

When Compromised Readers Meet RFID

Extended version

Gildas Avoine, Cédric Lauradoux, and Tania Martin

Université catholique de Louvain
Information Security Group
B-1348 Louvain-La-Neuve, Belgium

Abstract. RFID-based access control solutions for mobile environments, e.g. ticketing systems for sport events, commonly rely on readers that are not continuously connected to the back-end system. The readers must so be able to perform their tasks even in offline mode, what commonly requires the management by the readers of sensitive data.

We stress in this paper the problem of compromised readers and its impact in practice. We provide a thorough review of the existing authentication protocols faced to this constraint, and extend our analysis with the privacy property. We show that none of the reviewed protocols fits the required properties in case of compromised readers. We then design a sporadically-online solution that meets our expectations in terms of both security and privacy.

1 Introduction

Radio Frequency Identification (RFID) is getting more and more popular in access control, especially in mobile environments, such as sport events and public transportations. In such applications, the typical framework consists in customers who each holds an RFID ticket, that is a microcircuit with an antenna, called *tag*; some agents who carry an RFID *reader* to control the tags; and a centralized back-end system that manages data about the tickets and customers.

The readers in this context are mobile embedded devices that have an intermittent access to the back-end. For example, the ticket validator of a flying agent in the site of a sport event is connected only when the agent is back to the headquarter; or, the ticket validator in a bus has access to the back-end system only when the vehicle is parked in its lot, usually during the night. Consequently, readers must be able to authenticate offline the customers.

The tickets in this context are reasonably-costed RFID passive tags. They commonly offer some reasonable computation capabilities that allow them to use symmetric key cryptography.

The common RFID-friendly authentication protocols are based on either the ISO/IEC 9798 standard or some dedicated protocols [4, 20, 24]. They all consider an adversary model where the communication channel between the tag and the reader is basically not secure, and the tag can be tampered with. The security policies are thus designed following the assumption that the readers and any other infrastructure are secured. This is not enough to enforce strong security. Indeed, readers carry some sensitive information, and the security of the whole system is threatened if an adversary can compromise some of them. For instance, a PDA reader used to check

RFID tickets at Beijing Olympic Games [15] could have been stolen. Hence, it is critical for an access control system to be able to restore its integrity upon detection of such an event, without renewing all the delivered tickets.

In those applications, authentication protocols are used to prevent unauthorized entries. But they also require to enforce strong privacy policies to protect the customers. The challenge of designing authentication protocols consists in providing both security and privacy. Today, there is no authentication protocol resistant to compromised readers in terms of both security and privacy. The goal of this paper is to provide a practical and deployable solution that fits our expectations.

The contributions of the paper are as follows. We raise the problem of compromised readers, and we analyze the security of the existing protocols in this new scenario. We focus on candidates whose security without compromised readers is already well-established. All of them can be considered as a specific instance of the well-known ISO/IEC 9798 standard, from the basic challenge/response to more advanced protocols such as GPS, WIPR and TanSL. We show that none of these protocols preserves all the security properties under the assumption of a compromised reader, and especially tag privacy. We design a solution based on a symmetric-key challenge/response protocol. The secret key shared between a tag and a reader is computed on-the-fly by the tag at each authentication. This computation depends on an attack counter, the identity of the reader, and a long-term key shared by the tag and the back-end system. Our authentication protocol comes along with an update protocol to renew the tag's authentication key through the attack counter when a compromised reader is detected.

The paper is organized as follows: in Section 2, the different assumptions used to analyze the security of RFID protocols are detailed. Section 3 describes the authentication protocols defined by ISO/IEC 9798 and other protocols dedicated to RFID. The security analysis of these protocols is done in Section 4. Section 5 addresses the privacy issue, presents our protocol, and provides its security analysis. We conclude the paper in Section 6.

2 Security Analysis in RFID

In this section, we discuss the architecture, the threat models chosen to analyze RFID authentication protocols, and the different classes of attacks.

2.1 Architecture

We consider an RFID system composed of a trusted back-end, a set of readers, and m tags. The readers can be sealed for their own protection, and are sporadically connected to the back-end through a secure channel. We also assume the use of low-cost tags: they have a low gate complexity (≈ 4000 GE) and tamper-resistance is limited. The latter assumption implies that a unique secret per tag is needed to prevent a large-scale attack where an adversary recovers the secret of all the tags by breaking only one of them.

2.2 Threat Models

We examine two threat models in which an adversary can have different power levels. They mainly differ on the assumption made on the vulnerability of the readers. These two scenarios are described as follows.

Scenario 1. The RFID system does not have any compromised reader. This threat model is the most widespread in the RFID body of literature. The adversary can be either passive or active. She can eavesdrop, delete, swap, or alter the messages sent between the readers and the tags. She can inject her own messages. She has also the possibility to analyze the tag or reader behavior (e.g. timing attack).

Scenario 2. An adversary can compromise some readers. Thus, the adversary is as strong as in Scenario 1, but she can additionally obtain all the information stored in these readers which makes her extremely powerful. We assume that the back-end is able to detect this event: a physical alteration of the reader can be seen (its seal broken or it did communicate with the back-end for a long time). This scenario is particularly realistic in applications where the adversary can get access to the readers.

2.3 Attacks

An adversary can use different strategies to undermine an RFID system. The classes of attacks go from tag impersonation to tag privacy. The aim of a well-suitable RFID authentication protocol is to thwart these attacks, especially considering our two threat models.

