A holistic approach to RFID security and privacy

Evangelos Rekleitis, Panagiotis Rizomiliotis, and Stefanos Gritzalis
Department of Information and Communications Systems Engineering, University of the Aegean
Karlovassi, Samos, 83200,Greece
Email: {erekl,prizomil,sgritz}@aegean.gr

***Abstract*—RFID technology constitutes an important part of what has become known as the IoT; i.e accessible and interconnected machines and everyday objects that form a dynamic and complex environment. In order to be able to secure the IoT in a cost-efficient manner we need to build security and privacy into the design of its components. Thus, in this paper, we first introduce the use of security and privacy policies that can offer fine granularity and context-aware information control in RFID systems, and with this in mind, we propose a novel secure and privacy preserving tag management protocol to implement such policies. The new protocol has a modular design in order to support all the basic management operations (tag authentication, delegation and ownership transfer), while imposing minimal hardware and computational requirements on the tag side.**

I. Introduction

The term Internet of Things describes a vision of a tighter integration between the physical and the virtual world. Building on the rapid growth of the Internet, visionaries imagined a world where everyday objects (things) and machines will be interconnected and networked, revolutionizing our way of life. The increase of available data and the emerging new ways of interacting and managing with everyday objects will bring an unpresented level of automation.

The IoT is expected to form a dynamic and complex environment, consisting of some billions of networked and interrelated things and machines. This vision raises many security and privacy concerns, as today's tools and techniques might prove not enough to ensure a safe IoT. This comes at no surprise, considering the difficulty at which we provide security and privacy in current systems. It is therefore crucial that IoT components are designed from their inception with a privacy- and security-by-design mindset and comprehensively include user requirements [1]. Already several technologies exist that are to become the basis of the IoT, such as IPv6, web services, SOA, Radio Frequency Identification (*RFID*) etc. Especially regarding RFID, which is the focus of this research, there is an on-going effort to provide a secure and privacy-respecting system. More precisely, several protocols have been proposed aiming to provide secure tag management operations, like tag authentication and ownership transfer. However, the vast majority of these proposals offer standalone security services and do not consider the security and privacy of the tag in a unified way.

In this paper we propose mechanisms that can achieve usable security and privacy in an RFID system. First we discuss the application of security and privacy policies to provide fine-grain access control to tags information in the back-end and control tag operations. Then, in this context, we introduce a secure management protocol for low-cost RFIDs. The protocol has a modular design, supporting tag authentication, secure delegation and privacy respecting ownership transfer, as different operations, supporting a policy-based fine-grain access control to the tag.

The paper is organized as follows. In section II, we define the notion of RFID and identify a set of important security and operational requirements that a protocol needs to satisfy and discuss related research work. In section III we discuss the use of security and privacy policies to provide fine-grain control of both tag operations (tag data) and object information. In section IV, we describe our proposal for a modular secure tag management protocol. Section V contains the security evaluation of the new protocol. Finally, in Section VI, we present some concluding remarks and provide directions for future research work.

II. Background

RFID is a sensor-based technology, used, primarily, to identify and track products or living organisms [2]. An RFID system can be viewed as consisting of two components: a front-end and a back-end (tag repository) part [3]. The front-end consists of embedded integrated circuit (IC) tags (transponders) that can be queried by reader devices (transceivers); while the back-end of a server infrastructure that manages the tag/object related information. In its simplest form when a reader queries a tag, the tag responds with an ID thus identifying the tagged object.

RFID tags may, either be self-powered (active) or require power from an external source (passive), usually the reader, or a hybrid, using both internal and external power sources. Tag related information can be grouped into tag data and object information. Tag data include data that support tag operations, like tag secrets (keys), unique identifiers etc. On the other hand, object information comprises of data related to the tagged object (e.g. description, owner, manufacturer, etc.) or the supported actions and services (e.g. physical access control, inventory management, etc.)

To make RFID systems economically viable, strict restrictions have been placed, mainly, on the tag side, whose implementation has to be power, space and time efficient. However, these restrictions cause sever security and privacy problems, since well known and trusted solutions, like public-key cryptography, are no longer applicable, and efficient alternatives are required.

In [4], Chien proposed a rough classification of RFID authentication protocols based on the computational cost and the operations supported by the tag. As shown in table I, we can distinguish four protocol classes, viz. ‘full-fledged’, ‘simple’, ‘lightweight’ and ‘ultralightweight’; with diminishing hardware requirements, respectively. In order to protect tag holders’ privacy and provide adequate security we identify five important security requirements that a security protocol should satisfy:

- **Resistance to Tag impersonation:** an adversary should not be able to impersonate a legitimate tag to the reader.
- **Resistance to Reader impersonation:** an adversary should not be able to impersonate a legitimate reader/server to the tag.
- **Resistance to Denial of Service (DoS) attacks:** manipulating or blocking communication during a given number of sessions between the tag and the reader should not prevent any future normal interaction between the legitimate reader and tag. This kind of attacks are also called desynchronization attacks.
- **Indistinguishability** (**tag anonymity**): tag output must be indistinguishable from truly random values. Moreover, they should be unlinkable to the static ID of the tag. To achieve a stricter notion of tag anonymity, we further define:
  - **Forward security/untraceability:** Even if an adversary acquires all the internal states of a target tag at time t, she should not be able to ascribe past interactions, that occurred at time $t' < t$, to the said tag.
  - **Backward security/untraceability:**[1] similarly to forward security, it requires that even if an adversary gains knowledge of a tag’s internal state at time t, she should not be able to ascribe future/subsequent interactions, that occur at time $t' > t$, to the said tag.

