Reliable RFID Offline Privacy

Cristian Hristea[1,2(✉)]

[1] Simion Stoilow Institute of Mathematics of the Romanian Academy, Bucharest, Romania
cristi.hristea@gmail.com
[2] Advanced Technologies Institute, Bucharest, Romania

Abstract. The paper discusses a privacy definition for offline RFID schemes, called *privacy+*. We analyse this notion and we describe an attack that proves that it can not be achieved by the accompanying protocol. In order to achieve offline privacy we develop a novel approach based on using PUFs on the reader together with encrypting the reader database. Our approach contradicts the standard assumption that privacy must be lost when a reader is compromised and that privacy restoring mechanisms must be developed. We design a protocol that implements this idea and prove it to be secure, destructive-private and immune to reader corruption in a slightly modified version of Vaudenay's model.

1 Introduction

The potential of RFID technology has become evident as more and more applications have employed the benefits of contactless communication. Domains such as asset tracking, animal or object identification, public transportation or access control have come to rely on this technology [1]. Typically, RFID involves two main entities: a small device, called *tag*, that gets attached to an object that it identifies and a powerful device, called *reader* or interrogator, that communicates wirelessly with the tag. Besides the above components one may also introduce a backend *server* that stores a database with relevant information and communicates through a secure channel with the reader.

The widespread adoption of RFID comes, however, with a potential for security and privacy violations. The wireless nature of the communication between reader and tag allows malicious scanning of tags and traffic interception. Tracking a person through her RFID possessions (access card, bus ticket) becomes a reality. The lack of physical protection of the tag gives rise to another serious threat: corruption. An attacker can gain access to a tag and extract its secrets, allowing him to permanently violate the user's privacy. The research community has addressed these concerns by designing authentication protocols for RFID [2–5] and formal privacy models [2,6–8] for analysing these protocols.

The connection type of the reader with the server (permanent or not) gives rise to RFID schemes that are online or offline. In the first scenario, the reader is considered to always be connected to the backend server through a secure connection. Furthermore, the reader cannot function offline as it does not hold any

D. Maimut et al. (Eds.): SecITC 2020, LNCS 12596, pp. 212–226, 2021.
https://doi.org/10.1007/978-3-030-69255-1_14

database information. This approach is the most common one and has received more attention from the research community in terms of security and privacy [2,4,6,7].

In contrast, offline RFID schemes assume that the reader is only sporadically connected to the central database. Since most or all of the reader's activity must be conducted without access to the server, the reader must accommodate a partial (or full) database with tag information. Applications that fit this descriptions are access control systems where many individual rooms are equipped with electronic locks [1], sporting events or public transportation [9]. For example, bus readers connect to the central database only at the end of the day. Thus, it is natural to consider the privacy implications of the attacker compromising a reader. The common approach to this threat is to assume that privacy is inherently lost after an adversary corrupts a reader [5,9–11]. Therefore, privacy-restoring mechanisms have to be defined in order to regain the privacy of the scheme after such an event. In order to implement this view, special privacy experiments, such as the one from [10] or *privacy+* from [5], need to be created.

Contribution. First of all, in this paper we discuss the notion of *privacy+* from [5], that was proposed as a modification of Vaudenay's privacy experiment [2,6] for offline schemes with privacy-restoring mechanims. We show that *privacy+* is not adequately described and does not provide the intended privacy level. We present an attack against the scheme from [5] that breaks *privacy+*. Secondly, we suggest a general approach to construct RFID authentication protocols that do not lose privacy when the reader is compromised. As far as we know, this is the first proposal of its kind. Our idea is employing Physically Unclonable Function (PUF) [12] technology on the reader (secure key storage) and a symmetric encryption scheme for protecting sensitive information stored in the reader's local database. PUFs are lightweight constructions that have been proposed as a solution against corruption on tags [4] and have become a frequently used building block for RFID protocols [3–5,13,14]. We propose a protocol that follows this idea and does not lose privacy when the reader is compromised. The protocol is analysed in a slightly modified version of Vaudenay's model. Proof sketc.hes are provided for the protocol's security and privacy properties.