Tag Impersonation. In this case, an adversary would like to impersonate a legitimate tag T to fool a legitimate reader. The latter should be convinced to interact with an authorized tag. Various methods exist to carry out this attack: a tag can be cloned or an adversary can replay messages previously sent by T . The adversary's chances of success mainly depends on how much information she has.

Denial of Service. In RFID systems, a denial of service (DoS) is when an adversary wants to make the tag unusable by any means. Here, we only consider DoS attacks related to the authentication protocol. The adversary can only exploit the protocol to mount a DoS.

Tag Privacy vs. Reader Complexity. The tag privacy refers to the protection of the owner personal data. This security issue is based on two fundamental properties: the information leakage and the malicious traceability of a tag [3]. The information leakage concerns data exchanged during an RFID communication, which can be inherent to the environment or to the tag particularly. For instance, when Alice validates her transit pass, this latter can send in the clear the type of subscription: one-year or 10-travel ticket. The adversary learns some information about Alice. The malicious traceability arises when an adversary can correlate messages from a given tag over different protocol executions. In mass transportation, if Alice's pass always sends a unique identifier when she checks in, the adversary is able to recognize Alice anywhere. The problem of compromised readers is very critical for tag's privacy: an adversary may be able to track all the tags from the data stolen in a reader.

An efficient protocol preserving the tag privacy should have a low computation complexity for the reader, with respect to the number of tags. Unfortunately, there exist a duality between tag privacy and reader complexity. As explained in [2], there exists in our framework no symmetric-key based protocol that ensures privacy with a reader complexity better than $O(m)$, where m is the number of tags in the system.

3 Available Authentication Protocols

In this section, the content of the ISO/IEC 9798 standard from part 2 to 5 is described. It covers the well-known authentication protocols based on the classical cryptographic primitives. Moreover, three RFID-dedicated protocols are analyzed: GPS, WIPR and TanSL. GPS and WIPR are respectively based on zero-knowledge authentication and on public-key encryption. It has been shown that these two protocols can meet the hardware constraint of RFID. GPS is also mentioned in ISO/IEC 9798. Finally, we study the TanSL protocol because of its interesting secret recomputation mechanism.

3.1 ISO/IEC 9798

ISO/IEC 9798 [14] is currently the international standard for entity authentication and it is widely used in RFID. Parts 2 to 4 of the standard provides four authentication protocols that are respectively based on symmetric encryption algorithms, digital signature, and cryptographic hash functions. Each part describes three mechanisms for achieving authentication: unilateral authentication with timestamps, unilateral authentication with random numbers, and mutual authentication with random numbers. The unilateral authentication with random numbers, designated as Mechanism 2 in the standard, is discussed here. Basically, the reader sends to the tag a nonce n_R called the challenge and the tag responds to this challenge. To do so, it can use:

- a symmetric encryption function (ISO/IEC 9798-2),
- a signature scheme (ISO/IEC 9798-3), or
- a cryptographic hash function (ISO/IEC 9798-4).

The protocol described in Fig. 1 corresponds to a challenge/response based on symmetric-key (SK) encryption and Fig. 2 represents a challenge/response based on a signature scheme. To simplify the notations, the figures and protocols represent the readers data for one tag T . Id_T, s (resp. k_T and p_T), n_R and n_T denote the identifier of the tag, the symmetric key shared by the tag and the reader (resp. the private and public keys of the tag), the nonce picked by the reader, and finally the nonce picked by the tag. The functions E and S respectively represent encryption and signature algorithms.

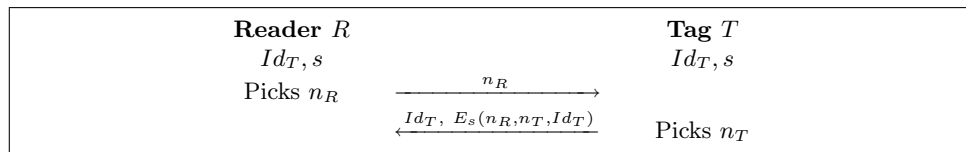


Fig. 1. A SK-based challenge/response protocol.

Part 5 of the standard provides several mechanisms that are respectively based on integer factorization, discrete logarithms with respect to prime or composite numbers, and asymmetric encryption. Such mechanisms can be zero-knowledge (ZK) proofs, such as Fiat-Shamir [10] protocol.

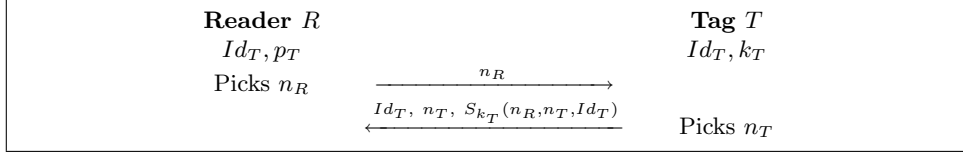


Fig. 2. A signature-based authentication protocol.

3.2 Protocols Dedicated to RFID

GPS, WIPR and TanSL protocols are reminded in this section.

GPS Protocol. The GPS authentication protocol [4] is an interactive zero-knowledge authentication protocol initially proposed by Girault, Poupard, and Stern. It provides provable security based on the composite discrete logarithm problem. It also combines short transmissions and minimal on-line computation, using precomputed “coupons”. This protocol has been selected in the NESSIE portfolio [18] and it is mentioned in the ISO/IEC 9798-5 Clause 8 as a reference. Throughout the paper, we will refer GPS as this variant “with coupons”.

