J Intell Manuf (2014) 25:273–281
DOI 10.1007/s10845-012-0655-4

Unbalanced states violates RFID privacy

Imran Erguler · Emin Anarim · Gokay Saldamli

Received: 3 October 2011 / Accepted: 25 April 2012 / Published online: 20 May 2012

Abstract Designing privacy preserving authentication protocols for massively deployed Radio Frequency IDentification (RFID) systems is a real world challenge that have drawn significant attention from RFID community. This interest yields considerable amount of proposals targeting to overcome the main bottleneck (i.e. the exhaustive search over the list of all tag entries) which appears in the back-end database for large-scale RFID tag deployments. A class of these proposals contains RFID protocols where the server authenticates the tag in a negligible constant/sub-linear time for a more frequent normal state and needs a linear search in a rare abnormal states. In this study, however, we show that such protocols having unbalanced states are subject to side-channel attacks and do not preserve the RFID privacy. To illustrate this brutal security flaw, we conduct our analysis on different RFID protocols.

Keywords RFID authentication protocols · Side-channel attacks · Untraceability

I. Erguler
TUBITAK BILGEM-UEKAE, PO Box 74, 41470 Gebze, Kocaeli, Turkey
e-mail: ierguler@uekae.tubitak.gov.tr

E. Anarim
Electrical-Electronics Engineering Department, Bogazici University, 34342 Bebek, Istanbul, Turkey
e-mail: anarim@boun.edu.tr

G. Saldamli (✉)
MIS Department, Bogazici University, 34342 Bebek, Istanbul, Turkey
e-mail: gokay.saldamli@boun.edu.tr

Introduction

Radio Frequency IDentification (RFID) technology is the main drive behind the pervasive computing and has been applied in various fields such as supply-chain management, inventory monitoring, payment systems, automobile immobilizers, and medical management. Because of its low production costs and tiny size, RFID gadgets are considered as a replacement technology for bar codes and other means of traditional identification tools.

Typically, an RFID system incorporates three components: Tags, one or more readers, and a back-end server. On top this hardware, a set of networking rules including the authentication protocols reside. Despite their advantages, RFID systems have some computational constraints mostly driven by the cost concerns of low-cost RFID tags. It hampers use of public key cryptography and only permits to have security schemes based on symmetric key primitives at the tag side. Indeed, this fact makes design of a fully privacy-preserving authentication protocol as a challenging task. Solving this delicate task has taken attention from security community and numerous authentication protocols have been proposed like in Ohkubo et al. (2003), Henrici and Müller (2004), Molnar and Wagner (2004), Rhee et al. (2005), Dimitriou (2005), Karthikeyan and Nesterenko (2005), Nguyen Duc et al. (2006), Chien and Chen (2007), Ha et al. (2007b), Tsudik (2007), Song and Mitchell (2008), Shaoying et al. (2009). Note that, these protocols are much more restricted than what is needed for the applications ranging from wireless networks (Xie et al. 2010; Sarkar and Saha 2011; Wang et al. 2010), to smart grids (Ling and Masao 2011).

Although details can vary dramatically from one protocol to another, these RFID protocols can be classified depending on time complexity taken by the server for its overall computations in tag identification process. In Alomair and Poovendran (2010), such a classification is presented and the protocols are categorized into three groups based on the computational complexity performed by the server: Constant-time, logarithmic-time and linear-time protocols. As can be inferred from their names, for these RFID protocols the server accomplishes tag identification with complexity $O(1)$, $O(\log N)$ and $O(N)$ respectively, where N denotes the number of tags in the database.

In addition to these classes, some recent security protocols, which we call *unbalanced authentication protocols*, like in Ha et al. (2007a,b), Burmester et al. (2008), Chang and Wu (2009),Song and Mitchell (2010), have been proposed to reduce the computational load on the back-end database by defining different states for which the server accomplishes tag identification in different order of computational effort. In other words, these protocols allow tags tags to be in different states such that the server authenticates the tag in constant/sub-linear time in a more frequent normal state and needs a linear search in a rare abnormal states. Since the server identifies tags in $O(1)$ for most of the time, such protocols achieve computation efficiency at the server side. Nevertheless, an adversary can utilize this computation difference with respect to different states in breaking privacy of the system if he has access to side channel information that leaks the computational complexity of the back-end database. In this paper, we point out this security risk and show that such protocols fail to fulfill their privacy claims due to this security flaw.

The rest of this paper is organized as follows: In the next section, we introduce the notation and definitions that will be used in this study. In "Classification of RFID protocols based on server computational effort", we shortly examine the characteristics of RFID classes that are formed depending on the required computational complexity taken by the server. In "Related work", we briefly discuss related work. "Description of attack" describes our attack model in detail and proves that for an RFID system, having unbalanced states, an adversary can trace the tags. We investigate untraceability of some RFID protocols in "Analysis of some RFID protocols" and draw our conclusions in "Conclusions".

