Distance-bounding Identification

Ahmad Ahmadi and Reihaneh Safavi-Naini
Department of Computer Science, University of Calgary, Calgary, Canada
{*ahmadi, rei*}*@ucalgary.ca*

Keywords: Distance-bounding, Identification, Public-key, MiM.

Abstract: Distance bounding (DB) protocols allow a prover to convince a verifier that they are within a distance bound. We propose a new approach to formalizing the security of DB protocols that we call distance-bounding identification (DBID), and is inspired by the security definition of cryptographic identification protocols. Our model provides a natural way of modeling the strongest man-in-the-middle attack, making security of DB protocols in line with identification protocols. We compare our model with other existing models, and give a construction that is secure in the proposed model.

1 INTRODUCTION

Distance bounding protocols were first proposed (Desmedt, 1988) as a mechanism for providing security against Man-in-the-middle (MiM) attacks in authentication protocols. In a MiM attack, the attacker does not have access to the secret key of the user, but exploits their communications to get accepted by the verifier. DB protocols use the distance between the prover and the verifier as a second factor in authentication. Distance bounding protocols have been widely studied in recent years and the security arguments have been formalized (Dürholz et al., 2011), (Boureanu et al., 2013). Most DB protocols are symmetric key protocols requiring the prover and the verifier to share a secret key. More recently public-key protocols –where the prover has a registered public-key– have been proposed. In this paper we consider public-key DB protocol.

Distance-bounding (DB) protocols are authentication protocols that consider the distance between the prover and the verifier as an extra factor, and guarantee that the prover knows (i) the correct secret and, (ii) their distance to the verifier is upper-bounded. The distance between the two parties is measured by running a fast challenge and response message exchange phase consisting of a sequence of single bit challenges sent by the verifier. The prover's response to each challenge bit depends on their secret key and the received bit. If the prover is located within a defined distance bound $\mathcal{D}$ to the verifier, then they are able to respond within a certain time period and their authentication claim will be accepted.

In a DB setting, the participants who are located within the distance $\mathcal{D}$ to the verifier, are called *close-by* participants, and those who are outside the range are called *far-away* participants. DB protocols have been studied under the following attacks. We use *prover* to refer to a participant who has a shared key with the verifier, and *adversary* for one who does not have a shared key.

(A1) *Distance-Fraud (Brands and Chaum, 1994);* a dishonest far-away prover tries to succeed in the protocol. In a sub-class of this attack, called *Distance-Hijacking (Cremers et al., 2012)*, a far-away prover takes advantage of honest close-by provers to succeed in the protocol.

(A2) *Mafia-Fraud (Desmedt, 1988);* an adversary who is close to the verifier tries to use the communications of a far-away honest prover, to succeed in the protocol.

(A3) *Impersonation (Avoine et al., 2011);* a close-by adversary tries to impersonate the prover in a new session. This original definition does not include a learning phase for the adversary.

(A4) *Strong-Impersonation (this paper);* a close-by adversary learns from past executions of the protocol by an honest prover and tries to impersonate the prover in a new execution when the prover is not close-by.

(A5) *Slow-Impersonation (Dürholz et al., 2011);* an adversary communicates simultaneously with the prover and the verifier, and tries to succeed in the protocol. The adversary will modify the protocol messages that are sent in the slow-

Ahmadi, A. and Safavi-Naini, R.
Distance-bounding Identification.
DOI: 10.5220/0006211102020212
In *Proceedings of the 3rd International Conference on Information Systems Security and Privacy (ICISSP 2017)*, pages 202-212
ISBN: 978-989-758-209-7

phase of the protocol but can relay the messages of the fast-phase.

(A6) *Terrorist-Fraud (Desmedt, 1988);* this is a colluding attack by a dishonest far-away prover and a close-by helper. The prover tries to succeed in the protocol with the requirement that their secret key will not be leaked to the helper.

DB protocols are commonly shown to be secure against a subset of these attacks. Shared-key DB protocols have application in the authentication of small devices such as RFID tags, while public-key DB protocols are used when more computation is affordable and one needs to protect the privacy of users against the verifier. In public-key DB model, a prover has a unique secret-key and the verifier has all the public-keys. Public-key DB protocols have the convenience of not needing to share a secret key but are significantly slower than shared-key DB protocols, and need more complex design to deal with individual bits of the private-key.

Related Works. The main models and constructions of public-key DB protocols are in (Hermans et al., 2013), (Ahmadi and Safavi-Naini, 2014), (Gambs et al., 2014), and (Vaudenay, 2014). In the following, we discuss and contrast the security model of these works to be able to put our new work in context.

(Hermans et al., 2013) presented an informal model for *Distance-Fraud*, *Mafia-Fraud* and *Impersonation* attack as defined above, and provided a secure protocol according to the model. (Ahmadi and Safavi-Naini, 2014) 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.

(Gambs et al., 2014) uses an informal model that captures *Distance-Fraud*, *Mafia-Fraud*, *Impersonation*, *Terrorist-Fraud*, *Distance-Hijacking* and a new type of attack called *Slow-Impersonation* that is defined in A5. In their model, the definition of *Terrorist-Fraud* is slightly different from A6: 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, (Dürholz et al., 2011) 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* (A5) and is inspired by the basic MiM attack in authentication protocols. 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 (Gambs et al., 2014) and (Dürholz et al., 2011), the protocol is secure against *Slow-Impersonation*, however it is not secure againt *Strong-Impersonation* (A4). 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, we propose *Strong-Impersonation* (A4), 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 (Vaudenay, 2014) 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 (Vaudenay, 2014) 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.

