Privacy-Aware Authentication in the Internet of Things*

Hannes Gross[1], Marko Hölbl[2], Daniel Slamanig[1], and Raphael Spreitzer[1]

[1] Graz University of Technology, Inffeldgasse 16a, 8010 Graz, Austria
{hannes.gross,daniel.slamanig,raphael.spreitzer}@iaik.tugraz.at
[2] University of Maribor, Smetanova ulica 17, 2000 Maribor, Slovenia
marko.holbl@um.si

Abstract. Besides the opportunities offered by the all-embracing Internet of Things (IoT) technology, it also poses a tremendous threat to the privacy of the carriers of these devices. In this work, we build upon the idea of an RFID-based IoT realized by means of standardized and well-established Internet protocols. In particular, we demonstrate how the Internet Protocol Security protocol suite (IPsec) can be applied in a privacy-aware manner. Therefore, we introduce a privacy-aware mutual authentication protocol compatible with restrictions imposed by the IPsec standard and analyze its privacy and security properties. In order do so, we revisit and adapt the RFID privacy model (HPVP) of Hermans et al. (ESORICS'11). With this work, we show that privacy in the IoT can be achieved without relying on proprietary protocols and on the basis of existing Internet standards.

Keywords: Internet of Things, privacy, privacy-aware authentication, EPC Gen2, RFID, IPsec, IKEv2

1 Introduction

The Internet of Things (IoT), in particular the secure and privacy-aware integration of RFID tags into the Internet, is a demanding area of research. In contrast to other technologies that form the IoT (e.g., wireless sensor nodes, mobile phones, bluetooth devices, or ZigBee), passively-powered RFID technology represents a cheap and maintenance-free solution for interconnecting objects. Motivated by the tight integration and interconnection of objects to share information autonomously among the Internet, research on security and privacy issues has gained increasing attention. More specifically, if the corresponding information is not protected properly, it can be misused to track or profile the carrier of the RFID tags. For instance, RFID tags that respond with their unique ID to requests from any (malicious) reader easily allow anyone to track the corresponding carrier.

Even though privacy aspects of RFID protocols have gained increasing attention, e.g., the proposal of tag-authentication protocols [36, 38] as well as RFID security and privacy models [15, 16, 23, 25, 26, 28, 32, 33, 38] for the theoretical investigation of the proposed protocols, so far mostly proprietary solutions have been considered to solve specific security and privacy issues. Clearly, these proprietary solutions impede the establishment of an RFID-based Internet of Things for the following reasons. First, many of these "light-weight" protocols (cf. [19, 20, 24]) are shown to be insecure [7, 8, 12, 24] immediately after their publication.

* An extended abstract of this paper appears in the proceedings of the 14th International Conference on Cryptology and Network Security (CANS 2015).

Second, a seamless integration into the existing Internet environment is not possible, since different communication protocols between tags and readers as well as between readers and the actual Internet are employed.

Chang et al. [13] were one of the first to present the idea of connecting passive RFID tags to the Internet by using standardized Internet technologies (e.g., IPv6). However, they only considered standardized technologies for the connection of reader devices to the Internet, but still suggested the employment of proprietary protocols between the readers and the tags. In contrast, Dominikus et al. [18] proposed to employ mobile IPv6 technology to connect RFID tags directly to the Internet, i.e., without trusting the readers. As Mobile IPv6 relies on IPsec to secure the communication between entities, Gross et al. [22] investigated and also demonstrated the feasibility of integrating IPsec into the EPC Gen2 standard.

Even though the work of Gross et al. advances the establishment of a secure Internet of Things, the privacy issue still remains an unsolved problem and has not been considered in a standard-conform setting. Especially privacy-aware authentication represents an unsolved issue in the Internet of Things. Privacy-aware authentication aims for the authentication of RFID tags, but only authentic counterparts, e.g., genuine clients or backends, are able to identify specific RFID tags. More specifically, any sensitive information (e.g., unique IDs) on the tag must be protected against any other party except a genuine backend. In contrast, privacy of readers (or backends) is not considered to be sensitive as they are not carried around by specific persons. Note that privacy-aware authentication is not as strong as anonymous authentication (e.g., [4, 11]) from a privacy perspective, as anonymous authentication aims at hiding tag identities even from genuine readers, i.e., insiders. While anonymous authentication is very strong, only reasonable within specific applications and often *not* even desired, privacy-aware authentication, i.e., protecting the privacy against outsiders, should be considered as *absolutely necessary* in the IoT in order to prevent malicious readers from tracking specific RFID tags and their carriers.

Contribution. The outlined issues clearly show the immediate need for standardized privacy-aware authentication mechanisms in the IoT, as otherwise tracking of specific tags becomes trivial. Even though privacy-aware authentication protocols already exist, these existing protocols do not allow standard-conform implementations. Therefore, we first evaluate existing Internet security protocols regarding their suitability for privacy-aware authentication protocols, and find that only IPsec provides a way to integrate privacy-aware protocols. Hence, we pick up the ideas of an Internet of Things based on IPsec technology and advance the field of security and privacy by designing a privacy-aware mutual authentication mechanism which is IPsec conform. Thereby, we get rid of proprietary solutions and protocols which represent a significant drawback in the context of the Internet of Things. Furthermore, since existing privacy models either deal with proprietary protocols, consider tag-only authentication, or have been shown to be flawed, we adapt the privacy model of Hermans et al. [25] (HPVP) to formally prove the privacy and security properties of the presented mutual authentication protocol. Our privacy-aware authentication protocol is wide-strong private and is conform with the IPsec standard. A performance estimation and a comparison of our protocol against existing (proprietary) protocols complete our contribution.

Outline. In Section 2, we briefly recall some cryptographic primitives with their security notions. In Section 3, we investigate the possibilities of integrating privacy-aware authentication into existing Internet security standards. Then, in Section 4, we recall established RFID models for the evaluation of privacy and security properties in authentication proto-

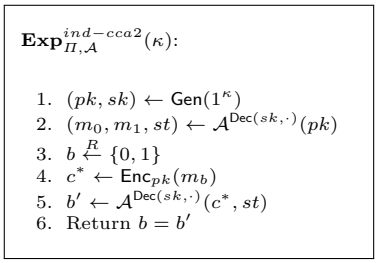

Fig. 1. IND-CCA2 experiment

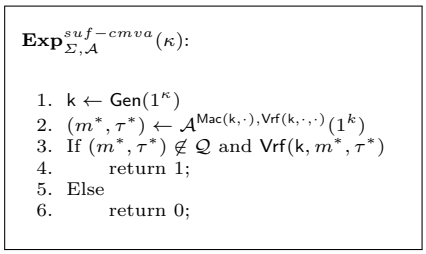

Fig. 2. sUF-CMVA experiment

cols and we adapt the model of Hermans et al. in order to prove the security and privacy properties of our protocol. In Section 5, we discuss the authentication mechanisms in IPsec and we present our instantiation of a privacy-aware authentication protocol. Subsequently, we demonstrate how this protocol can be integrated into IPsec and we also analyze the privacy and security properties of our protocol. Finally, we conclude this work in Section 6.

2 Preliminaries

In this section, we briefly introduce the required cryptographic primitives along with formal notions of security required in the analysis of the proposed privacy-aware mutual authentication protocol. In the remainder of the paper we denote a negligible function by ϵ. Thereby, a function $\epsilon : \mathbb{N} \to \mathbb{R}^+$ is called negligible if for all $c > 0$ there is a k_0 such that $\epsilon(k) < 1/k^c$ for all $k > k_0$. We call a probability p overwhelming if $p := 1 - \epsilon(k)$. Let $s \xleftarrow{R} S$ denote the sampling of an element s uniformly at random from a finite set S and $a|b$ the concatenation of two strings a and b such that a and b can be uniquely recovered. Moreover, we write $a \leftarrow A(b_1, \ldots, b_m)$ to denote that a is assigned the output of algorithm A run on input $b_1, \ldots, b_m$.

2.1 Public-Key Encryption

A public-key encryption (PKE) scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a tuple of probabilistic polynomial-time (PPT) algorithms. Gen is a probabilistic key generation algorithm that takes a security parameter κ and outputs a key pair (sk, pk). The encryption algorithm Enc is a (probabilistic) algorithm which takes a public key pk, and a message $m \in \mathcal{M}$ from some message space $\mathcal{M}$ and outputs a ciphertext $c \in \mathcal{C}$ in the ciphertext space (we will omit to explicitly mention the spaces henceforth). The deterministic decryption algorithm Dec takes a secret key sk as well as a ciphertext c and outputs a message m or a special symbol $\perp$ in case of failure.

A PKE scheme Π is called correct if for all $\kappa \in \mathbb{N}$ and all (sk, pk) generated by Gen, the probability $\mathsf{Pr}[\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(m)) = m]$ is overwhelming. Let us define the advantage of adversary $\mathcal{A}$ as $\mathbf{Adv}^{ind-cca2}_{\Pi,\mathcal{A}}(\kappa) := |2 \cdot \mathsf{Pr}[\mathbf{Exp}^{ind-cca2}_{\Pi,\mathcal{A}}(\kappa) = 1] - 1|$ for the experiment defined in Figure 1 (where $\mathcal{A}$ is not allowed to submit c^* to $\mathsf{Dec}(sk, \cdot)$ in the second phase). A PKE scheme Π is called indistinguishable under adaptively chosen ciphertext attacks (IND-CCA2 secure) if for all PPT adversaries $\mathcal{A}$ there exists a negligible function ϵ such that $\mathbf{Adv}^{ind-cca2}_{\Pi,\mathcal{A}}(k) \leq \epsilon(k)$. We note that the conventional IND-CCA2 game depicted in Figure 1 is the so-called single-query (sq) setting—as $\mathcal{A}$ can only request a single challenge.