Definitions & notation

Unless otherwise is stated, the notation depicted in Table 1 is used throughout the paper:

Table 1 Notations

$\mathcal{T}$	RFID tag
$\mathcal{R}$	RFID reader
$\mathcal{DB}$	The back-end database
$\mathcal{A}$	Adversary
ID	Identity of a tag
HID	Hashed value of ID
PID	Previous identity of a tag
r_R	Random nonce generated by reader $\mathcal{R}$
r_T	Random nonce generated by tag $\mathcal{T}$
$H()$	One-way hash function
$e(), f(), g()$	Keyed one-way hash functions
$SecReq$	Secret update request message
$\|\|$	Concatenation operator
N	Number of tags in the database

Privacy definitions

Privacy, both in terms of tag anonymity and tag untraceability (an adversary should not able to recognize a tag he previously observed or interacted with), is a significant concern that needs to be addressed if RFIDs are to be as widely deployed as conceived by proponents (Ouafi and Phan 2008). Untraceability issue has been treated formally in different security models, notably driven by Avoine (2005), by Vaudenay (2007) and by Juels and Weis (2007). In this part, instead of giving whole detailed definitions we briefly describe here the basic ideas of Vaudenay's privacy model (Vaudenay 2007) and its an extended version the Avoine's privacy model (Avoine et al. 2010) which will be sufficient to describe our security analysis. The Juels–Weis privacy model is based on indistinguishability of the tags and indeed this makes it more practical. Nevertheless, Vaudenay's model is more flexible by allowing adversaries with variant degrees of capabilities. Thus, our privacy analysis delicately combines definitions of these two models and defines untraceability in terms of a privacy experiment by which an adversary could distinguish two different tags by using all accessible oracles. According to Vaudenay's model, an adversary $\mathcal{A}$ takes a public key K_P as input and can run the following oracles:

- $O^{\text{CreateTag}}(ID, b)$: This oracle creates a free tag, either legitimate ($b = 1$) or not ($b = 0$), with unique identifier ID. By convention, b is implicitly 1 when omitted.
- $O^{\text{DrawTag}}(distr) \rightarrow (t_1, b_1, \ldots, t_k, b_k)$: This oracle moves from the set of free tags to the set of drawn tags a tuple of k tags at random following the probability distribution $distr$. For each chosen tag, the oracle gives it a new pseudonym denoted t_i and changes its status from free to drawn. Finally, the oracle returns all the generated pseudonyms $(t_1, b_1, \ldots, t_k, b_k)$ in any order.
- $O^{\text{Free}}(t)$: This oracle moves the tag with pseudonym t from the status drawn to the status free. It makes this tag unreachable for $\mathcal{A}$.
- $O^{\text{Launch}} \rightarrow \pi$: This oracle makes the reader launch a new protocol instance π which is returned as the output.

- $O^{\text{SendReader}}(m,\pi) \rightarrow r$: This oracle sends a message m to the reader $\mathcal{R}$ for a protocol instance π. It outputs the response r from the reader.
- $O^{\text{SendTag}}(m,t) \rightarrow r$: This oracle sends a message m to the tag $\mathcal{T}$ with pseudonym t. It outputs the response r from $\mathcal{T}$.
- $O^{\text{Execute}}(t) \rightarrow (\pi, transcript)$: This oracle executes a complete protocol between $\mathcal{R}$ and $\mathcal{T}$ with pseudonym t. It returns the transcript of the protocol, i.e. the list of successive protocol messages.
- $O^{\text{Return}}(\pi) \rightarrow x$: When π is completed this oracle outputs $x = 1$ if the output of the reader is $\neq\perp$, and $x = 0$ otherwise.
- $O^{\text{Corrupt}}(t) \rightarrow tk_t$: This oracle returns the tag-dependent key tk_t of tag t. If t is no longer used after this oracle call, we say that t is destroyed.

In addition to these oracles, we describe two new oracles[1]

$O^{\text{Complexity}}(\pi,\alpha) \rightarrow \delta$: This oracle returns computational complexity performed by back-end database for its overall computations during the protocol instance π as δ in units of α, e.g. α may be relevant to time, power consumption, electromagnetic leaks, sound emission etc.
$O^{\text{SetState}}(S,t)$: This oracle sets state of the tag with pseudonym t to S.

Classification of RFID protocols based on server computational effort

In this section, we briefly describe different classes of RFID protocols—constant-time, logarithmic-time and linear-time protocols—in order to facilitate expression of Unbalanced Authentication Protocol.

Linear-time protocols

In a protocol of this class, the authentication of a tag imposes a linear search at the server side. As the number of the tags in the system increases, the server suffers from the heavy overloads in tag identification process and this results in the major disadvantage of this class—the efficiency and the scalability issue. The protocols presented in Ohkubo et al. (2003), Rhee et al. (2005), Nguyen Duc et al. (2006), Chien and Chen (2007), Song and Mitchell (2008), Weis et al. (2003) are some examples of this class.

Logarithmic-time protocols

In order to solve the scalability problem of linear-time protocols, Molnar and Wagner proposed to use a tree-based key space such that the tags hold a set of keys arranged in a tree (Molnar and Wagner 2004). A single particular path in the tree is assigned for each tag, while the server knows all the secrets. To identify a tag the reader performs a challenge-response protocol through the tree from its root to the leaves looking for a match to the tag's response. This scheme requires logarithmic time complexity, so the work for the server to identify a tag has complexity of $O(\log N)$. Nevertheless, in Avoine and Oechslin (2005) it is shown that the protocol could degrade the privacy if one or more tags are compromised.

Furthermore, inspired by the well-known "meet-in-the-middle" strategy used in the past to attack certain symmetric ciphers, Cheon et al. proposed a simple and efficient RFID tag identification and authentication technique that reduces server computation to $O(\sqrt{N}\log N)$ in Cheon et al. (2009). However, just like Molnar and Wagner's protocol, if one or more tags are compromised, the privacy of system is affected. For example, if secrets of t tags are compromised, then an adversary can identify $t^2 - t$ uncompromised tags in the system (Alomair and Poovendran 2010).

Constant-time protocols

For a protocol of this class, the server achieves tag identification in constant time. As discussed below, however, such protocols have significant security or privacy shortcomings.

Henrici and Müller (2004) proposed a scheme in which the server only needs to perform $O(1)$ work in tag identification process to reduce the computational load at the server side. A tag sends two hashed values as its response to a query, and updates its stored values, including its ID, after a successful authentication. Since a tag always replies with the same hashed ID before it is successfully authenticated, the scheme is vulnerable a degree of tag tracking (Chien and Chen 2007).