Our Work. We follow the formalization of cryptographic identification protocols due to (Kurosawa and Heng, 2006), and extend it to include the proximity of the prover as an extra required property. By replacing the framework of authentication protocols/proof-of-knowledge in (Vaudenay, 2014), with that of identification protocols, we can provide appropriate security definitions that realistically capture security of public-key DB protocols. A basic identification protocol, referred to as Σ-protocol, is a three round protocol between a prover and a verifier in which the prover convinces the verifier of their identity. Security is defined as a game between a challenger and an adversary, and the success of the adversary depends on its ability to provide a new valid protocol transcript, given their knowledge of the past transcripts. This is a weaker requirement for adversary's success in the sense that an adversary that can extract the secret key of the prover can construct a valid transcript, but constructing a transcript does not imply success in extracting the key. Using this approach we can model a MiM setting where an honest prover and the adversary are close-by, and the adversary uses the past communications of the prover, to launch an impersonation attack. Our soundness definition correctly captures this setting in the sense that success of the adversary in this setting implies violation of the soundness in our model. Our model of public-key DB protocols captures *Distance-Fraud*, *Mafia-Fraud*, *Strong-Impersonation* and *Terrorist-Fraud*. We provide a construction that is provably secure in this model.

The organization of the paper is as follows: Section2 is preliminaries; Section3 includes the model and definitions; Section4 presents a protocol and proves their security in our proposed model; Section5 concludes the paper.

2 PRELIMINARIES

In the following we recall a widely used definition of identification schemes.

Identification Scheme (Kurosawa and Heng, 2006). An identification scheme (ID), a prover $\mathcal{P}$ convinces a verifier $\mathcal{V}$ that they know a witness w_I related to a public value p_I. The scheme is given by the tuple ID=(KeyGen;Commit;Response; Check) which consists of a PPT algorithm KeyGen that generates (w_I, p_I), and the algorithms, Commit, Response and Check that specify an interactive protocol (P,V) between the prover $\mathcal{P}$ and the verifier $\mathcal{V}$ as follows. Let $\mathbb{CMD}$, $\mathbb{CMT}$, $\mathbb{CHA}$ and $\mathbb{RES}$ denote domains of the three variables denoting committed, commitment, challenge and response.

Step 1. P chooses $a \in_R \mathbb{CMD}$, computes the commitment $A = \texttt{Commit}(a) \in \mathbb{CMT}$ and sends to V.

Step 2. V chooses $c \in_R \mathbb{CHA}$ and sends it to P.

Step 3. P computes $r = \texttt{Response}(w_I, a, c) \in \mathbb{RES}$ and sends to V.

Step 4. V accepts if the following check holds;

$$A = \texttt{Check}(p_I, c, r), \tag{2.1}$$

The above protocol is called Σ-protocol. We say that $(A,c,r) \in \mathbb{CMT} \times \mathbb{CHA} \times \mathbb{RES}$ is a *valid transcript* for p_I if it satisfies Equation2.1. The main properties of ID-schemes are:

- **Completeness.** If a prover is honest, then Equation2.1 holds.
- **Soundness.** It is hard to compute two valid transcripts (A,c,r) and (A,c',r') such that $c \neq c'$ on input p_I.

This definition of soundness is against an adversary who observes transcripts of the protocol and will succeed if they can construct a new one. (Bellare and Palacio, 2002) however considers soundness as a game between the adversary, prover and the verifier, in two distinct steps.

- **Impersonation under Concurrent Attack.** The adversary $\mathcal{A} = (\mathcal{V}^*, \mathcal{P}^*)$ is a pair of randomized polynomial algorithms, called *cheating verifier* and *cheating prover*. Consider a game having two phases; In the first phase the key-pair of the prover (w_I, p_I) is generated and $\mathcal{V}^*(p_I)$ interacts concurrently with multiple clones of the prover $P(w_I, p_I)$ with independent randomness. In the second phase, the cheating prover $\mathcal{P}^*(View_{\mathcal{V}^*})$ that is initialized with the final view of $\mathcal{V}^*$, interacts with $V(p_I)$. A protocol is *imp-ca secure* if the acceptance probability of V is negligible.

In a stronger definition of MiM attack (Lyubashevsky and Masny, 2013), the attacker simultaneously interacts with the prover and the verifier.

- **MiM Resistance.** Consider an adversary who can interact with the prover and the verifier at the same time and run this interaction over multiple instances of the protocol. Upon receiving a message from a party, the adversary adds some value to the message and sends it to the other party, hence replacing the message A with $A+A'$, c with $c+c'$ and r with $r+r'$. Note that this captures the most general modification of such adversary, as they effectively change the received protocol message to anything they like. A protocol is MiM-resistant if the success chance of adversary in finding $(A',c',r') \neq (0,0,0)$, such that $(A+A',c,r+r')$ is a valid transcript, is negligible.

3 SECURITY MODEL AND DEFINITION

In our model, each participant has a *location* which is an element of a metric space $\mathcal{S}$ and stays the same during the protocol execution. There is a distance function $d(loc_1,loc_2)$ that returns the distance between any two locations. When a participant I_1 (located at loc_1), sends a message m at time t to another participant I_2 (located at loc_2), the message is received by I_2 at time $t+\frac{d(loc_1,loc_2)}{\mathcal{L}}$, where $\mathcal{L}$ indicates the speed of light. The message that is delivered to $\mathcal{P}_2$ is a noisy version of the sent message m, i.e. each bit of m may have been flipped with probability p_{noise}.

