Privacy-friendly Authentication in RFID Systems: On Sub-linear Protocols based on Symmetric-key Cryptography

Gildas Avoine, Muhammed Ali Bingöl, Xavier Carpent, Siddika Berna Ors Yalcin

Abstract—The recent advent of ubiquitous technologies has raised an important concern for citizens: the need to protect their privacy. So far, this wish was not heard of industrials, but national and international regulation authorities, as the European Commission recently published some guidelines to enforce customers' privacy in RFID systems: "Privacy by design" is the way to be followed as stated in EC Recommendation of 12.5.2009. Research on privacy is an active domain but there is still a wide gap between theory and everyday life's applications. Filling this gap will require academia to design protocols and algorithms that fit the real life constraints. In this paper, we provide a comprehensive analysis of privacy-friendly authentication protocols devoted to RFID that: (1) are based on well-established symmetric-key cryptographic building blocks; (2) require a reader complexity lower than $O(N)$ where N is the number of provers in the system. These two properties are *sine qua non* conditions for deploying privacy-friendly authentication protocols in large-scale applications, e.g., access control in mass transportation. We describe existing protocols fulfilling these requirements and point out their drawbacks and weaknesses. We especially introduce attacks on CHT, CTI, YA-TRAP*, and the variant of OSK/AO with mutual authentication. We also raise that some protocols, such as O-RAP, O-FRAP and OSK/BF are not resistant to timing attacks. Finally, we select some candidates that are, according to our criteria, the most appropriate ones for practical uses.

Index Terms—RFID, Authentication, Privacy, Attacks, Complexity

1 Introduction

Radio Frequency Identification (RFID) is a pervasive technology deployed in many applications to identify or authenticate objects and subjects with neither physical nor visual contact. An RFID system usually consists of tags (i.e., a microcircuit with an antenna), carried by the object or subject, some readers that remotely query the tags, and a back-end system.

A common idea is that an RFID tag is just a transponder that backscatters a unique identifier, used for supply chains, libraries, and pet identification. A tag can actually do much more than simply backscattering an identifier, and it is even tricky to define the limits between RFID and the other evolved pervasive technologies. We describe precisely in Sect. 2 the capabilities we confer to tags in this paper.

The fact that no contact is needed to read an RFID tag allows to use it where traditional smartcards are not invited: pet identification, electronic passports, but also access control for ski lifts, ... RFID also brings advantages in access control applications by speeding up the flow of customers, typically in mass transportation. Such a kind of application requires authentication protocols that scale well when there is a large number of tags registered to the system.

While RFID has existed for several decades, it is its recent wide-spread that made privacy a major concern for everyone. Authorities are aware of the privacy issues and react accordingly. For example, in its recommendation SEC(2009) 585/586 about RFID, the European Commission states: "Because of its potential to be both ubiquitous and practically invisible, particular attention to privacy and data protection issues is required in the deployment of RFID. Consequently, privacy and information security features should be built into RFID applications before their widespread use (principle of security and privacy-by-design)" [64]. Similar recommendations also arose in North America [23], [43], [65]. Fulfilling these recommendations may be partially done by designing authentication or identification protocols that ensure privacy against an external adversary. Among privacy, one may distinguish information leakage where the tag or the back-end reveals some personal information, from illicit tracking that consists in tracking a tag and so its holder.

The large body of literature about RFID Security and Privacy [3] demonstrates that designing a privacy-friendly protocol is still a challenging task and finding the appropriate one is quite awful for industrials. Indeed, although many protocols have been proposed over the years, none can be deemed as

- *Gildas Avoine and Xavier Carpent (corresponding author) are with the Information Security Group, ICTEAM institute, Université catholique de Louvain, B-1348, Louvain-la-Neuve, Belgium. E-mail: xavier.carpent@uclouvain.be*
- *Muhammed Ali Bingöl is with TÜBİTAK BİLGEM UEKAE, Gebze, Kocaeli, Turkey and Istanbul Technical University, Faculty of Electrical Electronics Engineering, Maslak, Istanbul.*
- *Siddika Berna Ors Yalcin is with Istanbul Technical University, Faculty of Electrical Electronics Engineering, Maslak, Istanbul.*

ideal. In this paper, we examine most of the proposals in the field, categorize them according to common features, analyze them, compare their properties and discuss about which can be considered as the best ones to date. We provide many new attacks on several of these protocols, as well as some patches.

We emphasize that we do not consider in our work low-level criteria such as gate count or power consumption of tags because, although important, these depend on the implementation of the building blocks. Instead, we focus on the protocols, their efficiency, and the security and privacy level they achieve.

In order to analyze the privacy of the protocols we consider, several models are available in the literature [4], [22], [42], [69]. We decided to use Juels and Weis' model [42], which is based on Avoine's seminal work [4]. The model of Juels and Weis defines privacy with the following game (here simplified for the sake of simplicity):

1) An adversary interacts with the system
2) She chooses two tags $\mathcal{T}_0$ and $\mathcal{T}_1$
3) She interacts with the system except $\mathcal{T}_b$ ($b \in \{0,1\}$ chosen randomly and unknown to her)
4) She must guess whether $b = 0$ or 1

The system is deemed private if there exists no adversary capable of winning with a probability non-negligibly higher than $1/2$. Although Juels and Weis' model is less powerful than Vaudenay's model [69], it is more intuitive and provides an adversary granularity more suited to our analysis than the one provided by [69]. Beyond the concept of *privacy* as defined in [42], we address in this paper the *forward privacy*, as described in [42] as well, and intuitively introduced in [54]. To complete our analyses, we also consider the *timing attacks* against the readers, as introduced in [7].

We give in Sect. 2 the criteria we used to thoroughly select the protocols we analyze in this paper. We then categorize the selected protocols in three sections: protocols with shared secrets (Sect. 3), protocols using hash-chains (Sect. 4), and counter-based protocols (Sect. 5). We compare the most valuable protocols in Sect. 6 and provide a summary of their properties and performances in Tab. 2. Our analysis finally yields in Sect. 7 the best protocols in terms of security, privacy, forward-privacy, desynchronization, reader and tag complexity, and memory.

2 Protocol Selection Criteria

In this section, we list some characteristics that we consider relevant for the protocols to have in the problem at hand. We discard in the rest of our analysis all the protocols that do not meet these criteria. We emphasize that these characteristics do not form a partition, but should cover all existing solutions in RFID authentication, up to our knowledge.

In the following, we consider that the communication between tags and readers is insecure, meaning that it can be easily eavesdropped on, interrupted or modified on the fly by an external entity. However, the communication between readers and the database is secure, and we will in general refer to these two entities as a single one, since it makes no difference for an attacker. This is compliant with the model of Juels and Weis. We consider RFID tags as not tamper resistant. Therefore, solutions in which each tag has the same key, for instance, are discarded.

2.1 Time Complexity of Identification

The ISO-9798 defines challenge-response authentication protocols, which are commonly used in RFID. These are used in the MIFARE Classic for instance. Other standards are also in application, such as the ISO-11770, used for example in the Basic Access Control of e-passports.

However, as explained in [48], when the tag does not directly reveal its identifier, this solution takes $O(N)$ cryptographic operations (where N is the number of tags in the database), which is inefficient in large systems. The problem is important for most reasonably-sized systems, and we therefore restrict our analysis to protocols designed to reduce the complexity of the identification. Protocols such as the HB family ([15], [32], [33], [38], [41], [49]), or protocols such as [13], [21] are therefore not considered below.

2.2 Public-key Cryptography

Public-key cryptography (PKC) seems to be a solution to the identification problem stated above. The randomized Schnorr protocol [16], for instance, uses public-key encryption to provide both strong privacy and constant-time identification.

However, PKC is expensive, being in terms of gates required on the tag, or of time and especially energy necessary to perform the computations on a tag. Although some recent studies point otherwise (see, e.g., [35], [39], [46]), it is generally acknowledged that PKC is not affordable on low-cost tags, as most of the proposals for authentication in RFID use symmetric-key building blocks. We can hope that further research in that area will improve the feasibility of PKC for low-cost RFID, but there will always be a market for symmetric-key solutions. For these reasons, we only consider symmetric-key schemes in the following.

2.3 Privacy

By trying to lower the identification procedure complexity, some solutions also lower the privacy or the security considerably. For instance, one could imagine a very simple scheme where each tag has a limited amount of ephemeral pseudonyms (or "coupons"), using one each time a reader wants to authenticate it. This solution is both private and efficient, but has a limited lifetime and an adversary could perform

denial-of-service attacks very easily. Juels proposes in [40] a similar protocol in which each tag loops through a sequence of secrets to authenticate itself to a reader, again providing efficiency, but limited privacy. Henrici and Müller proposes in [37] that tags communicate to the reader the number of failed authentication attempts since the last legitimate authentication. While this allows the reader to efficiently identify the tags, it also allows an adversary to trace them, as pointed in [4].

In other proposals, such as [27], [29], [45], [60], [61], [62], [58], [72], each tag uses pseudonyms that change after each successful authentication. However, an adversary is able to trace its victim between two of them, which is a serious threat in some applications.

Note that despite the fact that these solutions are not private strictly speaking, there might be scenarios where they can be applied, since *some* privacy is better than none at all. However, for the reasons argued in Sect. 1, we will only consider protocols that have no obvious privacy or availability issues in this study.

2.4 Building Blocks

Finally, there are some other proposals that use non-classical cryptographic building blocks, deemed more lightweight than usual hash functions and ciphers, in order to lower the gate count on tags, and thus their price. An example is the family of so-called *ultra-lightweight* authentication protocols (see e.g. [27], [29], [45], [60], [61], [62], [58], [72]). Although innovative and interesting, this branch is rather recent and to date, all proposals suffer from miscellaneous security and privacy weaknesses.