In Dimitriou (2005) proposed an RFID authentication protocol which needs $O(1)$ effort for a server to authenticate a tag. Nevertheless, this protocol is also vulnerable to a tag tracking attack, since the tag ID is not changed in case of an unsuccessful authentication session (Dimitriou 2005).

In Burmester et al. (2008) proposed an RFID authentication protocol that achieves tag identification in constant time by supporting constant keylookup, using a pseudo-random function. However, the protocol suffers from the security flaw such that after an unsuccessful authentication session, a tag reuses the same pseudonym in the following session (Song and Mitchell 2009).

In Song and Mitchell (2009) proposed a scalable RFID pseudonym protocol based on the protocol of Song and

[1] In Avoine's privacy model $O^{\text{Timer}}(\pi)$ oracle is defined. It outputs the time δ taken by the reader for its overall computations in protocol instance π. We extend functionality of this oracle for our needs as $O^{\text{Complexity}}(\pi,\alpha)$ which may give side channel information in disparate domains depending on α such as power consumption, sound, time etc.

Mitchell (2008), that takes $O(1)$ work to authenticate a tag. However, it is shown in Erguler and Anarim (2010) that the protocol is vulnerable to tag tracking and denial of service attacks.

Unbalanced authentication protocols

From the previous parts of this section we conclude that a linear-time protocol can provide demanded security and privacy conditions, but suffers from scalability in case of large-scale RFID deployments. Unlike linear-time protocols, constant-time and partially logarithmic-time protocols are performance efficient schemes. However, they have serious security and privacy flaws that contradict with their design objectives. From these remarks, some designers drawn lessons and unintentionally introduced a new class of RFID protocols according to which the back-end server accomplishes the tag identification in constant or sub-linear time for a more frequent normal state and needs a linear search in a rare abnormal states. To the best of our knowledge, this type of protocols have not been formally categorized as an RFID class in previous work, so we call such protocols as unbalanced authentication protocols (UAP).

Related work

Use of side channel information—e.g. computational time—in security analysis of RFID authentication protocols has been recognized in a number of reported studies and particularly in the following ones. In Juels and Weis (2007) introduced the idea that adversary may access to the protocol output which shows whether a reader succeeded to identify a legitimate tag or not. For instance, opening a door with a proximity card or acceptance of a payment card can give this information. Of course such an information give a hint to an adversary to distinguish two different tags, i.e. break the privacy of the protocol. Moreover, they mentioned that computation time of the reader can shed critical light on protocol design and showed O–TRAP protocol, described in Burmester et al. (2006), cannot provide strong privacy if this side channel information is used. The idea that the adversary knows whether a reader succeeded to identify a legitimate tag or not is also formalized in Vaudenay's privacy model (Vaudenay 2007). In Burmester et al. (2006), timing attacks have been briefly considered by Burmester et al. and the following has been claimed: "In particular the time taken for each pass must be constant. This can be done by inserting an artificial delay on the trusted server…". Recent studies presented by Erguler and Anarim (2010), Erguler et al. (2009) have benefited from variance in computational time of reader/server with respect to different tag states to distinguish the tags. By which, they have shown that two protocols described in Song and Mitchell (2009) and Ha et al. (2007a) are vulnerable to timing attack. Also, in Avoine et al. (2010) a privacy model extending the Vaudenay's one has formalized computational time of the reader. A new privacy level TIMEFUL which is determined by leaked information from the computational time of the reader is added to privacy levels of model in Vaudenay (2007). Recently, Erguler et al. (2011) analyzes database search mechanism of linear-time protocols and shows that such protocols are vulnerable to timing attacks that could easily jeopardize the systems untraceability criteria, if the database querying is performed through a static process.

In this study, we firstly introduce Unbalanced Authentication Protocols UAPs that have not been formally categorized as an RFID class in previous work and then within a formal model we prove that such protocols suffer from a side-channel attack which breaks privacy of the schemes. As a consequence, we stress that use of unbalanced states must be avoided in designing RFID authentication protocols.

Description of attack

Before we elaborate on our attacking strategies, we will first state our assumption that an adversary may access to all previously mentioned oracles.

Definition 1 Suppose that for an RFID protocol a tag $\mathcal{T}$ is allowed to be in one of the states of finite set $\mathbf{S}_{\mathcal{T}} = \{S_0, S_1, \ldots, S_k\}$, where $k \geq 1$, through interaction with the RFID readers. Then this protocol is called Unbalanced Authentication Protocol (UAP) and the states are named as unbalanced states, if there exists any two states $S_i, S_j \in \mathbf{S}_{\mathcal{T}}$ for which the server performs identification of the tag in different order of computational complexities.

Definition 2 Let $S_i, S_j \in \mathbf{S}_{\mathcal{T}}$ be two unbalanced states of an UAP. An adversary $\mathcal{A}$ having the ability to set a tag's state from S_i to S_j or vice versa is denoted by $\mathcal{A}^{Stat}$ and called a stateful–adversary.

As can be inferred from the above definition, a stateful–adversary $\mathcal{A}^{Stat}$ has access to the oracle $O^{\text{SetState}}(S, t)$.

Theorem 1 *Any RFID authentication protocol involving unbalanced states are traceable to a stateful–adversary.*

Proof To prove the statement we describe the following privacy experiment $\mathbf{Exp}_{UAP}^{priv}$:

1. $\mathcal{A}^{Stat}$ creates two legitimate tags using twice $O^{\text{CreateTag}}(ID)$ and calls $O^{\text{DrawTag}}(\frac{1}{2}, 2)$. So, he obtains two pseudonyms t_0 and t_1.
2. $\mathcal{A}^{Stat}$ puts the tag t_0 into state S_i and t_1 into state S_j by calling $O^{\text{SetState}}(S_i, t_0)$ and $O^{\text{SetState}}(S_j, t_1)$ respectively.

