Untraceability of RFID Protocols

Ton van Deursen, Sjouke Mauw, and Saša Radomirović

Université du Luxembourg
Faculté des Sciences, de la Technologie et de la Communication
6, rue Richard Coudenhove-Kalergi
L-1359, Luxembourg

Abstract. We give an intuitive formal definition of untraceability in the standard Dolev-Yao intruder model, inspired by existing definitions of anonymity. We show how to verify whether communication protocols satisfy the untraceability property and apply our methods to known RFID protocols. We show a previously unknown attack on a published RFID protocol and use our framework to prove that the protocol is not untraceable.

Keywords: RFID protocols, untraceability, formal verification.

1 Introduction

Radio frequency identification (RFID) systems aim to identify tags to readers in an open environment. Communication between readers and tags is even possible when there is no physical or visual contact between tags and readers. RFID tags can be very small and cheap [1] and can therefore be embedded in a wide variety of objects. They have, for instance, been embedded in passports [2] and there are plans to embed them in bank notes [3] and groceries [4, 5].

The absence of physical contact during communication and the expected ubiquity of RFID systems will only encourage nefarious entities to trace and observe tags through time and space. If at any such point a tag is linked to a person, the tracing of a tag becomes the tracing of a person.

The need for RFID protocols to be resistant against this kind of attack on privacy has been realized early on. Intuitively, protocols are said to provide untraceability, if an adversary is not able to recognize a tag he previously observed. Although untraceability is mainly mentioned in the context of RFID systems, it is an issue for any protocol which is used with a mobile device. In the Bluetooth setting, it is known as location privacy [6, 7].

History has shown that designing protocols is a difficult and error-prone task and that formal verification of security properties is necessary [8, 9]. While traditional security properties such as authentication and secrecy have been studied thoroughly, untraceability has only become relevant with the introduction of travelling devices. Until now it has typically been treated rather informally. In some cases, protocol designers prove untraceability of their protocols without even defining it properly.

In this paper, we propose an intuitive, formal definition for untraceability that is inspired by existing definitions for anonymity [10, 11]. We demonstrate the usability of our definition on two protocols. In particular, we prove that the mutual authentication protocol by Feldhofer, Dominikus, and Wolkerstorfer [12] is untraceable and that the Di Pietro and Molva protocol [13] is untraceable only for a *restricted choice of parameters* and assuming that their constructed function, DPM, is a *perfect hash function*. By removing the assumptions and analyzing the algebraic properties of the DPM function we demonstrate the first, efficient method to trace tags running the Di Pietro-Molva protocol. We then relate this insight back to our definition of untraceability by exhibiting a trace of the protocol which violates our definition.

Our paper is structured as follows. In the next section we discuss related work. In Section 3 we formally define untraceability. In Section 4 we prove the Feldhofer *et. al.* protocol untraceable and in Section 5 we discuss the Di Pietro-Molva protocol. We conclude with an outlook on future work in Section 6.

2 Related Work

A discussion of the importance of untraceability can be found in [14–18]. Several RFID protocols have been proposed with informal reasoning about their untraceability property [19–22] or based on the belief that protocols with random nonces in all messages are untraceable [23–25]. On the opposite end of the spectrum, pseudonyms and frequent changes of IDs are claimed to be necessary to avoid the tracking problem [26–29]. Among the cryptographic notions of untraceability, worth mentioning are [30–36].

The notion of untraceability defined in this paper is stronger than the notions considered in [37, 38] in the following sense. These works consider RFID tags which an adversary could recognize between any two successful communications with a trusted RFID reader to be untraceable, while under the present definition they are not.

The untraceability notion considered here is only weakly related to the unlinkability notion that has been studied extensively in privacy enhancing technologies (PET) literature. A formal definition for unlinkability was given in [39, 40]. Unlinkability considers whether links can be established between senders and receivers, while untraceability considers whether different communications can be attributed to the same agent. It is difficult to give a precise relation between anonymity and untraceability due to the many differing definitions of anonymity [10, 11, 41]. In general however, untraceability implies anonymity.

Security properties such as secrecy and authentication, implemented by a protocol at a certain layer, are maintained in the lower layers. However, for untraceability, the property can be compromised by the protocols in the lower layer [42]. In this paper, we will focus on untraceability in the application layer.

3 Definitions

3.1 Security Protocol Model

The purpose of this section is to introduce basic notation and definitions concerning security protocols. Rather than providing a full description of security protocol syntax and semantics, we will only present the basic requirements needed for defining and analyzing untraceability. In short, we require that the behavior of a number of agents executing a security protocol is described by a set of traces in which we can identify the events belonging to the same run. A full semantics satisfying our requirements can be found in [43].

The starting point is the specification of a security protocol. A security protocol defines the behavior of a set of *roles* (e.g. initiator, responder, server). A role specification consists of a sequence of events (e.g. the sending or reception of a message). The messages contained in the events are *role terms*. Role terms are built from *basic role terms*, such as nonces (typically denoted by n), role names (e.g. R or T), or keys (typically denoted by k). Complex terms can be constructed using functions, such as tupling (denoted by $(t_1, \ldots, t_n)$), encryption ($\{t\}_k$), hashing ($h(t)$), and exclusive or (denoted by $\oplus$). Throughout this paper we will use Message Sequence Charts to present security protocol specifications (see e.g. Figure 1 in Section 4). In such a diagram, we use a hexagon at the end of a role specification to denote a security claim, such as untraceability.