In addition, a number of works, such as [25], [28], [50], [73], aim to provide secure protocols conforming to EPC Class-1 Gen-2 standards. Unfortunately, these attempts fall short of meeting the desired security objectives because EPC Class-1 Gen-2 supports only simple building blocks such as a 16-bit Pseudo-Random Number Generator and a 16-bit Cyclic Redundancy Code. Many analysis papers (see e.g., [34], [63], [57], [59]) show that it seems that enforcing privacy and security under the EPC Class-1 Gen-2 specifications is an almost impossible task due to the "bad" properties of the building blocks used.

For these reasons, we only consider the protocols that use the classical cryptographic primitives, and we focus our analysis on the protocols, not on the underlying building blocks.

2.5 Remaining Protocols

In the following, we consider all protocols of which we are aware that match the criteria developed above.

With time, some of these protocols were renamed. To avoid confusion, we present in Table 1 the matches between the protocols proposed with different names. The papers and publication years are given in the first row. Each other row represents one protocol, showing names given in each paper. In what follows we will use the most recently appeared names (in bold).

Table 1
Matching of the names of some protocols

[67] 2006	[18] 2006	[66] 2007	[19] 2009
YA-TRAP	-	YA-TRIP	**RIP**
-	-	YA-TRAP	**RIP+**
-	YA-TRAP+	-	**RAP**
-	O-TRAP	-	**O-RAP**
-	-	-	**O-RAKE**
-	-	**YA-TRAP***	-
-	-	**YA-TRAP*& fwd**	-

3 Protocols with Shared Secrets

Some recent protocols have the common feature that several tags in the system share their secrets (at least partially). They manage to lower the online complexity of the reader by storing tag secrets in a particular structure (a tree, a grid, etc.). While these protocols provide that very desirable property and bring new and interesting ideas, they all have traceability issues.

In this section, we describe Molnar and Wagner's tree-based protocol [48], Alomair, Clark, Cuellar, and Poovendran's protocol [2], Avoine, Buttyán, Holczer, and Vajda's group-based protocol [6], and Cheon, Hong, and Tsudik's meet-in-the-middle protocol [26]. We also discuss some attacks on these protocols, especially new attacks we suggest against [26] and [2].

3.1 Tree-based and Group-based Protocols

As stated previously, privacy-friendly challenge-response protocols do not scale well: the reader must check $O(N)$ keys to authenticate a tag, where N is the total number of tags in the system.

Molnar and Wagner propose in [48] an approach that reduces the complexity from $O(N)$ to $O(\log N)$. The fundamental idea is to manage the tags' keys in a tree structure instead of using a flat structure. More precisely, the tags are assigned to the leaves of a balanced tree with branching factor b at each level of the tree. Each edge of the tree carries a random key. Each tag stores the keys along the path from the root to the leaf corresponding to the given tag, while the reader stores the whole tree. During the authentication process, the reader performs one challenge-response per tree level in order to identify the sub-tree the tag belongs to. Each challenge-response requires from the reader an exhaustive search in a set containing b keys only. The overall reader's complexity of the authentication is $b \log_b N$ in the worst case.

The significant complexity improvement due to Molnar and Wagner's technique (MW) has however an unacceptable drawback: the level of privacy provided by the scheme is quickly decreasing when an adversary tampers with tags. Giving the adversary the ability to tamper with some tags makes sense because

MW is useless without this assumption: in such a case, the same key can be stored in all the tags and the complexity problem no longer occurs. On the other side, giving to the adversary the ability to tamper with tags significantly degrades the privacy in MW.

Avoine, Dysli, and Oechslin raise this attack in [8] and evaluate the trade-off between complexity and privacy according to the branching factor. Buttyán, Holczer, and Vajda in [20] also identified weaknesses of MW and introduce an improvement with variable branching factors. Nohl and Evans in [53] provided another approach to analyze MW. Later on, Halevi, Saxena, and Halevi [33] present a lightweight privacy-friendly authentication protocol that combines Hopper and Blum's HB protocol [38] and the tree-based key infrastructure suggested by Molnar and Wagner [48]. However, [38] inherits from the weaknesses of MW as demonstrated by Avoine, Martin, and Martin in [9]. Finally, Beye and Veugen further analyze the improvement of Buttyán *et al.* in [12].

One may also cite some other attempts to design tree-based protocols, e.g., [71] or the saga [1], [31], [47], [70]. However, we have seen that tree-based secret sharing is definitely not suited when the adversary is capable of tampering with tags, and the tree structure is even not the best solution in that case. Indeed, Avoine, Buttyán, Holczer, and Vajda demonstrate in [6] that a simpler structure than the tree, namely when tags are grouped and each group share a same key, achieves a higher level of privacy and a better efficiency. Finding a better structure, that does not avoid the traceability problem but that mitigates it is still an open problem.

3.2 Cheon, Hong, and Tsudik's Protocol

3.2.1 Description.

The protocol proposed by Cheon, Hong, and Tsudik in [26] is an innovative proposal to reduce the reader complexity. It uses a *meet-in-the-middle* strategy, similar to the one used in several famous attacks on double-encryption schemes [30]. The idea is the following. During the initialization, the system chooses two sets of keys $\mathcal{K}_1$ and $\mathcal{K}_2$ such that $|\mathcal{K}_1| = |\mathcal{K}_2| = n$, where $N = n^2$ is the number of tags in the system, and $\mathcal{K}_1 \cap \mathcal{K}_2 = \emptyset$. It then initializes each tag $\mathcal{T}_{i,j}$ with a unique pair of keys $\langle K_1^i, K_2^j \rangle$, where $K_1^i \in \mathcal{K}_1$ and $K_2^j \in \mathcal{K}_2$, yielding an $n \times n$ grid in which each cell represents a tag, as depicted in Fig. 1.

The identification procedure, represented in Fig. 2, is as follows. The reader $\mathcal{R}$ first picks a nonce r and sends it to a tag $\mathcal{T}_{i,j}$ entering its field. The latter then picks another nonce r', and computes $C = PRF_{K_1^i}(r, r') \oplus PRF_{K_2^j}(r, r')$, where PRF is a pseudo-random function. The tag $\mathcal{T}_{i,j}$ then sends the pair $\langle C, r' \rangle$ to $\mathcal{R}$. In order to identify the tag, $\mathcal{R}$ computes $PRF_{K_1^x}(r, r')$ for $x \in [1, n]$, and then computes $C \oplus PRF_{K_2^y}(r, r')$ for $y \in [1, n]$, and tries to find a match between two values. This search requires $2n = 2\sqrt{N}$ PRF evaluations at worst, rather than N for a standard linear search[1]. An adversary eavesdropping r, r', and C however would have to search the entire key space, since she does not know the key sets $\mathcal{K}_1$ and $\mathcal{K}_2$.

	K_1^1	K_1^2	K_1^3	...	K_1^i	...	K_1^n
K_2^1	$\mathcal{T}_{1,1}$	$\mathcal{T}_{2,1}$	$\mathcal{T}_{3,1}$	...	$\mathcal{T}_{i,1}$	...	$\mathcal{T}_{n,1}$
K_2^2	$\mathcal{T}_{1,2}$	$\mathcal{T}_{2,2}$	$\mathcal{T}_{3,2}$	...	$\mathcal{T}_{i,2}$	...	$\mathcal{T}_{n,2}$
⋮	⋮	⋮	⋮	⋱	⋮	⋱	⋮
K_2^j	$\mathcal{T}_{1,j}$	$\mathcal{T}_{2,j}$	$\mathcal{T}_{3,j}$	...	$\mathcal{T}_{i,j}$	...	$\mathcal{T}_{n,j}$
⋮	⋮	⋮	⋮	⋱	⋮	⋱	⋮
K_2^n	$\mathcal{T}_{1,n}$	$\mathcal{T}_{2,n}$	$\mathcal{T}_{3,n}$	...	$\mathcal{T}_{i,n}$	...	$\mathcal{T}_{n,n}$

Figure 1. Tags' secrets organized in a grid in CHT.

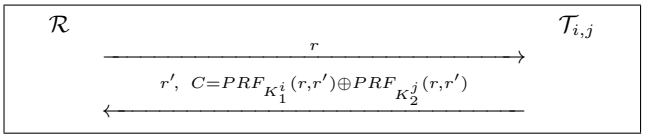


Figure 2. Cheon-Hong-Tsudik plain protocol.

The protocol presents an efficient search procedure, but is not synchronized (i.e., the tag has no *state* that changes over time). This implies that it does not provide any forward-privacy, because an adversary having compromised a tag gets its two keys, and can thus recompute messages previously produced by the tag, in this way "tracing" the tag in the past.

Moreover, the authors themselves identify an important issue. Indeed, when a tag is compromised, its two sub-keys are disclosed, but this does not leak any information on other tags' keys as the combination of subkeys is unique. However, when the adversary compromises several tags, she gains knowledge of key-pairs of legitimate tags. For instance, If the adversary compromises the tags $\mathcal{T}_{a,b}$ and $\mathcal{T}_{c,d}$, she also discovers the keys of the tags $\mathcal{T}_{a,d}$ and $\mathcal{T}_{b,c}$. These will respectively be referred as *directly compromised tags* and *indirectly compromised tags* hereafter (a *compromised tag* refers to either situation). We also name *partially compromised* the tags for which we only know one key.