3. $\mathcal{A}^{Stat}$ gets $(\pi_0, transcript_0)$ and $(\pi_1, transcript_1)$ by calling $O^{\text{Execute}}(t_0)$ and $O^{\text{Execute}}(t_1)$ respectively.
4. $\mathcal{A}^{Stat}$ requests the computational complexities of each authentication by using $O^{\text{Complexity}}(\pi_0, \alpha)$ and $O^{\text{Complexity}}(\pi_1, \alpha)$ to obtain δ_0 and δ_1.
5. Again $\mathcal{A}^{Stat}$ puts the tag t_0 into state S_i and t_1 into state S_j with requesting $O^{\text{SetState}}(S_i, t_0)$ and $O^{\text{SetState}}(S_j, t_1)$.
6. $\mathcal{A}^{Stat}$ frees both of the tags by calling $O^{\text{Free}}(t_0)$ and $O^{\text{Free}}(t_1)$ and reaffects only one of them with $O^{\text{DrawTag}}\left(\frac{1}{2}, 1\right)$. The adversary gets a new pseudonym t_b^*.
7. $\mathcal{A}^{Stat}$ runs an instance protocol by calling $O^{\text{Execute}}(t_b^*) \rightarrow (\pi^*, transcript^*)$ and gets δ^* with requesting $O^{\text{Complexity}}(\pi^*, \alpha)$.
8. If $\delta^* = \delta_0$ $\mathcal{A}$ guesses $b = 0$ and decides $t_b^* = t_0$, otherwise guesses $t_b^* = t_1$.

Clearly, the success probability of the adversary in guessing b is 1, because δ_0 and δ_1 are outputs of different order of computational complexities and they are comparably different. Thus, a stateful–adversary can distinguish between two different tags, i.e. breaks privacy of the system.

This theorem shows an impossibility for obtaining any form of untraceability for an RFID system as long as it posses unbalanced states that can settled by an adversary. Notice that as can be inferred from the above privacy experiment, a stateful-adversary can distinguish two different tags without requesting the oracle $O^{\text{Corrupt}}(t)$.

Analysis of some RFID protocols

In this section, we apply our privacy experiment described in "Description of attack" to reveal traceability in some proposed RFID schemes that have unbalanced states in their protocol descriptions. Then, we point out that all of these protocols suffer from the presented side-channel attack and they fail to fulfill their untraceability claims. In description of the protocols, we omit whole details of the protocol processes and shortly give the protocol flow (interested readers may refer to corresponding study).

The scalable Song–Mitchell protocol and analysis

In Song and Mitchell (2010) have proposed a scalable RFID authentication protocol which we call SSM due to name of the authors. Initially, a secret l-bit string, s_j, and its hash value (computed by the server), $k_j = h(s_j)$, is attached to the tag entries $\mathcal{T}_j \in \mathcal{DB}$, where $h()$ is a one-way hash function. Moreover, for every tag $\mathcal{T}_j$, $\mathcal{DB}$ stores a hash-chain $\{x_0, x_1, \ldots, x_m\}$ where m is a positive integer, $x_t = e_k(x_{t-1})$ for $1 \leq t \leq m$ and x_0 is a random l-bit string. In case of a need to resynchronize the tag, $\mathcal{DB}$ further stores $\{\hat{s}_j, \hat{k}_j\}$ as the most recent secrets assigned to $\mathcal{T}_j$. On the other hand, each $\mathcal{T}_j$ stores k, x and a counter c, where x is initially set to x_0 and c is set to m. It is claimed that the system provides untraceability, authentication, and robustness against replay and spoofing attacks. The operation of the protocol can be divided into three cases:

(C1) For each of the first $m - 1$ queries of a tag, where m is a positive integer determined by the server, the protocol process requires just two message flows and only involves tag authentication. To authenticate a tag, the reader/server searches a look-up table, thereby taking only $O(1)$ work to identify and authenticate a tag, without needing a linear search. Indeed, thanks to this feature the server can authenticate a registered tag in constant time and hence the scheme provides desirable scalability properties.

(C2) On the mth query of a tag, as in Case 1 the tag is authenticated in constant time by the server. However, the server realizes that all identifiers in the look-up table for the corresponding tag were used in the previous queries. In order to maintain Case 1 operation, the server needs to update the secrets of the tag and generate new identifiers. Also following the update process, the server sends an additional message to initiate a secret update process at the tag side. At the end of these steps, for the first $m' - 1$ queries of the tag, the protocol again operates as in Case 1, where m' is new value of m.

(C3) If a tag is queried more than m times, which should not normally happen, then tag produces two messages with demanding a secret update. In this case, the authentication of the tag imposes a linear search with complexity $O(N)$, where N is number of tags in the back-end database. After this as in Case 2 the server updates the secrets of the tag and replies to the tag with an additional message to invoke a secret update procedure at the tag side. □

In secret update process, the server picks a random l-bit string s' and an integer m', and computes a key $k' = h(s')$ and a sequence of m' identifiers $x_i' = e_{k'}\left(x_{i-1}'\right)$ for $1 \leq i \leq m'$, where x_0' is set to x_m. Figure 1 summarizes protocol flow of SSM.

