Directional Distance-Bounding Identification Protocols

Ahmad Ahmadi Reihaneh Safavi-Naini

Abstract

Distance bounding (DB) protocols allow a prover to convince a verifier that they are within a distance bound. A public key distance bounding relies on the public key of the users to prove their identity and proximity claim. There has been a number of approaches in the literature to formalize security of public key distance bounding protocols. In this paper we extend an earlier work that formalizes security of public key DB protocols using an approach that is inspired by the security definition of identification protocols, and is referred to it as distance-bounding identification (`DBID`). We first show that if protocol participants have access to a directional antenna, many existing protocols that have been proven secure, will become insecure, and then show to revise the previous model to include this new capability of the users. `DBID` approach provides a natural way of modeling man-in-the-middle attack in line with identification protocols, as well as other attacks that are commonly considered in distance bounding protocols. We propose a new `DBID` scheme, called `Poxy`, with security proof. We compare the existing public key DB models, and prove the security of the scheme known as `ProProx`, in our model.

1 Introduction

Distance upper bounding (DB) protocols were first proposed in [12] to provide security against Man-in-the-middle (MiM) attack in authentication protocols. They have found wide diversity of applications in location and proximity based services [6, 25, 15, 20, 10]. Most DB protocols are symmetric key protocols where the prover and the verifier share a secret key. More recently public key DB protocols have been proposed where the prover is only known through their public keys, while their secret key remains private to them [3, 28, 2]. In these models, the verifier only has access to system public parameters as well as public keys of the participants.

In a DB setting there are three types of *participants*: *provers* who are registered in the system and have secret keys, a *verifier* who is honest and has access to correct public keys of provers, and *actors* who are not registered in the system, but want to be accepted and may collude with a dishonest prover. The distance between the prover and the verifier is measured by using a “fast challenge-response phase" during which a sequence of

one bit *challenges* are sent by the verifier to the prover, and the corresponding responses by the prover is recorded and used for distance estimation. A *challenge-response table* includes responses that are required for all possible challenges and is calculated by the prover before the fast challenge-response rounds start. The challenge-response table is constructed using the provers' secret key, and some nonces that are communicated during the slow phase of the protocol.

In symmetric key setting, the challenge-response table can also be constructed by the verifier and used for the verification of responses In public key setting however, the verifier only knows the prover's public key, and cannot calculate the challenge-response table. In this case, the verifier verifies the correctness of the prover's responses using their relation with the provers' public key.

For a DB protocol with distance bound $\mathcal{D}$, we refer to participants whose distance to the verifier are less than $\mathcal{D}$ as *close-by* participants (set $\mathcal{S}$) and those who are farther away than $\mathcal{D}$, as *far-away* participants (set $\mathcal{F}$).

Important attacks against DB protocols are:

(A1) *Distance-Fraud [7];* where a dishonest far-away prover tries to be accepted in the protocol. *Distance-Hijacking [10]* is a special case of this attack, where a far-away prover takes advantage of the communication of honest close-by provers to succeed in the protocol.

(A2) *Mafia-Fraud (MF) [12];* a close-by actor tries to use the communications of a far-away honest prover, to succeed in the protocol.

(A3) *Strong-Impersonation [2];* a close-by actor learns from past executions of the protocol by a close-by honest prover and tries to impersonate the prover in a new execution when the prover is either inactive or is not close-by anymore.

(A4) *Terrorist-Fraud (TF) [12];* a dishonest far-away prover colludes with a close-by actor to succeed in the protocol. In the *original TF*, it's assumed that the prover does not leak their secret key to the actor. In the *recent TF* [27] this restriction is removed, but it is required that non-negligible success of TF attack results in non-negligible improvement in future impersonation attacks by the actor.

To prove security of the existing public key DB protocols, such as [3, 7, 8, 28, 17, 2, 16], `PoPoK` [28] proposed a formal security model that uses a cryptographic PoK system and considers distance bound as an additional property of the system. In `DBID` [2] an alternative approach was proposed that follows the security formalization of `identification` protocols (using Σ-protocols), and includes distance-bound as an extra property. The `ProProx` scheme [28] was first proven secure in the former model [28], and later in the latter model [2].

`ProProx` uses polynomial times more fast phase operations, compared to normal DB protocols. The communications of the fast phase of DB protocols are generally more expensive, less reliable and more noise sensitive compared to the slow phase, as the data is sent in plain form. This fact makes the `ProProx` protocol to be an inefficient

scheme. DBPoK-log$^+$ [3] is another public key DB protocol that uses a different cryptosystem and uses the normal amount of fast phase communications, which makes it more efficient compared to ProProx. However, the security proof of this protocol has not been yet provided and it is not reliable in presence of noisy channel.

Our work: We consider provers that have access to directional antennas. Such antennas allow point to point communication with minimum interception by eavesdroppers who are outside the main transmission direction [1]. Advances in beamforming techniques and smart antennas in recent years [1] has made these antennas readily accessible to users. Distance bounding protocols, during the fast challenge-response phase, rely on physical layer communication and so it is important to consider this extra attacking capability for protocol participants. We will show that indeed directional antenna affects the security evaluation of DB protocols, and in particular effectively allows a malicious prover to launch a successful TF attack against protocols that had provable security against this attack. In Section 3 we show how this extra capability can be used by a malicious prover who is aided by a helper to break security of VSSDB [16] and DBPoK-log$^+$ [3] schemes. Directional antennas had been previously considered for actors during MF attack. In this paper we consider a dishonest prover with access to this type of antenna. For distance fraud, a directional antenna does not appear to affect security. In TF however, the dishonest prover is aided by a helper and directional antenna and this affects the security definition.

We extend the DBID formal security model [2] to include this new attacker's capability. The directional TF attack is captured in the revised TF-resistance (Property 4.4). We propose a new DBID scheme, called Poxy, and provide the security proof. We also prove that the existing ProProx scheme is indeed secure in this new model.

This paper is the full version of conference paper [4].

Organization. Section 2 is preliminaries. Section 3 shows directional TF attack on some public key DB protocols. Section 4 presents our model, Section 5 and Section 6 describe the construction of Poxy and ProProx, respectively, and give security theorems and proofs. Section 7 gives a summary of related works, and Section 8 concludes the paper.

2 Preliminaries

In this section we introduce a primitive that will later be used in our model.

A Σ-protocol is a 3-round cryptographic protocol between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$, in which the two parties interact, and at the end of the protocol, $\mathcal{V}$ is convinced about validity of $\mathcal{P}$'s statement. $\mathcal{P}$ has a private input x that satisfies the relation $\mathcal{R}(x,y)$, where y is a public value that is also known to $\mathcal{V}$. A Σ-protocol is used in cryptographic systems such as proof-of-knowledge schemes [11, 18, 23, 19, 26]. The Σ-protocol [23] is defined as follows;

Definition 2.1. **(Σ-protocol).** *A prover P and verifier V execute three' algorithms* $(\texttt{Commit}, \texttt{Response}, \texttt{Check})$ *using inputs* (x, y) *and* (y)*, respectively in the following order. x is private and y is public.*

Let $\mathbb{C}$*,* $\mathbb{H}$ *and* $\mathbb{R}$ *denote three sets defined as follows.* $\mathbb{C}$ *is the set of possible input that is chosen by the prover;* $\mathbb{H}$ *is the set of possible challenges chosen by the verifier; and* $\mathbb{R}$ *is the set of possible responses of the prover. The steps of the protocol are as follows:*

1. *P randomly chooses* $a \in \mathbb{C}$*, computes the commitment* $A = \texttt{Commit}(a)$*, and sends A to V.*
2. *Challenge and Response messages that are defined as follows:*
   (a) *V randomly chooses a challenge* $c \in \mathbb{H}$ *and sends it to P,*
   (b) *P computes* $r = \texttt{Response}(x, a, c) \in \mathbb{R}$ *and sends it to V,*
3. *V calculates* $ret = \texttt{Check}(y, c, r, A)$*, where* $ret \in \{accept, reject\}$.

At the end of the protocol, V outputs $Out_V = 1$ *if* $ret = accept$*, and* $Out_V = 0$ *otherwise.*

In an *identification scheme* (ID), a prover $\mathcal{P}$ convinces a verifier $\mathcal{V}$ that they know a witness x related to a public value y. The scheme is given by the tuple $\texttt{ID=(KeyGen;Commit;Response;Check)}$. KeyGen is a PPT algorithm that generates (x, y). The PPT algorithms Commit, Response and Check specifying an interactive protocol between the prover $\mathcal{P}$ and the verifier $\mathcal{V}$ as a Σ-protocol (Definition 2.1).

An identification scheme is correct if the Check function outputs *accept* if $\mathcal{R}(x, y)$ holds, and *reject* otherwise. An identification scheme is secure if an adversary with access to a set of valid transcripts $\mathcal{T} = \{(A, c, r)\}$, cannot generate a valid transcript (A', c', r') for a c' that has not appeared in $\mathcal{T}$. Note that a transcript (A, c, r) is valid according to public key y, if the function $\texttt{Check}(y, c, r, A)$ returns *accept*.

Here we define a more general form of Σ-protocols, called Σ^*-protocols, in which the verifier consecutively sends multiple challenges, each (except the first one) after receiving the response to the previous challenges.

Definition 2.2. **(Σ^*-protocol).** *A prover P and verifier V run the following*

Let $\mathbb{C}$*,* $\mathbb{H}$ *and* $\mathbb{R}$ *denote three sets defined as follows.* $\mathbb{C}$ *is the set of possible input that is chosen by the prover;* $\mathbb{H}$ *is the set of possible challenges chosen by the verifier; and* $\mathbb{R}$ *is the set of possible responses of the prover. The steps of the protocol are as follows:*

1. *P randomly chooses* $a \in \mathbb{C}$*, computes the commitment* $A = \texttt{Commit}(a)$*, and sends A to V.*
2. *Challenge and Response messages that are defined as follows:*
   (a) *V randomly chooses a challenge* $c \in \mathbb{H}$ *and sends it to P,*

(b) P computes $r = \texttt{Response}(x,a,c,\bar{c}) \in \mathbb{R}$, where $\bar{c}$ is the list of previous challenges before c, and sends it to V,

Steps 2-(a) and 2-(b) may be repeated a number of times.

3. *V calculates $ret = \texttt{Check}(y,[c],[r],A)$, where $ret \in \{accept, reject\}$ and $[c]$ and $[r]$ are lists of all challenges and responses, respectively.*

At the end of the protocol, V outputs $Out_V = 1$ if $ret = accept$, and $Out_V = 0$ otherwise.

3 Directional Attacks on Public-Key DB Protocols

Directional attacks assume that participants have access to directional antennas that allow them to direct messages to specific participants, and prevent other participants from receiving them. Figure 1 shows how such an antenna can be exploited by a malicious prover in a TF attack. The helper does not receive slow phase messages that are sent by the prover, as prover uses a directional antenna (orange ribbon in Figure 1) for communication in this phase. Before the start of the fast-phase, the prover sends all fast-phase responses (*e.g.*, the fast challenge-response table) to the helper, making the helper in-charge of responding to the fast-phase challenges.

This means that the adversary is able to separate the slow phase messages of the protocol from the fast-phase messages. In a vulnerable protocol, the prover may succeed in TF attack without leaking their long term key to the helper, using this separation technique. Therefore, the attacker's success in TF will not imply success in future impersonation.

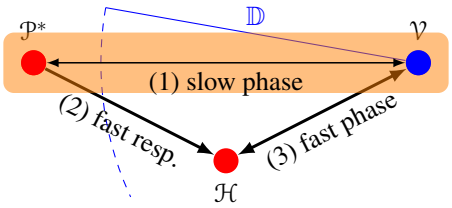

Figure 1: Directional TF

In the following we describe how this setting helps a malicious prover to succeed in terrorist-fraud against VSSDB [16] and DBPoK-log$^+$ [3] schemes.

3.1 Attack against VSSDB [16]

Using Definition 4.1 for a DB scheme, Figure 2 presents the Π protocol (Definition 4.1) of VSSDB scheme. This is a protocol between the prover and the verifier where the

prover has access to the public key of the verifier and their own secret key, and the verifier has access to their private key and the public key of the prover.

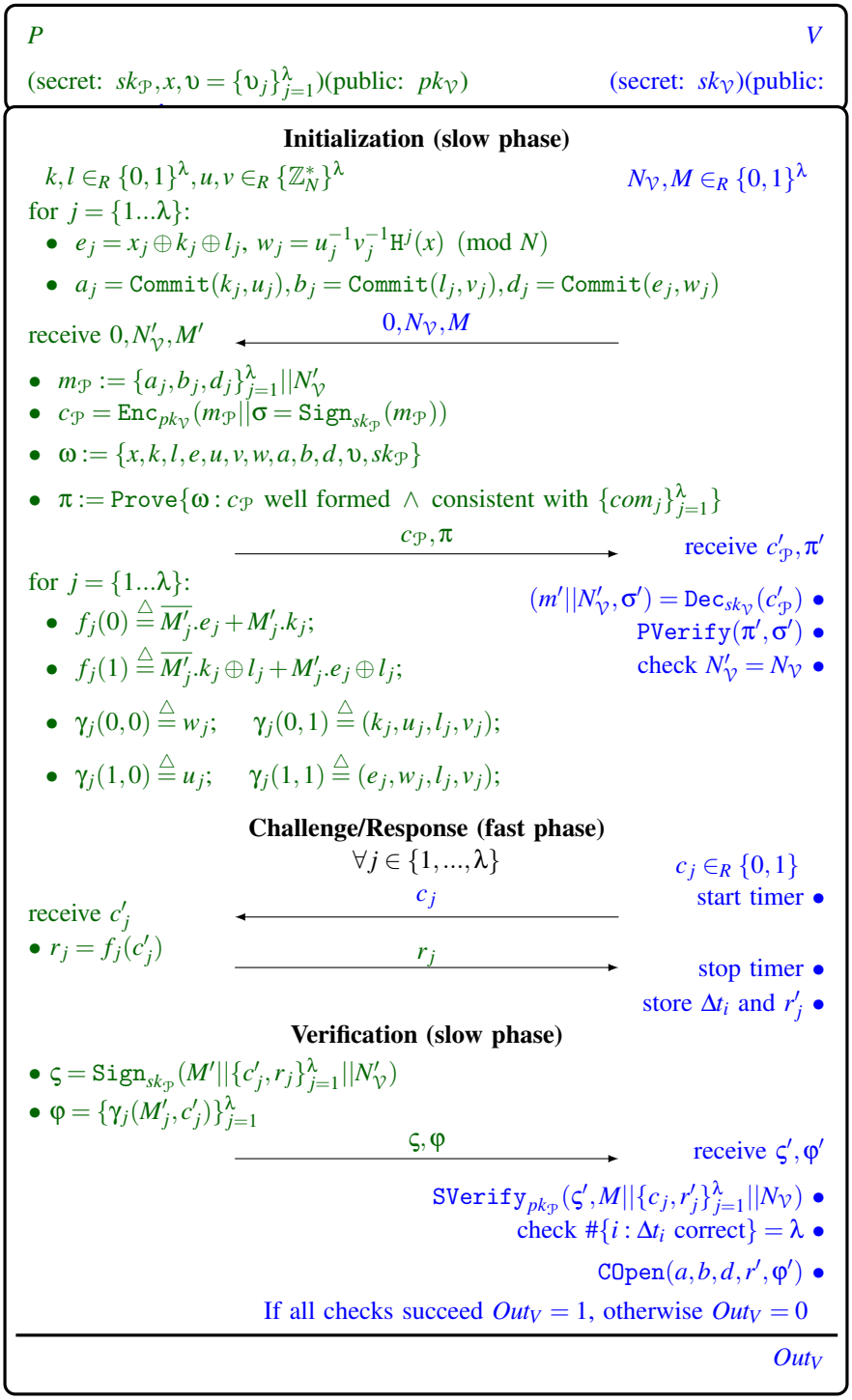

Figure 2: Π protocol of VSSDB scheme. (Commit, COpen) is a commitment scheme. (Enc, Dec) is a secure public key encryption scheme. (Sign, SVerify) is a signature scheme. (Prove, PVerify) is a proof-of-knowledge scheme. H is a secure hash function with pseudo-random output. x is the private key of the prover, with random distribution. $\upsilon = \{\upsilon_j\}_{j=1}^{\lambda}$ where $\upsilon_j = \mathrm{H}^j(x)$. $com_j = \mathtt{Commit}(x_j, \upsilon_j)$. The fast challenge-response table has λ columns with the j^{th} column defined by the boolean function $f_j()$.