The parameters used in this protocol are the following:

- S, B, A are public integers, where $|S| \approx 180$, $|B| = 32$ and $|A| = |S| + |B| + 80$,
- $n = p \times q$ is a public composite modulus, where p and q are secret primes, $|n| = 1024$, $|p| = |q| = 512$,
- g is an element of \mathbb{Z}_n^* ,
- $\Phi = (B - 1) \times (S - 1)$,
- $s \in [0, S[$ and $I = g^{-s} \pmod n$,
- a coupon i is a couple $(r_i, x_i = g^{r_i} \pmod n)$, where $r_i \in [0, A[$ is a random number.

At the beginning, the tag T has a unique identifier Id_T , a unique pair of keys (s is the private one and I is the public one) and a set of *coupons* computed by a higher trusted entity (the back-end). Every reader knows the tag’s identifier and public key.

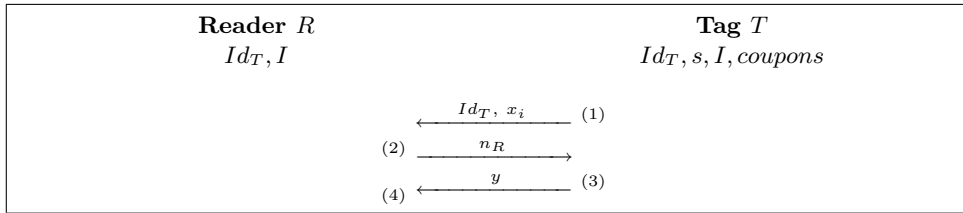


Fig. 3. GPS protocol.

GPS, depicted in Fig. 3, works as follows.

- (1) The tag T chooses a coupon (r_i, x_i) , and sends its identifier Id_T and x_i to the reader R .
- (2) The reader answers a challenge n_R randomly chosen in the interval $[0, B[$.
- (3) The tag computes $y = r_i + n_R \times s$, and sends y to the reader.
- (4) The reader checks if:
 - $g^y \times I^{n_R} \pmod n = x_i$
 - $y \in [0, A + \Phi[$

WIPR Protocol. This is a variant of the well-known Rabin cryptosystem [22]. WIPR was proposed by Oren and Feldhofer in [20], and improved by the same authors in [21]. It is based on early works of Shamir [23] and Naccache [17]. Recently, Wu and Stinson [25] also presented a version of WIPR with a proven security. We describe here the original version of WIPR presented at RFIDSec 2008.

During the system set up, a large number $n = p \times q$ is chosen, $|n| = 1024$ bits, p and q are two prime numbers. n is public while p and q are kept secret by the reader. α and β are two security parameters, such that $\alpha = 128$ and $\beta = 80$. Each tag T has a unique secret identifier Id_T only known by the readers of the system.

As depicted in Fig. 4, WIPR is a challenge/response protocol that works as follows:

- (1) The reader R sends a challenge n_R to the tag T , where $|n_R| = \alpha$.
- (2) The tag picks two random numbers $n_{T,1}$ and $n_{T,2}$, where $|n_{T,1}| = |n| - \alpha - |Id_T|$ and $|n_{T,2}| = |n| + \beta$. Then it generates a plaintext $P = \text{BYTE_MIX}(n_R || n_{T,1} || Id_T)$ where $\text{BYTE_MIX}()$ is a classic byte-interleaving operation. Finally, T sends to R the encryption $A = P^2 + n_{T,2} \times n$.
- (3) The reader decrypts it with its private key (p, q) . Like in Rabin cryptosystem, there are 4 plaintext candidates. Then R checks if one of these contains its challenge n_R . If so, then R also recovers Id_T .

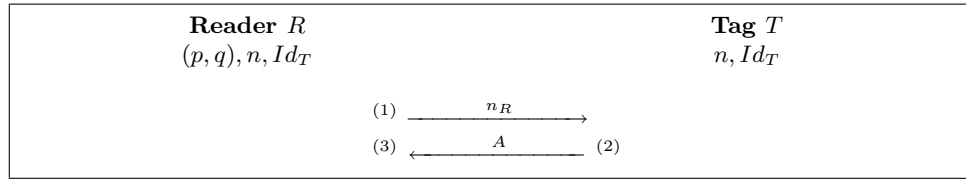


Fig. 4. WIPR protocol.

TanSL Protocol. TanSL protocol denotes the first protocol presented in [24] by Tan, Sheng, and Li at PerCom 2007. It is a challenge/response protocol based on a single hash operation. The secret shared between the reader and the tag is computed by the tag at each authentication.

At the initialization of the system, every tag T has a unique identifier Id_T and a secret t_T . Every reader R has an identifier Id_R and a list L containing all the tags' identifiers and a hash value of their secret concatenated with the reader's identifier: for every tag T , $L = [Id_T : h(Id_R || t_T)]$, where h is a cryptographic hash function.

As represented in Fig. 5, TanSL works as follows.

- (1) The reader R sends a request to the tag T .
- (2) The tag T answers a random number n_T .
- (3) The reader sends its identifier Id_R and a random number n_R .
- (4) The tag computes a hash $H = h(h(Id_R || t_T) || n_R || n_T)$, where $\ell := |H|$ (e.g. 160 bits for SHA-1).

Hb and He represent the b first bits and the $\ell - b$ last bits of H , respectively. That is $H = Hb || He$, $|Hb| = b$ and $|He| = \ell - b$.

