Wide Strong Private RFID Identification based on Zero-Knowledge

Roel Peeters and Jens Hermans

KU Leuven - ESAT/COSIC and iMinds
Kasteelpark Arenberg 10/2446, 3001 Leuven, BELGIUM
firstname.lastname@esat.kuleuven.be

Abstract. We present the first wide-forward-insider and wide-strong RFID identification protocols that are based on zero-knowledge. Until now these notions have only been achieved by schemes based on IND-CCA2 encryption. We discuss why wide-forward-insider privacy is sufficient for most practical applications. Rigorous proofs in the standard model are provided for the security and privacy properties of our protocols. Furthermore, our protocols are the most efficient solution presented in the literature. Using only Elliptic Curve Cryptography (ECC), the required circuit area can be minimized such that our protocol even fits on small RFID tags. Concerning computation on the tag, we only require two scalar-EC point multiplications for the wide-forward-insider protocol.

Keywords. RFID, Private Identification, Zero-Knowledge, Elliptic Curve Cryptography.

1 Introduction

RFID tags are deployed in various consumer applications such as physical access tokens, car keys, contactless payment systems and electronic passports. For these applications, it is crucial that the underlying protocols protect not only security but also the (location) privacy of the end user. Yet, all communication with RFID tags can easily be eavesdropped or modified, tags respond to any query and RFID tags can be corrupted, which renders these vulnerable to attacks. On top of this, an adversary can typically learn the outcome of the identification protocol. Successful identifications result in an unlocked door, unlocked car or processed payment; while failure has no outcome.

Privacy of RFID identification protocols is evaluated in terms of achieved privacy notions. The notion of strong privacy provides the strongest privacy guarantees: no adversary actively interacting with the tags and the reader is able to infer any information on a tag's identity from tag communication, even when given all secrets stored on the tag. The notion of wide-strong privacy corresponds to strong privacy against adversaries that also learn the outcome of the protocol.

Our new notion of 'wide-forward-insider' 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.* [27], 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 tag, *e.g.*,a legitimate public transportation ticket. For most practical applications 'wide-forward-insider' privacy seems to suffice: if the adversary has a very strong corruption capability, such as in wide-strong privacy, much easier attacks exist that fall outside of the model. An adversary could, for example, modify the tag or attach an additional tracking device to the tag or object to break privacy instead.

Our primary goal is to design and evaluate an RFID identification protocol with the strongest possible privacy guarantees, i.e. wide-strong. This privacy notion cannot be achieved when considering only symmetric identification protocols [28], where some cryptographic secret is shared between tag and reader. Additionally, Damgård and Pedersen [11] showed that privacy for symmetric RFID identification protocols, comes at the cost of a non-scalable lookup procedure at the reader. Examples of symmetric RFID identification protocols can be found in [18, 13, 5]. The main reason behind using symmetric identification protocols is the perception that public-key cryptography requires either too much time, power or circuit area to implement on low-cost devices. However, Lee *et al.* [21] and Hein *et al.* [16] showed that public key cryptography, in particular Elliptic Curve Cryptography (ECC), can be realized on RFID tags. Previously, wide strong privacy has only been achieved by schemes relying on an IND-CCA2 encryption scheme (or variants of such schemes) [28, 10]. Our scheme only needs an ECC architecture without additional components typically required for IND-CCA2 encryption (e.g. hash function), resulting in a smaller hardware footprint, which is a great improvement. As a secondary goal we design a wide-forward-insider private scheme that is the most efficient of its kind and offers a sufficient privacy and security level for most practical applications.

Outline Section 2 first introduces the required definitions. An overview of relevant previously proposed private RFID protocols is given in Sect. 3. In Sect. 4 and 5, we propose our protocols and analyze their security and privacy properties. We also propose an optimized version of our protocol. Section 6 takes into account some implementation considerations and compares the different protocols.

2 Definitions

We consider a system comprised of multiple tags and only one reader, where a tag and the reader carry out an identification protocol. Each tag stores a state and the reader keeps a database of all legitimate tags, to which tags can be dynamically added by the adversary. More in general, the reader could be a central back-end server that is connected to multiple readers, however tags can only identify to one server. Adversaries are allowed to communicate with all tags and the reader. For privacy, 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.

In this section, we will first give a short overview of the selected privacy model and the different privacy notions. Then the properties of a private identification protocol are discussed. Finally we give an overview of the necessary number-theoretic assumptions.

2.1 Privacy Model

The privacy model of Vaudenay [28] was one of the first and most complete privacy models that featured the notion of strong privacy. This model is based on simulatability; for the strongest privacy notions a separate blinder between the adversary and the oracles is required. Vaudenay shows that wide strong privacy cannot be achieved in this model by using a specific feature of the blinder. Armknecht *et al.* [2] later pointed out some issues of this model with regard to the blinder. Canard *et al.* [10] also proposed a simulation based model that resolves these issues by introducing a trivial adversary. However, their model is less general, as the focus is on finding non-trivial links between messages communicated by the same tag. The Juels-Weis model [19] is a well-known privacy model based on indistinguishability. This model lacks generality since it does not allow the adversary to corrupt challenge tags. Hermans *et al.* [17] provide a general privacy model for RFID based on indistinguishability; it is more robust and easier to apply. For these reasons we selected this model, a brief description of the model is given below. For more details on the different RFID privacy models and a comparison between these, the reader is referred to [17].

Privacy Model of Hermans et al. [17] The intuition behind the RFID privacy model of Hermans *et al.* is that privacy is guaranteed if an adversary cannot distinguish with which one of two RFID tags (of its choosing), it is interacting through a set of oracles.

Privacy is defined as a distinguishability 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: $\texttt{CreateTag}(ID)$, $\texttt{Launch}()$, $\texttt{DrawTag}(T_i,T_j)$, $\texttt{Free}(vtag)_b$, $\texttt{SendTag}(vtag, m)_b$, $\texttt{SendReader}(\pi, m)$, $\texttt{Result}(\pi)$ and $\texttt{Corrupt}(T_i)$.

We extend this with a new oracle $\texttt{CreateInsider}(ID)$ to create insider tags: after creation of the tag it is immediately corrupted. The insider tag cannot be used in any `DrawTag` query. Futhermore, we 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. An detailed overview of the (extended) oracles is given in App. A.

By using the `DrawTag` oracle the adversary can arbitrarily select which tags to interact with. Based upon the challenge bit b the system that the challenger presents to the adversary will behave as either the 'left' tags T_i or the 'right' tags T_j. After $\mathcal{A}$ called the oracles, it outputs a guess bit g. The outcome of the game will be $g == b$, *i.e.*, 0 for an incorrect and 1 for a correct guess. The adversary wins the privacy game if it can distinguish correctly the 'left' from the 'right' world being executed. The advantage of the adversary $\mathbf{Adv}_{\mathcal{S},\mathcal{A}}(k)$ is defined as:

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

2.2 Privacy Notions

The following privacy notions were introduced by Vaudenay [28] 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 privacy the `CreateInsider` can be simulated using the normal `CreateTag` and `Corrupt` oracles, *i.e.*, strong-insider is equivalent to strong. The privacy notions are related as follows:

$$\begin{tikzcd}
& \text{wide-forward-insider} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \text{wide-weak-insider} \arrow[d, Rightarrow] \\
\text{wide-strong} \arrow[ur, Rightarrow] \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \text{wide-forward} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \text{wide-weak} \arrow[d, Rightarrow] \\
\text{narrow-strong} \arrow[r, Rightarrow] & \text{narrow-forward} \arrow[r, Rightarrow] & \text{narrow-weak}
\end{tikzcd}$$

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.