The set of desirable tag management operations contains:

- **Tag authentication:** the reader/back-end system should be able to authenticate the tag.
- **Revocable access delegation:** (aka tag delegation), the capability to allow a third party, tag authentication and read access to an owned tag, while maintaining the right to revoke this privilege, under some predefined conditions.
- **Ownership transfer:** the capability to pass ownership of a tag to a third party, without compromising backward untraceability for the said party, or forward untraceability for the previous owner.
- **Permanent and temporal tag invalidation:** more commonly known as kill and sleep operations; were initially proposed to offer a minimal degree of command over the tag. A legitimate tag owner can issue a command to disallow the tag from emitting any signals; in the case of the sleep operation this ban of communication can easily be revoked by the owner. Implementing them is trivial and it is obvious that these operations can also be achieved by physical means, e.g. breaking the tag or placing them in a faraday cage.

While an ultralightweight solution would be most welcomed, unfortunately most, such, proposed protocols have been shown vulnerable to attacks. Vajda and Buttyán, in [6], proposed a set of extremely lightweight challenge-response authentication algorithms that by design could be broken by a powerful attacker. Peris-Lopez et al. designed a series of very efficient ultralightweight authentication protocols (viz. LMAP [7], M^2AP [8] and EMAP [9]), using simple bit-wise operations (XOR, OR, AND) and addition $\bmod m$. But these schemes where, also, successfully attacked by Li and Deng [10] and Li and Wang [11], who found that a powerful adversary can mount a de-synchronization and a fulldisclosure attack against all three protocols and proposed some improvements, and by Barasz et al., who described a full-disclosure passive attack (eavesdropping) against LMAP [12] and M^2AP [13]. Chien and Huang [14], further, found weakness in Li-Wang’s improved schemes. Toiruul et al. proposed another ultralightweight authentication protocol, based on modular exponentiation, whose traceability was attacked by Hernádez-Castro et al. using a metaheuristic-based attack [15]. Similarly, a protocol by Chien in [4] was successfully attacked by Phan in [16], where it was shown that a passive attacker could track a tag by obtaining information about its static ID. The final blow, on ultraligthweight protocols, came from Alomair and Poovendran, who contacted a study [17] in which they claimed that ”relying only on bitwise operation for authentication cannot lead to secure authentication in the presence of an active adversary” (sic)[2]. Respectively, in the lightweight camp, protocols have, as well, been notorious for their flaws. A striking example, is the series of corrections and counter-corrections proposed on a series of lightweight protocols based on the Learning Parity with Noise (LPN) problem. Another example being [18], were we demonstrated that a lightweight Song-Mitchel authentication protocol [19] could be successfully attacked by a passive adversary and proposed a simple correction. We therefore maintain a cautious stance as to the security, achievable by ultralightweight and lightweight protocols.

Going through the corpus of published research work on RFID security and privacy, we detect an uneven imbalance between offered services; even among *simple* RFID protocols. That is, the vast majority of published work proposes tag authentication protocols, while other important operations are less explored. Indeed, the bibliography is rather limited; viz. Molnar et al. [20] propose an authentication protocol using pseudonyms and secrets, organized in a tree structure, to offer secure ownership transfer and time-limited, recursive delegation; the tree scheme was compromised in [21]. Fouladgar et

[1] In some research work, e.g. [5], the terms are interchanged, i.e. backward security is called forward security.

[2] We stress, again, that the hardware constraints refer only to the tag; the reader can satisfy more complex requirements, e.g. a random number generator.

TABLE I
HARDWARE CLASSIFICATION OF RFID SECURITY PROTOCOLS

Class	Hardware Requirements (Cryptographic primitives)
full-fledged	conventional cryptographic functions; e.g. symmetric and/or asymmetric encryption algorithms
simple	cryptographic one-way hash function
lightweight	random number generator and simple functions; e.g. Cyclic Redundancy Code (CRC) checksum
ultralightweight	simple bitwise operations; e.g. XOR, AND, OR

al. [22] also used pseudonyms to construct an authentication protocol, where delegation lasts for a predetermined number of queries. And a similar protocol, supporting a limited kind of delegation, was proposed in [21]. Ownership transfer, by itself, is also addressed in [5], [23]–[28].

Hence, we believe that a shift of focus is needed. Instead of offering standalone security services, we propose a holistic approach, that is governed by security and privacy policies to allow secure tag/object management. To this end, first we describe an abstract framework for using policies to control tag information dissemination and then design a 'complete' *simple* protocol that covers all the identified (RFID tag) security and privacy requirements (such as data confidentiality, backward and forward untraceability, etc.); supporting in a unified way operations like tag authentication, tag ownership transfer and time-based tag delegation.

III. PRIVACY AND SECURITY POLICIES

While a RFID security protocol can help reduce information leakage of tag data, by itself it does not give to the user control over the disseminated tag/object information. A complete approach should provide the necessary tools to describe how and by whom resources may be used. By resources we mean both the tag data (secret keys, IDs etc.), the object related information and the tag devices.

Traditionally, resources are protected using access control techniques. For data resources, mechanisms like Access Control Lists (ACL), Capability-based access control, Mandatory Access Control (MAC), Role Based Access Control (RBAC) and more recently Attribute and Rule-based Access Control (ABAC and RuBAC), have been used in traditional systems.