A message sent by I_1 may be seen and modified by other participants. A malicious participant I_3 located at loc_3, can send a malicious signal (message) at time t_0. This can only affect a message sent from I_1 to I_2 at time t, if $t+d(loc_1,loc_2) \geq t_0+d(loc_3,loc_2)$. If this happens, the message of I_1 may not be received by I_2. We allow I_1 to send multiple messages to multiple destinations at the same time, or receive multiple messages simultaneously. In the latter case, the recipients will receive a message that is the combination of all the messages that are received at the same time. If the signal strength of a message m_1 is much higher than that of a message m_2, then m_1 would stay recoverable and m_2 would appear as noise. This extends the communication model of (Vaudenay, 2014) to better capture the setting of many communicating devices.

In this system, we consider a set of *users* $U = \{u_1,\ldots,u_m\}$ that are registered and identifiable with their secret-key. We assume that a single user can be associated with multiple devices (called *provers*) of the user that share the same secret-key, but possibly have different locations.

We consider one protocol between the verifier and a user (with provers' of the user) at each time, and consider the following three types of *participants* in the system; a list of *provers* $\mathcal{P} = \{\mathcal{P}_1,\ldots,\mathcal{P}_k\}$ that share the secret-key of a single user (u_i), a *verifier* $\mathcal{V}$ who has access to the public parameters of system, and a set of *actors*. The *actors* set consists of all other users (i.e. their provers) and adversaries that may be active at the time. The difference between a malicious prover and a malicious actor is that the malicious prover has access to the secret-key that is needed for execution of the protocol, while a malicious actor does not.

When a prover of a user is compromised, then the user's key is compromised and the adversary can choose devices with that key at locations of their choice. We refer to these as corrupted provers who are controlled by the adversary. We assume provers that are not corrupted follow the protocol, and that a user uses only one of its devices at a time. Corrupted provers that are controlled by the adversary however can be invoked simultaneously.

3.1 Distance-bounding Identification

In this section we describe the setting, operations and properties of DBID protocols. The approach is to enhance Σ-protocol to include distance-bounding properties.

Definition 1. (Distance-bounding Identification). *Let λ denote the security parameter. A distance-bounding identification (*DBID*) is a tuple* $(\mathbb{K};\texttt{KeyGen};\texttt{Commit};\texttt{Response};\texttt{Check};\mathcal{D};p_{noise})$*, where $\mathbb{K}$ the key space whose size is determined by a trusted party;* KeyGen *that generate a pair (x,y) of public and private key; a two-party probabilistic polynomial-time (PPT) distance-bounding protocol $(P(x),V(y))$, where $P(x)$ is the prover algorithm and $V(y)$ is the verifier algorithm; and a distance bound $\mathcal{D}$. A trusted party runs the key generation algorithm, gives the private-key x to the user and the corresponding public-key to the verifier. To run an instance of the protocol, the following steps are taken.*

1. *P randomly chooses a from the set $\mathbb{CMD}$ and computes the commitment of that $A =$ $\texttt{Commit}(a)$. P then sends A to V. They use a reliable channel (using error correcting codes).*
2. *Fast challenge/response over noisy channel:*
   (a) *V randomly chooses a challenge c from the domain $\mathbb{CHA}^{\mathbb{F}}$ and sends it to P.*
   (b) *P computes the response $r =$ $\texttt{Response}(x,a,c)$ that is from a certain domain $\mathbb{RES}^{\mathbb{F}}$ and sends r to V.*

Steps 2-(a) and 2-(b) are repeated multiple times, over a physical channel that is noisy and may cause the challenge or response to flip (with a constant probability).

3. *In Step 2, let t_1 and t_2 denote the clock values at the verifier when the challenge c is sent, and the response r is received, respectively. V verifies if $t_2 - t_1 < \frac{\mathcal{D}}{\mathcal{L}}$, where $\mathcal{L}$ is the speed of light.*
4. *Slow challenge/response over noise-free channel:*
   (a) *V chooses a challenge c at random from a certain domain $\mathbb{CHA}^{\mathbb{S}}$ and sends it to P.*
   (b) *P computes the response $r = \texttt{Response}(x,a,c)$ that is in a certain domain $\mathbb{RES}^{\mathbb{S}}$ and sends r to V.*

   In this phase the communication is over reliable channel.
5. *V accepts only if the following holds;*

$$Accept = \texttt{Check}(y,[c],[r],A), \qquad (3.1)$$

Here $[c]$ and $[r]$ are the sequence of challenges and the sequence of corresponding responses, in the fast and the slow phases of the protocol. V outputs $Out_{\mathcal{V}} = 1$ for accept, or 0 for reject.

Note that in this paper, the notation [.] indicates a list of variables. In the rest of this section we will define the security properties. We start by defining an experiment that describes the operation of a distance-bounding identification protocol.