2.3 Private Identification Protocol

A private identification protocol has the following three properties: correctness, soundness and privacy. Correctness and soundness are necessary to establish the security of the identification protocol. Privacy will ensure that no party can infer any information on the tag's identity from the protocol messages, except the reader for which the tag is identifying to.

A function $f : \mathbb{N} \to \mathbb{R}$ is called 'polynomial' in the security parameter $k \in \mathbb{Z}$ if $f(k) = O(k^n)$, with $n \in \mathbb{N}$. It is called 'negligible' if, for every $c \in \mathbb{N}$ there exists an integer k_c such that $f(k) \leq k^{-c}$ for all $k > k_c$.

Correctness ensures that a legitimate tag is always accepted by a reader.

Definition 1. *Correctness. A scheme is correct if the identification of a legitimate tag only fails with negligible probability.*

Soundness is the property that a fake tag is not accepted by the reader. We only consider adversaries that cannot interact with the tag they try to impersonate during the identification protocol (i.e., we do not consider relay or concurrent attacks). Concurrent attacks are impossible in the RFID setting, since tags can only participate in one session at the time. To avoid relay attacks, distance bounding protocols can be deployed. Rasmussen *et al.* [24] and Ranganathan *et al.* [23] proposed the first practical implementations of such a protocols by using analogue components, which allows for the necessary small processing delay. The following definition differs from most models as we do not require matching conversations, but impersonation resistance as in [7] is sufficient.

Definition 2. *Soundness. A scheme is resistant against impersonation attacks if no polynomially bounded strong adversary succeeds, with non-negligible probability, in being identified by a verifier 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.*

In a more general setting, a tag could be allowed to identify privately to multiple readers (not connected to the same central back-end server). In such a setting one RFID tag can be used to gain access to multiple independent locations, *e.g.*, office and home. However, even for a subset of corrupted readers, the adversary should not gain an advantage in authenticating as a valid tag to an uncorrupted reader. In this setting there is a clear advantage for protocols that provide extended soundness, since the tag can use the same private/public key pair to identify to each reader.

Definition 3. *Extended Soundness. Identical to Def. 2, but the adversary is also given the secret key of the reader and the full reader database.*

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 $\mathbf{Adv}^{X}_{\mathcal{S},\mathcal{A}}(k) \leq \epsilon$, for negligible ϵ.*

2.4 Number-theoretical Assumptions

Our proposed protocol is based on Elliptic Curve Cryptography, hence we make use of additive notation. Points on the curve are represented by capital letters while scalars are represented by lowercase letters.

The $[\cdot]_x$ function is the ECDSA conversion function [8], which comes almost for free when using elliptic curves. Assuming an elliptic curve $\mathbb{E}$ with prime order ℓ over $\mathbb{F}_p$, then for a point $Q = \{q_x, q_y\}$ with $q_x, q_y \in [0 \ldots p-1]$, $[Q]_x$ maps Q to $q_x \mod \ell$. We define $[O]_x = 0$, where O is the point at infinity.

Discrete Logarithm Let P be a generator of a group $\mathbb{G}_\ell$ of order ℓ and let A be a given arbitrary 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 The one more discrete logarithm (OMDL) problem was introduced by Bellare *et al.* [3]. Let P be a generator of a group $\mathbb{G}_\ell$ of order ℓ. Let $\mathcal{O}_1()$ be an oracle that returns random elements $A_i = a_i P$ of $\mathbb{G}_\ell$. 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 $m > 0$).

x-Logarithm 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.

Diffie-Hellman Let P be a generator of a group $\mathbb{G}_\ell$ of order ℓ and let aP, bP be 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 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 Abdalla *et al.* [1] introduced the ODH assumption:

Definition 5. *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 $H(Z) := [Z]_x P$.

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

3 Previously Proposed Protocols

In this section, we give an overview of previously proposed protocols that are based on public key cryptography. Each of these protocols is correct, sound and achieves narrow-strong privacy.

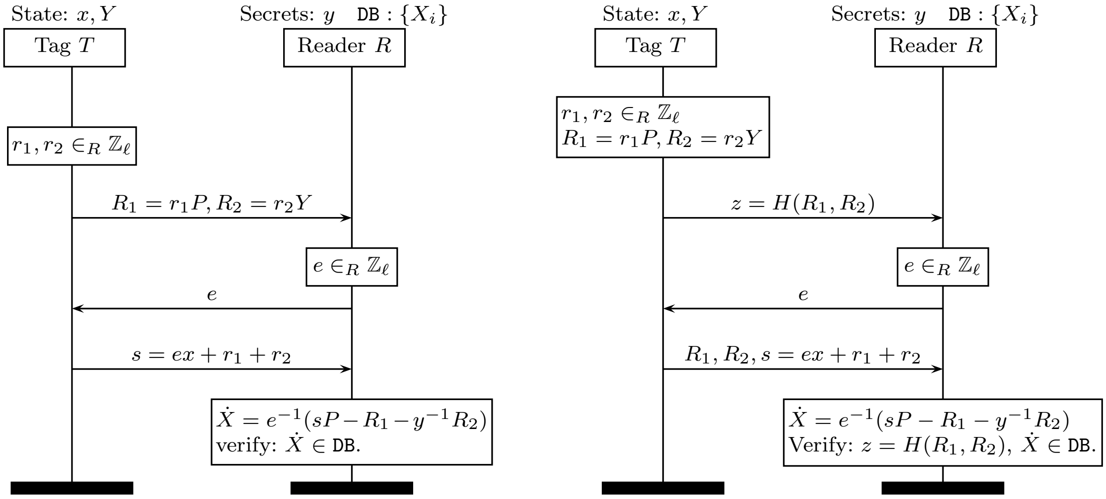
[1] The adversary can set $Z = rA$ for a known r and compute $r^{-1}(bZ) = bA$.

3.1 Zero-Knowledge Based Protocols

The zero-knowledge based protocols are proofs of knowledge for a specific verifier (reader) with public key $Y = yP$. The prover (tag) proves knowledge of the private key $x \in \mathbb{Z}_\ell$, which is the discrete logarithm of the corresponding public key $X = xP$, for P a publicly agreed-on generator of $\mathbb{G}_\ell$. The public key X of the tag will serve as its identity and has been registered with the reader.

Randomized Schnorr was proposed by Bringer *et al.* [6] (see Fig. 1(a)). It achieves extended soundness and narrow-strong privacy. This protocol requires only two scalar-EC point multiplications at the tag side.

