Prover Anonymous and Deniable Distance-Bounding Authentication[*]

Sébastien Gambs[1], Cristina Onete[1], and Jean-Marc Robert[2]

[1]*IRISA/INRIA, Univ. Rennes 1*
[2]*ETS Montréal*

Abstract

In distance-bounding authentication protocols, a verifier confirms that a prover is (1) legitimate and (2) in the verifier's proximity. Proximity checking is done by running time-critical exchanges between both parties. This enables the verifier to detect relay attacks (a.k.a. mafia frauds). While most distance-bounding protocols offer resistance to mafia and distance frauds as well as to impersonation attacks, only few protect the privacy of the authenticating prover.

One exception is the protocol due to Hermans, Peeters, and Onete developed in 2013, which offers strong privacy guarantees with respect to a Man-in-the-Middle adversary. However, this protocol provides no privacy guarantees for the prover with respect to a malicious verifier, who can fully identify the prover. Having in mind possible verifier corruption or data leakage from verifiers to a centralized server, we suggest that stronger privacy properties are needed.

In this paper, we propose an efficient distance-bounding protocol that gives strong prover privacy guarantees even with respect to the verifier or to a centralized back-end server, storing prover information and managing revocation and registration. Specifically, we formally model and define *prover anonymity*, a property guaranteeing that verifiers infer only the legitimacy of the prover but not his identity, and *deniability*, which ensures that the back-end server cannot distinguish prover behavior from malicious verifier behavior (*i.e.*, provers can deny that they authenticated). Finally, we present an efficient protocol that achieves these strong guarantees, give exact bounds for each of its security properties, and prove these statements formally.

1 Introduction

Distance-bounding protocols [BC93] were originally designed to counter *relay attacks* [DGB88] in authentication protocols. In these attacks, an adversary uses two proxies to relay messages between two devices (*i.e.,* the *prover* and the *verifier*). The prover is typically a mobile device or an identification RFID tag. The verifier is usually a reader or a terminal equipped with an internal clock and is either offline or connected to a back-end server. Repeatedly measuring the time span of some round trips of lightweight communication allows the verifier to check that the prover is within his vicinity. Indeed, only the prover should be able to appropriately respond to the challenges sent by the verifier during these rounds. Thus, distance-bounding protocols provide two

[*]A short version of this paper was accepted at AsiaCCS 2014.

functionalities: prover-to-verifier authentication and a prover proximity-check for the verifier.

While the seminal paper on distance bounding by Brands and Chaum [BC93] did not focus on the RFID setting, due to the lightweight nature of the distance-approximation phase, distance-bounding protocols have been widely adopted in this community. Most RFID distance-bounding protocols employ symmetric cryptography (*i.e.*, shared secrets between legitimate provers and the verifier) for efficiency reasons. Unfortunately, most of these protocols do not (strongly) protect the privacy of the provers against a Man-In-the-Middle (MIM) adversary and *no* protocol does so against the verifier, who can directly identify and trace a prover through his secret. In addition, most papers assume that there exists a list of trusted verifiers that is always accessible to the provers. The fact that these verifiers are trusted (*i.e.*, uncorrupted) and will not trace provers is a very strong assumption. Nevertheless, even if uncorrupted verifiers might not trace provers, it is sometimes easier to gain access to the (static) verifier – and his data – than to the (mobile) prover. This threat to privacy is magnified if provers can be tracked by a centralized back-end server storing their credentials.

Besides the original protocol of Brands and Chaum, which uses public-key digital signatures, other distance-bounding protocols in the literature rely on asymmetric primitives, such as the constructions due to Reid *et al.* [RNTS07], Bussard and Bagga [BB05] and the more recent one of Hermans, Peeters, and Onete [HPO13] (which we thereafter call the HPO protocol). Using public-key primitives is more promising to obtain good privacy properties. Indeed, Vaudenay [Vau07] and Paise and Vaudenay [PV08] showed that it is impossible to achieve strong privacy guarantees by relying only on symmetric primitives. Thus, we ask the following fundamental question, that has not (to our knowledge) been so far treated in the literature.

- Can we design an efficient distance-bounding protocol providing strong security guarantees, while preserving the prover privacy even against verifiers?

In this paper, we give a positive answer to this question by proposing an efficient distance-bounding protocol, in which the prover is authenticated in a fully-private and secure manner, such that even a MIM adversary controlling a verifier cannot trace the prover. Our construction is a proof-of-concept that a protocol with such strong security and privacy properties *can* be designed, and is targeted for computationally stronger mobile devices than common RFID tags. In a nutshell, our scheme is an extension of the HPO protocol [HPO13], which already provided very strong privacy guarantees (*i.e.*, *narrow-strong* and *wide-insider-forward* privacy – as defined by Vaudenay [Vau07]) with respect to a MIM adversary. In fact, our construction achieves better privacy guarantees (*i.e.*, *wide-strong* privacy) both against a MIM adversary and against an adversary that has access to the verifier's full internal state. This protocol also provides the very strong property of *prover deniability* with respect to a back-end server in possession of all the provers' credentials. These last two requirements are stronger than the typical ones used for payment systems [Vau15], in which a fraudulent payer should not be able to deny a payment. In the other hand, these requirements may be desirable for a lot of applications in which strong privacy properties for users are required, such as privacy-preserving public transportation system or private proximity-checking.

More precisely, our contributions are threefold. First after having introduced the preliminary notions in Section 2, in Section 3 we define the property of prover anonymity, extending the (RFID) privacy model of Hermans *et al.* [HPVP11]. Our adversary runs in three modes, depending on whether she personifies a MIM adversary, an honest-but-curious verifier, or a malicious verifier. Second, in Section 4.2 we propose a distance-bounding protocol building on the HPO protocol (outlined in Section 4.1), in a setting in which an entity called the back-end server, handling prover creation and revocation, provides an infrastructure enabling provers to authenticate simply as a member of the set of legitimate users. Finally, in Section 4.3 we prove that our construction inherits the near-optimal mafia-fraud and the distance-fraud resistances (as outlined in Section 2), and the impersonation security of the HPO protocol, has similar soundness and correctness guarantees, but provides better privacy properties, even with respect to the verifier and to the back-end server. Specifically, our scheme is wide-strong prover anonymous, and deniable with respect to the back-end server.

2 Preliminaries

In this section, we describe the system model, the problem statement, as well as the security and privacy requirements.

2.1 System Model

We consider a (distance-bounding) authentication system with a single verifier V, a set of provers $\mathcal{P} = \{\mathsf{P}_1, \ldots, \mathsf{P}_k\}$ and a back-end server Srv.

The *server* Srv is offline most of the time and takes care of prover creation and revocation. At every update of $\mathcal{P}$, Srv updates the authentication information available on a public board $\mathcal{B}$. We associate Srv with one-time private/public key pair (z, Z), used with an unforgeable signature scheme $\mathcal{S} = (\mathsf{SS.KGen}, \mathsf{Sign}, \mathsf{Vfy})$, and we assume that the server can self-certify these keys.

Each *prover* P_i has a key pair (x_i, X_i) for authentication. The public part X_i is stored on $\mathcal{B}$ and updated by Srv. In contrast, the private[1] part x_i is known only by the prover and Srv. In our security and privacy models, we give the adversary the possibility to register *insider* provers, who share their private keys with the adversary.

The *verifier* has a private/public key pair (y, Y) used for authentication (*e.g.*, a static Diffie-Hellman key pair) and may also store auxiliary information in his internal state.

The *adversary* has access to (1) the public information contained on the board $\mathcal{B}$, (2) the private information of the insider provers, and (3) for our prover-anonymity and deniability models, to the full internal state of the verifier. The attacker can (1) change the information on $\mathcal{B}$, (2) corrupt provers (thus learning their non-volatile internal state), or (3) act as a MIM between provers and verifiers (this includes strictly interacting with one party, or relaying/modifying messages between parties). Possible objectives of the adversary could be to break the correctness of the protocol (*e.g.*, causing a Denial-of-Service or DoS), to attack the security of the distance-bounding authentication (*e.g.*, by committing mafia or distance fraud, slow-phase prover impersonation or impersonation in the presence of prover corruption, called soundness), or to break the privacy of the prover (*e.g.*, prover anonymity or deniability).

[1]In this paper, we use interchangeably the terms “private” and “secret” when referring to a key that is not public.

2.2 Problem Statement

Concretely, we are aiming for a protocol solving the following problems.

Problem 1 (Authentication problem)**.** *Prevent an illegitimate party from authenticating to an honest verifier, while ensuring that legitimate parties always authenticate.*

Problem 2 (Distance-Bounding problem)**.** *Prevent mafia and distance frauds, as well as impersonation attacks.*

Problem 3 (Privacy problem)**.** *Prevent MIM adversaries, (honest-but-curious or malicious) verifiers, and even the back-end server from linking two prover authentication sessions.*

These problems are addressed in the literature in slightly different settings. Specifically, the formal model of distance-bounding security presented by Dürholz *et al.* [DFKO11] defines mafia, distance, and (simulation-based) terrorist-fraud resistance, as well as lazy-phase impersonation security *for a single prover and a single verifier.* This model is used by many protocols [DFKO11, FO13a, HPO13, FO13b] as it is sufficiently formal to give precise security bounds. In contrast, the notion of prover (MIM) privacy (in authentication *and* distance bounding) *must assume* the existence of *multiple provers* (and usually a single verifier). The privacy framework for (RFID) authentication due to Hermans *et al.* [HPVP11], which we use in this work, considers many potentially corrupted provers. Ideally, one should also account for multiple provers in distance-bounding. While this is beyond the scope of this paper, the recent work of Boureanu *et al.* [BMV13] makes a step towards generalizing distance-bounding security to the multi-prover setting. However, this model is not compatible with the framework of Dürholz *et al.*. As our protocol extends the HPO distance-bounding protocol, whose properties are proved within the framework of Dürholz *et al.*, we rely on the same model in this paper.

To summarize, our objective is to develop a protocol based on a model using the strong security requirements (correctness and soundness) of the multi-prover authentication framework of Hermans *et al.* while extending this model to capture our stronger privacy requirements. We proceed by outlining the models we use in the following subsections.

2.3 Correctness and Soundness

Authentication protocols must respect two *security* properties: namely, *correctness* and *soundness* [Vau07, HPVP11]. Hence, a legitimate prover should (almost) always be authenticated by an honest verifier (correctness). This implicitly rules out transmission errors. Meanwhile, no adversary, interacting arbitrarily with the verifier and any corrupted prover, should be authenticated as an honest (*i.e.*, non-corrupted) target prover by an honest verifier (soundness).

These definitions do *not* capture the fact that the adversary can interact with the public board $\mathcal{B}$. Thus, we updated them to take this aspect into account in the following manner (the formal security definition is given in Section 3).