Then T sends Hb and a question $ques_R = (ques_R^1, ques_R^2, \dots, ques_R^k)$, which represents k randomly chosen bit positions from He (notice that $k \leq \frac{\ell - b}{2}$).

- (5) For every entry T in L , the reader computes $H' = h(h(Id_R || t_T) || n_R || n_T)$ with $H' = Hb' || He'$, and checks if Hb matches with Hb' :
- If so and $k \leq \frac{\ell-b}{2}$, it sends to T the answer ans_R to the question $ques_R$. Thus, ans_R represents the actual bits in positions $ques_R^1, ques_R^2, \dots, ques_R^k$ of He' .
 - Else it sends $ans_R = rand$ where $rand$ is a random bit string of length k .
- In turn, it sends a question $ques_T = (ques_T^1, ques_T^2, \dots, ques_T^k)$, built like $ques_R$.
- (6) The tag T checks if ans_R is correct:
- If so and $\{\forall x, y, ques_R^x \neq ques_T^y\}$, it sends to R the answer ans_T to the question $ques_T$.
 - Else it sends $ans_T = rand$.
- (7) The reader R verifies the answer ans_T .

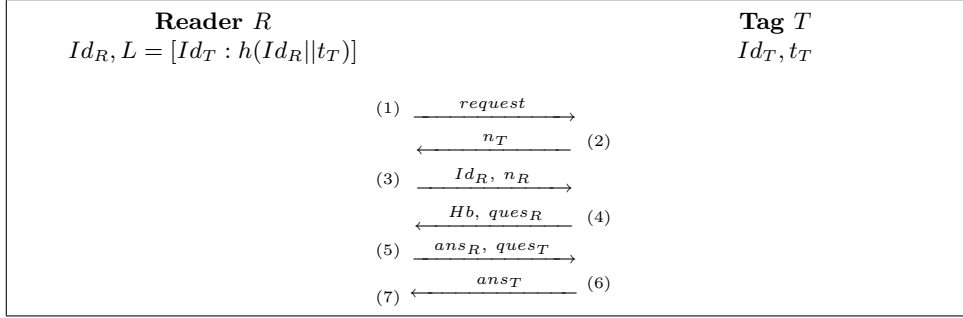


Fig. 5. TanSL protocol.

4 Security Analysis

We now study the security of all the previous protocols in the different scenarios defined in Section 2.

4.1 Tag Impersonation

As authentication protocols are designed by nature to be secure in the context of *Scenario 1*, we only focus in Section 4.1 on *Scenario 2*.

Scenario 2. For SK-based challenge/response protocols, once the adversary compromised a reader, she knows all the secrets stored by the reader. She is so able to impersonate any tag.

For signature schemes and zero-knowledge protocols (including GPS), the private key used to answer to the challenges is only known by the tag. Thus even if the adversary compromises a reader, she does not know the tags' private keys. She cannot impersonate them.

Regarding WIPR, an adversary who compromised a reader R knows its public and private keys (n and (p, q)) and the tags' identifiers. The result is that she will be able to impersonate any tag to every reader.

For the TanSL protocol, an adversary can obtain from a compromised reader R its identifier Id_R and the list L containing all $(Id_T : h(Id_R || t_T))$, for every tag T . The adversary will not be able to impersonate a tag T in front of any other non-compromised reader R' . Indeed, she does not know the tag's secret t_T , thus she is not able to compute the symmetric key $h(Id_{R'} || t_T)$ shared between R' and T .

4.2 Denial of Service

The problem of denial of service remains the same for either *Scenario 1* or *Scenario 2*. When an authentication protocol does not modify the content of the tags, no DoS attack is possible in both scenarios. Therefore, all the protocols presented in this paper, except GPS, are resistant to such an attack.

As GPS uses coupons, a DoS attack is feasible. Actually, a tag can perform a limited number of authentications, i.e., one authentication consumes one coupon. The number of coupons available is bounded by the tag memory. As there is no reader authentication, an adversary can ask many authentications to a tag T in a very short time. She can exhaust all the tag's coupons almost instantaneously without T 's agreement. T will no longer be able to successfully perform the protocol. If GPS is used "without coupons", there is no DoS attack. However, it increases the number of computations for the tag. This version of GPS has not been considered in [4] for lightweight applications such as RFID.

4.3 Comparison

In practical terms, a first comparison is necessary to find out which protocols can be implemented in wired logic for low-cost RFID passive tags.

For typical challenge/response protocols, we compare in Table 1 various hardware implementations of symmetric, asymmetric encryptions and hash functions available in the literature. As it is quoted, symmetric encryption has good results: AES is efficient in such tags, so is PRESENT. For asymmetric encryption, NTRUEncrypt has been considered as a great candidate in [1, 12], however the parameters of the system achieving good security are not yet known [11]. Also it is commonly admitted that an RSA encryption core cannot fit in low-cost passive tags [12]. The same observation can be made for classical elliptic curves cryptosystems [13]. Actually, the major problem of cryptosystem based on integer factorization or discrete logarithm problems, *e.g.* signature schemes or zero-knowledge protocols (included GPS "without coupons"), is that their implementation is too costly to fit in less than about 4000 GE. An exception is GPS: it is suitable for low-cost tags, since it requires only 1642 GE [16]. However, the coupons limit the number of authentications. WIPR is a great candidate for asymmetric encryption, as its chip area is reasonable.

In the case of ticketing applications, we consider that the AES encryption time is the reference. WIPR is not fast enough in comparison to the AES ($\times 66$ slower), whereas GPS is faster than the AES ($\times 2.5$ faster). TanSL and SK-based challenge/response are the only protocols relying on a common primitive (SHA-1, AES, PRESENT-80) that achieves a reasonable speed.