A role specification is only a blueprint of some actual behavior. It serves as the program that an agent (typically denoted by *Alice* or *Bob*) can execute. An execution of a role R specified by a protocol P is called a *run* of R. Such a run will be denoted by $R\#\theta$, where θ is a (unique) run identifier. A run can thus be viewed as an instantiation of a role. Therefore, we will also have to instantiate the abstract role events, yielding the *run events*. Run events are constructed from role events by instantiating the contained role terms. An instantiated role term is a *run term*. Run terms are similar to role terms, except that roles are replaced by agents and that basic role terms are suffixed with the identifier of the run. An instantiated nonce n is denoted by $n\#\theta$ if it occurs in run $R\#\theta$. In this way occurrences of the same nonce in different runs can be distinguished.

We assume a standard Dolev-Yao adversary, characterized by its *knowledge*. This knowledge consists of the set of run terms that the adversary initially knows, extended with the terms obtained by observing the runs. We assume that the adversary has unlimited inference capabilities, meaning that he can combine the information in his knowledge to construct or interpret new terms. However, this capability is restricted due to the assumption of perfect cryptography. This means that the adversary cannot reverse hash functions and that he is not able to learn the contents of an encrypted term, unless he knows the decryption key. We denote the inference of term t from term set M by $M \vdash t$. We model corrupted agents by assuming that all secrets of these agents (e.g. secret keys) are contained in the initial knowledge of the adversary. When evaluating security claims, we will only be interested in claims made by trusted (i.e. non-corrupt) agents.

Finally, we assume that the behavior of a collection of agents executing a security protocol is given as a set of traces. Each trace consists of a number of interleaved runs or run prefixes. A run prefix occurs if an agent cannot finish his execution of a role specification (e.g. because the expected input is never provided). We assume that within a trace t the events belonging to run $R\#\theta$ can be identified. Let $t_{R\#\theta}$ denote the subtrace of t consisting of the events of run (or run prefix) $R\#\theta$ which are observable by the adversary. We enumerate all non-empty subtraces $t_{R\#\theta}$ according to the occurrence of their first observable event in trace t. The i-th such subtrace is denoted by t_i^R. The agent executing the events in this subtrace is denoted by $agent(t_i^R)$.

3.2 Untraceability

We define untraceability as a trace property of a role in a protocol. Informally, a role is called untraceable if for every instantiation of the role which is linked to another instantiation of the role, there is a trace that is indistinguishable to the adversary, in which the two instantiations are not linked.

We will first define linkability, reinterpretation, and indistinguishability before presenting the definition of untraceability.

Definition 1 (linkability of subtraces). *Two subtraces t_i^R and t_j^R are linked, denoted by $L(t_i^R, t_j^R)$, if they are instantiated by the same agent:*

$$L(t_i^R, t_j^R) \equiv (agent(t_i^R) = agent(t_j^R)).$$

The notion of reinterpretation has been introduced in [10]. It will be used to express that subterms of a message can be substituted by other terms if the adversary is not able to read (or interpret) these subterms. All terms that the adversary can interpret remain unchanged.

Definition 2 (reinterpretation). *A map π from run terms to run terms is called a reinterpretation under knowledge set M if it and its inverse π^{-1} satisfy the following conditions:*

$\pi(m) = m$	*if m is a basic run term*
$\pi(m) = (\pi(m_1), \ldots, \pi(m_n))$	*if $m = (m_1, \ldots, m_n)$ is an n-tuple*
$\pi(\{m\}_k) = \{\pi(m)\}_k$	*if $M \vdash k^{-1}$*
	or $M \vdash m \wedge M \vdash k$
$\pi(f(m)) = f(\pi(m))$	*if $M \vdash m$*
	or f is not a hash function.

Note that the condition $\pi(f(m)) = f(\pi(m))$, when f is not a hash function, leads to an under-approximation of the intended notion of reinterpretation. This means that for certain functions f, there might be untraceable protocols which cannot be proven to be untraceable. In such cases, the condition would need to be refined based on the specific properties of such a function.

Reinterpretations generalize in the obvious way to traces. We use reinterpretations to define indistinguishability of traces. Two traces are indistinguishable to the adversary, if the adversary cannot see any meaningful difference between the two traces, based on the knowledge he has.

Definition 3 (indistinguishability of traces). *Let M be the adversary's knowledge at the end of trace t. The trace t is indistinguishable from a trace t', denoted by $t \sim t'$, if there is a reinterpretation π under M, such that $\pi(t_i^R) = t_i'^R$ for all roles R and subtraces t_i^R.*

We now have all ingredients to formally define untraceability. Untraceability is the property that for every trace of a protocol in which two subtraces are linked, there is a trace that is indistinguishable to the adversary in which these two subtraces are not linked.

Definition 4 (untraceability). *A protocol P is untraceable with respect to role R if*

$$\begin{aligned}&\forall_{t \in Traces(P)} \\ &\quad \forall_{i \neq j} L(t_i^R, t_j^R) \Rightarrow \\ &\qquad \exists_{t' \in Traces(P)} t \sim t' \wedge \neg L(t_i'^R, t_j'^R).\end{aligned}$$

4 An Untraceable Protocol

In [12], Feldhofer *et al.* present an AES hardware implementation for RFID tags along with two simple protocols for unilateral and mutual authentication, of which the unilateral authentication protocol can be proven traceable. In this section, we prove that the mutual authentication protocol is untraceable.