Definition 2. (DBID Experiment). *An experiment exp for a DBID scheme $(\mathbb{K};\texttt{KeyGen};\texttt{Commit};\texttt{Response};\texttt{Check};\mathcal{D};p_{noise})$ is defined by a set of participants and interactions between them. Participants are; a verifier $\mathcal{V}$, an ordered list of provers $\mathcal{P}$ that share the same secret-key (a user is a set of provers that share the same key), and other actors from a set $\mathcal{C}$ (including potential adversaries). Participants who are within the distance at most $\mathcal{D}$ from $\mathcal{V}$ are called close-by participants. Participants who are at a distance more than $\mathcal{D}$ to $\mathcal{V}$ are called far-away participants.*

If provers are honest, the secret-key $x \in \mathbb{K}$ of their associated user is randomly chosen and the public-key $y = \texttt{KeyGen}(x)$ is generated using a trusted process. Note that the values (x,y) are chosen before the experiment. x is given to members of $\mathcal{P}$ and the first prover is activated. Each active prover runs the defined algorithm, then stops and activates the next prover in the sequence of $\mathcal{P}$. All members of $\mathcal{P}$ run the algorithm $P(x)$, and this continues until all provers in $\mathcal{P}$ are finished.

If provers are malicious, the key pair (x,y) is set arbitrarily, and provers are allowed to be at any location and run any PPT algorithm concurrently.

The actors can run any PPT algorithms and can communicate with other participants simultaneously. The value of y is given to all participants. The verifier follows the algorithm $V(y)$ in an instance of the protocol with any participant, and returns $Out_{\mathcal{V}}$ at the end of that instance. The experiment finishes by returning the last output of verifier.

We define four distinct properties for public-key DB protocols using games between the adversary and a challenger.

Game Setting. The challenger assigns the keys of users and their provers. The adversary can corrupt a user, gain their secret-key, set the location of their provers and control those provers. The adversary can also corrupt some actors, set their locations and control them.

Property 1. (Completeness). Consider a DBID $(\mathbb{K};\texttt{KeyGen};\texttt{Commit};\texttt{Response};\texttt{Check};\mathcal{D};p_{noise})$ protocol between the verifier and a honest close-by prover, as defined in Definition1. Let p_{noise} denote the flip probability of noise in the environment.

The protocol is (τ,δ)-complete, $0 \leq \tau,\delta \leq 1$, if the verifier returns $Out_{\mathcal{V}} = 1$ with probability at least $1-\delta$, under the following assumption;

- for all bits of prover's secret, we have at least τ fraction of fast challenge/response rounds not affected by the noise, and
- $\tau < 1 - p_{noise} - \varepsilon$ for some constant $\varepsilon > 0$.

Completeness requires the absence of an adversary. A robust protocol must have negligible δ to be able to function in the presence of physical layer noise.

Next we consider a setting where the prover is honest, but there is a MiM adversary.

Property 2. (MiM-resistance). Let an adversary $\mathcal{A}$ select a set of honest provers, corrupt a set of actors (both sets of polynomial size in the security parameter), and set their locations, and runs the DBID experiment in Definition2. The corrupted actors are controlled by the adversary, and can communicate with provers and $\mathcal{V}$ at the same time. i.e. $\mathcal{A}$ can receive a message m from $\mathcal{P}$ and send m' to $\mathcal{V}$, and vice versa. A protocol is γ-MiM-resistant if for any such experiment, the probability of verifier outputting $Out_{\mathcal{V}} = 1$, while there is no active close-by prover during the session is limited by γ.

This general definition captures relay attack (Brands and Chaum, 1994), mafia-fraud (Desmedt, 1988) and impersonation attack (Avoine et al., 2011). In addition, this definition captures the case that there are some close-by provers, but they are inactive. In this case, if there is a successful protocol session, then the protocol is not considered as MiM-resistant.

We consider two types of attacks by a dishonest prover: a stand alone dishonest prover (Property3), and a dishonest prover with a helper (Property4).

Property 3. (Distance-Fraud). Let the adversary $\mathcal{A}$ corrupt the provers and set their locations far-away from the verifier. The experiment is run after this setting.

A protocol is called α-DF-resistant if, for any DBID experiment *exp*, where all participants (except $\mathcal{V}$) are far-away from $\mathcal{V}$, we have $\Pr[Out_{\mathcal{V}} = 1] \leq \alpha$.

In the following we define the soundness of public-key DB protocols that captures security against *Terrorist-Fraud*.

Definition 3. Let O be an oracle that takes the key pair of a user and the locations of participants and actors, simulates a DBID experiment as in Definition2, and supplies the view of close-by participants, including the verifier, to the caller algorithm.

A transcript sampler algorithm is an algorithm that calls O once and generates a valid transcript $\sigma = (A, [c], [r])$. A covering sampler algorithm $\mathcal{S}$ calls O once to generate a set of valid transcripts $\Sigma = \{\sigma_1, ..., \sigma_m\}$ such that, (i) all transcripts in Σ have the same commitment value A, (ii) for each round of the fast challenge/response phase (Step 2) there is a pair of transcripts $(\sigma', \sigma'') \in \Sigma$ such that the challenge values of the two transcripts of that round are different (complement), and (iii) there are at least two transcripts $(\sigma', \sigma'') \in \Sigma$ that have different challenge values for the slow challenge/response phase (Step 3).

A response sampler algorithm $\mathcal{R}$ is an algorithm that calls O once and is able to generate a valid transcript $\sigma = (A', [c'], [r'])$ for any value of $[c']$.

Property 4. (Soundness). Consider an adversary who corrupts provers and actors, and sets the location of actors and provers, close-by and far-away respectively. For a key assignment of the prover, and location set of provers and actors with the above restrictions, a protocol is sound if the following holds:

If there is a covering sampler $\mathcal{S}$ as defined in Definition3 that is successful with non-negligible probability, then there is a PPT response sampler $\mathcal{R}$ as defined in Definition3 that is successful with non-negligible probability.