We also need a protocol which achieves all the security properties in addition to lightweight implementation and efficiency. For *Scenario 1*, TanSL and SK-based challenge/response provide all the security features. GPS is vulnerable to DoS attacks in any of the scenarios. However, SK-based challenge/response protocols do not prevent from tag impersonation in *Scenario 2*. TanSL and GPS are the most attractive solutions in our context. Table 2 is an overview of the properties studied in this section for the protocols presented till to now.

Table 1. Comparison of different cryptosystem implementations.

Type	Algorithm	Frequency [kHz]	Chip Area [GE]	Clock Cycle
Symmetric	AES-128 [8]	100	3400	1032
	PRESENT-80 [5]	100	1570	32
Asymmetric	WIPR [21]	100	4682	66048
	ECC-163 [13]	100	14976	296000
	NTRUEncrypt [1]	500	3000	28390
Hash	SHA-1 [19]	100	5527	344
	SHA-256 [6, 9]	100	10868	1128
ZK	GPS [16]	100	1642	401

Table 2. Comparison of the presented protocols.

Scenario	SK Ch./Re.		Sign. - ZK		GPS		WIPR		TanSL	
	1	2	1	2	1	2	1	2	1	2
Implementation	+		-		+		+		+	
Efficiency	+		-		+		-		+	
No Tag Impersonation	+	-	+	+	+	+	+	-	+	+
No Denial of Service	+	+	+	+	-	-	+	+	+	+

5 The Problem of Privacy with Compromised Readers

We focus in this section on the problem of tag privacy when a system has compromised readers. We first show that currently none of the remaining candidates (TanSL and GPS) ensures privacy in such a context. Then, we propose a new authentication protocol with a key update to preserve tag privacy in *Scenario 2*. The security recovery is done using an update procedure when the legitimate readers get connected to the back-end after the detection of a compromised reader.

5.1 Privacy Analysis of the Candidates

Scenario 1. The TanSL protocol provides tag privacy because the tag never sends to the reader its identifier in the clear and, more generally, an adversary cannot distinguish the tag’s response from a random value. However, the reader does not know which tag it is communicating with and must so carry out an exhaustive search through the list L . Consequently, the reader complexity is $O(m)$ where m is the number of tags in the system. The duality privacy vs. reader complexity allows nevertheless to reduce to $O(1)$ the complexity if the privacy is abandoned.

GPS does not provide by design tag privacy because the tag public key I is required by the reader in order to complete the authentication. Since this key I is known by everyone, an adversary is free to perform herself an authentication on a tag. One may think that the duality tag privacy vs. reader complexity can also be applied here in order to get a protocol ensuring privacy at the cost of $O(m)$

computations. This is actually more tricky in this case. In fact, even if I is kept secret by the system and Id_T is not sent in the clear, Bringer, Chabanne, and Icart explain in [7] that an adversary is able to recognize a tag from another by eavesdropping two communications. Indeed, by observing two GPS executions, an adversary cannot recover I but she can determine whether or not the same I has been used in the two executions. Thus, classic GPS does not provide tag privacy, whether the tag public key (and identifier) is disclosed or not.

It is important to notice that the authors of GPS [4] proposed an improvement: the coupons can be pairs of $(r_i, x_i = h(g^{r_i} \bmod n))$, where h is a cryptographic hash function; in that case, the reader has to verify if $h(g^y \times I^{n_R} \bmod n) = x_i$. Bringer *et al.*'s attack does not apply to this modification, initially introduced to reduce the size of the coupons. In this case, the reader complexity is $O(m)$.

We propose below a way to reduce the above-mentioned complexity: the tag can send a random session identifier Id_T^i used once and only known by the reader and the tag. Therefore the tag cannot be recognized by an adversary during the authentication, but the reader who knows in advance the session identifiers can directly identify the public key I to use from Id_T^i . The reader complexity is so $O(1)$. The session identifiers can be integrated into the coupons, such that each coupon is a triple $(Id_T^i, r_i, x_i = h(g^{r_i} \bmod n))$. We will call this protocol the *modified GPS* (mGPS).

Scenario 2. In all the previous protocols, there is no reader's authentication: consequently the tag cannot distinguish a legitimate reader from a compromised one. Moreover, if the system detects a compromised reader R , there is no mechanism to take this information into account better than changing physically all the tags and readers.

For TanSL, the adversary knows R 's identifier Id_R and the list L that contains all the information required to allow R to communicate with the tag T . The adversary can track every tag, the TanSL protocol does not supply tag privacy anymore.

We also notice that, if tag privacy is not provided in *Scenario 1*, it can neither be achieved in *Scenario 2*. GPS has an unchanged security profile. Concerning mGPS, the adversary knows all the hidden tags' public keys I . Thus, she is able to identify which tag is involved in every communication. mGPS no longer provides tag privacy.

None of these protocols provides tag privacy in *Scenario 2*.

5.2 Solving the Privacy Issue

In a system with mobile readers, it is more attractive for an adversary to compromise a reader than a tag. We propose a new challenge/response authentication protocol to handle this threat. It is based on a symmetric encryption algorithm to achieve reasonable chip area and efficiency. Our protocol has two steps. First, the secret key shared by the tag and the reader is computed on-the-fly by the tag, as in TanSL. This mechanism is enhanced with an attack counter to allow an update of the system. Second, the tag sends a classical answer in a challenge/response protocol using the previous key. We describe our protocol. Then, we show how this solution supplies tag privacy in the context of compromised readers.

