Efficient, Secure, Private Distance Bounding without Key Updates

Jens Hermans Roel Peeters
KU Leuven, ESAT/COSIC & iMinds
Kasteelpark Arenberg 10
3001 Leuven, Belgium
firstname.lastname@esat.kuleuven.be

Cristina Onete
CASED & TU Darmstadt
Mornewegstrasse 30
64293 Darmstadt, Germany
cristina.onete@gmail.com

ABSTRACT

We propose a new distance bounding protocol, which builds upon the private RFID authentication protocol by Peeters and Hermans [25]. In contrast to most distance-bounding protocols in literature, our construction is based on public-key cryptography. Public-key cryptography (specifically Elliptic Curve Cryptography) can, contrary to popular belief, be realized on resource constrained devices such as RFID tags. Our protocol is wide-forward-insider private, achieves distance-fraud resistance and near-optimal mafia-fraud resistance. Furthermore, it provides strong impersonation security even when the number of time-critical rounds supported by the tag is very small. The computational effort for the protocol is only four scalar-EC point multiplications. Hence the required circuit area is minimal because only an ECC coprocessor is needed: no additional cryptographic primitives need to be implemented.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Authentication*

Keywords

RFID, Distance bounding, Privacy, Cryptographic protocol

1. INTRODUCTION

Authentication protocols are used for a wide range of applications, such as tracing goods for logistics, payment for public transport, Passive Keyless Entry and Start (PKES) systems used in cars, and personal identification for access control. While authentication protocols provide protection against impersonation, relay attacks are not considered. In relay attacks, an adversary just forwards data between the prover and the verifier. Francillon *et al.* [16] showed that for PKES, this vulnerability can be exploited in practice: one can simply drive off in another person's car by forwarding messages between the car and the owner's passive authentication device.

In order to prevent relay attacks (also called mafia fraud by Desmedt [13]), Brands and Chaum [7] proposed the first distance-bounding protocol, using the fact that pure relaying over a large distance introduces a processing delay for the adversary, which the reader can detect if equipped with a clock. Most distance-bounding protocols are round based and may be grouped into phases, which are called either lazy (no clock is used) or time-critical. In time-critical phases, the roundtrip time between sending a challenge and receiving a response is measured; the measured timings provide an upper bound on the distance between the communicating parties.

There are four main security threats for distance bounding protocols:

1. **Impersonation Security.** The adversary attempts to impersonate the prover during the lazy phases, but without pure relay.
2. **Distance Fraud.** The adversary is a malicious prover that tries to prove that it is closer to the honest verifier than it really is.
3. **Mafia Fraud.** The adversary impersonates an honest prover in the presence of this prover and the honest verifier.
4. **Terrorist Fraud.** The adversary impersonates a prover with that prover's aid to the honest verifier. However, the adversary is unable to impersonate the prover when unaided.

Lazy phase impersonation security was only recently classified as a desirable property. In general, impersonation security used to be achieved only during time-critical rounds. However, as noted by Avoine and Tchamkerten [4], resource constrained devices such as RFID tags cannot support many time-critical rounds and thus lazy-phase impersonation resistance is required.

Distance fraud is quite easy to achieve when considered independently of other properties. Optimal distance-fraud resistance can be achieved by having the verifier send random challenge bits to the prover and waiting for the prover to echo them back. However, any party within the legitimate distance (whether or not in possession of legitimate credentials) can echo challenge bits in time. This breaks in particular mafia-fraud resistance. In order to attain mafia-fraud resistance, distance-bounding protocols in the symmetric setting usually employ a PseudoRandom Function

WiSec'13, April 17-19, 2013, Budapest, Hungary.

(PRF), returning bits from the PRF's output depending on the reader's challenges. However, Boureanu *et al.* [6] recently showed that the PRF assumption is not sufficient to prove distance-fraud resistance in such cases.

Distance-bounding protocols in the literature address two or more of the above threats [4, 7, 10, 18, 20, 21, 28, 30]. Note that such protocols are also hard to design, since they should be as lightweight as possible so as to be implementable on resource-constrained devices. Furthermore, one needs to take measures to keep the processing at the device as small as possible. Rasmussen *et al.* [27] and Ranganathan *et al.* [26] proposed the first practical implementations of such protocols by using analog components, which allows for the necessary small processing delay.

That most of the proposed lightweight authentication and distance-bounding protocols use symmetric cryptography, is a result of the myth that public-key cryptography cannot be implemented on resource constrained devices such as RFID tags. However, Lee *et al.* [22, 23] and Wenger and Hutter [33] showed that public-key cryptography *can* be implemented. More specifically, these papers proposed efficient dedicated coprocessors that can do elliptic curve arithmetic on curves suitable for Elliptic Curve Cryptography (ECC).

Most distance bounding protocols do not address privacy. An exception is the Swiss-Knife protocol of Kim *et al.* [21], where tags have a shared secret with the reader which is used for authentication. However, from the moment the prover's secret is compromised, the tag is traceable, since the shared secret is never updated. In recent work Onete [24] shows how to achieve private distance bounding, but only with key updates; the same approach is taken by Yang *et al.* [34], whose protocol also fails to attain the claimed distance-fraud and terrorist-fraud resistance [15].

Vaudenay [32] showed that stronger privacy requires key agreement, and thus cannot be achieved in the symmetric setting (as is the case in [34]). Intermediate privacy levels can be achieved in the symmetric setting with key updates. However, key updates are unsuitable for resource-constrained devices as the write operation typically requires high energy. Moreover, protocols relying on key updates can be vulnerable to desynchronization attacks.

The notion of 'wide-forward-insider' [25] privacy covers the case where an adversary uses the internal state from a corrupted tag to attack the privacy of other tags. These insider attacks where described by van Deursen *et al.* [31], clearly showing that wide-forward privacy protocols are not sufficient. For two wide-forward private protocols it was shown that the adversary can link uncorrupted tags if he can to learn the outcome of the protocol and the state of a legitimate 'insider' tag. Note that an adversary can easily get a legitimate e.g. a legitimate public transportation ticket.

Our Contribution. We propose the first distance-bounding protocol that attains wide-forward-insider privacy in the sense of Hermans *et al.* [19]. Our protocol relies on the recently-proposed public-key (ECC) secure, wide-forward-insider RFID authentication protocol due to Peeters and Hermans [25], such that the resulting protocol resists distance- and mafia-fraud attacks while remaining secure and wide-forward-insider private. The proposed scheme has nearly optimal mafia-fraud resistance and a very high impersonation security, resulting from the soundness of the underlying protocol.

2. PRELIMINARIES

Our work builds on the work of Peeters and Hermans [25]. For this reason we briefly recall the privacy model and definitions used in [25] in the following sections. We also give an overview of the definitions related to distance bounding, as defined by Dürholz et al. [14].

2.1 Notation

To relate the security level of a protocol to the security of the underlying cryptographic primitives or number theoretic assumptions, we often transform a successful adversary $\mathcal{A}$ into one or more adversaries $\mathcal{A}'$ against the primitive(s) or assumption(s). Let $\mathbf{Adv}_{\mathcal{S}}^{\mathrm{Exp}}(\mathcal{A}')$ denote the advantage of an adversary $\mathcal{A}'$ in breaking a cryptographic scheme or assumption $\mathcal{S}$ in some experiment Exp. An overview of the relevant number theoretic assumptions is given in Sect. 2.2.

In this paper we consider elliptic curves $\mathbb{E}$ and subgroups $\mathbb{G}_\ell$ of points on $\mathbb{E}$ of prime order ℓ over $\mathbb{F}_p$, usually generated by a point P. Points on the elliptic curve are denoted by uppercase characters. In general, we denote scalars by lowercase letters. We denote by aP the scalar multiplication of the point P by the scalar $a \in \mathbb{Z}_\ell^*$. For a any scalar $x \in \mathbb{Z}_\ell^*$, the corresponding uppercase letter X is defined as xP. The key-generation algorithm of our scheme outputs a pair $(priv, Pub)$ such that $priv \in_R \mathbb{Z}_\ell^*$ and $Pub = priv \cdot P \in \mathbb{G}_\ell$. We denote by O the point at infinity of the elliptic curve.

Our construction relies on the $\texttt{xcoord}(\cdot)$ function, which is the DSA conversion function [8]. This function, returns the x-coordinate of a point. For a point $Q = \{q_x, q_y\} \in \mathbb{G}_\ell$, with $q_x, q_y \in [0 \dots p-1]$, $\texttt{xcoord}(Q)$ maps Q to $q_x \mod \ell$. Additionally, we define $\texttt{xcoord}(O) = 0$.