Definition 1 (Extended Correctness)**.** *A legitimate prover should be (almost) always be authenticated by an* honest verifier*, even in the presence of an adversary that can change the board in a way that is undetectable by both the prover and the verifier.*

An adversary can change the information on the board $\mathcal{B}$, but if this is detected by the prover or the verifier, they abort the protocol execution. We disregard attacks in which the adversary changes $\mathcal{B}$ to deny the prover the possibility to authenticate, assuming sufficient redundancy of the board. In contrast, if the prover and verifier *cannot* detect the adversary's changes, the two parties will engage in an authentication process that will fail. Such DoS attacks are a real threat and must be included in our model. Thus, we give the following definition of extended soundness.

Definition 2 (Extended Soundness)**.** *No adversary, having (unregistered) insider provers and interacting arbitrarily with the verifier, the board, and corrupted provers, should be authenticated as legitimate by an honest verifier.*

This definition of extended soundness is very strong. To achieve this goal, as well as strong privacy guarantees, we have to make two security assumptions. First, the server Srv is assumed to revoke all corrupted provers upon corruption, which implies that it must be informed whenever the prover token is stolen or damaged. Another way of seeing this assumption is that the adversary plays a two-phase game: initially, in phase (1) she can corrupt any provers at will (without their being revoked), but she does not win if she authenticates on behalf of any provers, corrupted or not; in phase (2) she cannot corrupt provers, but she wins if she authenticates successfully. Before phase (2) starts, the server is assumed to revoke all the corrupted provers. Thus, esentially, we guarantee that as soon as the server detects corruption, the adversary is no longer able to authenticate on behalf of corrupted users. Second, when the insider provers yield their private keys to an adversary, we assume they will no longer be registered by Srv. Otherwise, the adversary would be able to authenticate trivially as a legitimate prover by using the insider prover credentials. These assumptions are necessary since we require that the verifier cannot distinguish which prover is authenticated. In usual soundness models, the guarantee we have is that an adversary who can corrupt provers cannot authenticate on behalf of an uncorrupted prover. However, in our case, there is no way for the verifier to know on behalf of which prover the adversary has authenticated, unless corrupted and insider provers are in fact made illegitimate by non-registration, resp. revocation. In fact, our protocol provides the same kind of soundness guarantees as usual authentication schemes.

2.4 Privacy

Privacy is hard to define for authentication protocols, a fact reflected in the numerous models proposed in the literature [JW07, Vau07, PV08, NSMSN08, MLDL09, DLYZ11, HPVP11]. In [HPVP11], Hermans *et al.* captured privacy against MIM adversaries through an indistinguishability game between an adversary and legitimate provers, extending efforts by Juels and Weis [JW07] and including prover corruption as introduced by Vaudenay [Vau07].

In a typical security game, an adversary is defined by her objective (*i.e.*, the game she plays), her potential actions (*i.e.*, the oracles she queries) and any possible restriction in her interactions with the system (*i.e.*, the rules of the game). In a generic privacy indistinguishability game, the adversary picks two legitimate entities of her choice. A challenger then selects one of them and allows the adversary to interact with it through predefined oracles. The adversary wins the game if she can determine the entity selected by the challenger.

The adversary considered in the model proposed in [HPVP11] can register valid provers (called insiders), interact arbitrarily with the verifier and the prover selected by the challenger, query the result of an authentication session, and corrupt provers to obtain their private information. The adversary can request the challenger to run the selection process multiple times with multiple input pairs. Interactions take place through predefined oracles described in [HPVP11]. Thereafter, we present an intuitive description of the most important ones.

Prover Creation. The adversary inputs an identifier id and the CreateProver oracle creates P_i along with his new private/public key pair (x_i, X_i). The adversary obtains as output the handle[2] P_i.

Insider Creation. The adversary inputs an identifier id and the CreateInsider oracle internally runs CreateProver, before leaking the generated private part x_i to the adversary.

Draw and Release Provers. For the DrawProver oracle, the adversary inputs the identities of two available provers P_i and P_j. Then, the oracle returns a handle to one of them, thus allowing an anonymous access to him. Both provers become unavailable. Subsequently, the adversary runs the Free oracle to release both P_i and P_j. DrawProver and Free may be run multiple times, resulting in the challenger consistently drawing either the first or the second input prover.

Interactions. Through oracles, the adversary can interact as a MIM between the verifier and the selected (anonymised) prover, launching sessions and relaying messages or changing/injecting arbitrary messages to any of the two parties.

Prover Corruption. The adversary inputs the identity of provers P_i and the oracle Corrupt returns the non-volatile internal state of this prover.

Result. The adversary can also learn the result of a completed execution of the game. The Result oracle returns 1 if the prover is authenticated, 0 if this is not the case, and $\perp$ if the protocol has aborted.

At the end, the adversary *wins* the game if she can identify which prover is consistently selected by DrawProver.

Following the classification proposed by Vaudenay [Vau07], Hermans *et al.* [HPVP11] consider several classes of adversaries. These adversaries can be classified as *wide* or *narrow* depending on whether respectively they use Result or not. In parallel, they can be *weak* (if they cannot use Corrupt), *forward* (if only Corrupt can be used after a Corrupt query), *destructive* (if Corrupt destroys the corrupted prover's handle), or *strong* (if there is no restriction on the use of the oracles). Note that destructive and strong privacy in authentication protocols *cannot* be achieved with the symmetric key paradigm as shown by Paise and Vaudenay [PV08].

The original model of Hermans *et al.* [HPVP11] does not allow the adversary to register malicious insiders for which she knows the private keys (this situation was already considered by Vaudenay [Vau07]). By adding the notion of *insider* privacy, Peeters and Hermans [PH12] extended the model to capture this addition and proposed a protocol achieving this notion. Their protocol has strong security guarantees and provides narrow-strong and wide-forward insider privacy. This scheme is the basis of the HPO protocol [HPO13]. The formal privacy definition of Hermans *et al.* is the following.

[2]In a nutshell, a *handle* is a variable associated with an object (such as a protocol session or a prover), which enables the adversary to interact with this object.

Definition 3 (MIM-Privacy). *A protocol is Γ-private if and only if for all polynomial time MIM adversaries $\mathcal{A}$ in class Γ, the adversary cannot win the privacy game with a non-negligibly advantage over the probability obtained by a random guess.*

Three aspects are not considered in the above model. First (and specific to our setting), the presence of the server Srv and its corresponding public board $\mathcal{B}$ introduces new interactions such as an adversary changing the board contents.

More importantly, the above definition does *not* guarantee prover privacy with respect to a legitimate verifier. In fact, in both the protocol of Hermans *et al.* [HPVP11] and in the related HPO scheme [HPO13] (as well as in most distance-bounding and authentication protocols), the verifier can *always* distinguish which prover has actually authenticated. The extension of the privacy notion to capture prover unlinkability with respect to the verifier is addressed in Section 3.

Finally (again specific to our setting), we require prover privacy with respect to Srv. In particular, we demand prover *deniability* with respect to a server, even if it has access to the verifier's full internal state (see Section 3).

2.5 Distance-Bounding Security

In the single-prover/single-verifier model of Dürholz *et al.* [DFKO11], the adversary can open adversary-prover or adversary-verifier sessions, or simply observe prover-verifier sessions. The verifier has a clock that can accurately measure the time elapsed between sending a challenge and receiving the response. If a round trip transmission is timed, it is simply called a *time-critical* (fast) phase. Otherwise, it is called a *lazy* (slow) phase. The four main security notions of distance-bounding protocols are the following ones.

Definition 4 (Distance Fraud Resistance). *No legitimate prover located outside the prover's proximity should be able to authenticate.*

Definition 5 (Mafia Fraud Resistance). *No MIM adversary should authenticate to a verifier even in the presence of a prover, except by purely relaying messages.*

The notion of pure relay is a controversial one. Indeed, Dürholz *et al.* [DFKO11] conservatively rule out only relays of exactly the same messages in the same order. This model also allows errors or noise in the transmissions. We skip these complexities here, noting that it is easy to modify protocols to take them into account.

Definition 6 (Lazy Impersonation Resistance). *No MIM adversary should be able to authenticate to the verifier in the lazy phases except by purely relaying messages.*

This notion is independent of the notion of soundness, as defined by Hermans *et al.* [HPO13]. We refer the reader to the appendix for an in-depth discussion of this relationship.

Definition 7 (Terrorist Fraud Resistance). *If a MIM adversary aided by a malicious, offline prover can authenticate to the verifier, then the information provided by the prover will give him at least an equal chance to authenticate later, even if he is unaided.*

The latter notion, called simulation-based terrorist fraud resistance by Fischlin and Onete [FO13b], is a very strong notion. While we advocate the use of schemes achieving this strong notion, we leave the improvement of our protocol to gain terrorist-fraud resistance as future work.

3 Prover Anonymity

We have to extend the privacy model of Hermans *et al.* [HPVP11] in two directions: (1) model the interactions of the adversary with the board $\mathcal{B}$ and (2) generalize the privacy notion to protect provers against malicious verifiers and the server Srv.

For the first extension, we introduce the $\mathsf{ChangeBoard}$ oracle that allows an adversary to alter the contents of $\mathcal{B}$ with her own input. For the second one, we first introduce the notion of *prover anonymity*, requiring that an adversary cannot link sessions of provers *even* if she learns the verifier's full internal state. We next define the concept of *deniability*, demanding that Srv (knowing all the provers' private keys) cannot distinguish a real protocol transcript from a simulated one, even if the verifier gives his full internal state.

For prover anonymity, we rely on the notion of *collusion* between an adversary and a verifier to capture *honest-but-curious* and *malicious* verifiers. This enables us to present our model as an extension to the privacy model of Hermans *et al.* [HPVP11]. Both in our model and the original model, the honest-but-curious verifier always follows the protocol but tries to link prover sessions. Thus, we have to restrict an MIM adversary to be a simple proxy between the verifier and the prover as opposed to a fully-functional MIM colluding with a verifier, which is equivalent to a malicious verifier. We call this extended privacy property prover anonymity.

In order to make the distinction between the different behaviors of the verifier, a new parameter flag is associated with the game that specifies the oracle behavior. This parameter takes one of three different values: hon, hbc, mal, standing respectively for an honest verifier, an honest-but-curious verifier or a malicious verifier.

***The Effect of* flag.** At the beginning of the privacy game, the adversary selects the value of flag. If $\mathsf{flag} \in \{\mathsf{hbc}, \mathsf{mal}\}$, the adversary receives the verifier's full internal state. In addition, whenever the verifier sends a message to the adversary (*e.g.*, via $\mathsf{SendVerifier}$ or Launch queries, see [HPVP11]), the adversary also receives an update of the internal state. This information enables the adversary to check the legitimacy of the prover without Result queries. Essentially, for $\mathsf{flag} \in \{\mathsf{hbc}, \mathsf{mal}\}$, the verifier always behaves honestly, but passes on his full internal state to the MIM adversary. If $\mathsf{flag} = \mathsf{hbc}$, the adversary may only relay the prover-verifier communication, modeled by the oracle $\mathsf{Execute}$. In this case, the adversary cannot interact with $\mathsf{Corrupt}$, $\mathsf{CreateInsider}$, nor any other interaction oracle[3]. If $\mathsf{flag} = \mathsf{mal}$, the adversary cannot corrupt provers or create insiders but may use any interaction oracle as in the original game. Finally, if $\mathsf{flag} = \mathsf{hon}$, the adversary runs the original MIM-privacy game, while also being able to modify the content of $\mathcal{B}$.