The authors describe an extension to mitigate this problem by introducing proper authentication in the protocol. In this extension, each tag has a third, unique subkey K_3. The key sets $\mathcal{K}_1$ and $\mathcal{K}_2$ have a size of N^α, with $0 \leq \alpha \leq \frac{1}{2}$ being a system parameter and $\mathcal{K}_3$ has a size of N, such that $N^{1-2\alpha}$ tags have the same $\langle K_1, K_2 \rangle$ key-pair. The tag further computes $C' = PRF_{K_3}(r, r')$, and sends it to the reader. After the usual search procedure, $\mathcal{R}$ checks the value C' to authenticate the tag. Since K_3 is unique to each tag, the impersonation attack is prevented, but there is still a traceability issue, as detailed in Sect. 3.2.3.

1. Note that in [26], the authors state that the search is $O(\sqrt{N} \log N)$. We consider only the cryptographic operations in the online time, so we suggest $O(\sqrt{N})$ instead.

3.2.2 Impersonation Attack on the Plain Protocol.

After having compromised some tags, an adversary can perform the following impersonation attack. She listens to a legitimate authentication session between $\mathcal{R}$ and $\mathcal{T}_{i,j}$. When $\mathcal{T}_{i,j}$ outputs (r', C), she blocks the message. She can now change C in order to authenticate another tag than $\mathcal{T}_{i,j}$. Because the protocol is stateless, $\tilde{C}$, the modified C, will be accepted (provided it is valid), and the corresponding tag will be identified. Two situations may occur for an adversary:

1) She wants to authenticate a tag that is compromised instead of $\mathcal{T}_{i,j}$.
2) She wants to authenticate a tag that is partially compromised.

In case 1, the adversary can replace the authenticating tag with another compromised tag, say, $\mathcal{T}_{a,b}$, by simply replacing C by $\tilde{C} = PRF_{K_1^a}(r, r') \oplus PRF_{K_2^b}(r, r')$. This problem was already highlighted in [26].

In case 2, the adversary must at least know one of the keys of $\mathcal{T}_{i,j}$ to succeed (i.e. $\mathcal{T}_{i,j}$ must be partially compromised). Let us suppose that the adversary knows K_1^i but not K_2^j, and that she also knows another key K_1^k. She can then replace C by $\tilde{C} = PRF_{K_1^k}(r, r') \oplus PRF_{K_2^j}(r, r')$ by computing $\tilde{C} = C \oplus PRF_{K_1^i}(r, r') \oplus PRF_{K_1^k}(r, r')$, and by doing so, authenticate $\mathcal{T}_{k,j}$, which is only partially compromised. Of course, she does not know the keys of the victim in advance, so the attack is probabilistic. She can thus iterate on all the tags for which she knows the secrets partially. A side-effect of this is that when $\mathcal{R}$ accepts the authentication, the adversary gets $PRF_{K_2^j}(r, r')$, which can lead to a traceability attack.

In [26], the authors state that, when compromising t tags, the number of indirectly compromised tags is $t^2 - t$. This is actually rather optimistic (from an attacker viewpoint) and only accurate when t is small. We provide a more precise result in Lemma 1, which proof is given in the Appendix **??**.

Lemma 1: Let T denote the number of *directly compromised* tags and S the total number of *compromised* tags, that is the ones for which we know both keys. Then, the expected number of compromised tags given that we compromised t tags is:

$$\mathbb{E}\left[S|T=t\right] = N\left[1 - \frac{2\binom{N-n}{t} - \binom{N-2n+1}{t}}{\binom{N}{t}}\right],$$

where $n = \sqrt{N}$. A similar result applies for the authentication extension, and S here denotes the number of compromised *cells*:

$$\mathbb{E}\left[S|T=t\right] = n^2\left[1 - \frac{2\binom{N-N/n}{t} - \binom{N-2N/n+N/n^2}{t}}{\binom{N}{t}}\right],$$

with $n = N^\alpha$.

This result allows to quantify the probability of success of our attacks and confirms their feasibility, as we will see below.

3.2.3 Traceability Attack on Authentication Extension.

Recall that in the authentication extension, the grid can now be seen as $N^\alpha \times N^\alpha$ "cells" of $N^{1-2\alpha}$ tags secrets. No two tags share the K_3 key, but each $\langle K_1, K_2\rangle$ is shared among $N^{1-2\alpha}$ tags. As the authors mentioned, this leads to a traceability issue because if an attacker knows a $\langle K_1^i, K_2^j\rangle$ pair, she can track $\mathcal{T}_{i,j}$ with probability $1/N^{1-2\alpha}$ by using the fact that there are $N^{1-2\alpha}$ tags with the same pair.

In this section, we point out a more dangerous issue. Let us assume that the adversary has obtained the keys related to s cells. For the sake of simplicity, we assume that the compromised tags are put back into circulation. Since this number is supposedly small compared to N, the number of tags in the system, this is a reasonable assumption.

Let X denote the set of tags which secrets belong to one of the s cells known by the adversary. In a Juels and Weis game [42], when two tags $\mathcal{T}_0$ and $\mathcal{T}_1$ are presented to her, the adversary is asked to answer which of these tags is her target. Several cases occur:

- $E_1 = \mathcal{T}_0 \in X \wedge \mathcal{T}_1 \notin X$
- $E_2 = \mathcal{T}_0 \notin X \wedge \mathcal{T}_1 \in X$
- $E_3 = \mathcal{T}_0 \in X \wedge \mathcal{T}_1 \in X \wedge \langle K_1, K_2\rangle_{\mathcal{T}_0} \neq \langle K_1, K_2\rangle_{\mathcal{T}_1}$
- $E_4 = \mathcal{T}_0 \in X \wedge \mathcal{T}_1 \in X \wedge \langle K_1, K_2\rangle_{\mathcal{T}_0} = \langle K_1, K_2\rangle_{\mathcal{T}_1}$
- $E_5 = \mathcal{T}_0 \notin X \wedge \mathcal{T}_1 \notin X$

The obvious strategy for an adversary is, after choosing r, to query $\mathcal{T}_0$ and $\mathcal{T}_1$, and compare their answer with what would have answered the tags of which she knows the keys. If there is a match, then she identifies the tag and deduces its keys. In E_1 and E_2, only either of $\mathcal{T}_0$ and $\mathcal{T}_1$ is identified, and the adversary is able to determine correctly whether it is her target or not in all cases. If both tags are identified, the adversary succeeds only when they have a different key-pair (E_3, but not E_4). Finally, if neither is identified, the adversary is unable to tell her target apart in any better way than at random. Therefore, in the first three events, the adversary succeeds in the attack, and in the other two she fails. It is clear that the first two cases are symmetric:

$$\Pr(E_1) = \Pr(E_2) = \frac{NM - M^2}{N^2}, \tag{1}$$

where $M = sN^{1-2\alpha}$, that is the number of tags for which the adversary knows the secrets. Likewise,

$$\Pr(E_3) = \frac{M^2}{N^2}(1 - 1/s). \tag{2}$$

The overall probability that the adversary succeeds after corrupting s cells is thus

$$\begin{aligned}\Pr(E_1 \vee E_2 \vee E_3) &= \Pr(E_1) + \Pr(E_2) + \Pr(E_3)\\ &= 2\frac{M}{N} - \frac{M^2}{N^2}(1 + 1/s),\end{aligned}$$

because these events are mutually exclusive. This probability can become much higher than the one

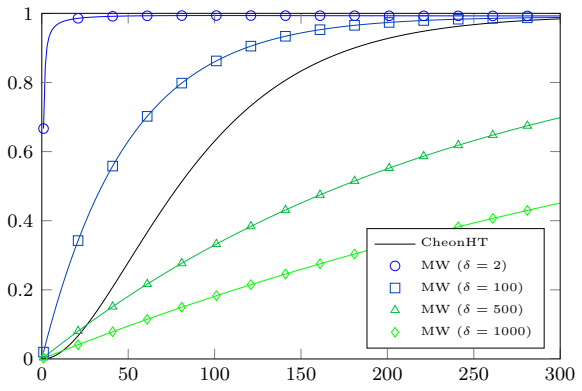


Figure 3. Probability of tracing a tag in CHT and in MW with respect to the number of compromised tags.

presented in [26]. For instance, in a system with $N = 10^6$ tags, configured with $\alpha = \frac{1}{3}$ (as suggested by the authors), an adversary having compromised $t = 300$ tags has roughly $s = 8750$ compromised cells (Lemma 1), and a probability of roughly 0.984.

3.2.4 *Discussion.*

We have introduced two important attacks on CHT. The first one regards the plain protocol and allows an adversary to change the tag being authenticated. The targeted tag need not be completely indirectly compromised, as a probabilistic approach can be carried out. The second attack regards the authentication extension, and allows an adversary to trace a tag.

The second attack is similar to the one [8] against MW. Although quite different technically, MW and CHT have in common the fact that tags share parts of their secrets. This property yields efficient tag identification, but compromising tags becomes far more dangerous. We present in Fig. 3 a comparison of the probability of tracing in CHT and MW protocols (with different values for the branching factor), in a system with $N = 10^6$ tags.

3.3 Alomair, Clark, Cuellar, and Poovendran's Protocol

3.3.1 *Description.*

The protocol introduced by Alomair, Clark, Cuellar, and Poovendran in [2] provides Constant-Time Identification (CTI). We classify this protocol in the shared-secret family in the sense that the system manages a pool of shared secret pseudonyms such that each tag is paired with a pseudonym for a while, and is reassigned to another one each time it is legitimately authenticated. Consequently, different tags may use the same pseudonym, at different times. Using re-usable pseudonyms was first introduced by Juels in [40] where each tag manages its own pool of pseudonyms and uses linear combination of them once all the

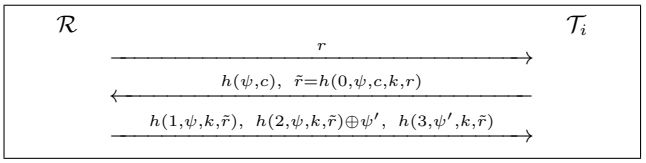


Figure 4. The CTI Protocol.

pseudonyms have been used. However, tags do not exchange their pseudonym in [40], contrarily to [2].