For a bitstring x, we denote by $[x]_k$ the least significant k bits of the string.

2.2 Number Theoretic Assumptions

Discrete Logarithm (DL). Let A be a given, arbitrarily chosen element of $\mathbb{G}_\ell$. The discrete logarithm (DL) problem is to find the unique integer $a \in \mathbb{Z}_\ell^*$ such that $A = aP$. The DL assumption states that it is computationally hard to solve the DL problem.

One More Discrete Logarithm (OMDL). The one more discrete logarithm (OMDL) problem was introduced by Bellare *et al.* [5]. Let $\mathcal{O}_1()$ be an oracle that returns random elements $A_i = a_i P$ of $\mathbb{G}_\ell$, and let $\mathcal{O}_2(\cdot)$ be an oracle that returns the discrete logarithm of a given input base P. The OMDL problem is to return the discrete logarithms for each of the elements obtained from the m queries to $\mathcal{O}_1()$, while making strictly less than m queries to $\mathcal{O}_2(\cdot)$ (with $m > 0$). The OMDL assumption is that it is computationally hard to solve the OMDL problem.

x-Logarithm (XL). Brown and Gjøsteen [9] introduced the x-Logarithm (XL) problem: given an elliptic curve point, determine whether its discrete logarithm is congruent to the x-coordinate of an elliptic curve point. The XL assumption states that it is computationally hard to solve the XL problem. Brown and Gjøsteen also provided some evidence that the XL problem is almost as hard as the DDH problem (see below).

Diffie Hellman (DH). Let aP, bP be any two given arbitrary elements of $\mathbb{G}_\ell$, with $a, b \in \mathbb{Z}_\ell^*$. The computational Diffie Hellman (CDH) problem is, given P, aP and bP, to

find abP. The 4-tuple $\langle P, aP, bP, abP\rangle$ is called a Diffie Hellman (DH) tuple. Given a fourth element $cP \in \mathbb{G}_\ell$, the decisional Diffie Hellman (DDH) problem is to determine if $\langle P, aP, bP, cP\rangle$ is a valid Diffie-Hellman tuple or not. The DDH assumption states that it is computationally hard to solve the DDH problem.

Oracle Diffie Hellman (ODH). Abdalla *et al.* [1] introduced the ODH assumption:

Definition 1 *Oracle Diffie Hellman (ODH) Given $A = aP$, $B = bP$, a function H and an adversary $\mathcal{A}$, consider the following experiments:*

Experiment $\mathbf{Exp}^{odh}_{H,\mathcal{A}}$ *:*

- $\mathcal{O}(Z) := H(bZ)$ *for* $Z \neq \pm A$
- $g = \mathcal{A}^{\mathcal{O}(\cdot)}(A, B, H(C))$
- *Return* g

The value C is equal to abP for the $\mathbf{Exp}^{odh-real}_{H,\mathcal{A}}$ *experiment, chosen at random in $\mathbb{G}_\ell$ for the* $\mathbf{Exp}^{odh-random}_{H,\mathcal{A}}$ *experiment.*

We define the advantage of $\mathcal{A}$ violating the ODH assumption as:

$$|\Pr\left[\mathbf{Exp}^{odh-real}_{H,\mathcal{A}} = 1\right] - \Pr\left[\mathbf{Exp}^{odh-random}_{H,\mathcal{A}} = 1\right]|.$$

The ODH assumption consists of the plain DDH assumption combined with an additional assumption on the function $H(\cdot)$. The idea is to give the adversary access to an oracle $\mathcal{O}$ that computes bZ, without giving the adversary the ability to compute bA, which can then be compared with C. To achieve this one restricts the oracle to $Z \neq \pm A$, and moreover, only $H(bZ)$ instead of bZ is released, to prevent the adversary from exploiting the self reducibility of the DL problem.[1] The crucial property that is required for $H(\cdot)$ is one wayness. In the following part we use a one way function based on the DL assumption. We define the function $H(Z) := \texttt{xcoord}(Z)P$.

Theorem 1 *The function $H(\cdot)$ is a one-way function under the DL assumption.*

2.3 Privacy Model

Hermans *et al.* [19] provided a general game-based privacy model for RFID,which is robust and easy to apply. For more details on the different existing RFID privacy models and a comparison between these, the reader is referred to [19].

The intuition behind the RFID privacy model of Hermans *et al.* is that of tag indistinguishability, i.e. privacy is guaranteed if an adversary cannot distinguish with which one of two RFID tags (of its choosing) it is interacting by means of a set of oracles. The main ideas of this model resemble previous frameworks: the adversary interacts with the tags by means of handles, called virtual tags (vtags), and privacy is defined as an indistinguishability game (or experiment **Exp**) between a challenger and the adversary.

This game is defined as follows. First the challenger picks a random challenge bit b and then sets up the system $\mathcal{S}$ with a security parameter k. Next, the adversary $\mathcal{A}$ can use a subset (depending on the privacy notion) of the following oracles to interact with the system:

- `CreateTag`$(ID) \rightarrow T_i$: on input a tag identifier ID, this oracle creates a tag with the given identifier and corresponding secrets, and registers the new tag with the reader. A reference T_i to the new tag is returned.
- `Launch`$() \rightarrow \pi$: this oracle launches a new protocol run on the reader R_j, according to the protocol specification. It returns a session identifier π, generated by the reader.
- `DrawTag`$(T_i, T_j) \rightarrow vtag$: on input a pair of tag references, this oracle generates a virtual tag reference, as a monotonic counter, $vtag$ and stores the triple $(vtag, T_i, T_j)$ in a table $\mathcal{D}$. Depending on the value of b, $vtag$ either refers to T_i or T_j. If one of the two tags T_i or T_j is in the list of insider tags $\mathcal{I}$, $\perp$ is returned and no entry is added to $\mathcal{D}$. If T_i is already references as the left-side tag in $\mathcal{D}$ or T_j as the right-side tag, then this oracle also returns $\perp$ and adds no entry to $\mathcal{D}$. Otherwise, it returns $vtag$.
- `Free`$(vtag)_b$: on input $vtag$, this oracle retrieves the triple $(vtag, T_i, T_j)$ from the table $\mathcal{D}$. If $b = 0$, it resets the tag T_i. Otherwise, it resets the tag T_j. Then it removes the entry $(vtag, T_i, T_j)$ from $\mathcal{D}$. When a tag is reset, its volatile memory is erased. The non-volatile memory, which contains the state S, is preserved.
- `SendTag`$(vtag, m)_b \rightarrow m'$: on input $vtag$, this oracle retrieves the triple $(vtag, T_i, T_j)$ from the table $\mathcal{D}$ and sends the message m to either T_i (if $b = 0$) or T_j (if $b = 1$). It returns the reply from the tag (m'). If the above triple is not found in $\mathcal{D}$, it returns $\perp$.
- `SendReader`$(\pi, m) \rightarrow m'$: on input π, m this oracle sends the message m to the reader in session π and returns the reply m' from the reader (if any) is returned by the oracle.
- `Result`(π): on input π, this oracle returns a bit indicating whether or not the reader accepted session π as a protocol run that resulted in successful authentication of a tag. If the session with identifier π is not finished yet, or there exists no session with identifier π, $\perp$ is returned.
- `Corrupt`(T_i): on input a tag reference T_i, this oracle returns the complete internal state of T_i. Note that the adversary is not given control over T_i.
- `CreateInsider(ID)` $\rightarrow T_i, S$: create an insider tag T_i. This runs `CreateTag` to create a new tag T_i and `Corrupt` on the newly created tag. The tag T_i is added to the list $\mathcal{I}$ of insider tags.

By using the `DrawTag` oracle the adversary $\mathcal{A}$ can arbitrarily select which two tags to interact with. Based upon the challenge bit b chosen initially, a virtual tag is then associated to either the 'left' tags T_i or the 'right' tags T_j. At the end of the adversary's interaction, $\mathcal{A}$ outputs a guess bit g. The outcome of the game will be $g \stackrel{?}{=} b$, *i.e.*, 0 for an incorrect and 1 for a correct guess. Thus, the adversary wins the game if it can distinguish whether it has interacted with the 'left' or the 'right' world.