However, this can easily be extended to the so-called multi-query (mq) setting, where $\mathcal{A}$ can obtain q challenge ciphertexts for messages of its choice. Using a straightforward hybrid argument, the following Lemma can be shown (cf. [9]).

Lemma 1. *For any $\kappa \in \mathbb{N}$, PKE scheme Π, and any multi-query adversary $\mathcal{A}$ making q queries to its challenge oracle, there exists a single-query adversary $\mathcal{B}$ such that*

$$\mathbf{Adv}^{ind-cca2-mq}_{\Pi,\mathcal{A}}(\kappa) \leq q \cdot \mathbf{Adv}^{ind-cca2-sq}_{\Pi,\mathcal{B}}(\kappa).$$

2.2 Message Authentication Codes

A message authentication code (MAC) scheme $\Sigma = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrf})$ is a tuple of PPT algorithms. Gen is a probabilistic key generation algorithm that takes a security parameter κ and outputs a key k with $|\mathsf{k}| \geq \kappa$. The tag generation algorithm Mac takes a key k, a message $m \in \mathcal{M}$ and outputs a tag τ. The verification algorithm Vrf takes a key k, a message m, and a tag τ and outputs either `true` or `false`.

A MAC scheme Σ is called correct if for all $\kappa \in \mathbb{N}$ and all k generated by Gen, the probability $\mathsf{Pr}[\mathsf{Vrf}_{\mathsf{k}}(\mathsf{m}, \mathsf{Mac}_{\mathsf{k}}(\mathsf{m})) = \mathtt{true}]$ is overwhelming. Let us define the advantage of adversary $\mathcal{A}$ as $\mathbf{Adv}^{suf-cmva}_{\Sigma,\mathcal{A}}(\kappa) := \mathsf{Pr}[\mathbf{Exp}^{suf-cmva}_{\Sigma,\mathcal{A}}(\kappa) = 1]$ for the experiment defined in Figure 2, where $\mathcal{Q}$ is the set of tuples (m, τ) of messages m queried to the Mac oracle and its corresponding answers τ. A MAC scheme Σ is called strongly existential unforgeable against chosen message and verification attacks (sUF-CMVA secure, cf. [10]), if for all PPT adversary $\mathcal{A}$ there is a negligible function ϵ such that $\mathbf{Adv}^{suf-cmva}_{\Sigma,\mathcal{A}}(\kappa) \leq \epsilon(\kappa)$.

3 Internet Security Protocols

There are essentially two predominant technologies to secure the communication over the Internet, namely the Transport Layer Security protocol (TLS) [17] and the Internet Protocol Security protocol suite (IPsec) [30]. While TLS is integrated in the Transport Layer of the OSI protocol stack and is therefore visible for applications, IPsec has the advantage of being transparent for application-layer protocols. Consequently, IPsec is closer to the physical layer, which has the advantage of less overhead in terms of additional headers added by upper-layer protocols. This is especially important, as minimizing the communication overhead is crucial for RFID tags. Note that multiple RFID tags usually share only one half-duplex communication channel and the communication speed is limited to a few kilo bits per second. Thus, IPsec seems to be preferable over TLS. However, for the sake of a complete and fair comparison, we investigate the possible implementations of privacy-aware authentication in IPsec and TLS within the following paragraphs.

Privacy-Aware Authentication in IPsec and TLS. Comparing IPsec—respectively the Internet Key Exchange protocol (IKEv2) used by IPsec—and TLS, we observe that both protocols support symmetric authentication by means of a pre-shared secret (PSK) as well as asymmetric authentication via certificates. However, the way both protocols implement the key agreement and the authentication is different.

In case of IKEv2, both communication parties establish a confidential and integrity-protected communication channel by means of a Diffie-Hellman (DH) key agreement and the authentication is then performed subsequently over the already encrypted channel. Hence, passive attackers cannot gain any information on eavesdropped authentication procedures.

Active attackers, on the other hand, can exploit the fact that the tags need to claim their identity in order to prove it. Later, in Section 5 we demonstrate how to counteract such active attacks.

In contrast, TLS does not encrypt the communication until the authentication phase is finished. Even considering a passive attacker only, the identity of the involved parties cannot be protected as the identities are claimed in plaintext[3] in the PSK setting of TLS. As the identities are also claimed in plaintext for certificate-based authentication, we conclude that privacy-aware authentication cannot be achieved by TLS alone.

Nevertheless, both protocols also support completely anonymous DH communication channels (without authentication). Based on such an anonymous DH channel, one could implement a privacy-aware authentication mechanism on the application layer[4] (in a non-standard conform way). However, we do not consider this approach as a viable option. The discussed drawbacks of TLS also hold for DTLS [37], which has been promoted for wireless sensor nodes in the IoT (e.g., [27, 31]).

4 Existing Privacy Models

In this section, we give a brief overview on existing RFID privacy models. In addition, we introduce the model of Hermans et al. [25] (HPVP) in detail, as we adapt this model in order to analyze the privacy and security properties of our protocol.

A classical RFID system is defined as follows. It consists of a central reader $\mathcal{R}$ and a set of tags $\mathcal{T} = \{T_1, T_2, \ldots, T_\ell\}$. Each of the tags T_i has an identifier (ID) that needs to be protected against the adversary $\mathcal{A}$. The task of the reader is to identify all legitimate tags in its communication range.[5] Furthermore, each tag has an internal state which contains static parts (S), like the stored secret K, but also volatile parts that may change or are associated to a specific protocol run, e.g., internal randomness.

Vaudenay [38] introduced a formal model of an RFID system (scheme) that is now widely accepted and used by different security models presented in this work. Following the notion of Vaudenay, we formally define an RFID system as follows.

Definition 1 *RFID Scheme [38]. An RFID scheme consists of the following polynomial-time algorithms:*

- `SetupReader`(1^κ)*, initializes the reader by generating the public parameters as well as the required key material depending on the security parameter* κ*. Afterwards, the public parameters and the private key are stored in the reader backend and the public parameters (including the public key) are available for all other parties of the scheme.*
- `SetupTag`(ID)*, returns the unique secret* K *and the initial state* S *for the tag with id* ID*. The pair* (ID, K) *is stored in the reader backend, and the tag is initialized with the state* S*. Depending on the used protocol, the secret* K *as well as the public parameters might be part of the state* S*.*
- `Prot`*, is a polynomial-time interactive protocol between a reader and a tag that ends with a separate tape* $Output_{\mathcal{R}}$ *for the reader and* $Output_T$ *for the tag.*

[3] See Section *7.3 Identity Privacy* of RFC 4279 [21] which explicitly states that the "PSK identity is sent in cleartext".

[4] For instance, the privacy-aware authentication mechanism for IPsec presented in this paper can be implemented at the application layer as well.

[5] We refer to this setting as a classical RFID system as the readers identify tags. In our setting (cf. Section 4.2) we slightly adapt this setting in order to cover untrusted readers.

First, we define what it means for an RFID scheme to be correct.

Definition 2 *Correctness [25, 35]. An RFID scheme is correct, if its output is correct with overwhelming probability for any polynomial-time experiment which can be described as follows:*

1. *set up the reader*
2. *create a number of tags including a subject named ID*
3. *execute a complete protocol between reader and tag ID*

The output is correct if and only if $Output_{\mathcal{R}} = \perp$ *and tag ID is not legitimate or* $Output_{\mathcal{R}} = ID$ *and tag is legitimate as well as* $Output_T = \perp$ *if the reader is not legitimate and* $Output_T = OK$ *if the reader is legitimate.*

We do not consider stronger correctness guarantees such as those provided by Deng et al. [16], where the correctness of the system still needs to hold for uncorrupted tags even after attacks, e.g., desynchronization attacks. However, we take their definition of a matching session which is adopted from [35,38] (and rules out trivial "cutting-last message" attacks when considering reader authentication) that allows to compactly formalize the security property.

Definition 3 *Matching Session [16]. Let* $(\pi, m^0_{\rightarrow\mathcal{R}}, m^0_{\rightarrow T}, \ldots, m^n_{\rightarrow\mathcal{R}}, m^n_{\rightarrow T})$ *be a transcript of a session with identifier* π *of the protocol* `Prot` *between a reader/backend* $\mathcal{R}$ *and a tag* T *run by tag* T*, where* $m^i_{\rightarrow\mathcal{R}}$ *denotes the* i*'th message sent to the reader/backend and* $m^i_{\rightarrow T}$ *denotes the* i*'th message sent to the tag* T*. We say that a session has a matching session at the side of the reader* $\mathcal{R}$*, if* $\mathcal{R}$ *ever successfully completed a session with an identical transcript.*

Let $(\pi', m'^0_{\rightarrow\mathcal{R}}, m'^0_{\rightarrow T}, \ldots, m'^n_{\rightarrow\mathcal{R}}, m'^n_{\rightarrow T})$ *be a transcript of a session run by* $\mathcal{R}$*. This session has a matching session at the side of some tag* T*, if either of the following conditions hold:*

- T *ever completed, whether successfully finished or aborted, a session of the identical transcript prefix* $(\pi', m'^0_{\rightarrow\mathcal{R}}, m'^0_{\rightarrow T}, \ldots, m'^n_{\rightarrow\mathcal{R}})$*;*
- *or,* T *is now running a session with partial transcript* $(\pi', m'^0_{\rightarrow\mathcal{R}}, m'^0_{\rightarrow T}, \ldots, m'^n_{\rightarrow\mathcal{R}})$ *and is now waiting to send the last-round message of the session* π'*.*

We note that a successfully completed session for $\mathcal{R}$ means that $Output_T \neq \perp$ and for T means that $Output_{\mathcal{R}} \neq \perp$.