Because of the envisioned dynamic and complex nature of RFID systems and the IoT, static approaches such as ACLs and RBAC, are deemed unsuitable. Instead research points out that rule and/or attribute based access control systems seem a more suitable candidate for such services [29]. RuBAC and ABAC access decisions are based on the evaluation of rules expressed in terms of attributes and obligations of the subject, action, resource and environment. This allows finer granularity and context-aware authorization were required, even when the involved entities don't have predefined relationships (in contrast an ACL mechanism would require that all entities be known in advanced).

Policies themselves are expressed through the use of policy languages that define specialized grammar, syntax and enforcement mechanisms; e.g. XACML [30]. There is a rich literature on policy languages [31], but a critical review of these is out of scope. In the remainder chapter we will provide a high level description of how a non-monolithic security protocol can be coupled with privacy and security policies to provide fine-grain control to the end users.

Assuming that an RFID tag has an abstract four step lifecycle from birth (creation) to death (end-of-life/recycling), as depicted bellow:

1) Creation: a tag is created, initialized (viz. given a (unique) identifier, secret and public data stored on tag etc.), and bound to a data entry on the managing back-end infrastructure (e.g. a database server or an intelligent agent [3] etc.)
2) Attachment: the tag is attached to an object (inanimated item or living organization) and the data entry is expanded to include information pertaining the tagged 'thing'; possibly in a new back-end managed by the object's owner.
3) Operation: the tag's daily usage, were authorized entities acquire access to the tag's operations (viz. tag querying, tag delegation, secret updating, ownership transfer) and information.
4) End-of-life: the tag is no longer usable and is (hopefully) recycled.

The governing policies come as a natural extension of the tag information stored in the back-end. Each tag, from its creation, may be bound to a policy that defines the attributes that an entity must hold, the obligation he/she must make and the conditions under which tag operations are allowed. When a tag is attached to an object, along with the object information, suitable policies will be created to control access to this data.

Assuming a generic RFID system that uses an RFID authentication protocol (e.g. the one described in section IV), we have the following scenario.

- When a tag query request first arrives to the managing back-end, a first layer policy will define whether the user/reader (requester) is allowed access to the back-end's services. If the user holds the needed attributes his query is forwarded to the back-end storage module that holds tag related information (viz. tag data, object information and privacy policies). Otherwise access is denied.
- At the tag information entry, a second layer security policy will be consulted to check if the requester is authorized to perform the specific operation (in this case tag authentication/query). If yes the operation proceeds. Otherwise access is denied.
- If the back-end does not have an entry for the queried

tag a relevant message is returned. The contents of the message depend on the requester's trust level; as a policy may define that certain entities are not entitled to learn whether a tag is not managed by the back-end.

- If the correct tag is found, a policy should define how much of the object information will be released to the requester.
- Tag protocols may support extra operations beyond simple tag authentication/query. Whether the requester is allowed to perform these depends again on tag policy. In essence, since these operations require that the back-end returns the result of certain computations/data (e.g. decrypting a ciphertext), the policies allow or disallow them by controlling access to these computations/data.

Although the use of privacy policies might prove beneficial, there are problems that need to be addressed first. Such include:

- Efficiency issues: This includes policy evaluation at the infrastructure, storage costs etc. [29].
- Policy and rule construction: Although many policies use the XML to provide a form that is not only machine readable, but can also be reviewed by human users; nonetheless this may become a barrier for non-technical users.
- Access control complexity: When moving from a closed well-managed RFID system to a highly dynamic, interconnected and complex system like the IoT, there is a considerable amount of complexity that will need to be expressed into the policies. A good balance between fine-grain control, usability, manageability and cost will need to be reached.
- Privacy issues regarding use of attributes: Attributes hold information about entities, releasing more attributes than necessary to gain access to a resource could lead to sensitive information disclosure.
- Interoperability: To achieve a unified IoT, not only heterogeneous RFID hardware, RFID protocols and back-end infrastructures but also policies will need to be able to communicate and operate with each other.

While fine-grain control is required, it is nonetheless assumed that in the general case policies won't differ in excess. A user will most probably group her items according to her privacy, security and usability needs. Thus, the labor of writing individual policies for each and every tag is greatly reduced.

In addition, the literature provides research on efforts made to construct machine readable policies using 'natural' language rule editors [32], allowing not only easy policy creation but also policy revision from the user. It is thus easy to envision interested organizations, such as privacy rights NGOs, providing ready made rules and policies for every day use. Tweaking grouping and generic policies will both provide the required level of control and abstraction needed.

Another challenging task is providing a privacy-preserving trust negotiation mechanism. Trust negotiation simply put is the bilateral exchange of digital credentials to establish trust gradually. When entities set up access policies and try to satisfy them by exchanging proofs that they hold the necessary attributes, they release sensitive objects (e.g. credentials) about themselves. Over the years researchers have proposed several mechanisms that try to build trust and at the same time preserve users privacy, including trust managing systems and attribute release strategies [33], [34].

Standardization efforts have been made to provide an interoperable environment, both in the hardware and software level. For example, the OASIS consortium has standardized an XML based access control language (XACML), but more research is needed on the interoperability (bridging services) of existing mainstream languages.

In the next section, a protocol with a modular design that supports all identified tag management operations, is presented. Being modular means that the owner can enforce fine-grained access control to the tag, by selectively allowing or disallowing specific tag operations. As already discussed, this selection could be automated with the use of suitable policies at the back-end, which would authorize tag operations by disclosing or withholding relevant tag data (i.e. secret values and cryptographic computations results).