Randomized Hashed GPS was later proposed by Bringer *et al.* [7] (see Fig. 1(b)). The protocol has extended soundness and narrow-strong privacy. The authors also claim wide-PI-forward privacy, i.e., wide-forward privacy even when the list of registered tags' identities is known. This approach requires two scalar-EC point multiplications and the evaluation of a hash function, for which additional hardware will be needed.


State: x, Y
Tag T
Secrets: y $\texttt{DB} : \{X_i\}$
Reader R
$r_1, r_2 \in_R \mathbb{Z}_\ell$
$R_1 = r_1P, R_2 = r_2Y$
$e \in_R \mathbb{Z}_\ell$
e
$s = ex + r_1 + r_2$
$\dot{X} = e^{-1}(sP - R_1 - y^{-1}R_2)$
verify: $\dot{X} \in \texttt{DB}$.
State: x, Y
Tag T
Secrets: y $\texttt{DB} : \{X_i\}$
Reader R
$r_1, r_2 \in_R \mathbb{Z}_\ell$
$R_1 = r_1P, R_2 = r_2Y$
$z = H(R_1, R_2)$
$e \in_R \mathbb{Z}_\ell$
e
$R_1, R_2, s = ex + r_1 + r_2$
$\dot{X} = e^{-1}(sP - R_1 - y^{-1}R_2)$
Verify: $z = H(R_1, R_2)$, $\dot{X} \in \texttt{DB}$.

(a) Randomized Schnorr [6]

(b) Randomized Hashed GPS [7]

Fig. 1. Zero-knowledge based protocols.

Privacy-wise, both protocols suffer from the adversary having complete freedom over the exam e it sends to the tag and the fact that the final message from the tag s contains a term that is linearly dependent on this exam and the secret of the tag x. For this reason these protocols cannot be wide-strong private.[2] Furthermore, there exist a linear relation between the commitments (R_1, R_2) and the answer s. This, together with the above, makes that Randomized Schnorr cannot be wide-weak private.[3] Both protocols are also vulnerable to insider-attacks.[4]

[2] An attacker in the middle sends $e - 1$ to the virtual tag and responds to the reader with $s + x$. For a correct guess of the tag's identity with known internal state x, the `result` oracle returns 1.

[3] For an observed protocol run π_0, an adversary can test, using the `result` oracle, that the current virtual tag is the tag of π_0. The adversary mounts a Man-In-The-Middle attack, sends to the reader $(R_1 + R_{1,0}, R_2 + R_{2,0})$, challenges the tag with $e - e_0$ and returns to the reader $s + s_0$.

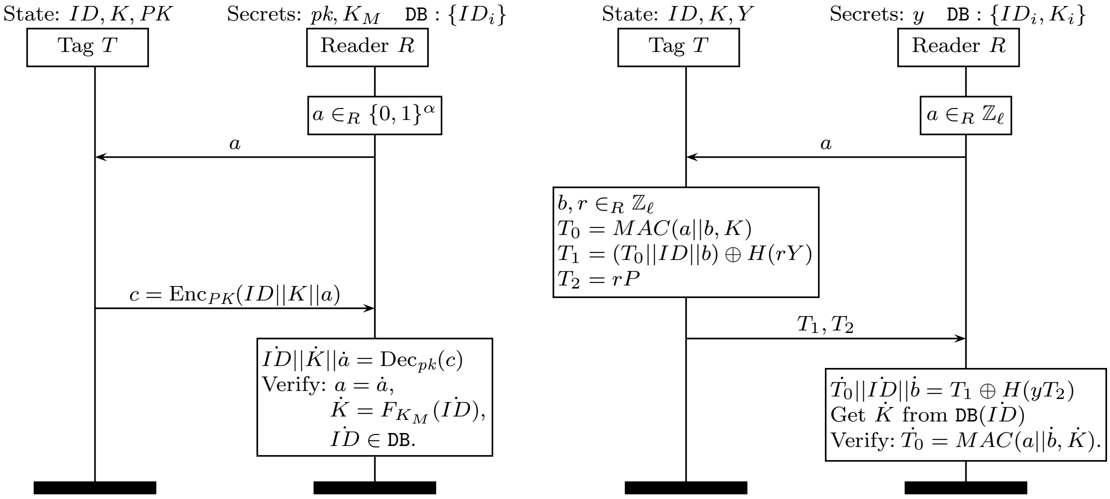
[4] Similar to the above. The attacker sends the exam e_0 to the virtual tag in protocol run π_1. When subtracting the answers $s_0 - s_1$, the tag specific part should cancel out. The attacker starts a protocol run π_2 between its insider tag (with private key x') and the reader. The attacker sets $R_1 = R_{1,0} - R_{1,1}$, $R_2 = R_{2,0} - R_{2,1}$ and replies with $s' = s_0 - s_1 + e_2x'$.

3.2 Public Key Encryption Based Protocols

For these protocols, the reader has a public/private key pair (PK, pk). The identities ID of tags that registered are stored in the reader's database. The tag and reader share a symmetric key K.

Vaudenay's Public Key Protocol [28] (see Fig. 2(a)) requires the tag to compute the public key encryption of one message. This cryptosystem needs to be secure against adaptive chosen ciphertext attacks (IND-CCA2) to have a secure identification scheme that achieves narrow-strong and wide-forward privacy. When evaluating this protocol in the privacy model of Hermans *et al.* [17], this protocol achieves wide-strong privacy. One of the most efficient IND-CCA2 cryptosystems in the standard model is DHIES [1]. This cryptosystem requires two scalar-EC point multiplications, one evaluation of a hash function, one evaluation of a MAC algorithm and the invocation of symmetric encryption scheme per encryption.

Hash ElGamal Based Protocol was proposed by Canard *et al.* [10] (see Fig. 2(b)). This protocol is secure, narrow-strong private and future untraceable. It is unclear how future untraceability (as defined by Canard *et al.* [10]) and wide-strong privacy relate to each other, however, these seem to be closely related. It makes use of a cryptosystem that is secure against chosen plaintext attacks (IND-CPA), Hash ElGamal; and a MAC algorithm. This scheme is more efficient than Vaudenay's public key scheme since the underlying encryption does not need to be IND-CCA2. Note that the combination of a MAC and IND-CPA encryption used in this specific protocol in fact provides IND-CCA2 encryption for the type of plaintext messages used [20]. The tag is required to compute two scalar-EC point multiplications, one evaluation of a hash function and one evaluation of a MAC algorithm.


State: ID, K, PK
Secrets: pk, K_M $\mathtt{DB} : \{ID_i\}$
Tag T
Reader R
$a \in_R \{0,1\}^\alpha$
a
$c = \mathrm{Enc}_{PK}(ID||K||a)$
$\dot{ID}||\dot{K}||\dot{a} = \mathrm{Dec}_{pk}(c)$
Verify: $a = \dot{a}$,
$\dot{K} = F_{K_M}(\dot{ID})$,
$\dot{ID} \in \mathtt{DB}$.
State: ID, K, Y
Secrets: y $\mathtt{DB} : \{ID_i, K_i\}$
Tag T
Reader R
$a \in_R \mathbb{Z}_\ell$
a
$b, r \in_R \mathbb{Z}_\ell$
$T_0 = MAC(a||b, K)$
$T_1 = (T_0||ID||b) \oplus H(rY)$
$T_2 = rP$
T_1, T_2
$\dot{T_0}||\dot{ID}||\dot{b} = T_1 \oplus H(yT_2)$
Get $\dot{K}$ from $\mathtt{DB}(\dot{ID})$
Verify: $\dot{T_0} = MAC(a||\dot{b}, \dot{K})$.