Lemma 1. *In the* Π *protocol of* VSSDB *scheme, the fast challenge-response table does not leak information about the secret value* $sk_{\mathcal{P}}$ *of prover, assuming that* $sk_{\mathcal{P}}$ *and x are independently chosen.*

Proof. The elements of the fast challenge-response table are calculated as $r_j = f_j(c'_j) \in \{0,1\}$ for $j = \{1,...,\lambda\}$. Therefore, by knowing the table, one can, at the most, extract the values of e_j, k_j, l_j for $j = \{1,...,\lambda\}$. By finding these values, one can extract the value of x using the equation $x_j = e_j \oplus k_j \oplus l_j$ for $j = \{1,...,\lambda\}$. Since k_j and l_j are chosen randomly, therefore, this table only contains information about randomly chosen values k and l, and the value of x, which are independent of the secret value $sk_{\mathcal{P}}$. □

Attack: In this attack, the prover sends the messages of the slow phase to the verifier using directional antenna. The prover then sends the fast challenge-response table (*i.e.*, $\forall j \in \{1,...,\lambda\}$: either (e_j, k_j) or $(k_j \oplus l_j, e_j \oplus l_j)$) to the helper before running the fast phase. Note that the fast challenge-response table does not leak the prover long-term secret $sk_{\mathcal{P}}$ according to Lemma 1.

This allows the helper to respond to the verifier's challenges during the fast phase. The collusion of the prover and the helper will make the verifier to accept (*i.e.*, $Out_V = 1$) and this is without the prover sending to the helper any information that is dependent on the secret key $sk_{\mathcal{P}}$. The secret $sk_{\mathcal{P}}$ is required to generate a valid signature σ in the message π. This means that the helper's success chance in a future impersonation attack will not improve. This completes a successful TF.

3.2 Attack against DBPoK-log$^+$ [3]

The presented model of [3] follows the original definition of TF (Attack 4), and so our attack can be seen as outside their model. However, we showed that the original definition of TF is not suitable for anonymous distance-bounding protocols. In this section we present a TF attack against DBPoK-log$^+$, using the more recent definition of TF (Attack 4).

The Π protocol in DBPoK-log$^+$ scheme consists of the following four sub-protocols between the verifier (V) and the prover (P). The prover takes secret-key (sk_i, r) as input, and the verifier takes prover's public-key $pk_i = g_1^{sk_i}.g_2^r$ as input. The following is an improved version of the scheme presented in [3], in terms of being noise resistant.

Step (i) Bit Commitment is a commitment protocol, in which the prover uses the secret key sk_i as input. In this protocol, the prover decides on the "fast challenge-response table" and commits to each bit in the table. The verifier learns the committed values of every single bit of the fast challenge-response table. This table consists of two rows: $\{r_b[l]\}_{l=\{1,...,\lambda\},b\in\{0,1\}}$, where $r_b[l]$ is the response in the i^{th} fast challenge-response round. The corresponding

committed values are $\{C_b[l]\}_{l=\{1,...,\lambda\},b=\{0,1\}}$, and the corresponding randomness of commitments are indicated by $\{v_b[l]\}_{l=\{1,...,\lambda\},b=\{0,1\}}$, where $v_b[l] \in \mathbb{Z}_p^*$. The commitment function is as follows: $C_b[l] = g_1^{r_b[l]}.h^{v_b[l]}$ for $b \in \{0,1\}$, $l = \{1...\lambda\}$, and $g_1, h \in \mathbb{Z}_p$. The committed table and the randomness is kept secret at the prover, while the commitments are sent to the verifier. Figure 3 shows the details of this step. The parts that are shown in a box, are sub-protocols whose details are omitted.

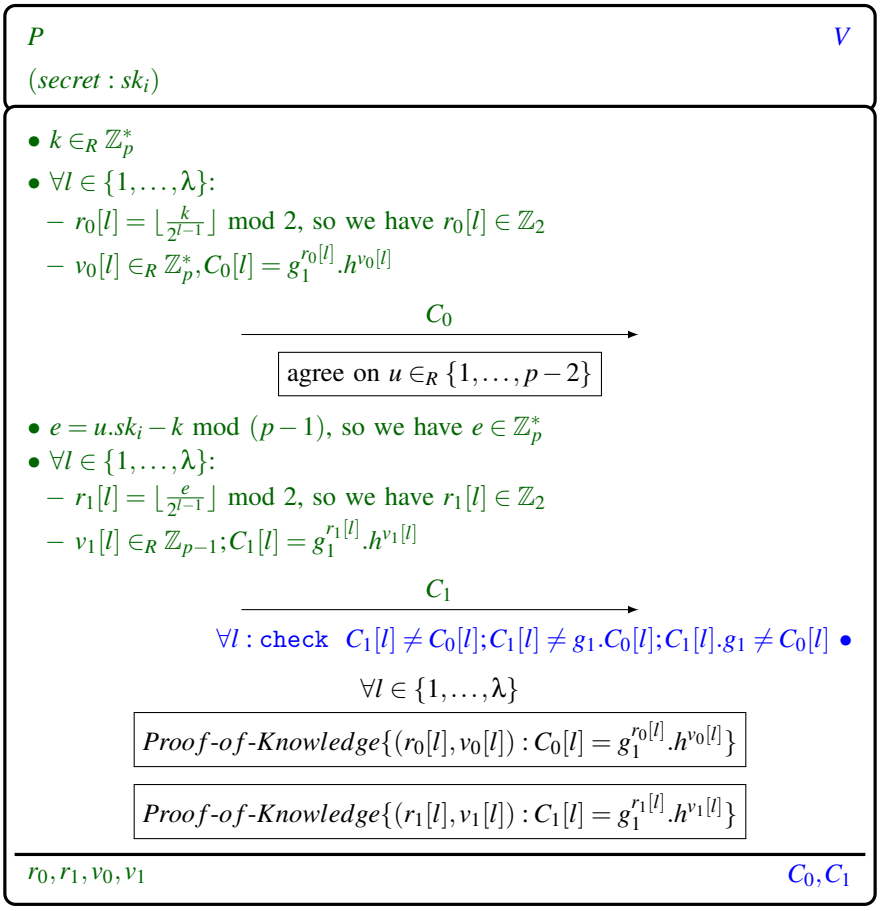

Figure 3: DBPoK-log$^+$ Step (i): Bit Commitment

Step (ii) Fast Challenge/Response is the protocol in which the prover uses the calculated "fast challenge-response table" $\{r_b[l] : l = \{1...\lambda\}, b = \{0,1\}\}$, generated in Bit Commitment step, as input. They run the protocol in Figure 4.

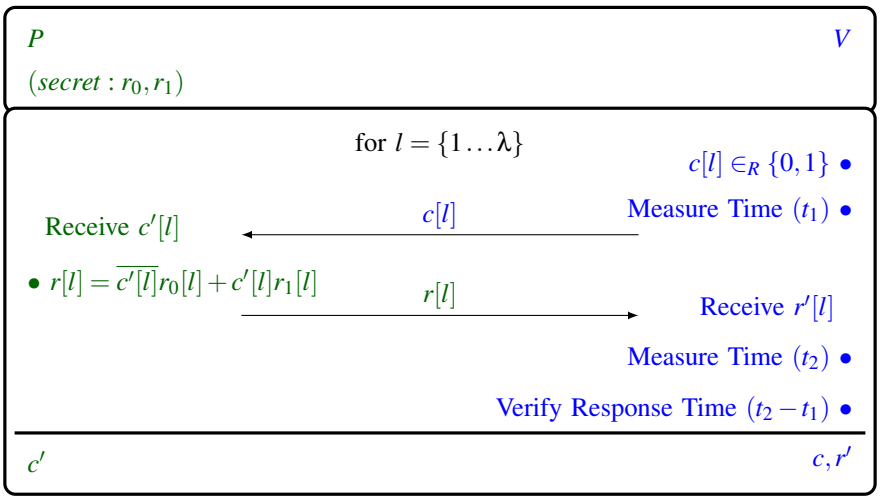

Figure 4: DBPoK-log^{+} Step (ii): Fast Challenge/Response

Step (iii) Commitment Opening is used to open half of the commitments, that correspond to the challenge bits sent by the verifier in Fast Challenge/Response step. In this step, the prover uses the secret commitment randomness (*i.e.*, $\{v_b[l] : l = \{1...\lambda\}, b = \{0,1\}\}$) and the challenge values of Fast Challenge/Response step (*i.e.*, c'), and the verifier uses the committed values (*i.e.*, $\{C_b[l] : l = \{1...\lambda\}, b = \{0,1\}\}$) and the challenge and response values of step (iii) (*i.e.*, c and r') as input. This protocol is shown in Figure 5, which improves the original PDB protocol [3] by adding noise resistance to the protocol. This step succeeds, if the noise counter is less than the threshold (*i.e.*, $count_{noise} < \tau$).

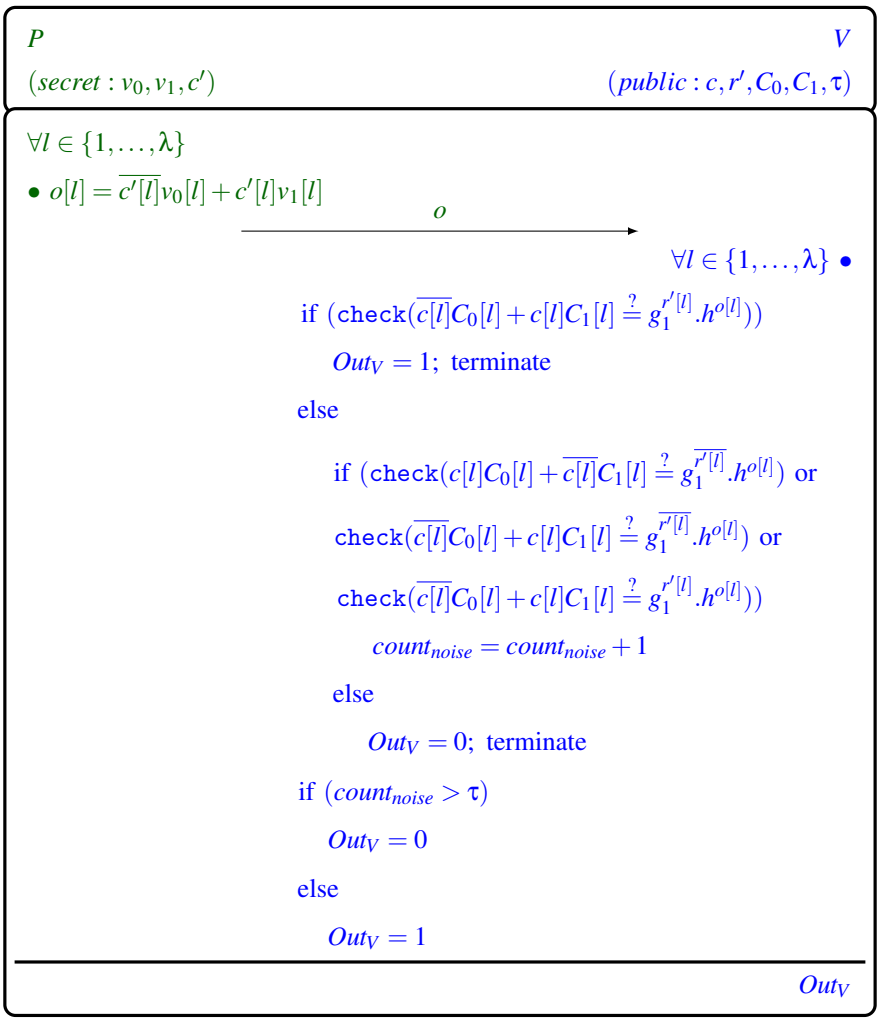

Figure 5: DBPoK-log$^+$ Step (iii): Commitment Opening

Step (iv) Proof-of-Knowledge is a protocol for zero-knowledge proof of equality that shows the secret key and the bitwise committed secret key of Bit Commitment step are the same. In this protocol, the prover uses the secret key (sk_i, r) and the commitment randomness of Bit Commitment step, and the verifier uses the public-key and bit committed values of Bit Commitment step, as input. z is the accumulation of the committed values of Bit Commitment step as $z = \prod_{l=1}^{\lambda}(C_0[l]C_1[l])^{2^{l-1}} \bmod p$, and v is the accumulation of the commitment randomness of step (ii) as $v = \sum_{l=1}^{\lambda} 2^l.(v_0[l] + v_1[l]) \bmod (p-1)$.

This protocol runs t iterations of ZK proof where z and C satisfy the following relation: $PoK[(sk_i, v, r) : z = g^{u.sk_i}.h^v \wedge pk_i = g^{sk_i}.g_2^r]$.

If all steps terminate successfully, then the verifier outputs $Out_V = 1$.

Lemma 2. *In* DBPoK-log$^+$ *scheme, the fast challenge-response table does not leak information about the randomness of secret-key of the prover (r), unless negligible probability.*

Proof. We know that by having the fast challenge-response table, we can calculate part of the secret-key of the prover, as $sk_i = \frac{k+e}{u} \bmod (p-1)$ where k is fresh randomness. Note that the fast challenge-response table is the output of random function that takes sk_i as input. So it cannot leak any information about other independent secrets of the prover, including r.

If there is an adversary $\mathcal{A}$ that can calculate the randomness r from the secret-key sk_i and $pk_i = g^{sk_i}.g_2^r$, then $\mathcal{A}$ can solve the discrete log problem for g_2^r. Therefore, since we assume discrete log is a hard problem, then the success chance of $\mathcal{A}$ is negligible. □

Attack: In directional TF attack (Figure 1), a malicious far-away prover will use a directional antenna for the slow phase of DBPoK-log$^+$ protocol (all steps except step (ii)) to communicate directly with the verifier, without the helper being able to intercept the messages. The prover sends the fast challenge-response table to the helper before running step (ii). Note that the fast challenge-response table does not leak any information about r, according to Lemma 2. In this way, the helper can respond in time and correctly to the challenges of the verifier during the fast challenge-response rounds. This attack makes the verifier to accept the protocol.

Since the fast challenge-response table does not leak any information about the randomness r, the helper will not be able to pass step (iv) in future and so it cannot impersonate the prover. This completes a successful terrorist fraud on DBPoK-log$^+$.

4 Model

First we define the settings of our system. This includes entities, their communication, their views, and the adversarial capabilities. Then we define distance-bounding identification scheme (DBID) and describe DBID experiment, which simulates an instance of DBID scheme. Finally we formalize four properties: (*Completeness*, *Soundness*, *DF-resistance*, and *TF-resistance* of distance-bounding identification schemes, using a game-based approach and described as a DBID experiment where adversary is active, and the game is between a challenger that sets up the system, taking into account the adversary's input. Finally in fundamental lemma, we discuss the relation between the location of participants, the timing of messages and their content.

Entities. We consider a set $\mathcal{U}$ of users. The user $u \in \mathcal{U}$ can have multiple provers that are denoted by the set $\mathcal{P}$. This captures the scenario that a single user has multiple devices.

A trusted group manger generates the public parameters of the system, and registers users and issues a key pair to each user. The user u is identifiable by its' private key. The private key, that must be kept secret, forms the secret input of the user in providing authentication proof. The private key of a user u is shared by all their provers $\mathcal{P}$. The corresponding public key of the user is published by the group manager.

There is a single verifier in the system, that for uniformity of notations, we refer to it as a set $\mathcal{V}$ that has a single member. The verifier only access to the public parameters of the system.

There is a set of *actors* ($\mathcal{T}$) that only have access to the public parameters of the system. In this paper we refer to the members of the sets $\mathcal{P}$, $\mathcal{V}$ and $\mathcal{T}$ as *participants.*

Each participant has a location $loc = (x,y) \in \mathbb{R} \times \mathbb{R}$, that is an element of a metric space equipped with Euclidean distance, and is fixed during the protocol. The distance function $d(loc_1, loc_2)$ returns the distance between two locations. Message travel time between locations loc_1 and loc_2 is $\frac{d(loc_1,loc_2)}{\mathcal{L}}$, where $\mathcal{L}$ is the speed of light. A bit sent over the channel may flip with probability p_{noise} ($0 \leq p_{noise} \leq 1$).

Participants that are located within a predefined distance bound $\mathcal{D}$ from the verifier, excluding the verifier, are called *close-by* participants (set $\mathcal{S}$), and those who are outside the distance bound from the verifier are called *far-away* participants (set $\mathcal{F}$).

Communication Structure. All participants have access to directional antennas: a participant A in loc_A can send a message to participant B at loc_B, such that others who are not on the straight line connecting loc_A and loc_B, cannot intercept it. Using omni-directional antenna however allows a message to be seen and modified by other participants. A participant may have multiple antennas that can be either directional or omni-directional. We allow a participant to send multiple messages to multiple parties at the same time, each from a separate antenna. Multiple messages that are received at the same time on the same antenna are combined and received as a single message.