Extended interaction model. We introduce two additional oracles, namely $\mathsf{ChangeBoard}$ and $\mathsf{Execute}$. The first oracle allows the adversary to change $\mathcal{B}$, while the second triggers an honest prover-verifier protocol execution in which the adversary acts as a proxy.

$\mathsf{ChangeBoard}(\mathcal{B}')$. This oracle changes the contents of the public board $\mathcal{B}$ with the new contents $\mathcal{B}'$.

$\mathsf{Execute}(\cdot)$. When given as input a prover index I, this oracle executes the protocol be-

[3] This restriction can be relaxed, for both the hbc and the mal modes, to allow $\mathsf{CreateProver}$ and $\mathsf{CreateInsider}$ queries. This is equivalent to achieving both deniability and honest-but-curious prover anonymity.

tween the prover I and the verifier. The prover index I is either an index $i \in \{1, \ldots, k\}$, for k, the number of provers of the system, or the anonymized handle as returned by $\mathsf{DrawProver}$. More specifically, when calling $\mathsf{Execute}$, then Launch is first called, yielding a handle prot and a first verifier message. Afterwards, the oracle executes $\mathsf{SendProver}$ for I with the message returned by Launch. This process is repeated by alternating queries to $\mathsf{SendVerifier}_{\mathsf{flag}}$ and $\mathsf{SendProver}$, essentially relaying the communication between the two parties. At the end of the protocol, the transcript of prot is returned. If the protocol aborts prematurely, a special symbol $\perp$ is appended at the end of the transcript.

We also modify the following oracles defined in the original model of Hermans *et al.* [HPVP11].

$\mathsf{CreateProver}(\cdot)$. When given as input an identifier id, this oracle runs prover initialization for a prover P_i. The private/public key pair is generated and stored in P_i and Srv. The server updates the board $\mathcal{B}$ according to $\mathcal{B}'$.

$\mathsf{CreateInsider}(\cdot)$. When given as input an identifier id, this oracle creates a new prover P_i. The private/public key pair of this prover is given to the adversary. In the privacy model, the key pair is registered with Srv and $\mathcal{B}$ is updated accordingly. In the soundness model, the private key is *not* given to the server. As already discussed, we make this assumption in view of the strong privacy guarantees we require.

$\mathsf{SendProver}(\cdot,\cdot,\cdot)$. This oracle takes as input either an anony–mised index I and a message m (in the privacy model) or an index $i \in \{1, \ldots, k\}$, a message m, and a protocol handle prot (in the soundness model). For the privacy model, this oracle works as in the original (privacy) model. For the soundness model, m is sent to P_i in the protocol session prot.

$\mathsf{Corrupt}(\cdot)$. This oracle takes as input either an anonymised index I (in the privacy model) or a prover index $i \in \{1, \ldots, k\}$ (in the soundness model). For the privacy model, this oracle works as in the original (privacy) model. For the soundness model, as soon as this oracle is used, the prover is revoked and Srv removes the information of the prover from its database and updates $\mathcal{B}$ accordingly.

$\mathsf{SendVerifier}_{\mathsf{flag}}(\cdot,\cdot)$. This oracles takes as input a message m and a protocol handle. If $\mathsf{flag} = \mathsf{hon}$, the oracle works as originally defined. If $\mathsf{flag} \in \{\mathsf{hbc}, \mathsf{mal}\}$, it returns the message m' as done by the verifier V and the full internal state of V.

Note that the oracles $\mathsf{SendProver}$, $\mathsf{Corrupt}$, and $\mathsf{CreateInsider}$ run differently for the soundness and the privacy models. To distinguish these modes, we use the notation Oracle^* if this oracle runs currently in the soundness model.

The games. We start by describing the *extended correctness* game. The adversary is given access to $\mathsf{CreateProver}$, $\mathsf{ChangeBoard}$, Result, and $\mathsf{Execute}$. The adversary *wins* if there is a session prot for which $\mathsf{Result}(\mathsf{prot})$ returns 0. The adversary's advantage $\mathbf{Adv}_{\mathsf{EC}}(\mathcal{A})$ corresponds to her winning probability.

For the *extended soundness* game, the adversary $\mathcal{A}$ is given access to $\mathsf{CreateProver}$, $\mathsf{CreateInsider}^*$, $\mathsf{ChangeBoard}$, Launch, $\mathsf{SendProver}^*$, $\mathsf{SendVerifier}$, Result and $\mathsf{Corrupt}^*$. $\mathcal{A}$ *wins* if there is a session prot, for which $\mathsf{Result}(\mathsf{prot})$ returns 1 and there is no $\mathsf{SendProver}^*$ query with input $(\cdot,\cdot,\mathsf{prot})$. Her advantage $\mathbf{Adv}_{\mathsf{Sound}}(\mathcal{A})$ is her winning probability.

For the *prover anonymity* game, the adversary $\mathcal{A}$ starts by selecting a value $\mathsf{flag} \in \{\mathsf{hon}, \mathsf{hbc}, \mathsf{mal}\}$. Then, $\mathcal{A}$ is given access to $\mathsf{ChangeBoard}$, $\mathsf{DrawProver}$, and Free. If $\mathsf{flag} = \mathsf{hon}$, $\mathcal{A}$ gets access also to $\mathsf{CreateProver}$ and $\mathsf{Corrupt}$ (depending on the adversary model). The adversary also has access to Launch, $\mathsf{SendProver}$, $\mathsf{SendVerifier}_{\mathsf{flag}}$ and possibly Result (if $\mathsf{flag} \in \{\mathsf{hon}, \mathsf{mal}\}$) or to $\mathsf{Execute}$ (if $\mathsf{flag} = \mathsf{hbc}$). Note that for $\mathsf{flag} \in \{\mathsf{hbc}, \mathsf{mal}\}$, the use of Result is no longer necessary, as the adversary knows the verifier's full internal state (in this case, narrow privacy is equivalent to wide privacy). $\mathcal{A}$ *wins* if she can find which prover has been selected by $\mathsf{DrawProver}$. Her advantage $\mathbf{Adv}_{\mathsf{PAnon}}(\mathcal{A})$ is her winning probability minus the guessing probability $\frac{1}{2}$. We preserve the same definition and adversary classes as Hermans *et al.*.

Finally, we also define the notion of *deniability* in the presence of an honest-but-curious back-end server Srv[4]. Let $\mathsf{state}_{\mathsf{V}}$ denote the full internal state of the verifier V and let Sim denote a simulator taking as input $\mathsf{state}_{\mathsf{V}}$ and returning a transcript τ_{Sim}. We require that the transcript of a honest session between V and a (legitimate) prover chosen by Srv is identical to a simulated transcript generated by V (from Srv's point of view, for the same state $\mathsf{state}_{\mathsf{V}}$). In reach this property, we introduce a final oracle. $\mathsf{DenyRoR}_{b,\mathsf{state}_{\mathsf{V}}}(\cdot)$. This oracle is conditioned on a bit b and on the verifier's state $\mathsf{state}_{\mathsf{V}}$ given by the verifier V to the server Srv. When given as input an index $i \in \{1, \dots, k\}$, the oracle runs either an honest session between P_i and V in state $\mathsf{state}_{\mathsf{V}}$ (if $b = 0$) and returns the session transcript τ_0, or it runs Sim on $\mathsf{state}_{\mathsf{V}}$ and returns the simulated transcript τ_1 (if $b = 1$). The oracle also outputs the updated $(\tau_b, \mathsf{state}_{\mathsf{V}})$.

For the *deniability* game, the adversary (*i.e.*, here the server Srv) is given $\mathsf{state}_{\mathsf{V}}$ and access to the $\mathsf{CreateProver}$ and $\mathsf{DenyRoR}$ oracles. The adversary *wins* if she outputs the correct bit used as input to $\mathsf{DenyRoR}_{b,\mathsf{state}_{\mathsf{V}}}(\cdot)$. Her advantage $\mathbf{Adv}_{\mathsf{deny}}(\mathcal{A})$ is her probability to win minus the guessing probability $\frac{1}{2}$. A protocol is *perfectly deniable* if the advantage of the best adversary is $\frac{1}{2}$.

4 Our construction

Our protocol extends the existing HPO protocol of Hermans, Peeters, and Onete [HPO13], which is narrow-strong and wide-insider-forward MIM-private, as well as correct, sound, and resistant to mafia and distance frauds. We briefly outline this protocol and then modify it in order to achieve our privacy notions of prover anonymity and deniability.

4.1 The HPO Distance-Bounding Protocol

Consider the classical elliptic curve cryptography setting on a subgroup $\mathbb{G} =< P >$ of prime order q defined by the points on the elliptic curve $\mathcal{E}$ over a finite field $\mathbb{F}_q$. The point P is a generator of this group. Each prover P_i has a private/public key pair (x_i, X_i), for $x_i \in \mathbb{Z}_q$ and $\mathrm{X}_i \coloneqq x_i P$. The verifier has a private/private key pair (y, Y), for $y \in \mathbb{Z}_q$ and $\mathrm{Y} \coloneqq yP$, and knows all X_i.

Overview. The protocol presented in Figure 1 merges authentication and distance-bounding (with near-optimal mafia-fraud resistance). First, the parties exchange nonces, which are points on $\mathcal{E}$: two on the prover side (R_1, R_2) and one on the verifier side (R_3). Intuitively, the prover needs two nonces to preserve his MIM privacy. These nonces will effectively blind (from the point of view of a MIM adversary) the prover's key in the final authentication string s.

Following the nonce exchange, n time-critical phases take place. Before starting them, the verifier chooses an integer $e \in \mathbb{Z}_q$, and truncates it to n bits. These bits are

[4]This can be extended to a malicious server by adding further NIZK-PK in our protocol.