IV. A Novel Protocol for Secure RFID Management

In this section, we describe a 'simple' tag management protocol. The proposed protocol supports all basic tag operations, viz. authentication, tag delegation and ownership transfer, while it covers the identified security and privacy requirements. More precisely tag delegation is achieved by using time-based and temporal pseudonyms, while privacy preserving ownership transfer is achieved by renewing the value of the secret key. The protocols falls into the 'simple' protocol class as it imposes limited hardware requirements on the tag side, as the tag must implement a secure one-way function $h(\cdot)$ and a pseudorandom number generator (random selection of an element from a finite set using a uniform probability distribution is denoted as $\in_{\mathbb{R}}$). In addition, the tag needs to share only two values with the back-end system, namely an l-bit secret value $secret$ and a time value $horizon$, which designates a specific point in time and is publicly known. Time is an important concept for the delegation of the tag and we assume that its representation comforts to the ISO 8601 international standard [35].

Figure 1 provides a concise schematic of the proposed protocol which presents all supported operations. To de-clutter the schematic we choose to depict the reader and the back-end as one entity and skip the command signals that the reader sends to the tag; in practice the reader would act as a middleman forwarding messages and might, also, be given the capability to generate certain data items, such as random nonces or timestamps. Especially when delegating tag access, the reader may act without the support of the original back-end (e.g. off-line mode). It is assumed that the communication protocol supports suitable command signals/codes that instruct the tag on the desired operation, ensuring the authenticity

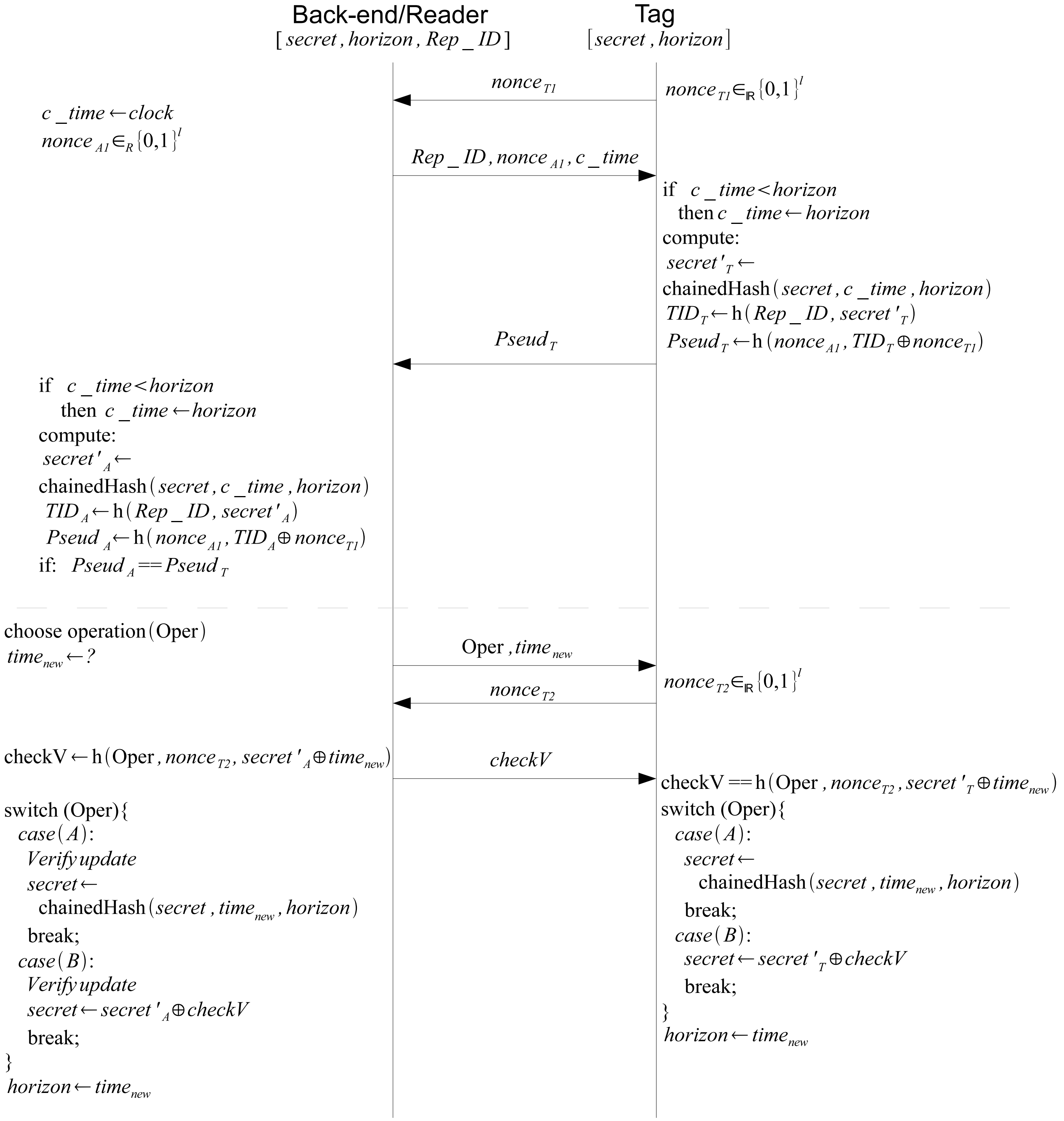
Back-end/Reader
$[secret, horizon, Rep\_ID]$
Tag
$[secret, horizon]$
$nonce_{T1}$
$nonce_{T1} \in_R \{0,1\}^l$
$c\_time \leftarrow clock$
$nonce_{A1} \in_R \{0,1\}^l$
$Rep\_ID, nonce_{A1}, c\_time$
if $c\_time < horizon$
then $c\_time \leftarrow horizon$
compute:
$secret'_T \leftarrow$
chainedHash$(secret, c_time, horizon)$
$TID_T \leftarrow \mathrm{h}(Rep\_ID, secret'_T)$
$Pseud_T \leftarrow \mathrm{h}(nonce_{A1}, TID_T \oplus nonce_{T1})$
$Pseud_T$
if $c\_time < horizon$
then $c\_time \leftarrow horizon$
compute:
$secret'_A \leftarrow$
chainedHash$(secret, c_time, horizon)$
$TID_A \leftarrow \mathrm{h}(Rep\_ID, secret'_A)$
$Pseud_A \leftarrow \mathrm{h}(nonce_{A1}, TID_A \oplus nonce_{T1})$
if: $Pseud_A == Pseud_T$
choose operation(Oper)
$time_{new} \leftarrow ?$
$\mathrm{Oper}, time_{new}$
$nonce_{T2}$
$nonce_{T2} \in_R \{0,1\}^l$
$\mathrm{checkV} \leftarrow \mathrm{h}(\mathrm{Oper}, nonce_{T2}, secret'_A \oplus time_{new})$
$checkV$
$\mathrm{checkV} == \mathrm{h}(\mathrm{Oper}, nonce_{T2}, secret'_T \oplus time_{new})$
switch (Oper){
$case(A)$:
$Verify\ update$
$secret \leftarrow$
chainedHash$(secret, time_{new}, horizon)$
break;
$case(B)$:
$Verify\ update$
$secret \leftarrow secret'_A \oplus checkV$
break;
}
$horizon \leftarrow time_{new}$
switch (Oper){
$case(A)$:
$secret \leftarrow$
chainedHash$(secret, time_{new}, horizon)$
break;
$case(B)$:
$secret \leftarrow secret'_T \oplus checkV$
break;
}
$horizon \leftarrow time_{new}$