Proposition 1 *SSM protocol is vulnerable to the proposed attack and cannot provide untraceability.*

Proof The statement is confirmed, if we show that the SSM protocol is an UAP and a stateful–adversary $\mathcal{A}^{Stat}$ exists for this system. As given in the protocol description the server authenticates a registered tag in $O(1)$ if $c \neq 0$; we call this state as S_0. On the other hand, the server performs a linear search costing $O(N)$ in case of $c = 0$ and this state is represented as S_1. Note that two unbalanced states as S_0 and S_1 exist for SSM i.e. it is an UAP. Furthermore, an adversary

Data Base / Reader $[\hat{s}, \hat{k}, s, k, x_0, \cdots, x_i \cdots, x_m]$		**Tag** $[k, x, c]$
Generate r	r $\longrightarrow$	**If** $c \neq 0$, $M_T = f_k(r\|\|x)$ $x \leftarrow e_k(x)$, $c \leftarrow c - 1$
	$\{r, M_T, x\}$ $\longleftarrow$	
Case 1: Search for $x_t = x$ in the DB Check $M_T = f_k(r\|\|x_{t-1})$ **Case 2:** **If** $x = x_m$, $M_S = g_k(r\|\|M_T) \oplus (s\|\|k'\|\|m')$ Update secrets for Tag $\hat{s} \leftarrow s, \hat{k} \leftarrow k, s \leftarrow s', k \leftarrow k', x_0 \leftarrow x$ $x_t(1 \leq t \leq m) \leftarrow x'_t(1 \leq t \leq m')$	$\{r, M_S\}$ $\longrightarrow$	$(s\|\|k'\|\|m') = M_S \oplus g_k(r\|\|M_T)$ **If** $h(s) = k$, $k \leftarrow k'$ and $c \leftarrow m'$
	$\{r, M_1, M_2\}$ $\longleftarrow$ $SecReq$	**If** $c = 0$, Generate r_T $M_1 = f_k(r\|\|r_T)$ $M_2 = (r_T \oplus x)$
Case 3: Search for $x = x_m$ or x_0 for which $M_1 = f_k(r\|\|(M_2 \oplus x))$ $r_T = M_2 \oplus x$ **If** $x = x_m$, $M_S = g_k(r\|\|r_T) \oplus (s\|\|k'\|\|m')$ **If** $x = x_0$, $M_S = g_{\hat{k}}(r\|\|r_T) \oplus (\hat{s}\|\|k\|\|m)$ Update secrets for Tag $s \leftarrow s', k \leftarrow k', x_0 \leftarrow x$ $x_t(1 \leq t \leq m) \leftarrow x'_t(1 \leq t \leq m')$	$\{r, M_S\}$ $\longrightarrow$	$(s\|\|k'\|\|m') = M_S \oplus g_k(r\|\|r_T)$ **If** $h(s) = k$, $k \leftarrow k'$ and $c \leftarrow m'$

Fig. 1 The Song and Mitchell's scalable RFID authentication protocol

can put any tag into state S_0 or S_1 by running the following procedures respectively:

Procedure 1 *Setstate* S_0

```
1: For a selected tag with pseudonym $t_i$, $\mathcal{A}$ transmits some random nonce $r$ to $\mathcal{T}_i$ by calling $O^{\text{SENDTAG}}(r, t_i)$.
2: $\mathcal{A}$ repeats the previous step until $\mathcal{T}_i$ response contains $SecReq$, indicating a secret update request.
3: $\mathcal{A}$ calls the oracle $O^{\text{EXECUTE}}(t_i)$.
```

By using the above procedure, the tag updates its secrets and sets $c = m'$, where $m \neq 0$. Hence, it is guaranteed that the tag is in state S_0.

Procedure 2 *Setstate* S_1

```
1: For a selected tag with pseudonym $t_i$, $\mathcal{A}$ transmits some random nonce $r$ to $\mathcal{T}_i$ by calling $O^{\text{SENDTAG}}(r, t_i)$.
2: $\mathcal{A}$ repeats the previous step until $\mathcal{T}_i$ response contains $SecReq$, demanding a secret update.
```

After execution of *Setstate* S_1 procedure the tag will be in state S_1, because $c = 0$. Thus, in the next protocol run it requests a secret update which imposes a linear search at the server side. Notice that the adversary is able to put any selected tag into one of the unbalanced states. Therefore, the adversary is a stateful–adversary and can request the oracle $O^{\text{SETSTATE}}(S, t)$. From Theorem 1, it is apparent that SSM protocol is vulnerable to the proposed attack and cannot provide untraceability.

The Ha's protocol and analysis

Ha et al. (2007b) proposed a mutual authentication protocol based on a hash function. The protocol allows a tag to be in one of two states: In a synchronized state, the server authenticates the tag in constant time by using a look-up table, so requires only $O(1)$ work. However; in the case of a desynchronized state, the server needs to perform a linear search, i.e. $O(N)$.

Initially, the back-end database $\mathcal{DB}$ stores the ID, hashed values HID, and PID for each tag, while the tag keeps ID and state status flag $SYNC$. The protocol is depicted in Fig. 2 and a step by step description is also given below:

