag, 1987.
18. M. Fischlin, C. Onete. Terrorism in Distance Bounding: Modelling Terrorist-Fraud Resistance. In *Applied Cryptography and Network Security ACNS'13*, Banff AB, Canada, Lecture Notes in Computer Science 7954, pp. 414–431, Springer-Verlag, 2013.
19. S. Gambs, C. Onete, J.-M. Robert. Prover Anonymous and Deniable Distance-Bounding Authentication. In *ACM Symposium on Information, Computer and Communications Security (ASIACCS'14)*, Kyoto, Japan, pp. 501–506, ACM Press, 2014.
20. S. Goldwasser, S. Micali. Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information. In *Proceedings of the 14th ACM Symposium on Theory of Computing*, San Fransisco, California, U.S.A., pp. 365–377, ACM Press, 1982.
21. S. Goldwasser, S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, vol. 28, pp. 270–299, 1984.
22. J. Hermans, R. Peeters, C. Onete. Efficient, Secure, Private Distance Bounding without Key Updates. In *ACM Conference on Security and Privacy in Wireless and Mobile Networks WISEC'13*, Budapest, Hungary, pp. 195–206, ACM, 2013.
23. W. Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, vol. 58, pp. 13–30, 1963.
24. H. Kılınç, S. Vaudenay. Optimal Proximity Proof Revisited. To appear in ACNS'15.
25. S. Vaudenay. On Modeling Terrorist Frauds. In *Provable Security ProvSec'13*, Melaka, Malaysia, Lecture Notes in Computer Science 8209, pp. 1–20, Springer-Verlag, 2013.
26. S. Vaudenay. Private and Secure Public-Key Distance Bounding: Application to NFC Payment. In *Financial Cryptography and Data Security (FC'15)*, San Juan, Puerto Rico, Lecture Notes in Computer Science 8975, pp. 207–216, Springer-Verlag, 2015.
27. S. Vaudenay. On Privacy for RFID. In these proceedings.