Paper Structure. The paper is divided in six sections. The first section corresponds to the introduction. In the second section we fix some notations and present useful definitions. Section 3 represents a presentation of Vaudenay's RFID privacy model and the needed modifications for offline schemes. In Sect. 4 we present a state of the art of offline privacy and discuss the notion of privacy+. Section 5 represents the proposed protocol description and the security and privacy proofs.

2 Notations, Definitions and Concepts

In this section we recall a few concepts from cryptography. For details, the reader is referred to [15].

We use *probabilistic polynomial time* (PPT) algorithms as defined in [16] that can consult *oracles*. An oracle is a black box that can perform a particular computation. When considering an oracle, we do not consider its implementation or the way it works. Whenever a PPT algorithm $\mathcal{A}$ sends a value x to some oracle $\mathcal{O}$, the oracle returns to $\mathcal{A}$ a given value in $O(1)$ time.

For a set A, $a \leftarrow A$ means that a is uniformly at random chosen from A. If $\mathcal{A}$ is a probabilistic algorithm, then $a \leftarrow \mathcal{A}$ means that a is an output of $\mathcal{A}$ for some given input.

The asymptotic approach to security makes use of security parameters, denoted by λ in our paper.

Definition 1. *A positive function $f(\lambda)$ is called* negligible *if for any positive polynomial $poly(\lambda)$ there exists n_0 such that $f(\lambda) < 1/poly(\lambda)$, for any $\lambda \geq n_0$. $f(\lambda)$ is called* overwhelming *if $1 - f(\lambda)$ is negligible.*

We say that a function F is computationally indistinguishable from a random function g if no PPT algorithm can decide with more than a negligible probability whether a given value is an output of F or g.

Physically Unclonable Functions (PUFs) [12] are hardware constructions that use variations in the manufacturing process of integrated circuits (ICs) to produce IC-specific outputs. The typical analogy for PUFs is that they can provide device identification similar to human fingerprints. Thus, PUFs have a specific challenge-response behaviour, *i.e.* when queried with a challenge they produce a response that depends not just on the challenge but also on the IC on which the PUF is implemented. Common requirements for PUFs are that they are *physically unclonable* (it is infeasible to produce two PUFs that cannot be distinguished based on their challenge/response behavior), *unpredictable* (it is infeasible to predict the response to an unknown challenge), and *tamper-evident* (any attempt to physically access the PUF irreversible changes the challenge/response behaviour).

When considering PUFs for cryptographic usage one must alleviate the unstable nature of PUFs. This can be performed by using techniques such as Helper Data Algorithms [17] or with PUF constructions that offer zero bit error rate [18].

Since provable security requires ideal primitives, we adopt the concept of *ideal PUF*, that was used in several papers [3,4,14]. This concept treats PUFs from a theoretical perspective and considers them to be tamper-evident constructions that provide consistent responses (no noise) with good entropy. Our definition is the same as the one from [3].

Definition 2. *An* ideal PUF *is a physical object with a challenge/response behaviour that implements a function $P : \{0,1\}^p \rightarrow \{0,1\}^k$, where p and k are of polynomial size in λ, such that (1) P is computationally indistinguishable from a random function and (2) any attempt to physically tamper with the object implementing P results in destruction of P (P cannot be evaluated any more).*

A *pseudo-random function* (PRF) is a family of functions with the property that if we randomly choose a function from this family then its input-output behaviour is computationally indistinguishable from that of a random function.