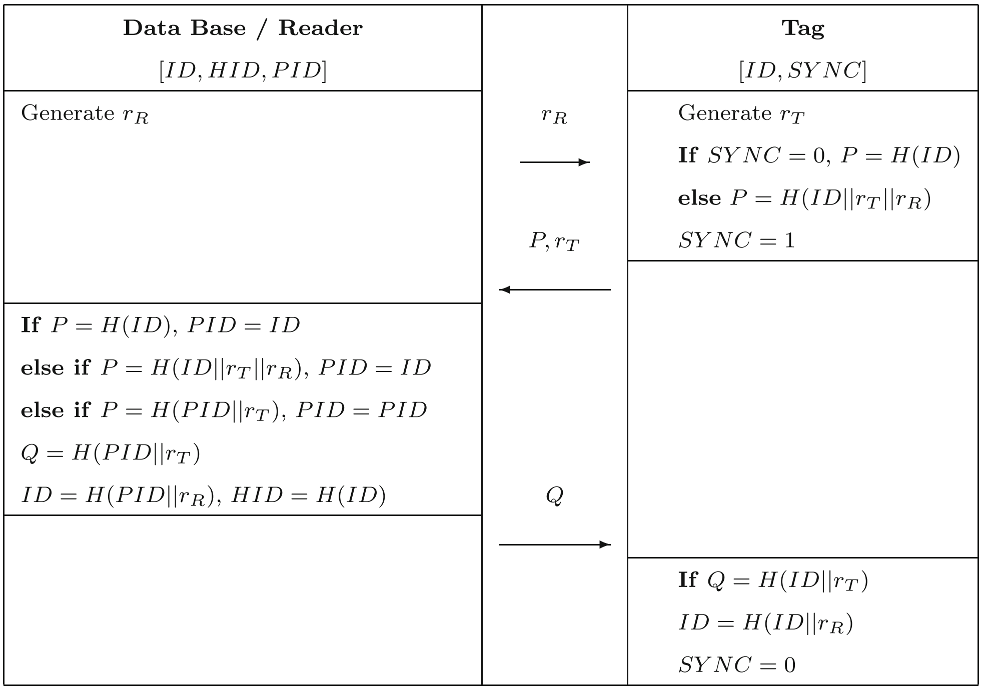
Data Base / Reader
$[ID, HID, PID]$
Tag
$[ID, SYNC]$
Generate r_R
r_R
Generate r_T
If $SYNC = 0$, $P = H(ID)$
else $P = H(ID||r_T||r_R)$
$SYNC = 1$
P, r_T
If $P = H(ID)$, $PID = ID$
else if $P = H(ID||r_T||r_R)$, $PID = ID$
else if $P = H(PID||r_T)$, $PID = PID$
$Q = H(PID||r_T)$
$ID = H(PID||r_R)$, $HID = H(ID)$
Q
If $Q = H(ID||r_T)$
$ID = H(ID||r_R)$
$SYNC = 0$

Fig. 2 The Ha's protocol

- $\mathcal{R}$ challenges $\mathcal{T}$ with a random nonce r_R.
- $\mathcal{T}$ chooses a random nonce r_T and computes P differently according to the state of $SYNC$. If $SYNC = 0$, then computes $P = H(ID)$, otherwise $P = H(ID||r_T||r_R)$, and then sets $SYNC = 1$. $\mathcal{T}$ responds with $\{P, r_T\}$.
- $\mathcal{R}$ delivers the messages from $\mathcal{T}$ to $\mathcal{DB}$ with r_R.
- $\mathcal{DB}$ firstly searches P with the HID values saved in the database. If the values match, $\mathcal{DB}$ regards the ID as the identity of $\mathcal{T}$. This is a general case when the previous session is closed normally. If $\mathcal{DB}$ cannot find any match in the first searching case, then it goes through a linear search for a match $P = H(ID||r_T||r_R)$. If $\mathcal{DB}$ finds a match in any of the two searching cases, then it sets $ID = PID$, otherwise it then computes $H(PID||r_T||r_R)$ and compares it with P. If $\mathcal{DB}$ finds a match in any of the three searching cases, then it calculates $Q = H(PID||r_T)$ and transmits it to the tag through $\mathcal{R}$. Next, it computes $ID = H(PID||r_R)$ and updates $HID = H(ID)$.
- To verify the correctness of Q received from the $\mathcal{DB}$, $\mathcal{T}$ checks $Q = H(ID||r_T)$. If yes, it updates its ID as $ID = H(ID||r_R)$ and sets $SYNC = 0$. □

Proposition 2 *The Ha's protocol suffers the proposed attack and cannot achieve untraceability.*

Proof To prove the statement we have to show that the Ha's protocol is an UAP and a stateful–adversary $\mathcal{A}^{Stat}$ exists for this system. According to the protocol description the server authenticates a registered tag in $O(1)$ if $SYNC = 0$; we call this state as S_0. On the other hand, the server performs a linear search requiring $O(N)$ in case of $SYNC = 1$ and this state is denoted as S_1. Notice that the state of the tag is determined by the value of $SYNC$ variable and two unbalanced states as S_0 and S_1 exist. Thus, the protocol is an UAP. Also, an adversary has ability to set $SYNC$ value. In other words, he can put any tag to state S_0 or S_1 as follows: By using the oracle $O^{\text{EXECUTE}}(t_i)$ for a selected tag with pseudonym t_i, $\mathcal{A}$ puts the tag state into S_0, because at the end of a successful authentication the tag sets $SYNC = 0$. On the other hand if the adversary transmits some random nonce r_R to for a selected tag with pseudonym t_i and breaks the protocol, the tag will be in state S_1. The reason is obvious the protocol is not completed and $SYNC = 1$ value is kept by the tag. □

By considering these facts one can conclude that the adversary is able to put any selected tag into one of the unbalanced states. Hence, the adversary is a stateful–adversary and can call the oracle $O^{\text{SETSTATE}}(S, t)$. From Theorem 1, it is apparent that the Ha's protocol cannot provide untraceability.

The CW protocol and analysis

The protocol proposed by Chang and Wu (2009) relies on a hybrid scheme from the randomized hash lock scheme of Weis et al. (2003) and Dimitriou's mutual authentication scheme (Dimitriou 2005). Initially, a random secret, $A_{i,t}$ shared both in a particular tag and the server is generated, where $A_{i,t}$ denotes the secret of $\mathcal{T}_i$ at instance t. Let

$\alpha_1, \ldots, \alpha_m$ be the enumerate of all possible random strings α of length $\log m$. To be efficient to find the ID of a tag, the indexes $H(\alpha_k||A_{i,t})$ is stored $\forall k \in \{1, \ldots, m\}$.