(a) Vaudenay's Public Key RFID Protocol [28]

(b) Hash ElGamal Based Protocol [10]

Fig. 2. Public key encryption based protocols.

Neither protocol achieves extended soundness. The tag and the reader need to store some shared (secret) data. These shared data consist of an identifier ID and a shared secret key K. Both protocols achieve wide-strong privacy and soundness can also be proven under the more strict definition of matching conversations. Wide-strong privacy rules out insider attacks on privacy.

4 A New Wide-Forward-Insider Protocol

Our proposed protocol is a modified version of the Schnorr [25] identification protocol. The original protocol is proven secure by Bellare and Palacio [4] under the OMDL assumption. Our starting point is a variant of the Schnorr identification protocol, where the exam of the reader is applied to the tag's randomness instead of its secret. This variant is equivalent to the original protocol, except for the case that $e = 0$. In the original Schnorr identification protocol this results in the adversary learning the tag's randomness while in the variant the adversary will learn the tag's secret. This situation can easily be avoided by having the tag check that e is not equal to 0.

Privacy is ensured by introducing a blinding factor d that can only be computed by the tag and the reader. The blinding factor is applied to the secret x. This blinding factor only depends on input of the tag and the public key of the reader, which is known to the tag. As such an adversary cannot influence the value of this blinding factor. This is an important difference with previously proposed zero-knowledge based protocols (see Sect. 3.1) where the adversary has influence on the factor that is applied to the secret of the tag, leading to insider attacks against privacy.

An overview of the proposed protocol is given in Fig. 3. The tag generates two random numbers r_1 and r_2, where the former is needed for extended soundness and the latter is used to ensure privacy. The tag commits to its randomness by sending R_1, R_2 to the reader. The reader verifies that neither $R_1 = O$ nor $R_2 = O$, the point at infinity. The tag's response is $s = dx + er_1$, with d the blinding factor as computed by the tag. Note that the tag must check that $d, e \neq 0$.[5] The reader verifies that a tag with public key $\dot{X} = \dot{d}^{-1}(sP - eR_1)$, with $\dot{d}$ the blinding factor as computed by the reader, has been registered. The reader keeps a list of all incomplete sessions. If a session timeout occurs or the tag fails to identify for a given challenge, the session is also considered to be completed.
The blinding factor contains $r_2 Y = yR_2$. Given the CDH assumption, this value can only be computed when given either r_2 or y. To prevent an adversary of exploiting the self-reducibility of the DL problem, this value is encapsulated in a one-way function. An obvious one-way function is a cryptographic hash function. However, to implement a cryptographic hash function on an RFID tag, additional logic is required. Current hash functions [26] require at least 50% of the circuit area of the most compact ECC implementation. For this reason we propose the following one-way function, that is build using only EC operations $H(r_2 Y) = [r_2 Y]_x P$. The value d is set to the x-coordinate of the EC point.

4.1 Analysis

The first two theorems deal with the security properties of the proposed protocol. The last theorem deals with the privacy properties of the proposed protocol.

[5] By appropriate selection of the elliptic curve (e.g. a curve without points $(0, y)$), checking $d \neq 0$ is not necessary if $R_2 \neq O$.

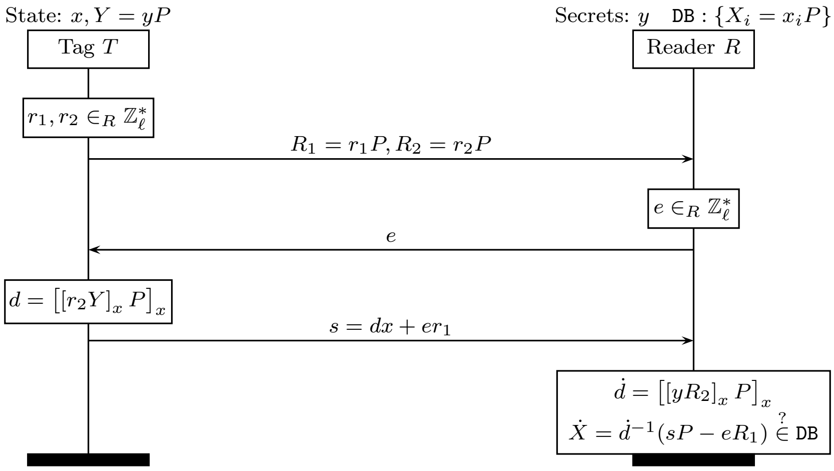

State: $x, Y = yP$
Tag T
Secrets: y $\mathtt{DB} : \{X_i = x_iP\}$
Reader R
$r_1, r_2 \in_R \mathbb{Z}_\ell^*$
$R_1 = r_1P, R_2 = r_2P$
$e \in_R \mathbb{Z}_\ell^*$
e
$d = [[r_2Y]_x P]_x$
$s = dx + er_1$
$\dot{d} = [[yR_2]_x P]_x$
$\dot{X} = \dot{d}^{-1}(sP - eR_1) \overset{?}{\in} \mathtt{DB}$

Fig. 3. Wide-forward-insider private RFID identification protocol.

Theorem 2. *The proposed protocol is correct according to Def. 1.*

Proof. Since $d = [[r_2Y]_x P]_x = [[yR_2]_x P]_x = \dot{d}$, it follows that $\dot{X} = \dot{d}^{-1}(sP - eR_1) = d^{-1}((dx + er_1)P - er_1P) = X$. □

Theorem 3. *The proposed protocol has extended soundness according to Def. 2 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.[6] We construct an adversary $\mathcal{B}$ that wins the OMDL game as follows:

- Set $X = \mathcal{O}_1()$. X 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 `SendTag`$(vtag)$ query of the i'th protocol run: return $R_{2,i} = r_{2,i}P$ with $r_{2,i} \in_R \mathbb{Z}_\ell$ and $R_{1,i} = \mathcal{O}_1()$.
  - On the second `SendTag`$(vtag, e_i)$ query of the i'th protocol run: set $d_i = \left[[r_{2,i}Y]_x P\right]_x$ and return $s_i = \mathcal{O}_2(d_iX + e_iR_{1,i})$
- During the second phase of $\mathcal{A}$, $\mathcal{B}$ proceeds as follows:
  - On the first call of $\mathcal{A}$ to `Result`(π), compute $d = [[yR_2]_x P]_x$ and store (s, d). Next, rewind $\mathcal{A}$ until right before the call to `SendReader`(π, R_1, R_2). On the next call to `SendReader`(π, R_1, R_2), return a new random e'.
  - On the next call of $\mathcal{A}$ to `Result`(π): compute $r_1 = {}^{(s-s')}/_{(e-e')}$ and $x = d^{-1}(s - er_1)$ return $(x, e_1^{-1}(s_1 - xd_1), \ldots, e_k^{-1}(s_k - xd_k))$.