Definition 3. *Let* $F : \{0,1\}^{\lambda} \times \{0,1\}^{\ell_1(\lambda)} \rightarrow \{0,1\}^{\ell_2(\lambda)}$ *be an efficiently computable, keyed function, where* $\ell_1(\lambda), \ell_2(\lambda)$ *are two polynomials in* λ. *F is called a pseudo-random function if* F_K *is computationally indistinguishable from a random function g, where* $K \leftarrow \{0,1\}^{\lambda}$ *is chosen uniformly at random.*

To prove that F is a PRF, we usually use a *bit guessing game* between a *challenger* $\mathcal{C}$ and an adversary $\mathcal{A}$ (with a security parameter λ) where, based on a random bit b, the challenger provides $\mathcal{A}$ with oracle access to either F ($b = 1$) or a random function ($b = 0$). At the end, $\mathcal{A}$ outputs a guess b'. The probability that $\mathcal{A}$ wins the game is denoted $P(b' = b)$. We can say that F is a PRF if it is efficiently computable and $Adv^{prf}_{\mathcal{A},F}(\lambda) = |P(b = b') - 1/2|$ is negligible.

Definition 4. *A symmetric-key encryption (SKE) scheme is a triple of PPT algorithms* $\mathcal{S} = (\mathcal{G}, \mathcal{E}, \mathcal{D})$, *where* $\mathcal{G}$ *outputs a secret key* K *and takes as input a security parameter* λ, $\mathcal{E}$ *outputs a ciphertext* y *and takes as input a key* K *and a plaintext* x, *and* $\mathcal{D}$ *is deterministic and outputs a plaintext and takes as input a key* K *and a ciphertext, such that* $x = \mathcal{D}(K, y)$, *for any* $y \leftarrow \mathcal{E}(K, x)$.

$\mathcal{S}$ is called *IND-CPA secure* if no PPT algorithm $\mathcal{A}$ that is allowed to query the encryption algorithm $\mathcal{E}$ of $\mathcal{S}$ has a non-negligible advantage to distinguish between two plaintexts of equal length, given a ciphertext of one of them.

3 RFID Systems

3.1 RFID Schemes

Let $\mathcal{R}$ be a *reader identifier* and $\mathcal{T}$ be a set of *tag identifiers* whose cardinal is polynomial in some security parameter λ.

An *RFID scheme over* $(\mathcal{R}, \mathcal{T})$ [2,6] is a triple $\mathcal{S} = (SetupR, SetupT, Ident)$ of PPT algorithms, where $SetupR$ initialises the reader and its database DB, $SetupT$ initialises a tag and stores a corresponding entry in DB and $Ident$ is an interactive protocol between the reader identified by $\mathcal{R}$ (with database DB) and a tag identified by ID (with state S). At the end of $Ident$ the reader outputs either an ID or $\perp$, while the tag outputs OK or $\perp$ (*mutual authentication*).

For mutual authentication RFID schemes, *correctness* means that, regardless of how the system is set up, after each complete execution of the interactive protocol between the reader and a legitimate tag, the reader outputs the tag's identity and the tag outputs OK with overwhelming probability.

3.2 Adversarial Model

There have been several proposals for an adversarial model [2,6–8,19] for RFID schemes. In this paper we follow *Vaudenay's model* [2,6].

In Vaudenay's model, a tag can be either *drawn* or *free* based on adversarial access to the tag (proximity). An adversary can access a drawn tag only through a temporary unique identifier *vtag*.

Within this model, the adversary is given access to the following oracles:

1. $CreateTag^b(ID)$: Creates a free tag $\mathcal{T}_{ID}$ with the identifier ID by calling the algorithm $SetupT(pk, ID)$ to generate a pair (K, S). If $b = 1$, $(ID, f(S), K)$ is added to DB and the tag is considered *legitimate*; otherwise ($b = 0$), the tag is considered *illegitimate*;
2. $DrawTag(\delta)$: This oracle chooses a number of free tags according to the distribution δ, let us say n, and draws them. That is, n temporary identities $vtag_1, \ldots, vtag_n$ are generated and then the oracle outputs $(vtag_i, b_i)$, where b_i specifies whether the tag $vtag_i$ is legitimate or not;
3. $Free(vtag)$: The tag identified by $vtag$ becomes free and the identifier $vtag$ will no longer be used. It is assumed that the temporary state of the tag is erased when the tag is freed. This is a natural assumption that corresponds to the fact that the tag is no longer powered by the reader;
4. $Launch()$: Launches a new protocol instance and assigns a unique identifier to it. The oracle outputs the identifier;
5. $SendReader(m, \pi)$: Outputs the reader's answer when the message m is sent to it as part of the protocol instance π. When m is the empty message, abusively but suggestively denoted by $\emptyset$, this oracle outputs the first message of the protocol instance π, assuming that the reader does the first step in the protocol;
6. $SendTag(m, vtag)$: outputs the tag's answer when the message m is sent to the tag referred to by $vtag$. When m is the empty message, this oracle outputs the first message of the protocol instance π, assuming that the tag does the first step in the protocol;
7. $Result(\pi)$: Outputs $\perp$ if in session π the reader has not yet made a decision on tag authentication (this also includes the case when the session π does not exist), 1 if in session π the reader authenticated the tag, and 0 otherwise (this oracle is both for unilateral and mutual authentication);
8. $Corrupt(vtag)$: Outputs the current permanent (internal) state of the tag referred to by $vtag$, when the tag is not involved in any computation of any protocol step (that is, the permanent state before or after a protocol step).