4.1 History of Privacy Models

One of the first models for the analysis of privacy and security in RFID systems has been presented by Vaudenay [38]. In this model an adversary is allowed to interact with an RFID system by means of the oracles `CreateTag`, `Launch`, `DrawTag`, `Free`, `SendTag`, `SendReader`, `Result`, `Corrupt` (cf. Section 4.2 for their precise definition). Based on the adversary's restrictions in its access to specific oracles, Vaudenay defined different privacy notions and analyzed the relations among them (cf. Figure 3).

The first distinction is made between *wide* and *narrow* adversaries. Contrary to *wide* adversaries, *narrow* adversaries cannot access the `Result` oracle. This distinction models the restriction that for some protocols an adversary cannot observe whether the authentication succeeded or failed, which usually results from the fact these protocols terminate after the authentication process finishes and so an attacker does not learn the outcome of the

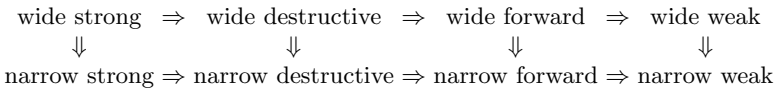

Fig. 3. Privacy notions as defined in [38].

authentication process. While for some RFID scenarios this might be a realistic restriction, the class of narrow adversaries is not relevant for IoT applications, where an adversary *can always* distinguish between a successful and an unsuccessful authentication by just looking at the communication flow, i.e., whether the communication continues after an authentication attempt or terminates. However, considering a scenario where the only purpose of the tag is to allow the reader to identify specific tags, the communication does not continue after the authentication attempt. Hence, the adversary does not learn whether or not the authentication was successful by observing the communication flow. In order to deal with such applications of our proposed privacy-aware authentication protocol, we still consider the `Result` oracle in our model.

Orthogonally to the classification of *wide* and *narrow* adversaries, Vaudenay distinguishes between *strong*, *destructive*, *forward*, and *weak* adversaries. These notions regulate the usage of the `Corrupt` oracle, which reveals the internal state of an attacked tag. Note that even if the adversary corrupted specific tags (i.e., she knows the static parts of the tag's memory, including the TID and secret key material), she should not be able to decide whether or not a particular authentication process corresponds to a specific tag. The classification is as follows:

Strong: Adversaries in the class *strong* have unrestricted access to the `Corrupt` oracle. The attacker's capabilities include all kinds of active and passive physical attacks that allow the extraction of the tag's internal state without destroying the tag, e.g., side-channel attacks.

Destructive: *Destructive* adversaries can call the `Corrupt` oracle only once for each tag, then the tag is destroyed. This covers attacks that require invasive methods like etching or the usage of a focused ion beam (FIB) in order to extract the tag's state. Hence, these tags either do not work anymore after the corruption or are conspicuously damaged.

Forward: *Forward* adversaries, after the first call to the `Corrupt` oracle can (from this point onwards) only query the `Corrupt` oracle anymore. As a result, after a corruption the adversary can no longer actively or passively follow the communication flow between tags and readers. So the attack is subdivided into two phases. In the first phase the adversary can actively communicate or eavesdrop on an ongoing authentication, and in the second phase she tries to link the state of corrupted tags to a previous authentication process in order to identify a tag.

Weak: For *weak* adversaries, no corruptions are allowed at all. In this case the attacker has either no physical access to a tag or does not have the skills and tools to perform physical attacks.

As the model of Vaudenay considered only tag authentication, Paise and Vaudenay [35] extended the model to also cover mutual authentication. They even proved that if the employed public-key encryption scheme is IND-CPA (resp. IND-CCA) secure, then the authentication mechanism is narrow-strong private (resp. forward private). However, Armknecht et al. [5, 6] observed some flaws in the aforementioned Paise-Vaudenay model

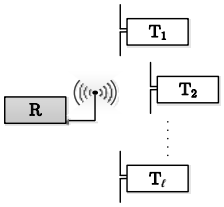

Fig. 4. Classic RFID scenario

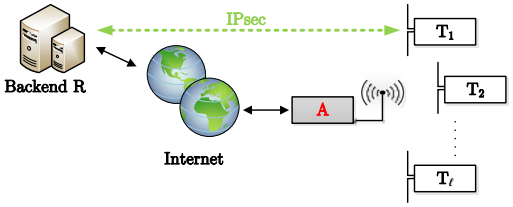

Fig. 5. Internet of Things scenario

and showed that mutual authentication and narrow-forward privacy (resp. narrow-strong privacy) cannot be achieved if tag corruption reveals both the static and the temporary/volatile memory (resp. the static memory only). Even though Vaudenay claimed in [38] that strong privacy (wide strong) is not possible—at least in their model, Ouafi and Vaudenay [34] showed that by slightly changing the initial model and relying on plaintext-aware encryption, strong privacy is indeed possible. However, their work did not consider mutual authentication.

In 2011, Hermans et al. [25] presented a new model (HPVP) in order to overcome some issues (cf. Armknecht et al. [5, 6]) of the Paise-Vaudenay model [35]. In particular, the HPVP model gets rid of the so-called blinders and instead relies on an indistinguishability-style privacy game. Hermans et al. also mention that the corruption might be restricted, i.e., to the static memory only, in case of multi-pass protocols. Since their model is not based on the blinder construction [35], the above mentioned impossibility results [5] do not apply anymore. Protocols analyzed in the HPVP model achieve wide-strong privacy in case the used public-key encryption scheme is IND-CCA2 secure. They note that their model can also be applied in case of mutual authentication, but did not go into detail in [25]. In a follow-up paper by Hermans et al. [26], the model has been extended to multiple readers which can be corrupted. Nevertheless, for our scenario we do not build upon their model as we consider a single backend in an Internet of Things scenario with multiple untrusted readers (cf. Section 4.2 for more details on this scenario). Multiple-backend scenarios are thus still possible, but do not need to be considered in our RFID privacy model. A detailed comparison of multiple RFID privacy models as well as evaluations of existing protocols according to the compared models is provided by Coisel and Martin [14].

Given the fact that some issues have been identified in the Paise-Vaudenay model and the fact that it is rather inconvenient to use, we build upon the HPVP model.

4.2 Adaption of the HPVP Model

Recall, a classic RFID system consists of a central reader $\mathcal{R}$ and a set of tags $\mathcal{T} = \{T_1, T_2, \ldots, T_\ell\}$ like shown in Figure 4. In this scenario the reader tries to identify all tags inside its range while the tags try to protect their identity against illegitimate readers.

Our IoT scenario, however, looks a bit different. As shown in Figure 5, the tags establish a secure communication channel (IPsec) to other IoT participants—like their associated backend(s)—over an untrusted communication path (the Internet). We emphasize that readers act as pure routers in this IoT setting and solely bridge the IP packets between the tags and the backend over the Internet. The communication path between the backend and a tag thus contains untrusted readers, routers, and other Internet participants which are not

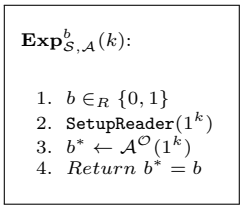

Fig. 6. Privacy experiment

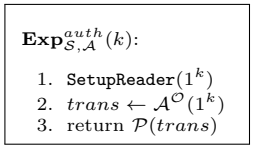

Fig. 7. Authentication experiment

directly modeled here but instead are considered as possible adversaries $\mathcal{A}$. However, for reasons of compatibility with the HPVP model, we refer to the backend in our system as $\mathcal{R}$, or reader, respectively.

For our IPsec-conform mutual authentication, we also need to slightly adapt the oracles used in the HPVP model. The main differences are as follows:

- `Launch` needs to be adapted to model authentication protocols where the tag initiates the protocol.
- `SendReader` returns an error message ERR if tag authentication failed on the reader side (before reader authentication is completed). Thereby, passive adversaries (eavesdroppers) can possibly learn the result of the tag's authentication procedure on the reader side by simply monitoring the exchanged messages.
- `Result` returns whether or not reader authentication was successful on the tag side.

The adversary interacts with the system $\mathcal{S}$ by means of a set of oracles $\mathcal{O} = \{$`CreateTag`, `Launch`, `DrawTag`, `Free`, `SendTag`, `SendReader`, `Result`, `Corrupt`$\}$ which are discussed subsequently.

For an authentication protocol to be secure, one requires (in addition to correctness as given in Definition 2) the properties privacy and security as defined below.