View. The view of an entity at a point of a protocol consists of: all the inputs of the entity (including random coin tosses) and the set of messages that they have received up to that point in the protocol.Receiving a message is called an *event*. $View_x^{\Gamma}(e)$ is a random variable that denotes the view of an entity (or a set of entities) x right after the event e in protocol Γ. The short notation $View_x^{\Gamma}$ is used to indicate the view of x at the end of the protocol Γ, *i.e.*, $View_x^{\Gamma} = View_x^{\Gamma}(e_{last})$ where e_{last} is the last event in the protocol Γ.

Adversary. An adversary can corrupt a subset of participants $\mathcal{X}^* \subset \mathcal{P} \cup \mathcal{V} \cup \mathcal{T}$. As we will see later in this section, for each security property, $\mathcal{X}^*$ will have certain restrictions; in *Completeness* $\mathcal{X}^* = \emptyset$, in *Soundness* $\mathcal{X}^* \subseteq \mathcal{T}$, in *DF-resistance* $\mathcal{X}^* \subseteq \mathcal{P}$, and in *TF-resistance* $\mathcal{X}^* \subseteq \mathcal{P} \cup \mathcal{T}$.

When a prover of a user u is compromised, the user u's secret private key is compromised and the adversary can choose devices with that key at locations of their choice. In other words, all the provers in $\mathcal{P}$ become compromised. This is because all the provers of a user share the same private key. We refer to them as *corrupted provers*,

who are controlled by the adversary and may be activated simultaneously. However, we assume the non-corrupted provers follow the protocol, and a user only uses one of its devices at a time (*i.e.*, the execution time of the provers $\mathcal{P}$ do not overlap). This is because an honest user does not use multiple devices simultaneously.

Definition 4.1. (Distance-Bounding Identification Scheme). *For a security parameter λ, a distance-bounding identification scheme (*DBID*) is defined by a tuple $(\mathbb{X}, \mathbb{Y}, \mathbb{S}, \mathbb{P}, \mathcal{D}, p_{noise}, \texttt{Init}, \texttt{KeyGen}, \Pi, \texttt{Revoke})$, where*

(I) *$\mathbb{X}$ and $\mathbb{Y}$ are the sets of possible master keys and public keys of the system, respectively, chosen based on the security parameter λ. The system master key $msk \in \mathbb{X}$, and group public key $gpk \in \mathbb{Y}$ are generated using $(msk, gpk) \leftarrow \texttt{Init}(1^\lambda)$ algorithm;*

(II) *$\mathbb{S}$ and $\mathbb{P}$ are sets of possible private keys and public keys of the users respectively, chosen according to the security parameter λ. The user private key $sk \in \mathbb{S}$, and public key $pk \in \mathbb{Y}$ are generated using either $(sk, pk) \leftarrow \texttt{KeyGen}(1^\lambda, msk, gpk)$ algorithm or $\texttt{KeyGen}\{U(1^\lambda, gpk) \leftrightarrow GM(1^\lambda, msk)\}$ protocol;*
The KeyGen *algorithm is run by the group manager and the output is a user key pair and updated group public key. The user key pair is securely sent to the user, and the public key is published by the group manager, i.e., $gpk' := gpk \cup \{pk\}$. However, the* KeyGen *protocol is run between the group manager $GM(1^\lambda, msk)$ and a user $U(1^\lambda, gpk)$. The user outputs a key pair (sk, pk), and the group manager outputs the updates group public key $gpk' := gpk \cup \{pk\}$.*

(III) *Π is a Σ^*-protocol between a prover $P(sk, pk, gpk)$ and the verifier $V(pk, gpk)$, in which V verifies if the prover is authentic and is located within the distance bound $\mathcal{D} \in \mathbb{R}$ to the verifier.*

(IV) *The transmitted bits of a fast challenge-response round in Π protocol are affected by noise where $p_{noise} \in [0, 1]$ is the probability of a bit flip on each fast challenge-response message.*

(V) *$(gpk') \leftarrow \texttt{Revoke}(msk, gpk, i)$ is an algorithm that takes the master secret key, the group public key and the index of a user. The algorithm removes the corresponding user u_i from the system and updates the group public key accordingly, i.e., $gpk \rightarrow gpk'$. The* Revoke *operation is optional in* DBID *scheme.*

Below we describe execution of an instance of the DBID scheme, which we call DBID experiment.

Definition 4.2. (DBID Experiment). *A* DBID *experiment is defined by a tuple $(\texttt{DBID}; \mathcal{U}; \mathcal{P}; \mathcal{V}; \mathcal{T})$, where*

(i) DBID *is a distance-bounding identification scheme as defined in Definition 4.1.*

(ii) *$\mathcal{U}$ is the set of users that are members of the group; each user $u_j \in \mathcal{U}$ has three attributes:*

- *u_j.Key that is a secret key generated by the group manager,*

- u_j.*RT that is the registration time of the user that can be any time, and*
- u_j.*Rev that is a flag that shows if the user is revoked.*

(iii) $\mathcal{P}$ *is the set of provers; each prover has access to the secret key of a single user.*

(iv) $\mathcal{V}$ *is the set of verifiers; that have access to the public parameters of the* DBID *system. We consider the case where* $\mathcal{V}$ *has a single member.*

(v) $\mathcal{T}$ *is the set of actors; each actor has access to the public parameters of the* DBID *system.*

Members of the set $\mathcal{X} = \mathcal{P} \cup \mathcal{V} \cup \mathcal{T}$ *are called participants of the system. Each of the participants* $x \in \mathcal{X}$*, has the following attributes:*

a1. x.*Loc is the location of the participant,*

a2. x.*Code is the code to run by the participant,*

a3. x.*St that is the start time of the* x.*Code execution, and*

a4. x.*Corr is a flag indicating if the participant is corrupted or not.*

In addition to these attributes, each prover $p \in \mathcal{P}$ *has one extra attribute:*

a5. p.*Key that is the secret key of the corresponding user, i.e.,* $p.Key = u_j.Key$ *for user* $u_j \in \mathcal{U}$*.*

The start time of all provers is after registration time of the user, i.e., $\forall u \in \mathcal{U}, \forall p \in \mathcal{P} : p.St > u.RT$*.*

The provers of a user are either all honest or all dishonest. Because of users' keys are independently chosen, we can only consider a single user and so for simplicity we omit other users. i.e., $\forall p \in \mathcal{P} : p.Corr = flag$*, where* $flag \in \{true, false\}$*. Honest provers* $p \in \mathcal{P}$ *follow the* Π *protocol (i.e.,* $p.Code = \texttt{DBID}.\Pi.P(.)$*) and there is no overlap in the execution time of the honest provers. If the verifier is honest, then it follows the* Π *protocol (i.e.,* $v.Code = \texttt{DBID}.\Pi.V(.)$ *for* $v \in \mathcal{V}$*).*

The experiment is run by a simulator *that sets the attributes of the participants, and interacts with the group manager to assign keys to the provers of a user. If there is an adversary in the system, the simulator interacts with the adversary and follow their requested operations, that will influence the experiment.*

The experiment, without an adversary, proceeds as follows:

1. **Setup.**
   - *(a)* **Initiate:** *The group manager runs* $(\texttt{msk}/\texttt{gpk}) \leftarrow \texttt{DBID.Init}(1^\lambda)$ *algorithm to generate the master secret key and group public key.*
   - *(b)* **Generate Players:** *The simulator forms the sets* $(\mathcal{U}, \mathcal{V}, \mathcal{P}, \mathcal{T})$ *and sets their attributes. The simulator interacts with the group manager obtain and assign keys of the provers.*

2. **Run:** *The simulator starts the execution of x.Code for all participants* $x \in \mathfrak{X} = \mathcal{P} \cup \mathcal{V} \cup \mathcal{T}$ *at time x.St.*

The simulation uses a clock. $Time(e)$ indicates the time of event e. The start and finish time of a protocol Γ is indicated as $stTime(\Gamma)$ and $fshTime(\Gamma)$ respectively, which form the execution time $exTime(\Gamma) = (stTime(\Gamma), fshTime(\Gamma))$ as the range of time and the execution time period $exLen(\Gamma) = fshTime(\Gamma) - stTime(\Gamma)$. Different provers have different execution time period (*i.e.*, they participate in a protocol from time t_1 to t_2), and possibly different locations.

In the following, we define security properties of DBID scheme, using a game between a challenger and an adversary. This game is a DBID experiment that is run by the challenger who interacts with an adversary. In this game we only consider one user, *i.e.*, $|\mathcal{U}| = 1$. The challenger plays both roles of the simulator and the group manager in the DBID experiment (Definition 4.2). The adversary's capabilities is modelled as access to a query that it presents to the challenger.

Definition 4.3. (DBID **Game).** *A* DBID *game between a challenger and adversary is a* DBID *experiment that is defined by a tuple* $(\texttt{DBID}; \mathcal{U}; \mathcal{P}; \mathcal{V}; \mathcal{T}; \texttt{CorruptParties})$ *where*

- DBID *is a distance-bounding identification scheme as defined in Definition 4.1.*
- $\mathcal{U}, \mathcal{P}, \mathcal{V}, \mathcal{T}$ *are the sets of users, provers, verifiers and actors as defined in Definition 4.2, that are determined through interaction of the challenger and the adversary.*
- $\texttt{CorruptParties}(Q)$ *is a query that allows the adversary to plan (program) their attack.* Q *is a set of participants, that may exist in the system or be introduced by the adversary.*

The game setup phase is by the challenger while playing the roles of the simulator and the group manager, and interacting with the adversary.

In more details:

1. **Setup.**
   (a) **Initiate:** *Challenger runs* $(\texttt{msk}/\texttt{gpk}) \leftarrow \texttt{DBID.Init}(1^\lambda)$ *and publishes gpk. Note that the execution codes of an honest prover and verifier are known by the challenger and the adversary at this point, and are referred to as* $\texttt{DBID}.\Pi.P$ *and* $\texttt{DBID}.\Pi.V$*, respectively.*
   (b) **Generate Players:** *The sets* $(\mathcal{U}, \mathcal{V}, \mathcal{P}, \mathcal{T})$ *are formed through the interaction of the challenger and the adversary as follows:*
      i. *The challenger creates the sets* $(\mathcal{U}, \mathcal{V}, \mathcal{P}, \mathcal{T})$ *as follows:*
         - *Chooses a verifier* $\mathcal{V} = \{v\}$*, with the following attributes:*

           a1. $v.Loc = loc_0$*,*

           a2. $v.Code = \texttt{DBID}.\Pi.V$*,*

a3. $v.St = 0$*, and*

a4. $v.Corr = false$*.*

- *Runs* $(sk, pk) \leftarrow \texttt{DBID.KeyGen}(1^\lambda, msk, gpk)$ *once and forms the set* $\mathcal{U} = \{u\}$*. The user key is set as* $u.Key = sk$*, the registration time of the user is set as* $u.RT = 0$ *and the revocation flag is set as* $u.Rev = false$*. The group public key is updated as* $gpk' := gpk \cup \{pk\}$*.*
- *Creates a prover set* $\mathcal{P}$ *and for each member* p *of* $\mathcal{P}$*, assigns their attributes as:*

  a1. $p.Loc$ *is set arbitrarily,*

  a2. $p.Code = \texttt{DBID}.\Pi.P$*,*

  a3. $p.St$ *is set arbitrarily such that there is no overlap in the execution time of the provers (i.e.,* $\nexists p_1, p_2 \in \mathcal{P} : p_1.St < p_2.St \wedge p_1.St + exLen(\texttt{DBID}.\Pi) > p_2.St$*),*

  a4. $p.Corr = false$*, and*

  a5. secret key $p.Key = u.Key$*.*

- $\mathcal{T} = \emptyset$

ii. The challenger sends the attributes $(x.Loc, x.Code, x.St)$ *for all* $x \in \mathcal{P} \cup \mathcal{V} \cup \mathcal{T}$ *to the adversary. The size of the set* $\mathcal{X}$ *is n.*

iii. The adversary uses the published values to form their corruption query $\texttt{CorruptParties}(Q)$ *that is sent to the challenger. The secret information of the corrupted participants in* Q *is given to the adversary and the behaviour (Code) of the corresponding participants, is assigned according to the adversary instruction.*

More specifically, the parameter of this query is $Q = \{q_1, ..., q_{n'}\}$*. Each* q_i *consists of the location, the execution start time and the execution code of a participant. i.e.,* $q_i = (type, location, code, time)$*, where* $type \in \{verifier, prover, actor, user\}$ *indicates the type of the participant,* $location \in \mathbb{R} \times \mathbb{R}$*,* $code \in \{0,1\}^*$ *indicates the location of the participant and* $time \in \mathbb{N}$ *indicates the execution start time of the participant.*

If $q_i \in \mathcal{X} = \mathcal{P} \cup \mathcal{V} \cup \mathcal{T} \cup \mathcal{U}$*, it determines the settings of an existing participant, and if* $q_i \notin \mathcal{X}$*, it determines the settings of a new participant.*

iv. Upon receiving the $\texttt{CorruptParties}(Q)$ *where* $Q = \{q_1, ..., q_{n'}\}$*, the challenger runs:*

- *For a* q_i *that* $q_i.type = verifier$*, then* $v.Code = q_i.code$ *and* $v.Corr = true$ *for* $v \in \mathcal{V}$*.*

- *For each q_i that $q_i.type = user$, sets the users' revocation flag as $u.Rev = true$ where $u \in \mathcal{U}$, runs $(gpk') \leftarrow \texttt{Revoke}(msk, gpk, 1)$, and then updates the group public key $gpk \leftarrow gpk'$. This applies only if the* DBID *scheme provides user revocation.*
- *If there is a q_i that $q_i.type = prover$, then for each member p of the set $\mathcal{P}$, sets their corruption flag $p.Corr = true$. If q_i is not corresponding to an existing prover, then create a new prover p and add it to the prover set $\mathcal{P}$. Set the attributes of the participant p as follows:*

  a1. location $p.Loc = q_i.location$,

  a2. execution code $p.Code = q_i.code$,

  a3. start time $p.St = q_i.time$,

  a4. corruption flag $p.Corr = true$, and

  a5. secret key $p.Key = u.Key$.
- *For each q_i that $q_i.type = actor$, add a new actor x to the set $\mathcal{T}$, and assign its attributes as follows:*

  a1. location $x.Loc = q_i.location$,

  a2. execution code $x.Code = q_i.code$,

  a3. start time $x.St = q_i.time$, and

  a4. corruption flag $x.Corr = true$.

v. The challenger sends the key of the corrupted provers and the key of revoked user to the adversary, i.e., $p.Key$ for all $p \in \mathcal{P}$ such that $p.Corr = true$ and $u.Key$ for all $u \in \mathcal{U}$ such that $u.Rev = true$.

2. **Run:** *Challenger activates all participants $x \in \mathcal{X} = \mathcal{P} \cup \mathcal{V} \cup \mathcal{T}$ at time $x.St$ for execution of $x.Code$.*

The game ends when the last participant's code completes its execution.

Using the above game, we define four distinct properties for distance-bounding identification schemes. The winning condition of the above game, varies for each property.

Property 4.1. (DBID **Completeness).** *Consider a* DBID *scheme and a* DBID *game when $Q = \emptyset$ in the* CorruptParties(Q) *query and the set $\mathcal{P}$ is not empty.*

The DBID *scheme is (τ, δ)-complete for $0 \leq \tau, \delta \leq 1$, if the verifier returns $Out_V = 1$ with probability at least $1 - \delta$, under the following assumptions:*

- *the fast challenge-response rounds are independently affected by noise and at least τ portion of them are noiseless, and*
- *$\tau > 1 - p_{noise} - \varepsilon$ for some constant $\varepsilon > 0$.*

A complete scheme must have negligible δ to be able to function in the presence of communication noises.