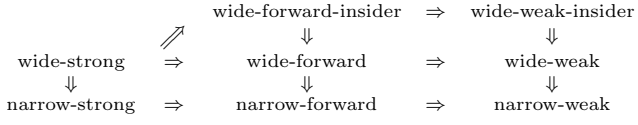
The advantage of the adversary $\mathbf{Adv}_{\mathcal{S},\mathcal{A}}(k)$ is defined as:

$$\left|Pr\left[\mathbf{Exp}^0_{\mathcal{S},\mathcal{A}}(k) = 1\right] + Pr\left[\mathbf{Exp}^1_{\mathcal{S},\mathcal{A}}(k) = 1\right] - 1\right|.$$

[1] The adversary can set $Z = rA$ for a known r and compute $r^{-1}(bZ) = bA$.

The following privacy notions were introduced by Vaudenay [32] and are also present in Hermans *et al.*'s framework. *Strong* attackers are allowed to use all the oracles available. *Forward* attackers are only allowed to do other corruptions after the first corruption, protocol interactions are no longer allowed. *Weak* attackers cannot corrupt tags. Independently of these classes, there is the notion of *wide* and *narrow* attackers. A *wide* attacker is allowed to get the result from the reader, *i.e.* whether the identification was successful or not; while a *narrow* attacker does not.

If an adversary is allowed to call `CreateInsider` the privacy notion is called 'insider', so we can speak of forward-insider and weak-insider adversaries. For strong and destructive the `CreateInsider` can be simulated using the normal `CreateTag` and `Corrupt` oracles, i.e. strong-insider and destructive-insider are equivalent to strong and destructive respectively. The privacy notions are related as follows:



We use arrows between two notions to denote that any protocol that is private in the sense of the first notion is also private in the sense of the second notion.

For most practical applications, wide-forward-insider privacy is sufficient. By contrast, the weaker notion of wide-forward privacy is *not* sufficient; indeed, in e.g. transportation systems an adversary has easy access to an insider tag and can thus abuse any privacy guarantees of the system. Furthermore, it seems that the wide-strong notion captures a scenario exceeding the practical requirements for privacy, where an adversary may first corrupt a tag and then release it again for future tracking. However, in practice this can be done more easily, without physically tampering with the tag itself (i.e. corrupting it). For instance the attacker could, when having physical access to the tag, attach his own tracking device to it.

Note that we further restrict the `Corrupt` oracle, such that it only returns the non-volatile state of the tag. This restriction allows to exclude trivial privacy attacks on multi-pass protocols, that require the tag to store some information in volatile memory during the protocol run.

2.4 Private Authentication Protocol

The definition of a private authentication protocol is due to Peeters and Hermans [25]. This definition is specific for the RFID setting in the sense that it assumes that concurrent attacks are impossible, since tags can only participate in one session at the time. Furthermore their security definition does not model physical distance, as a result relay attacks are not considered.

Private authentication protocols have the following three properties: *correctness*, *soundness*, and *privacy*. Correctness and soundness are necessary to establish the security of the authentication protocol. Correctness ensures that the reader (verifier) does not reject legitimate tags (provers). Soundness ensures that an illegitimate tag (*i.e.* an adversary not in possession of legitimate credentials) is always rejected by the reader. Privacy will ensure that all parties cannot infer any information on the tag's identity from the protocol messages, except the reader to which the tag is authenticating. Only the content of the exchanged messages is taken into account, not the physical characteristics of the radio links as studied by Danev *et al.* [12], which should be dealt with at the hardware level.

Definition 2 *Correctness. A scheme is correct if the authentication of a legitimate tag only fails with negligible probability.*

Definition 3 *Soundness. A scheme is resistant against impersonation attacks if no polynomially bounded strong adversary succeeds, with non-negligible probability, in being authenticated by the reader as the tag it impersonates. Adversaries may interact with the tag they want to impersonate prior to, and with all other tags prior to and during the protocol run. All tags, except the impersonated tag, can be corrupted by the adversary.*

Definition 4 *Privacy. A privacy protecting protocol, modeled by the system $\mathcal{S}$, is said to computationally provide privacy notion X, if and only if for all polynomially bounded adversaries $\mathcal{A}$, it holds that $\boldsymbol{Adv}^X_{\mathcal{S},\mathcal{A}}(k) \leq \epsilon$, for negligible ϵ.*

2.5 Distance Bounding

The security model of Dürholz *et al.* [14], which formalizes security notions for distance-bounding protocols (in particular taking into consideration relay attacks) considers a single verifier and a single prover, in particular for the RFID setting. Here, the single prover $\mathcal{P}$ is an RFID tag and the verifier $\mathcal{V}$ is the reader. The reader uses a clock to measure the time elapsed between sending a challenge and receiving the response. Dürholz et al. consider round-based distance-bounding protocols, where rounds are called *time-critical* if a clock is used to measure the roundtrip time, and *lazy* otherwise.

In the following, we provide intuitive descriptions of impersonation security, mafia fraud, distance fraud, and terrorist fraud. For the formal definitions and for further insight, we refer the reader to the original paper [14].

Impersonation Resistance. Impersonation resistance refers to lazy-phase tag authentication. The idea, introduced by Avoine and Tchamkerten [4], is that even without the time-critical phases (relay attacks are not considered), the prover is still authenticated. By contrast, Avoine *et al.* [2] define impersonation security for the entire protocol. Note that impersonation resistance as defined by Avoine *et.al.* is also achieved by protocols that are lazy-phase impersonation secure and mafia-fraud resistant.

Distance Fraud. Distance-fraud adversaries control the tags themselves. The adversary is further away than allowed from the reader, but aims to convince the reader of the contrary. Since the reader's clock measures time accurately, the adversary must anticipate the reader's challenges and respond in advance.

Mafia Fraud. Mafia-fraud resistance considers a Man-In-The-Middle (MITM) attack, where pure relay is prevented by the reader's clock. Informally, the attacker consists of two parts: a *leech*, which impersonates the reader to an honest tag, and a *ghost*, which impersonates the tag to an honest reader. Both the reader and the honest tag are unaware of the MITM attack.

Terrorist Fraud. In terrorist-fraud attacks, the dishonest prover cooperates with an adversary in order to enable this adversary to authenticate. The informal restriction is that the prover does not forward trivial data, like the secret key. This attack is rather controversial, as we discuss it at length in Section 3.2.

Attacks in [14]. All the attacks above are formalized by Dürholz et al. [14] by introducing an abstract clock, which keeps track of the messages sent in several protocol executions called sessions. These sessions can be: reader-tag (the adversary is a passive eavesdropper), reader-adversary (the adversary impersonates the tag to the reader), and adversary-tag (the adversary impersonates the reader to the tag). Relaying is considered round-wise. In mafia-fraud attacks, a phase is called *tainted* if the adversary purely relays communication between a reader-adversary and an adversary-tag session. Here, *pure* relay refers to an adversary receiving a message in a session sid and *then* relaying the exact, same message in a session sid'. Having received a response, the adversary relays it back again between sid' and sid, for all subsequent rounds in the tainted phase. If the adversary changes any of the messages in one session before it forwards them in the other session, this is not pure relay. Also, if the adversary queries one session with some message m before receiving the same m in the other session, this is not relaying. In distance fraud, phases are *tainted* if the adversary does not commit in advance to the responses of time-critical phases before the phase has started. In terrorist fraud, the adversary taints a time-critical phase by querying the adversary during that phase.

Attack Parameters. Apart from the upper bound $t_{\max}$ of the roundtrip transmission time and the number of time-critical rounds n, we also allow for at most $T_{\max}$ phases with delayed responses and at most $E_{\max}$ phases with wrong responses. Though most existing protocols do not provide for erroneous/delayed communication, fault tolerance is essential in resource-constrained environments, e.g. RFID.

When specifying the adversary's characteristics one considers its runtime t and the number $q_{\mathcal{V}}$, $q_{\mathcal{P}}$,q_{OBS} of respectively reader-adversary, adversary-tag, and reader-tag sessions.

3. THE PROTOCOL

In several distance-bounding protocols (e.g., Hancke-Kuhn [18]), the tag and reader use a long-term shared secret to compute an ephemeral, session-specific shared secret. Afterwards, during each of the n time-critical rounds, the reader sends a challenge bit and expects a single response bit, either from the left or from the right half of the computed ephemeral value, depending on the challenge. This ephemeral secret is the result of a PseudoRandom Function (PRF), often instantiated with an H-MAC. Our proposed protocol follows this structure; very importantly, however, our protocol is in the asymmetric setting. Recall from the introduction that the need for the asymmetric setting arises from our desire to design a protocol without key updates[2] that guarantees strong privacy and also protects against distance fraud.