Fig. 1. Compact schematic of Tag Query

and integrity of those is not explicitly discussed here, but the proposed methods and techniques can be suitably extended.

Motivated by the observation that practically, all the protocols supporting standalone tag operations (ownership transfer, revocable tag delegation etc.) begin with the authentication of the tag by the back-end/reader, our protocol is divided in two phases denoted by the dashed line. In the first phase, the tag is authenticated by the (owner's or delegated entity's) back-end/reader. In the second phase, initially the reader is authenticated by the tag and then some of the tag's data is updated. We distinguish the following cases: the secret key is updated according to a predefined function and the secret key or the publicly known value $horizon$ are reloaded with a specific value. All the tag operations are supported without influencing the number or the length of the exchanged messages. The protocol description follows:

Phase I: Tag Authentication

- **Tag $\longrightarrow$ back-end/Reader:** Generates and forwards a random $nonce_{T1}$.
- **Back-end/Reader $\longrightarrow$ Tag:** Forwards the identity Rep_ID of the reader's repository, an l-bit random $nonce_{A1}$ and the current time c_time.
- **Tag $\longrightarrow$ Back-end/Reader:** If c_time designates a point in time 'older than' $horizon$, then the tag computes a time-dependent secret key, using the key update process $secret'_T \leftarrow chainedHash(secret, c_{time}, horizon)$. Further, it computes the corresponding identifier $TID_T \leftarrow h(Rep\_ID, secret'_T)$. And finally, computes a pseudonym $Pseud_T \leftarrow h(nonce_{A1}, TID \oplus nonce_T)$. Then the tag forwards pseudonym $Pseud_T$. The function $chainedHash(s, t_1, t_2)$ is just the hashing of s repeated $t_1 - t_2$ times.
- **Back-end [Tag authentication phase]:** At the back-end, for each tag entry in the back-end server storage an individual time-depended key $secret'_A$, identifier ($TID_A \leftarrow h(Rep\_ID, secret'_A)$) and subsequently pseudonym ($Pseud_A \leftarrow h(nonce_{A1}, TID_A \oplus nonce_{T1})$) are computed. If the computed pseudonym is equal to the received one, then the tag has been successfully identified. Note that this step has to be computed twice, if no tag is authenticated. using the old value of the secret stored to prevent desynchronization attacks.

Phase II: Tag Data update

- **Back-end/Reader $\longrightarrow$ Tag:** Chooses the desirable operation and forwards it along with the new horizon value $time_{new}$.
- **Tag $\longrightarrow$ Back-end/Reader:** Generates and forwards an l-bit random $nonce_{T2}$.
- **Back-end/Reader $\longrightarrow$ Tag:** Computes a checksum value for the update ($checkV \leftarrow h(Oper, nonce_{T2}, secret'_A \oplus time_{new})$). Forwards the value $checkV$.
- **Tag [Data update phase]:** Checks if the received $checkV$ is equal to $h(Oper, nonce_{T2}, secret'_T \oplus time_{new})$; if yes based on the received operation it either updates the time-dependent secret, using the secret update process $chainedHash(secret, time_{new}, horizon)$, and then the tag uses the already computed $secret'_T$, or sets it to $secret'_T \oplus checkV$. Value $time_{new}$ is the new $horizon$.
- **Back-end [Data update phase]:** The back-end system stores both the new and the old values for the tag.