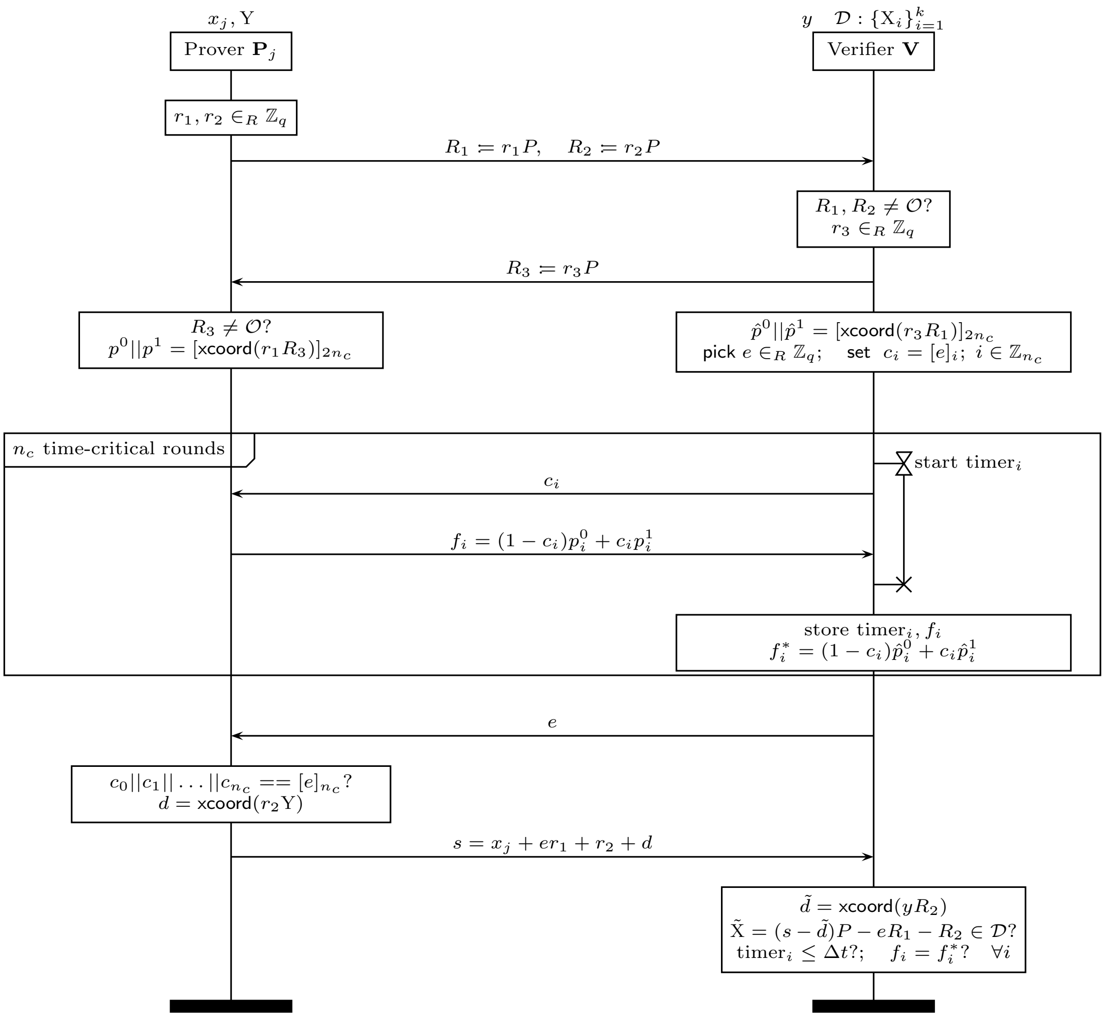
x_j, Y
$y \quad \mathcal{D} : \{\mathrm{X}_i\}_{i=1}^k$
Prover $\mathbf{P}_j$
Verifier $\mathbf{V}$
$r_1, r_2 \in_R \mathbb{Z}_q$
$R_1 \coloneqq r_1 P, \quad R_2 \coloneqq r_2 P$
$R_1, R_2 \neq \mathcal{O}?$
$r_3 \in_R \mathbb{Z}_q$
$R_3 \coloneqq r_3 P$
$R_3 \neq \mathcal{O}?$
$p^0 || p^1 = [\mathsf{xcoord}(r_1 R_3)]_{2n_c}$
$\hat{p}^0 || \hat{p}^1 = [\mathsf{xcoord}(r_3 R_1)]_{2n_c}$
pick $e \in_R \mathbb{Z}_q$; set $c_i = [e]_i$; $i \in \mathbb{Z}_{n_c}$
n_c time-critical rounds
start timer$_i$
c_i
$f_i = (1 - c_i) p_i^0 + c_i p_i^1$
store timer$_i$, f_i
$f_i^* = (1 - c_i) \hat{p}_i^0 + c_i \hat{p}_i^1$
e
$c_0 || c_1 || \dots || c_{n_c} == [e]_{n_c}?$
$d = \mathsf{xcoord}(r_2 \mathrm{Y})$
$s = x_j + e r_1 + r_2 + d$
$\tilde{d} = \mathsf{xcoord}(y R_2)$
$\tilde{\mathrm{X}} = (s - \tilde{d})P - eR_1 - R_2 \in \mathcal{D}?$
timer$_i \leq \Delta t?$; $f_i = f_i^*?$ $\forall i$

Figure 1: Secure, wide-strong insider-private distance-bounding protocol.

associated with the challenges c_i, for $i = 1, \dots, n$. The prover's responses are computed by taking the first $2n$ bits of $\mathsf{xcoord}()$, which correspond to uniformly distributed random bits [HPO13]. The function $\mathsf{xcoord}()$ computes the x-coordinate of a point on $\mathcal{E}$. Note that the number n of time-critical phases is small compared to the group size. For each challenge c_i, the prover responds with a bit either from p^0 or p^1.

Intuition. This protocol can be viewed either as a privacy-enhanced distance-bounding protocol or as a distance-approximation-enhanced MIM-private authentication protocol. The randomness of the $2n$ bits ensures the distance-fraud resistance. All distance-bounding responses are session-specific, and the string s hides the prover's secret key from a MIM adversary. Finally, binding the response s to the challenge string e ensures nearly perfect mafia-fraud resistance. Thus, this protocol ensures both distance bounding and MIM-private authentication while being efficient. The exact security properties of this protocol are outlined in the appendix.

4.2 Prover Anonymous and Deniable Scheme

In order to preserve the prover anonymity against a (honest-but-curious or malicious) verifier, we adapted the HPO protocol in two distinct places. The first change concerns the initial nonce-exchange phase, in which the prover and the verifier will rely on an homomorphic encryption scheme and a Non-Interactive-Zero-Knowledge (NIZK) Proof of Knowledge system to derive a session key. The second change occurs during the last authentication phase, in which the prover uses his private key to compute a response attesting of his legitimacy without revealing his identity. To ensure long-term forward privacy, we assume that all ephemeral keys and random nonces are discarded as soon as possible. We obtain better prover anonymity guarantees (*i.e.*, wide-strong prover anonymity) than the corresponding degree of MIM privacy of the HPO protocol (*i.e.* narrow-strong and wide-insider-forward). We also preserve the same (high) correctness, soundness, and distance-bounding security guarantees, while also achieving deniability with respect to an honest-but-curious back-end server.

Setting. We specify a homomorphic encryption scheme[5] $\mathcal{HE} = (\mathsf{HE.KGen}, \mathsf{HEnc}, \mathsf{HDec})$ with message-space $\mathbb{G}$, which is IND-RCCA and IK-CCA secure (see the appendix for the definitions of these concepts).

We assume that Srv generates and self-certifies its private/public key pairs (z, Z). The keys are updated and certified at every prover creation and revocation, and the certificates of old keys are placed on a revocation list. These keys are used with an EUF-CMA signature scheme $\mathcal{S} = \{\mathsf{SS.KGen}, \mathsf{Sign}, \mathsf{Vfy}\}$.

Finally, the prover will use a NIKZ-PK system $\mathcal{PK} = (\mathsf{par}, \mathsf{Prove}, \mathsf{Vf}, \mathsf{Sim}, \mathsf{Ext})$. In this system, the witness is a private/public encryption key pair and the randomness used to encrypt, while the statement is that a specific ciphertext c is the encryption of the value $\omega \mathrm{Q}_i$ with $i \in \{1, \ldots, k\}$ and some ephemeral ω, for publicly known values Q_i with probability exactly $\frac{1}{k}$ *and* that the prover knows the ephemeral key ω. This scheme should have perfect completeness.

Prover setup. Prover creation associates a prover P_i, for $i \in \{1, \ldots, k\}$, with a secret[6] key $x_i \in \mathbb{Z}_q$ and an auxiliary public key Q_i stored by P_i and Srv. The key Q_i is defined as $\prod_{j=1;j\neq i}^{k} x_j P$, for $i = 1, \ldots, k$, while the public key Q is defined as follows $\mathrm{Q} \coloneqq \prod_{i=1}^{k} x_i P$. These values are updated dynamically as provers are created and revoked. The server publishes on $\mathcal{B}$ a tuple containing the message $m \coloneqq \mathrm{Q}_1, \ldots, \mathrm{Q}_k, \mathrm{Q}$, and generates a signature key pair, certifies it, and publishes the signature $\sigma \coloneqq \mathsf{Sign}(z, m)$ on $\mathcal{B}$.

Prover revocation. Whenever prover P_i is revoked, the server removes his secret key x_i from his database and updates all the values Q_j, Q and σ in its internal database and on the public board $\mathcal{B}$.

Protocol intuition. Most of the structure of the HPO protocol is preserved. However, instead of authenticating as the prover in possession of key x_j, P_j proves that he can compute $x_j \mathrm{Q}_j = \mathrm{Q}$. At each session, the (potentially malicious) verifier generates a random value r and expects to retrieve the value $r\mathrm{Q}$ from the string S sent by P_j. The

[5]In practice, we use the homomorphic EC ElGamal encryption scheme. Thus, knowing $\mathsf{HEnc}(\mathsf{pk}, \mathrm{Q}_i)$, anyone can compute $\mathsf{HEnc}(\mathsf{pk}, r\mathrm{Q}_i)$. This is equal to $r \circ \mathsf{HEnc}(\mathsf{pk}, \mathrm{Q}_i)$, which denotes the multiplication of the encryption r times by itself.

[6]The term *secret* is used here to emphasize the fact that x_i and Q_i cannot be derived one from the other.

prover is not given the value r explicitly. Instead, he computes $\mathsf{HEnc}(\mathsf{epk}, \mathrm{Q}_i)$ under an ephemeral private/public key pair $(\mathsf{esk}, \mathsf{epk})$ known from the prover only. Afterwards, the verifier uses the homomorphic property of the encryption scheme to return $\mathsf{HEnc}(\mathsf{epk}, r\mathrm{Q}_i)$. In order to prevent attacks against anonymity, the prover also chooses an ephemeral secret (which is thus not corruptible by the adversary) and must also include a NIZK proof that the ciphertext computed with the homomorphic encryption scheme is well-formed AND that the prover knows the ephemeral secret value. Finally, the prover decrypts the value, multiplies it by the inverse of his ephemeral value, and computes $x_i(r\mathrm{Q}_i) = r\mathrm{Q}$.

Retrieving information from the board. Though not explicitly depicted in Figure 3, both the prover and the verifier *only* retrieve information from $\mathcal{B}$ if the signature σ and the corresponding certificate are valid.

The full interaction model is shown in Figure 2 and our protocol is depicted in Figure 3.