Our proposed protocol is depicted in Fig. 1. All tags are initialized with a private/public key pair $(x, X = xP)$ and the tags' public keys are registered in the reader's database. The reader's private/public key pair is $(y, Y = yP)$ of which the public key is known to all tags.

To generate the ephemeral shared secret, an anonymous Diffie-Hellman key agreement, with fresh random values from both sides ($R_1 = r_1P$ and $R_3 = r_3P$), takes place, resulting in a shared point r_1r_3P on the elliptic curve. To map this point to a uniformly distributed element in $\mathbb{Z}_\ell^*$, a cryptographic hash function can be used. Unfortunately, current hash functions [29] require at least 50% of the circuit area of the most compact ECC coprocessor implementation. Instead, we propose to use the ECDSA conversion function [8], which comes almost for free when using elliptic curves. This function simply returns the x-coordinate of a point on the elliptic curve. Note that the set of x-coordinates does not span $\mathbb{Z}_\ell^*$ entirely, as such the x-coordinates are not uniformly-randomly distributed in $\mathbb{Z}_\ell^*$. However, we only need $2n$ bits. Chevalier *et al.* [11] showed that binary truncation of the x-coordinate of the last element of an instance of the DDH problem is statistically indistinguishable from the uniform distribution :

$$\langle aP, bP, U_k\rangle \approx_S \langle aP, bP, \text{lsb}_k(\text{xcoord}(cP))\rangle\,.$$

If there are no transmission errors and no Man-in-the-Middle (MITM) interference, then $t^0||t^1 = u^0||u^1$; these values will be used by the tag to answer the reader's subsequent challenges. Now the reader chooses a random challenge $e \in \mathbb{Z}_\ell^*$ and sends the first n bits, one per round, as its time critical challenges, expecting a bit from the corresponding response vector, i.e. for a challenge bit b sent in the i^{th} round, the tag should respond with t_i^b. The round trip time is measured and compared with a maximal round trip time t_{max}.

Finally, the protocol ends in a second lazy phase, in which the last messages of the underlying private authentication protocol are broadcast. This underlying private authentication protocol is due to Peeters and Hermans [25] and has the following structure: commit, exam, response. The tag's commitment is now the point R_2 sent in the first lazy phase. We cannot reuse the point R_1; indeed, if R_1 is used, an attacker could impersonate the verifier and send the prover Y instead of a random r_3P, thus having a better probability to distinguish the tag. The reader's full exam value e is sent to the tag, which in turn compares this to the received bit challenges in the time-critical phase. As such, we can enforce a higher level of mafia-fraud resistance. The tag must also verify that $e \neq 0$, to prevent trivial attacks. The response is similar to the Schnorr authentication protocol, providing a very high level of impersonation resistance. To achieve privacy, the response contains an additional blinding factor d. This blinding factor is computed using a static DH key exchange, with the randomness r_2 of the tag it already committed to (by sending R_2) and the public key Y of the reader. To map this point r_2Y to a scalar while breaking the homomorphisms that exist between the input and the output, again the `xcoord`$(\cdot)$ function is used. Due to the non-uniformity subgroup of x-coordinates in $\mathbb{Z}_\ell^*$, a privacy adversary could build a distinguisher. However, this adver-

[2]For tags updating their keys, it is important that the reader and tag stay synced, meaning that measures should be taken to prevent desynchronization attacks. Moreover, updating keys requires high energy.

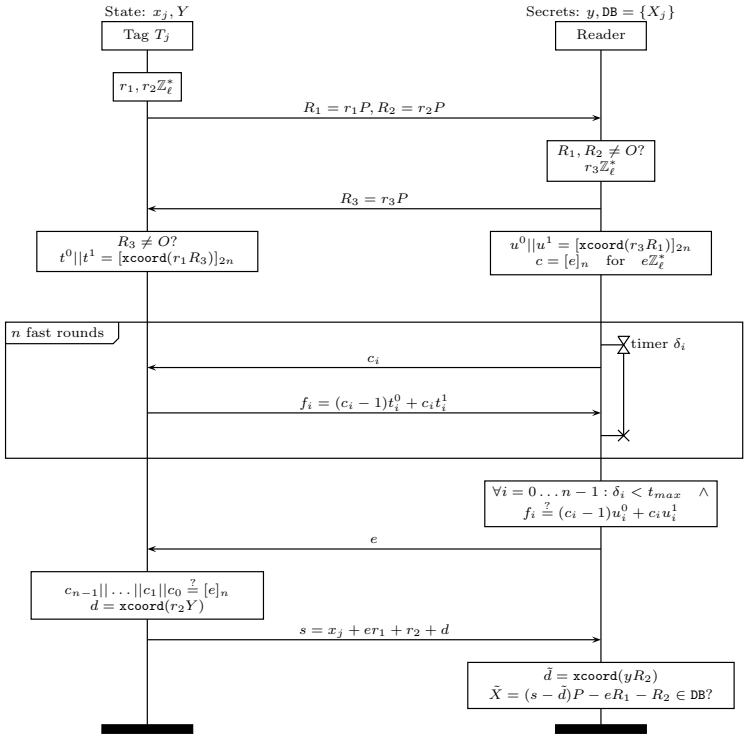


Figure 1: Efficient, secure, wide-forward-insider private distance bounding protocol.

sary has no information on d directly, only on $d + r_2$. Given the XL assumption, this poses no threat.

3.1 Properties

Let DB be the distance-bounding authentication scheme in Fig. 1 with parameters $(t_{\max}, n)$. We proceed to give the formal security statements for this protocol.

Impersonation Resistance. This property follows directly from the correctness and soundness of the underlying private authentication protocol, due to Peeters and Hermans [19]. As a result, our protocol has very high impersonation security that does not depend on the number of performed round in the time-critical phase.

Theorem 2 (Correctness) DB *is correct in the sense of Def. 2.*

The proof of correctness is trivial due to the fact that our protocol does not make use of key updates. As a result, desynchronization attacks need not to be taken into account. Therefore, this proof is omitted.

Theorem 3 (Soundness) DB *is sound according to Def. 3 under the OMDL assumption.*

Proof. Assume an adversary $\mathcal{A}$ that can break the extended soundness with non-negligible probability, i.e. that can perform a fresh, valid authentication with the verifier. Without loss of generality we will assume the target tag is known at the start of the game. [3] We construct an adversary $\mathcal{B}$ that wins the OMDL game as follows:

- Set $X = \mathcal{O}_1()$; this value will be used as the public key of the target tag.
- $\mathcal{B}$ executes $\mathcal{A}$. During the first phase of $\mathcal{A}$, $\mathcal{B}$ simulates the `SendTag` oracles for the target tag as follows (all other oracles are simulated as per protocol specification):
  - On the first $\mathtt{SendTag}(vtag)$ query of the i'th protocol run:
    return $R_{1,i} = \mathcal{O}_1()$ and $R_{2,i} = r_{2,i} P$.
  - On the third $\mathtt{SendTag}(vtag, e_i)$ query of the i'th protocol run:
    set $d_i = \mathtt{xcoord}(y R_{2,i})$ and return $s_i = \mathcal{O}_2(X + d_i P + e_i R_{1,i} + R_{2,i})$
- During the second phase of $\mathcal{A}$, $\mathcal{B}$ proceeds as follows:
  - On the first call of $\mathcal{A}$ to $\mathtt{Result}(\pi)$, compute $d = \mathtt{xcoord}(y R_2)$ and store (s, d). Next, rewind $\mathcal{A}$ until right before the call to $\mathtt{SendReader}(\pi, R)$.

[3] Otherwise, the proof can be adapted by choosing the public keys of the tags as $X_i = \mathcal{O}_1()$. All tag queries are simulated as for the target tag, until the tag is corrupted. When corrupting a tag, call $\mathcal{O}_2(X_i)$ for that tag and use the result as private key for simulating all following queries to that tag. At the end of the game, use the $\mathcal{O}_2(\cdot)$ oracle to extract all remaining discrete logarithms, except for the target tag.

On the next call to $\texttt{SendReader}(\pi, R)$, return a new random e'.

– On the next call of $\mathcal{A}$ to $\texttt{Result}(\pi)$: compute $r_1 = {}^{(s-s')}/_{(e-e')}$ and $x = s - d - er_1 - r_2$ return $(x, e_1^{-1}(s_1 - x - d_1 - r_{2,1}), \ldots, e_k^{-1}(s_k - x - d_k - r_{2,k}))$.