Property 4.2. (DBID **Soundness).** *Consider a* DBID *scheme and a* DBID *game with the following restrictions:*

- $\mathcal{P}$ *is nonempty and* $\forall p \in \mathcal{P}, v \in \mathcal{V} : d(p.Loc, v.Loc) > \texttt{DBID}.\mathcal{D}$*, and*
- *in the* $\texttt{CorruptParties}(Q)$ *query,* $q_i.type \in \{actor, user\}$ *for all* $q_i \in Q$*.*

In this game the verifier and provers are honest, while the adversary $\mathcal{A}$ *corrupts a set of actors and sets their locations (and, if applicable) revokes some users. The corrupted actors are controlled by the adversary, and can simultaneously communicate with multiple provers and the verifier. They can receive a message m from a prover and send* m' *to the verifier, and vice versa. The certificate of the revoked users are sent to the adversary.*

The DBID *scheme is* γ*-sound if the probability of the verifier outputting* $Out_V = 1$ *is at most* γ*.*

This general definition captures the following attacks by considering special values for the parameters of the game;

- *relay attack* [7] where the MiM attacker only relays the messages between the honest verifier and a far-away honest prover. The MiM attacker tries to convince the verifier that the prover is located close to the verifier. This attack is achieved by adding extra restrictions on the adversary of Property 4.2 as follows:
  • $\forall q_i \in Q$ we have $q_i.code =$ "relay messages".
- *mafia-fraud* [12] is when there is an honest verifier, an honest far-away prover, and a close-by MiM attacker who tries to convince the verifier that the prover is located close to the verifier. The attacker listens to the legitimate communications for a while, before running the attack as the learning phase. This attack corresponds to adding extra restrictions on the adversary in Property 4.2 as follows:
  • $\mathcal{P}$ is nonempty, and
  • $\forall q_i \in Q$ we have $d(q_i.location, v.Loc) \leq \texttt{DBID}.\mathcal{D}$ for $v \in \mathcal{V}$.
- *impersonation attack* [5] happens when there is an honest verifier and a single close-by attacker who tries to convince the verifier that the prover is located close to the verifier. The attacker can have a learning phase before running the attack. We can achieve this attack by adding extra restrictions on the adversary of Property 4.2 as follows:
  • $\mathcal{P}$ is nonempty, and
  • $\forall q_i \in Q$ we have $d(q_i.location, v.Loc) \leq \texttt{DBID}.\mathcal{D}$ for $v \in \mathcal{V}$, and
  • among all the successful $\texttt{DBID}.\Pi$ protocols (Π^{succ} set) during the game, $\exists \pi \in \Pi^{succ}, \forall p \in \mathcal{P} : t = fshTime(\pi), t \notin [p.St, p.St + exLen(p.Code)]$.
- *strong-impersonation* [2] happens when either *mafia-fraud* or *impersonation* hap-

pens. We can achieve this attack by adding extra restrictions on the adversary of Property 4.2 as follows:
• $\mathcal{P}$ is nonempty, and
• $\forall q_i \in Q$ we have $d(q_i.location, v.Loc) \leq \texttt{DBID}.\mathcal{D}$ for $v \in \mathcal{V}$, and
• among all the successful $\texttt{DBID}.\Pi$ protocols (Π^{succ} set) during the game, at least one of the following conditions hold:

(i) $\exists \pi \in \Pi^{succ}, \forall p \in \mathcal{P} : t = fshTime(\pi), t \notin [p.St, p.St + exLen(p.Code)]$

(ii) $\exists p \in \mathcal{P}, \exists \pi \in \Pi^{succ}, v \in \mathcal{V} : t = fshTime(\pi), t \in [p.St, p.St + exLen(p.Code)] \wedge d(p.Loc, v.Loc) > \texttt{DBID}.\mathcal{D}$.

We consider two types of attacks by a dishonest prover: far-away dishonest provers (Property 4.3), and far-away dishonest provers with a close-by helper (Property 4.4).

Property 4.3. (DBID **Distance-Fraud).** *Consider a* DBID *scheme and a* DBID *game with the following restrictions:*

- $\mathcal{P}$ *is nonempty and* $\forall p \in \mathcal{P}, v \in \mathcal{V} : d(p.Loc, v.Loc) > \texttt{DBID}.\mathcal{D}$*, and*
- *in the* $\texttt{CorruptParties}(Q)$ *query,* $q_i.type = prover$ *and* $d(q_i.location, v.Loc) > \texttt{DBID}.\mathcal{D}$ *for all* $q_i \in Q$ *and* $v \in \mathcal{V}$.

The DBID *scheme is* α*-DF-resistant if, for any* $\texttt{DBID}.\Pi$ *protocol in such game, we have* $\Pr[Out_V = 1] \leq \alpha$.

In the following we define the TF-resistance of DBID protocols.

Property 4.4. (DBID **Terrorist-Fraud).** *Consider a* DBID *scheme and a* DBID *game with the following restrictions:*

- $\mathcal{P}$ *is nonempty and* $\forall p \in \mathcal{P}, v \in \mathcal{V} : d(p.Loc, v.Loc) > \texttt{DBID}.\mathcal{D}$*, and*
- *in the* $\texttt{CorruptParties}(Q)$ *query,* $q_i.type \in \{prover, actor\}$ *and* $d(q_i.location, v.Loc) > \texttt{DBID}.\mathcal{D}$ *for all* $q_i \in Q$ *that* $q_i.type = prover$ *and* $v \in \mathcal{V}$.

The DBID *scheme is* μ*-TF-resistant, if the following holds about the above game:*

- *If the verifier returns* $Out_V = 1$ *in the* Π *protocol of game* Γ *with non-negligible probability* κ*, then there is an impersonation attack as a* DBID *game* Γ' *with honest verifier, no prover and one close-by actor that takes the view of close-by participants* ($View_S^{\Gamma}$) *as input, and makes the verifier return* $Out_V = 1$ *with probability at least* $\kappa - \mu$ *in the* Π *protocol of* Γ' *game.*

Note that this is a formal definition of the terrorist-fraud resistance (A4) that is based on the recent definitions (such as [27]) and is different from the definition of TF in [12], which is the original version of this work. This change in the definition of TF is necessary because here we consider directional antennas. With this new capability, a malicious prover can use directional communication with the verifier and the helper, such that although the TF succeeds, the leaked information does not allow a response generator to be constructed. Using the original approach, and removing contribution of the verifier's view, allow us to define TF security.

In Lemma 3 we show that if a DBID scheme is TF-resistant (Property 4.4), using a directional antenna (as in Figure 1) will not affect its security. We only provide an informal proof because a formal proof needs formalizing properties of directional antennas.

Lemma 3. *If a* DBID *scheme is TF-resistant (Property 4.4), it is directional TF-resistant.*

Proof. The main observation is that in a TF attack (Property 4.4), all close-by participants, except the verifier, are controlled by the adversary. So, using a directional antenna to communicate with close-by participants such that the verifier is excluded, adds the transmitted message to the view of adversary, and replacing the directional antenna with an omni directional one, does not change this view.

The messages that are sent to the verifier using directional antenna, will not be included in the impersonation adversary view, *i.e.*, $View_{\mathcal{S}}^{\Gamma}$.

Using property 4.4, if there is a successful TF attack against a DBID scheme, the *TF-resistant* property guarantees existence of an impersonation attacker with non-negligible probability that takes the $View_{\mathcal{S}}^{\Gamma}$ as input. Since the view of actors in a directional TF attack will include this view, therefore, in a *TF-resistant* DBID scheme, having a successful directional TF attack implies future impersonation attack. □

In fundamental lemma, we relate (i) the local timing of a received message at the verifier, (ii) physical distances traveled by the message, and (iii) the message content. It shows that any response r for the challenge c, that is received by the verifier $\mathcal{V}$, can be split into two parts $r_{\mathcal{S}}$ and $r_{\mathcal{F}}$ based on the distance of the sender, and each part can be computed from two separate inputs; (a) $r_{\mathcal{S}}$ from the challenge c and the views of close-by participants before seeing c, and (b) $r_{\mathcal{F}}$ from the views of far-away participants before seeing c.

The difference between our fundamental lemma compared to the proposed fundamental lemma of [28] is twofolds: we allow a received message to be combination of multiple sent messages which is more realistic in wireless communications, and we do not require a global clock.

Lemma 4. (Fundamental Lemma). *Consider a multi-party protocol execution Γ with a distinguished participant $v \in \mathcal{V}$ that measures the local time of events, the set of far-away participants $\mathcal{F}$ and the set of close-by participants $\mathcal{S}$. At local time t, v broadcasts a random message c and waits for a response r.* Acc *denotes the event that r was received by v at local time $t' \leq t + \frac{2\mathcal{D}}{\mathcal{L}}$. The message from a participant x is independent from c, if it is the result of running the participant algorithm with $View_x^{\Gamma}(\bar{c})$ as input.*

If Acc *occurs, then r consists of two components $(r_{\mathcal{F}}, r_{\mathcal{S}})$ (i.e., $r = sum(r_{\mathcal{F}}, r_{\mathcal{S}})$ for a deterministic function summation $: \mathbb{M}^* \rightarrow \mathbb{M}$, where $\mathbb{M}$ is the set of all possible messages that v may receive), where $r_{\mathcal{F}}$ is sent from members of $\mathcal{F}$ and $r_{\mathcal{S}}$ is sent from members of $\mathcal{S}$ that: $r_{\mathcal{F}} = Msg_{\mathcal{F}\rightarrow v(t')}(\bar{c})$ and $\exists \mathcal{J}' : r_{\mathcal{S}} = \mathcal{J}'(View_{\mathcal{S}}^{\Gamma}(\bar{c}), c, Msg_{\mathcal{F}\rightarrow\mathcal{S}}(\bar{c}))$. $Msg_{\mathcal{F}\rightarrow\mathcal{S}}(\bar{c})$ is all messages from any member of $\mathcal{F}$ to any member of $\mathcal{S}$ that are independent of c, and $Msg_{\mathcal{F}\rightarrow v(t')}(\bar{c})$ is the summation of all messages from members*

of $\mathcal{F}$ that get received by the verifier at time t', which are independent of c. Note that $Msg_{\mathcal{F}\to v(t')}(\bar{c}) \in \mathbb{M}$.

Proof. We consider three cases for possible sources of the response r, received by v; (i) r is solely sent by a subset $F \subseteq \mathcal{F}$, (ii) r is solely sent by a subset $C \subseteq \mathcal{S}$, or (iii) otherwise. In this proof, we show that the response can be generated according to the lemma in all the cases.

Case (i) In the first case, let's consider the received message $r = r_{\mathcal{F}}$ by the verifier, as the summation[1] of multiple messages $(r_1,...,r_l)$ that are sent by participants $F = \{F_1,...,F_l\}$ at time $\{t'_1,...,t'_l\}$, respectively. All these messages arrive at the verifier at the same time, so the sending times are according the mutual distance between the sender and the verifier. Without lose of generality, we assume that for $i = \{1...l-1\}$, the participant F_i is closer to v than participant F_{i+1}, and so $t'_i \geq t'_{i+1}$.

If `Acc` occurs, we have $t'_1 \leq t + 2\mathcal{D} - \frac{d(v,F_1)}{\mathcal{L}}$, and since $d(v,F_1) > \mathcal{D}$, then we have $t'_1 < t + \frac{d(v,F_1)}{\mathcal{L}}$ which is before the time F_1 could see the challenge c. We have the same inequality for other participants, so $r = r_{\mathcal{F}} = Msg_{\mathcal{F}\to v(t')}(\bar{c}) = sum(r_1,...,r_l)$ is independent of c.

Case (ii) In the second case, let's consider the received message $r = r_{\mathcal{S}}$ by the verifier, as the summation of multiple messages $(r_1,...,r_l)$ that are sent by participants $C = \{C_1,...,C_l\}$ at time $\{t'_1,...,t'_l\}$, respectively. All these messages arrive at the verifier at the same time, so the sending times are according the mutual distance between the sender and the verifier. Without lose of generality, we assume that for $i = \{1...l-1\}$, the participant F_i is closer to v than participant F_{i+1}, and so $t'_i \geq t'_{i+1}$.

We make the algorithm $\mathcal{J}'$ simulate all close-by participants $x \in \mathcal{S}$ in the time range $[t + \frac{d(v,x)}{\mathcal{L}}, t + \frac{2\mathcal{D}-d(v,x)}{\mathcal{L}}]$ (*i.e.*, from the event of seeing c, until before it's too late to send a message to the verifier and make `Acc` occur). The simulation is in parallel and in chronological order. The output of $\mathcal{J}'$ is the message $r = r_{\mathcal{S}}$ delivered to v as the summation of messages sent from $C \subseteq \mathcal{S}$ at time $t' \leq t + \frac{2\mathcal{D}-d(v,C_1)}{\mathcal{L}}$, where C_1 is the closest responder.

Here we claim that the input (m) of each responder $x \in C$ is either part of the input of $\mathcal{J}'$ in the lemma. In order to prove it, we consider four cases about the source of m;

- m comes from the internal view of x before seeing c, *i.e.*, $View^{\Gamma}_x(\bar{c})$; since $x \in \mathcal{S}$, then $View^{\Gamma}_x(\bar{c}) \in View^{\Gamma}_{\mathcal{S}}(\bar{c})$ that is included in the input of the simulator.
- m comes from far-away participants; in this case it must be independent from c due to the distance constraints, and so $m \in Msg_{\mathcal{F}\to\mathcal{S}}(\bar{c})$, that is included in simulator's input.

[1]The summation of multiple messages that are received at the same time, depends on the physical properties of the communication channel. In this model, we just assume that it is a deterministic function.

- m comes from v; since v only sends c before r, then the message m can be either c, or already is in the view of the close-by participants before c, *i.e.*, $View_{\mathcal{S}}^{\Gamma}(\bar{c})$.
- m comes from a close-by participant $y \in \mathcal{S}$; then the above three cases about x, applies to y too. This part is a recursive argument till these is no close-by participant left to send the message.

Case (iii) In the third case, let's consider the received message r by the verifier, as the summation of messages sent from close-by participants ($r_{\mathcal{S}}$) and messages sent from far-away participants ($r_{\mathcal{F}}$). Note that all messages are delivered at the same time t' to v. We have thus showed that there are algorithms that can generate $r_{\mathcal{S}}$ and $r_{\mathcal{F}}$ separately. Therefore by applying the summation function $r = sum(r_{\mathcal{S}}, r_{\mathcal{F}})$, the algorithm can compute the response message r with correct timing. □

5 Poxy Scheme

In this section we present a new DBID construction as an extension of DBPK-log^{+} [3] and DBPK-log [8]. This protocol uses Pedersen [24] cryptosystem. As a DBID $(\mathbb{X};\mathbb{Y};\mathbb{S};\texttt{Init};\texttt{KeyGen};\Pi;\mathcal{D};p_{noise})$ protocol, Poxy consists of all of the operations in the scheme as follows;

$(msk, gpk) \leftarrow \texttt{Init}(1^{\lambda})$ The group manager initializes a Pedersen commitment with λ bit security: chooses a large λ-bit strong prime p, such that $p = 2q+1$ for a prime q. It also chooses the generator $g = p-1$ and a random element $h \in_R \mathbb{Z}_p^*$. Note that for the selected generator we have $g = g^{-1}$ in the multiplicative group $\mathbb{Z}_p$.

The group manager initiates a certificate mechanism for validating the public key of the user, *i.e.*, creates a certificate key pair (sk^{Cert}, pk^{Cert}). We omit this mechanism from the rest of this section for simplicity. So we have $msk = (sk^{Cert})$ and $gpk = (p, q, g, h, pk^{Cert}, \Xi)$ where $\Xi = \emptyset$.

$(sk, pk) \leftarrow \texttt{KeyGen}(msk, gpk)$ Assume $l-1$ users have joined the group and their public keys are in the set $\Xi = \{pk_1, ..., pk_{l-1}\}$ that is published by the group manager. For the l^{th} user, the group manager generates a key pair (sk, pk), such that $sk \in_R \mathbb{Z}_{p-1}$ and $pk = \texttt{Commit}(sk; 0) = g^{sk} \pmod{p}$, where $\texttt{Commit}(u; v)$ is Pedersen commitment ($= g^u h^v \pmod{p}$). The group manager securely sends the key pair to the new user and adds the public key pk to the set Ξ.