4.1 Protocol Description

The protocol assumes that every pair of reader R and tag T shares a unique key $k(R, T)$. These shared keys are initially not part of the adversary's knowledge. The reader initiates the protocol by sending a freshly generated nonce nr to the tag. The tag generates a nonce nt encrypts the pair (nr, nt) under the shared key $k(R, T)$, and sends it to the reader. The reader decrypts the message using the same shared key, reverses the order of the two nonces, encrypts the message under the shared key, and sends it to the tag. Figure 1 shows a graphical representation of the protocol specification.

4.2 Untraceability

Theorem 1. *The protocol depicted in Figure 1 is untraceable.*

Proof. We notice first that $k(R, T)$ and nt remain secret throughout the protocol execution. This can be easily verified by hand or with an automated tool.

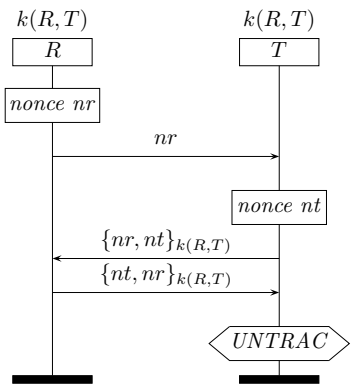

Fig. 1. An untraceable mutual authentication protocol.

Let t be a trace with subtraces t_i^T and t_j^T for $i \neq j$. We need to show that whenever $L(t_i^T, t_j^T)$ we can find a trace $t' \sim t$ such that $\neg L(t_i'^T, t_j'^T)$. For ease of notation, we set $agent(t_i^T) = agent(t_j^T) = agent(t_i'^T) = Alice$ and $agent(t_j'^T) = Bob$. The general idea of the proof is that t' can be constructed from t by replacing all occurrences of $Alice$ in t_j^T by Bob. We will make this more precise below and motivate that the adversary cannot distinguish between t and t'.

Since we are verifying the untraceability claim for an agent in role T, we may assume that the agent is trusted, i.e. that it executes all read and send events according to the specification. By definition, there is a θ such that the subtrace t_j^T contains the event where $\{nr\#\theta, nt\#\theta\}_{k(R\#\theta,T\#\theta)}$ is sent.

We consider the map π with the following properties:

$$\begin{array}{lll} \pi(\{x, nt\#\theta\}_{k(y,Alice)}) = \{x, nt\#\theta\}_{k(y,Bob)} & & \text{for any } x \text{ and } y, \\ \pi(\{nt\#\theta, x\}_{k(y,Alice)}) = \{nt\#\theta, x\}_{k(y,Bob)} & & \text{for any } x \text{ and } y, \\ \pi(m) = m & & \text{elsewhere.} \end{array}$$

Note that π is a reinterpretation under the adversary's knowledge, by Definition 2 and secrecy of $k(R,T)$.

Let $t' = \pi(t)$. We show that t' is a valid trace. The only difference between the traces t and t' occurs in messages containing the nonce $nt\#\theta$. By construction, the changes produce a valid run for Bob while keeping the reader's run valid. It follows from the secrecy of nt and $k(R,T)$ that any further occurrence of $nt\#\theta$ must be in $\{nr\#\theta, nt\#\theta\}_{k(R\#\theta,T\#\theta)}$ or $\{nt\#\theta, nr\#\theta\}_{k(R\#\theta,T\#\theta)}$. Since $nr\#\theta$ is produced by $R\#\theta$, no other run of R will accept the former message, and similarly, since $nt\#\theta$ is produced by $T\#\theta$, no other run of T will accept the latter message.

Finally, $t_i^T = t_i'^T$ thus $\neg L(t_i'^T, t_j'^T)$.

□

5 A Traceable Protocol

Di Pietro and Molva describe in [13] an identification and authentication protocol aimed at enhancing the security and privacy of RFID-based systems. We will first describe the Di Pietro-Molva protocol and then prove it untraceable for a *restricted choice of parameters* and the assumption that Di Pietro and Molva's DPM function is a *perfect hash function*. By lifting the restrictions and analyzing the algebraic properties of the DPM function we will demonstrate an efficient method to trace tags and discuss its practicality. Finally, we will relate the insight back to our definition of untraceability by exhibiting a trace of the Di Pietro-Molva protocol for which there is no valid, to the adversary indistinguishable, trace with unlinked subtraces.

5.1 Protocol Description

Let h be a cryptographic hash function, M, the majority function of three bits, defined by

$$\begin{aligned} M &: \mathbb{F}_2^3 \to \mathbb{F}_2 \\ (a,b,c) &\mapsto ac + bc + ab \end{aligned}$$

and for $\ell \in 3\mathbb{N}$,

$$\begin{aligned} DPM &: \mathbb{F}_2^\ell \to \mathbb{F}_2 \\ (x_1, \ldots, x_\ell) &\mapsto \sum_{i=1}^{\ell/3} M(x_{3i-2}, x_{3i-1}, x_{3i}). \end{aligned}$$

It is easy to verify that the functions M and DPM are identical to the corresponding functions in [13], except that we have defined them over vector spaces over the finite field with two elements instead of bit strings. In the remainder of this section we will identify elements in vector spaces over $\mathbb{F}_2$ with bit strings. We will denote the tags' and readers' unique ids by ID_T and ID_R, respectively. Every tag has a unique key k_T assigned to it by the key distribution center (KDC). A reader authorized to identify a tag T will be given by the KDC the key $k_{T,R} = h(k_T, ID_R, k_T)$. The keys are ℓ bits long.