The simulation by $\mathcal{B}$ is perfect during both phases. At the end of the game $\mathcal{B}$ will successfully win the OMDL with non-negligible probability, unless $s = s'$, which happens with negligible probability since both e and e' are randomly chosen after $R \neq O$ is fixed. □

Distance-Fraud Resistance. Intuitively, distance-fraud resistance requires both the unpredictability of challenges and that the response has sufficient entropy, even with respect to a party having the secret key, i.e. a dishonest prover. The flaws in the proofs for distance-fraud resistance as identified by Boureanu *et al.* [6], have not yet been resolved in the symmetric setting. By contrast, in our case, we use a public-key setting, where the ephemeral secret is the truncation of the x-coordinate of a point on the elliptic curve. The prover first selects an integer nonce r_1 and sends the value $R_1 = r_1 P$. Then the verifier (honestly) selects another nonce r_3 and truncates the x-coordinate of $r_3 R_1$; the output is then a bitstring which is distributed according to the uniform random distribution.

Theorem 4 (Distance-Fraud Resistance) *For any $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$ distance fraud adversary $\mathcal{A}$, it holds that*

$$\boldsymbol{Adv}^{dist}_{\mathsf{DB}}(\mathcal{A}) \leq q_{\mathcal{V}} \cdot \left(\tfrac{3}{4}\right)^n .$$

Proof. First we argue that $r_1 \cdot r_3$ can be replaced by a random integer r. Second, we argue that the bits in the binary truncation $t^0||t^1$ are distributed according to the random distribution. Finally, we argue that the given bound holds. The first part holds because, on the one hand, the reader is honest in this attack and thus r_3 is chosen an random *after* the tag has committed to r_1, and on the other hand because the reader checks that $R_1 \neq O$, thus that $r_1 \neq 0$. The second part holds in view of the results of Chevalier et al. [11]. Thus, for every i, it holds that $Pr[t_i^0 = t_i^1] = \frac{1}{2}$. Finally, it holds that the adversary wins every round where $t_i^0 = t_i^1$ with probability 1. However, if $t_i^0 \neq t_i^1$, the adversary has only guessing probability to win, i.e. $\frac{1}{2}$. This adds up to a success probability of $\frac{3}{4}$ per round. Accounting for $q_{\mathcal{V}}$ attempts, we have the bound above. □

Mafia-Fraud Resistance. The basic-most requirement for mafia-fraud resistance is that the ephemeral secret is hard to compute without knowing the long-term secret key. However, in order to increase mafia-fraud resistance we need to prevent the adversary from performing a MITM attack, which we call the Go-Early strategy following the notation of Dürholz et al. [14] and Fischlin and Onete [15]. Briefly, this attack works as follows: having first forwarded the lazy-phase messages between a prover and a verifier (i.e. the adversary opens a reader-adversary session sid and an adversary-prover session sid^* that are "related" in the sense that the time-critical responses will be the same), the MITM adversary will then be queried by the reader with a challenge bit c and will expect a response bit r. However, in the Go-Early strategy, the adversary first queries the prover in session sid^* with a random bit c^*, receiving r^* in response, and will use this response to answer to the reader subsequently, in session sid. In other words, if $c = c^*$, then the adversary wins the round by forwarding $r = r^*$, else, if $c = c^*$, it guesses the correct response with probability $\frac{1}{2}$, totaling a success probability of $\frac{3}{4}$ per round. In our protocol we reduce this success probability by using a strategy similar to [7, 21], i.e. we add a lazy authentication phase depending on the challenges received by the prover. Thus, as soon as the adversary mis-guesses one challenge, it makes the prover compute a different response in this lazy phase, which cannot be used by the adversary in session sid. In particular, we merge authentication with distance bounding and use the reader's challenge bits in order to compute the authentication string.

Theorem 5 (Mafia-Fraud Resistance) *For any $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$-mafia-fraud adversary $\mathcal{A}$ against the scheme there exist: a (t', q')-distinguisher $\mathcal{A}'$ that can distinguish the truncated output of the x-coordinate of the last element of a DDH element from random; an adversary $\mathcal{A}''$ that can solve the DL problem; and an adversary $\mathcal{B}$ against the soundness of the underlying protocol such that:*

$$\begin{aligned}\boldsymbol{Adv}^{mafia}_{\mathsf{DB}}(\mathcal{A}) \quad \leq \quad & q_{\mathcal{V}} \cdot \left(\tfrac{1}{2}\right)^n + \binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-\ell} + \\ & \boldsymbol{Adv}_{\mathsf{dist}}(\mathcal{A}') + 2q_{\mathcal{V}} \boldsymbol{Adv}_{DL}(\mathcal{A}'') + \\ & \boldsymbol{Adv}_{\mathsf{Sound}}(\mathcal{B}) + \binom{q_{\mathcal{P}}}{2} \cdot 2^{-\ell},\end{aligned}$$

with ℓ the order of the elliptic curve subgroup $\mathbb{G}_\ell$.

Proof. The proof proceeds as follows:

1. We show that one can safely replace the output $T^0||T^1$ by truly random values, for each new nonce pair (r_1, r_3).
2. Show that nonce pairs are (almost) unique, except for possibly one adversary-tag session sid^* having the same nonce pair as a reader-adversary session sid (here the adversary relays the nonces between sessions).
3. Bound the probability that the adversary passes the time-critical phases for at most one adversary-tag interaction.

The first step goes as follows. First, note that the adversary can learn the values r_1 and r_3 (and thus compute the ephemeral secret by using the public keys) with at most probability $2\mathbf{Adv}_{\mathsf{DL}}(\mathcal{A}''')$ per authentication attempt. In this case, the adversary can bypass tainting the phase by querying the prover *after* it has successfully completed the time-critical phases, in order to learn the final authentication string. We assume now that the adversary cannot guess these values. By the results of Chevalier *et al.* [11], indicating that binary truncation of the x-coordinate of the last element of an instance of the DDH problem is statistically indistinguishable from the uniform distribution. Thus, replacing $t_0||t_1$ by random values decreases the adversary's success probability by at most $\mathbf{Adv}_{\mathsf{dist}}(\mathcal{A}')$.

Next we consider all the nonces appearing in an attack of the adversary $\mathcal{A}$ mounting a mafia fraud attack. Assume that there exist two sessions (between adversary and tag or

reader, or between both honest parties) with the same pair (r_1, r_3). This can only be a reader-adversary session and an adversary-tag session, except with probability (see [14]):

$$\binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-\ell} + \binom{q_{\mathcal{P}}}{2} \cdot 2^{-\ell}.$$

Now declare $\mathcal{A}$ to lose if a collision appears, decreasing its success probability by this negligible term, but allowing us to consider collision-free executions. In particular, except for the matching session, all values $T^0||T^1$ in the attack are independent.

Now consider a reader-adversary session sid in which $\mathcal{A}$ successfully impersonates the tag $\mathcal{P}$ to $\mathcal{V}$, such that the same nonce pair appears (by assumption) in at most one other adversary-tag session. If such a (unique) matching adversary-tag session sid^* exists, then this session (we claim) must taint sid with high probability (if sid^* does not exist we have the case below, where the adversary does not use the additional session). If even a single phase of the protocol is tainted, this invalidates session sid. Thus, suppose to the contrary, that the matching session sid^* taints no time-critical phase in sid.

Consider an untainted time-critical phase of sid where $\mathcal{V}$ sends $c_i = b$ and expects t_i^b. The adversary has thus successfully passed the first $i-1$ time-critical phases and can choose to do one of the following in the i-th phase:

The Go-Early Strategy. In session sid^* the adversary has sent some bit c_i^* to $\mathcal{P}$ before having received $\{t_i^b\}^*$. The probability that $c_i^* \neq c_i$ is $\frac{1}{2}$, in which case $\mathcal{A}$ does not know the value t_i^b and must guess it or taint the round. However, note that if the adversary sends $c_i^* \neq c_i$ in sid^*, this invalidates the lazy authentication step following the protocol, where the value s is computed based on the received challenges. Thus, this strategy invalidates the attack with probability $1/2$ per round.

The Go-Late Strategy. In session sid the adversary responds to c_i with some $\{t_i^{c_i}\}^*$ before receiving $\{t_i^{c_i}\}^*$ in session sid^*. Now $\mathcal{A}$ succeeds only with probability $\frac{1}{2}$ for this phase.