In the rest of this section we show that *terrorist-fraud* (A6) is captured by the *soundness* (Property4). First we rephrase the definition of terrorist-fraud resistance.

Definition 4. (TF-resistance). Consider an experiment in which provers are far-away from the verifier while the actors are close-by and all are controlled by the adversary. A DB protocol is *TF-resistant* if the following statement holds:

If there is a TF adversary that succeeds with non-negligible probability, then there exists a PPT close-by adversary that takes the view of all close-by actors as input and is able to impersonate the prover with non-negligible probability.

Lemma 1. A DBID protocol that is sound according to Property4, is *TF-resistant* according to Definition4.

Proof. Consider the following diagram that starts with a successful TF attack and ends with a successful impersonation.

$$\boxed{Successful\ TF} \xrightarrow{(i)} \boxed{\mathcal{S}} \xrightarrow{(ii)} \boxed{\mathcal{R}} \xrightarrow{(iii)} \boxed{Impersonation}$$

Our soundness definition of DBID protocols imply that this diagram holds for the protocol. We will show that (i) and (iii) hold for any DBID protocol, and so if the protocol is sound, then the link in (ii) holds, and the diagram is complete.

To show (iii), we note that for a DBID protocol, if there is a response sampler $\mathcal{R}$ that takes the view of close-by participants, and returns a valid transcript for any challenge sequence (concatenated fast and slow challenges), then $\mathcal{R}$ can be used to construct a successful impersonation attacker. This immediately follows from the definition of a successful impersonation attacker and $\mathcal{R}$: the impersonation attacker runs a learning phase during which it plays the role of the verifier for a close-by prover, and provides the view of the close-by participants to $\mathcal{R}$. In the attack phase, the impersonation attacker will provide the commitment of the close-by prover in the learning phase, to the verifier, and use $\mathcal{R}$ to provide responses for the challenges of the verifier. The success of the impersonation attacker follows because $\mathcal{R}$ is able to generate correct response to any challenge sequence.

To show (i) we need to show that if there is a successful TF attacker (Definition4) then there is a covering sampler. Suppose there is a TF adversary $\mathcal{A}$ that succeeds with non-negligible probability p_1. The challenge sequence $[c]$ is chosen randomly by the honest verifier after they receive the commitment A. During the fast phase, the response to the verifier

cannot depend on the response in the same round, of the far-away prover as such responses will be dropped because they are outside the acceptable time-interval (associated with the bound). The view of the close-by participants however can be used to construct a valid transcript with non-negligible probability p_1. This means that there is a sampler algorithm $\mathcal{J}$ that uses this view and succeeds with non-negligible probability p_1 in constructing a valid transcript. $\mathcal{J}$ can simulate the behavior of the honest verifier, which is generating the challenge bits randomly. We divide $\mathcal{J}$ into two sub-algorithms $(\mathcal{J}_1, \mathcal{J}_2)$ based on time. $\mathcal{J}_1$ is the sub-algorithm from the beginning of $\mathcal{J}$ up to the generation of commitment A, and $\mathcal{J}_2$ is the sub-algorithm from after generation of A up to the end of $\mathcal{J}$.

To construct a covering sampler $\mathcal{S}$ we do the following: run $\mathcal{J}_1$ once, followed by ℓ times of $\mathcal{J}_2$. Before running $\mathcal{J}_2$ at any time, we rewind the memory state of the algorithm to the end of $\mathcal{J}_1$. Let n denote the number of challenge rounds in the fast exchange round, and for simplicity assume binary challenges. The following calculation shows that the probability of constructing a covering sampler approaches 1 for any n and p_1, if ℓ is chosen sufficiently large.

Suppose the covering sampler algorithm $\mathcal{S}$ has generated ℓ transcripts $\Sigma = \{\sigma_1, \sigma_2, ..., \sigma_\ell\}$, each with n challenge bits. Each transcript is valid with probability p_1. For a bit $1 \leq i \leq n$, consider the list of ℓ bits $\hat{b^i} = (b^i_1, b^i_2, ...b^i_\ell)$ belonging to $\sigma_1, \sigma_2 \cdots \sigma_\ell$ respectively. The bit is bad, if no pair of transcripts in Σ have complementary values in this position. This only happens if all the bits of $\hat{b^i}$ are 0, or all are 1. So the probability that the bit is good is, assuming all bits belong to valid transcripts, is $1 - 2^{-\ell+1}$. Note that out of the ℓ bits, $\ell \times p_1$ bits belong to valid transcripts and so probability that the bit is good is, is $1 - 2^{-p_1\ell+1}$. This means the probability that all n bits are covered is, $(1 - 2^{-p_1\ell+1})^n \geq 1 - n \times 2^{-p_1\ell+1}$. Since p_1 is constant, and ℓ and n are polynomial in the security parameter, then for any n, ℓ can be chosen such that the success chance of constructing the covering sampler is arbitrarily close to 1. This implies that we can build a covering sampler $\mathcal{S}$ that follows the success chance the TF attack (i.e. p_1). □

4 PROPROX PROTOCOL

In this section we introduce ProProx (Figure1) public-key DB protocol (Vaudenay, 2014) and show that the protocol fits in DBID model. We also prove security of the protocol in this model.

ProProx has λ as the secret the size, n as the number of rounds that is linear in λ, and t as the minimum ratio of number of noiseless rounds to all rounds. The vector $b \in Z_2^n$ has Hamming weight $\lfloor \frac{n}{2} \rfloor$ and is fixed.