There has been consistent debate on whether the *Corrupt* oracle returns only the permanent state or the volatile state of a tag as well [3,20]. As stated in the oracle description, we consider that *Corrupt* returns only the permanent state. Based on access to the *Corrupt* oracle, adversaries are classified into: *weak* (no access to *Corrupt*), *forward* (no other oracles can be used after *Corrupt*), *destructive* (after corrupting a tag it is considered destroyed) and *strong* (no restrictions).

Another class of adversaries called *narrow* is created when the adversary is denied access to the *Result* oracle. This class can be combined with the previous categories and we obtain another four classes of adversaries, *narrow weak*, *narrow forward*, *narrow destructive*, and *narrow strong*.

3.3 Security

Security for RFID schemes is composed of two complementary notions: *tag authentication* and *reader authentication*.

The tag authentication property is defined by means of an experiment denoted $RFID^{t\text{-}auth}_{\mathcal{A},\mathcal{S}}(\lambda)$ where a challenger sets up for a strong adversary $\mathcal{A}$ an RFID scheme $\mathcal{S}$ in which $\mathcal{A}$ must impersonate a legitimate uncorrupted tag to the reader. The adversary is compelled to compute at least one of the messages exchanged in the protocol. In the end $\mathcal{A}$ outputs a bit b. The advantage of $\mathcal{A}$ in the experiment $RFID^{t\text{-}auth}_{\mathcal{A},\mathcal{S}}(\lambda)$ is defined as the probability that the adversary outputs 1. We say that $\mathcal{S}$ achieves *tag authentication* if $Adv^{t\text{-}auth}_{\mathcal{A},\mathcal{S}}$ is negligible, for any strong adversary $\mathcal{A}$.

The experiment $RFID^{r\text{-}auth}_{\mathcal{A},\mathcal{S}}(\lambda)$ for reader authentication is identical to the $RFID^{t\text{-}auth}_{\mathcal{A},\mathcal{S}}(\lambda)$ except that $\mathcal{A}$ has to impersonate the reader to a legitimate uncorrupted tag. An RFID scheme $\mathcal{S}$ achieves *reader authentication* if the adversarial advantage in this experiment, $Adv^{r\text{-}auth}_{\mathcal{A},\mathcal{S}}$, is negligible, for any strong adversary $\mathcal{A}$.

3.4 Privacy

The *privacy* notion that was defined in Vaundenay's model basically means that the communication between the reader and the tags does not leak any information to an eavesdropping adversary. This is modelled through the use of a blinder.

A *blinder* for an adversary $\mathcal{A}$ that belongs to some class V of adversaries is a PPT algorithm $\mathcal{B}$ that: (1) simulates the *Launch*, *SendReader*, *SendTag*, and *Result* oracles for $\mathcal{A}$, without having access to the corresponding secrets and (2) passively looks at the communication between $\mathcal{A}$ and the other oracles allowed to it by the class V. When the adversary $\mathcal{A}$ interacts with the RFID scheme by means of a blinder $\mathcal{B}$, we say that $\mathcal{A}$ is *blinded by* $\mathcal{B}$ and denote this by $\mathcal{A}^{\mathcal{B}}$. We emphasize that $\mathcal{A}^{\mathcal{B}}$ is allowed to query the oracles *Launch*, *SendReader*, *SendTag*, and *Result* only by means of $\mathcal{B}$; all the other oracles are queried as a standard adversary.