During the set up phase, each of the N_T tags is assigned with a secret key k, a cycling counter c that is incremented modulo C each time the tag is queried (initially $c = 0$), and an initial pseudonym ψ drawn from a pool $\mathcal{E}$ of size $N > N_T$. A sketch of CTI is depicted in Fig. 4 and we refer the reader to [2] for a detailed description.

The key-point of CTI is that each time a tag is legitimately authenticated, it releases its current pseudonym in order to get a new one from the reader randomly drawn from $\mathcal{E}$; it also updates its secret key k with the value $h(k)$ where h is a hash function. CTI provides constant time identification but this property is obtained after pre-calculation of all the NC possible answers from the tags. In that sense, CTI is not far from OSK [54]: the table of pairs (pseudonym, counter) in [2] is in some way similar to the table of pairs (identifier, counter) in [54]. A few differences can nevertheless be raised: (1) a denial of Service (DoS) occurs with OSK after M illegitimate authentications, while CTI is DoS-resistant; (2) OSK with authentication requires to compute between 3 (if there is no attack) and $2m+1$ hash calculations per identification, while CTI requires 4 hash calculations in any case; (3) CTI needs a larger memory than OSK and provides a lower privacy-resistance, as explained below. Note that both of them are resistant to timing attacks as stated in [7]. OSK will be studied in Sect. 4.1.

3.3.2 *Intra-legitimate authentication Attack.*

The main drawback of CTI, already mentioned in [2] is the cycling counter because a tag can be easily tracked between two legitimate authentications if an adversary is able to query it C times. Indeed, recording each of the answers $h(0, \psi, c, k, r)$ $(0 \leq c < C)$, the adversary can definitely track the tag till the next legitimate authentication. This attack is especially meaningful when considering tags that are not frequently used, e.g., passports or tickets used for ephemeral event and kept by the customer as souvenir... Increasing C makes the attack harder, but this also significantly increases the memory consumption (and the reader's workload during the setup). This attack makes CTI not traceability-resistant in the Juels and Weis model [42].

3.3.3 *Inter-legitimate authentication Attack.*

The pseudonyms used in the system are originally secret and can only be revealed in case of tampering

attack. In such a case the current pseudonym of the compromised tag is revealed (and the secret key as well) but the adversary can also obtain additional pseudonyms by impersonating the tag in the system. This attack is mentioned in [2] but we refine its analysis and show that its impact should not be underestimated. First of all, the number of pseudonyms obtained by the adversary after tampering with only one tag is [2] $N\,(1-(1-1/N)^q)$, where q is the number of protocol executions[2]. Let $\mathcal{E}^q \subset \mathcal{E}$ the set of pseudonyms so revealed, the adversary can track a tag (even after legitimate authentications) as follows: in the learning phase as defined in the model of Juels and Weis [42], the adversary queries the targeted tag $\mathcal{T}_{\text{target}}$ once and so obtains a value $h(\psi_{\text{target}}, c_{\text{target}})$. Trying an exhaustive search on all values in $\mathcal{E}^q$ and all counter values, she obtains c_{target} if and only if $\psi_{\text{target}} \in \mathcal{E}^q$, which occurs with probability $|\mathcal{E}^q|/N$. In the challenge phase, given $\mathcal{T}_0$ and $\mathcal{T}_1$, the adversary must decide which one is $\mathcal{T}_{\text{target}}$. To do so, she applies the same technique and so possibly obtains c_0 and c_1. From c_0 and c_1, she could be able to decide which of $\mathcal{T}_0$ and $\mathcal{T}_1$ is $\mathcal{T}_{\text{target}}$. For example, if the adversary knows that her target is rather new while c_i ($i = 0$ or 1) is large, it may be safe to conclude that $\mathcal{T}_{\text{target}}$ is $\mathcal{T}_{1-i}$. To illustrate this attack, consider the following practical parameters: $N = 2N_T$, $N_T = 10^6$, $C = 10^3$, and $q = 10^3$. The probability to track a given tag is therefore 0.1%, assuming that one of the two tags only is rather new.

3.4 Discussion

While protocols using shared secrets all aim mainly to decrease the identification time on the reader, they all have issues when facing adversaries capable of compromising tags. One could argue that a protocol using only one "master key" is the extreme case in that direction: it has constant-time identification, but no privacy/security as soon as one tag is compromised.

All of the proposals we analyzed in this section have important problems, mostly due to the fact that compromising one tag reveals information on other tags too. However, we have no element showing that sharing secrets between tags is a definitely flawed way of reducing identification time. It remains an open question whether it is possible to design such a protocol without any loss of security or privacy.

4 Protocols Based on Hash-Chains

An early family of sub-linear protocols uses hash-chains to update the internal state of the tags. In this section, we describe Ohkubo, Suzuki, and Kinoshita's protocol (OSK) [54] two of its improvements, OSK/AO [8], [10] and OSK/BF [52]. We then describe the O-RAP [19] protocol.

We show a traceability attack on the mutual authentication extension of OSK/AO protocol, and we suggest a solution to overcome this problem. We also show new weaknesses of O-RAP and OSK/BF.

4.1 OSK Protocol

OSK [54] is a well-known synchronized identification protocol[3], and was one of the earliest of its kind. However, beside its traceability issue, and although the protocol is very efficient when all tags are synchronized, the worst-case complexity of the search makes the protocol unsuitable for most practical systems.

The authors later introduced in [55] some ideas to improve the efficiency of the search at the cost of lowering privacy. Since strong privacy is one of the design goals of OSK, we do not consider them.

4.2 OSK/AO Protocol

Avoine and Oechslin propose in [10] to apply Hellman's time-memory trade-offs [36] to the search procedure of OSK, which has two main implications. First, the complexity of the search procedure varies from $O(1)$ to $O(N)$, depending on the amount of memory we are willing to devote to the time-memory trade-off[4]. Moreover, the search is intrinsically randomized, which prevents timing attacks [7].

Avoine, Dysli, and Oechslin also suggest in [8] a variant of OSK that ensures authentication as OSK is originally designed to provide private identification only (i.e., it does not resist to replay attacks). To do so, they suggest using nonces: instead of simply sending a *request* message, the reader sends a nonce r, and the tag answers $G(s_i^k \oplus r)$ along with $G(s_i^k)$.

Finally, Avoine proposes in [5] an extended version of OSK that provides reader authentication to the tag: the reader sends a last message $G(s_i^{k+1} \oplus w)$, where w is a public static value.

However, we point out a traceability issue in this extension: an adversary can eavesdrop a legitimate authentication between $\mathcal{R}$ and $\mathcal{T}_i$, and record the last message (i.e. $G(s_i^{k+1} \oplus w)$); after a while, she sends w as a nonce to a tag, and if the tag answers with the previously recorded value, this tag is almost certainly $\mathcal{T}_i$, and it has not been queried since then.

Preventing this attack can be done easily using a third hash function for the last message. In practice, a single hash function is implemented and an additional input enables to derive it into several functions, for

2. Note that [2] suggests to limit the number of requests to a reader per tag, but bounding q to a value less than 1000 does not seem realistic in most applications as the adversary can avoid being detected, using a slow attack.

3. PFP, introduced by Berbain, Billet, Etrog, and Gilbert in [11] is strongly inspired by OSK. The building blocks in PFP are different than the ones in OSK, and they are used in a different way, but the global scheme is the same, and the security and privacy properties of the two protocols are equivalent. Hence, we will not detail PFP further.

4. The authors mention that, for instance, a complexity of $O(N^{2/3})$ can be reached with a memory of size $O(N^{2/3})$.

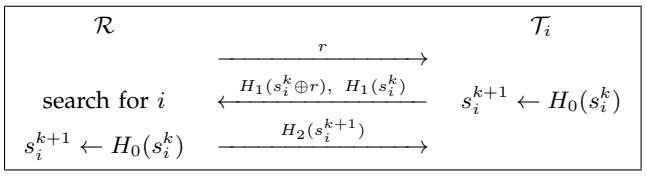


Figure 5. Patched OSK with replay-attack protection and reader authentication.

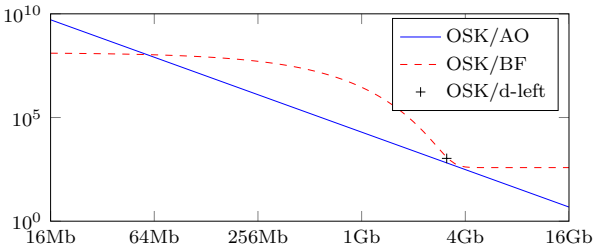


Figure 6. Average number of cryptographic hashes during identification for variants of OSK.

instance, by concatenating 0, 1, or 2 to the value to hash. Fig. 5 shows our modification to the mutual authentication extension of OSK/AO.

4.3 OSK/BF Protocol

4.3.1 Description.

Nohara, Inoue and Yasuura propose in [52] another innovative time-memory trade-off for OSK, which we label OSK/BF in the following. They use Bloom Filters [14], a space-efficient data structure, to store all the hash-chains of each tag. When identifying a tag, the reader first queries all the Bloom Filters for the received σ, and then computes the whole hash-chain of each candidate to confirm the identity of the tag. Once identified, the corresponding Bloom Filter is re-computed for the next hash-chain. On that point OSK/BF contrasts with OSK/AO, in which updates of the database occur less frequently but are more costly.