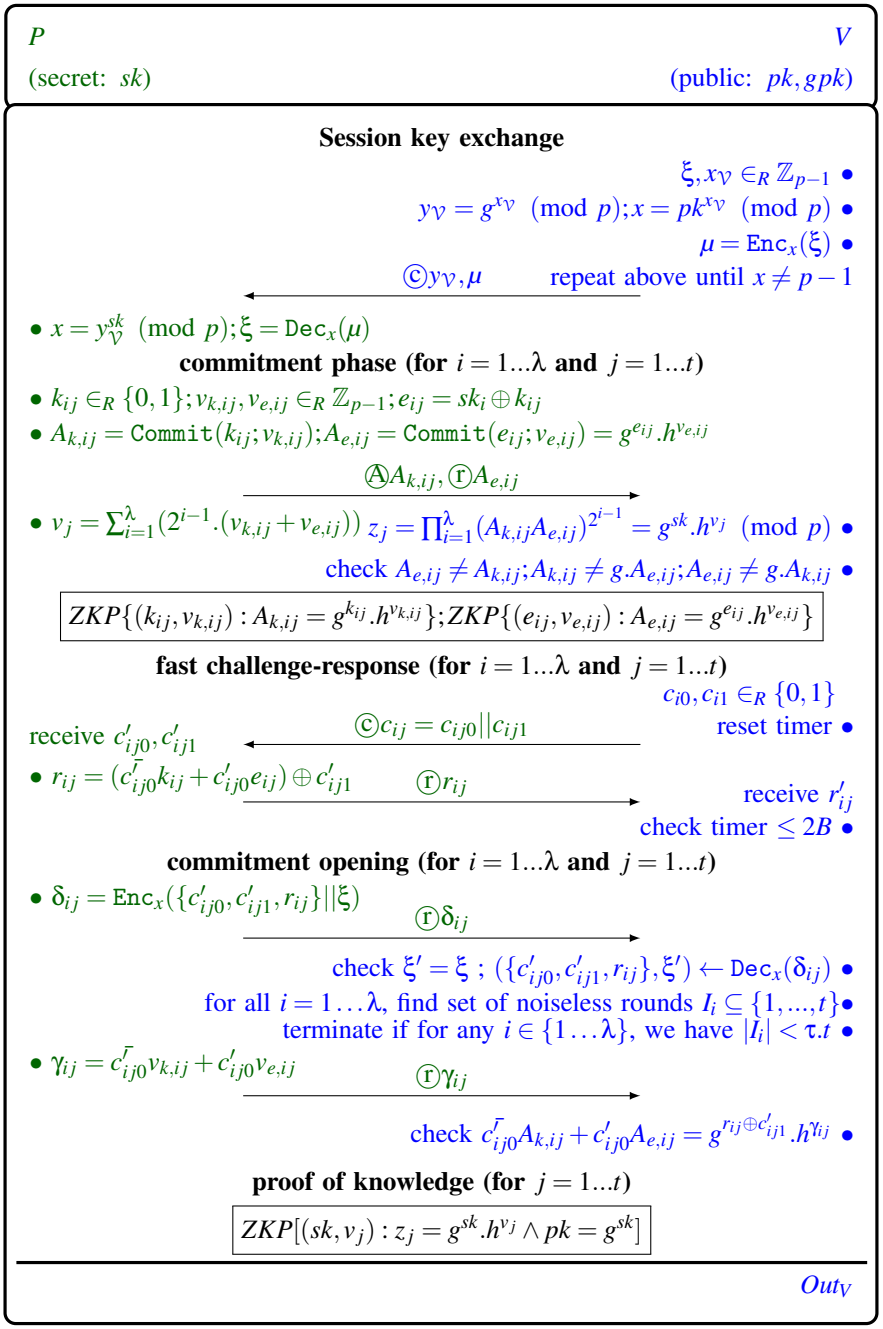

Figure 6: Π protocol of Poxy scheme. Commit is Pedersen commitment scheme. (Enc, Dec) is a secure symmetric encryption scheme. *ZKP* is a zero-knowledge proof-of-knowledge protocol. The notation Ⓐ by a message, indicates that the message is considered as commitment of Σ^*-protocol. The notations ⓒ and ⓡ indicate the challenge and response messages of Σ^*-protocol, respectively.

$accept/reject \leftarrow \Pi\{P(sk,pk) \leftrightarrow V(pk)\}$ When a prover of a registered user wants to run the DBID.Π protocol with the verifier, they will follow the protocol described in Figure6.

The current form of the protocol is for better readability. However, the actual order of the messages is in the order of Σ^*-protocol, that consists of three type of messages: commitment, challenge and response. In Figure6, the messages are mark by three signs; Ⓐ as commitment, ⓒ as challenge, and ⓡ as response of a Σ^*-protocol. If we rearrange the messages (including the messages of ZKP sub-protocols) based on the type, according to Definition2.2, then Poxy.Π becomes a Σ^*-protocol.

$(msk', gpk') \leftarrow \texttt{Revoke}(msk, gpk, i)$ The group manager removes the i^{th} public key from the set Ξ. *i.e.*, $\Xi := \Xi \setminus \{pk_i\}$.

5.1 Security Analysis

In this section we provide the security analysis of Poxy protocol.

Theorem 1. *Assuming ZKP is a κ-sound (Definition.1) and ζ-zero-knowledge authentication protocol (Definition.4) for negligible values of κ and ζ, and* (Enc, Dec) *is an IND-CCA symmetric encryption scheme;*

Poxy *is (τ,δ)-complete, μ-TF-resistant, γ-sound, α-DF-resistant and zero-knowledge (Definition.4)* DBID *scheme for negligible values of δ, μ, γ and α, when t is linear in security parameter λ, and $\lambda.(1-p_{noise}-\varepsilon) > \lambda.\tau \geq \lambda - (\frac{1}{2}-2\varepsilon)\lceil\frac{\lambda}{2}\rceil$ for some constant $\varepsilon > 0$.*

Lemma 5. (Completeness). Poxy *is a (τ,δ)-complete* DBID *protocol for negligible value of δ, assuming $1-p_{noise}-\varepsilon > \tau$ for some constant $\varepsilon > 0$ and* (Enc, Dec) *is an IND-CCA symmetric encryption scheme.*

Proof. Consider a DBID game with Poxy scheme, in which there is no actor, and the provers and the verifier are honest, *i.e.*, $\forall x \in \mathcal{P} : x.Code = \texttt{Poxy}.\Pi.P(.) \wedge d(x.Loc, v.Loc) \leq \mathcal{D}$ for $v \in \mathcal{V} : v.Code = \texttt{Poxy}.\Pi.V(.)$. The verifier has access to the correct public key of provers. In this proof, we calculate the success chance of an honest prover in a Π protocol.

let's assume the verifier sends the challenge sequence $[c] = ([a],[b])$, where $[a] = [a^1...a^\lambda]$, and $[b] = [b_1...b_m]$ for $m \geq 0$. The prover receives $[c'] = ([a'],[b'])$ such that $\forall i \in \{1,...,\lambda\}, j \in \{1,...,t\} : \Pr[a_{ij} = a'_{ij}] = 1 - p_{noise}$. Correspondingly, the prover sends the response sequence $[r] = ([d],[e])$ and the verifier receives $[r'] = ([d'],[e'])$, where $\forall j \in \{1,...,\lambda\}, \forall j \in \{1,...,t\} : \Pr[d_{ij} = d'_{ij}] = 1 - p_{noise}$.

After the commitment opening phase, the verifier is able to find the noisy rounds (except with probability $Adv^{\texttt{Enc}}_{Corr}$ that is negligible). The probability of having at least τ noiseless fast challenge-response rounds for each $j \in \{1,...,t\}$ is $Tail(\lambda, \tau.\lambda, 1-p_{noise})$. As a result, the failure chance of the protocol is t times $1 - Tail(\lambda, \tau.\lambda, 1-p_{noise})$, which

is less than $e^{-2\varepsilon^2\lambda}$ based on Chernoff bound (Lemma14). And we have $t.e^{-2\varepsilon^2\lambda} < negl(\lambda)$ if t is linear to λ. □

Lemma 6. (Distance-Bounding). Poxy *is an* α*-DF-resistant* DBID *protocol for negligible value of* δ*, assuming* $\tau.\lambda \geq \lambda - (\frac{1}{2} - \varepsilon)\lfloor \frac{\lambda}{2} \rfloor$ *for some constant* $\varepsilon > 0$*, Diffie-Hellman key exchange is computationally unforgeable, ZKP is a* κ*-sound (Definition.1), and* ζ*-zero-knowledge authentication (Definition.4) for negligible* κ *and* ζ*.*

Proof. Consider a DBID game of Poxy scheme with no actors (*i.e.*, $\mathcal{T} = \emptyset$), honest verifier (*i.e.*, $v \in \mathcal{V} : v.Code = \texttt{Poxy}.\Pi.V(.)$) and far-away corrupted provers (*i.e.*, $\forall x \in \mathcal{P} : d(x.Loc, v.Loc) > \mathcal{D} \wedge x.Code \neq \texttt{Poxy}.\Pi.P(.)$) that might overlap in their execution time (*i.e.*, $x, y \in \mathcal{P} : y.St < x.St + exLen(x.Code) \leq y.St + exLen(y.Code)$).

In a successful Π protocol, the verifier gets $\xi' = \xi$ at the end of commitment opening phase, which implies that the prover has the correct value of x, unless negligible probability $Adv^{\texttt{DH}}_{forge}$, as the forgery chance of the semi-fresh Diffie-Hellman key exchange protocol [13], which is used in session key exchange phase. If the adversary succeeds in commitment and proof-of-knowledge phases, then they know a certain $\{(e'_j, k'_j)\}_{j=1...t}$ that satisfies "$e'_j \oplus k'_j = sk$" for all $j \in \{1, ..., t\}$. Therefore, they can efficiently find sk, unless negligible probability $(2\lambda + t)Adv^{ZKP}_{sound} + Adv^{DH}_{forge} = (2\lambda + t)\kappa + Adv^{DH}_{forge}$.

Since the value of sk is chosen randomly, then we have $\Pr[e_{ij} = k_{ij}] = \frac{1}{2}$ for $i = 1...\lambda$ and $j = 1...t$. In a DF attack, any collaboration of far-away provers in sending the response r_i is independent from the challenge bit c_i, according to the fundamental lemma (Lemma 4). Therefore, for the cases that $e_{ij} \neq k_{ij}$ (*i.e.*, half of the rounds), the success chance of adversary in sending the correct r_{ij} is $\frac{1}{2}$. As a result, the success chance of adversary in guessing the correct responses is limited by $Tail(\lfloor \frac{\lambda}{2} \rfloor, \tau.\lambda - \lceil \frac{\lambda}{2} \rceil, \frac{1}{2})^\lambda$, which is negligible based on Chernoff bound (Lemma14). □

We prove TF-resistance of Poxy in Lemma 8 that uses the following lemma.

Lemma 7. (Extractor). *Consider a* DBID *game* Γ *with TF attack (Property 4.4), for* Poxy *scheme. If there is a* Π *protocol in the game* Γ *in which, the verifier returns* $Out_V = 1$ *with non-negligible probability p, then there is a PPT extractor* $\mathcal{E}$*, that takes the view of all close-by participants, except the verifier (*$View^{\Gamma}_{\bar{S}}$*) as input, and outputs* $sk' = sk$ *with probability* $p - \mu$ *for negligible value of* μ*. This holds assuming that ZKP is* κ*-sound (Definition .1).*

Proof. **(Extractor).** Let's assume there is a TF adversary $\mathcal{A}$ that succeeds in Π protocol with non-negligible probability p, *i.e.*, generates a transcript $\xi = (A, [c], [r])$ that is accepted by the verifier with probability p. We construct a PPT extractor algorithm $\mathcal{E}$ for the secret key.

In a Π protocol from game Γ, the sequence of all challenges $[c]$ (slow and fast) is chosen randomly and broadcasted by the honest verifier. We define $[r] = [r]^{fast}||[r]^{slow}$ and $[c] = [c]^{fast}||[c]^{slow}$ where the superscripts show the type of the phase of the challenges.

Let S be the event that for all $i = \{1...\lambda\}$, and $j \in I_i$, the verifier's check $r_{ij} \oplus c_{ij1} = \bar{c_{ij0}} k_{ij} + c_{ij0}(sk_i \oplus k_{ij})$ hold true, where $c_i = c_{i0}||c_{i1}$. This can be verified by checking success of all *ZKP*'s in `commitment` phase, all the *ZKP*'s in `PoK` phase and all the checks in `commitment opening` phase. In other words, when in `commitment opening` phase all checks succeed, we have $r_{ij} \oplus c_{ij1} = \bar{c_{ij0}} k_{ij} + c_{ij0}(e_{ij})$. And when all *ZKP*'s of `commitment` phase succeed, we have the commitment to every bit of k_{ij} and e_{ij} for $i = \{1...\lambda\}$ and $j = \{1...t\}$, that builds z_j as the commitment to $e_j \oplus k_j$. And when all *ZKP*'s of `commitment` phase succeed, we have $sk = e_j \oplus k_j$ for all $j = \{1...t\}$. This implies the occurrence of S.

Since *ZKP* is κ-sound, we conclude that at least for one *ZKP* we have $\Pr[\text{succ } ZKP|\neg S] \leq \kappa$ and then $\Pr[\text{succ } ZKP, \neg S] \leq \kappa$, where $\neg S$ indicates negation of S. So we have $\Pr[\text{valid } \xi, \neg S] \leq \kappa$.

Based on the fundamental lemma (Lemma 4), any valid response r_{ij} that is received by the verifier consists of two parts $r_{\mathcal{F}}$ and $r_{\mathcal{S}}$ that $r_{\mathcal{F}} = Msg_{\mathcal{F}\to v(t')}(\bar{c_{ij}})$ and $\exists \mathcal{J}' : r_{\mathcal{S}} = \mathcal{J}'(View^{\Gamma}_{\mathcal{S}}(\bar{c_{ij}}), c_{ij}, Msg_{\mathcal{F}\to\mathcal{S}}(\bar{c_{ij}}))$. We have $r_{ij} = sum(r_{\mathcal{F}}, r_{\mathcal{S}})$, for a deterministic function $sum()$ that is determined by the physical communication channel. We assume there exist a deterministic subtraction function $sub()$ such that for any $x = sum(z,y)$, we have $z = sub(x;y)$ and $y = sub(x;z)$.

We consider the view of close-by participants before sending the response r_{ij}, *i.e.*, $View^{\Gamma}_{\mathcal{S}}(\neg r_{ij})$, relative to the view of the close-by participants before seeing the challenge c_{ij}, *i.e.*, $View^{\Gamma}_{\mathcal{S}}(\neg c_{ij})$. In the time period between receiving the challenge c_{ij} and sending r_{ij}, the close-by participants can receive messages from two different sources: the verifier, and the far-away participants. The only message from the verifier in this period is c_{ij} and we indicate the messages from the far-away participants as $Msg_{\mathcal{F}\to\mathcal{S}}(\neg c_{ij})$ which is independent from c_{ij}. So we have $View^{\Gamma}_{\mathcal{S}}(\neg r_{ij}) = View^{\Gamma}_{\mathcal{S}}(\neg c_{ij})||c_{ij}||Msg_{\mathcal{F}\to\mathcal{S}}(\neg c_{ij})$.

We conclude that there is an algorithm $\mathcal{J}'$ that takes $View^{\Gamma}_{\mathcal{S}}(\neg r_{ij})$ and generates $r_{\mathcal{S}}$, such that the correct response r_{ij} is calculated as $r_{ij} = sum(Msg_{\mathcal{F}\to v(t')}(\bar{c_{ij}}), r_{\mathcal{S}})$. Note that $View^{\Gamma}_{\mathcal{S}}(\neg r_i)$ includes the challenge c_{ij}.

Since there is an algorithm $\mathcal{J}'$ that generates $resp_{ij} = \mathcal{J}'(View^{\Gamma}_{\mathcal{S}}(\neg c_{ij})||c_{ij}||Msg_{\mathcal{F}\to\mathcal{S}}(\neg c_{ij}))$, then we construct the algorithm $\mathcal{J}$ that calls $\mathcal{J}'$ four times for different values of c_{ij} with the following inputs: $resp^d_{ij} = \mathcal{J}'(View^{\Gamma}_{\mathcal{S}}(\neg c_{ij})||d||Msg_{\mathcal{F}\to\mathcal{S}}(\neg c_{ij}))$ for $d = \{00, 01, 10, 11\}$. As a result, $\mathcal{J}$ returns $\{resp^d_{ij}\}_{d=\{00,01,10,11\}}$ such that the output of the function $sum(resp^d_{ij}, r_{\mathcal{F}})$ is the correct answer when the challenge is $c_{ij} = d$, which makes correct $resp^d_{ij}$. Note that the value of $r_{\mathcal{F}}$ is not known to the extractor.

From the final view of the close-by participants, we can find their partial view at the time of sending response r_{ij} for all $j = 1\ldots\lambda$ and $i = 1\ldots t$, and then call the algorithm $\mathcal{J}$. So we can calculate $(resp^{00}_{ij}, resp^{01}_{ij}, resp^{10}_{ij}, resp^{11}_{ij})$ for all $i = 1\ldots\lambda$ and $j = 1\ldots t$, from the final view of the close-by participants.

Since we have $sub(sum(resp^{11}_{ij}, r_{\mathcal{F}}); sum(resp^{01}_{ij}, r_{\mathcal{F}})) = sub(resp^{11}_{ij}; resp^{01}_{ij})$, and $sub(sum(resp^{10}_{ij}, r_{\mathcal{F}}); sum(resp^{00}_{ij}, r_{\mathcal{F}})) = sub(resp^{10}_{ij}; resp^{00}_{ij})$, then for all values of $resp^{00}_{ij}$,

$resp_{ij}^{01}$, $resp_{ij}^{10}$, $resp_{ij}^{11}$ and $r_{\mathcal{F}}$, calculating the difference between the two correct responses for both values of the challenge bit c_{ij0} given c_{ij1}, is equal to the difference between $resp_{ij}^{1||c_{ij1}}$ and $resp_{ij}^{0||c_{ij1}}$, which are computable from the final view of close-by participants ($View_{\mathcal{S}}^{\Gamma}$). In the following, we use function $sub^{\oplus}(.;.)$ that is defined as: $sub^{\oplus}(a;b) := 0$ if $sub(a;b) = 0$, and $sub^{\oplus}(a;b) := 1$ otherwise.