The Di Pietro-Molva protocol, depicted in Figure 2, begins with the reader sending its ID and a random nonce n_j to the tag. The tag replies with the message $\alpha_1, \ldots, \alpha_q, V, \omega$, where $\alpha_i = k_{T,R} \oplus r_i$ for randomly chosen r_i (an ℓ-bit vector), the i-th bit of V (a q-bit vector) is $DPM(r_i)$, and $\omega = h(k_{T,R}, n_j, r_1, k_{T,R})$. The reader has a database of keys $k_{T,R}$. The reader can find the key $k_{T,R}$ with the help of the vectors α_i and values $DPM(r_i)$ by iterating through all possible keys. It is expected that each α_i reduces the number of possible keys by approximately one half. ω can be used to uniquely identify the correct key. The last message of the protocol allows the tag to authenticate the reader.

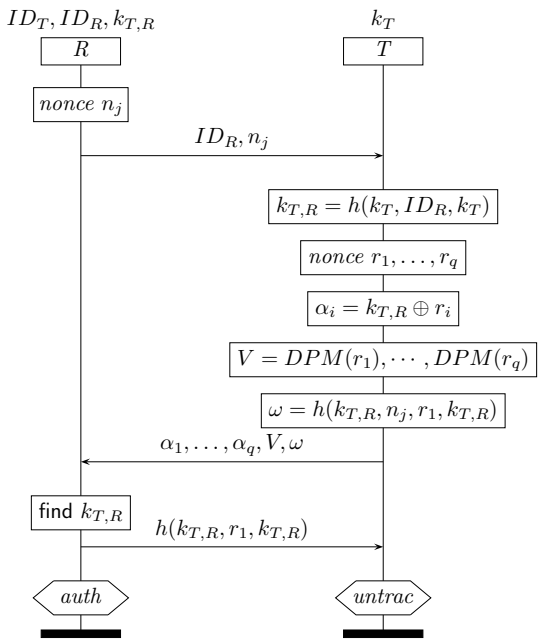

Fig. 2. Di Pietro-Molva protocol.

5.2 Untraceability Under Ideal Assumptions

We show that under the following assumptions the Di Pietro-Molva protocol indeed has the untraceability property with respect to RFID tags. We assume that the random numbers used in the protocol are perfect nonces, we restrict the number q of nonces used by the tag in the protocol to one, and we treat the DPM function appearing in V as a perfect hash function.

Theorem 2. *If the DPM function is a perfect hash function then the protocol depicted in Figure 2 is untraceable for $q = 1$.*

Proof. It can be easily verified with automated tools that $k_{T,R}$ and r_1 are secret. Let t be a trace with subtraces t_i^T and t_j^T for $i \neq j$. We proceed similarly to the proof of Theorem 1, and set $agent(t_i^T) = agent(t_j^T) = agent(t_i'^T) = Alice$ and $agent(t_j'^T) = Bob$.

Let t_j^T contain the event where $\alpha_1\#\theta, V\#\theta, \omega\#\theta$ is sent. We consider the map π which for any term R has the following properties

$$\begin{aligned}
&\pi(h(k_{Alice,R}, n_j\#\theta, r_1\#\theta, k_{Alice,R})) = h(k_{Bob,R}, n_j\#\theta, r_1\#\theta, k_{Bob,R})\\
&\pi(h(k_{Alice,R}, r_1\#\theta, k_{Alice,R})) \quad\quad = h(k_{Bob,R}, r_1\#\theta, k_{Bob,R})\\
&\pi(h(k_{Alice}, ID_R, k_{Alice})) \quad\quad\quad\quad = h(k_{Bob}, ID_R, k_{Bob})
\end{aligned}$$

and is equal to the identity map everywhere else. Note that according to the specification, $k_{Alice,R} = h(k_{Alice}, ID_R, k_{Alice})$ and $\alpha_1\#\theta = k_{Alice,R} \oplus r_1\#\theta$. By

the definition of reinterpretation, $\pi(k_{Alice,R} \oplus r_1\#\theta) = \pi(k_{Alice,R}) \oplus \pi(r_1\#\theta) = k_{Bob,R} \oplus r_1\#\theta$. For convenience, we set $\alpha_1'\#\theta = k_{Bob,R} \oplus r_1\#\theta$.

Let $t' = \pi(t)$. It follows from the construction that the map produces a valid run for Bob while keeping the reader's run valid. The only differences between the traces t and t' occur in messages containing the hashes $h(k_{Bob,R}, r_1\#\theta, k_{Bob,R})$ and $h(k_{Bob,R}, n_j\#\theta, r_1\#\theta, k_{Bob,R})$. Aside from Bob's run, the hashes and $\alpha_1\#\theta$ may be replayed by the adversary. Because $r_1\#\theta$ is generated by $T\#\theta$, no other run of T will accept $h(k_{Bob,R}, r_1\#\theta, k_{Bob,R})$ or $h(k_{Alice,R}, r_1\#\theta, k_{Alice,R})$. Similarly, since $k_{T,R}$ is secret and at most one run of R could have generated $n_j\#\theta$, at most one run of R accepts $\alpha_1\#\theta$, $\alpha_1'\#\theta$, $h(k_{Bob,R}, n_j\#\theta, r_1\#\theta, k_{Bob,R})$, and $h(k_{Alice,R}, n_j\#\theta, r_1\#\theta, k_{Alice,R})$.