As presented in [52], OSK/BF is an identification protocol and does not resist impersonation. However, we point out that it can be easily adapted to an authentication scheme using the same construction as the one in [8].

In [51], Nohara and Inoue present an analogous protocol using a similar architecture but a different data structure, d-left Hash Tables [17], an extension of Bloom Filters. The resulting protocol has, according to the authors, a better update efficiency than OSK/BF, but it turns out to be the same. Furthermore, the identification time seems to be very comparable to that of [52], and it has the further disadvantage of being less parameterizable.

4.3.2 Traceability Timing Attacks.

We point out two potential traceability weaknesses of OSK/BF due to timing analysis, not mentioned in [52]. The first one uses the fact that the search is linear in [52], meaning that $\mathcal{T}_1$ will on average be authenticated much faster than $\mathcal{T}_N$, for instance. The reason is that when a tag has a record (and a corresponding Bloom Filter) at the start of the table, the reader has to go through few false positives invalidations before actually confirming the identity of the tag, whereas when it has a record near the end of the table, it might go through several of them. The second attack uses the fact that it is possible to trace a tag being desynchronized more than M times by observing whether the identification time remains constant (it should be constant when the reader refuses identification, but not when the Bloom Filters get updated). Countermeasures might exist against these attacks (simply shuffling the search seems to be a solution to the first one), but in any case, OSK/BF is more fragile regarding timing analysis than OSK/AO, and avoiding them without artificially waiting for $O(N)$ cryptographic operations does not seem to be trivial.

4.3.3 Comparison with OSK/AO.

As in OSK/AO, the time of identification can be lowered by increasing the memory of the reader. In OSK/BF, this is done by tuning the false positive rate of the Bloom Filters. Doing so results in more time needed to compute the hash-chains in order to infirm false positives, increasing identification time, but also in a decrease of the size of Bloom Filters and thus of memory. In OSK/AO, this is done by tuning the size of the Rainbow table, and also determining the amount of intermediate columns stored.

A slight advantage of OSK/BF over OSK/AO is that, despite it also has a probabilistic nature, the successful identification rate is of 100% while being *close to* 100% (fixed by parameters) in OSK/AO. However, the two protocols have the same disadvantage regarding desynchronization, i.e., a tag desynchronized more than M times is lost.

Regarding the trade-off efficiency, OSK/AO seems slightly more efficient than OSK/BF, although comparable. We used numbers from [8], i.e. a system of 2^{20} tags and chains of 2^7 hashes, to provide a comparison between the two protocols, which we depict in Fig. 6. The saturation in OSK/BF after some point comes from the fact that the update part takes $2M$ cryptographic operations, no matter how much memory is dedicated to the trade-off. Note also that we did not take the random hash calculations into account, which, depending on the functions used, could increase the identification time significantly.

4.4 O-RAP Protocol

4.4.1 Description.

O-RAP, which stands for *Optimistic RFID Authentication Protocol*, has been originally introduced in [18]

(its former name was O-TRAP — see Table 1) and a slightly modified version is re-presented in [19]. They call the protocol "optimistic" for the reason that the security overhead is minimal when the system is not under attack. The steps of O-RAP are shown in Fig. 7. The reader contains a hash table indexed by r_{tag} with entries K_i (the static keys of the tags). When starting an authentication, the reader sends a random number r_{sys} to the tag. The tag computes the hash of r_{sys} and r_{tag} with its key K_i and gets r and h output values. Then the tag sends h and r_{tag} values to the reader. The tag also updates r_{tag} with r value. The system searches r_{tag} to find the corresponding K_i in the database, and if found, it checks the correctness of the hash. If r_{tag} is not found, then it exhaustively searches among all the keys. If found, it validates the tag and updates r_{tag} with r value. This allows the reader to re-synchronize the tag automatically.

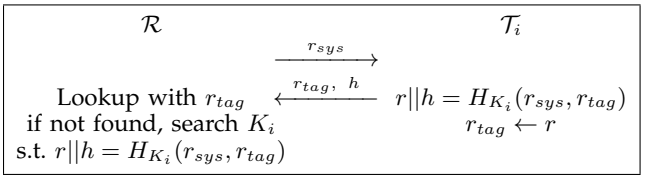


Figure 7. O-RAP Protocol.

4.4.2 Attack by Ouafi and Phan.

In [56], Ouafi and Phan propose a traceability attack on O-RAP based on the desynchronization of a tag. The idea is that an adversary can make enough queries to a tag in order to make it update its secret r_{tag} a lot of times to the point that a legitimate reader is unable to authenticate it anymore.

However, we point out that this attack is erroneous. Indeed, the tag always sends r_{tag} in its answer so the resynchronization is trivial, and because K_i does not change, the authentication is always correct, regardless of how many queries the attacker has performed.

4.4.3 Forward-Privacy Issue and O-FRAP.

Although the authors raise the problem in [18], no particular attention has been drawn on the forward-privacy of O-RAP. An attacker compromising $\mathcal{T}_i$ at some point can recover r_{tag} and K_i. This allows him to trace $\mathcal{T}_i$ in the past, because r_{tag} is sent in the clear and is updated by r. This update can be computed by the adversary, since K_i does not change.

The authors propose in [68] the O-FRAP protocol, adding the forward-privacy to O-TRAP. This comes at the cost of an extra pass in order to authenticate the reader to the tag, as well as a memory overhead for storing previous keys. However, we point out that the protocol is not forward-private strictly speaking. Indeed, suppose than an adversary queries the tag some times without answering to it. Afterwards, she compromises the tag, and if the tag has not been authenticated since, she will be able to trace it in the past. This is the same idea as protocols using pseudonyms for identification, discarded in Sect. 2.3.

Also note that in [19] and [68], the authors propose key exchange extensions to O-RAP and O-FRAP respectively, namely O-RAKE and O-FRAKE. Their goal is to provide features outside of authentication, which is beyond the scope of our paper.

4.4.4 Traceability Timing Attack.

The fact that O-RAP behaves differently according to synchronization makes it work very efficiently in "normal" situations, but allows an adversary to carry out the following timing attack. The adversary first sends a random number to a tag and ignores its answer. The tag will thus be desynchronized with the system, and the next legitimate reader trying to authenticate it will take much more time, because in that case, the search is linear. The adversary can easily notice that by measuring time differences, and can thus trace the tag she desynchronized.

A possible countermeasure is to artificially add time for the search in a normal situation, but this would be equivalent to a protocol with linear complexity.

4.5 Discussion

OSK and O-RAP are two convincing proposals with a simple design and interesting properties.

As pointed by Avoine and Oechslin in [10] and by Nohara *et al.* in [52], OSK can be easily accommodated to using time-memory trade-offs, which make the identification procedure efficient. It also provides forward-privacy to the tags. However, the synchronization issue present in OSK and its variants, although mitigatable, remains significant.

In that regard, the O-TRAP protocol has no such synchronization issue because tags automatically "re-synchronize" with each authentication attempt. It is also the reason why the identification procedure is constant-time in normal situations. However, it is very easy to make the next search linear by querying the tag once. This also leads to traceability issues using reader-side timing analysis. Additionally, it provides no forward-privacy.

Despite their respective weaknesses, these protocols are nonetheless probably the most solid solutions we analyzed.

5 Counter-Based Protocols

The *counter-based protocols* all share the same characteristics: they use a strictly increasing number[5] and maintain a periodically updated hash table for each

5. In some previous papers [18], [19], [67], [66] the name "*timestamp*" is used to denote a strictly increasing number. Since the tags do not have any clock and this number is not a cryptographic timestamp, we prefer using the more generic term *counter*.

counter. The idea is to pre-compute the table at each counter tick, in order to reduce the online search to a constant time on the server-side.

In this section we examine a family of counter-based protocols, namely RIP, RIP+, RAP, and YA-TRAP* (see Table 1 for the names given in different papers). We show a traceability attack on the most advanced protocol proposed in [66], namely YA-TRAP*, based on timing analysis.

5.1 YA-TRAP Family

A family of tag identification and authentication protocols that use strictly increasing counters is proposed in the papers [18], [19], [67], [66]. The first protocol, RIP, stands for *RFID Identification Protocol*. It is followed by authentication protocols called RIP+, YA-TRAP*, and a variant of YA-TRAP* with forward-privacy (we call this protocol YA-TRAP*&fwd).

5.1.1 Description

We describe below the RIP [19] protocol, which is the simplest and earliest proposal in the family.

Each tag $\mathcal{T}_i$ is initialized with a starting counter T_0 and a maximum counter value $T_{\max}$, as well as with a unique secret key K_i. When initiating an authentication, the reader sends its current counter T_r. The tag checks that T_r is less than $T_{\max}$ and that the received counter is bigger than the one it currently stores, T_t, which it received during the last successful identification. If these conditions hold, it stores the new counter and computes and sends the hash of T_r with its key K_i. Otherwise, the tag sends a random number to prevent an adversary from drawing any conclusion. The authors added that to avoid timing attacks against a tag at this point, the nonce generation must be designed to take approximately the same time as the hash computation.

As stated above, every now and then, the server increases the value of the counter, and re-computes the table accordingly. This allows for a constant time identification online, but takes time offline.