In day-to-day operations the tag's $horizon$ time value is expected to be set to the current time and the secret to be updated using the chained hash process. There are cases, however, where the owner may use a specific horizon value, different than c_time. This may be the case when we wish to invalidate a granted delegation, as we will explain at the end of this section. As well as cases, where the owner wants to disrupt linkability between subsequent secret values. This may be done to invalidate all granted delegations, to achieve secure ownership transfer (the new owner changes the secret information to avoid tracking from the previous owner), because we suspect that (at some point in the past) an adversary tampered with the tag (thus gaining access to its data) or for managerial purposes. We assume that these steps are performed during a 'safe slot', that is a time period during which no adversary, with knowledge of the current $secret$ value, eavesdrops the communication; further information is provided in section V.

All the basic tag management operations are supported. More precisely,

- **Tag Authentication:** The authentication can is achieved in the first phase, using the Rep_ID identity value of the owning back-end. The protocol may terminate here, with no further data sent to the tag, if authentication is the only desired operation.
- **Delegated Tag Authentication:** The owner of the tag can delegate, to another entity, the right to successfully authenticate the tag for a given period of time c_{time}. To achieve this it produces a time dependent tag identifier $TID = h(Rep\_ID, secret')$, where $secret' = chainedHash(secret, c_{time}, horizon)$. This identifier is unique for each system with identity Rep_ID for the given time period c_{time}. While $horizon \leq c_{time}$, the Rep_ID system can authenticate the tag using Phase I of the protocol.
- **Revocation of Tag Delegation:** By selecting a new value for $horizon$ greater than c_{time} the delegation is revoked. The owning back-end system can use the protocol with $Oper =$'A' (fig. 1) to update the value of $horizon$ with the new value $time_new$.
- **Ownership Transfer:** In Phase II, the owning back-end system can use the protocol with $Oper =$'B' to update the value of the secret value.

V. Protocol Security Analysis

To ease the security analysis, we first define an adversary model matrix, according to the available attack actions/capabilities [36]–[38]. More precisely, attackers can be distinguished into those that can tamper the tag (*corruptive*), i.e. can take the IC apart and extract, delete or alter data and those that cannot (*weak*). Further an attacker is characterized *wide* if she has access to (side channel) information about the outcome of the protocol (e.g. whether tag identification process was successful or not). Table II details the different adversarial types and highlights their capabilities.

For all defined adversary models, we impose two limitations:

- Existence of 'safe time slots'; that is there exist time periods (albeit small and few), during which no adversary eavesdrops or manipulates the tag-reader communication. This is an assumption made by all published protocol (implicitly or explicitly); as without it we would not be able to initialize the tags or perform secure ownership transfer.

TABLE II
ADVERSARY MATRIX

	Weak		Corruptive		
Actions	Passive	Active	Forward	Destructive	Strong
1. Eavesdrops	✓	●	●	●	●
2. Full control of network operations	—	✓	✓	✓	✓
3. Tag corruption at the end of the attack	—	—	✓	●	●
4. Destructive tag corruption	—	—	—	✓	●
5. Arbitrary tag corruption	—	—	—	—	✓
6. Side channel knowledge	wide	wide	wide	wide	wide

— The action is not available ✓ The action is available ● A more powerful action is available

- Existence of a secure communication channel between the reader(s) and the back-end system (back channel). We assume trusted and tested countermeasures have been taken to ensure the back channel.

According to Table II, a *wide-Passive* adversary is one that can only eavesdrop on the unencrypted communication between the tag and the reader and has knowledge of whether the tag authentication was successful or not. A *wide-Strong* adversary, on the other hand, is one that, not only, can manipulate the communication channel (according to the Dolev-Yao threat model; i.e. eavesdrop, corrupt, insert, etc. messages, mount MIM and replay attacks), but, as well, can corrupt the tag (altering and/or reading the data stored in the tag) whenever she sees fit.

It is important to clearly define actions 3–5 of Table II, to avoid any misconception. According to the forward privacy model, a *Forward* attacker is allowed to corrupt the data stored in the tag, but only at the end of the attack, so that no further active action happens after corruption. Whereas action 4 defines that a (*Destructive*) attacker may corrupt the tag, whenever he sees fit, but, after that, the tag is destroyed; the adversary may continue his attack, e.g. by simulating the tag. The 5th action, allows the attacker to access and manipulate the tag at his convenience, without further limitations.

For every identified security requirement, we will describe how it is satisfied by our protocol for the strongest possible adversary model.

- **Resistance to Tag and Reader Impersonation**: This requirement is studied under the weak adversary models; to prevent stronger (*corruptive*) attackers one would need to employ hardware anti-tampering techniques, which are out of scope. For an *Active* attacker the protocol can prevent malicious manipulation of the tag data. Any changes to the tag data or to the relevant data stored to the back-end are done after authenticating the received input and verifying its integrity. Replay attacks are thwarted by using random nonces. A MIM attack on the unprotected front channel, would not yield anything for the attacker as all secret information is enciphered.
- **Resistance to DoS**: In order to avoid desynchronization, the last two values of tag data, i.e. the current and previous secret and $horizon$ values, are stored at the back-end system for each tag.
- **Indistinguishability** (**tag anonymity**): The tags always reply using pseudonyms, which depend on the current secret and the exchanged random nonces. Even when the secret is not updated, the tag's reply will seem random, to those that don't have access to the current secret or temporal ID. Thus, the protocol can defend itself against active attackers. (For corruptive attackers v.i.)
- **Forward security/untraceability**: The protocol provides forward security, even under the Strong attacker model. If a corruptive attacker gains access to the tag data in time t, he cannot correlate past interactions to the tag (that were done using older keys), thanks to the one-wayness of the secret update process.
- **Backward security/untraceability**: As soon as a corruptive adversary gains access to the tag data in time t, he becomes able to trace all subsequent tag interactions — for the destructive adversary this type of attack is not applicable, since the tag is destroyed and no further interaction is possible. The only way to regain untraceability is by exploiting a safe slot to disrupt the chained-hash update process and change the secret to a new unrelated value.