[6] 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.

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 $R_2 \neq O$ is fixed. □

Before giving the privacy proof we introduce a crucial conjucture 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 $x_i, \iota_j \in_R \mathbb{Z}_\ell$. We conjecture that the adversary has negligible probability in winning the following game:

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) $\mathcal{O}_1(\alpha, \beta) := \begin{cases} (i, d_i x_\alpha) & \text{if} \quad b = 0 \\ (i, d_i x_\beta) & \text{if} \quad b = 1 \end{cases}$
   with $d_i \in_R \mathbb{Z}_\ell$ and let i be a counter that is incremented at every call
   (b) $\mathcal{O}_2(s, i) := d_i^{-1} s \in \mathcal{X} \cup \mathcal{I}$
   (c) $\mathcal{O}_3(s) := s \in \mathcal{X}$ [7]
3. The adversary $\mathcal{A}$ is given $\mathcal{X}$ and outputs a bit g.

The adversary wins the game if $b == g$.

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 multiplied with 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 d_i is multiplied with 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 d_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 4. *The proposed protocol is wide-forward-insider and narrow-strong private according to Def. 4 under the ODH assumption, the XL assumption and Conjecture 1.*

Proof. Assume an adversary $\mathcal{A}$ that wins the privacy game with non-negligible advantage. Using a standard hybrid argument [30, 15], 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:

[7] 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.

- `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 and R_2.
  - $j = i$: Generate $r_1 \in_R \mathbb{Z}_\ell$. Take $R_1 = r_1 P, R_2 = A$. Return R_1 and 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 $d = [H(rP)]_x$. Return $s = dx + er_1$.
  - $j = i$: Take $d = [H(C)]_x$. Return $s = dx + er_1$.
  - $j > i$: Take $d = [H(r_2 Y)]_x$. Return $s = dx + er_1$.
- `Result`(π): If the received R_2 in session π matches A from the ODH problem take $\dot{d} = [H(C)]_x$. 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 $\dot{d} = [H(rP)]_x$. Otherwise, take $\dot{d} = [\mathcal{O}(R_2)]_x$. Finally, compute $\dot{X} = \dot{d}^{-1}(sP - eR_1)$. 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 $(\tilde{b} == g)$.

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 $H(C)$, which gives the same result.

We use $\mathcal{A}^i$ (with $i \in [1 \dots k]$) 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. Note that $\mathcal{A}^i$ wins if $\tilde{b} == g$.

By the hybrid argument we get that:

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

- 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 $d = [[rP]_x P]_x$. Under the XL assumption it follows that d is indistinguishable from a random value from the x-coordinate distribution and that d is independent of R_1, R_2 and e.

Narrow-strong privacy Since $s = dx + er_1$ and $R_1 = r_1 P$, it follows under the XL assumption that $(dx + er_1, e, R_1 = r_1 P)$, with d a random value from the x-coordinate distribution, is indistinguishable from $(\tilde{r}, e, R_1 = r_1 P)$, 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. We can now do a straightforward reduction to the game from Conjecture 1. All $\texttt{SendTag}(vtag, e)$ 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 $\mathcal{O}_2(sP - eR_1, i)$ if the R_1 received by the server matches an R_1 resulting from a $\texttt{SendTag}()$, otherwise $\dot{d}$ is computed as in the original protocol and $\mathcal{O}_3(\dot{d}^{-1}(sP - eR_1))$ is used to validate the resulting secret. □

4.2 Efficiency Optimisation

Only one random value r is generated by the tag ($r_1 = r_2$). As such, the tag has to compute one less scalar-EC point multiplication and has to transmit one less element. The blinding factor is changed to $d = [rY]_x$. This reduces the computational effort required from the tag with another scalar-EC point multiplication. The function to compute the blinding factor is no longer one-way for rY, however, the function to compute the response s is. This means that, for the privacy proof, we need an extended variant of the oracle diffie helmann assumption. To be able to verify the tag's response, we only need $dP = [rY]_x P$, which is already given by the original oracle diffie helmann assumption. However, to be able to compute s, in the case that $R = A$, the value $[C]_x + ea$ is required. An overview of the protocol is given in Fig. 4.

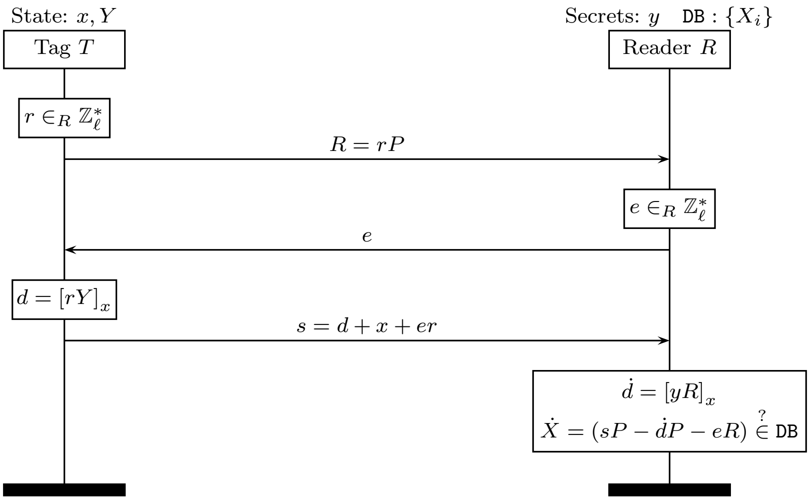

State: x, Y
Tag T
Secrets: y $\texttt{DB} : \{X_i\}$
Reader R
$r \in_R \mathbb{Z}_\ell^*$
$R = rP$
$e \in_R \mathbb{Z}_\ell^*$
e
$d = [rY]_x$
$s = d + x + er$
$\dot{d} = [yR]_x$
$\dot{X} = (sP - \dot{d}P - eR) \overset{?}{\in} \texttt{DB}$

Fig. 4. Optimised wide-forward-insider private RFID identification protocol.

Theorem 5. *The optimised protocol has extended soundness 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.[8] We construct an adversary $\mathcal{B}$ that wins the OMDL game as follows:

- Set $X = \mathcal{O}_1()$. X 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 `SendTag`$(vtag)$ query of the i'th protocol run: return $R_i = \mathcal{O}_1()$.
  - On the second `SendTag`$(vtag, e_i)$ query of the i'th protocol run: set $d_i = [yR_i]_x$ and return $s_i = \mathcal{O}_2(d_iP + X + e_iR_i)$
- During the second phase of $\mathcal{A}$, $\mathcal{B}$ proceeds as follows:
  - On the first call of $\mathcal{A}$ to `SendReader`(π, R), return a random e and store R.
  - On the first call of $\mathcal{A}$ to `Result`(π), compute $d = [yR]_x$ and store (s, d). Next, rewind $\mathcal{A}$ until right before the call to `SendReader`(π, R). On the next call to `SendReader`(π, R), return a new random e'.
  - On the next call of $\mathcal{A}$ to `Result`(π): compute $r = {}^{(s-s')}/_{(e-e')}$ and $x = s-d-er$. Return $(x, e_1^{-1}(s_1 - x - d_1), \ldots, e_k^{-1}(s_k - x - d_k))$ as the result of the OMDL game.