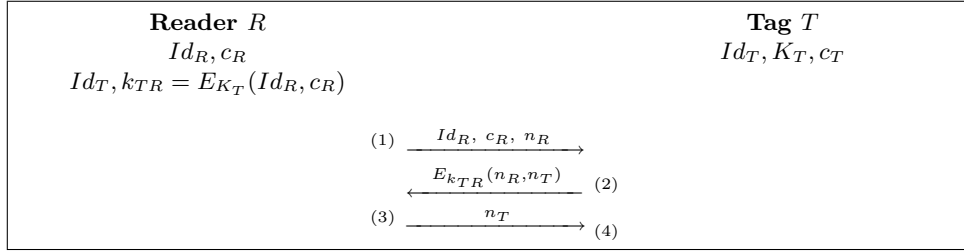


Fig. 6. Authentication protocol.

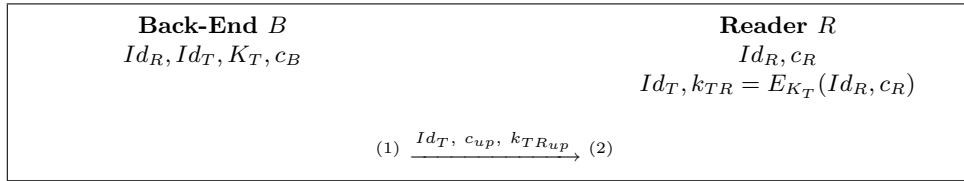


Fig. 7. Key update protocol.

Initialization. When the system is set up, each tag T is assigned with the following values:

- a unique identifier Id_T ,
- a long-term key K_T ,
- three counters c_B , c_R and c_T , initially synchronized and all equal to zero.

And during this set up, each reader R is assigned with the following values:

- a unique identifier Id_R ,
- for every tag T , its identifier and an encryption of its secret: $Id_T, k_{TR} = E_{K_T}(Id_R, c_R)$.

B stores Id_R, Id_T, K_T and c_B .

R stores Id_R, c_R, Id_T and $k_{TR} = E_{K_T}(Id_R, c_R)$.

T stores Id_T, K_T, c_T .

Authentication. The authentication protocol consists of four steps (see Fig. 6):

- (1) The reader sends to the tag its identifier Id_R , the counter c_R and a nonce n_R .
- (2) The tag checks the value c_R it receives:
 - If $c_R \geq c_T$, it computes the key $k_{TR} = E_{K_T}(Id_R, c_R)$. Then, it picks a nonce n_T and answers the encryption $E_{k_{TR}}(n_R, n_T)$ to the reader.
 - If $c_R < c_T$, the protocol aborts.
- (3) The reader decrypts the received message with the symmetric key k_{TR} , and verifies the value n_R . Then, it sends to the tag the recovered value n_T .
- (4) T checks the validity of n_T : if so and $c_R > c_T$, it updates c_T to the value c_R ($c_T \leftarrow c_R$).

Key Update. The update protocol is carried out when a compromised reader is detected (see Fig. 7). We consider that all the readers are synchronized at the same time:

- (1) The back-end increments c_B and associates this new value to c_{up} .
For every reader R and every tag T , it generates a new key $k_{TR_{up}} = E_{K_T}(Id_R, c_{up})$.
Finally, for every tag T , it sends Id_T , c_{up} , and $k_{TR_{up}}$ to every reader R .
- (2) Each reader R updates its array as follows:
 - $c_R \leftarrow c_{up}$.
 - $k_{TR} \leftarrow k_{TR_{up}}$, for every tag T .

Our protocol is based on a single cryptographic operation (symmetric encryption) and it consists in two computations for the tag. The first computation is done to generate the secret key k_{TR} used between the tag T and the reader R . Usually in a classical challenge/response protocol, each tag is assigned with a fixed key, which is used to communicate with every reader. In our protocol, the tag computes on-the-fly the key to interact with the reader R . This key k_{TR} is unique for T and R , since it is the encryption of the reader’s identity Id_R and the counter c_R with the tag long-term key K_T (shared by the back-end and the tag T). The counter c_R represents the number of updates achieved by the system. At the beginning of the authentication, the reader sends this counter alongside with its identity Id_R and a nonce n_R to indicate the current state to the tag. The second computation corresponds to the tag’s answer in the challenge/response protocol. The nonces n_R and n_T are encrypted with the key k_{TR} . When the reader decrypts successfully this latter message, it replies n_T as an acknowledgment.

We now analyze the security of our protocol against the different classes of attacks.

Our protocol introduces a key update for the reader and tag update. The tag update corresponds to the c_T update. These events have an impact on *Scenario 2*. Let consider \mathcal{E} a set of m tags, where at least one tag has already been updated. The set $\mathcal{S} \subset \mathcal{E}$ contains all the non updated tags. We assume that the tag $T \in \mathcal{S}$ recovers its security after it has been updated. We define \mathcal{P} the period spent between the detection of the attack and T ’s update.

Tag Impersonation. Our protocol inherits from the positive security properties of the classical challenge/response protocol in *Scenario 1*, that is it prevents from tag impersonation. Indeed, the only difference is that the long-term key stored by the tag in a classical challenge/response protocol is replaced in our solution by a key computed on-the-fly by the tag.