The j^{th} bit of public-key is $y_j = \texttt{KeyGen}(x_j) = Com(x_j; H(x, j))$ where $Com(b; \rho) = \theta^b \rho^2 (\text{mod } N)$ for a bit b, and the random value ρ is given by Goldwasser-Micali cryptosystem (Goldwasser and Micali, 1984). For a fixed θ that is a residue modulo $N = P.Q$ such that $(\theta/P) = (\theta/Q) = -1$, for example $\theta = -1$, the algorithm Com is a homomorphic commitment scheme that provides computational hiding in the sense of zero-knowledgeness. More details can be found in (Vaudenay, 2014).

In the verification phase, the prover and the verifier determine a set I_j of $\tau.n$ round indices that they believe has not been affected by noise. The verifier checks whether I_j has cardinality $\tau.n$ and responses are within the required time interval. The prover and the verifier then run an interactive zero-knowledge proof (ZKP_κ) to show that the responses $r_{i,j}$'s are consistent with $A_{i,j}$'s and y_j's. If the verification fails, the verifier aborts the protocol by sending $Out_\mathcal{V} = 0$. Otherwise, they send $Out_\mathcal{V} = 1$.

Theorem 1. ProProx *is* (τ, δ)*-complete, sound,* γ*-MiM-resistant and* α*-DF-resistant* DBID *protocol for negligible values of* δ*,* γ *and* α*, when* $n.(1 - p_{noise-}\varepsilon) > n.\tau \geq n - (\frac{1}{2} - 2\varepsilon)\lceil \frac{n}{2} \rceil$ *for some constant* $\varepsilon > 0$.

ProProx is shown to be *distance-fraud resistant* and *zero-knowledge* (Definition8) in (Vaudenay, 2014). Our definitions of those properties remain unchanged and so we only need to prove *completeness*, *soundness* and *MiM-resistance* properties of DBID model about ProProx protocol.

Lemma 2. (Completeness). ProProx *is a complete* DBID *protocol.*

Proof. Let's assume the verifier sends challenge sequence $C = ([c], [\bar{c}])$, where $[c] = [[c^1]...[c^\lambda]]$, $[c^j] = [c^j_1 ... c^j_n]$ for all $j \in \{1, ..., \lambda\}$, and $[\bar{c}] = [\bar{c}_1 ... \bar{c}_m]$ for some $m \geq 0$, while the prover receives $C' = ([c'], [\bar{c}])$ such that $\Pr[c^j_i = c'^j_i] = 1 - p_{noise}$ for all $j \in \{1, ..., \lambda\}$ and $i \in \{1, ..., n\}$. Correspondingly, the prover sends the response sequence $R = ([r], [\bar{r}])$ and the verifier receives $R' = ([r'], [\bar{r}])$ such that $\Pr[r^j_i = r'^j_i] = 1 - p_{noise}$ for all $j \in \{1, ..., \lambda\}$ and $i \in \{1, ..., n\}$.

The verifier is honest and for an honest prover, they are able to find and agree on the set of uncorrupted (noiseless) messages and so the protocol succeeds if for all values of $j \in \{1, ..., \lambda\}$ there exist at least $\tau.n$ noiseless challenge/response rounds. The failure chance of the protocol is not more than the chance of having at least one $j \in \{1, ..., \lambda\}$ that has less than $\tau.n$ noiseless challenge/response rounds.