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. □

For privacy an extended ODH variant is required. The original ODH variant from Def. 5 gives direct access to an oracle for computing the blinding factor d. This is no longer possible, since d is defined as $[rY]_x$, which does not involve a one-way-function as required by the ODH definition.

Theorem 6. *The optimised protocol is narrow-strong and wide-forward-insider private according to Def. 4 under the extended ODH assumption and the additive variant of Conjecture 1.*

The privacy of the optimised protocol can be shown under an extended ODH assumption where the adversary, in addition to $A = aP, B = bP$, $[C]_x P$ and the oracle $\mathcal{O}(Z)$, is also given an oracle $\mathcal{O}'(z) := [C]_x + za$ that can be called once with $z \neq 0$.

A similar privacy proof as in Sect. 4.1 can be used, with different oracle calls in the `SendTag` and `Result` simulation.

Proof. Assume an adversary $\mathcal{A}$ that wins the privacy game with non-negligible advantage. Using a hybrid argument 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:

[8] 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.

- `SendTag`$(vtag)$:
  - $j \neq i$: Generate $r \in_R \mathbb{Z}_\ell$. Take $R = rP$. Return R.
  - $j = i$: Take $R = A$. Return R.
- `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 $d = [r'P]_x$. Return $s = x + er + d$.
  - $j = i$: Return $s = x + \mathcal{O}'(e)$.
  - $j > i$: Take $d = [rY]_x$. Return $s = x + er + d$.
- `Result`(π): If the received R in session π matches A from the ODH problem take $\dot{d}P = [C]_x P$. If not, check if R matches any of the R's generated during the first $i - 1$ `SendTag` queries. If so, use the r' generated in that query and compute $\dot{d}P = [r'P]_x P$. Otherwise, take $\dot{d}P = \mathcal{O}(R)$. Finally, compute $\dot{X} = sP - eR - \dot{d}P$. 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 $(\tilde{b} == g)$.

The above simulation to $\mathcal{A}$ is perfect, since validation is done in the same way as the protocol specification. If $R = A$, the oracle $\mathcal{O}(\cdot)$ cannot be used. However, in this case we know the corresponding value of dP by directly using $[C]_x P$, 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. Note that $\mathcal{A}^i$ wins if $\tilde{b} == g$.

By the hybrid argument we get that:

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

- 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 $d = [rP]_x$.

Narrow-strong privacy Since $s = x + er + d$ and $R = rP$, it follows under the XL assumption that $(x + er + d, e, R = r)$, with d a random value from the x-coordinate distribution, is indistinguishable from $(\tilde{r}, e, R = rP)$, 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 $e, d \neq 0$.

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. We can now do a straightforward reduction to the game from (the additive variant of) Conjecture 1. All $\texttt{SendTag}(vtag, e)$ 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 $\mathcal{O}_2()$ if the R_1 received by the server matches an R_1 resulting from a $\texttt{SendTag}()$, otherwise $\dot{d}$ is computed as in the original protocol and $\mathcal{O}_3()$ is used to validate the resulting secret. □

5 A New Wide-Strong Protocol

Figure 5 shows a modification of the protocol that is wide-strong private. Instead of sending an exam e, the reader sends a point R_2, which is used to compute a Diffie-Hellman key $r_1 r_2 P$. After applying a one-way function, the tag obtains e, which together with d is used to compute the tag reply $s = dx + e$. The reader performs a similar computation to obtain $\dot{d}$ and $\dot{e}$.

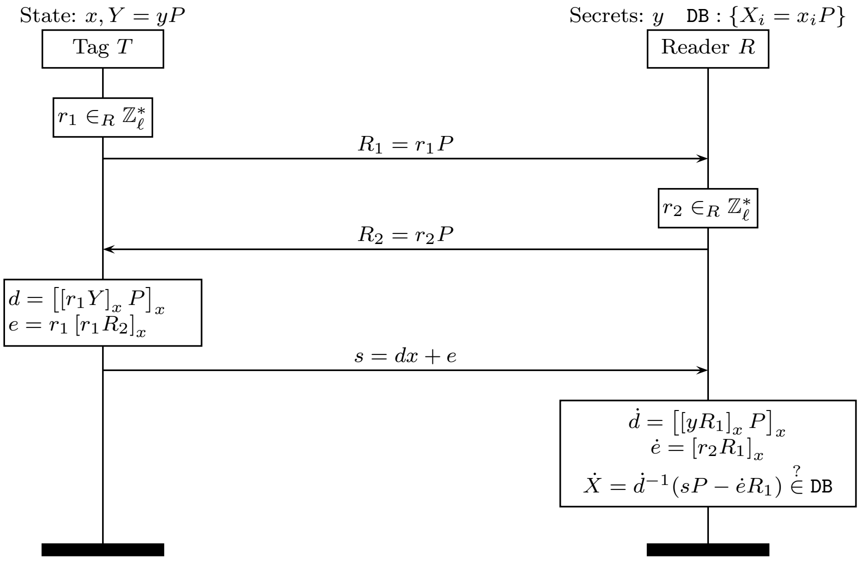

State: $x, Y = yP$
Tag T
Secrets: y $\texttt{DB} : \{X_i = x_i P\}$
Reader R
$r_1 \in_R \mathbb{Z}_\ell^*$
$R_1 = r_1 P$
$r_2 \in_R \mathbb{Z}_\ell^*$
$R_2 = r_2 P$
$d = \left[[r_1 Y]_x \, P\right]_x$
$e = r_1 \, [r_1 R_2]_x$
$s = dx + e$
$\dot{d} = \left[[y R_1]_x \, P\right]_x$
$\dot{e} = [r_2 R_1]_x$
$\dot{X} = \dot{d}^{-1}(sP - \dot{e} R_1) \overset{?}{\in} \texttt{DB}$

Fig. 5. Wide-strong private RFID identification protocol.

The proofs for correctness and soundness are omitted, since these are similar to the proofs for the wide-forward insider protocol. For privacy we introduce a variant of the ODH assumption, the dual ODH assumption, and a new conjecture.

Definition 6. *Dual Oracle Diffie-Hellman (D-ODH) Given $A = aP, B = bP, B = b'P$, a function H, H' and an adversary $\mathcal{A}$, consider the following experiments:*

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

- $\mathcal{O}_1(Z) := H(bZ)$ *for* $Z \neq \pm A$
- $\mathcal{O}_2(Z) := H'(b'Z)$ *for* $Z \neq \pm A$

- *$\mathcal{O}_3(Z) := H'(aZ)$ for $Z \neq \pm B, B'$*
- *$g = \mathcal{A}^{\mathcal{O}_i(\cdot)}(A, B, H(C), H'(C'))$*
- *Return g*

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

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

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

For our protocol we set $H(Z) := [Z]_x\, P$ and $H'(Z) := a\,[Z]_x$.