The Modify-it Strategy. The adversary schedules the message such that it receives c_i in sid, sends some $c_i^* = b$ in sid^*, receives t_i^b in sid^*, and forwards some t_i^* in sid. Hence, the scheduling corresponds to a pure relay attack, but $c_i \neq c_i^*$ or $t_i^* \neq t_i^b$. If $b = c_i^*$ is wrong then t_i^b is never sent by $\mathcal{P}$ in sid^* and the adversary can thus only guess t_i^* with probability $\frac{1}{2}$; if $b = c_i = c_i^*$ then $t_i^* \neq t_i^b$ makes the reader reject.

The Taint-it Strategy. The adversary taints this phase of sid through sid^*. This is equivalent here to losing in sid.

Thus, the most successful strategy is the Go-Early strategy, which, however, invalidates the attack with high probability. The overall success probability thus amounts to the value claimed in the theorem. □

Privacy. Since the underlying authentication protocol is wide-forward-insider private, we merely have to ensure that the challenge and response strings reveal no information about the secret key of the tag x. The challenges are chosen at random; furthermore, we use a binary truncation of the x-coordinate output for the ephemeral secret, which ensures that the response is indistinguishable from random and reveals no information about the secret.

The privacy of the protocol can be shown under an extended ODH assumption where the adversary, in addition to $A = aP, B = bP, \texttt{xcoord}(C)P$ and the oracle $\mathcal{O}(Z)$, is also given $\texttt{xcoord}(C) + a$.

Before giving the privacy proof we first introduce a conjecture that is used as building block for obtaining wide-forward-insider privacy.

Conjecture 1 *Assume a set $\mathcal{X} = \{x_1, \ldots, x_n\}$ and $\mathcal{I} = \{\iota_1, \ldots, \iota_m\}$ with $x_i, \iota_j \in_R \mathbb{Z}_\ell^*$. The game proceeds as follows:*

1. *$b \in_R \{0, 1\}$.*
2. *The adversary $\mathcal{A}$ is given $\mathcal{I}$ and can interact with the system through the following oracles:*
   (a) $\mathcal{O}_1(\alpha, \beta) := \begin{cases} (i, \tilde{r}_i + x_\alpha) & \text{if} \quad b = 0 \\ (i, \tilde{r}_i + x_\beta) & \text{if} \quad b = 1 \end{cases}$
   with $\tilde{r}_i \in_R \mathbb{Z}_\ell^*$ and let i be a counter that is incremented at every call
   (b) $\mathcal{O}_2(s, i) := s - \tilde{r}_i \in \mathcal{X} \cup \mathcal{I}$
   (c) $\mathcal{O}_3(s) := s \in \mathcal{X}$ [4]
3. *The adversary $\mathcal{A}$ is given $\mathcal{X}$ and outputs a bit g.*

The adversary wins the game if $b \stackrel{?}{=} g$.

We conjecture that the adversary has negligible probability in winning the above game.

The intuition behind the experiment described above is that the adversary has a set of insider tags for which it knows the secret keys ($\mathcal{I}$) and that there is a set of tags for which the keys remains secret ($\mathcal{X}$). Through $\mathcal{O}_1$ the adversary can obtain output of the non-corrupted tags, which is a random value added to the tag secret. Just as in the privacy definition, a random bit determines which tag secret x_i is selected. Since a fresh random value $\tilde{r}_i$ is added to every tag output, it is obvious that the adversary has negligible advantage in winning the game when only given $\mathcal{O}_1$.

The oracles $\mathcal{O}_2$ and $\mathcal{O}_3$ let the adversary verify the tag output. Both oracles only return a binary value indicating whether validation succeeded. The random $\tilde{r}_i$'s are used in $\mathcal{O}_2$ to verify the input. Intuitively, the only way that the adversary can win the game is by either guessing some x_i and checking it through oracle $\mathcal{O}_3$ or by giving an input (s, i) to $\mathcal{O}_2$ that did not directly originate from a call to $\mathcal{O}_1$ (i.e. that maps to a different x_i than the call to $\mathcal{O}_1$ did). The probability of both these events happing however seems negligible.

Theorem 6 (Privacy) DB *is narrow-strong and wide-forward-insider private according to Def. 4 under an extended ODH, the XL assumption and Conjecture 1.*

[4] Due to a technicality in the privacy proof, we need to replace this oracle by $\mathcal{O}_3(S) := dlog(S) \in \mathcal{X}$. Note that it is the challenger, which is computationally unbounded, that computes the discrete logarithm in this oracle. This definition is equivalent to the one given here, since the adversary can always call $\mathcal{O}_3$ with sP instead of s.

PROOF. Assume an adversary $\mathcal{A}$ that wins the privacy game with non-negligible advantage. Using a standard hybrid argument [35, 17], we construct an adversary that breaks the ODH-assumption. We set $Y = B$. $\mathcal{B}_i$ plays the privacy game with $\mathcal{A}$. $\mathcal{B}_i$ selects a random bit $\tilde{b}$, which will indicate which world is simulated to $\mathcal{A}$. All oracles are simulated in the regular way, with the exception of the `SendTag` and `Result` oracle for the target tag:

- `SendTag`($vtag$):
  - $j \neq i$: Generate $r_1, r_2 \in_R \mathbb{Z}_\ell^*$. Take $R_1 = r_1 P$, $R_2 = r_2 P$. Return R_1, R_2.
  - $j = i$: Generate $r_1 \in_R \mathbb{Z}_\ell^*$ Take $R_1 = r_1 P$ and $R_2 = A$. Return R_1, R_2.
- `SendTag`($vtag, e$), j'th query: retrieve the tuple $(vtag, T_0, T_1)$ from the table $\mathcal{D}$. Take the key x for tag $T_{\tilde{b}}$.
  - $j < i$: Generate $r \in_R \mathbb{Z}_\ell^*$. Take $d = \texttt{xcoord}(rP)$. Return $s = x + er_1 + d + r_2$.
  - $j = i$: Return $s = x + er_1 + (\texttt{xcoord}(C) + a)$.
  - $j > i$: Take $d = \texttt{xcoord}(r_1 Y)$. Return $s = x + er_1 + d + r_2$.
- `Result`(π): If the received R_2 in session π matches A from the ODH problem take $\dot{d}P = \texttt{xcoord}(C)P$. If not, check if R_2 matches any of the R_2's generated during the first $i-1$ `SendTag` queries. If so, use the r generated in that query and compute $\dot{d}P = \texttt{xcoord}(rP)P$. Otherwise, take $\dot{d}P = \mathcal{O}(R_1)$. Finally, compute $\dot{X} = sP - (\dot{d}P) - eR_1 - R_2$. Check $\dot{X}$ with the database, return true if $\dot{X}$ is found, false otherwise.

At the end of the game $\mathcal{A}$ outputs its guess g for the privacy game. $\mathcal{B}_i$ outputs $(\tilde{b} \stackrel{?}{=} g)$.

The above simulation to $\mathcal{A}$ is perfect, since validation is done in the same way as the protocol specification. If $R_2 = A$, the oracle $\mathcal{O}(\cdot)$ cannot be used. However, in this case we know the corresponding value of d by directly using $\texttt{xcoord}(C)P$, which gives the same result.

We use $\mathcal{A}^i$ (with $i \in [1 \ldots k]$) to denote the case that $\mathcal{A}$ runs with the first i `SendTag` queries random instances, and the other queries real instances. This is the case when $\mathcal{B}_{i+1}$ runs with a real ODH instance, or $\mathcal{B}_i$ with a random ODH instance.

By the hybrid argument we get that

$$\| \Pr\left[\mathcal{A}^0 \text{ wins}\right] - \Pr\left[\mathcal{A}^k \text{ wins}\right] \| \leq \sum \mathbf{Adv}_{B_i} .$$

Note that $\mathcal{A}^i$ wins if $\tilde{b} \stackrel{?}{=} g$.

In the case of $\mathcal{A}^0$, it is clear $\Pr\left[\mathcal{A}^0 \text{ wins}\right] = \Pr\left[\mathcal{A} \text{ wins}\right]$ since all oracles are simulated exactly as in the protocol definition.

In the case of $\mathcal{A}^k$, all `SendTag` queries are simulated with $r \in_R \mathbb{Z}_\ell^*$ and $d = \texttt{xcoord}(rP)$.

Narrow-strong privacy Since $s = x + er_1 + d + r_2$ and $R_1 = r_1 P, R_2 = r_2 P$, it follows under the XL assumption that $(x + er_1 + d + r_2, e, R_1 = r_1 P, R_2 = r_2 P)$, with d a random value from the x-coordinate distribution, is indistinguishable from $(\tilde{r}, e, R_1 = r_1 P, R_2 = r_2 P)$, with $\tilde{r}$ a uniformly random value. Hence it follows that s is indistinguishable from a uniformly random value independent of x, as long as $e, d \neq 0$. Note that this only holds in the absence of a `Result` oracle (which is able to distinguish $\tilde{r}$ from random).