We denote the privacy experiment the adversary is required to win by $\mathbf{Exp}^{b}_{\mathcal{S},\mathcal{A}}(k)$ (cf. Figure 6). Here, a challenger sets up a system $\mathcal{S}$ in which the adversary can access different oracles in order to win. The experiment is based on the notion of (left-or-right) indistinguishability. Thereby, a protocol provides privacy if an attacker cannot distinguish the case where the challenger selected the "left" world ($b = 0$) from the case the right world ($b = 1$) was selected. The set of oracles is defined as follows:

`CreateTag`(`ID`) $\rightarrow T_i$: a new tag is created in the system and a reference to the created tag T_i is returned.

`DrawTag`$(T_i, T_j)_b \rightarrow vtag$: adds T_i and T_j to a virtual table $\mathcal{D}$ together with $vtag$ that refers either to T_i or T_j depending on b. If $\mathcal{D}$ already contains T_i or T_j, the oracle returns $\perp$, otherwise $vtag$.

`Free`$(vtag)_b$: retrieves and removes the corresponding entry $(vtag, T_i, T_j)$ from table $\mathcal{D}$. Depending on b, the volatile state of either T_i or T_j is reset.

`Launch`$(vtag)_b \rightarrow (\pi, m)$: launches a new protocol session according to the specifications, and returns the session identifier π as well as m.

$\texttt{SendTag}(vtag, m)_b \rightarrow m'$: forwards the message m to the tag referenced by $vtag$ (depending on b either T_i or T_j). If a reply is available m' is returned. If no reply is available[6] or in case the $vtag$ is not found, $\perp$ is returned.

$\texttt{SendReader}(\pi, m) \rightarrow m'$: forwards the message m to the reader in session π and returns its response m'. If no valid session π exists, $\perp$ is returned. In case the tag authentication failed, the oracle outputs an error message ERR.

$\texttt{Result}(\pi)$: this oracle returns whether or not the reader authentication succeeded in session π. If the authentication failed, or π is not a valid session or has not been completed (terminated) yet, the oracle returns $\perp$.[7]

$\texttt{Corrupt}(T_i)$: returns the non-volatile part of the state of the tag T_i.[8]

Finally, the adversary $\mathcal{A}$ outputs its guess b^*. Let us denote $\mathbf{Adv}^X_{\mathcal{S},\mathcal{A}}(k) := |\mathsf{Pr}[\mathbf{Exp}^0_{\mathcal{S},\mathcal{A}}(k) = 1] + \mathsf{Pr}[\mathbf{Exp}^1_{\mathcal{S},\mathcal{A}}(k) = 1] - 1|$. Privacy in the HPVP model is defined as follows:

Definition 4 *Privacy [25]. An RFID system $\mathcal{S}$, is said to unconditionally provide privacy notion X, if and only if for all adversaries $\mathcal{A}$ of type X, it holds that $\mathbf{Adv}^X_{\mathcal{S},\mathcal{A}}(k) = 0$. Similarly, we speak of computational privacy if for all polynomial-time adversaries $\mathcal{A}$, it holds that $\mathbf{Adv}^X_{\mathcal{S},\mathcal{A}}(k) \leq \epsilon(k)$.*

Definition of Oracles for the Security Notion. The HPVP model only focuses on privacy and does neither discuss nor present the security notion. Vaudenay [38] and Deng et al. [16] present explicit notions of security, but the former security notion is not very convenient to use and the latter is defined within a different model, which we adapt to our setting. We complete the presented RFID privacy model by explicitly stating the security property via an authentication experiment.

As the privacy of tags is not relevant for the adversary $\mathcal{A}$ in case of security, we can simply *ignore* the parameter b of the oracles defined above. Hence, for all the oracles available to the adversary in the authentication experiment (cf. Figure 7) we fix b, i.e., we set $b = 0$. Thereby, in case of the `DrawTag` oracle the second parameter is ignored and $vtag$ always refers to the tag T_i.

We illustrate the authentication experiment $\mathbf{Exp}^{auth}_{\mathcal{S},\mathcal{A}}(k)$ in Figure 7. Let $\mathbf{Adv}^{auth}_{\mathcal{S},\mathcal{A}}(k) := \mathsf{Pr}[\mathbf{Exp}^{auth}_{\mathcal{S},\mathcal{A}}(k) = 1]$. Here, a challenger sets up the system $\mathcal{S}$ in which the adversary interacts with the reader $\mathcal{R}$ and multiple tags $\mathcal{T}$ via oracles $\mathcal{O}$ which are defined as stated above. At the end of the experiment, $\mathcal{A}$ outputs a transcript $trans$ of a session. The goal of the adversary is to output a transcript $trans$, such that $trans$ represents a successfully completed session run by $\mathcal{R}$ (resp. an uncorrupted tag T_i) to identify an uncorrupted tag T_i (resp. the reader $\mathcal{R}$), but this session has no matching session at the side of the uncorrupted tag T_i (resp. the reader $\mathcal{R}$). We use predicate $\mathcal{P}(\cdot)$ to indicate whether this holds for a given transcript $trans$. Consequently, an adversary $\mathcal{A}$ should only be able to honestly relay messages actually

[6] At the end of an authentication protocol (regardless of whether it is successful or not) the tag might not return any message. For instance, our protocol discussed in Section 5.4 does not return any message after reader authentication.

[7] In some scenarios the `Result` oracle is not even required, as the adversary can learn whether or not the authentication was successful by observing the communication flow (cf. discussion in Section 4.1). However, for the sake of completeness, we explicitly model the `Result` oracle.

[8] As already mentioned by Armknecht et al. [5], the corruption of the volatile state would allow an adversary to trivially break the privacy of mutual-authentication protocols.

generated and sent by the reader $\mathcal{R}$ and the uncorrupted tag, which in turn means that $\mathcal{A}$ cannot break the security property. The security notion given below has been adapted from the notion of mutual authentication from Deng et al. [16].

Definition 5 *Secure Tag Authentication. We consider any adversary in the class strong. We say that an adversary wins if it outputs a transcript of a successfully completed session run by $\mathcal{R}$ to identify an uncorrupted tag T, but this session has no matching session at the side of the uncorrupted tag T.*

Definition 6 *Secure Reader Authentication. We consider any adversary in the class strong. We say that an adversary wins if it outputs the transcript of a successfully completed session run by an uncorrupted tag T to identify the reader $\mathcal{R}$, but this session has no matching session at the side of the reader $\mathcal{R}$.*

Definition 7 *Security. We say that the RFID scheme is secure if it provides secure tag authentication as well as secure reader authentication. More formally, for every $k \in \mathbb{N}$ and PPT adversary $\mathcal{A}$ we require that $\mathbf{Adv}^{auth}_{\mathcal{S},\mathcal{A}}(k) \leq \epsilon(k)$.*

5 IPsec-Conform Authentication

In this section, we give an overview of the Internet Key Exchange protocol (IKEv2) as it is used in IPsec for the negotiation of the security functionality and for the authentication of the involved parties. We also discuss how privacy can be achieved based on IKEv2, and why—despite the existence of privacy-aware mutual authentication protocols (e.g., PKC [38] and IBIHOP [36])—a new privacy-aware authentication protocol needs to be defined. Furthermore, we introduce our IPsec-conform authentication protocol, and prove its security and privacy properties.

5.1 Authentication with IKEv2

IPsec relies on the IKEv2 [29] for the negotiation of the security functionality and for the authentication of the involved parties. Thus, in order for a protocol to be IPsec-compatible it must precisely follow the message flow and data processing steps of IKEv2. In Figure 8, we illustrate the initial IKEv2 messages to generate a so-called *security association* (SA), which is then used by IPsec for securing the communication channel. An SA contains connection-specific parameters, cryptographic algorithms, and key material that is required for the secure communication in both IKEv2 and IPsec. Subsequently, we sketch the message flow in more detail.

`IKE_SA_INIT_REQ` $\leftrightarrow$ `IKE_SA_INIT_RSP:` The *initiator* starts the communication by sending the IKE_SA_INIT_REQ request. This request contains the message header (HDR), a proposal for a security association SA_{i1} consisting of one or multiple supported algorithms, the Diffie-Hellman (DH) value KE_i, and the initiator's nonce N_i. In its reply, the *responder* chooses one of the proposed security associations in SA_{r1}, and sends its DH value KE_r, and the nonce N_r. The certificate request is optional in IKEv2 and is only required if a certificate-based authentication is chosen instead of an authentication via a pre-shared key (PSK). With the information exchanged, the communicating parties derive the key material for the security association. The key material consists

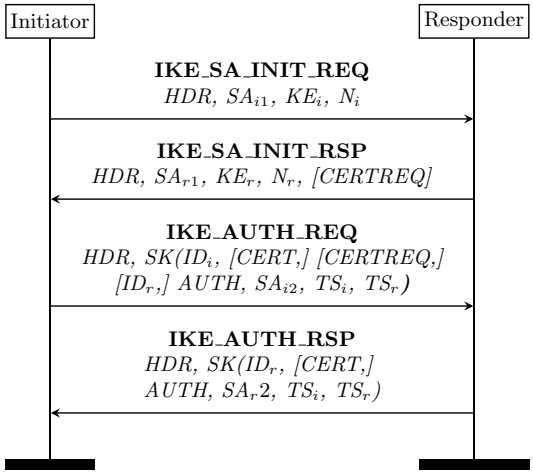

Fig. 8. Authentication process in IKEv2

of keys for encryption and decryption, keys for ensuring the integrity and authenticity of the messages, another key for the generation of pseudo-random numbers, and a key for deriving further key material as required for the IPsec message exchange. Despite the derivation key, all keys are derived pairwise since IKEv2 uses different keys for both communication directions.