The probability of having at least $\tau.n$ noiseless

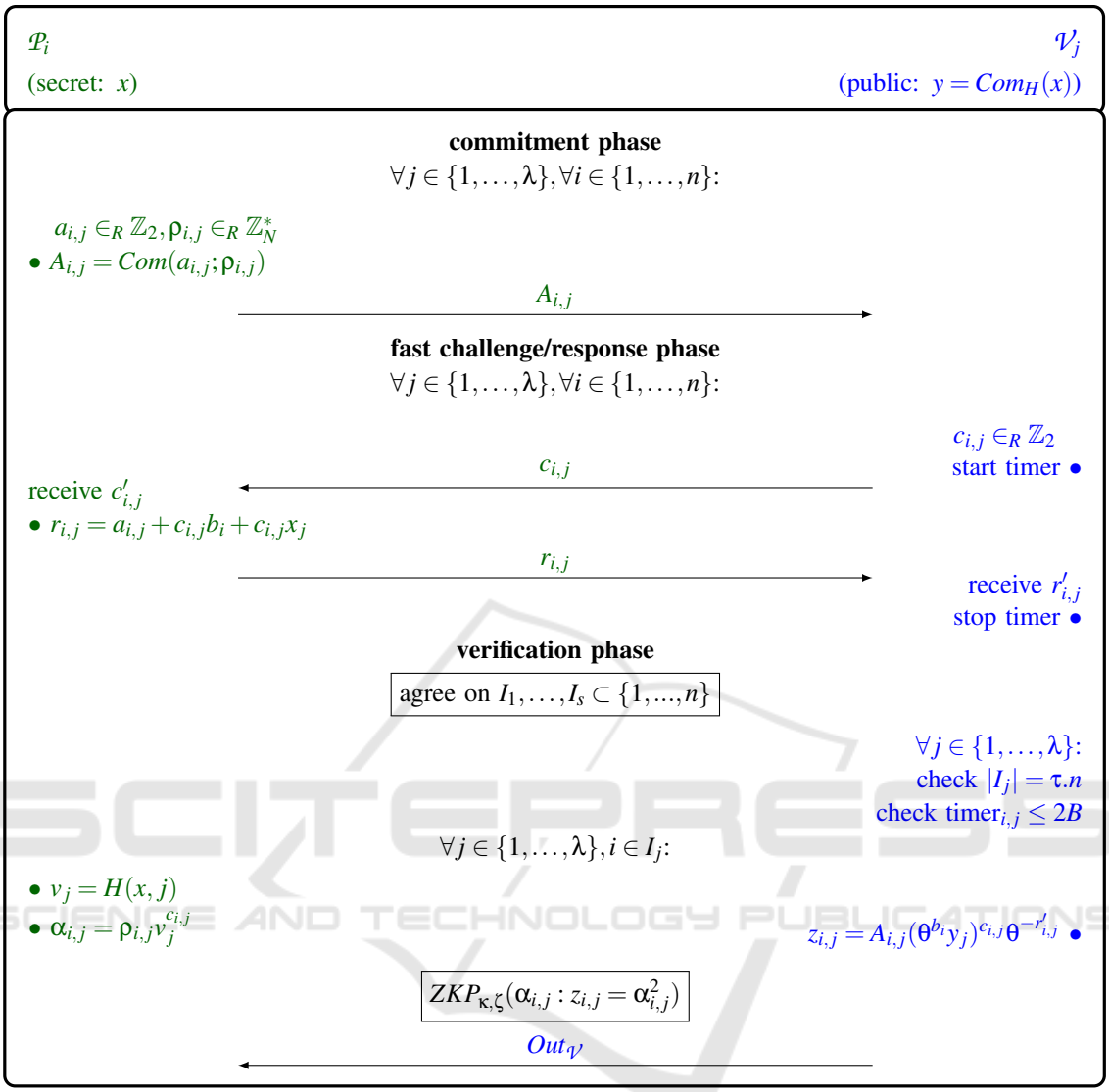

Figure 1: ProProx Protocol.

challenge/response rounds for a single $j \in \{1,...,\lambda\}$ is equal to $Tail(n, \tau.n, \rho) = \sum_{i=\tau.n}^{n} \binom{n}{i} (\rho)^i (1-\rho)^{n-i}$ for $\rho = 1 - p_{noise}$. As a result, the failure chance of the protocol is limited by $\lambda.(1 - Tail(n, \tau.n, 1 - p_{noise}))$.

Based on Chernoff bound (Lemma5) we know that $1 - Tail(n, \tau.n, 1 - p_{noise}) < e^{-2\varepsilon^2 n}$. Therefore, the failure chance of the protocol is limited by $\lambda.e^{-2\varepsilon^2 n}$, which is negligible. □

Lemma 3. (Soundness). ProProx *is a sound* DBID *protocol.*

Proof. The challenger sets the secret of the provers. The adversary sets the location of provers and actors before start of the experiment and possibly corrupts provers that are outside the bound.

According to the soundness definition (Property4), we need to show that if there is a PPT covering sampler ($\mathcal{S}$) that can create a covering set of valid transcripts using the view of close-by participants, then there exists a response sampler for the protocol. In the following we will achieve this by first proving a stronger property: we will show that existence of $\mathcal{S}$ implies existence of a PPT extractor ($\mathcal{E}$) that uses $\mathcal{S}$ to find an x' such that $\mathtt{KeyGen}(x') = y$. That is the extractor algorithm $\mathcal{J}$ will use $\mathcal{S}$ as a subroutine to extract x'.

In ProProx the relation between each challenge and response bit in the fast phase is $r_{i,j} = a_{i,j} + c_{i,j} b_i + c_{i,j} x_j$. Since $\mathcal{S}$ can find a covering set $\sigma = \{\sigma_1, ..., \sigma_m\}$

that for any $j \in \{1,...,\lambda\}$, there is at least one pair of transcripts $(\sigma',\sigma'') \in \sigma$ that $c'_{i,j} \neq c''_{i,j}$, where $c'_{i,j}$ belongs to σ' and $c''_{i,j}$ belongs to σ''. Since the committed values ($a_{i,j}$) are fixed in the two transcripts and the value of b_i is public and fixed in the protocol, then we have $r'_{i,j} - r''_{i,j} = a_{i,j} + c'_{i,j}b_i + c'_{i,j}x_j - (a_{i,j} + c''_{i,j}b_i + c''_{i,j}x_j) = (c'_{i,j} - c''_{i,j})b_i + (c'_{i,j} - c''_{i,j})x_j$. So for all $j \in \{1,...\lambda\}$, the value of x_j can be computed as $x_j = \frac{r'_{i,j} - r''_{i,j} - (c'_{i,j} - c''_{i,j})b_i}{c'_{i,j} - c''_{i,j}}$. By using the extracted key x in honest prover algorithm $P(x)$, we build the response sampler of Property4. □

Lemma 4. (MiM-resistance). ProProx *is a* γ*-MiM-resistant* DBID *(Prop2) protocol 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 Definition6; ZKP$_\kappa$ is a* κ*-sound (Definition5) and* ζ*-zero-knowledge authentication (Definition8) for negligible* κ *and* ζ*.*

Proof. The adversary $\mathcal{A}$ sets the locations of provers and actors, and can control the actors who can communicate with $\mathcal{P}$ and $\mathcal{V}$ simultaneously. $\mathcal{P}$ and $\mathcal{V}$ are assumed to behave honestly. The $\mathcal{P}$'s input in an experiment consist of challenge values (i.e. $\{c_{i,j}\}$ bits and challenge values of ZKP$_\kappa$), that can take arbitrary values, and will be responded by the prover only once. Because of zero-knowledge property of ProProx against any PPT verifier (that can be malicious), the learning phase of the MiM adversary cannot provide any information to the adversary as otherwise there will be a malicious verifier that will violate *negl*(λ)*-zero-knowledge* property of the protocol.