Finally, $t_i^T = t_i'^T$, since $r_1\#\theta$ is generated by $T\#\theta$, thus $\neg L(t_i'^T, t_j'^T)$.

It remains to show that π is a reinterpretation under the adversary's knowledge. This follows from the fact that $r_1\#\theta$ is secret. □

Note that the assumption $q > 1$ would invalidate the untraceability proof, because t' would not necessarily be a valid trace anymore. In fact, for $q > 1$ an adversary may be able to determine that a tag is *not* identical to a previously observed tag. This insight can be exploited with an active as well as a passive attack. In an active attack, for each consecutive bit-triplet in α_2, the adversary would change one bit, during one execution. In such a case, the reader replies to the tag with a third message if and only if the two unchanged bits of the corresponding bit-triplet of the nonce r_2 are the same. Such an attack would, after several iterations, lead to the same information as the passive attack demonstrated in the following section.

5.3 Analysis of the DPM Function

We consider how much information about the tag is leaked through the DPM function and the resulting relation between α_i and $V[i]$. We first observe that for $(a, b, c), (x, y, z) \in \mathbb{F}_2^3$,

$$M(a + x, b, c) = ac + bc + ab + cx + bx$$

with analogous equations for $M(a, b + y, c)$ and $M(a, b, c + z)$. Furthermore, we have

$$M(a+x, b+y, c+z) = M(a+x, b, c)+M(a, b+y, c)+M(a, b, c+z)+M(x, y, z).$$

It follows that

$$M(a + x, b + y, c + z) = M(a, b, c) + M(x, y, z) + a(y + z) + b(x + z) + c(x + y)$$

which after reordering we write as

$$(y + z, x + z, x + y) \cdot \begin{pmatrix} a \\ b \\ c \end{pmatrix} = M(a+x, b+y, c+z)+M(a, b, c)+M(x, y, z). \quad (1)$$

We define, for convenience, the function

$$\begin{aligned}\text{cross} : \mathbb{F}_2^\ell &\to \mathbb{F}_2^\ell \\ (x_1, y_1, z_1, \ldots, x_{\ell/3}, y_{\ell/3}, z_{\ell/3}) &\mapsto (y_1 + z_1, x_1 + z_1, x_1 + y_1, \ldots, \\ &\qquad y_{\ell/3} + z_{\ell/3}, x_{\ell/3} + z_{\ell/3}, x_{\ell/3} + y_{\ell/3}).\end{aligned}$$

Note that $\text{cross}(r_1) + \text{cross}(r_2) = \text{cross}(r_1 + r_2) = \text{cross}(\alpha_1 + \alpha_2)$.

From equation (1) and the definition of $DPM(\cdot)$ we obtain the following identity in which the left-hand side is a product between the row vector $\text{cross}(r_1)$ and $k_{T,R}$ written as a column vector $k_{T,R}^T$.

$$\text{cross}(r_1) \cdot k_{T,R}^T = DPM(k_{T,R} + r_1) + DPM(k_{T,R}) + DPM(r_1) \tag{2}$$

and similarly

$$\text{cross}(r_2) \cdot k_{T,R}^T = DPM(k_{T,R} + r_2) + DPM(k_{T,R}) + DPM(r_2). \tag{3}$$

By adding up equations (2) and (3) we obtain

$$\text{cross}(\alpha_1 + \alpha_2) \cdot k_{T,R}^T = DPM(\alpha_1) + DPM(\alpha_2) + DPM(r_1) + DPM(r_2).$$

For $i = 2, \ldots, \ell + 1$, let the $\ell \times \ell$ matrix A be given by the row vectors $\text{cross}(\alpha_1 + \alpha_i)$ and let the column vector v be given by the entries $DPM(\alpha_1) + DPM(\alpha_i) + DPM(r_1) + DPM(r_i)$. Consider then the linear equation $Ax = v$, *viz.*

$$\begin{pmatrix} \text{cross}(\alpha_1 + \alpha_2) \\ \text{cross}(\alpha_1 + \alpha_3) \\ \vdots \\ \text{cross}(\alpha_1 + \alpha_\ell) \end{pmatrix} \cdot x = \begin{pmatrix} DPM(\alpha_1) + DPM(\alpha_2) + DPM(r_1) + DPM(r_2) \\ DPM(\alpha_1) + DPM(\alpha_3) + DPM(r_1) + DPM(r_3) \\ \vdots \\ DPM(\alpha_1) + DPM(\alpha_\ell) + DPM(r_1) + DPM(r_\ell) \end{pmatrix}$$

By construction, the vector $x = k_{T,R}^T$ is a solution of the equation and so is any vector of the form $k_{T,R}^T + y$, where y is in the null space of A. Thus, the null space of A in this equation can be considered the adversary's uncertainty about $k_{T,R}$. From the definition of the $\text{cross}(\cdot)$ function, it is easy to see that the null space of A contains the vectors

$$(1, 1, 1, 0, \ldots, 0)^T, (0, 0, 0, 1, 1, 1, 0, \ldots, 0)^T, \ldots, (0, \ldots, 0, 1, 1, 1)^T. \tag{4}$$

The following theorem states that the null space of A is actually spanned by these vectors whenever A is constructed from linearly independent vectors $\alpha_1, \ldots, \alpha_{\ell+1}$. Thus, the adversary can learn all bits of $k_{T,R}$ modulo the vectors in (4), that is, up to complements of $\ell/3$ consecutive bit-triplets.