We build the extractor $\mathcal{E}$ as follows: $\mathcal{E}$ runs $(resp_{ij}^{00}, resp_{ij}^{10}) \leftarrow \mathcal{J}(View_{\mathcal{S}}^{\Gamma})$ for all $i = 1 \ldots \lambda$ and $j = 1 \ldots t$, and finds $\xi_{ij} = sub^{\oplus}(resp_{ij}^{10}; resp_{ij}^{00})$, and then calculates the key bits $sk_i' = majority(\xi_{ij})$ over $j = \{1...t\}$.

Note that in a simple case that the received message at receiver is sent from a single source, we have $sub^{\oplus}(resp_{ij}^{10}; resp_{ij}^{00}) = resp_{ij}^{10} \oplus resp_{ij}^{00}$ and also for all $d = \{0,1\}$ we have $resp_{ij}^{d||1} = resp_{ij}^{d||0} \oplus 1$. In the following we calculate the success chance of this extractor.

A received response $r_{ij}^{d||b}$ for $d, b \in \{0,1\}$ is correct if $r_{ij}^{d||b} = (\bar{d}.k_{ij} + d.e_{ij}) \oplus b$. For succeeding in the Π protocol (event S), for at least $\tau.t$ values of $j \in \{1...t\}$ the received response should be correct for all values of $i \in \{1, ..., \lambda\}$. Since the challenge $c_{i,j}$ is chosen randomly and we assume the secret sk is chosen randomly too, then the distribution of a correct response is uniform. So the messages of far-away provers, *i.e.*, $r_{\mathcal{F}}$, have at most $\frac{1}{2}$ chance of being the correct response. Therefore, for at least $\tau.t$ values of $j = \{1...t\}$, the close-by actors help the prover.

An extracted response $resp_{ij}^{d||0}$ for $d \in \{0,1\}$ is correct if $sum(resp_{ij}^{d||0}, r_{\mathcal{F}}) = \bar{d}.k_{ij} + d.e_{ij}$. We define $R_{ij} \in \{0,1,2\}$ as the number of challenge bits $d \in \{0,1\}$ for which the extracted response $resp_{ij}^{d||0}$ is correct. If we have $R_{ij} = 2$ then $\xi_{ij} = sk_i$, but if $R_{ij} = 1$ then we might have $\xi_{ij} \neq sk_i$.

Consider $R = (R^1, ..., R^{\lambda})$ for vector $R^i = (R_{i1}, ..., R_{it})$, where R_{ij} is defined as above and calculated by comparing the $(resp_{ij}^{00}, resp_{ij}^{10})$ with correct responses for all $i = 1 \ldots \lambda$ and $j = 1 \ldots t$. We define the set $\mathcal{R}$ as all vectors in $\{0,1,2\}^t$ such that at least $\lfloor \frac{t}{2} \rfloor + 1$ values are 2. For a vector $R^i \in \mathcal{R}$ we have $\xi_{ij} = sk_i$.

Since the verifier selects the challenges randomly, then knowing R_{ij} allows us to find the probability that the response $r_{ij} = sum(resp_{ij}, r_{\mathcal{F}})$ is correct: if $R_{ij} = 2$ then this probability is 1, otherwise this probability is at most $\frac{1}{2}$ (a response can always be guessed randomly). For any $R^i \notin \mathcal{R}$, at least $\lceil \frac{t}{2} \rceil$ values of $R_{i,j} \neq 2$, and at least $\tau.t - \lfloor \frac{t}{2} \rfloor$ of them have to be guesses randomly for success. Therefore, the probability of success in the i^{th} round by having $R^i \notin \mathcal{R}$ is limited by $p_B = Tail(\lceil \frac{t}{2} \rceil, \tau.t - \lfloor \frac{t}{2} \rfloor, \frac{1}{2})$, where $Tail(n,k,\rho) := \sum_{i=k}^{n} \binom{n}{i} \rho^i (1-\rho)^{n-i}$.

We define W as the random variable showing the number of i's that vector $R^i \notin \mathcal{R}$. So we have $\Pr[S|W = w] \leq p_B^w$. So $\Pr[S, W = w] \leq p_B^w \Pr[W = w]$ and then $\Pr[S, W \geq w] \leq p_B^w$. As a result, we have:

$$\Pr[W \geq w, \text{valid } \xi] \leq \Pr[\neg S, \text{valid } \xi] + \Pr[S, W \geq w] \leq \kappa + p_B^w$$

Each index j where $sk_j \neq sk'_j$, corresponds to $R_j \notin \mathcal{R}$. Therefore, having the verifier outputting $Out_V = 1$ (*i.e.*, ξ is valid) and the extractor giving at least w errors occurs with probability bounded by $\mu = \kappa + p_B^w$. This implies that we can build a key extractor from $View_S^\Gamma$ that follows the success chance of the TF attack (*i.e.*, p), except with probability μ, which is negligible due to Chernoff bound (Lemma14). □

Lemma 8. (TF-resistance). `Poxy` *is a* μ-**(TF-resistance)** `DBID` **scheme for negligible value of** μ**.**

Proof. Consider a `DBID` game with single limitation of having honest verifier (*i.e.*, $\mathcal{V}.Code = \texttt{ID}.V(.)$). Based on the TF definition (Property4.4), here we show that if there is a non-negligible TF attacker, then these existence impersonator that take the view of close-by actors and succeeds with same chance, except negligible probability μ.

If there is a TF attacker that succeeds with probability p, then based on Lemma 7, there is an extractor that takes the view of close-by actors and returns the secret key of prover $sk' = sk$ with success chance $p - \mu$ for negligible μ.

Therefore, the impersonator first runs the extractor and gets prover's secret key sk', and then runs the prover protocol $P(sk')$ with the verifier. The success chance of the impersonator is at least $p - \mu$. □

Lemma 9. (Zero-Knowledge). *By assuming ZKP and* `Enc` *operations are computationally negl-zero-knowledge, then* Π *protocol of* `Poxy` *scheme with honest prover is a computationally* ζ*-zero-knowledge protocol according to Definition.4, where* $\zeta = negl$*.*

Proof. Here we show that in Π operation, given two participants $P(sk)$ and $V^*(pk, gpk)$, there exists a simulator $S(pk, gpk)$ such that the view of $V^*(pk, gpk)$ in the interaction $V^*(pk, gpk) \leftrightarrow P(sk)$ is computationally indistinguishable from the output of $S(pk, gpk)$. We basically show that there is a simulator that consist of some smaller simulators for each part of the protocol.

The view of adversary at the end is as follows: $\forall j \in \{1,...,t\}, i \in \{1,...,\lambda\}, b \in \{\text{"}k\text{"},\text{"}e\text{"}\}$: $(View_{\mathcal{V}}(\texttt{ZKP})^{t.\lambda+t}, x_{\mathcal{V}}, y_{\mathcal{V}}, \mu, \xi, A_{b,ij}, r_{ij}, c_{ij}, \delta_{ij}, \gamma_{ij})$ for $\forall j \in \{1,...,t\}, i \in \{1,...,\lambda\}$, where the adversary chooses values $x_{\mathcal{V}}, y_{\mathcal{V}}, \xi, \mu, \{c_{ij}\}$.

Since *ZKP* and `Enc` operations are assumed to be computationally *negl*-zero-knowledge, then by definition there is a simulator for the view of V^* after these operations, that only takes the view of V^* as input and returns an indistinguishable output from the normal case. Now we can remove the prover side of *ZKP* operations. The same thing applies to the Diffie-Helman key exchange session. Therefore, the view of adversary is reduced to: $\forall j \in \{1,...,t\}, i \in \{1,...,\lambda\}$: $(A_{k,ij}, A_{e,ij}, r_{ij}, \gamma_{ij}, c_{ij})$, where the adversary chooses the values $\{c_{ij}\}$.

Since the relation between e and k is no longer checked by the above removals, then we can make their mutual relation to be random on the prover side. *i.e.*, choose e randomly. Therefore both values of e and k will be random and the view of verifier in

this case is indistinguishable from the case when e is computed as $e_{ij} = sk_i \oplus k_{ij}$ for $j \in \{1,...,t\}, i \in \{1,...,\lambda\}$. In this modified case, the value of sk is no longer used. Therefore, we can replace sk at the prover with randomness and still be indistinguishable at the verifier. As a result, we built a simulator that doesn't use the secret of prover (sk) and produces indistinguishable output from the output of verifier while it's interacting with real prover. □

Lemma 10. (Soundness). `Poxy` *is a* γ*-sound* `DBID` *(Prop4.2) protocol for* $\gamma = negl$*, under the following assumptions;* t *is polynomial,* $\tau > \frac{1}{2} + \varepsilon$ *for some constant* ε*, ZKP is a* κ*-sound (Definition.1) and* ζ*-zero-knowledge authentication (Definition.4) for negligible* κ *and* ζ*, and* `Enc` *is* ζ*-zero-knowledge.*

Proof. According to the `DBID` game settings, we have some lists of provers $\mathcal{P}^j \in \mathcal{P}$ that there is no overlap in the execution time of any list $\mathcal{P}^j$, however the provers of two different lists $\mathcal{P}^j \neq \mathcal{P}^i$ can run simultaneously. The corrupted actors $\mathcal{T}$ are controlled by the adversary. In this game, the adversary succeeds if any of the following is true;

(i) $\exists t \in \mathbb{N}, \forall p \in \mathcal{P} : t = Time(Out_{\mathcal{V}} = 1), t \notin [p.St, p.St + exLen(p.Code)]$, or

(ii) $\exists (p,t) \in (\mathcal{P}, \mathbb{N}) : t = Time(Out_{\mathcal{V}} = 1), t \in [p.St, p.St + exLen(p.Code)], d(p, \mathcal{V}) > \mathcal{D}$

In this proof, we calculate the success chance of the adversary in both conditions.

Here we introduce two time periods: the learning phase is from the beginning of the game till the beginning of the first session that makes the adversary to win, and the attack session, that is right after the *learning phase* till end of the `ID` operation that makes the adversary to win.

Let's assume the adversary uses $\{A_{e,ij}\}$ and $\{A_{k,ij}\}$ in the commitment phase of the attack session, which are the bit commitments of e and k. Let's assume the response table used in fast phase is e' and k', which is known by the close-by participant.

In the first condition (*i.e.*, no active prover during attack session), since there is `ZKP` for all $2\lambda t$ of the committed values ($\{A_{e,ij}\}$ and $\{A_{k,ij}\}$), the adversary should know the committed values e and k, unless with negligible probability $2\lambda.t.Adv^{ZKP}_{sound} = 2\lambda.t.\kappa$. If e and k are correct (*i.e.*, $e_{ij} = sk_i \oplus k_{ij}$ for all $j \in \{1,...,t\}, i \in \{1,...,\lambda\}$), then the adversary can efficiently calculate the value of sk, which is in contradiction with Lemma9 unless with probability $\zeta = Adv_{\texttt{ZKP}}$. If e and k are incorrect, then the adversary would not be able to pass the `ZKP` of `proof-of-knowledge` phase, unless with negligible probability $t.Adv^{ZKP}_{sound} = t.\kappa$.

In the second condition (all active provers are far-away from verifier during attack session), based on Lemma9, we know that the close-by participants (actors) have negligible information about sk. In this case, for each row of the response table (*i.e.*, the two bits e'_{ij} and k'_{ij}) there is one bit uncertainty ($\Pr[e_{ij} = e'_{ij}, k_{ij} = k'_{ij}] = \frac{1}{2}$). Otherwise, the adversary would gain some information about sk_i. Therefore, by assuming that the challenge bits of fast phase are chosen randomly, then for half of the fast rounds, the verifier sends the challenge value that the actor needs to make a guess for the correct answer. As a result, for each value of $i \in \{1,...,\lambda\}$ the success chance of related round

in fast phase is $\frac{1}{2}^{\frac{t}{2}-(1-\tau)t} = \frac{1}{2}^{(\tau-\frac{1}{2})t}$. Therefore, the success chance of the adversary is limited by $2\lambda.t.\kappa + t.\kappa + \frac{1}{2}^{(\tau-\frac{1}{2})t.\lambda} + \zeta$, which is negligible. □

6 ProProx Scheme [28]

ProProx scheme is a public key DB protocol [28] that fits our DBID model. We also prove security of the protocol in this model. The details of the operations of ProProx is given below. Let λ and n be the security parameters that are linearly related.

$(msk, gpk) \leftarrow \texttt{Init}(1^\lambda)$ The group manager initializes a Goldwasser-Micali cryptosystem with λ bit security: chooses $N = p.q$ and chooses θ that is a quadratic residue modulo N. It also chooses chooses $b \in \{0,1\}^n$ with Hamming weight of $\lfloor \frac{n}{2} \rfloor$. The group master key is $msk = (p,q)$; the group public key is $gpk = (N, b, \theta, \Xi)$ where $\Xi = \emptyset$.

$(sk, pk) \leftarrow \texttt{KeyGen}(msk, gpk)$ Assume $l-1$ users have joined the group and their public keys are in the set $\Xi = \{pk_1, ..., pk_{l-1}\}$ that is published by the group manager. For the l^{th} user, the group manager generates a key pair (sk, pk), such that $sk \in_R \{0,1\}^\lambda$ and pk is the output of a homomorphic and deterministic commitment scheme $Com_H()$ on $sk = (sk_1 ... sk_\lambda)$; that is $pk = Com_H(sk) = (Com(sk_1; H(sk,1)), ..., Com(sk_\lambda; H(sk,\lambda)))$, where $Com(u;v)$ is Goldwasser-Micali encryption ($= \theta^u v^2 \pmod N$) and H is a one-way hash function. The group manager securely sends the key pair to the new user and adds the public key pk to the set Ξ.

$accept/reject \leftarrow \Pi\{P(sk, pk) \leftrightarrow V(pk)\}$ When a prover ($\mathcal{P}_l$) of a registered user wants to run the DBID.Π protocol with the verifier, they will follow the protocol described in Figure7.

In the verification phase, the prover and the verifier agree on a list $I = (I_1, ..., I_\lambda)$, where each I_j consists of $\lceil \tau.n \rceil$ indices from 1 to n. Both parties believe $\forall j = \{1...\lambda\}, i \in I_j : c_{i,j} = c'_{i,j}$ and $r_{i,j} = r'_{i,j}$. The verifier then checks whether responses are within the required time interval. The prover and the verifier then run an interactive zero-knowledge proof (ZKP) to show that the responses $r_{i,j}$, $j = \{1...\lambda\}, i \in I_j$ are consistent with the corresponding $A_{i,j}$'s and y_j's. If the verification fails, the verifier aborts and outputs $Out_v = 0$, otherwise, outputs $Out_V = 1$.

$(msk', gpk') \leftarrow \texttt{Revoke}(msk, gpk, i)$ The group manager removes the i^{th} public key from the set Ξ. *i.e.*, $\Xi := \Xi \setminus \{pk_i\}$.

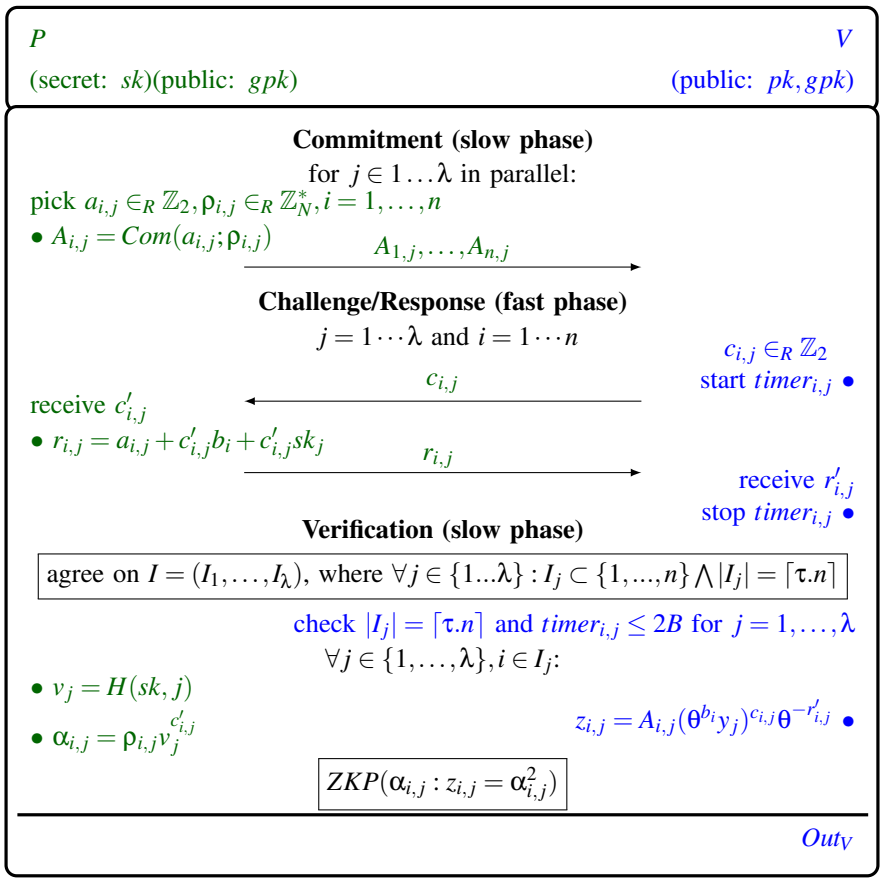

Figure 7: Π protocol of `ProProx` scheme. *Com* is Goldwasser-Micali encryption. τ is the minimum threshold ratio of noiseless fast rounds. *ZKP* is an interactive zero-knowledge proof. The number of fast rounds is $n.\lambda$. In each fast round, the verifier sends one-bit challenge, and receives the corresponding response.