VI. CONCLUSION

In this paper, we have discussed the use of security and policy languages to control access to tag information and tag operations, in order to allow for finer granularity and context-aware authorization in RFID systems. We believe that this is an interesting topic that needs more research, especially in integrating the so far proposed systems and mechanisms and transforming them into a suitable tool for use with RFID related operations. In the second part of the paper we described a unified novel tag management protocol that supports, among others, secure and privacy preserving tag authentication, delegation and ownership transfer. The protocol has minimal requirements on the tag side and follows a clear modular design.

REFERENCES

[1] "Communication to the european parliament, the council, the EESC and the committee of the regions: Internet of things - an action plan for europe," Commission to the European Parliament, the Council, the

European Economic and Social Committee and the Committee of the Regions, Tech. Rep., Jun. 2009.

[2] OECD, "Radio-Frequency identification (RFID): drivers, challenges and public policy considerations," Organisation for Economic Co-operation and Development (OECD), Paris, Tech. Rep. DSTI/ICCP(2005)19/FINAL, Mar. 2006.

[3] E. Rekleitis, P. Rizomiliotis, and S. Gritzalis, "An agent based back-end RFID tag management system," in *Lecture Notes in Computer Science LNCS*, S. Katsikas and J. Lopez, Eds. Bilbao, Spain: Springer, Aug. 2010.

[4] H.-Y. Chien, "Sasi: A new ultralightweight rfid authentication protocol providing strong authentication and strong integrity," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, pp. 337–340, 2007.

[5] C. H. Lim and T. Kwon, "Strong and robust RFID authentication enabling perfect ownership transfer," in *ICICS*, ser. LNCS, P. Ning, S. Qing, and N. Li, Eds., vol. 4307. Raleigh, North Carolina, USA: Springer, Dec. 2006, pp. 1–20.

[6] I. Vajda and L. Buttyán, "Lightweight Authentication Protocols for Low-Cost RFID Tags," in *Second Workshop on Security in Ubiquitous Computing – Ubicomp 2003*, Seattle, Washington, USA, October 2003.

[7] P. Peris-Lopez, J. C. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda, "LMAP: A Real Lightweight Mutual Authentication Protocol for Low-cost RFID tags," in *Workshop on RFID Security – RFIDSec'06*. Graz, Austria: Ecrypt, July 2006.

[8] ——, "M2AP: A Minimalist Mutual-Authentication Protocol for Low-cost RFID Tags," in *International Conference on Ubiquitous Intelligence and Computing – UIC'06*, ser. Lecture Notes in Computer Science, vol. 4159. Wuhan and Three Gorges, China: Springer-Verlag, September 2006, pp. 912–923.

[9] ——, "EMAP: An Efficient Mutual Authentication Protocol for Low-Cost RFID Tags," in *OTM Federated Conferences and Workshop: IS Workshop – IS'06*, ser. Lecture Notes in Computer Science, vol. 4277. Montpellier, France: Springer-Verlag, November 2006, pp. 352–361.

[10] T. Li and R. H. Deng, "Vulnerability Analysis of EMAP - An Efficient RFID Mutual Authentication Protocol," in *Second International Conference on Availability, Reliability and Security – AReS 2007*, Vienna, Austria, April 2007.

[11] T. Li and G. Wang, "Security Analysis of Two Ultra-Lightweight RFID Authentication Protocols," in *IFIP SEC 2007*. Sandton, Gauteng, South Africa: IFIP, May 2007.

[12] M. Barasz, B. Boros, P. Ligeti, K. Loja, and D. Nagy, "Breaking LMAP," in *Conference on RFID Security*, Malaga, Spain, July 2007.

[13] M. Bárász, B. Boros, P. Ligeti, K. Lója, and D. Nagy, "Passive Attack Against the M2AP Mutual Authentication Protocol for RFID Tags," in *First International EURASIP Workshop on RFID Technology*, Vienna, Austria, September 2007.

[14] H.-Y. Chien and C.-W. Huang, "Security of ultra-lightweight rfid authentication protocols and its improvements," *SIGOPS Oper. Syst. Rev.*, vol. 41, no. 4, pp. 83–86, 2007.

[15] J. C. Hernandez-Castro, J. Estevez-Tapiador, P. Peris-Lopez, J. A. Clark, and E.-G. Talbi, "Metaheuristic Traceability Attack against SLMAP, an RFID Lightweight Authentication Protocol," in *Proceedings of the 23rd IEEE International Parallel and Distributed Processing Symposium – IPDPS 2009*, Rome, Italy, May 2009.

[16] R. C. W. Phan, "Cryptanalysis of a new ultralightweight rfid authentication protocol — sasi," *IEEE Transactions on Dependable and Secure Computing*, vol. 6, pp. 316–320, 2009.