`IKE_AUTH_REQ` ↔ `IKE_AUTH_RSP`**:** From this point on, the communication is already confidential and integrity protected (denoted by $SK(...)$), but the authenticity of the involved parties still needs to be verified in the next step. Therefore, the initiator sends the IKE_AUTH_REQ request containing the identity claim ID_i, the $AUTH$ value (which is computed over the IKE_SA_INIT_REQ messages sent by the initiator), the responder's nonce and the MACed ID of the initiator. Depending on the chosen authentication method, the initiator either computes a MAC on the basis of the PSK or a signature for the $AUTH$ value. Furthermore, an SA proposal (SA_{i2}) for the IPsec communication, and the so-called traffic selectors (TS_i and TS_r) are transmitted. The traffic selectors bind the negotiated SA to a certain IP address and port range, which, however, is of less interest for an RFID-based IoT scenario. Upon receiving the message, the responder decrypts the ciphertext and extracts the identity of the tag. Afterwards, the responder verifies the authenticity of the initiator according to the $AUTH$ value, either by means of the PSK or the public key of the initiator.

The IKE_AUTH_RSP response is assembled similarly to the IKE_AUTH_REQ request of the initiator. After verifying the authenticity of the responder, both nodes derive the key material for the subsequent communication over IPsec by using the negotiated SA and the derivation key.

5.2 IPsec Conformance of Existing Protocols

As outlined in Section 5.1, the IKEv2 protocol already defines the message flow and data processing steps. Thus, a privacy-aware authentication protocol needs to conform to this

message flow and data processing steps in order to fit into the IPsec standard. When looking at existing literature [14, 26] we observe that there are only very few mutual-authentication protocols that have been formally analyzed. To the best of our knowledge, only the PKC [38] and IBIHOP [36] meet these requirements (mutual authentication and a formal analysis), but these protocols do not conform to the message flow as required by IKEv2. Hence, a privacy-aware authentication protocol that fits into IKEv2 still needs to be defined.

5.3 Possible Realizations

In general, privacy-aware authentication in the IKEv2 protocol can be realized for either a *tag-initiated* or a *backend-initiated* authentication scenario[9]. However, when the tag initiates the IKEv2 protocol, the tag needs to reveal its identity to an unauthenticated communication partner through the third protocol message, which violates the privacy of the tag.

On the other hand, if the backend initiates the communication, a unique PSK per tag leads to the situation that the backend does not know for which of the tags to compute the $AUTH$ value. A straightforward realization would thus be to use a PSK that is shared among all the tags, but this has the major drawback that one broken tag results in a broken system. As an alternative, a backend-first scenario could be realized by using certificates. Again, in a scenario where the tag starts the communication this would inevitably reveal the tag's identity. Nevertheless, if the backend initiates the protocol, the identity of the backend can be verified by first ensuring the validity of the certificate, and then checking the $AUTH$ value with the public key of the backend. Only if both are valid, the tag continues the protocol execution by sending its response containing the identity claim, and the $AUTH$ value. The tag authentication can then either be certificate-based or PSK-based. Because the tag reveals its identity only after the backend has been successfully authenticated and the communication channel in the second protocol phase is already protected, neither active nor passive attackers can identify specific tags.

However, the validation of certificates on the tag side, i.e., verifying the certificate chain up to a trusted root certificate, represents an enormous effort for a constrained RFID tag. Even if the tag uses PSK-based authentication and only a single backend certificate is used—and therefore no certificate chain needs to be verified—, at least the certificate as well as a signature must be verified on the tag side. A PSK-based authentication mechanism for both sides is thus the desired choice in terms of computational overhead.

Furthermore, in a typical IoT scenario, an RFID tag is usually the initiator of the conversation as the tag updates the backend with the information gathered from its environment. Thus, the resulting requirements for an IPsec-conform privacy-aware authentication are: (1) a tag-initiated protocol, and (2) the corruption of one tag should not lead to a broken system regarding the remaining tags. In the next section, we propose a new privacy-aware authentication protocol, which is wide-strong private under our model and furthermore fits into the IKEv2 protocol, i.e., allows tag-initiated conversations, and uses PSK-based authentication on both sides.

5.4 IPsec-Conform Privacy-Aware Authentication

Figure 9 shows our IPsec-conform privacy-aware mutual authentication protocol, which relies on the Diffie-Hellman Integrated Encryption (DHIES) scheme [3]. Subsequently, we use the

[9] Note that tag-initiated authentication does not necessarily mean tag-first authentication, i.e., that the tag is indeed authenticated first, but only that the tag initiates the protocol.

additive notation as in the elliptic curve setting for the description of our protocol. We denote by $\mathbb{G}$ the description of an additive group of prime order q with some fixed generator G.

DHIES Excursus. DHIES [3] is a public-key encryption scheme $\Pi_{DHIES} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. Here, the Gen algorithm generates a private key $k \xleftarrow{R} \mathbb{Z}_q$ and a public key $K \leftarrow k \cdot G$. The Enc algorithm takes the public key K and a message m. It computes an ephemeral public key $R \leftarrow r \cdot G$ for $r \xleftarrow{R} \mathbb{Z}_q$ and the secret DH value $P \leftarrow r \cdot K$. P is then used to derive two symmetric keys via a key-derivation function as $(k_{mac}, k_{enc}) \leftarrow \mathsf{KDF}(P)$. These keys are used to obtain $c \leftarrow \mathsf{SymEnc}_{k_{enc}}(m)$ using a symmetric encryption algorithm and to generate a tag as $t \leftarrow \mathsf{Mac}_{k_{mac}}(m)$ to authenticate the message (MAC-and-encrypt). Finally, Enc outputs the tuple (R, c, t). The Dec algorithm takes a secret key k and ciphertext (R, c, t). It computes the DH value $P = k \cdot R$. Then, the KDF is computed over P and the two keys k_{mac} and k_{enc} are used to decrypt c and to verify the tag t. If t is valid, it returns $m \leftarrow \mathsf{SymDec}_{k_{enc}}(c)$ and $\perp$ otherwise. We note that DHIES has been shown to provide IND-CCA2 security in [3].

Our protocol. Figure 9 omits the IKEv2 parameters that are not relevant for the properties of our protocol for brevity reasons. The $\mathsf{SetupReader}$ algorithm generates a secret and public key pair (k_B, $K_B = k_B \cdot G$) representing the backend's secret integer (scalar) and the corresponding static Diffie-Hellman (DH) parameter. Furthermore, the $\mathsf{SetupTag}$ algorithm generates a unique pre-shared secret key k_{PSK} for each tag, which is shared between the tag and the backend.

The protocol follows the notion of a tag-initiated challenge-response protocol to be compatible with the IPsec's IKEv2 protocol. Each tag contains the backend's public key K_B (the static DH parameter), which ensures that only the genuine backend (in possession of k_B) can decrypt the received data. The tag starts the protocol by generating a nonce N_T (of suitable bitsize λ) and the ephemeral DH parameter $R \leftarrow r_T \cdot G$ for $r_T \xleftarrow{R} \mathbb{Z}_q$. Based on the resulting shared secret $P = r_T \cdot k_B \cdot G$ and the nonces, the tag and the backend derive symmetric encryption and authentication keys for both sides by means of a KDF. In contrast to a single DHIES instance—and to fit into the IKEv2 protocol—, two separate key derivation functions ($\mathsf{KDF_T}$ and $\mathsf{KDF_B}$) are used to derive the keys (k_{mac}, k_{enc}) for the tag and the backend, respectively. Instead of referring directly to the encryption or authentication keys, we use k_{ae*} for $*$ being T or B to denote these tuples. Furthermore, we denote by $\widehat{\mathsf{Enc}}_{k_{ae*}}$ that the symmetric encryption SymEnc and the Mac function of the DHIES is implicitly called such that $\widehat{\mathsf{Enc}}_{k_{ae*}}$ returns the tuple (c, t) under the key set k_{ae*}. The decryption $\widehat{\mathsf{Dec}}_{k_{ae*}}$ works analogously.

After the backend decrypted the tag's ID (TID), the corresponding PSK (k_{PSK}) is used to verify the tag authentication. Therefore, we use an additional PSK (k_{PSK}) for the authentication of both sides—by generating a and a', respectively.

Implicit Backend-First Authentication. Analyzing our protocol, we observe that our protocol is a tag-first authentication protocol with an "implicit" backend-first authentication. Implicit backend-first authentication is achieved as only the genuine backend is able to decrypt the message containing the tag's ID that is required for the subsequent tag authentication.

In Appendix A, we prove that our protocol provides strong privacy under wide adversaries.