According to the protocol description, $\mathcal{R}$ is considered as a single entity consists of the back-end server and the reader. The authentication process of the CW protocol goes as follows:

- $\mathcal{R}$ sends a long random string r_R to a tag $\mathcal{T}_i$.
- If the counter C_i is less than some threshold value δ, $\mathcal{T}_i$ generates a short random string α and a long random string r_T. Then it computes two hash values $M_1 = H(\alpha||A_{i,t})$ and $M_2 = H(A_{i,t}||r_T||r_R)$. Next, the tag transmits $\langle M_1, M_2, r_T, \alpha\rangle$ to the reader.
- The $\mathcal{R}$ uses M_1 to quickly search for the index in the database and find out ID of the tag in constant time. If a match is found in the list, then the reader checks $M_2 = H(A_{i,t}||r_T||r_R)$. If the equation is verified, $\mathcal{R}$ replies to $\mathcal{T}_i$ with $M_3 = H(A_{i,t}||r_T)$ and updates $A_{i,t}$ to $A_{i,t+1}$.
- Upon reception of M_3, the tag checks whether it is a true value by computing $H(A_{i,t}||r_T)$. If yes, then it allows the reader to use its all functionality, sets $C_i = 0$ and updates $A_{i,t}$ to $A_{i,t+1}$. If not, however, the tag rejects and records this illegitimate query by incrementing the counter C_i.
- On the other hand, in case of $C_i > \delta$, the tag starts to run the fully randomized hash lock protocol (Weis et al. 2003) with readers. This mode of moderation requires a linear search with $O(N)$ work.

Proposition 3 *The CW protocol is vulnerable the proposed attack and cannot ensure untraceability.*

Proof If we show that the CW protocol is an UAP and a stateful–adversary $\mathcal{A}^{Stat}$ exists for this scheme, then we prove the statement. The protocol is run depending on two different modes: If $C_i \leq \delta$ at the tag side, the server authenticates tag in sub-linear time via searching the value in an indexed list. However, if $C_i > \delta$, then the tag executes a different protocol process which results in a linear search at the server side in tag authentication process. Note that the state of the tag is depends on the counter value C_i, so two unbalanced states as S_0 and S_1 exist for the cases whether C_i is less than or greater than δ respectively. Thus, the protocol is an UAP. Also, an adversary has ability to put a tag into one of two states: By requesting the oracle $O^{\text{EXECUTE}}(t_i)$ for a selected tag with pseudonym t_i, $\mathcal{A}$ puts the tag state into S_0, because at the end of a successful authentication the tag sets $C_i = 0$. On the side if the adversary queries a tag more than δ times, the tag will be in state S_1 due to C_i exceeding δ. Therefore, the adversary is a stateful–adversary and can request the oracle $O^{\text{SETSTATE}}(S, t)$. Considering Theorem 1, we can say that the CW protocol cannot provide untraceability. □

Conclusions

In this study, we proved that any RFID authentication protocol involving unbalanced states are traceable to a stateful-adversary. We apply our result to examine the privacy of some existing protocols and showed that they fail to fulfill untraceability property. We believe that this highlighted privacy issue facilitates development of stronger schemes and will be taken into account as a design criteria for RFID authentication protocols.

References

Alomair, B., & Poovendran, R. (2010). Privacy versus scalability in radio frequency identification systems. *Computer Communication, Elsevier, 33*(18), 2155–2163.

Avoine, G. (2005). *Adversarial model for radio frequency identification*. Cryptology ePrint Archive, Report 2005/049.

Avoine, G., & Oechslin, P. (2005). RFID traceability: A multilayer problem. In *Financial cryptography—FC'05. Lecture notes in computer science* (Vol. 3570, pp. 125–140). Springer.

Avoine, G., Coisel, I., & Martin, T. (2010). Time measurement threatens privacy-friendly RFID authentication protocols. In S. O. Yalcin (Ed.), *Workshop on RFID Security—RFIDSec'10. Lecture notes in computer science* (Vol. 6370, pp. 138–157). Springer.

Burmester, M., Le, T. v., & Medeiros, B. d. (2006). Provably secure ubiquitous systems: Universally composable RFID authentication protocols. In *Conference on security and privacy for emerging areas in communication networks—secureComm 2006* (pp. 1–10). IEEE, IEEE Computer Society, Baltimore, Maryland, USA.

Burmester, M., de Medeiros, B., & Motta, R. (2008). Anonymous RFID authentication supporting constant-cost key-lookup against active adversaries. *International Journal of Applied Cryptography, 1*(2), 79–90.

Chang, J. C., Wu, H. L. (2009). A hybrid RFID protocol against tracking attacks. In *Fifth international conference on intelligent information hiding and multimedia signal processing*. (pp. 865–868). IEEE, IEEE Computer Society, Los Alamitos, CA, USA.

Cheon, J. H., Hong, J., & Tsudik, G. (2009). *Reducing RFID reader load with the meet-in-the-middle strategy*. Cryptology ePrint Archive, Report 2009/092.

Chien, H. Y., & Chen, C. H. (2007). Mutual authentication protocol for RFID conforming to EPC class 1 generation 2 standards. *Computer Standars & Interfaces, Elsevier Science Publishers, 29*(2), 254–259.

Dimitriou, T. (2005). A lightweight RFID protocol to protect against traceability and cloning attacks. In *Conference on security and privacy for emerging areas in communication networks—secureComm*. IEEE, Athens, Greece.

Erguler, I., & Anarim, E. (2010). Scalability and security conflict for RFID authentication protocols. *Wireless Personal Communications* doi:10.1007/s11277-010-0188-0.

Erguler, I., Akgun, M., & Anarim, E. (2009). Cryptanalysis of a lightweight RFID authentication protocol—LRMAP. In: *Western European workshop on research in cryptology—WEWoRC 2009*. Graz, Austria.