6.1 Security Analysis

To prove that the above protocol satisfies our security definition, we first note that the Π protocol of `ProProx` scheme (*i.e.*, Figure 7) can be seen as a Σ^*-protocol (Definition 2.2). This is true because, assuming the agreement step (on the value of I) and the ZKP step can be written as Σ^*-protocols, their concatenation is also a Σ^*-protocol because one can consider all first message commitment of the protocols as a single commitment phase, and all verification functions stay at the end. The remaining challenge and response messages are concatenated and form the challenge and responses of the combined protocol.

Theorem 2. *Assuming $Com(u;v)$ is a perfect binding and computationally hiding*

homomorphic bit commitment scheme (Definition .2), $Com_H()$ is one-way function (Definition .3), and ZKP is a κ-sound (Definition.1) and ζ-zero-knowledge authentication protocol (Definition.4) for negligible values of κ and ζ;

ProProx *is* (τ,δ)*-complete,* μ*-TF-resistant,* γ*-sound and* α*-DF-resistant* DBID *scheme for negligible values of* δ*,* μ*,* γ *and* α*, when n is linear in security parameter* λ*, and* $n.(1-p_{noise}-\varepsilon) > n.\tau \geq n-(\frac{1}{2}-2\varepsilon)\lceil\frac{n}{2}\rceil$ *for some constant* $\varepsilon > 0$*.*

ProProx is proven to be *complete*, *DF-resistant* and *zero-knowledge* (Definition.4) in [28]. Our definitions of these properties remain unchanged. So we only need to prove *TF-resistance* and *soundness* properties of ProProx scheme.

We prove TF-resistance of ProProx in Lemma 12 that uses the following lemma.

Lemma 11. (Extractor). *Consider a* DBID *game* Γ *with TF attack (Property 4.4), for* ProProx *scheme. If there is a* Π *protocol in the game* Γ *in which, the verifier returns* $Out_V = 1$ *with non-negligible probability p, then there is a PPT extractor* $\mathcal{E}$*, that takes the view of all close-by participants, except the verifier (*$View_S^{\Gamma}$*) as input, and outputs* $sk' = sk$ *with probability* $p-\mu$*, for negligible value of* μ*. This holds assuming,* $Com(u;v)$ *is a perfect binding computational hiding homomorphic bit commitment scheme (Definition .2), and ZKP is a* κ*-sound authentication protocol (Definition .1).*

Note that the extractor of this lemma, has a critical difference from the extractor that is considered in security analysis of ProProx scheme in the original paper [28]; the input of the extractor of the original paper takes the view of the verifier is the view of all close-by participants, including the verifier, but the input of the above extractor is the view of all close-by participants (excluding the verifier). By excluding the view of the verifier of a TF attack from the view of the extractor, the close-by participants can extract the secret-key of the prover, even when the prover is using directional antenna to communicate directly to the verifier.

In the security analysis of the extractor of the original paper, it is assumed that a correct response is solely sent from a single close-by participant. However, there might be a case that the received message $r_{i,j}$ is the combination of a message that is sent from a far-away source and a message that is sent from a close-by source. In our extractor, we include this case.

Proof. **(Extractor).** Let's assume there is a TF adversary $\mathcal{A}$ that succeeds in Π protocol with non-negligible probability p, *i.e.*, generates a transcript $\xi = (A,[c],[r])$ that is accepted by the verifier with probability p. We construct a PPT extractor algorithm $\mathcal{E}$ for the secret key.

In a protocol Π from game Γ, the sequence of all challenges $[c]$ (slow and fast) is chosen randomly and broadcasted by the honest verifier. We define $[r] = [r]^{fast}||[r]^{slow}$ and $[c] = [c]^{fast}||[c]^{slow}$ where the superscripts show the type of the phase of the challenges.

Because of the perfect binding commitment (Definition .2), the value of the public key pk uniquely determines $sk = Com^{-1}(pk)$, and the value of $A_{i,j}$ uniquely determines

$a_{i,j} = Com^{-1}(A_{i,j})$. We emphasis that these values are not being calculate, but we just mathematically define them based on the view of the verifier.

Let S be the event that for all j, and $i \in I_j$, the verifier's checks $r_{i,j} = a_{i,j} + c_{i,j}b_i + c_{i,j}sk_j$ hold true. This can be verified by checking success of ZKP, for all the corresponding j and $i \in I_j$. In other words, when ZKP succeeds for all j, and $i \in I_j$, we have $z_{i,j}$ as commitment to $a_{i,j} + c_{i,j}b_i + c_{i,j}sk_j - r_{i,j}$, which implies the occurrence of S.

Since ZKP is κ-sound, we conclude that $\Pr[\text{succ } ZKP|\neg S] \leq \kappa$ and then $\Pr[\text{succ } ZKP, \neg S] \leq \kappa$, where $\neg S$ indicates negation of S. So we have $\Pr[\text{valid } \xi, \neg S] \leq \kappa$.

Based on the fundamental lemma (Lemma 4), any valid response $r_{i,j}$ that is received by the verifier consists of two parts $r_{\mathcal{F}}$ and $r_{\mathcal{S}}$ that $r_{\mathcal{F}} = Msg_{\mathcal{F}\to v(t')}(\overline{c_{i,j}})$ and $\exists \mathcal{J}' : r_{\mathcal{S}} = \mathcal{J}'(View^{\Gamma}_{\mathcal{S}}(\overline{c_{i,j}}), c_{i,j}, Msg_{\mathcal{F}\to\mathcal{S}}(\overline{c_{i,j}}))$. We have $r_{i,j} = sum(r_{\mathcal{F}}, r_{\mathcal{S}})$, for a deterministic function $sum()$ that is determined by the physical communication channel. We assume there exist a deterministic subtraction function $sub()$ such that for any $x = sum(z,y)$, we have $z = sub(x;y)$ and $y = sub(x;z)$.

We consider the view of close-by participants before sending the response $r_{i,j}$, *i.e.*, $View^{\Gamma}_{\mathcal{S}}(\neg r_{i,j})$, relative to the view of the close-by participants before seeing the challenge $c_{i,j}$, *i.e.*, $View^{\Gamma}_{\mathcal{S}}(\neg c_{i,j})$. In the time period between receiving the challenge $c_{i,j}$ and sending $r_{i,j}$, the close-by participants can receive messages from two different sources: the verifier, and the far-away participants. The only message from the verifier in this period is $c_{i,j}$ and we indicate the messages from the far-away participants as $Msg_{\mathcal{F}\to\mathcal{S}}(\neg c_{i,j})$ which is independent from $c_{i,j}$. So we have $View^{\Gamma}_{\mathcal{S}}(\neg r_{i,j}) = View^{\Gamma}_{\mathcal{S}}(\neg c_{i,j})||c_{i,j}||Msg_{\mathcal{F}\to\mathcal{S}}(\neg c_{i,j})$.

We conclude that there is an algorithm $\mathcal{J}'$ that takes $View^{\Gamma}_{\mathcal{S}}(\neg r_{i,j})$ and generates $r_{\mathcal{S}}$, such that the correct response $r_{i,j}$ is calculable as $r_{i,j} = sum(Msg_{\mathcal{F}\to v(t')}(\overline{c_{i,j}}), r_{\mathcal{S}})$. Note that $View^{\Gamma}_{\mathcal{S}}(\neg r_{i,j})$ includes the challenge $c_{i,j}$.

Since there is an algorithm $\mathcal{J}'$ that $resp_{i,j} = \mathcal{J}'(View^{\Gamma}_{\mathcal{S}}(\neg c_{i,j})||c_{i,j}||Msg_{\mathcal{F}\to\mathcal{S}}(\neg c_{i,j}))$, then we construct the algorithm $\mathcal{J}$ that calls $\mathcal{J}'$ two times with the following inputs: $resp^0_{i,j} = \mathcal{J}'(View^{\Gamma}_{\mathcal{S}}(\neg c_{i,j})||0||Msg_{\mathcal{F}\to\mathcal{S}}(\neg c_{i,j}))$ and then $resp^1_{i,j} = \mathcal{J}'(View^{\Gamma}_{\mathcal{S}}(\neg c_{i,j})||1||Msg_{\mathcal{F}\to\mathcal{S}}(\neg c_{i,j}))$. As a result, $\mathcal{J}$ returns a pair $(resp^0_{i,j}, resp^1_{i,j})$ such that the output of the functions $sum(resp^0_{i,j}, r_{\mathcal{F}})$ and $sum(resp^1_{i,j}, r_{\mathcal{F}})$ are the correct answer for the two possible cases of the challenge bit $c_{i,j}$. We say $resp^0_{i,j}$ (or $resp^1_{i,j}$) is correct if $sum(resp^0_{i,j}, r_{\mathcal{F}})$ (or $sum(resp^1_{i,j}, r_{\mathcal{F}})$) is the correct response to the challenge $c_{i,j} = 0$ (or $c_{i,j} = 1$). Note that the value of $r_{\mathcal{F}}$ is not known to the extractor.

From the final view of the close-by participants, we can find their partial view at the time of sending response $r_{i,j}$ for all $j = 1 \ldots \lambda$ and $i = 1 \ldots n$, and then call the algorithm $\mathcal{J}$. So we can calculate $(resp^0_{i,j}, resp^1_{i,j})$ for all $j = 1 \ldots \lambda$ and $i = 1 \ldots n$, from the final view of the close-by participants. Since we have

$$sub(sum(resp^1_{i,j}, r_{\mathcal{F}}); sum(resp^0_{i,j}, r_{\mathcal{F}})) = sub(resp^1_{i,j}; resp^0_{i,j}),$$

therefore, calculating the difference between the two correct responses for both values of the challenge bit $c_{i,j}$ is equal to the difference between $resp^1_{i,j}$ and $resp^0_{i,j}$, which are

computable from the final view of close-by participants ($View_{\mathcal{S}}^{\Gamma}$).

We build the extractor $\mathcal{E}$ as follows: $\mathcal{E}$ runs $(resp_{i,j}^0, resp_{i,j}^1) \leftarrow \mathcal{J}(View_{\mathcal{S}}^{\Gamma})$ for all $j = 1 \ldots \lambda$ and $i = 1 \ldots n$. Then it guesses the bits of secret $sk'_j = majority(\delta_{1,j} ... \delta_{n,j})$ for $j = \{1...\lambda\}$, where $\delta_{i,j} = sub(resp_{i,j}^1; resp_{i,j}^0) - b_i$ for $i = \{1...n\}$.

Note that in a simple case that the received message at receiver is sent from a single source, we have $sub(resp_{i,j}^1; resp_{i,j}^0) = resp_{i,j}^1 - resp_{i,j}^0$. In the following we calculate the success chance of this extractor;

A response $resp_{i,j}^d$ for $d \in \{0,1\}$ is correct if $sum(resp_{i,j}^d, r_{\mathcal{F}}) = a_{i,j} + d.b_i + d.sk_j$. We define $R_j = [R_{1,j} ... R_{n,j}]$ where $R_{1,j}$ is the number of challenge bits $d \in \{0,1\}$ for which the response $resp_{i,j}^d$ is correct. If we have $R_{i,j} = 2$ then $\delta_{i,j} = sk_j$, but if $R_{i,j} = 1$ then we might have $\delta_{i,j} \neq sk_j$.

Consider $R = (R^1, ..., R^{\lambda})$ for vector $R^i = (R_{i1}, ..., R_{in})$, where $R_{i,j}$ is defined as above and calculated by comparing the $(resp_{i,j}^0, resp_{i,j}^1)$ with correct responses for all $j = 1 \ldots \lambda$ and $i = 1 \ldots n$. We define the set $\mathcal{R}$ as all vectors in $\{0,1,2\}^n$ that have at least $\lfloor \frac{n}{2} \rfloor + 1$ values of 2. For a vector $R_j \in \mathcal{R}$, we have a majority of i's that $\delta_{i,j} = sk_j$, which implies $sk'_j = sk_j$.

Since the verifier selects the challenges randomly, then knowing $R_{i,j}$ allows us to find the probability that the response $r_{i,j} = sum(resp_{i,j}, r_{\mathcal{F}})$ is correct: if $R_{i,j} = 2$ then this probability is 1, otherwise this probability is at most $\frac{1}{2}$ (this is true that a response can always be guessed randomly). If W is the random variable giving the number R_js that $R_j \notin \mathcal{R}$, we have $\Pr[S|W = w] = p_B^w$ where $p_B = Tail(\lceil \frac{n}{2} \rceil, \tau - \lfloor \frac{n}{2} \rfloor, \frac{1}{2})$, defined in Lemma 14. So $\Pr[S, W = w] \leq p_B^w \Pr[W = w]$ and then $\Pr[S, W \geq w] \leq p_B^w$. As a result, we have:

$$\Pr[W \geq w, \text{valid}\xi] \leq \Pr[\neg S, \text{valid } \xi] + \Pr[S, W \geq w] \leq \kappa + p_B^w$$

Each index j where $sk_j \neq sk'_j$, corresponds to $R_j \notin \mathcal{R}$. Therefore, having the verifier outputting $Out_V = 1$ (*i.e.*, ξ is valid) and the extractor giving at least 1 error occurs with probability bounded by $\mu = \kappa + p_B$. This implies that we can build a key extractor from $View_{\mathcal{S}}^{\Gamma}$ that follows the success chance of the TF attack, except with probability μ that is negligible because of Chernoff bound (Lemma14). □

Lemma 12. (TF-resistance). `ProProx` *is a μ-TF-resistant* `DBID` *(Prop4.4) scheme for negligible value of μ, assuming Com is a perfectly binding computational hiding homomorphic bit commitment (Definition .2), and ZKP is a κ-sound authentication protocol (Definition .1).*

Proof. According to the TF-resistance definition (Property 4.4), we need to show that if there is a game Γ for a Π protocol in which that the verifier returns $Out_V = 1$ with non-negligible probability κ, then there exists a close-by actor $\mathcal{R}$ that for any challenge sequence $[c]$ can create a valid transcript with probability at least $\kappa - \mu$ for a negligible μ, using the view of all close-by participants, excluding the verifier ($View_{\mathcal{S}}^{\Gamma}$).

Based on Lemma 11, the existence of TF attacker with non-negligible success probability κ, implies the existence of a key extractor $sk' \leftarrow \mathcal{E}(View_S^{\Gamma})$ with the success chance of at least $\kappa - \mu'$ for a negligible μ'.

So we make $\mathcal{R}$ as follows: After a successful TF attack, $\mathcal{R}$ calls the above extractor $\mathcal{E}$ to find sk'. Then $\mathcal{R}$ runs the $\texttt{ProProx}.\Pi.P(sk', gpk)$ interactive algorithm in order to generate valid transcript with correct timing for any challenge that is generated by the verifier. Since ProProx is (τ, δ)-complete, the verifier outputs $Out_V = 1$, unless negligible probability δ. Therefore, the success chance of $\mathcal{R}$ is at least $\kappa - \mu$ for negligible $\mu = \mu' - \delta$. □

Lemma 13. (Soundness). ProProx *is a* γ*-sound* DBID *(Prop4.2) scheme for* $\gamma = negl(\lambda)$*, if the followings hold:* $\tau.n \geq n - (\frac{1}{2} - 2\varepsilon)\lceil \frac{n}{2} \rceil$ *for some constant* ε*; Com*$_H$ *is one-way; Com is homomorphic bit commitment with all properties of Definition.2; ZKP is a* κ*-sound (Definition.1) and* ζ*-zero-knowledge authentication (Definition.4) for negligible* κ *and* ζ*.*

Proof. In a Π protocol, the verifier receives a transcript $\xi = (A, [c], [r])$. There are two possible participant arrangements for the winning conditions of a soundness adversary that result in the verifier returning $Out_V = 1$: (i) all active provers are far-away from the verifier, (ii) there is no active prover during the Π protocol (*i.e.*, there might be close-by provers but they are not active). In the following we show that the success probability of the adversary in both cases is negligible. In other words, the success probability of generating a valid transcript $\xi = (A, [c], [r])$ when the challenge sequence $[c]$ is generated by the verifier, is negligible.