Theorem 3. *If $\alpha_1, \ldots, \alpha_{\ell+1}$ are linearly independent, then the rank of A is $\frac{2}{3}\ell$.*

Proof. We know that the $\ell/3$ vectors listed in (4) are in the null space of A. Since they are linearly independent, the rank of A is at most $\frac{2}{3}\ell$.

Conversely, consider the matrix $\tilde{A}$ obtained from A by deleting every third column of A. $\tilde{A}$ can also be obtained from the matrix B consisting of the rows $\alpha_1 + \alpha_2, \ldots, \alpha_1 + \alpha_{\ell+1}$ as follows. We add every third column to the preceding two columns and swap those preceding two columns. We call the resulting matrix $\tilde{B}$. Clearly B and $\tilde{B}$ have the same rank. By deleting every third column of $\tilde{B}$, we obtain $\tilde{A}$. Since deletion of a column decreases the rank of the matrix by at most one and $\tilde{B}$ had full rank, it follows that the rank of $\tilde{A}$ is at least $\frac{2}{3}\ell$ and thus the rank of A is at least $\frac{2}{3}\ell$. □

5.4 Practical Considerations

The probability of a random $(n+1) \times n$ matrix over $\mathbb{F}_2$ to have rank n is greater than $1/2$. This follows from a simple computation along the lines of equation (1) in [44]. So we may over-estimate the expected number of random vectors needed to obtain ℓ linearly independent vectors to be 2ℓ. Hence after roughly $2\ell/q$ communications between an adversary and a tag, the adversary is able to compute a secret key of the tag up to complements of consecutive bit-triplets. We will now show that this information is very likely to distinguish one tag from almost all of the other tags in the system.

It follows from Theorem 3 that for each of the 2^ℓ possible secret keys, there are $2^{\ell/3}$ possible keys which cannot be distinguished from it solely based on the information contained in $\alpha_1, \ldots, \alpha_q$ and V. We may assume that the entries of the secret keys are uniformly randomly distributed since they are obtained by applying a cryptographic hash function. If we further assume that the number of tags ν in the system is small compared to 2^ℓ, then the probability that for a given tag, there are one or more tags indistinguishable by the above method is approximately $1-(1-\frac{1}{2^{2\ell/3}})^\nu$. If, as suggested by the authors, we use the values $\ell = 117$, $q = 2\log\nu$ and assume that there are $\nu = 2^{16}$ tags in the system, then the probability to find one or more tags which would be indistinguishable from a given tag is approximately $2.17 \cdot 10^{-19}$ and the number of communications necessary with the tag to be able to distinguish it with that probability would be 10. In fact, even the probability that there are two or more indistinguishable tags among 2^{16} tags is vanishingly small, namely $7.1 \cdot 10^{-13}$.

Finally, note that the same method reduces the complexity of computing the secret key of a tag to a brute force search of a space with $2^{\ell/3}$ elements, which for $\ell = 117$ is feasible.

5.5 Traceability

In this section we show that the Di Pietro-Molva protocol without idealizing assumptions on the DPM function is traceable by our definition.

We say that the lookup process is *efficient* if any authorized reader can uniquely identify a tag based on $\alpha_1, \ldots, \alpha_q$ and V.

Theorem 4. *Assuming that the lookup process is efficient, the protocol depicted in Figure 2 is traceable.*

Proof. Let t be a trace in which a reader *Ray* interacts twice with the same tag *Alice*. Let t_1^T and t_2^T be the two subtraces containing the send event of the tag, i.e. $agent(t_1^T) = agent(t_2^T) = Alice$. We need to show that there is no valid trace $t' \sim t$ such that $\neg L(t_1'^T, t_2'^T)$.

By observing t_1^T, t_2^T the adversary can compute $k_{Alice,Ray}$ up to a null space N_1 as shown in Section 5.3. We may assume that t is such that N_1 is the smallest possible null space shown in (4). Note that no other key $k_{T,Ray}$ is equal to $k_{Alice,Ray} + n$ for any $n \in N_1$ because the lookup process is efficient.

Let t' be any valid trace where $agent(t_1'^T) = agent(t_1^T) = Alice$, $agent(t_2'^T) = Bob$. By construction, we have $\neg L(t_1'^T, t_2'^T)$.

By observing $t_2'^T$, the adversary can compute $k_{Bob,Ray}$ up to a null space N_2 with $N_1 \subseteq N_2$ by minimality of N_1. There are no $n_1 \in N_1$, $n_2 \in N_2$ with $k_{Alice,Ray} + n_1 = k_{Bob,Ray} + n_2$ because the lookup process is efficient and $N_1 \subseteq N_2$.

Therefore the adversary can distinguish t from t'.