The authors identified several drawbacks in this protocol. First, it is vulnerable to a trivial DoS attack: the adversary can temporarily or permanently incapacitate a tag by sending a future counter. Although the authors point out that DoS resistance is not the main goal of this protocol, the attack is very easy to perform and very hard to recover from. Second, it is implicitly assumed that a tag is never identified more than once between two consecutive counter ticks. A short time interval (e.g., a second) between two counter updates makes this assumption realistic, but it causes heavy computational burden for the server. RIP is also vulnerable to replay attacks: an adversary can send a counter slightly ahead to a tag and wait until this counter is sent by the server. She can repeat this attack and thus impersonate its victim for a long time without the original tag being present. RIP is depicted in Fig. 8.

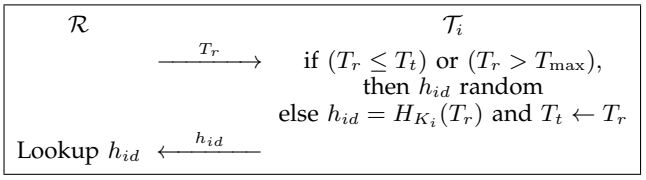


Figure 8. RIP protocol.

In RIP+, the protocol is modified in order to provide authentication. The reader sends a random nonce R_r along with the counter. The tag chooses its random nonce R_t and computes a hash for authentication $h_{auth} = H_{K_i}(R_t, R_r)$. The reader first identifies the tag, then checks the correctness of the hash.

Note that although this prevents the replay attack, the two aforementioned issues are still present.

In order to cope with DoS attacks, Tsudik proposed YA-TRAP*, which is illustrated in Fig. 9. DoS resistance is achieved by using a system-wide hash-chain. At setup, the system initializes a long Lamport-chain [44] of hashes, and sets the value ET_t of all tags to the last hash computed. Every INT counter ticks, a value of the hash-chain is popped, and the next one is used as ET_r. During an authentication session, a tag receiving T_r, R_r and ET_r will compute the number of intervals skipped since the last authentication (i.e. $\nu = \lfloor T_r/INT \rfloor - \lfloor T_t/INT \rfloor$), and will verify that the hash ET_r is the corresponding predecessor of ET_t by checking whether $H^{\nu}(ET_r) = ET_t$[6].

Note that DoS resistance in YA-TRAP* is limited by the magnitude of INT value. When ET_r is sent by the system it is no longer secret. Therefore, the adversary can still incapacitate tags up to the duration of INT by querying the tag with the maximum possible T_r value within the current epoch.

All the aforementioned protocols do not provide forward-privacy because the long-term key of the tags are static. Tsudik introduces an additional operation for updating the keys of the tags. In this extension, which we denote YA-TRAP*&fwd hereafter, a tag takes ν times hash of the key for each authentication namely $K_i^{\nu} = H^{\nu}(K_i)$. With this modification, the tag's key is changed once per INT interval, and this brings ν additional hash operations on the tag-side.

5.1.2 Attacks on YA-TRAP*.

In YA-TRAP*, the tag computes ν times the hash function depending on the difference between the T_t and T_r values. If the received T_r value is within the same interval as T_t, the tag computes no hash function for the interval check. If the difference between these two counters is large, the tag has to compute many hash functions. This leads to two potential attacks.

6. Note that in [66], the authors mistakenly stated this check was $H^{\nu}(ET_t) = ET_r$.

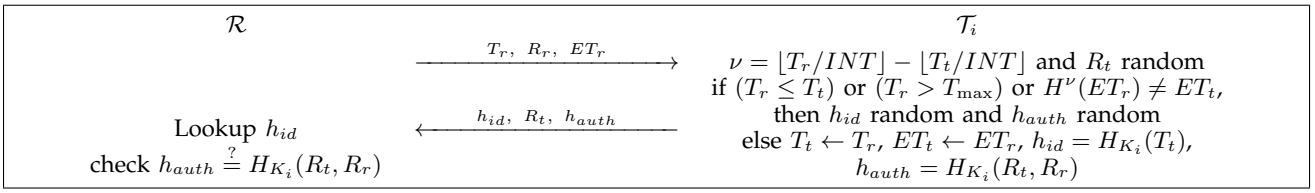


Figure 9. YA-TRAP* protocol.

The first one is a traceability attack. It is simply that if a tag has not been authenticated in a long time, it is traceable due to the amount of time it spends computing the hashes. Distinction is thus possible between two tags in some situations.

The second one is a DoS. If an adversary sends a big T_r and a random ET_r to a tag, the latter needs to compute many hashes, even if it will eventually discard the request since the ET_r is not correct. Depending on INT, this can make the authentication impossible due to the amount of time needed by the tag to complete its calculation.

The parameter INT must be carefully chosen: the bigger, the less it mitigates the DoS already present in RIP+; and the smaller, the more computation on tags, leading to the two problems described above.

5.1.3 Other Protocols.

Another counter-based protocol called YA-TRAP+ is proposed by Burmester *et al.* in 2006 [18], [24]. A slightly modified version of it is presented in [19] with a new name "RAP"). This protocol is very similar to O-RAP in terms of security properties. In [19] it is also stated that "O-RAP is simpler than RAP, at the cost of not supporting kill-keys. The security for O-RAP is similar to that of RAP." In particular, the two issues mentioned in Sect. 4.4 are also applicable to RAP. Additionally, in O-RAP a desynchronized tag is resynchronized automatically after each legitimate authentication, however RAP does not support automatic resynchronization. For these reasons, we only analyses O-RAP among those two similar protocols.

5.2 Discussion

Counter-based protocols, embodied by the YA-TRAP family, provide an interesting approach to constant-time identification. However, since the counter must be provided in the clear and, as such, is not authenticated, DoS attacks are extremely easy to accomplish and hard to prevent. YA-TRAP* attempts to alleviate this problem but at the same time introduces other weaknesses as indicated in section 5.1.2.

6 Comparison

In this section we summarize most of the protocols we analyzed and compare them on several criteria, as shown in Table 2. We evaluate the schemes that provide sub-linear complexity, at least during the normal case online interaction, and that intend to provide at least user privacy (not necessarily forward-privacy). We also include those for which we highlight new weaknesses in this paper. For clarity reasons, we provide additional remarks (superscripted capital letters in the table) below the table. It is difficult to compare these protocols objectively because of the number of criteria available. Nonetheless, we attempt to give some insights below about their comparison.

A major design goal for authentication protocols is the protection against impersonation attacks. Cheon, Hong, and Tsudik's plain protocol can therefore be discarded since it does not satisfy this requirement. OSK and its variants do not satisfy it either, but they are identification protocols, and it is possible to extend them to authentication protocols as explained in Sect. 4.2. For the rest of the protocols, we observe generally a trade-off between efficiency of the reader authentication complexity, and privacy weaknesses and/or other issues.

The protocols using shared secrets, although presenting alluring identification efficiency, have important security and privacy problems, as stated earlier. They are particularly vulnerable in scenarios where tag corruption is easy. Nonetheless, future ideas might lower the impact of tag compromise, and this approach remains interesting.

The counter-based protocols, embodied by the YA-TRAP family, seem to be promising as well, but are easily desynchronized, which decreases the privacy they provide. Their usability (a maximum of one authentication per counter tick) might be a problem in some applications too. If this is not a problem, YA-TRAP* provides a decent level of privacy and allows automatic re-synchronization.

Finally, although not ideal, the protocols based on hash-chains seem to be the most solid solutions to date among the protocols we analyzed. OSK/AO and OSK/BF provide forward-privacy and a very good efficiency for the authentication on the reader side, but have desynchronization issues due to the finite size of the chains. O-RAP is also quite good and does not have desynchronization problems. However, if we consider an adversary capable of performing timing analysis, it has a lower privacy. O-FRAP also brings some forward-privacy to O-RAP (but not completely as we point out in Sect. 4.4.3).

Some protocols require fewer assets on the tag. For instance, hash-chains protocols do not require ran-

Table 2
Comparison of the protocols analyzed in this article. Letters in brackets link to comments described below.

CLASS	SHARED SECRETS			HASH-CHAINS					COUNTER-BASED		
PROTOCOL	CHT plain	CHT with auth	CTI	OSK	OSK/AO with Auth	OSK/BF with Auth	O-RAP	O-FRAP	RIP+	YA-TRAP*	YA-TRAP* &fwd
MAIN REFERENCE	[26]	[26]	[2]	[54]	[5], [8], [10]	[52]	[18]	[68]	[66]	[66]	[66]
YEAR OF PUBLICATION	2009	2009	2010	2003	2005	2008	2006	2007	2007	2007	2007
Identification/ Authentication	auth.[A]	auth.	auth.	id.	auth.	auth.	auth.	auth.	auth.	auth.	auth.
Off-line Computation Complexity[B]	0	0	$O(NC)$[C]	$2N$[C]	$\frac{NM^2}{2}$ ([8])[D]	$2NM$[C]	0	0	N/counter update[E]	N/counter update[E]+ Lamport Chain	N/counter and key update[E] + Lamport Chain
Normal case online complexity	$O(\sqrt{N})$	$O(N^a)$	4	2	$O(N^{2/3})$[F]	$M(\epsilon N+3)$ on average	1	2	0[E]+1[G]	0[E]+1[G]	0[E]+1[G]
Desynchronized case online complexity	N/A	N/A	N/A	lower than $2N(M-1)$[H]	$O(N^{2/3})$[F]	$M(\epsilon N+3)$ on average	$O(N)$	$O(N)$	out of order after desync.[H]	out of order after desync.[I]	out of order after desync.[I]
Memory Complexity	$2\sqrt{N}$	$2N^a+N$	$O(N)$[J]	N	$O(N^{2/3})$[F]	$\frac{NM\log\epsilon}{-\log^2 2}$	$2N$	$3N$	N[E]	N[E]	N[E]
Tag Computation	2 PRFs + 1 Nonce	3 PRFs + 1 Nonce	5 hashes	2 hashes	3 hashes	3 hashes	2 hashes	4 hashes	2 hashes + 1 Nonce	$\nu+2$ hashes + 1 Nonce	$2\nu+2$ hashes + 1 Nonce
Tag Resources	PRF, PRNG	PRF, PRNG	PRNG, Hash func.	Hash func.	Hash func.	Hash func.	Hash func.	Hash func.	PRNG, Hash func.	PRNG, Hash func.	PRNG, Hash func.
Privacy	no	no★	no★	yes[K]	yes◊,[K]	no★	no★,[K]	no★,[K]	not private after desync.	not private after desync.★,[L]	not private after desync.★,[L]
Forward-privacy	no	no	no[M]	yes[K]	yes[K]	no[M]	no	no★	no	no	no[N]
Desynchronization resistance	N/A	N/A	yes	yes up to M consecutive[O]	yes up to M consecutive[P]	yes up to M consecutive[O]	no[Q]	no[Q]	no[R]	yes[S]	yes[S]
Impersonation Resistance	no★	yes	yes	N/A	yes	yes	yes	yes	yes	yes	yes

★ : Weaknesses discovered in this paper. ◊ : Weaknesses discovered and fixed in this paper.