So $\mathcal{A}^k$ has probability $1/2$ of winning the privacy game, since it obtains no information at all on x from a tag.

$$\begin{aligned} \| \Pr\left[\mathcal{A}^0 \text{ wins}\right] - \Pr\left[\mathcal{A}^k \text{ wins}\right] \| &= \| \Pr\left[\mathcal{A} \text{ wins}\right] - \frac{1}{2} \| \\ &= \frac{1}{2} \mathbf{Adv}_{\mathcal{A}}^{privacy} \\ &\leq \sum \mathbf{Adv}_{B_i} \end{aligned}$$

It follows that at least one of the $\mathcal{B}_i$ has non-negligible probability to win the ODH game.

Wide-forward-insider privacy For proving wide-forward-insider privacy, we also have to simulate the `Result` oracle, which was ommitted in the case of narrow-strong privacy. After applying the XL assumption to show that $(d + r_2, R_2)$ is indistinguishable from $(\tilde{r}, R_2)$, we can now do a straightforward reduction to the game from Conjecture 1. All `SendTag`($vtag, e$) calls are simulated using $\mathcal{O}_1(i, j)$ for the tags T_i and T_j passed to `DrawTag`. Calls to `Result` are simulated using $\mathcal{O}_2(sP - eR_1 - R_2, i)$ if the R_2 received by the server matches an R_2 resulting from a `SendTag`(), otherwise $\dot{d}$ is computed as in the original protocol and $\mathcal{O}_3(sP - eR_1 - R_2 - \dot{d}P)$ is used to validate the resulting secret. □

3.2 Terrorist Fraud

General distance-bounding models mention four main security threats: impersonation security, distance fraud, mafia fraud and terrorist fraud. Our protocol is resistant against the former three attacks, but not to the latter one. Indeed, a dishonest prover can simply send the values $t_0 || t_1$ to the adversary for a given session sid, thus helping it win with probability 1; however, these values cannot be reused to win a different session with independent nonces. We discuss here the notion of terrorist-fraud resistance, its applicability, and its attainability.

Terrorist-fraud attacks are in general very strong, as they consider a misbehaving, or malicious prover, willing to aid the adversary. Such attacks could be considered for instance when two entities wish to share the same identity without being caught. For instance, one entity, say Alice, might want to share her public transport privileges with another entity, called Bob, but only for a given amount of time. However, Alice does not wish to let Bob abuse her kindness; thus, she wants to make it hard for Bob to authenticate later, without her permission.

Formal models of terrorist-fraud resistance disagree about what constitutes a valid terrorist-fraud attack. Indeed, the model due to Avoine *et al.* [2] stipulates that the attack is only valid if the adversary has no further advantage from the information forwarded by the prover. In latter work, Avoine *et al.* [3] rely on the fact that adversary strategies need to information-theoretically hide the secret. However, this restriction is unnecessarily strong, since the prover could forward information about the secret which does not help the adversary in future authentication sessions. The model due to Dürholz *et al.* [14] is more lenient towards the adversary: the malicious prover can forward any information to the adversary, provided that this information does not help a simulator (given the adversary's view) authenticate with the same probability. In this model, the adversary may be willing to leak some information about the secret key, as

long as these cannot be used directly by the adversary to authenticate.

In the symmetric setting, protocols aiming to attain terrorist-fraud resistance (e.g. [3, 10, 21, 28]) relate the two responses used during time-critical phases by means of a secret key. If the prover reveals both time-critical responses (corresponding to the response bits for a 0 and 1 challenge bits) for any given round, the adversary learns a bit of the secret key that relates the two responses. Since the prover only helps the adversary offline, it cannot know the challenges that the adversary will receive at every round. As a result, it cannot help the adversary authenticate by forwarding only one of the responses. While such protocols might attain some form of terrorist fraud resistance, they are not terrorist-fraud resistant in the definition of [14], as proved in the recent result of Fischlin and Onete [15].

In the context of public-key cryptography, one could use similar strategies in order to attain the same intuitive form of terrorist-fraud resistance. In particular, the tag would compute a binary truncation of the x-coordinate of $r_1 R_3$ as t^0, and then set $t^1 = t^0 \oplus priv$ for some the private key $priv$. Note that this notion might be too weak. Indeed, Fischlin and Onete [15] show a generic attack in which the adversary forwards one of the session responses, say t^0 (this does not reveal any information about the secret). Thus, the adversary is able to respond correctly to any round in which the challenge is 0. For the other rounds, the adversary can guess the response; thus the overall winning probability is roughly $\frac{3}{4}$ per time-critical round. However, once the prover withdraws its support, the adversary (more formally a simulator having access to the adversary's view) is unable to use the information learned during the prover-aided phase of the attack. Thus, if we used the same strategy for our protocol, this attack would still apply.

It is unclear whether such an attack captures the intuition of terrorist fraud. On the one hand, the model of Avoine et al. [2] seems too restrictive: indeed, it seems unreasonable to require the prover to forward *no* information at all about the secret key. Since terrorist fraud resistance is the strongest type of attack against distance bounding protocols, it may be quite feasible for a prover to accept leakage of a few bits of the secret key as the price of a successful attack. On the other hand, the stronger notion due to Dürholz et al. [14] also seems too strong, enabling a prover to forward quite a lot of information.

In order to achieve provable terrorist-fraud resistance it seems the protocol must include a weakness, i.e. a back door for the simulator to authenticate. This is why we do not aim to address terrorist-fraud resistance here. It remains an open question which model captures the intuition behind terrorist-fraud resistance best and how to attain this property.

3.3 Allowing for Errors

The protocol as shown in Fig. 1 does not allow for transmission errors or delays in communication. However, communication is not that reliable in the typical RFID environments. In particular, transmissions are susceptible to delays and they might also be incorrect,*e.g.*, in the case of collisions.

In order to account for such weaknesses, tolerance parameters are introduced for faulty and for delayed transmissions, respectively, as outlined in Sect. 2.5. Our protocol can also be modified to be robust with respect to transmission errors.

The tag will check, upon receiving the value e from the reader, whether the Hamming distance between the first n bits of this value and the concatenation of the received challenges c_i is greater than the tolerance level $E_{\max}$. If so, the tag may choose to abort or simply forward a random value for s. The reader allows for a maximum of $E_{\max}$ erroneous time-critical responses (with respect to its computed values $u^0||u^1$). Furthermore, the reader also allows for a maximum of $T_{\max}$ number of rounds where the roundtrip times exceeds $t_{\max}$. If there are too many delayed or erroneous rounds, the reader rejects the tag.

3.4 Performance of the Protocol

General Infrastructure Our protocol assumes that the tag is able to know the public key Y of the reader to which it attempts to authenticate. In practice, this can be achieved by storing (a small number of) public keys on the tag itself. At the reader side, the public keys of the tags are either stored locally or kept at a central server that is connected to the readers.

Protocol Complexity and Parameters Our protocol requires the tag to generate randomness of bitlength $\log_2(l)$. During each protocol run, the tag must store, apart from its own secret key and the public key(s) of the reader(s), four registers of size $\log_2(l)$ to store the necessary information to complete the protocol. The tag performs four (costly) EC point multiplications and some scalar arithmetic. Note that the time-critical responses do not require arithmetic and is done by a simple if-else statement. The bottleneck in the implementation constitutes of the EC point multiplications. For a 80 bit security level, an elliptic curve over a field of about 160 bits is needed. As an indication, the ECC co-processor of Lee *et al.* [23], implementing such a curve, requires less than 15 kGEs (kilo-Gate Equivalents), consumes around $13.8\mu W$ of power, and requires 85 ms for a single point-multiplication.

4. CONCLUSION

We proposed a new distance bounding protocol and provide rigorous proofs for all achieved properties. Our protocol achieves a very high impersonation resistance independent of the number of rounds in the time-critical phase. The protocol has distance fraud resistance of about $\frac{3}{4}$ per time-critical round. The proof bypasses the flaws identified by Boureanu *et al.* [6] which affect most distance bounding protocols in the literature. Our protocol achieves mafia fraud resistance at a near optimal rate of about $\frac{1}{2}$ per time-critical round. However, it does not achieve terrorist fraud resistance since we are not willing to introduce weaknesses as argued in Sect. 3.2. Finally, the protocol achieves one of the strongest possible degrees of privacy, namely wide-forward-insider privacy.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments. We would like to thank Saartje Verheyen, without whom this paper would not have been possible.