Conjecture 2. Assume a set $\mathcal{X} = \{x_1, \ldots, x_n\}$ with $x_i \in_R \mathbb{Z}_\ell$. We conjecture that the adversary has negligible probability in winning the following game:

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

The adversary wins the game if $b == g$.

Theorem 7. *The proposed protocol is wide-strong private according to Def. 4 under the Dual ODH assumption, the XL assumption and Conjecture 2.*

Proof (Sketch). The first step in the proof is similar to the proof of theorem 4. By reducing from the Dual ODH assumption to the privacy game (with a hybrid argument) we can replace the tag and reader oracles by the following simulations:

- Tag oracles: use $R_1 = A$ and $d = H(C)$. If $R_2 = B'$, then take $e = H'(C')$, otherwise take $e = \mathcal{O}_3(R_2)$.
- Reader oracles: use $R_2 = B'$ and $Y = B$. If $R_1 = A$, then take $d = H(C)$ and $e = H'(C')$. Otherwise, set $d = \mathcal{O}_1(R_1)$ and $e = \mathcal{O}_2(R_1)$.

In these simulations we can take $H(C) = \tilde{d}$ and $H'(C') = \tilde{e}$ with $\tilde{d}$ and $\tilde{e}$ uniformly random values from their respective spaces.

If the privacy adversary has no access to the `Result` oracle then the reduction above is sufficient for proving privacy: the output from the tag will be $s = \tilde{d}x_i + \tilde{e}$, which completely masks the key x_i with the values $\tilde{d}$ and $\tilde{e}$, which are random and independent of all other values.

Now consider the case that the privacy adversary does have access to the `Result` oracle. Let's assume the reader is invoked with R_1, sends out R_2 and receives s.

- R_1 was not generated by a tag: the reader will use $d = \mathcal{O}_1(R_1)$ to compute the result of the protocol, while all tags will have used a random value $\tilde{d}$. Hence the output s of the tag is independent of the d used by the reader, so the authentication will fail.

- R_1 was generated by a tag: the value of $\tilde{d}$ will be used by both tag and reader. If R_2 is modified to R_2', authentication will fail, since the reader will use $\tilde{e}$ and the tag will use $\mathcal{O}_3(R_2')$, which are independent of each other. For the remaining case that R_1 and R_2 are passed unmodified (and both tag and reader use $\tilde{d}$ and $\tilde{e}$) we reduce to the game of Conjecture 2, i.e. the tag sends out the value of s generated by $\mathcal{O}_1$ from Conj. 2 and the reader validates this value using $\mathcal{O}_2$.

□

6 Implementation Considerations

Our protocol requires the evaluation of scalar-EC point multiplications and the generation of a random number. For 80 bit security, we need an elliptic curve over a field that is approximately 160 bits in size. The protocol can be implemented on the architecture proposed by Lee *et al.* [22]. Their ECC coprocessor can be built with less than 15 kGEs (Gate Equivalent), consumes $\pm 13,8 \mu W$ of power and takes around 85 ms for one scalar-EC point multiplication. More recently, Wenger and Hutter [29] proposed an ECC coprocessor that only requires 9 kGEs, consumes $\pm 32,3 \mu W$ of power and takes around 286 ms for one scalar-EC point multiplication. Aside from the ECC coprocessor, circuit area is required for the ROM (Read Only Memory), RAM (Random Access Memory) and RNG (Random Number Generator).

6.1 Coupons

Several papers [7, 10] proposed to optimise their private RFID authentication protocols by means of precomputation. These precomputed values are stored in the form of coupons. When using coupons, the time needed by the tag to do the necessary computations drops. The most striking example is the Randomized Hashed GPS: the tag does not need to compute complex scalar-EC point multiplications and evaluate a hash function anymore, instead only some simple scalar arithmetic is performed. Of course, the use of coupons comes with a price, i.e. storage also requires circuit area. As introduced by Girault *et al.* [14], the size of coupons can be minimized by not including the randomness in the coupons, but instead implementing a pseudo-random function with a seed on the tag to generate these random numbers when the coupons are used. But even so, only a limited number of coupons can be stored on the tag.[9]

The question on how to securely get these coupons on the tag remains. These coupons can be generated by the tag itself, whenever energy is available. In this case, at the expense of having a slightly bigger design, private authentication protocols might be executed faster. Another option is that the coupons are generated by a third party and pushed on the tag. In this case, one can sometimes save on circuit area. For instance, the tag might only need to compute EC point additions or even only need scalar arithmetic. This approach has two disadvantages: first of all an attacker can quite easily mount a denial of service attack, since tags respond to any query; second, transferring these coupons securely is not straightforward. Lastly, it can be argued that strong privacy is not achievable when using coupons or a pseudo-random function instead of a true random number generator. Through the `Corrupt` oracle, the adversary learns the complete internal state of the tag, which also comprises coupons

[9] Abstracting away from the necessary control logic, one needs about one floating gate for each bit of storage. This means that we can only store 6-7 elements for a circuit area equivalent to 1kGE.

and/or the seed of the pseudo-random function. For these reasons we do not consider coupons.

6.2 Comparison

Now we will compare our protocol and its variants to previously proposed protocols, described in Sect. 3. A general overview of the protocols is given in Table 1.

Both the Randomized Schnorr and our proposed protocol benefit from a compact hardware design, only an ECC coprocessor is needed. The other protocols require additional hardware to evaluate a cryptographic hash function, which makes the design substantially larger. Recall that current hash functions [26] require at least 50% of the circuit area of the most compact ECC implementation.

The scalar-EC point multiplication is more complex than the evaluation of a hash/MAC. For a fair comparison between the performance of protocols that require the evaluation of a hash/MAC and protocols that do not, we assume the same total available circuit size. This means that our protocol can be implemented using a larger but faster ECC processor.

When also considering the more general setting, where a single tag can identify the end-user privately to multiple readers, the tags not only need to store an extra public key for every reader but also corresponding shared data, if any. In this setting there is a clear advantage for protocols that provide extended soundness, since the tag can use the same private/public key pair to identify to each reader.

Table 1. Overview different proposed protocols.

Protocol	Strongest Privacy	Insider Private	Extended Soundness	Operations
Randomized Schnorr [6]	narrow-strong	no	yes	2 EC mult
Randomized Hashed GPS [7]	narrow-strong wide-forward	no	yes	2 EC mult 1 hash
Vaudenay [28] + DHIES [1]	wide-strong	yes	no	2 EC mult 1 hash 1 MAC 1 symm enc
Hash ElGamal [10]	wide-strong	yes	no	2 EC mult 1 hash 1 MAC
Proposed Protocol (Sect. 4)	narrow-strong wide-forward-insider	yes	yes	4 EC mult
- optimised (Sect. 4.2)	narrow-strong wide-forward-insider	yes	yes	2 EC mult
Proposed Protocol (Sect. 5)	wide-strong	yes	yes	4 EC mult

7 Conclusions

This paper proposes a new wide-forward-insider and a wide-strong private RFID identification protocol. Unlike previous proposals, that are based on IND-CCA2 encryption, our protocol is based on zero-knowledge. Security and privacy of our protocols and the optimised variant of our wide-forward-insider protocol are proven in the standard model. All our proposed protocols can be implemented on RFID tags, using only Elliptic Curve Cryptography. This allows for a compact hardware design and requires minimal computational effort from the tag, namely two scalar-EC point multiplications for the wide-forward-insider protocol. As an additional benefit, our protocols do not require any shared secrets between readers and tags. Our optimised wide-forward-insider protocol is the most efficient in its kind.