[17] B. Alomair and R. Poovendran, "On the authentication of RFID systems with bitwise operations," in *New Technologies, Mobility and Security NTMS'08*. Tangier, Morocco: IEEE, Nov. 2008, pp. 1–6.

[18] P. Rizomiliotis, E. Rekleitis, and S. Gritzalis, "Security analysis of the song-mitchell authentication protocol for low-cost rfid tags," *Comm. Letters.*, vol. 13, pp. 274–276, April 2009.

[19] B. Song and C. J. Mitchell, "RFID Authentication Protocol for Low-cost Tags," in *ACM Conference on Wireless Network Security, WiSec'08*, V. D. Gligor, J. Hubaux, and R. Poovendran, Eds. Alexandria, Virginia, USA: ACM Press, April 2008, pp. 140–147.

[20] D. Molnar, A. Soppera, and D. Wagner, "A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags," in *SAC*, ser. LNCS, B. Preneel and S. E. Tavares, Eds., vol. 3897. Kingston, Canada: Springer, Aug. 2006, pp. 276–290.

[21] T. Dimitriou, "rfiddot: Rfid delegation and ownership transfer made simple," in *Proceedings of the 4th international conference on Security and privacy in communication netowrks*, ser. SecureComm '08. New York, NY, USA: ACM, 2008, pp. 34:1–34:8.

[22] S. Fouladgar and H. Afifi, "An efficient delegation and transfer of ownership protocol for RFID tags," in *First International EURASIP Workshop on RFID Technology*, Vienna, Austria, Sep. 2007.

[23] Y. Jin, H. Sun, and Z. Chen, "Hash-based tag ownership transfer protocol against traceability," *E-Business Engineering, IEEE International Conference on*, vol. 0, pp. 487–492, 2009.

[24] Y. Zuo, "Changing hands together: A secure group ownership transfer protocol for rfid tags," *Hawaii International Conference on System Sciences*, vol. 0, pp. 1–10, 1899.

[25] J. Saito, K. Imamoto, and K. Sakurai, "Reassignment scheme of an RFID tags key for owner transfer," in *EUC Workshops*, ser. LNCS, T. Enokido, L. Yan, B. Xiao, D. Kim, Y.-S. Dai, and L. T. Yang, Eds. Springer, 2005, vol. 3823, pp. 1303–1312.

[26] K. Osaka, T. Takagi, K. Yamazaki, and O. Takahashi, "An efficient and secure RFID security method with ownership transfer," in *Computational Intelligence and Security: International Conference, CIS 2006, Guangzhou, China, November 3-6, 2006, Revised Selected Papers*. Springer, 2007, pp. 778–787.

[27] K. H. Koralalage, S. M. Reza, J. Miura, Y. Goto, and J. Cheng, "POP method: An approach to enhance the security and privacy of RFID systems used in product lifecycle with an anonymous ownership transferring mechanism," in *Proceedings of the 2007 ACM symposium on Applied computing SAC'07*. Seoul, Korea: ACM, Mar. 2007, pp. 270–275.

[28] B. Song, "RFID tag ownership transfer," in *Conference on RFID Security*, Budaperst, Hungary, Jul. 2008.

[29] E. Grummt and M. Müller, "Fine-grained access control for EPC information services," in *Proceedings of the 1st international conference on The internet of things*. Zurich, Switzerland: Springer-Verlag, 2008, pp. 35–49.

[30] "OASIS eXtensible access control markup language (XACML) TC," http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml #XACML20, Retrieved September 10.

[31] P. Kumaraguru, F. L. Cranor, J. Lobo, and S. B. Calo, "A survey of privacy policy languages," in *In SOUPS '07: Proceedings of the 3rd symposium on Usable privacy and security*. New York, NY, USA: ACM, Mar. 2007.

[32] B. Stepien, A. Felty, and S. Matwin, "A non-technical User-Oriented display notation for XACML conditions," in *E-Technologies: Innovation in an Open World*, ser. Lecture Notes in Business Information Processing, W. Aalst, J. Mylopoulos, N. M. Sadeh, M. J. Shaw, C. Szyperski, G. Babin, P. Kropf, and M. Weiss, Eds. Springer Berlin Heidelberg, 2009, vol. 26, pp. 53–64, 10.1007/978-3-642-01187-0_5.

[33] K. E. Seamons, M. Winslett, T. Yu, L. Yu, and R. Jarvis, "Protecting privacy during on-line trust negotiation," in *Proceedings of the 2nd international conference on Privacy enhancing technologies*. San Francisco, CA, USA: Springer-Verlag, 2003, pp. 129–143.

[34] M. Winslett, "An introduction to trust negotiation," in *Proceedings of the 1st international conference on Trust management*. Heraklion, Crete, Greece: Springer-Verlag, 2003, pp. 275–283.

[35] ISO 8601:2004, *Data elements and interchange formats – Information interchange – Representation of dates and times*. ISO, Geneva, Switzerland, 2004.

[36] D. Dolev and A. C. Yao, "On the security of public key protocols," in *Foundations of Computer Science, Annual IEEE Symposium on*, vol. 0. Los Alamitos, CA, USA: IEEE Computer Society, 1981, pp. 350–357.

[37] G. Avoine, "Adversary model for radio frequency identification," EPFL, Lausanne, Switzerland, Technical Report LASEC-REPORT-2005-001, Sep. 2005.

[38] S. Vaudenay, "On privacy models for RFID," in *Advances in Cryptology - Asiacrypt 2007*, ser. Lecture Notes in Computer Science. Springer-Verlag, Dec. 2007, pp. 68–87.