Given an adversary $\mathcal{A}$ and a blinder $\mathcal{B}$ for it, let us define two experiments (privacy games) $RFID^{prv-0}_{\mathcal{A},\mathcal{S}}(\lambda)$ and $RFID^{prv-1}_{\mathcal{A},\mathcal{S},\mathcal{B}}(\lambda)$ where the adversary interacts, according to its class, with the real RFID scheme and, respectively, with the blinded scheme. After an interaction phase, the adversary receives the hidden table of the $DrawTag$ oracle, enters an analysis phase and outputs a bit b.

The *advantage* of $\mathcal{A}$ blinded by $\mathcal{B}$, denoted $Adv^{prv}_{\mathcal{A},\mathcal{S},\mathcal{B}}(\lambda)$, is

$$Adv^{prv}_{\mathcal{A},\mathcal{S},\mathcal{B}}(\lambda) = \mid P(RFID^{prv-0}_{\mathcal{A},\mathcal{S}}(\lambda) = 1) - P(RFID^{prv-1}_{\mathcal{A},\mathcal{S},\mathcal{B}}(\lambda) = 1) \mid$$

An RFID scheme achieves privacy for a class V of adversaries if for any adversary $\mathcal{A} \in V$ there exists a blinder $\mathcal{B}$ such that $Adv^{prv}_{\mathcal{A},\mathcal{S},\mathcal{B}}(\lambda)$ is negligible.

3.5 Vaudenay's Model for Offline Schemes

Vaudenay's model has been constructed for analysing online RFID schemes. Modifications for the offline setting have been proposed in [5]. In this paper we propose similar modifications, inspired from [5] and [8].

In an offline RFID scheme, the reader and the server are distinct entities. To accommodate this, the reader from Sect. 3.1 becomes the *server* of the offline one. Thus, a new PPT algorithm *SetupReader* has to be incorporated in the offline RFID scheme definition. *SetupReader* is responsible for creating a reader with an identifier ID_R, a state s and a database DB_{ID_R}. The reader database DB_{ID_R} is constructed from the system database DB.

Two additional oracles have to be incorporated in the adversarial model:

- $CreateReader(ID_R)$ - creates the reader ID_R and calls *SetupReader*;
- $CorruptReader(ID_R)$ - returns the internal state s of the reader ID_R as well as the reader database. The reader is considered destroyed and cannot be used anymore.

Furthermore, the *SendReader* oracle needs to take into account the reader identity besides the session identifier and the message.

After a legitimate tag is created and added in the server database, we require all reader databases to be updated with needed information regarding the created tag. For *tag authentication*, *reader authentication* and *privacy* we use the same security experiments as we do not allow the adversary to query the *CorruptReader* oracle. This is consistent with the model from [8] and with the modifications from [5].

We define a new privacy notion, *offline privacy* for which we use the same privacy experiment as in Sect. 3.4, with the modification that the adversary is given access to the *CorruptReader* oracle. The definition for the adversary's advantage remains unchanged.

We say that an RFID scheme achieves *offline privacy* if an adversary has negligible advantage in distinguishing the real RFID scheme from the blinded version, even in the presence of the *CorruptReader* oracle.