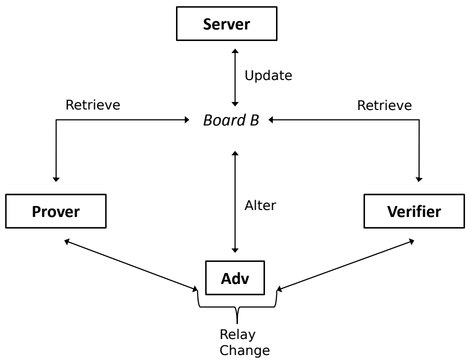

Figure 2: Interactions diagram between the different parties.

Security intuition. The construction keeps the distance-fraud resistance of the HPO protocol, as the time-critical responses are generated similarly in both schemes. For mafia-fraud resistance, the new response string S hides $r\mathrm{Q}$ in the same way as x_j is hidden in the HPO protocol. We effectively prevent adversaries from changing the board contents or replaying old boards (without being detected) by using one-time (certified) signature key pairs. For the soundness property, we show that no MIM adversary can modify the ciphertext or obtain $r\mathrm{Q}$. In fact, if the adversary encrypts a different public auxiliary key Q_j, she cannot produce the last response S without knowing x_j. This is impossible since any corrupted prover is automatically revoked, the board is always updated before authentication and the credentials gained cannot be reused by the MIM adversary.

Next, we analyze the privacy properties of the protocol, which is the main result of our paper. For MIM privacy (*i.e.*, an *honest* verifier), the following observations can be made. First, each prover authenticates under the same public key Q, sent with the response string S. Thus, the only way a MIM adversary can learn the identity of the prover by linking sessions in which the same value Q_i was used. Second, the corruption of P_i yields the long term secret x_i and the corresponding public value Q_i, but not the nonces or the ephemeral keys used for encryption. Thus, even upon corruption,

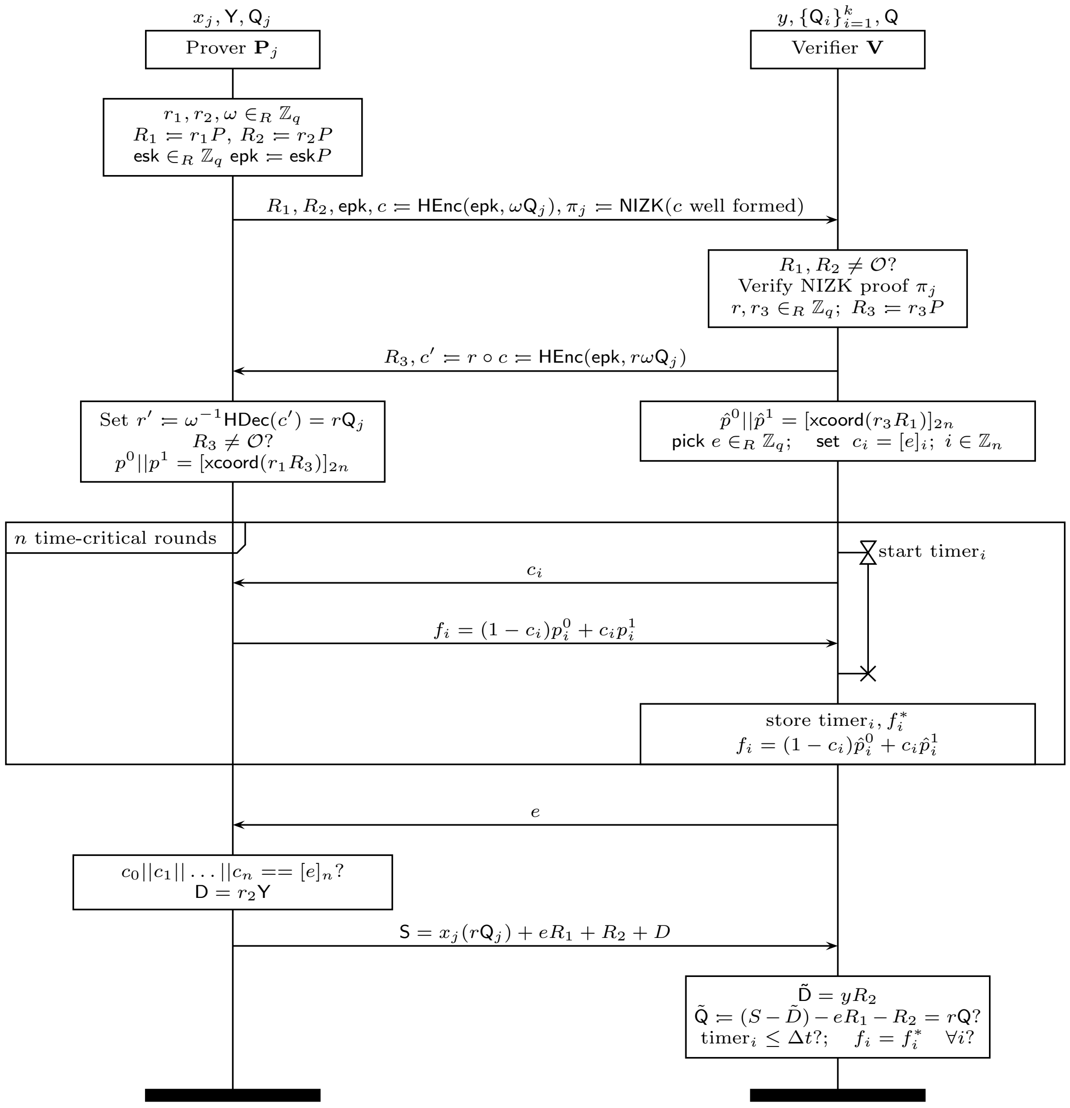
$x_j, \mathsf{Y}, \mathsf{Q}_j$
Prover $\mathbf{P}_j$
$y, \{\mathsf{Q}_i\}_{i=1}^k, \mathsf{Q}$
Verifier $\mathbf{V}$
$r_1, r_2, \omega \in_R \mathbb{Z}_q$
$R_1 \coloneqq r_1 P,\ R_2 \coloneqq r_2 P$
$\mathsf{esk} \in_R \mathbb{Z}_q\ \mathsf{epk} \coloneqq \mathsf{esk}P$
$R_1, R_2, \mathsf{epk}, c \coloneqq \mathsf{HEnc}(\mathsf{epk}, \omega\mathsf{Q}_j), \pi_j \coloneqq \mathsf{NIZK}(c \text{ well formed})$
$R_1, R_2 \neq \mathcal{O}?$
Verify NIZK proof π_j
$r, r_3 \in_R \mathbb{Z}_q;\ R_3 \coloneqq r_3 P$
$R_3, c' \coloneqq r \circ c \coloneqq \mathsf{HEnc}(\mathsf{epk}, r\omega\mathsf{Q}_j)$
Set $r' \coloneqq \omega^{-1}\mathsf{HDec}(c') = r\mathsf{Q}_j$
$R_3 \neq \mathcal{O}?$
$p^0||p^1 = [\mathsf{xcoord}(r_1 R_3)]_{2n}$
$\hat{p}^0||\hat{p}^1 = [\mathsf{xcoord}(r_3 R_1)]_{2n}$
pick $e \in_R \mathbb{Z}_q$; set $c_i = [e]_i;\ i \in \mathbb{Z}_n$
n time-critical rounds
start timer$_i$
c_i
$f_i = (1 - c_i)p_i^0 + c_i p_i^1$
store timer$_i, f_i^*$
$f_i = (1 - c_i)\hat{p}_i^0 + c_i \hat{p}_i^1$
e
$c_0||c_1||\dots||c_n == [e]_n?$
$\mathsf{D} = r_2\mathsf{Y}$
$\mathsf{S} = x_j(r\mathsf{Q}_j) + eR_1 + R_2 + D$
$\tilde{\mathsf{D}} = yR_2$
$\tilde{\mathsf{Q}} \coloneqq (S - \tilde{D}) - eR_1 - R_2 = r\mathsf{Q}?$
timer$_i \leq \Delta t?;\quad f_i = f_i^*\quad \forall i?$

Figure 3: Our prover-anonymous and deniable distance-bounding construction.

the MIM adversary cannot distinguish previous and future encryptions of Q_i. Third, the IND-RCCA and IK-CCA security of $\mathcal{HE}$ and the zero-knowledge and soundness properties of the NIZK proof essentially ensure that the encrypted values Q_i do not leak, and cannot be effectively tampered with by the adversary.

The same considerations hold for an *honest-but-curious verifier*, who also knows the verifier's private key y and the ephemeral value r chosen for each session. However, $\mathcal{HE}$ still hides the encrypted value Q_i.

Intuitively, a *malicious verifier* can perform one of the following actions: (1) choose a convenient value r (however, this affects all provers equally) or (2) try to learn the value Q_i encrypted by the prover (this fails due to the IND-RCCA and IK-CCA security of $\mathcal{HE}$).

Finally, our protocol is deniable with respect to the back-end server. The crucial detail for this property is that, while the verifier does not know the secret keys x_i, he can compute the value $r\mathrm{Q}$ since he knows Q. This is how the simulator computes the response, and thus the simulation is perfect.

4.3 Security Analysis

In this section, we prove the security properties of our protocol. Since this is common to all properties, we omit to model the certification scheme used by Srv. We simply remark that $\mathsf{A}^{\mathsf{Cert}} \coloneqq \mathbf{Adv}_{\mathsf{cert}}(\mathcal{A}) + \mathbf{Adv}_{\mathsf{unf}}(\mathcal{A}_1)$ accounts for forging a certificate (*e.g.*, its revocation date) or a signature. In the following, we always assumed that the adversaries considered are polynomial time-bounded (*i.e.*, they are assumed not to be able to break the computational assumptions on which the security of the cryptographic primitives rely).