There are two possible participant arrangements for winning conditions of a MiM adversary that result in the verifier accepting a DBID instance: (i) all active provers are far-away from the verifier, (ii) there is no active prover during this session. In the following we show that the success probability of the adversary in both cases 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 $x_j = \frac{r_{i,j} - r_{\bar{i},j} - (c_{i,j} - c_{\bar{i},j})b_i}{c_{i,j} - c_{\bar{i},j}}$ 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 ($\{x_j\}$) or the committed values ($\{a_{i,j}\}$) 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$_\kappa$, for at least $\tau.n$ values of i, so they need to find at least $\tau.n$ valid tuples $\pi_i = (X \in G, Y \in \{0,1\}, Z \in \mathbb{Z}_N^*)$ for random challenge bits such that $Z^2 = X(\theta^{b_i} y_j)^{c_{i,j}} \theta^{-Y}$ without having information about x_j. For $\pi = [\pi_i]$, $|\pi| \geq \tau.n$ and *chg* : *challenge variables*, $\Pr[valid\ \pi |\ random\ chg] = \prod_{i=1}^{\gamma} \Pr[valid\ \pi_i |\ random\ chg]$. So if $\Pr[\pi\ is\ valid \mid random\ chg] \geq negl(\lambda)$, then there is a value of i that $\Pr[\pi_i\ is\ valid \mid random\ chg] \geq \frac{1}{2} + poly(\lambda)$. Since X is sent to the verifier before seeing $c_{i,j}$, therefore we have $\Pr[valid\ (X,Y,Z) | c_{i,j} = 0] \geq \frac{1}{2} + poly(\lambda)$ and also $\Pr[valid\ (X,Y',Z') | c_{i,j} = 1] \geq \frac{1}{2} + poly(\lambda)$. 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 $y_j = \theta^{x_j}(v_j)$;

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

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

5 CONCLUSION

We motivated and 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 effectively includes physical distance as an additional attribute of the prover in identification protocol.

We motivated our work by examining the proof of proximity of knowledge that incorporates the prover's distance as an extra attribute in proof of knowledge systems, and showed that the soundness definition cannot correctly capture the situation that the prover is close-by but inactive. The DBID framework however correctly captures the soundness in this case. We also showed the ProProx protocol fits our model and proved its security in this model. Our future work includes designing more efficient DBID protocols, and extending the model to include the anonymity of the prover against the verifier.

REFERENCES

Ahmadi, A. and Safavi-Naini, R. (2014). Privacy-preserving distance-bounding proof-of-knowledge. In *16th International Conference on Information and Communications Security*.

Avoine, G., Bingöl, M. A., Kardaş, S., Lauradoux, C., and Martin, B. (2011). A framework for analyzing RFID distance bounding protocols. *Journal of Computer Security*, pages 289–317.

Bellare, M. and Palacio, A. (2002). Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In *Annual International Cryptology Conference*, pages 162–177. Springer.

Boureanu, I., Mitrokotsa, A., and Vaudenay, S. (2013). Practical & provably secure distance-bounding. In *The 16th Information Security Conference*.

Brands, S. and Chaum, D. (1994). Distance-bounding protocols. In *Advances in Cryptology–EUROCRYPT'93*, pages 344–359. Springer.

Chernoff, H. (1952). 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.

Cremers, C., Rasmussen, K. B., Schmidt, B., and Capkun, S. (2012). Distance hijacking attacks on distance bounding protocols. In *Security and Privacy*, pages 113–127.

Desmedt, Y. (1988). Major security problems with the ünforgeable(feige-)fiat-shamir proofs of identity and how to overcome them. In *Congress on Computer and Communication Security and Protection Securicom'88*, pages 147–159.

Dürholz, U., Fischlin, M., Kasper, M., and Onete, C. (2011). A formal approach to distance-bounding RFID protocols. In *Information Security*, pages 47–62. Springer.

Gambs, S., Killijian, M.-O., Lauradoux, C., Onete, C., Roy, M., and Traoré, M. (2014). Vssdb: A verifiable secret-sharing and distance-bounding protocol. In *International Conference on Cryptography and Information security (BalkanCryptSec'14)*.

Goldwasser, S. and Micali, S. (1984). Probabilistic encryption. *Journal of computer and system sciences*, 28(2):270–299.

Hermans, J., Peeters, R., and Onete, C. (2013). 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.

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

Kurosawa, K. and Heng, S.-H. (2006). The power of identification schemes. In *Public Key Cryptography-PKC 2006*, pages 364–377. Springer.

Lyubashevsky, V. and Masny, D. (2013). Man-in-the-middle secure authentication schemes from lpn and weak prfs. In *Advances in Cryptology – CRYPTO'13*, pages 308–325. Springer.

Vaudenay, S. (2014). Proof of proximity of knowledge. *IACR Eprint*, 695.

APPENDIX

Definition 5. (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 6. (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 7. (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 8. (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 5. (Chernoff-Hoeffding Bound (Chernoff, 1952), (Hoeffding, 1963)). *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 $Tail(n, \tau, q) < e^{-2\varepsilon^2 n}$*