This work was supported by the Flemish Government, IWT SBO MobCom, FWO G.0360.11N Location Privacy, and by the Research Council KU Leuven: GOA TENSE;

and by the European Commission through the FIDELITY project (contract number 284862) and the ICT programme under contract ICT-2007-216676 ECRYPT II. Jens Hermans is a research assistant, sponsored by the Fund for Scientific Research - Flanders (FWO).

5. REFERENCES

[1] M. Abdalla, M. Bellare, and P. Rogaway. The Oracle Diffie-Hellman assumptions and an analysis of DHIES. In D. Naccache, editor, *Cryptographer's Track at RSA Conference*, volume 2020 of *LNCS*, pages 143–158. Springer, 2001.

[2] G. Avoine, M. A. Bingol, S. Karda, C. Lauradoux, and B. Martin. A formal framework for analyzing RFID distance bounding protocols. In *Journal of Computer Security - Special Issue on RFID System Security, 2010*, 2010.

[3] G. Avoine, C. Lauradoux, and B. Martin. How secret-sharing can defeat terrorist fraud. In *Proceedings of the Fourth ACM Conference on Wireless Network Security WISEC 2011*, pages 145–156. ACM Press, 2011.

[4] G. Avoine and A. Tchamkerten. An efficient distance bounding RFID authentication protocol: Balancing false-acceptance rate and memory requirement. In *Conference on Information Security 2009*, volume 5735 of *LNCS*, pages 250–261. Springer, 2009.

[5] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. The One-More-RSA-Inversion problems and the security of Chaum's blind signature scheme. *Journal of Cryptology*, 16:185–215, 2003.

[6] I. Boureanu, A. Mitrokotsa, and S. Vaudenay. On the pseudorandom function assumption in (secure) distance-bounding protocols. In *Progress in Cryptology – LATINCRYPT 2012*, volume 7533 of *LNCS*, pages 100–120. Springer, 2012.

[7] S. Brands and D. Chaum. Distance-bounding protocols. In *Advances in Cryptology — EUROCRYPT'93*, volume 765 of *LNCS*, pages 344–359. Springer, 1993.

[8] D. R. Brown. Generic groups, collision resistance, and ECDSA. *Designs, Codes and Cryptography*, 35(1):119–152, 2005.

[9] D. R. L. Brown and K. Gjøsteen. A security analysis of the NIST SP 800-90 elliptic curve random number generator. In A. Menezes, editor, *Advances in Cryptology — CRYPTO*, volume 4622 of *LNCS*, pages 466–481. Springer, 2007.

[10] L. Bussard and W. Bagga. Distance-bounding proof of knowledge to avoid real-time attacks. In *Security and Privacy in the Age of Ubiquitous Computing*, volume 181 of *IFIP AICT*, pages 222–238. Springer, 2005.

[11] C. Chevalier, P.-A. Fouque, D. Pointcheval, and S. Zimmer. Optimal randomness extraction from a Diffie-Hellman element. In *Advances in Cryptology – EUROCRYPT '09*, number 5479 in LNCS, pages 572–589. Springer-Verlag, 2009.

[12] B. Danev, T. S. Heydt-Benjamin, and S. Čapkun. Physical-layer identification of RFID devices. In *USENIX*, pages 125–136. USENIX, 2009.

[13] Y. Desmedt. Major security problems with the 'unforgeable' (Feige)-Fiat-Shamir proofs of identity and how to overcome them. In *SecuriCom*, pages 15–17. SEDEP Paris, France, 1988.

[14] U. Dürholz, M. Fischlin, M. Kasper, and C. Onete. A formal approach to distance bounding RFID protocols. In *Proceedings of the* 14^{th} *Information Security Conference ISC 2011*, volume 7001 of *LNCS*, pages 47–62. Springer, 2011.

[15] M. Fischlin and C. Onete. Provably secure distance-bounding: an analysis of prominent protocols. 6^{th} Conference on Security and Privacy in Wireless and Mobile Networks ACM WISec 2013, 2013.

[16] A. Francillon, B. Danev, and S. Čapkun. Relay attacks on passive keyless entry and start systems in modern cars. Cryptology ePrint Archive, Report 2010/332, 2010. `http://eprint.iacr.org/`.

[17] O. Goldreich. *Foundations of Cryptography: Volume 1, Basic Tools*. Cambridge University Press, 2001.

[18] G. P. Hancke and M. G. Kuhn. An RFID distance bounding protocol. In *Conference on Security and Privacy for Emergency Areas in Communication Networks 2005*, pages 67–73. IEEE, 2005.

[19] J. Hermans, A. Pashalidis, F. Vercauteren, and B. Preneel. A new RFID privacy model. In V. Atluri and C. Diaz, editors, *ESORICS 2011*, volume 6879 of *LNCS*, pages 568–587. Springer, 2011.

[20] C. H. Kim and G. Avoine. RFID distance bounding protocol with mixed challenges to prevent relay attacks. In *Conference on Cryptology and Networks Security 2009*, volume 5888 of *LNCS*, pages 119–131. Springer, 2009.

[21] C. H. Kim, G. Avoine, F. Koeune, F. Standaert, and O. Pereira. The swiss-knife RFID distance bounding protocol. In *Information Security and Cryptology (ICISC) 2008*, LNCS, pages 98–115. Springer, 2008.

[22] Y. K. Lee, L. Batina, K. Sakiyama, and I. Verbauwhede. Elliptic curve based security processor for RFID. *IEEE Transactions on Computers*, 57(11):1514–1527, 2008.

[23] Y. K. Lee, L. Batina, D. Singelée, and I. Verbauwhede. Low-cost untraceable authentication protocols for RFID. pages 55–64. ACM, 2010.

[24] C. Onete. Key updates for RFID distance-bounding protocols: Achieving narrow-destructive privacy. Cryptology ePrint Archive, Report 2012/165, 2012. `http://eprint.iacr.org/`.

[25] R. Peeters and J. Hermans. Wide strong private RFID identification based on zero-knowledge. Cryptology ePrint Archive, Report 2012/389, 2012. `http://eprint.iacr.org/`.

[26] A. Ranganathan, N. O. Tippenhauer, D. Singelée, B. Skoric, and S. Capkun. Design and Implementation of a Terrorist Fraud Resilient Distance Bounding System. In S. Foresti, F. Martinelli, and M. Yung, editors, *ESORICS 2012*, volume 7459 of *LNCS*, pages 415–432. Springer-Verlag, 2012.

[27] K. B. Rasmussen and S. Čapkun. Realization of RF Distance Bounding. In *USENIX*, pages 389–402. USENIX, 2010.

[28] J. Reid, J. M. G. Nieto, T. Tang, and B. Senadji. Detecting relay attacks with timing-based protocols. In *ACM symposium on information, computer and*

communications security (ASIACCS) 2007, pages 204–213. ACM Press, 2007.

[29] SHA-3 Zoo. Overview of all Candidates for the Current SHA-3 Hash Competition Organized by NIST. `http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo`.

[30] R. Trujillo-Rasua, B. Martin, and G. Avoine. The Poulidor distance-bounding protocol. In *RFIDSec 2010*, volume 6370 of *LNCS*, pages 239–257. Springer, 2010.

[31] T. van Deursen and S. Radomirović. Insider attacks and privacy of RFID protocols. In S. Petkova-Nikova, A. Pashalidis, and G. Pernul, editors, *EUROPKI*, volume 7163 of *LNCS*, pages 65–80. Springer, 2011.

[32] S. Vaudenay. On privacy models for RFID. In *Advances in Cryptology — Asiacrypt 2007*, volume 4883 of *LNCS*, pages 68–87. Springer, 2007.

[33] E. Wenger and M. Hutter. A hardware processor supporting elliptic curve cryptography for less than 9 kGEs. In *CARDIS 2011*, volume 7079 of *LNCS*, pages 182–198. Springer, 2011.

[34] A. Yang, Y. Zhuang, and D. S. Wong. An efficient single-slow-phase mutually authenticated RFID distance-bounding protocol with tag privacy. In *Information and Communications Security*, volume 7618 of *LNCS*, pages 285–292. Springer, 2012.

[35] A. C.-C. Yao. Theory and applications of trapdoor functions (extended abstract). In *FOCS 1982*, pages 80–91. IEEE Computer Society, 1982.