Theorem 1 (Security and privacy of the protocol)**.** *Consider the protocol in Figure 3, instantiated for a subgroup $\mathbb{G} \coloneqq < P >$ of prime order q of an elliptic curve $\mathcal{E}$ over a field $\mathbb{F}_q$. Let $\mathsf{A}^{\mathsf{Cert}}$ be the advantage of a best forger for either the signature scheme $\mathcal{S}$ or for the certification scheme used to certify keys for $\mathcal{S}$. In this situation, the following holds:*

- *For any extended correctness adversary $\mathcal{A}$, there is an adversary $\mathcal{A}_1$ against the unforgeability of the signature scheme $\mathcal{S}$ such that $\mathbf{Adv}_{\mathsf{EC}}(\mathcal{A}) \leq \mathsf{A}^{\mathsf{Cert}}$.*
- *For any (q_{V})-adversary $\mathcal{A}$ against the distance-fraud resistance of the protocol, there is an adversary $\mathcal{A}_1$ against the indistinguishability from random of the truncation of a DDH product such that: $\mathbf{Adv}_{\mathsf{DF}}(\mathcal{A}) \leq q_{\mathsf{V}}\{\frac{3}{4}\}^n + \mathbf{Adv}_{\mathsf{dist}}(\mathcal{A}_1) + \mathsf{A}^{\mathsf{Cert}}$, in which q_{V} is the number of verifier-adversary sessions and n the number of time critical rounds.*
- *For any $(q_{\mathsf{P}}, q_{\mathsf{V}}, q_{OBS})$-mafia-fraud adversary $\mathcal{A}$ there are the adversaries $\mathcal{A}_1$ against the indistinguishability from random of the truncation of a DDH product, $\mathcal{A}_2$ against the hardness of the DLog problem, and $\mathcal{A}_3$ against the soundness of the protocol, such that*

$$\begin{aligned}\mathbf{Adv}_{\mathsf{MF}}(\mathcal{A}) \quad \leq \quad & q_{\mathsf{V}} \cdot \left(\tfrac{1}{2}\right)^n + \binom{q_{\mathsf{V}} + q_{OBS}}{2} \cdot 2^{-|\mathbb{G}|} + \binom{q_{\mathsf{P}}}{2} \cdot 2^{-|\mathbb{G}|} + \mathsf{A}^{\mathsf{Cert}} + \\ & \mathbf{Adv}_{\mathsf{dist}}(\mathcal{A}_1) + 2q_{\mathsf{V}}\mathbf{Adv}_{\mathsf{DL}}(\mathcal{A}_2) + \mathbf{Adv}_{\mathsf{Sound}}(\mathcal{A}_3).\end{aligned}$$

- *For any extended soundness adversary $\mathcal{A}$, there are the adversaries $\mathcal{A}_1$ against the soundness of the NIZK proof and $\mathcal{A}_2$ against the hardness of the DLog problem, such that*

$$\mathbf{Adv}_{\mathsf{Sound}}(\mathcal{A}) \quad \leq \quad \binom{q}{2} \cdot 2^{-|\mathbb{G}|} + \mathsf{A}^{\mathsf{Cert}} + q(\mathbf{Adv}_{\mathsf{NIZK.Snd}}(\mathcal{A}_1) + \mathbf{Adv}_{\mathsf{DL}}(\mathcal{A}_2)),$$

  in which q is the number of sessions run by $\mathcal{A}$.
- *For any wide-strong-prover anonymity adversary $\mathcal{A}$, there are the adversaries $\mathcal{A}_1$ against the zero-knowledge property of the NIZK-PK, $\mathcal{A}_2$ against the IND-RCCA of the homomorphic encryption scheme, $\mathcal{A}_3$ against the IND-RCCA of the encryption scheme (possibly different from $\mathcal{A}_2$), $\mathcal{A}_4$ against the IK-CCA of the encryption scheme, and $\mathcal{A}_5$ against the soundness of the NIZK-PK such that:*

$$\mathbf{Adv}_{\mathsf{PAnon}}(\mathcal{A}) \leq \mathsf{A}^{\mathsf{Cert}} + qA.$$

where it holds: $A = \mathbf{Adv}_{\textsf{NIZK.ZK}}(\mathcal{A}_1) + \mathbf{Adv}_{\textsf{IND-RCCA}}(\mathcal{A}_2) + \mathbf{Adv}_{\textsf{IND-RCCA}}(\mathcal{A}_3) + 2\mathbf{Adv}_{\textsf{IK-CCA}}(\mathcal{A}_4) + \mathbf{Adv}_{\textsf{NIZK.Snd}}(\mathcal{A}_5)$.
Moreover, for an honest-but-curious wide-strong prover-anonymity adversary $\mathcal{A}^{\mathsf{hbc}}$*, we have*

$$\mathbf{Adv}_{\textsf{PAnon}}(\mathcal{A}^{\mathsf{hbc}}) \leq \mathsf{A}^{\mathsf{Cert}} + qA^*,$$

in which A^* *is* $\mathbf{Adv}_{\textsf{NIZK.ZK}}(\mathcal{A}_1) + \mathbf{Adv}_{\textsf{IND-RCCA}}(\mathcal{A}_2) + \mathbf{Adv}_{\textsf{IND-RCCA}}(\mathcal{A}_3)$.

- *The protocol is perfectly deniable with respect to an honest-but-curious server.*
- *For any* $(q_{\mathsf{P}}, q_{\mathsf{V}}, q_{OBS})$*-impersonation adversary* $\mathcal{A}$*, there are adversaries* $\mathcal{A}_1$ *against the extended soundness of the protocol,* $\mathcal{A}_2$ *against the mafia-fraud resistance of the protocol, and adversaries* $\mathcal{A}_3$ *against the soundness of the NIZK proof,* $\mathcal{A}_4$ *against the key indistinguishability of the encryption scheme, and* $\mathcal{A}_5$ *breaking the DLog assumption such that:*

$$\begin{aligned}\mathbf{Adv}_{\textsf{Imp}}(\mathcal{A}) \quad \leq \quad & q_{\mathsf{V}}(\mathbf{Adv}_{\textsf{NIZK.Snd}}(\mathcal{A}_3) + \mathbf{Adv}_{\textsf{DL}}(\mathcal{A}_5) + \mathbf{Adv}_{\textsf{IK-CCA}}(\mathcal{A}_4)) + \\ & \mathbf{Adv}_{\textsf{Sound}}(\mathcal{A}_1) + \mathbf{Adv}_{\textsf{mafia}}(\mathcal{A}_2).\end{aligned}$$

Proof. ***Extended Correctness.*** This inequality follows trivially. Indeed, $\mathcal{A}$ can access only the oracles CreateProver, ChangeBoard, Result, and Execute. If ChangeBoard is not used, then the protocol has perfect correctness (as the NIZK-PK system is perfectly correct). If ChangeBoard is used and $\mathcal{A}$ succeeds with advantage $\mathbf{Adv}_{\mathsf{EC}}(\mathcal{A})$, we can construct either a forger against $\mathcal{S}$, or one against the certification scheme. Thus, $\mathbf{Adv}_{\mathsf{EC}}(\mathcal{A}) \leq \mathsf{A}^{\mathsf{Cert}}$ using our notation.

Extended Soundness. Consider the session in which $\mathcal{A}$ succeeds with probability $\mathbf{Adv}_{\mathsf{Sound}}(\mathcal{A})$ to authenticate. First, we account for the changes of values of the board, losing a term $\mathsf{A}^{\mathsf{Cert}}$. We can now make the *a priori* assumption that the prover will not use Corrupt or CreateInsider (both result in a change on $\mathcal{B}$ such that the obtained credentials are no longer useful). Next, we rule out the possibility of having collisions of values of r between this session and any other session, losing in the way a term $\binom{q}{2} \cdot 2^{-|\mathbb{G}|}$. Finally, we modify the original game such that $\mathcal{A}$ is considered to lose if she can send a first message to the prover consisting of a ciphertext c and a NIZK-PK π such that c does *not* encrypt a legitimate value Q_i. The modified game is equivalent up to a success term of $q\mathbf{Adv}_{\mathsf{NIZK.Snd}}(\mathcal{A}_1)$ with the original game (we can construct an adversary against the soundness of the NIZK-PK outputting the forged NIZK-PK).

Note that for soundness, the adversary cannot run a full MIM attack for the session that she succeeds in. Based on the soundness of the NIZK-PK, the adversary will receive in the challenge session, the value $r \circ c = \mathsf{HEnc}(\mathsf{epk}, \omega\mathrm{Q}_i)$, for a value Q_i and a private/public key pair $(\mathsf{esk}, \mathsf{epk})$ chosen by her. Thus, the adversary learns $r\mathrm{Q}_i$, but not the value of r (by the DLog assumption). As a result, we lose a term $\mathbf{Adv}_{\mathsf{DL}}(\mathcal{A}_2)$.

Distance and Mafia Fraud. These proofs are identical to the corresponding proofs for the HPO protocol [HPO13], only accounting for the additional term $\mathsf{A}^{\mathsf{Cert}}$. However, note that the advantage against soundness, which is used for the mafia-fraud resistance, is different for our construction.

Prover anonymity. We discuss each adversary mode in {hon, hbc, mal}. Consider an hypothetical adversary $\mathcal{A}$ that can win the wide-strong privacy game in hon mode. We assume *a priori* that all the provers have already been corrupted, and $\mathcal{A}$ knows the values x_i, for $i = 1, \dots, k$. Note that the volatile ephemeral key pair $(\mathsf{esk}, \mathsf{epk})$ is not leaked upon corruption. We first rule out any illegitimate change to $\mathcal{B}$ as in the soundness model, thus losing a term $\mathsf{A}^{\mathsf{Cert}}$. Then, we can rule out that $\mathcal{A}$ can determine which selection $\mathsf{DrawProver}$ has done by just performing honest executions with provers and verifiers. We use a standard hybrid argument, enabling us to progressively change the game in order to make interactions with an anonymised P_i (as selected by $\mathsf{DrawProver}$) identical to interactions with an anonymised P_j. In this argument, we progressively change each of the q executions initiated by the adversary such that in both executions we send an encryption of the same message, with a correct associated proof. Whenever $\mathcal{A}$ is set to receive from the prover selected by $\mathsf{DrawProver}$ (*i.e.*, either P_i or P_j) the values $c_* \coloneqq \mathsf{HEnc}(\mathsf{epk}, \omega \mathrm{Q}_*)$ and π_*[7], $\mathcal{A}$ receives instead the values c_i and π_i (not necessarily generated with the same randomness). We argue that this does not change $\mathcal{A}_1$'s success probability. First, note that the NIZK proofs do not reveal any information about which Q_i value is being encrypted, nor about the hidden ω (just that the encrypted value is of the form $\omega \mathrm{Q}_j$). At the expense of a term $q\mathbf{Adv}_{\mathsf{NIZK.ZK}}(\mathcal{A}_1)$, we rule out that $\mathcal{A}$ can distinguish π_i from π_j. Furthermore, at the expense of a term $q\mathbf{Adv}_{\mathsf{IND\text{-}RCCA}}(\mathcal{A}_2)$, we rule out that $\mathcal{A}$ can distinguish c_i from c_j. In the same way, we gradually replace the response strings $r \circ c_i$ and $r \circ c_j$ by $r \circ c_i$ every time, and lose a total term of $q\mathbf{Adv}_{\mathsf{IND\text{-}RCCA}}(\mathcal{A}_3)$. We now note that both transcripts are identically distributed, as they no longer depend on the auxiliary values x_i and Q_i. As a consequence, the adversary's best strategy is to guess.

Whereas the original HPO protocol is only *narrow*-strong private since the final response string s depends on x_i, our protocol is *wide*-strong private since the corresponding string S does not depend anymore on this key.