[A] Although the authors implicitly consider it to be an identification protocol (because of the existence of an *authentication* extension), we denote it by an authentication protocol, since the reader sends a nonce, and since the protocol is at first designed to cope with impersonation.
[B] Excluding key generation.
[C] Done during the setup.
[D] Done each time one tag reaches M authentications since the last table update.
[E] The whole hash table is periodically updated or can be precomputed for forecoming counters.
[F] Using rainbow tables. The complexity provided is an example, but the identification complexity can be set anywhere between $O(1)$ and $O(N)$ according to the memory available for the trade-off (see Sect. 4.2 for discussion).
[G] Additional hash for authentication.
[H] The desynchronized tag might not be identified/authenticated by the reader.
[I] The tag cannot be authenticated within the time interval. It gets resynchronized in the next one.
[J] Can be big due to constant terms (see Sect. 3.3 and [2]).
[K] Private if and only if the adversary is not able to tell whether the protocol session was successful.
[L] However, private after resynchronization. The tag resynchronize automatically at each interval start.
[M] Since it is not private.
[N] Not forward private until the last ET update.
[O] After M desynchronizations, the tag owner can go to some central office to fix the issue.
[P] Including legitimate authentications. If the tag owner goes to the office the tag can be resynchronized in the next precomputation. However, if the number of the illegitimate authentications is less than M, then the resynchronization of the tag will be done automatically during the next update.
[Q] Tags can be desynchronized, but resynchronize automatically after each authentication.
[R] Either up to $T_{\max}$ (tag becomes useless) or with a smaller counter (for to traceability and replay).
[S] Tags can be easily desynchronized within a time interval (e.g. one day), making them unusable and/or traceable. Tags are automatically resynchronized at the beginning of each interval. Tags have limited lifetime because the Lamport chain must have a start (although the system can be initialized with a really big hash-chain).

domness to originate from tags and might therefore be more easily implemented on low-cost tags.

In conclusion, the choice of the best protocol depends on the scenario, on the privacy and efficiency requirements. However, we can point that the protocols based on hash-chains clearly stand out.

7 Conclusion

In this paper, we studied a number of identification and authentication protocols based on classical symmetric-key cryptographic building blocks (e.g. hash functions) and providing sub-linear online complexity to identify users. We first evaluated each of the schemes by examining whether they satisfy a set of security properties under a well-known adversarial model [42]. We have shown two new attacks on the CHT protocol [26] which is a very efficient protocol in terms of key search complexity (i.e., $O(\sqrt{N})$). We also introduced two new traceability attacks on the CTI protocol [2]. Furthermore, we have shown a traceability weakness of the mutual authentication version of OSK/AO [8] protocol, and shown a possible way to repair this problem with no additional cost. We also introduce traceability attacks on OSK/BF [52], O-RAP [24] and YA-TRAP* [66], which emphasize the importance of timing attacks [7] on the reader side. Finally, we have extensively evaluated and compared all the candidates according to their security and performance. The security properties that we investigated include user privacy, forward privacy, impersonation resiliency and desynchronization resistance. Furthermore, we examined thoroughly their performance, in terms of computational and storage cost.

References

[1] M. Akgün, M. U. Caglayan, and E. Anarim, "Secure RFID Authentication with Efficient Key-lookup," in *Proceedings of the 28th IEEE conference on Global telecommunications – GLOBECOM'09*.

[2] B. Alomair, A. Clark, J. Cuellar, and R. Poovendran, "Scalable RFID Systems: a Privacy-Preserving Protocol with Constant-Time Identification," in *the 40th Annual IEEE/IFIP International Conference on Dependable Systems and Networks – DSN'10*.

[3] G. Avoine, "RFID lounge," http://www.avoine.net/rfid/.

[4] ——, "Adversary Model for Radio Frequency Identification," Technical Report, 2005.

[5] ——, "Cryptography in Radio Frequency Identification and Fair Exchange Protocols," Ph.D. dissertation, 2005.

[6] G. Avoine, L. Buttyán, T. Holczer, and I. Vajda, "Group-based private authentication," in *IEEE International Workshop on Trust, Security, and Privacy for Ubiquitous Computing – TSPUC*, 2007.

[7] G. Avoine, I. Coisel, and T. Martin, "Time Measurement Threatens Privacy-Friendly RFID Authentication Protocols," in *Workshop on RFID Security – RFIDSec'10*.

[8] G. Avoine, E. Dysli, and P. Oechslin, "Reducing Time Complexity in RFID Systems," in *Selected Areas in Cryptography – SAC 2005*.

[9] G. Avoine, B. Martin, and T. Martin, "Tree-Based RFID Authentication Protocols Are Definitively Not Privacy-Friendly," in *Workshop on RFID Security – RFIDSec'10*.

[10] G. Avoine and P. Oechslin, "A Scalable and Provably Secure Hash Based RFID Protocol," in *International Workshop on Pervasive Computing and Communication Security – PerSec 2005*.

[11] C. Berbain, O. Billet, J. Etrog, and H. Gilbert, "An Efficient Forward Private RFID Protocol," in *Conference on Computer and Communications Security – ACM CCS'09*.

[12] M. Beye and T. Veugen, "Improved anonymity for key-trees," Cryptology ePrint Archive, Report 2011/395.

[13] O. Billet, J. Etrog, and H. Gilbert, "Lightweight Privacy Preserving Authentication for RFID Using a Stream Cipher," in *Fast Software Encryption – FSE'10*.

[14] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, 1970.

[15] J. Bringer, H. Chabanne, and D. Emmanuelle, "HB^{++}: a Lightweight Authentication Protocol Secure against Some Attacks," in *IEEE International Conference on Pervasive Services, Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing – SecPerU 2006*.

[16] J. Bringer, H. Chabanne, and T. Icart, "Cryptanalysis of EC-RAC, a RFID identification protocol," in *7th International Conference on Cryptology And Network Security – CANS'08*.

[17] A. Broder and M. Mitzenmacher, "Using multiple hash functions to improve IP lookups," in *Procedings of the twentieth Annual Joint Conference of the IEEE Computer and Communications Societies – INFOCOM 2001*.

[18] M. Burmester, T. v. Le, and B. de Medeiros, "Provably Secure Ubiquitous Systems: Universally Composable RFID Authentication Protocols," in *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm 2006*.

[19] ——, "Universally Composable RFID Identification and Authentication Protocols," *ACM Transactions on Information and System Security – TISSEC'09*.

[20] L. Buttyán, T. Holczer, and I. Vajda, "Optimal Key-Trees for Tree-Based Private Authentication," in *Workshop on Privacy Enhancing Technologies – PET 2006*.

[21] S. Canard and I. Coisel, "Data Synchronization in Privacy-Preserving RFID Authentication Schemes," in *Workshop on RFID Security – RFIDSec'08*.

[22] S. Canard, I. Coisel, and M. Girault, "Security of Privacy-Preserving RFID Systems," in *IEEE International Conference on RFID-Technology and Applications – RFID-TA'10*.

[23] A. Cavioukan, "Privacy guidelines for RFID information systems (RFID privacy guidelines)," Office of the Information and Privacy Comissioner, Ontario, June 2006.

[24] C. Chatmon, T. van Le, and M. Burmester, "Secure Anonymous RFID Authentication Protocols," Technical Report, 2006.

[25] C.-L. Chen and Y.-Y. Deng, "Conformation of EPC Class 1 Generation 2 standards RFID system with mutual authentication and privacy protection," *Engineering Applications of Artificial Intelligence*, 2009.

[26] J. H. Cheon, J. Hong, and G. Tsudik, "Reducing RFID Reader Load with the Meet-in-the-Middle Strategy," Cryptology ePrint Archive, Report 2009/092.

[27] H.-Y. Chien, "SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity," *IEEE Transactions on Dependable and Secure Computing*, 2007.

[28] H.-Y. Chien and C.-H. Chen, "Mutual Authentication Protocol for RFID Conforming to EPC Class 1 Generation 2 standards," *Computer Standars & Interfaces, Elsevier*, 2007.

[29] M. David and N. R. Prasad, "Providing strong security and high privacy in low-cost RFID networks," in *Security and Privacy in Mobile Information and Communication Systems*, 2009.