6 Conclusion

The main contribution of this paper is a definition of untraceability which can be used in formal verification of RFID protocols. We showed how to apply our definition by proving that the protocol in [12] indeed satisfies untraceability. We also demonstrated a weakness in the published protocol in [13], that we could exploit by using linear algebra. We proved that the protocol does not satisfy our definition of untraceability.

In the future, we would like to refine our notion of untraceability. Under the current definition, for a tag to be untraceable, it suffices to find one other tag which could have been present to produce the same trace. A strengthening of this definition is therefore desirable.

Several other refinements are conceivable. One such refinement concerns a weaker notion of untraceability that allows an adversary to recognize a tag between any two successful communications with a trusted RFID reader. Another refinement could be 'untraceability groups' defining the set of agents from which a particular agent cannot be distinguished. A third, slightly stronger notion of untraceability that should be defined properly is the notion of 'forward untraceability', stating that compromising a tag does not compromise its untraceability in past interactions.

A difficult open problem concerns the condition $\pi(f(m)) = f(\pi(m))$ in the definition of reinterpretation. This condition expresses that the application of the function f can be reinterpreted only to the extent its arguments can be reinterpreted under a given knowledge set M. If f is a cryptographic hash function, we know by the perfect cryptography assumption that $f(m)$ can be freely reinterpreted whenever m is not inferable from M. For other functions, however, the reinterpretation depends on the algebraic properties of f and then

$\pi(f(m)) = f(\pi(m))$ is only an under-approximation. Finding the correct condition for a given function f is, in general, non-trivial.

Finally, we plan to automate the process of verifying or finding attacks on untraceability. This leads to new challenges as can be seen in Section 5.3. Under the perfect cryptography assumption, large parts of the verification can be automated, but even state-of-the-art verification tools still struggle with algebraic operations in security protocols.

Acknowledgments

We are grateful to Hugo Jonker, Jun Pang, and the anonymous reviewers for their valuable comments which helped to improve this work.

References

1. Murray, C.J.: RFID tags: driving toward 5 cents. Design News (April, 24 2006)
2. Hoepman, J.H., Hubbers, E., Jacobs, B., Oostdijk, M., Wichers Schreur, R.: Crossing borders: Security and privacy issues of the european e-passport. In Yoshiura, H., Sakurai, K., Rannenberg, K., Murayama, Y., Kawamura, S.i., eds.: Advances in Information and Computer Security, First International Workshop on Security – IWSEC. Volume 4266 of Lecture Notes in Computer Science., Kyoto, Japan, Springer-Verlag (October 2006) 152–167
3. Yoshida, J.: Euro bank notes to embed RFID chips by 2005. EETimes (December 19, 2001)
4. Gilbert, A.: Major retailers to test 'smart shelves'. CNET (January 8, 2003)
5. O'Conner, M.C.: Gilette fuses RFID with product launch. RFID Journal (March, 27 2006)
6. Wong, F.L., Stajano, F.: Location privacy in Bluetooth. In: ESAS. (2005) 176–188
7. Jakobsson, M., Wetzel, S.: Security weaknesses in Bluetooth. In: CT-RSA. (2001) 176–191
8. Clark, J.A., Jacob, J.L.: A survey of authentication protocol literature. Technical Report 1.0 (1997)
9. Lowe, G.: Breaking and fixing the Needham-Schroeder public-key protocol using fdr. In: TACAS. (1996) 147–166
10. Garcia, F.D., Hasuo, I., Pieters, W., van Rossum, P.: Provable anonymity. In: FMSE. (2005) 63–72
11. Mauw, S., Verschuren, J., de Vink, E.: A formalization of anonymity and onion routing. In Samarati, P., Ryan, P., Gollmann, D., Molva, R., eds.: ESORICS'04. Volume 3193 of Lecture Notes in Computer Science., Sophia Antipolis, Springer-Verlag (2004) 109–124
12. Feldhofer, M., Dominikus, S., Wolkerstorfer, J.: Strong authentication for RFID systems using the AES algorithm. In: CHES. (2004) 357–370
13. Di Pietro, R., Molva, R.: Information confinement, privacy, and security in RFID systems. In: ESORICS. (2007) 187–202
14. Weis, S., Sarma, S., Rivest, R., Engels, D.: Security and privacy aspects of low-cost radio frequency identification systems. In Hutter, D., Müller, G., Stephan, W., Ullmann, M., eds.: International Conference on Security in Pervasive Computing – SPC 2003. Volume 2802 of Lecture Notes in Computer Science., Boppard, Germany, Springer-Verlag (March 2003) 454–469