In the first case, the adversary cannot simply relay the messages because of the extra delay and the fact that the responses are from out of bound locations. In this case the verifier will reject the instance. If there is a PPT adversary $\mathcal{A}$ that can guess at least $\tau.n$ out of n responses for each key bit with non-negligible probability (*i.e.*, guessing all bits of $\forall j \in \{1, ..., \lambda\} I_j \subset \{1, ..., n\}$ such that $|I_j| \geq \tau.n$), then they can find the response table for at least $\tau.n$ elements for each $j \in \{1, ..., \lambda\}$ with the same probability. So for $\tau.n$ out of n values of i they can find correct $sk_j = \frac{r_{i,j} - r'_{i,j} - (c_{i,j} - c'_{i,j})b_i}{c_{i,j} - c'_{i,j}}$ with probability $\geq poly(\lambda)$. Therefore by taking the majority, they can find the correct key bit with probability $\geq 1 - (1 - poly(\lambda))^{\tau.n}$.

Thus if the adversary succeeds in the first case with non-negligible probability, then they can find the secret key with considerably higher probability than random guessing and this contradicts the zero-knowledge property of ProProx. Therefore, the adversary's success chance will be negligible in this case.

In the second case, the adversary succeeds in the protocol by providing the correct response to $\mathcal{V}$ for at least $\tau.n$ correct queries out of n fast rounds for all key bits.

We noted that the learning phase of the adversary cannot provide information about the secret key $(\{sk_j\}_{j=1}^{\lambda})$ or the committed values $(\{a_{i,j}\}_{j=\{1...\lambda\}, i=\{1...n\}})$ as otherwise the zero-knowledge property of the protocol, or the commitment scheme will be violated, respectively.

In order to succeed in the protocol with non-negligible probability, the adversary must succeed in ZKP, for at least $\tau.n$ values of i, so they need to find at least $\tau.n$ valid tuples $\pi_i = (X,Y,Z)$ for random challenge bits such that $Z^2 = X(\theta^{b_i} y_j)^{c_{i,j}} \theta^{-Y}$ without having information about sk_j. For $\pi = [\pi_i]$ with size at least $\tau.n$ and $[c]$, $\Pr[\pi \textit{ is valid} | [c] \textit{ is random}] = \prod_{i=1}^{\gamma} \Pr[\pi_i \textit{ is valid} | [c] \textit{ is random}]$. So if $\Pr[\pi \textit{ is valid } | [c] \textit{ is random}] \geq negl$, then there is a value of i that $\Pr[\pi_i \textit{ is valid } | [c] \textit{ is random}] \geq \frac{1}{2} + poly(\lambda)$.

Since X is sent to the verifier before seeing $c_{i,j}$, therefore we have $\Pr[\textit{valid } (X,Y,Z) | c_{i,j} = 0] \geq \frac{1}{2} + poly(\lambda)$ and also $\Pr[\textit{valid } (X,Y',Z') | c_{i,j} = 1] \geq \frac{1}{2} + poly(\lambda)$. Since both tuples are valid, then we have $Z^2 = X\theta^{-Y}$ and $Z'^2 = X(\theta^{b_i} y_j)\theta^{-Y'}$. Therefore we have the following for $pk_j = \theta^{sk_j}(v_j)$;

$$(\frac{Z'}{Z})^2 = pk_j \theta^{b_i - Y' + Y} = \theta^{sk_j + b_i - Y' + Y}(v_j)^2$$

Therefore, the adversary can conclude $sk_j + b_i - Y' + Y \notin \{1,3\}$ for the known bits b_i, Y' and Y. So they gain some information about sk_j, which is in contradiction with *zero-knowledge* property of ProProx. □

7 Related Works

The main models and constructions of public key DB protocols are in [21], [3], [16], and [28]. In the following, we discuss and contrast the security model of these works to be able to put our new work in context.

[21] presented an informal model for *Distance-Fraud*, *Mafia-Fraud* and *Impersonation* attack and provided a secure protocol according to the model. [3] formally defined *Distance-Fraud*, *Mafia-Fraud*, *Impersonation*, *Terrorist-Fraud* and *Distance-Hijacking* attack. The *Distance-Fraud* adversary has a learning phase before the attack session and is therefore stronger than the definition in A2. During the learning phase, the adversary has access to the communications of the honest provers that are close-by. The security proofs of the proposed protocol have been deferred to the full version, which is not available yet.

[16] uses an informal model that captures *Distance-Fraud*, *Mafia-Fraud*, *Impersonation*, *Terrorist-Fraud*, *Distance-Hijacking* and a special type of attack, called *Slow-Impersonation* [14]. In their model, the definition of *Terrorist-Fraud* is slightly different from A4: a TF attack is successful if it allows the adversary to succeed in future Mafia-Fraud attacks.

For the first time in distance-bounding literature, [14] considered normal MiM attacking scenario where both the honest prover and the adversary are close to the verifier. The adversary interacts with the prover in order to succeed in a separate protocol session with the verifier. The adversary has to change some of the received messages in the slow phases of protocol in order to be considered successful. The attack is called *Slow-Impersonation* and is inspired by the basic MiM attack in authentication protocols. In

Slow-Impersonation, a close-by MiM actor that communicates with both verifier and close-by prover, tries to succeed in the protocol. During the slow-phase, the actor modifies the received messages from a party, and then sends it to the other party. Although the basic MiM attack is proper for DB models, it may not be strictly possible in one phase of the protocol as their action could influence or be influenced by other phases of the protocol.

A MiM adversary may, during the learning phase, only relay the slow-phase messages but, by manipulating the messages of the fast phase, learn the key information and later succeed in impersonation. According to the definitions in [16] and [14], the protocol is secure against *Slow-Impersonation*, however it is not secure against *Strong-Impersonation* (A3). This scenario shows that *Slow-Impersonation* does not necessarily capture *Impersonation* attacks in general. Moreover, it's hard to distinguish the success in slow phases of a protocol without considering the fast phase, as those phases have mutual influences on each other.

As an alternative definition, [2] proposed *Strong-Impersonation* (A3), in which the MiM adversary has an active learning phase that allows them to change the messages. *Strong-Impersonation* captures the MiM attack without the need to define success in the slow rounds. One of the incentives of *Strong-Impersonation* is capturing the case when the prover is close to the verifier, but is not participating in any instance of the protocol. In this case, any acceptance by the verifier means that the adversary has succeeded in impersonating an inactive prover.

In [28] an elegant formal model for public key distance-bounding protocols in terms of proof of proximity of knowledge has been proposed. The model captures *Distance-Fraud*, *Distance-Hijacking*, *Mafia-Fraud*, *Impersonation* and *Terrorist-Fraud*. In this approach, a public key DB protocol is a special type of proof of knowledge (proximity of knowledge): a protocol is considered sound if the acceptance of the verifier implies existence of an extractor algorithm that takes the view of all close-by participants and returns the prover's private key. This captures security against *Terrorist-Fraud* where a dishonest far-away prover must succeed without sharing their key with the close-by helper.

According to the soundness definition in [28] however, if the adversary succeeds while there is an inactive close-by prover, the protocol is sound because the verifier accepts, and there is an extractor for the key simply because there is an inactive close-by prover and their secret key is part of the extractor's view. Existence of an extractor is a demanding requirement for the success of attacks against authentication: obviously an adversary who can extract the key will succeed in the protocol, however it is possible to have an adversary who succeeds without extracting the key, but providing the required responses to the verifier. Our goal in introducing identification based model is to capture this weaker requirement of success in authentication, while providing a model that includes realistic attacks against DB protocols.

8 Concluding remarks

This paper is a revised and extended version of [2] which proposed a new formal model (DBID) for distance-bounding protocols, inline with the cryptographic identification protocols, that captures and strengthens the main attacks on public key distance-bounding protocols. This approach effectively included physical distance as an additional attribute of the prover in identification protocol.

In this paper we assume a stronger adversary that has access to a directional antenna. We showed this additional capability can break security of protocols that had been proven secure. To include this capability of the adversary, we needed to revise the definition of TF in [2] which resulted in proving a new security proof for the ProProx protocol. Other parts of model and security definition remained unchanged.

We also proposed Poxy, as a new DBID scheme, and provided the security proof. Poxy and ProProx use two different cryptosystems, and the fast phase of the Π protocol in Poxy and ProProx schemes have more computations compared to typical DB schemes. This is the price we pay for achieving TF-resistance. Our future work includes designing more efficient DBID protocols, and extending the model to include the anonymity of the prover against the verifier.

References

[1] Mamta Agiwal, Abhishek Roy, and Navrati Saxena. Next generation 5g wireless networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 2016.

[2] Ahmad Ahmadi and Reihaneh Safavi-Naini. Distance-bounding identification. In *Proceedings of the 3rd International Conference on Information Systems Security and Privacy - Volume 1: ICISSP,*, pages 202–212. INSTICC, SciTePress, 2017.

[3] Ahmad Ahmadi and Reyhaneh Safavi-Naini. Privacy-preserving distance-bounding proof-of-knowledge. In *16th ICICS*, 2014.

[4] Ahmad Ahmadi and Reyhaneh Safavi-Naini. Directional distance-bounding identification. In *Proc. of the 3rd Int. Conference on International Conference in Information Systems Security and Privacy (ICISSP)*. Springer, 2017.

[5] Gildas Avoine, Muhammed Ali Bingöl, Süleyman Kardaş, Cédric Lauradoux, and Benjamin Martin. A framework for analyzing RFID distance bounding protocols. *Journal of Computer Security*, 2011.

[6] Ioana Boureanu, Aikaterini Mitrokotsa, and Serge Vaudenay. Secure & lightweight distance-bounding. In *International Workshop on Lightweight Cryptography for Security and Privacy*, 2013.

[7] Stefan Brands and David Chaum. Distance-bounding protocols. In *Advances in Cryptology–EUROCRYPT'93*, pages 344–359. Springer, 1994.

[8] Laurent Bussard and Walid Bagga. Distance-bounding proof of knowledge protocols to avoid terrorist fraud attacks. Technical report, Technical report, Institut Eurecom, France, 2004.

[9] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23(4):493–507, 1952.

[10] Cas Cremers, Kasper Bonne Rasmussen, Benedikt Schmidt, and Srdjan Capkun. Distance hijacking attacks on distance bounding protocols. In *Security and Privacy*, 2012.

[11] Ivan Damgård. On Σ-protocols. *Lecture Notes, University of Aarhus, Department for Computer Science*, 2002.

[12] Yvo Desmedt. Major security problems with the ünforgeable(feige-)fiat-shamir proofs of identity and how to overcome them. In *Securicom'88*, 1988.

[13] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 1976.

[14] Ulrich Dürholz, Marc Fischlin, Michael Kasper, and Cristina Onete. A formal approach to distance-bounding rfid protocols. In *International Conference on Information Security*, pages 47–62. Springer, 2011.

[15] Aurélien Francillon, Boris Danev, and Srdjan Capkun. Relay attacks on passive keyless entry and start systems in modern cars. In *NDSS*, 2011.

[16] Sébastien Gambs, Marc-Olivier Killijian, Cédric Lauradoux, Cristina Onete, Matthieu Roy, and Moussa Traoré. Vssdb: A verifiable secret-sharing and distance-bounding protocol. In *International Conference on Cryptography and Information security*, 2014.

[17] Sébastien Gambs, Cristina Onete, and Jean-Marc Robert. Prover anonymous and deniable distance-bounding authentication. In *ASIA CCS '14*, 2014.

[18] Rosario Gennaro. Multi-trapdoor commitments and their applications to proofs of knowledge secure under concurrent man-in-the-middle attacks. In *Annual International Cryptology Conference*, 2004.

[19] Louis C Guillou and Jean-Jacques Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In *EUROCRYPT '88*, 1988.

[20] Jens Hermans, Andreas Pashalidis, Frederik Vercauteren, and Bart Preneel. A new RFID privacy model. In *Computer Security–ESORICS 2011*, pages 568–587. Springer, 2011.

[21] Jens Hermans, Roel Peeters, and Cristina Onete. Efficient, secure, private distance bounding without key updates. In *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks*, pages 207–218. ACM, 2013.

[22] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[23] Kaoru Kurosawa and Swee-Huay Heng. The power of identification schemes. In *Public Key Cryptography-PKC 2006*, pages 364–377. Springer, 2006.

[24] Torben Pryds Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology – CRYPTO'91*, pages 129–140. Springer, 1992.

[25] Kasper Bonne Rasmussen and Srdjan Capkun. Realization of rf distance bounding. In *USENIX Security Symposium*, pages 389–402, 2010.

[26] C P Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 1991.

[27] Serge Vaudenay. On modeling terrorist frauds. In *Provable Security*. Springer, 2013.

[28] Serge Vaudenay. Proof of proximity of knowledge. *IACR Eprint*, 695, 2014.

Appendix

Definition .1. (Authentication). *An authentication protocol is an interactive pair of protocols $(P(\zeta),V(z))$ of PPT algorithms operating on a language L and relation $R = \{(z,\zeta) : z \in L, \zeta \in W(z)\}$, where $W(z)$ is the set of all witnesses for z that should be accepted in authentication. This protocol has the following properties:*

- *complete: $\forall (z,\zeta) \in R$, we have $\Pr[Out_{\mathcal{V}} = 1 : P(\zeta) \leftrightarrow V(z)] = 1$.*
- *κ-sound: $\Pr[Out_{\mathcal{V}} = 1 : P^* \leftrightarrow V(z)] \leq \kappa$ in any of the following two cases; (i) $z \notin L$, (ii) $z \in L$ while algorithm P^* is independent from any $\zeta \in W(z)$. $\Pr[Out_{\mathcal{V}} = 1 : \mathcal{A}_2(View_{\mathcal{A}_1}) \leftrightarrow V(z)] \leq negl$.*

Definition .2. (Homomorphic Bit Commitment). *A homomorphic bit commitment function is a PPT algorithm Com operating on a multiplicative group G with parameter λ, that takes $b \in \mathbb{Z}_2$ and $\rho \in G$ as input, and returns $Com(b;\rho) \in G$. This function has the following properties:*

- *homomorphic: $\forall b,b' \in \mathbb{Z}_2$ and $\forall \rho,\rho' \in G$, we have $Com(b;\rho)Com(b';\rho') = Com(b+b';\rho\rho')$.*
- *perfect binding: $\forall b,b' \in \mathbb{Z}_2$ and $\forall \rho,\rho' \in G$, the equality $Com(b;\rho) = Com(b';\rho')$ implies $b = b'$.*
- *computational hiding: for a random $\rho \in_R G$, the distributions $Com(0,\rho)$ and $Com(1,\rho)$ are computationally indistinguishable.*

Definition .3. (One-way Function). *By considering λ as the security parameter, an efficiently computable function $OUT \leftarrow FUNC(IN)$, is one-way if there is no PPT algorithm that takes OUT as input and returns IN with non-negligible probability in terms of λ.*

Definition .4. (Zero-Knowledge Protocol). *A pair of protocols* $(P(\alpha), V(z))$ *is* ζ*-zero-knowledge for* $P(\alpha)$*, if for any PPT interactive machine* $V^*(z, aux)$ *there is a PPT simulator* $S(z, aux)$ *such that for any PPT distinguisher, any* $(\alpha : z) \in L$*, and any* $aux \in \{0,1\}^*$*, the distinguishing advantage between the final view of* V^**, in the interaction* $P(\alpha) \leftrightarrow V^*(z, aux)$*, and output of the simulator* $S(z, aux)$ *is bounded by* ζ*.*

Lemma 14. (Chernoff-Hoeffding Bound [9], [22]). *For any* $(\varepsilon, n, \tau, q)$*, we have the following inequalities about the function* $Tail(n, \tau, \rho) = \sum_{i=\tau}^{n} \binom{n}{i} \rho^i (1-\rho)^{n-i}$*;*

- *if* $\frac{\tau}{n} < q - \varepsilon$*, then* $Tail(n, \tau, q) > 1 - e^{-2\varepsilon^2 n}$
- *if* $\frac{\tau}{n} > q + \varepsilon$*, then* $Tail(n, \tau, q) < e^{-2\varepsilon^2 n}$