Erguler, I., Anarim, E., & Saldamli, G. (2011). A salient missing link in rfid security protocols. *EURASIP Journal of Wireless Communications and Networking*, article id 541283, 2011.

Ha, J., Ha, J., Moon, S., & Boyd, C. (2007a). LRMAP: Lightweight and resynchronous mutual authentication protocol for RFID system.

In *International conference on ubiquitous convergence technology—ICUCT 2006. Lecture notes in computer science* (Vol. 4412, pp. 80–89). Springer.

Ha, J., Moon, S., Nieto, J. M. G., & Boyd, C. (2007b). Low-cost and strong-security rfid authentication protocol. In *Proceedings of the 2007 conference on emerging direction in embedded and ubiquitous computing—EUC'07. Lecture notes in computer science*. (Vol. 4809, pp. 795–807). Springer.

Henrici, D., & Müller, P. (2004). Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers. In: R. Sandhu & R. Thomas (Eds.), *International workshop on pervasive computing and communication security—PerSec 2004* (pp. 149–153). IEEE, IEEE Computer Society, Orlando, Florida, USA.

Juels, A., & Weis, S. (2007). Defining strong privacy for RFID. In *International conference on pervasive computing and communications—PerCom 2007* (pp. 342–347). IEEE, IEEE Computer Society, New York City, New York, USA.

Karthikeyan, S., & Nesterenko, M. (2005). RFID security without extensive cryptography. In *Workshop on security of ad hoc and sensor networks—SASN'05* (pp. 63–67). ACM, ACM Press, Alexandria, Virginia, USA.

Ling, A. P., & Masao, M. (2011). Selection of model in developing information security criteria for smart grid security system. *Journal of Convergence, 2*(1), 39–46.

Molnar, D., & Wagner, D. (2004). Privacy and security in library RFID: Issues, practices, and architectures. In B. Pfitzmann & P. Liu (Eds.), *Conference on computer and communications security—ACM CCS* (pp. 210–219). ACM, ACM Press, Washington, DC, USA.

Nguyen Duc, D., Park, J., Lee, H., & Kim, K. (2006). Enhancing security of EPCglobal Gen-2 RFID tag against traceability and cloning. In *Symposium on cryptography and information security*. Hiroshima, Japan.

Ohkubo, M., Suzuki, K., & Kinoshita, S. (2003). Cryptographic approach to "privacy-friendly" tags. In *RFID privacy workshop*. MIT, Massachusetts, USA.

Ouafi, K., & Phan, R. C. W. (2008). Privacy of recent RFID authentication protocols. In L. Chen, Y. Mu & W. Susilo (Eds.), *4th international conference on information security practice and experience—ISPEC 2008. Lecture notes in computer science* (Vol. 4991, pp. 263–277). Springer.

Rhee, K., Kwak, J., Kim, S., & Won, D. (2005). Challenge-response based RFID authentication protocol for distributed database environment. In D. Hutter & M. Ullmann (Eds.), *International conference on security in pervasive computing—SPC 2005. Lecture notes in computer science* (Vol. 3450, pp. 70–84). Springer-Verlag, Boppard, Germany.

Sarkar, P., & Saha, A. (2011). Security enhanced communication in wireless sensor networks using reed-muller codes and partially balanced incomplete block designs. *Journal of Convergence, 2*(1), 23–30.

Shaoying, C., Li, Y., Li, T., & Deng, R. (2009). Attacks and improvements to an RFID mutual authentication protocol and its extensions. In D. A. Basin, S. Capkun & W. Lee (Eds.), *Proceedings of the 2nd ACM conference on wireless network security—WiSec'09*. (pp. 51–58). ACM, ACM Press, Zurich, Switzerland.

Song, B., & Mitchell, C. J. (2008). RFID authentication protocol for low-cost tags. In V. D. Gligor, J. P. Hubaux & R. Poovendran (Eds.), *Proceedings of the 1st ACM conference on wireless network security—WiSec'08*. (pp. 140–147). ACM, ACM Press, Alexandria, Virginia, USA.

Song, B., & Mitchell, C. J. (2009). Scalable RFID authentication protocol. In *3rd International conference on network and system security—NSS 2009*. (pp. 216–224). IEEE, IEEE Computer Society, Gold Coast, Australia.

Song, B., & Mitchell, C. J. (2010). Scalable RFID security protocols supporting tag ownership transfer. *Computer Communication, Elsevier* doi:10.1016/j.comcom.2010.02.027.

Tsudik, G. (2007). *A family of dunces: Trivial RFID identification and authentication protocols*. Cryptology ePrint Archive, Report 2006/015.

Vaudenay, S. (2007) On privacy models for RFID. In K. Kurosawa (Ed.) *Advances in cryptology–f́basiacrypt 2007. Lecture notes in computer science* (Vol. 4833, pp. 68–87). Springer.

Wang, S. J., Tsai, Y. R., Shen, C. C., & Chen, P. Y. (2010). Hierarchical key derivation scheme for group-oriented communication systems. *International Journal of Information Technology, Communications and Convergence, 1*(1), 66–76.

Weis, S., Sarma, S., Rivest, R., & Engels, D. (2003). Security and privacy aspects of low-cost radio frequency identification systems. In D. Hutter, G. Müller, W. Stephan & M. Ullmann (Eds.), *International conference on security in pervasive computing—SPC 2003. Lecture notes in computer science* (Vol. 2802, pp. 454–469). Springer.

Xie, B., Kumar, A., Zhao, D., Reddy, R., & He, B. (2010). On secure communication in integrated heterogeneous wireless networks. *International Journal of Information Technology, Communications and Convergence, 1*(1), 4–43.