15. Saito, J., Ryou, J.C., Sakurai, K.: Enhancing privacy of universal re-encryption scheme for RFID tags. In: EUC. (2004) 879–890
16. Garfinkel, S., Juels, A., Pappu, R.: RFID privacy: An overview of problems and proposed solutions. IEEE Security and Privacy **3**(3) (May-June 2005) 34–43
17. Juels, A.: RFID security and privacy: A research survey. Manuscript (September 2005)
18. Tan, C.C., Sheng, B., Li, Q.: Severless search and authentication protocols for RFID. In: International Conference on Pervasive Computing and Communications – PerCom 2007, New York, USA, IEEE, IEEE Computer Society Press (March 2007)
19. Seo, Y., Lee, H., Kim, K.: A scalable and untraceable authentication protocol for RFID. In: EUC Workshops. (2006) 252–261
20. Ohkubo, M., Suzuki, K., Kinoshita, S.: Cryptographic approach to "privacy-friendly" tags. In: RFID Privacy Workshop, MIT, MA, USA (November 2003)
21. Kang, J., Nyang, D.: RFID authentication protocol with strong resistance against traceability and denial of service attacks. In Molva, R., Tsudik, G., Westhoff, D., eds.: European Workshop on Security and Privacy in Ad hoc and Sensor Networks – ESAS'05. Volume 3813 of Lecture Notes in Computer Science., Visegrad, Hungary, Springer-Verlag (July 2005) 164–175
22. Dimitriou, T.: A secure and efficient RFID protocol that could make big brother (partially) obsolete. In: PerCom. (2006) 269–275
23. Choi, E.Y., Lee, S.M., Lee, D.H.: Efficient RFID authentication protocol for ubiquitous computing environment. In Enokido, T., Yan, L., Xiao, B., Kim, D., Dai, Y., Yang, L., eds.: International Workshop on Security in Ubiquitous Computing Systems – secubiq 2005. Volume 3823 of Lecture Notes in Computer Science., Nagasaki, Japan, Springer-Verlag (December 2005) 945–954
24. Nguyen Duc, D., Park, J., Lee, H., Kim, K.: Enhancing security of EPCGlobal Gen-2 RFID tag against traceability and cloning. In: Symposium on Cryptography and Information Security, Hiroshima, Japan (January 2006)
25. Piramuthu, S.: On existence proofs for multiple RFID tags. In: IEEE International Conference on Pervasive Services, Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing – SecPerU 2006, Lyon, France, IEEE, IEEE Computer Society Press (June 2006)
26. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J., Ribagorda, A.: RFID systems: A survey on security threats and proposed solutions. In: 11th IFIP International Conference on Personal Wireless Communications – PWC06. Volume 4217 of Lecture Notes in Computer Science., Springer-Verlag (September 2006) 159–170
27. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J., Ribagorda, A.: Cryptanalysis of a novel authentication protocol conforming to epc-c1g2 standard. (2007)
28. Martinez, S., Magda, V., Concepcio, R., Fransesc, G., Josep, M.: An elliptic curve and zero knowledge based forward secure RFID protocol (2007)
29. Alomair, B., Lazos, L., Poovendran, R.: Passive attacks on a class of authentication protocols for RFID. In: ICISC. (2007) 102–115
30. Nohl, K., Evans, D.: Quantifying information leakage in tree-based hash protocols. Technical Report UVA-CS-2006-20, University of Virginia, Department of Computer Science, Charlottesville, Virginia, USA (2006)
31. Tsudik, G.: YA-TRAP: Yet another trivial RFID authentication protocol. In: International Conference on Pervasive Computing and Communications – PerCom 2006, Pisa, Italy, IEEE, IEEE Computer Society Press (March 2006)

32. Ateniese, G., Camenisch, J., de Medeiros, B.: Untraceable RFID tags via insubvertible encryption. In: Conference on Computer and Communications Security – CCS'05, Alexandria, Virginia, USA, ACM, ACM Press (November 2005)
33. Avoine, G.: Adversary model for radio frequency identification. Technical Report LASEC-REPORT-2005-001, Swiss Federal Institute of Technology (EPFL), Security and Cryptography Laboratory (LASEC), Lausanne, Switzerland (September 2005)
34. Juels, A., Weis, S.A.: Defining strong privacy for RFID. In: PerCom Workshops. (2007) 342–347
35. Chatmon, C., van Le, T., Burmester, M.: Secure anonymous RFID authentication protocols. Technical Report TR-060112, Florida State University, Department of Computer Science, Tallahassee, Florida, USA (2006)
36. Tsudik, G.: A family of dunces: Trivial RFID identification and authentication protocols. Cryptology ePrint Archive, Report 2006/015 (2007)
37. Dimitriou, T.: A lightweight RFID protocol to protect against traceability and cloning attacks. In: Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm, Athens, Greece, IEEE (September 2005)
38. Lee, S., Asano, T., Kim, K.: RFID mutual authentication scheme based on synchronized secret information. In: Symposium on Cryptography and Information Security, Hiroshima, Japan (January 2006)
39. Steinbrecher, S., Köpsell, S.: Modelling unlinkability. In: Privacy Enhancing Technologies. (2003) 32–47
40. Huang, D.: On measuring anonymity for wireless mobile ad-hoc networks. Local Computer Networks, Proceedings 2006 31st IEEE Conference on (2006) 779–786
41. Schneider, S., Sidiropoulos, A.: CSP and anonymity. In: ESORICS. (1996) 198–218
42. Avoine, G., Oechslin, P.: RFID traceability: A multilayer problem. In Patrick, A., Yung, M., eds.: Financial Cryptography – FC'05. Volume 3570 of Lecture Notes in Computer Science., Roseau, The Commonwealth Of Dominica, IFCA, Springer-Verlag (February–March 2005) 125–140
43. Cremers, C., Mauw, S.: Operational semantics of security protocols. In Leue, S., Systä, T., eds.: Scenarios: Models, Algorithms and Tools (Dagstuhl 03371 post-seminar proceedings, September 7–12, 2003). Volume 3466 of LNCS. (2005) 66–89
44. Cooper, C.: On the rank of random matrices. Random Structures and Algorithms **16**(2) (2000)