[30] W. Diffie and M. Hellman, "Special Feature Exhaustive Cryptanalysis of the NBS Data Encryption Standard," *Computer*, 1977.

[31] T. Dimitriou, "A Lightweight RFID Protocol to protect against Traceability and Cloning attacks," in *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm 2005*.

[32] D. N. Duc and K. Kim, "Securing HB+ against GRS Man-in-the-Middle Attack," in *Institute of Electronics, Information and Communication Engineers, Symposium on Cryptography and Information Security*, 2007.

[33] T. Halevi, N. Saxena, and S. Halevi, "Using HB Family of Protocols for Privacy-Preserving Authentication of RFID Tags in a Population," in *Workshop on RFID Security – RFIDSec'09*.

[34] D. Han and D. Kwon, "Vulnerability of an RFID authentication protocol conforming to EPC Class 1 Generation 2 Standards," *Comput. Stand. Interfaces*, 2009.

[35] D. Hein, J. Wolkerstorfer, and N. Felber, "ECC is Ready for RFID A Proof in Silicon," in *Workshop on RFID Security – RFIDSec'08*.

[36] M. Hellman, "A cryptanalytic time-memory trade-off," *Information Theory, IEEE Transactions on*, 1980.

[37] D. Henrici and P. Müller, "Hash-Based Enhancement of Location Privacy for Radio-Frequency Identification Devices Using Varying Identifiers," in *International Workshop on Pervasive Computing and Communication Security – PerSec 2004*.

[38] N. Hopper and M. Blum, "Secure Human Identification Protocols," *Advances in Cryptology – Asiacrypt 2007*, 2001.

[39] M. Hutter, M. Feldhofer, and T. Plos, "An ECDSA Processor for RFID Authentication," in *Workshop on RFID Security – RFIDSec'10*.

[40] A. Juels, "Minimalist Cryptography for Low-Cost RFID Tags," in *International Conference on Security in Communication Networks – SCN 2004*.

[41] A. Juels and S. Weis, "Authenticating Pervasive Devices with Human Protocols," in *Advances in Cryptology – CRYPTO'05*.

[42] ——, "Defining Strong Privacy for RFID," in *International Conference on Pervasive Computing and Communications – PerCom 2007*.

[43] T. Karygiannis, B. Eydt, G. Barber, L. Bunn, and T. Phillips, "Guidelines for Securing Radio Frequency Identification (RFID) Systems," NIST Special Publication, 2007.

[44] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, 1981.

[45] Y.-C. Lee, Y.-C. Hsieh, P.-S. You, and T.-C. Chen, "A New Ultralightweight RFID Protocol with Mutual Authentication," in *WASE International Conference on Information Engineering – ICIE '09*.

[46] Y. K. Lee, K. Sakiyama, L. Batina, and I. Verbauwhede, "Elliptic-Curve-Based Security Processor for RFID," *IEEE Transactions on Computers*, 2008.

[47] L. Lu, J. Han, L. Hu, Y. Liu, and L. Ni, "Dynamic Key-Updating: Privacy-Preserving Authentication for RFID Systems," in *International Conference on Pervasive Computing and Communications – PerCom 2007*.

[48] D. Molnar and D. Wagner, "Privacy and Security in Library RFID: Issues, Practices, and Architectures," in *Conference on Computer and Communications Security – ACM CCS'04*.

[49] J. Munilla and A. Peinado, "HB-MP: A further step in the HB-family of lightweight authentication protocols," *Computer Networks*, 2007.

[50] D. Nguyen Duc, J. Park, H. Lee, and K. Kim, "Enhancing Security of EPCglobal Gen-2 RFID Tag against Traceability and Cloning," in *Symposium on Cryptography and Information Security*, 2006.

[51] Y. Nohara and S. Inoue, "A Secure and Scalable Identification for Hash-based RFID Systems Using Updatable Precomputation," in *Proceedings of the 3rd ACM Conference on Wireless Network Security – WiSec'10*.

[52] Y. Nohara, S. Inoue, and H. Yasuura, "A secure high-speed identification scheme for RFID using bloom filters," in *Third International Conference on Availability, Reliability and Security – AReS 2008*.

[53] K. Nohl and D. Evans, "Quantifying Information Leakage in Tree-Based Hash Protocols," in *International Conference on Information and Communications Security – ICICS'06*.

[54] M. Ohkubo, K. Suzuki, and S. Kinoshita, "Cryptographic Approach to "Privacy-Friendly" Tags," in *RFID Privacy Workshop*, 2003.

[55] ——, "Efficient Hash-Chain Based RFID Privacy Protection Scheme," in *International Conference on Ubiquitous Computing – Ubicomp, Workshop Privacy: Current Status and Future Directions*, 2004.

[56] K. Ouafi and R. C.-W. Phan, "Privacy of Recent RFID Authentication Protocols," in *4th International Conference on Information Security Practice and Experience – ISPEC 2008*.

[57] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, T. Li, and J. C. van der Lubbe, "Weaknesses in Two Recent Lightweight RFID Authentication Protocols," in *Workshop on RFID Security – RFIDSec'09*.

[58] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "Advances in Ultralightweight Cryptography for Low-cost RFID Tags: Gossamer Protocol," in *Workshop on Information Security Applications – WISA'08*.

[59] ——, "Cryptanalysis of a Novel Authentication Protocol Conforming to EPC-C1G2 Standard," in *Workshop on RFID Security – RFIDSec'07*.

[60] ——, "EMAP: An Efficient Mutual Authentication Protocol for Low-Cost RFID Tags," in *OTM Federated Conferences and Workshop: IS Workshop – IS'06*.

[61] ——, "LMAP: A Real Lightweight Mutual Authentication Protocol for Low-cost RFID tags," in *Workshop on RFID Security – RFIDSec'06*.

[62] ——, "M2AP: A Minimalist Mutual-Authentication Protocol for Low-cost RFID Tags," in *International Conference on Ubiquitous Intelligence and Computing – UIC'06*.

[63] P. Peris-Lopez, J. C. Hernandez-Castro, J. E. Tapiador, and J. C. A. van der Lubbe, "Cryptanalysis of an EPC Class-1 Generation-2 standard compliant authentication protocol," *Engineering Applications of Artificial Intelligence*, 2011.

[64] V. Reding, "Commission recommendation of 12.05.2009 - sec(2009) 585/586, on the implementation of privacy and data protection principles in applications supported by radio-frequency identification," Commission of the European Communities, May 2009.

[65] J. Simitian, "Californian senate bill no.682," 2005.

[66] G. Tsudik, "A Family of Dunces: Trivial RFID Identification and Authentication Protocols," in *Workshop on Privacy Enhancing Technologies – PET 2007*.

[67] ——, "YA-TRAP: Yet Another Trivial RFID Authentication Protocol," in *International Conference on Pervasive Computing and Communications – PerCom 2006*.

[68] T. Van Le, M. Burmester, and B. de Medeiros, "Universally Composable and Forward-secure RFID Authentication and Authenticated Key Exchange," in *ACM Symposium on Information, Computer and Communications Security – ASIACCS 2007*.

[69] S. Vaudenay, "On Privacy Models for RFID," in *Advances in Cryptology – Asiacrypt 2007*.

[70] W. Wang, Y. Li, L. Hu, and L. Lu, "Storage-awareness: RFID private authentication based on sparse tree," in *Security, Privacy and Trust in Pervasive and Ubiquitous Computing, 2007. SECPerU 2007. Third International Workshop on*.

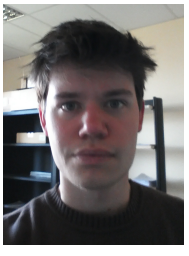
[71] Q. Yao, Y. Qi, J. Han, J. Zhao, X. Li, and Y. Liu, "Randomizing RFID private authentication," in *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*.

[72] K.-H. Yeh, N. Lo, and E. Winata, "An Efficient Ultralightweight Authentication Protocol for RFID Systems," in *Workshop on RFID Security – RFIDSec Asia'10*.

[73] T.-C. Yeh, Y.-J. Wang, T.-C. Kuo, and S.-S. Wang, "Securing RFID systems conforming to EPC Class 1 Generation 2 standard," *Expert System Applications*, 2010.

Gildas Avoine is a professor of information security and cryptography at the UCL in Louvain-la-Neuve (Belgium), where he leads the Information Security Group (GSI). Before joining the UCL, he was researcher at the MIT (USA) and at the EPFL (Switzerland), where he obtained a PhD degree in cryptography. Previously, he studied at the University of Caen (France) where he received a Bachelor degree in mathematics and Bachelor and Master degrees in computer science.

Muhammed Ali Bingöl received his B.Sc. and M.Sc. degrees from ITU (Istanbul Technical University), Turkey in 2008 and 2012, respectively. He has been a researcher at TUBITAK BILGEM UEKAE (National Research Institute of Electronics & Cryptology) since 2008. His primary research interests include designing and analyzing cryptographic protocols, RFID security and privacy and secret sharing schemes.



Xavier Carpent is a PhD student at the UCL in Louvain-la-Neuve (Belgium). He received his B.Sc dedgree from the UCL in 2008 and his M.Sc degree from the UCL and the UPC (Barcelona). His primary research interests include RFID authentication protocols and time-memory trade-offs.

Siddika Berna Ors Yalcin received the Electronics and Communication Engineering degree and the M.Sc in 1995 and 1998, respectively, both from the Istanbul Technical University (Turkey). She received the Electrical Engineering degree in Applied Siences in 2005 from the Katholieke Universiteit Leuven (Belgium). She is currently an associative professor at the ITU. Her main research interests are cryptography, embedded systems and side-channel attacks.