4 Offline Privacy in RFID Protocols

4.1 Related Work

Symmetric Encryption Protocol. In [9] an offline RFID protocol based on symmetric encryption is proposed. Each tag stores a unique secret key K_T, an identifier ID_T and a counter C_T. The readers store an identifier ID_R and for each tag ID_T a specific tag-reader secret K_{TR} and a counter C_R. The protocol debuts with the reader sending ID_R, C_R and a nonce n_R. The tag computes the tag-reader key as $K_{TR} = E_{K_T}(ID_R, C_R)$, generates a nonce n_T and sends to the reader $E_{K_T R}(n_R, n_T)$. The reader then searches in its database for a key K_{TR} that decrypts the message (n'_R, n'_T) such that $n'_R = n_R$. If so, the tag is authenticated and the reader sends n'_T. If the equality $n_T = n'_T$ holds then the tag authenticates the reader and decides if it updates its counter. The counters C_T, C_R are used to restore privacy. After a reader is compromised, the backend server updates the scheme counter C_B and all other readers are updated with new keys based on the new counter. Thus, the C_B counter becomes C_R. The scheme privacy is restored after all tags have replaced their C_T with the new C_R.

Indistinguishability and Hash Protocol. The subject of achieving offline privacy using symmetric cryptography was also tackled in [10]. The paper describes two versions of a protocol (simple and enhanced) that restore privacy after an adversary has corrupted a reader. The proposed protocol is a variant of the OSK protocol [21] and uses a hash function H as the cryptographic primitive. The tag is required to store a system constant C_0 and two keys K, K' (the first being a shared secret with the reader and the latter being a shared secret with the backend). The readers store for each tag the last known key K_T which is used to trigger the update procedure (that restores privacy), a communication key $\tilde{K}$ and a MAC computed as $H(K', C_0)$. Note that the reader does not posses key K'. In the protocol, the tag answers to a reader's challenge n with $c = H(K, n)$ and updates $K = H(K)$. For every entry in its database, the reader will iterate a number of times (N) for K_T and $\tilde{K}$ in order to identify the tag, that is find $0 < i < N$ such that $c = H(H^i(K_T))$ or $c = H(H^i(\tilde{K}))$. If any of the two conditions is met then the reader identifies and authenticates the tag. However, if the first condition is met the reader will trigger an update of the tag's secrets, which happens during a privacy restore phase. The proposed protocol offers only tag authentication and not mutual authentication. The paper also describes a privacy model used for the offline setting. The model is a combination between the models from [2] and [7]. The online privacy experiment is based on indistinguishability: the adversary is required to distinguish between two uncorrupted tags. In an initial step the adversary interacts with the system and outputs two uncorrupted tags T_0, T_1. The challenger then chooses a bit $b \leftarrow \{0, 1\}$ and gives the adversary access to T_b. After a second session of interacting with the system the adversary outputs a bit b'. The adversary wins if $b = b'$. For the offline case, the authors modify this experiment by adding a system synchronisation and successful protocol runs with T_0, T_1 after the challenger chooses b.

Hash and PUF Protocol. The first attempt at achieving offline privacy in Vaudenay's model has been performed in [5], where a PUF-based RFID scheme and an offline privacy experiment (*privacy+*) are proposed. We will present this scheme with some simplification: the double PUF protection method, proposed by the authors in order to thwart the *cold boot attack*, will be omitted.

The scheme is based on a hash function H and requires each tag to be equipped with a PUF P and to store a seed G, a counter C_T and an identifier ID. The secret key of the tag S is protected by the PUF. The reader needs an identifier ID_R, a counter C_R and a database DB_R that contains entries (ID_i, K_i) for each tag, where the key K_i is computed from the tag's secret key and from the reader identifier and counter by means of H. After receiving ID_R, C_R and a nonce r_1 from the reader, the tag also generates a nonce r_2. The tag then checks if the reader is up to date ($C_R \geqslant C_T$) and evaluates the PUF to obtain its key $S = P(G)$. Next, the reader specific key is obtained by $K = H(S, ID_R, C_R)$ and the tag computes $v_1, v_2 = H(K, r_1, r_2)$. v_1, r_2 will be sent to the reader, while v_2 will be kept to perform the reader authentication. The reader searches its database for an entry (ID_i, K_i) such that for $v_1', v_2' = H(K_i, r_1, r_2)$ the equality $v_1 = v_1'$ holds. If it finds such a tag then the reader authenticates the tag and

sends v_2' to the tag (otherwise a random number will be sent). If $v_2 = v_2'$ then the tag will authenticate the reader and perform an update if necessary (a reader was compromised and counters were increased). The protocol is depicted in Appendix A.