So far, the proof would apply for hbc mode as well, since $\mathcal{A}$ can only observe honest sessions and since the received values R, R_3, and y do not interfere with our reasoning above. Now, we argue that $\mathcal{A}$ gains nothing by changing any of the messages in a MIM interaction. Since the time-critical rounds do not depend on the secret key of the prover, without loss of generality we can replace the random strings R_1, R_2, R_3, the time-critical challenges, and the time-critical responses by independent and truly random values. We next rule out the possibility that $\mathcal{A}$ learns under which ephemeral public key epk the drawn prover encrypts his auxiliary value. This follows from the IK-CCA property of the encryption scheme, thus by a standard hybrid argument we lose a term $q\mathbf{Adv}_{\mathsf{IK\text{-}CCA}}(\mathcal{A}_4)$. Then, we rule out the possibility that $\mathcal{A}$ can send a verifiable NIZK proof for an encryption of a value other than one of the auxiliary values and the possibility that the sent ciphertext is anything but a possible re-randomization of the original ciphertext, losing a term $q\mathbf{Adv}_{\mathsf{NIZK.Snd}}(\mathcal{A}_5)$. Note that if $\mathcal{A}$ replaces the encrypted value Q_i by some other value (encrypting it under the same public key) and generates the corresponding NIZK-PK, the drawn prover will not be able to decrypt the received ciphertext to the correct value, since the ephemeral ω is unknown to any but the legitimate prover *irrespective of whether the adversary's re-encryption was done for the same value* Q_i *or for another one.* Thus we can assume that the first message, sent by the prover, cannot be tampered with by $\mathcal{A}$. In the same way, we can argue that $\mathcal{A}$ cannot replace the verifier's $r \circ c$ by anything that can help $\mathcal{A}$ to distinguish (we lose again a term $q\mathbf{Adv}_{\mathsf{IK\text{-}CCA}}(\mathcal{A}_4)$). At this point, the two protocol transcripts (as output by $\mathsf{Execute}$ or as collected at the end of a protocol execution) are again identically distributed. This concludes the proof for hon mode.

Finally, assume that the wide-strong adversary is in mal mode. The proof goes

[7] π_i is a NIZK-PK that the value encrypted in the ciphertext is well-formed and that the prover knows the corresponding ephemeral value ω.

exactly as for the honest case, with the only difference being that $\mathcal{A}$ knows which value r was used when sending the verifier's homomorphic encryption $r \circ \mathsf{HEnc}(\mathsf{epk}, \mathrm{Q}_i)$, and $\mathcal{A}$ can modify this value (this will make both provers fail in the same way). If some re-encryption is done under a different public key (we ruled out that $\mathcal{A}$ can learn the value of epk used by the prover), this will result in the same behavior (illegitimate) for both provers since decrypting with the wrong key will give a wrong plaintext.

Deniability. The Simulator takes as input the verifier's full internal state including y, but does not have any of the provers' secret keys. The Simulator works as follows: (1) it chooses random integers r_1, r_2, r_3, r, e and a random index $i \in \{1, \dots, k\}$, computing R_1, R_2, R_3, $r_2\mathrm{Y}$ and $[r_1 R_3]_{2n}$; (2) it generates a private/public ephemeral key pair $(\mathsf{esk}, \mathsf{epk})$; (3) it encrypts the value Q_i for the chosen i under the key epk (*i.e.*, $c \coloneqq \mathsf{HEnc}(\mathsf{epk}, \mathrm{Q}_i)$); (4) it (honestly) generates the NIZK-PK π that c encrypts the value Q_i with probability exactly $\frac{1}{k}$; (5) it honestly performs the homomorphic operation, yielding $r \circ c$; and (6) it computes the response $\mathrm{S} \coloneqq r\mathrm{Q} + eR_1 + R_2 + r_2\mathrm{Y}$. This simulation is perfect, and so we have deniability without any loss.

Impersonation resistance. In addition of proving the soundness of the protocol, we argue here that the MIM adversary cannot authenticate to an honest verifier *even in the presence of an honest prover*, except by purely relaying the lazy-phase transmissions. In the impersonation resistance model, no corruption is allowed. We assume that there exists an impersonator $\mathcal{A}$ succeeding with advantage $\mathbf{Adv}_{\mathsf{Imp}}(\mathcal{A})$. We first rule out the possibility that $\mathcal{A}$ can authenticate without simultaneous access to a prover, losing a term $\mathbf{Adv}_{\mathsf{Sound}}(\mathcal{A}_1)$. The transcript of the protocol (for the lazy rounds) has to be different in the successful impersonation attempt respectively for the adversary-prover session and for the verifier-adversary session. We exclude the possibility that $\mathcal{A}$ changes either R_1 or R_3 and manages to pass the time-critical rounds, losing a term equal to the mafia-fraud resistance of the protocol $\mathbf{Adv}_{\mathsf{mafia}}(\mathcal{A}_2)$. Then, we claim that if the ciphertext sent by the prover is modified, without modifying the proof, the verifier will detect it and abort (we lose a term $q\mathbf{Adv}_{\mathsf{NIZK.Snd}}(\mathcal{A}_3)$). Assume that $\mathcal{A}$ effectively modifies the ciphertext c, the randomness R_2, the NIZK proof π, and the verifier's response (which should contain $r \circ c$) such that the authentication succeeds. At the expense of a term $q\mathbf{Adv}_{\mathsf{IK\text{-}CCA}}(\mathcal{A}_4)$, we assume that $\mathcal{A}$ learns nothing about the ephemeral key used to encrypt the prover's ciphertext c. If $\mathcal{A}$ encrypts Q_i under a different public key, possibly also changing the verifier's response, then authentication succeeds only if $\mathcal{A}$ is able to essentially modify R_2 such that the response accounts for modification, which is equivalent to breaking the DLog assumption. □

5 Conclusion and Future Work

As our capacity to store and manipulate large amount of data increases, the risks associated with the leak of personal data are currently exploding. In this context, privacy with respect to Man-in-the-Middle adversaries in authentication and distance-bounding protocols is no longer enough to ensure sufficient privacy for the provers. In particular, in distance-bounding authentication, insiders such as the verifiers and the back-end server are able to forward large amounts of users' data to governmental agencies or to other third parties. Thus, strong privacy guarantees should also be given to provers with respect to parties interested in the tracking of users.

In this paper, we advocate the use of stronger models for prover privacy in authentication and distance bounding, and consequently introduce the notion of prover anonymity, capturing the guarantee of privacy with respect to both MIM adversaries *and* verifiers (honest-but-curious or malicious). To realize this, we have presented a protocol achieving wide-strong privacy in this sense. Furthermore, this protocol is deniable with respect to a centralized back-end server.

Whereas our scheme is mafia-fraud, distance-fraud, and impersonation resistant, it is still insecure against terrorist-fraud attacks. The attack is a type of insider attack in the sense that the prover helps (in some measure) the adversary to authenticate. The next step is to modify the protocol to gain terrorist-fraud resistance. Another important direction for further work is to investigate the minimal conditions under which a generic composition of distance-bounding protocols and authentication schemes that are private with respect to both insiders and outsiders can attain our notion of privacy.

Acknowledgements

We would like to thank the anonymous reviewers from AsiaCCS 2014 for their useful and constructive comments. Furthermore, we thank Serge Vaudenay for pointing out a weakness in our protocol with respect to MIM attacks that we have fixed in the current version of the protocol.

References

[BB05] Laurent Bussard and Walid Bagga. Distance-bounding proof of knowledge to avoid real-time attacks. In *Proc. of Security and Privacy in the Age of Ubiquitous Computing*, IFIP International Federation for Information Processing, pages 222–238. Springer-Verlag, 2005.

[BC93] Stefan Brands and David Chaum. Distance-bounding protocols. In *Proc. of Advances in Cryptology – EUROCRYPT'93*, volume 765 of *LNCS*, pages 344–359. Springer-Verlag, 1993.

[BMV13] Ioana Boureanu, Aikaterini Mitrokotsa, and Serge Vaudenay. Secure and lightweight distance-bounding. In *Proc. of the* 2^{nd} *Int. Workshop on Lightweight Cryptography for Security & Privacy (LightSec)*, volume 8162 of *LNCS*, pages 97–113. Springer-Verlag, 2013.

[CFPZ09] Céline Chevalier, Pierre-Alain Fouque, David Pointcheval, and Sébastien Zimmer. Optimal Randomness Extraction from a Diffie-Hellman Element. In *Proc. of Advances in Cryptology – EUROCRYPT'09*, number 5479 in LNCS, pages 572–589. Springer-Verlag, 2009.

[DFKO11] Ulrich Dürholz, Marc Fischlin, Michael Kasper, and Cristina Onete. A formal approach to distance bounding RFID protocols. In *Proc. of the* 14^{th} *Information Security Conference ISC*, volume 7001 of *LNCS*, pages 47–62. Springer-Verlag, 2011.

[DGB88] Yvo Desmedt, Claude Goutier, and Samy Bengio. Special uses and abuses of the fiat-shamir passport protocol. In *Proc. of Advances in Cryptology – CRYPTO'87*, volume 293 of *LNCS*, pages 21–39. Springer-Verlag, 1988.

[DLYZ11] Robert H. Deng, Yingjiu Li, Moti Yung, and Yunlei Zhao. A zero-knowledge based framework for RFID privacy. In *Journal of Computer Security*, volume 19(6), pages 1109–1146, 2011.

[FO13a] Marc Fischlin and Cristina Onete. Subtle kinks in distance-bounding: an analysis of prominent protocols. In *Proc. of the* 6th *ACM Conf. on Security and Privacy in Wireless and Mobile Networks WISec*, pages 195–206. ACM Press, 2013.

[FO13b] Marc Fischlin and Cristina Onete. Terrorism in distance bounding: Modeling terrorist fraud resistance. In *Proc. of the* 11th *Int. Conf. on Applied Cryptography and Network Security ACNS*, volume 7954 of *LNCS*, pages 414–431. Springer-Verlag, 2013.

[HPO13] Jens Hermans, Roel Peeters, and Cristina Onete. Efficient, secure, private distance bounding without key updates. In *Proc. of the* 6th *ACM Conf. on Security and Privacy in Wireless and Mobile Networks WiSec*, pages 207–218. ACM Press, 2013.

[HPVP11] Jens Hermans, Andreas Pashalidis, Frederik Vercauteren, and Bart Preneel. A new RFID privacy model. In *Proc. of the* 16th *European Symposium on Research in Computer Securitym (ESORICS)*, volume 6879 of *LNCS*, pages 568–587. Springer-Verlag, 2011.

[JW07] Ari Juels and Stephen A. Weis. Defining strong privacy for RFID. In *Proc. of the Int. Conf. on Pervasive Computing and Communications - Workshops (PerCom Workshops)*, pages 342–347. IEEE, 2007.

[MLDL09] Changshe Ma, Yingjiu Li, Robert H. Deng, and Tieyan Li. RFID privacy: relation between two notions, minimal condition, and efficient constructions. In *Proc. of the ACM Conf. on Computer and Communications Security (CCS)*, pages 54–65. ACM Press, 2009.

[NSMSN08] Ching Yu Ng, Willy Susilo, Yi Mu, and Reihaneh Safavi-Naini. RFID privacy models revisited. In *Proc. of the* 13th *European Symposium on Research in Computer Security (ESORICS)*, volume 5283 of *LNCS*, pages 251–266. Springer-Verlag, 2008.