Acknowledgements

The authors would like to thank everyone that contributed to some very fruitful discussions, came up with possible proof strategies, provided useful suggestions or tried to break the claimed privacy properties of the proposed protocols. Special thanks to: Julien Bringer, Ivan Damgård, Junfeng Fan, Jesper Buus Nielsen, Andreas Pashalidis, Dominique Raub, Koen Simoens, Dave Singelée, Serge Vaudenay and Frederik Vercauteren.

The work leading to these results has received funding from the European Community's Framework Programme (FP7/2007-2013) under grant agreement n^{o} 284862, and the Research Council KU Leuven: GOA TENSE (GOA/11/007). Jens Hermans is a research assistant, sponsored by the Fund for Scientic Research - Flanders (FWO).

References

1. M. Abdalla, M. Bellare, and P. Rogaway. The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES. In D. Naccache, editor, *CT-RSA*, volume 2020 of *LNCS*, pages 143–158. Springer, 2001.
2. F. Armknecht, A.-R. Sadeghi, A. Scafuro, I. Visconti, and C. Wachsmann. Impossibility Results for RFID Privacy Notions. In M. Gavrilova, C. Tan, and E. Moreno, editors, *Transactions on Computational Science XI*, volume 6480 of *LNCS*, pages 39–63. Springer, 2010.
3. 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.
4. M. Bellare and A. Palacio. GQ and Schnorr Identification Schemes: Proofs of Security against Impersonation under Active and Concurrent Attacks. In M. Yung, editor, *CRYPTO*, volume 2442 of *LNCS*, pages 162–177. Springer, 2002.
5. O. Billet, J. Etrog, and H. Gilbert. Lightweight Privacy Preserving Authentication for RFID Using a Stream Cipher. In S. Hong and T. Iwata, editors, *International Workshop — FSE*, volume 6147 of *LNCS*, pages 55–74. Springer, 2010.
6. J. Bringer, H. Chabanne, and T. Icart. Cryptanalysis of EC-RAC, a RFID Identification Protocol. In M. K. Franklin, L. C. K. Hui, and D. S. Wong, editors, *CANS*, volume 5339 of *LNCS*, pages 149–161. Springer, 2008.
7. J. Bringer, H. Chabanne, and T. Icart. Efficient Zero-Knowledge Identification Schemes which respect Privacy. In W. Li, W. Susilo, U. K. Tupakula, R. Safavi-Naini, and V. Varadharajan, editors, *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, ASIACCS, pages 195–205. ACM, 2009.
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, *CRYPTO*, volume 4622 of *LNCS*, pages 466–481. Springer, 2007.
10. S. Canard, I. Coisel, J. Etrog, and M. Girault. Privacy-Preserving RFID Systems: Model and Constructions. Cryptology ePrint Archive, Report 2010/405, 2010. `http://eprint.iacr.org/`.
11. I. Damgård and M. Ø. Pedersen. RFID Security: Tradeoffs between Security and Efficiency. In T. Malkin, editor, *CT-RSA*, volume 4964 of *LNCS*, pages 318–332. Springer, 2008.
12. B. Danev, T. S. Heydt-Benjamin, and S. Čapkun. Physical-layer Identification of RFID Devices. In *USENIX*, pages 125–136. USENIX, 2009.

13. H. Gilbert, M. J. Robshaw, and Y. Seurin. HB#: Increasing the Security and Efficiency of HB+. In N. P. Smart, editor, *EUROCRYPT*, volume 4965 of *LNCS*, pages 361–378. Springer, 2008.
14. M. Girault, G. Poupard, and J. Stern. On the Fly Authentication and Signature Schemes Based on Groups of Unknown Order. *J. Cryptology*, 19:463–487, 2006.
15. O. Goldreich. *Foundations of Cryptography: Volume 1, Basic Tools.* Cambridge University Press, 2001.
16. D. Hein, J. Wolkerstorfer, and N. Felber. *ECC Is Ready for RFID — A Proof in Silicon*, pages 401–413. Springer, Berlin, 2009.
17. J. Hermans, A. Pashalidis, F. Vercauteren, and B. Preneel. A New RFID Privacy Model. In V. Atluri and C. Diaz, editors, *ESORICS*, volume 6879 of *LNCS*, pages 568–587. Springer, 2011.
18. A. Juels and S. A. Weis. Authenticating Pervasive Devices with Human Protocols. In V. Shoup, editor, *CRYPTO*, volume 3621 of *LNCS*, pages 293–308. Springer, 2005.
19. A. Juels and S. A. Weis. Defining Strong Privacy for RFID. *ACM Trans. Inf. Syst. Secur.*, 13:7:1–7:23, November 2009.
20. H. Krawczyk. The Order of Encryption and Authentication for Protecting Communications (or: How Secure Is SSL?). In J. Kilian, editor, *CRYPTO*, volume 2139 of *LNCS*, pages 310–331. Springer, 2001.
21. 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.
22. Y. K. Lee, L. Batina, D. Singelée, and I. Verbauwhede. Low-Cost Untraceable Authentication Protocols for RFID. In C. Nita-Rotaru and F. Stajano, editors, *WISEC*, pages 55–64, Hoboken,NJ,USA, 2010. ACM.
23. A. Ranganathan, N. O. Tippenhauer, D. Singelée, B. koric, and S. Capkun. Design and Implementation of a Terrorist Fraud Resilient Distance Bounding System. In S. Foresti, F. Martinelli, and M. Yung, editors, *European Symposium on Research in Computer Security (ESORICS 2012)*, volume 7459 of *Lecture Notes in Computer Science*, pages 415–432, Pisa,Italy, 2012. Springer-Verlag.
24. K. B. Rasmussen and S. Čapkun. Realization of RF Distance Bounding. In *USENIX*, pages 389–402. USENIX, 2010.
25. C.-P. Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 4(3):161–174, 1991.
26. 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`.
27. 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.
28. S. Vaudenay. On Privacy Models for RFID. In K. Kurosawa, editor, *ASIACRYPT*, volume 4833 of *LNCS*, pages 68–87. Springer, 2007.
29. E. Wenger and M. Hutter. A Hardware Processor Supporting Elliptic Curve Cryptography for Less Than 9 kGEs. In E. Prouff, editor, *CARDIS*, volume 7079 of *LNCS*. Springer, 2011. in press.
30. A. C.-C. Yao. Theory and Applications of Trapdoor Functions (Extended Abstract). In *FOCS*, pages 80–91. IEEE Computer Society, 1982.

A Extended Oracles Model Hermans et al. [17]

- `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 entire non-volatile 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.