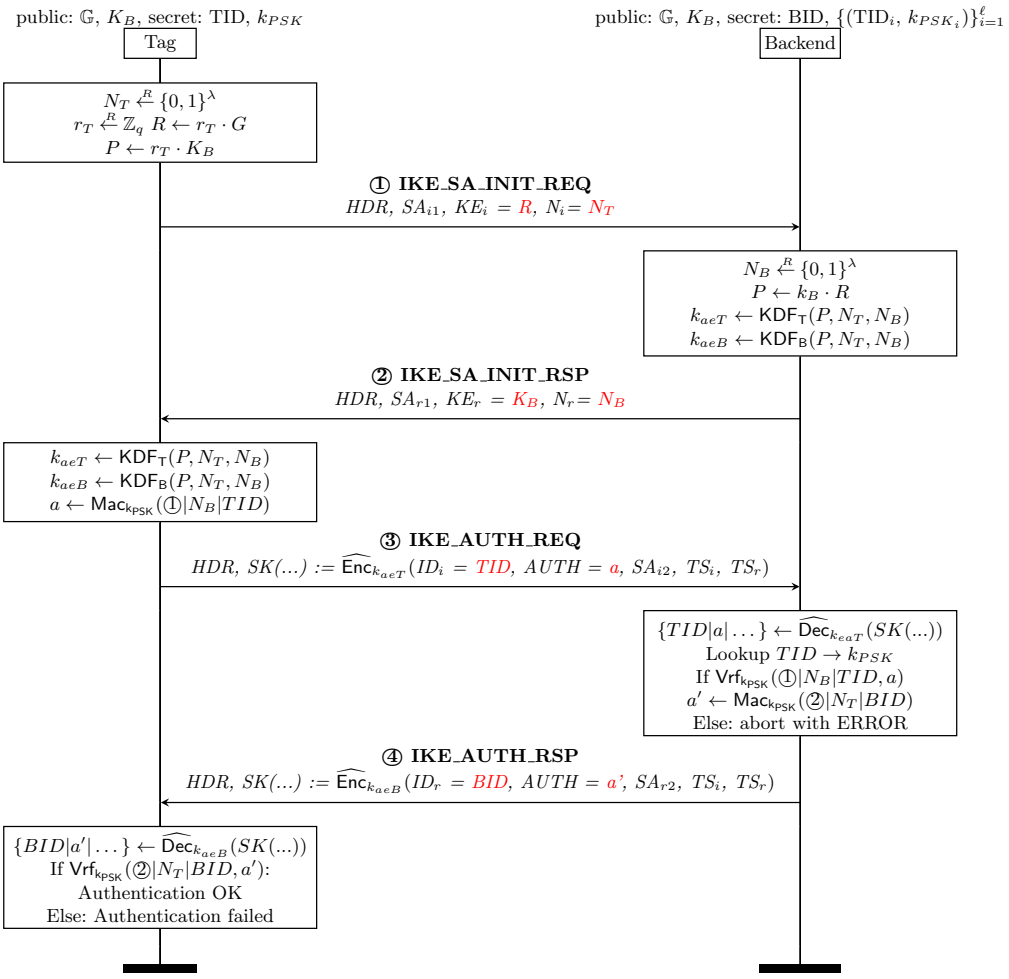

Fig. 9. IPsec-conform privacy-aware mutual authentication protocol

Theorem 1. *If DHIES used in the protocol of Figure 9 is IND-CCA2 secure and the DDH assumption holds, then the protocol is strong private against wide adversaries.*

Moreover, we prove the security of the protocol in Figure 9 against strong adversaries in Appendix B.

Theorem 2. *If the MAC* $\Sigma = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrf})$ *used in the protocol of Figure 9 is sUF-CMVA secure, then the protocol is secure against strong adversaries.*

Performance Evaluation and Comparison. As elliptic curve cryptography (ECC) is the most reasonable setting for public-key cryptography in resource-constrained environments, we assume an instantiation of DHIES in this setting. In [22], the standard-conform

integration of IPsec into the EPC Gen2 standard is presented and the requirements regarding the cryptographic primitives and the implementation overhead are evaluated. Therefore, [22] uses the NIST P-192 elliptic curve and the AES algorithm. Both ECC and AES cores were designed for low-power applications. As these results show, one ECC scalar multiplication consumes around 700 k cycles, which is significantly more than one AES operation with only 1 k cycles. Thus, the computational complexity of a protocol mainly depends on the number of required scalar multiplications if the number of symmetric-key operations is reasonably low.

Furthermore, PKC [38], IBIHOP [36], and our protocol are the only privacy-aware authentication protocols that provide mutual authentication. However, in contrast to our protocol IBIHOP is less efficient and like the PKC it is not standard conform. More specifically, we only require two scalar multiplications on the tag's side compared to the three multiplications required by the IBIHOP protocol. Therefore, our protocol is the only one that provides standard-conform mutual authentication.

6 Conclusion

Building on the recent work in [22], that focused on the integration of IPsec into RFID tags, we proposed an IPsec-conform privacy-aware mutual authentication mechanism between RFID tags and clients on the Internet. Thereby, we further paved the way for an IoT that is based on well-established standards. Our privacy-aware authentication does not reveal sensitive information like IDs unless the tag is assured to communicate with a genuine backend. Consequently, we reduce privacy concerns of carriers of RFID tags since undesired disclosure of sensitive information is prevented.

Acknowledgements. We would like to thank the anonymous reviewers of CANS 2015 for their valuable comments. This work has been supported by the Austrian Science Fund (FWF) under the grant number TRP251-N23 (Realizing a Secure Internet of Things - ReSIT), the Austrian Research Promotion Agency (FFG) and the Styrian Business Promotion Agency (SFG) under grant number 836628 (SeCoS) and by EU Horizon 2020 through project Prismacloud (GA No. 644962).

References

1. IEEE Standard Specifications for Public-Key Cryptography - Amendment 1: Additional Techniques. *IEEE Std 1363a-2004 (Amendment to IEEE Std 1363-2000)*, pages 1–167, Sept 2004.
2. SEC 1: Elliptic Curve Cryptography. *Standards for Efficient Cryptography (SEC)*, pages 1–144, May 2009.
3. M. Abdalla, M. Bellare, and P. Rogaway. The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES. In *CT-RSA*, volume 2020 of *LNCS*, pages 143–158. Springer, 2001.
4. F. Armknecht, L. Chen, A. Sadeghi, and C. Wachsmann. Anonymous Authentication for RFID Systems. In *RFIDSec*, volume 6370 of *LNCS*, pages 158–175. Springer, 2010.
5. F. Armknecht, A. Sadeghi, A. Scafuro, I. Visconti, and C. Wachsmann. Impossibility Results for RFID Privacy Notions. *Transactions on Computational Science*, 11:39–63, 2010.
6. F. Armknecht, A. Sadeghi, I. Visconti, and C. Wachsmann. On RFID Privacy with Mutual Authentication and Tag Corruption. In *ACNS*, volume 6123 of *LNCS*, pages 493–510, 2010.
7. G. Avoine and X. Carpent. Yet Another Ultralightweight Authentication Protocol That Is Broken. In *RFIDSec*, volume 7739 of *LNCS*, pages 20–30. Springer, 2012.

8. V. Banciu, S. Hoerder, and D. Page. Light-weight Primitive, Feather-weight Security: A Cryptanalytic Knock-out. In *WESS*, pages 3:1–3:10, New York, NY, USA, 2013. ACM.
9. M. Bellare, A. Boldyreva, and S. Micali. Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements. In *EUROCRYPT*, volume 1807 of *LNCS*, pages 259–274. Springer, 2000.
10. M. Bellare, O. Goldreich, and A. Mityagin. The Power of Verification Queries in Message Authentication and Authenticated Encryption. Cryptology ePrint Archive, Report 2004/309, 2004. `http://eprint.iacr.org/`.
11. M. Burmester, B. de Medeiros, and R. Motta. Anonymous RFID Authentication Supporting Constant-Cost Key-Lookup Against Active Adversaries. *IJACT*, 1(2):79–90, 2008.
12. J. C. H. Castro, P. Peris-Lopez, M. Safkhani, N. Bagheri, and M. Naderi. Another Fallen Hash-Based RFID Authentication Protocol. In *WISTP*, volume 7322 of *LNCS*, pages 29–37. Springer, 2012.
13. Y.-C. Chang, J.-L. Chen, Y.-S. Lin, and S. M. Wang. RFIPv6 - A Novel IPv6-EPC Bridge Mechanism. In *ICCE*, pages 1–2, 2008.
14. I. Coisel and T. Martin. Untangling RFID Privacy Models. *Journal Comp. Netw. and Communic.*, 2013:710275:1–710275:26, 2013.
15. I. Damgård and M. Ø. Pedersen. RFID Security: Tradeoffs between Security and Efficiency. In *CT-RSA*, volume 4964 of *LNCS*, pages 318–332. Springer, 2008.
16. R. H. Deng, Y. Li, M. Yung, and Y. Zhao. A New Framework for RFID Privacy. In *ESORICS*, volume 6345 of *LNCS*, pages 1–18. Springer, 2010.
17. T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246, 2008.
18. S. Dominikus, M. J. Aigner, and S. Kraxberger. Passive RFID Technology for the Internet of Things. In *ICITST*, pages 1–8. IEEE, 2010.
19. P. Dusart and S. Traoré. Lightweight Authentication Protocol for Low-Cost RFID Tags. In *WISTP*, volume 7886 of *LNCS*, pages 129–144. Springer, 2013.
20. A. Eghdamian and A. Samsudin. A Secure Protocol for Ultralightweight Radio Frequency Identification (RFID) Tags. In *ICIEIS*, volume 251 of *CCIS*, pages 200–213. Springer, 2011.
21. P. Eronen and H. Tschofenig. Pre-Shared Key Ciphersuites for Transport Layer Security (TLS). RFC 4279, 2005.
22. H. Gross, E. Wenger, H. Martín, and M. Hutter. PIONEER – a Prototype for the Internet of Things Based on an Extendable EPC Gen2 RFID Tag. In *RFIDSec*, volume 8651 of *LNCS*, pages 54–73. Springer, 2014.
23. J. Ha, S. Moon, J. Zhou, and J. Ha. A New Formal Proof Model for RFID Location Privacy. In *ESORICS*, volume 5283 of *LNCS*, pages 267–281. Springer, 2008.
24. M. H. Habibi, M. R. Alaghband, and M. R. Aref. Attacks on a Lightweight Mutual Authentication Protocol under EPC C-1 G-2 Standard. In *WISTP*, volume 6633 of *LNCS*, pages 254–263. Springer, 2011.
25. J. Hermans, A. Pashalidis, F. Vercauteren, and B. Preneel. A New RFID Privacy Model. In *ESORICS*, volume 6879 of *LNCS*, pages 568–587. Springer, 2011.
26. J. Hermans, R. Peeters, and B. Preneel. Proper RFID Privacy: Model and Protocols. *IEEE Trans. Mob. Comput.*, 13(12):2888–2902, 2014.
27. R. Hummen, H. Shafagh, S. Raza, T. Voigt, and K. Wehrle. Delegation-based Authentication and Authorization for the IP-based Internet of Things. In *SECON*, pages 284–292. IEEE, 2014.
28. A. Juels and S. A. Weis. Defining Strong Privacy for RFID. *ACM Trans. Inf. Syst. Secur.*, 13(1), 2009.
29. C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, and T. Kivinen. Internet Key Exchange Protocol Version 2 (IKEv2). RFC 7296, 2014.
30. S. Kent and K. Seo. Security Architecture for the Internet Protocol. RFC 4301, 2005.
31. T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, and G. Carle. DTLS based Security and Two-way Authentication for the Internet of Things. *Ad Hoc Networks*, 11(8):2710–2723, 2013.