The additional values c_R and n_T exchanged during the authentication protocol do not reveal any key material neither in *Scenario 1*, nor in *Scenario 2*: n_T is a random number, and c_R is the system state known by anybody.

For *Scenario 2*, an adversary cannot impersonate a tag in front of all the non compromised readers, since the keys are computed like for TanSL.

Denial of Service. The only value modified into the tag T is the counter c_T in *Scenario 2*. An adversary can try to desynchronize the tag by modifying c_T ,

impersonating a legitimate reader with a fake update. She cannot forge a fake key $k'_{TR} = E_{K_T}(Id_R, c'_R)$ corresponding to a fake counter $c'_R > c_T$, because she does not know K_T . Therefore, the tag will not accept to update c_T to a fake $c'_R > c_T$, since the adversary will not be able to answer correctly at step (3) of the authentication protocol (see Fig. 6). Our solution is so resistant to DoS attacks.

Tag Privacy vs. Reader Complexity. Our protocol provides tag privacy in *Scenario 1*: no tag's identifier sent in the clear and answers are randomized. But the reader has no clue on the key it should use to check the tags' responses: the reader complexity is so $O(m)$. An improvement for the reader complexity consists in using session identifiers as in our modification of GPS (mGPS). In this solution, the tag will store a given number of identifiers known by the reader and used once per authentication. The reader complexity is $O(1)$. However, there is only a limited number of authentications: a DoS attack is possible.

In *Scenario 2*, the adversary knows T 's key $k_{TR} = E_{K_T}(Id_R, c_R)$ during \mathcal{P} . Thus, she can track T . After \mathcal{P} , the adversary does not know the new key $k_{TR_{up}} = E_{K_T}(Id_R, c_{up})$ used for the further T 's authentications. Thus, she will not be able to decrypt correctly the T 's answers and to track T anymore. Such an incorrect decryption by the adversary represents an information leakage: she knows when T has been updated. But this cannot be avoided, since the adversary knows that T will be updated at one time or another. This is the only leakage. However, the adversary is no more able to distinguish any $T \in \mathcal{E} \setminus \mathcal{S}$, since $|\mathcal{E} \setminus \mathcal{S}| > 1$ after the period \mathcal{P} . Therefore, the tag privacy is restored. The reader complexity is still $O(m)$.

It should be noticed that the number of messages exchanged in our protocol is variable. Indeed, a legitimate tag answers or not to a reader depending on the attack counter c_R sent in the clear. However, a legitimate tag always answers to an updated legitimate reader. A variation is possible only between a legitimate tag and a rogue reader. If the tag answers ($c_R \geq c_T$), the rogue reader can use the secrets stolen from the compromised reader to try to decrypt the tag's answer. The adversary knows if the tag has been updated or not. If the tag does not answer ($c_R < c_T$), the rogue reader learns that the tag has been updated. Thus, we see that any answer of the tag can be exploited to know if the tag has been updated or not. This is the situation described in the previous paragraph.

Table 3 provides a comparison between GPS, GPS with our improvement (mGPS), TanSL, and our protocol.

Table 3. Comparison of our protocol.

Scenario	GPS		mGPS		TanSL		Our protocol	
	1	2	1	2	1	2	1	2
No Tag Impersonation	+	+	+	+	+	+	+	+
No Denial of Service	-	-	-	-	+	+	+	+
Tag Privacy vs. Reader Complexity	-	-	+	-	+	-	+	+
	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(m)$	$O(m)$	$O(m)$	$O(m)$

6 Conclusion

A new issue in RFID systems is presented: the threat of compromised readers. Such an attack is very likely to occur in access control solutions for mobile environments, e.g. ticketing systems for sport events. We present a state of the art of several existing protocols. Our security and practical analysis of these available authentication protocols showed their weaknesses in this context.

We proposed a solution based on a symmetric-key challenge/response authentication protocol with key update. It can face the problem of compromised readers while preventing from the security properties we defined: tag impersonation and DoS attacks. Moreover, our solution provides tag privacy in both cases w/o compromised readers. As it is based on symmetric encryption, it also carries out low chip area and efficiency, such that it is suitable for reasonably-costed RFID tags.

We have made the hypothesis that the period \mathcal{P}' during which the system has updated a single tag T cannot be exploited by an adversary to track this tag. During \mathcal{P}' , the T 's answers are the only ones which cannot be decrypted correctly by the adversary. In applications such as ticketing, it is very uncommon to have a single updated user during a long period: our hypothesis is fair in this context. If this hypothesis cannot be fulfilled, new solutions are required to preserve privacy. A future work will be to solve this open problem.