[PH12] Roel Peeters and Jens Hermans. Wide strong private RFID identification based on zero-knowledge. Cryptology ePrint Archive, Report 2012/389, 2012.

[PV08] Radu-Ioan Paise and Serge Vaudenay. Mutual authentication in RFID: Security and privacy. In *Proc. on the* 3rd *ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, pages 292–299. ACM, 2008.

[RNTS07] Jason Reid, Juan M. Gonzalez Nieto, Tee Tang, and Bouchra Senadji. Detecting relay attacks with timing-based protocols. In *Proc. of the* 2nd *ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, pages 204–213. ACM Press, 2007.

[Vau07] Serge Vaudenay. On privacy models for RFID. In *Proc. of Advances in Cryptology – Asiacrypt'07*, volume 4883 of *LNCS*, pages 68–87. Springer-Verlag, 2007.

[Vau15] Serge Vaudenay. Private and secure public-key distance bounding - application to NFC payment. In *Proceedings of the Financial Cryptography and Data Security - 19th International Conference on Financial Cryptography and Data Security (FC)*, volume 8975 of *LNCS*, pages 207–216. Springer-Verlag, 2015.

A Soundness vs. Impersonation

Though closely related, the notions of soundness [HPVP11] and lazy-phase impersonation [DFKO11] are independent. Thus, it makes sense to prove them both separately. Clearly, impersonation resistance does not imply soundness since it models a multiprover setting, considers the security of the whole protocol and enables corruption. However, a subtle difference in the definition of lazy-phase impersonation resistance [DFKO11] is required to argue that soundness does not imply impersonation resistance. Indeed, the soundness definition [HPVP11] stipulates that the adversary does not interact with the prover during a successful impersonation attempt.

In contrast, impersonation security [DFKO11] allows this, requiring that lazy-phase transcripts of the simultaneous adversary-verifier and adversary-prover sessions are different during the impersonation attempt. This subtle difference helps an adversary if the information of a legitimate prover (or some redundancy in the protocol) may allow the adversary to change the received information in order to authenticate.

Specifically, one can easily build a separating counterexample, in which the prover's authentication response (which is session-specific and unpredictable by a MIM adversary) can be slightly altered (for instance by flipping an "unimportant" bit) to obtain a legitimate response. This protocol is sound (because the response is session specific and cannot be generated by the MIM adversary), but not lazy-phase impersonation-resistant, because the adversary can authenticate in the presence of an honest prover, without generating an identical transcript.

B Assumptions

In this appendix, we present the number-theoretic assumptions and the security properties of the primitives used for both constructions. The following computationally hard problems are based on a subgroup $\mathbb{G} =< P >$ of prime order q defined by the points of an elliptic curve $\mathcal{E}$ over a finite field $\mathbb{F}_q$. All these parameters are publicly known. Finally, let $\mathsf{dlog}(Q)$ denote the discrete logarithm of a point $Q \in \mathbb{G}$ with respect to the group generator P.

Discrete Log (DLog). Given elements P and R $= r$P, for $r \in \mathbb{Z}_q$, compute the value r.

One More Discrete Log (OMDL). Given the access to an oracle GenP generating elements and an oracle DLog solving the discrete logarithm problem in $\mathbb{G}$, after having generated m elements in $\mathbb{G}$ and used DLog at most $m-1$ times, determine the corresponding m discrete logarithms.

x-Logarithm (XL). Given an element $aP \in \mathbb{G}$, determine if its discrete logarithm a is congruent to the x-coordinate of a point $\mathrm{R} \in \mathcal{E}$.

Decisional Diffie-Hellman(DDH). Given the elements aP, bP, and $cP \in \mathbb{G}$, determine if $cP = abP$.

Oracle Diffie-Hellman (ODH). Given the elements aP, $bP \in \mathbb{G}$, a one-way hash function H, the value of $H(cP)$ and an oracle returning $H(b\mathrm{Z})$, for any point $\mathrm{Z} \neq \pm aP$, determine if $H(cP) = H(abP)$.

Extended ODH. Given the elements aP, $bP \in \mathbb{G}$, a one-way hash function H, the values of $H(cP)$ and $\mathsf{xcoord}(cP) + a$ and an oracle returning $H(b\mathrm{Z})$, for any point $\mathrm{Z} \neq \pm aP$, determine if $H(cP) = H(abP)$.

Oracle Indistinguishability with Insiders (OII). This problem has been introduced by Hermans, Peeters and Onete [HPO13]. Given $\mathcal{X} = \{x_1, \ldots, x_n\} \subset_R \mathbb{Z}_q^*$ and $\mathcal{I} = \{i_1, \ldots, i_m\} \subset_R \mathbb{Z}_q^*$ the sets of private keys of the insiders and the legitimate provers, respectively, and a counter ctr initialised to 0. At the beginning of the game, the adversary is given $\mathcal{I}$ and access to the system through three oracles. The first oracle $\mathsf{ChGen}(j,k)$ returns $(\mathsf{ctr}, \hat{r}_{\mathsf{ctr}} + x_j)$ or $(\mathsf{ctr}, \hat{r}_{\mathsf{ctr}} + x_k)$, for $\hat{r}_{\mathsf{ctr}} \in \mathbb{Z}_q^*$. The choice depends on which prover $\mathsf{DrawProver}$ has chosen. The counter ctr is then incremented and the value $\hat{r}_{\mathsf{ctr}}$ is stored in a table. The second oracle $\mathsf{TestOut}(S, \mathsf{ctr})$ returns 1 if $(\mathsf{dlog}(S - \hat{r}_{\mathsf{ctr}})P) \in \mathcal{X} \cup \mathcal{I}$ and 0 otherwise. Finally, the third oracle $\mathsf{TestX}(S)$ oracle returns 1 if $\mathsf{dlog}(S) \in \mathcal{X}$ and 0 otherwise. At the end of the game, the adversary gets $\mathcal{X}$ and must determine which prover the $\mathsf{DrawProver}$ oracle has chosen.

IND-RCCA Encryption. The notion is defined with respect to an equivalence relation such that the decryption oracle does not decrypt ciphertexts equivalent to the challenge ciphertext (we call such equivalence a "replay of the ciphertext"). The game is played between an adversary and a challenger. At the beginning of the game, the challenger generates the private/public key pair $(\mathsf{sk}, \mathsf{pk})$ by running the key-generation algorithm. Given the public key, the adversary is given access to a decryption oracle instantiated with the private key. At some point, the adversary outputs two messages, and the challenger encrypts either the first, or the second message, outputting the so-called *challenge* ciphertext. The adversary can query the decryption oracle on any ciphertext that is not a replay of the challenge ciphertext, and she must eventually guess whether the challenge ciphertext encrypts the first or the second message. The advantage of the adversary is defined by her probability to win the game minus the guessing probability of $\frac{1}{2}$.

IK-CCA Encryption. The notion of key indistinguishability stipulates that a ciphertext does not reveal anything about the public key under which the message is encrypted. In this game, the challenger generates two public keys (providing decryption under each of the two private keys), and after several decryption queries, returns a message, which will be encrypted under either the first or the second public key. After more decryption queries, the adversary has to guess under which key the challenge ciphertext was encrypted. The advantage of the adversary is her probability to win minus the guessing probability of $\frac{1}{2}$.

EUF-CMA of Signature Schemes. The usual notion of existential unforgeability against chosen message attacks (EUF-CMA) applies to signature schemes ($\mathsf{SS.KGen}, \mathsf{Sign}, \mathsf{Vfy}$) and stipulates that an adversary, given the public key generated by $\mathsf{SS.KGen}$, and access to an oracle that produces a signature $\sigma = \mathsf{Sign}(\mathsf{sk}, m)$ on the input of an arbitrary message m, should be able to produce a tuple (m^*, σ) – for a new message m^* for which the oracle has not been queried – with at most a negligible probability.

NIZK correctness (completeness). For any legitimate witness/statement pair, the verification of any proof generated with the witness for that statement should (almost) always verify. The NIZK-PK system is perfectly complete if these proofs *always* verify.

NIZK soundness. No adversary can output a valid proof for a statement outside the language.

NIZK zero-knowledge. An adversary cannot distinguish a proof returned by the prover using a legitimate witness from a proof returned by a simulator without the witness. Thus, the proof leaks no information about the witness.

C HPO Security Properties

The following lemma summarizes the results presented by Hermans, Peeters, and Onete [HPO13].

Lemma 2. *Consider the protocol in Figure 1, instantiated for a subgroup $\mathbb{G} \coloneqq < P >$ of prime order q of an elliptic curve $\mathcal{E}$ over a field $\mathbb{F}_q$. The following holds:*

- *The protocol is (perfectly) correct in the sense of [HPVP11].*
- *The protocol is sound in the sense of [HPVP11], under the One More Discrete Logarithm (OMDL) assumption. The security loss is the probability that two challenges e and e' chosen at random coincide (*i.e.*, $\binom{q}{2}2^{-q}$).*
- *The protocol has a distance-fraud resistance of (loosely) $q_{\mathsf{V}}\{\frac{3}{4}\}^n$, in which q_{V} is the number of sessions the adversary opens with the verifier. The loss here is the advantage in distinguishing the truncated DDH product from random, denoted $\mathbf{Adv}_{\mathsf{dist}}(\mathcal{A})$, which is negligible according to the results of Chevalier* et al. *[CFPZ09]. Thus, the output $p^0||p^1$ is uniformly random and the probability that $p_i^0 = p_i^1$ is $\frac{1}{2}$ for each $i = 1, \ldots n$.*
- *The protocol has a (loose) mafia-fraud resistance of $q_{\mathsf{V}}\{\frac{1}{2}\}^n$. The loss consists of a collision term for the nonces R_1, R_3, the advantage in distinguishing the truncated DDH product from random, denoted $\mathbf{Adv}_{\mathsf{dist}}(\mathcal{A})$, and the advantage of breaking the soundness of the protocol, denoted $\mathbf{Adv}_{\mathsf{Sound}}(\mathcal{A}')$.*
- *The protocol is narrow-strong and wide-forward-insider private, under the XL assumption, the ExtODH assumption, and the OII assumption. The security loss for an adversary making at most n_{DrawTag} queries to the Prover-drawing oracle consists of: a term $2n_{\mathsf{DrawTag}} \cdot (\mathbf{Adv}_{\mathsf{extODH}}(\mathcal{A}) + \mathbf{Adv}_{\mathsf{XL}}(\mathcal{A}'))$ for narrow-strong privacy (coming from a standard hybrid argument reducing a narrow-strong privacy adversary to a successful ExtODH distinguisher, and from replacing the value computed d computed by the prover by a random value), and a term $2n_{\mathsf{DrawTag}}(\mathbf{Adv}_{\mathsf{extODH}}(\mathcal{A}) + \mathbf{Adv}_{\mathsf{XL}}(\mathcal{A}') + \mathbf{Adv}_{\mathsf{OII}}(\mathcal{A}''))$ for wide-forward-insider privacy.*