32. C. Ma, Y. Li, R. H. Deng, and T. Li. RFID Privacy: Relation Between Two Notions, Minimal Condition, and Efficient Construction. In *CCS*, pages 54–65. ACM, 2009.
33. C. Y. Ng, W. Susilo, Y. Mu, and R. Safavi-Naini. RFID Privacy Models Revisited. In *ESORICS*, volume 5283 of *LNCS*, pages 251–266. Springer, 2008.
34. K. Ouafi and S. Vaudenay. Strong Privacy for RFID Systems from Plaintext-Aware Encryption. In *CANS*, volume 7712 of *LNCS*, pages 247–262. Springer, 2012.
35. R. Paise and S. Vaudenay. Mutual Authentication in RFID: Security and Privacy. In *ASIACCS*, pages 292–299. ACM, 2008.
36. R. Peeters, J. Hermans, and J. Fan. IBIHOP: Proper Privacy Preserving Mutual RFID Authentication. In *RFIDSec Asia*, volume 11 of *CIS*, pages 45–56. IOS Press, 2013.
37. E. Rescorla and N. Modadugu. Datagram Transport Layer Security Version 1.2. RFC 6347 (Proposed Standard), Jan. 2012.
38. S. Vaudenay. On Privacy Models for RFID. In *ASIACRYPT*, volume 4833 of *LNCS*, pages 68–87. Springer, 2007.

A Proof of Theorem 1

Before we start analyzing the privacy property of our protocol, we have to discuss the implicit use of DHIES in our protocol. DHIES [3] is a family of public-key encryption schemes using symmetric encryption, message authentication, and hashing (key derivation). The security proof for the IND-CCA2 security of DHIES [3] either relies on the strong DH assumption and requires the hash function to be modeled as a random oracle (which gives more freedom in the choice of the hash function) or the oracle DH assumption (which somehow restricts the choice) but without relying on the random oracle heuristic. Standardization bodies [1, 2] thereby consider the hash function to be replaced with a key-derivation function that can also additionally be parametrized by some shared (public) information.

For our analysis below we thus assume that this variant (corresponding to the instantiation proposed by standardization bodies) still provides IND-CCA2 security. Namely, as tag and reader derive different key sets, we consider that DHIES can be used with two different key derivation functions KDF_B and KDF_T (so we are considering two members of the family of DHIES schemes) and furthermore the key derivation function takes additional common shared information N_T and N_B as input (as it is also the case for standardized versions of DHIES).

Consequently, we formally model the access to the challenge and the decryption oracle of an IND-CCA2 challenger for DHIES by parametrizing it with additional public information required by the internally used key-derivation function, namely $\mathsf{KDF}_\mathsf{B}(P, N_T, N_B)$ and $\mathsf{KDF}_\mathsf{T}(P, N_T, N_B)$.

Below, in our analysis we use what we call the n-DDH assumption, which is obviously equivalent to the DDH assumption. Basically, n-DDH states that it is hard to distinguish the distributions $(U = uP, V_1 = v_1P, \ldots, V_n = v_nP, W_1 = w_1P, \ldots, W_n = w_nP)$ where either $w_i = uv_i$ or w_i are random for all $i \in [n]$.

We prove that our protocol is strong private against wide adversaries, using a sequence of games, where we denote the event that the adversary $\mathcal{A}$ wins Game i by S_i. Let $\mathcal{C}_T$ and $\mathcal{C}_B$ be two lists that store challenge ciphertexts (plus additional information) received from a multi-query IND-CCA2 challenger and asked on behalf of a tag or the backend respectively.

Proof.
Game 0. The original game.

Game 1. We slightly modify the Game 0. Basically, for the third and the fourth pass of the protocol we do not compute the respective DHIES ciphertext for the correct shared key, but with respect to a randomly generated shared key. Since the adversary cannot obtain the volatile state from drawn tags, this cannot be detected under the DDH assumption. In particular, we modify the `SendTag` and `SendReader` oracles as follows:

- $\texttt{SendTag}(vtag, m) \rightarrow m'$: If the input message m is of type IKE_SA_INIT_RSP, choose $r' \xleftarrow{R} \mathbb{Z}_p$ and compute the DHIES ciphertext with respect to the DH key $Q = r' \cdot K_B$. Retrieve R from the volatile state of $vtag$ in session π and return (R, c', t') as ciphertext.
- $\texttt{SendReader}(\pi, m) \rightarrow m'$: If the input message m is of type IKE_AUTH_REQ, choose $r' \xleftarrow{R} \mathbb{Z}_p$ and compute the DHIES ciphertext with respect to the DH key $Q = r' \cdot K_B$. Retrieve R from the volatile state of $vtag$ in session π and return (R, c', t') as ciphertext.

Transition: Game 0 and Game 1. A distinguisher between Game 0 and Game 1 yields a distinguisher for n-DDH. Let n be the number of queries of the adversary to the `SendTag` and `SendReader` oracle (for the second pass each) and let $(U = uP, V_1 = v_1P, \ldots, V_n = v_nP, W_1 = w_1P, \ldots, W_n = w_nP)$ be a n-DDH instance padded from a DDH instance (U, V_1, W_1). Then, set $K_B \leftarrow U$. Let $R_i \leftarrow V_i$ be the value used in the i-th query and use W_i as DH key to produce a DHIES ciphertext in `SendTag` or `SendReader` respectively. If the distinguisher reports Game 0, then we have a valid (n-)DDH instance and an invalid (n-)DDH instance otherwise.

Game 2. We only make a conceptual change in that K_B is set to the public-key of a multi-query IND-CCA2 challenger of DHIES, the ciphertexts in `SendTag` and `SendReader` are queried from the challenge oracle of the IND-CCA2 challenger (and R is replaced as in Game 1) and ciphertexts that have not been produced within `SendTag` or `SendReader` are forwarded to the decryption oracle of the IND-CCA2 challenger.

- $\texttt{SendTag}(vtag, m) \rightarrow m'$: Parse the message m:
  - if it is of type IKE_SA_INIT_RSP containing (K_B, N_B), then retrieve the tag identities T_i and T_j corresponding to $vtag$, compute $a_i \leftarrow \mathsf{Mac}_{k_{PSK_i}}(①|N_B|ID_i)$ and $a_j \leftarrow \mathsf{Mac}_{k_{PSK_j}}(①|N_B|ID_j)$, and generate two messages $m_0 = ID_i|a_i|\ldots$ and $m_1 = ID_j|a_j|\ldots$, respectively. Then, query the IND-CCA2 challenger with (m_0, m_1) and (N_B, N_T) for $\mathsf{KDF_T}$, which returns the encryption of either m_0 or m_1 as a challenge ciphertext $C^* = (R', c, t)$. Add (R, R', c, t) to the list $\mathcal{C}_T$, where R is taken from the volatile state of $vtag$ in the corresponding session. Return the ciphertext $C^* \leftarrow (R, c, t)$ where R is taken from the volatile state of $vtag$ in the corresponding session.
  - if it is of type IKE_AUTH_RSP, then simply record m as the last-pass message for the session corresponding to $vtag$ (and it can then be processed by the `Result` oracle).
- $\texttt{SendReader}(\pi, m) \rightarrow m'$: Parse the message m:
  - if it is of type IKE_AUTH_REQ, then check for the received ciphertext $C^* = (R, c, t)$ whether $(R, \cdot, c, t)$ is contained in $\mathcal{C}_T$.
    - If this is the case, retrieve the tag identities T_i and T_j corresponding to $vtag$ in session π, compute $a'_i \leftarrow \mathsf{Mac}_{k_{PSK_i}}(②|N_T|BID)$ and $a'_j \leftarrow \mathsf{Mac}_{PSK_j}(②|N_T|BID)$, and generate two messages $m_0 = BID|a_i|\ldots$ and

$m_1 = BID|a_j|\ldots$, respectively. Then, query the IND-CCA2 challenger with (m_0, m_1) and (N_B, N_T) for $\mathsf{KDF_B}$, which returns the encryption of either m_0 or m_1 as a challenge ciphertext $C^* = (\hat{R}', \hat{c}, \hat{t})$. Add $(R, \hat{R}', \hat{c}, \hat{t})$ to the list $\mathcal{C}_B$. Return the ciphertext $C^* \leftarrow (R, \hat{c}, \hat{t})$ (note that R is from the volatile state of *vtag* in the corresponding session).