References

1. A. C. Atici, L. Batina, J. Fan, I. Verbauwhede, and S. B. O. Yalcin. Low-cost Implementations of NTRU for Pervasive Security. In *International Conference on Application-Specific Systems, Architectures and Processors – ASAP 2008*, pages 79–84, Leuven, Belgium, July 2008.
2. G. Avoine, E. Dysli, and P. Oechslin. Reducing Time Complexity in RFID Systems. In *Selected Areas in Cryptography – SAC 2005*, volume 3897 of *Lecture Notes in Computer Science*, pages 291–306, Kingston, Canada, August 2005. Springer-Verlag.
3. G. Avoine and P. Oechslin. RFID Traceability: A Multilayer Problem. In A. Patrick and M. Yung, editors, *Financial Cryptography – FC'05*, volume 3570 of *Lecture Notes in Computer Science*, pages 125–140, Roseau, The Commonwealth Of Dominica, February–March 2005. IFCA, Springer-Verlag.
4. O. Baudron, F. Boudot, P. Bourel, E. Bresson, J. Corbel, L. Frisch, H. Gilbert, M. Girault, L. Goubin, J.-F. Misarsky, P. Nguyen, J. Patarin, D. Pointcheval, G. Poupard, J. Stern, and J. Traoré. GPS - An Asymmetric Identification Scheme for on the Fly Authentication of Low Cost Smart Cards. A proposal to NESSIE, 2001. <https://www.cosic.esat.kuleuven.be/nessie/updatedPhase2Specs/gps/GPS-Bv2.pdf>.
5. A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In *Workshop on Cryptographic Hardware and Embedded Systems – CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466, Vienna, Austria, September 2007. Springer-Verlag.
6. A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, and Y. Seurin. Hash Functions and RFID Tags : Mind The Gap. In *Workshop on Cryptographic Hardware and Embedded Systems – CHES 2008*, volume 5154 of *Lecture Notes in Computer Science*, pages 283–299, Washington, DC, USA, August 2008. Springer-Verlag.
7. J. Bringer, H. Chabanne, and T. Icart. Efficient Zero-Knowledge Identification Schemes which respect Privacy. In *ACM Symposium on Information, Computer and Communication Security – ASIACCS'09*, pages 195–205, Sydney, Australia, March 2009. ACM.

8. M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. Strong Authentication for RFID Systems using the AES Algorithm. In *Workshop on Cryptographic Hardware and Embedded Systems – CHES 2004*, volume 3156 of *Lecture Notes in Computer Science*, pages 357–370, Boston, Massachusetts, USA, August 2004. Springer-Verlag.
9. M. Feldhofer and C. Rechberger. A Case Against Currently Used Hash Functions in RFID Protocols. In *Workshop On the Move to Meaningful Internet Systems – OTM 2006*, volume 4277 of *Lecture Notes in Computer Science*, pages 372–381, Montpellier, France, 2008. Springer-Verlag.
10. A. Fiat and A. Shamir. How To Prove Yourself: Practical Solutions to Identification and Signature Problems. In *Advances in Cryptology – CRYPTO’86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, Santa Barbara, California, USA, 1987. Springer-Verlag.
11. N. Gama and P. Q. Nguyen. New Chosen-Ciphertext Attacks on NTRU. In *Workshop on Practice and Theory in Public Key Cryptography – PKC 2007*, volume 4450 of *Lecture Notes in Computer Science*, pages 89–106, Beijing, China, June 2007. Springer-Verlag.
12. G. Gaubatz, J.-P. Kaps, and B. Sunar. Public Key Cryptography in Sensor Networks – Revisited. In *Workshop on Security in Ad-Hoc and Sensor Networks – ESAS 2004*, Heidelberg, Germany, August 2004.
13. D. Hein, J. Wolkerstorfer, and N. Felber. ECC is Ready for RFID – A Proof in Silicon. In *Conference on RFID Security*, Budapest, Hungary, July 2008.
14. International Organization for Standardization. ISO/IEC 9798 – Information technology – Security techniques – Entity authentication, 1997 – 2008. <http://www.iso.org>.
15. C. Mathas. Altera CPLDs go to the Beijing Olympics, 2008. <http://www.eetimes.com/showArticle.jhtml?articleID=208800197>.
16. M. McLoone and M. J. Robshaw. Public Key Cryptography and RFID Tags. In *The Cryptographers’ Track at the RSA Conference – CT-RSA*, volume 4377 of *Lecture Notes in Computer Science*, pages 372–384, San Francisco, California, USA, February 2007. Springer-Verlag.
17. D. Naccache. Method, Sender Apparatus and Receiver Apparatus for Modulo Operation. European patent application no. 91402958.2, 1992.
18. NESSIE consortium. Portfolio of recommended cryptographic primitives. Technical report, 2003. <https://www.cosic.esat.kuleuven.be/nessie/deliverables/decision-final.pdf>.
19. M. O’Neill (McLoone). Low-Cost SHA-1 Hash Function Architecture for RFID Tags. In *Conference on RFID Security*, Budapest, Hungary, July 2008.
20. Y. Oren and M. Feldhofer. WIPR - a Public Key Implementation on Two Grains of Sand. In *Conference on RFID Security*, Budapest, Hungary, July 2008.
21. Y. Oren and M. Feldhofer. A Low-Resource Public-Key Identification Scheme for RFID Tags and Sensor Nodes. In *Proceedings of the second ACM Conference on Wireless Network Security – WiSec’09*, Zurich, Switzerland, March 2009. ACM.
22. M. O. Rabin. Digitalized Signatures and Public-Key Functions as Intractable as Factorization. Technical report, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 1979.
23. A. Shamir. Memory Efficient Variant of Public-key Schemes for Smart Card Applications. In *Advances in Cryptology – EUROCRYPT’94*, volume 950 of *Lecture Notes in Computer Science*, pages 445–449, Perugia, Italy, January 1995. Springer-Verlag.
24. C. C. Tan, B. Sheng, and Q. Li. Serverless Search and Authentication Protocols for RFID. In *International Conference on Pervasive Computing and Communications – PerCom 2007*, New York City, New York, USA, March 2007. IEEE, IEEE Computer Society Press.
25. J. Wu and D. Stinson. How to Improve Security and Reduce Hardware Demands of the WIPR RFID Protocol. In *IEEE International Conference on RFID – RFID 2009*, Orlando, Florida, USA, April 2009.