  - ∗ If C^* is not contained in $\mathcal{C}_T$, then send the ciphertext C^* to the decryption oracle of the multi-query IND-CCA2 challenger. Then parse the response as in the real game. If it is a valid message, proceed as above and otherwise return ERROR.
- otherwise proceed as in the real game.

– `Result`(π): Check if there is a session π with a last-pass message and return $\perp$ if this is not the case. Otherwise, check if for the corresponding ciphertext $C^* = (R, c, t)$, the tuple $(R, \cdot, c, t)$ is contained in $\mathcal{C}_B$. If this is the case return `true`. If not, query the decryption oracle of the multi-query IND-CCA2 challenger with ciphertext C^* and parse the returned message as in the real oracle and return whatever the real oracle would return.

Analysis of Game 2. It is clear that we have $\mathsf{Pr}[S_2] = \mathbf{Adv}^{ind-cca2-mq}_{DHIES,\mathcal{A}}(\kappa)$ and $\mathsf{Pr}[S_2] = \mathsf{Pr}[S_1]$ as this is only a bridging step, as well as $|\mathsf{Pr}[S_1] - \mathsf{Pr}[S_0]| \leq \epsilon_{\mathrm{DDH}}(\kappa)$ as we have shown that distinguishing these games yields a DDH distinguisher. Consequently, we have that $\mathsf{Pr}[S_0] \leq \mathbf{Adv}^{ind-cca2-mq}_{DHIES,\mathcal{A}}(\kappa) + \epsilon_{\mathrm{DDH}}(\kappa)$ which concludes the proof. □

B Proof of Theorem 2

In the following we analyze the security of the proposed protocol. Recall, that security requires secure tag authentication as well as secure reader authentication which informally says that any adversary is unable to output a transcript of a successful (accepting) authentication attempt (for either the tag or the reader) such that there is no matching session (for the respective counterpart) recorded in the environment.

Subsequently, we provide a proof for Theorem 2 which shows that given an adversary $\mathcal{A}$ that wins the authentication game with non-negligible advantage, we show how to create an adversary $\mathcal{A}'$ that wins the sUF-CMVA game of the used MAC with non-negligible advantage. In the proof below, we assume that the nonces N_B and N_T are unique throughout the game.

Proof. The adversary $\mathcal{A}'$ simulates the system $\mathcal{S}$ for the adversary $\mathcal{A}$ and let us assume w.l.o.g. that $\mathcal{A}$ makes q_{cr} calls to the corrupt oracle and $q_{ct} = q_{cr} + 1$ calls to the `CreateTag` oracle. When it runs $\mathsf{SetupReader}(1^\kappa)$ to obtain BID, $\mathbb{G}$ and (k_B, K_B), it guesses an index $i^* \in [q_{ct}]$ uniformly at random. $\mathcal{A}'$ simulates the oracles as follows (note that we omit the bit b, i.e., `DrawTag` always selects T_i (cf. Section 4.2), and omit IKEv2 specific parts of the messages that are not relevant).

– `CreateTag`(`ID`) $\rightarrow T_i$: This oracle is executed as in the real game (let us denote the tag associated to the i^*'th call by T_{i^*}).
– `Launch`(*vtag*): This oracle is executed as in the real game and in particular samples an unused session identifier π , $r_T \xleftarrow{R} \mathbb{Z}_q$, $N_T \leftarrow \{0,1\}^\lambda$, computes $R \leftarrow r_T \cdot G$ and $P \leftarrow r_T \cdot K_B$. Then, it records (r_T, R, P, N_T) as the volatile state of the tag associated to *vtag* and returns (π, m) with $m := (R, N_T)$.

- $\texttt{DrawTag}(T_i, T_j)$: This oracle is executed as in the real game.
- $\texttt{Free}(vtag)$: This oracle is executed as in the real game.
- $\texttt{SendReader}(\pi, m) \rightarrow m'$: This oracle is simulated as follows, where we discuss every message type subsequently (if the wrong round message for a session π is provided the oracle returns $\perp$):
  ①: Let $m := (R, N_T)$, proceed as in the real game and return $m' \leftarrow (K_B, N_B)$.
  ③: Let $m := C$, proceed as in the real game, but if π is associated to tag T_{i^*}, then the value a is fed into the Vrf oracle of the MAC challenger to check the validity. Furthermore, in case that a is a valid MAC, a' is obtained via a call to the Mac oracle of the MAC challenger. Return ERROR on error or $m' \leftarrow C'$ on success.
- $\texttt{SendTag}(vtag, m) \rightarrow m'$: This oracle is simulated as follows, where we discuss every message type subsequently (if the wrong round message for a session π is provided the oracle returns $\perp$):
  ②: Let $m := (K_B, N_B)$, proceed as in the real game, but if $vtag$ is associated to tag T_{i^*} the value a is obtained via a call to the Mac oracle of the MAC challenger. Finally, return $m' \leftarrow C$.
  ④: Let $m := C'$, proceed as in the real game, but if $vtag$ is associated to tag T_{i^*}, then the value a' is fed into the Vrf oracle of the MAC challenger to check the validity. Write either ERROR on error or OK on success on the output tape of $vtag$.
- $\texttt{Result}(\pi)$: This oracle returns the content of the output tape of $vtag$ associated to π.
- $\texttt{Corrupt}(T_i)$: This oracle is simulated as in the real game, i.e., returns the non-volatile memory (TID_i, k_{PSK_i}) of tag T_i. However, if this oracle is queried for tag T_{i^*}, then the simulation is aborted.

$\mathcal{A}$ eventually outputs $trans$ s.t. $\mathcal{P}(trans)$ yields true and wins with probability at least $\epsilon(\kappa)$.

It is clear that if $\mathcal{A}'$ did not abort, the view of $\mathcal{A}$ in the simulation is identical to the view of $\mathcal{A}$ under a real attack and independent of the choice i^*. To complete the proof we need to calculate the probability that $\mathcal{A}'$ did not abort during the simulation. Therefore, we observe that $\mathcal{A}'$ aborts if event $\mathsf{E_1}$, meaning that $\mathcal{A}$ queries $\texttt{Corrupt}$ for i^*, happens. We obtain that the probability $\mathsf{Pr}[\neg\mathsf{E_1}] \geq \frac{1}{q_{ct}}$. Now, we know that $\mathcal{A}'$ is successful, whenever $\mathsf{E_1}$ does not happen, and $\mathsf{E_2}$, meaning that $\mathcal{A}$ produces a valid forgery, as well as $\mathsf{E_3}$, meaning that $\mathsf{E_2}$ happens and $\mathcal{A}'$'s guess is correct, happen simultaneously. We need to bound the probability $\mathsf{Pr}[\neg\mathsf{E_1} \wedge \mathsf{E_3}] = \mathsf{Pr}[\mathsf{E_2}|\neg\mathsf{E_1}] \cdot \mathsf{Pr}[\neg\mathsf{E_1}] \cdot \mathsf{Pr}[\mathsf{E_3}|\neg\mathsf{E_1} \wedge \mathsf{E_2}]$. We know that $\mathsf{Pr}[\neg\mathsf{E_1}] \geq \frac{1}{q_{ct}}$, $\mathsf{Pr}[\mathsf{E_2}|\neg\mathsf{E_1}] \geq \epsilon(\kappa)$ as well as $\mathsf{Pr}[\mathsf{E_3}|\neg\mathsf{E_1} \wedge \mathsf{E_2}] = \frac{1}{q_{ct}}$. Consequently, we obtain that $\mathsf{Pr}[\neg E_1 \wedge E_3] \geq \frac{\epsilon(\kappa)}{q_{ct}^2}$.

Now if $\mathcal{A}'$ is successful, we know that the transcript contains at least one message from $\mathcal{R}$ to T_{i^*} (in case of reader authentication) and vice versa in case of tag authentication such that the corresponding message ②$|N_T|BID$ or ①$|N_B|TID$ respectively, have not been queried to the Mac oracle (since the messages to be MACed are distinct due to the choice of the nonces). Since, this however constitutes a valid forgery for the MAC scheme and the MAC scheme is assumed to be sUF-CMVA secure, $\epsilon(\kappa)$ must be negligible, which concludes our proof. □