4.2 Privacy+ Discussion

In [5] *privacy+* is introduced as an adaptation of the privacy from Sect. 3.4 for offline schemes with privacy-restoring mechanisms, where the adversary is allowed to query the *CorruptReader* oracle. The definition for *privacy+* (definition 3.6 from [5]) states that an RFID scheme achieves this level of privacy if it is still private after (1) an adversary has corrupted some of the readers, (2) the remaining readers are updated with new information and (3) all existing tags run at least one successful protocol instance with an updated reader. This falls in line with the offline privacy experiment defined in [10] for the indistinguishability-based privacy model.

Unfortunately the details for the *privacy+* experiment are not adequately adapted for Vaudenay's model which uses a blinder-based approach. If we consider the original privacy experiment from [2] and add the conditions from above (i.e after a *CorruptReader* query, the system updates the readers and the tags) then the result of Theorem 5.7 from [5], claiming *destructive privacy+* (for the proposed protocol) becomes invalid. Since in Vaudenay's model the goal of the privacy adversary is to distinguish with non-negligible probability between the real RFID scheme and the blinded version, the adversary may simply perform a complete session between a reader and a tag, corrupt the reader and obtain the tag's key (K_i). With this key the adversary may check if the messages from the protocol run were exchanged correctly (the protocol is assumed to be correct). Clearly, in the real RFID case the verification will be successful while in the blinded version the result will be unsuccessful since the blinder simulates the messages without knowing the key K_i. This gives the adversary a non-negligible advantage of distinguishing between the two RFID schemes. The details of this attack are presented below.

1. $CreateTag^1(ID)$ ($\mathcal{A}$ creates a legitimate tag);
2. $CreateReader(ID_R)$ ($\mathcal{A}$ creates a reader ID_R);
3. $vtag \leftarrow DrawTag(P(ID) = 1)$ ($\mathcal{A}$ draws ID);
4. $\pi \leftarrow Launch()$;
5. $(ID_R, c_1, r_1) \leftarrow SendReader(ID_R, \perp, \pi)$;
6. $(r_2, v_1) \leftarrow SendTag(vtag, (ID_R, c_1, r_1))$;
7. $v_2 \leftarrow SendReader(ID_R, (r_2, v_1), \pi)$;
8. $b = Result(\pi)$;
9. $DB_{ID_R} \leftarrow CorruptReader(ID_R)$;
10. The system gets updated as definition 3.6 requires;
11. Find (ID, K) in DB_{ID_R};
12. if $(v1, v_2) == H(K, r_1, r_2)$ *and* $b == 1$
    then output 0 ($\mathcal{A}$ interacts with the real system)
    else output 1 ($\mathcal{A}$ interacts with the blinder)

We point out that the system wide update, of both readers and tags, is useless against this attack as the adversary uses the secrets from the reader to verify protocol runs that occurred before the *CorruptReader* query and not after.

Given the above, we consider that a different approach must be taken when designing a blinder-based privacy experiment for offline schemes with privacy-restoring mechanisms. We consider *privacy+* to be more in line with the *extended soundness* notion from [8], that defines tag and reader authentication when the adversary is allowed access to the *CorruptReader* oracle.

5 Proposed Protocol

In this section we build upon the efforts of [3,5,10] and we propose an RFID scheme that offers offline privacy, mutual authentication and destructive privacy. For our scheme we use symmetric cryptography (a PRF $F = (F_K)_{K\in\{0,1\}^k}, F_K : \{0,1\}^{2\alpha+1} \rightarrow \{0,1\}^k$ and an $IND-CPA$ symmetric encryption scheme $\mathcal{S} = (\mathcal{E}, \mathcal{D}, \mathcal{G})$ with key length k and block length k) and endow both tags and readers with PUFs ($P : \{0,1\}^p \rightarrow \{0,1\}^k$) in order to make them resilient to invasive adversaries. The parameters k, α, p are all polynomial in a security parameter λ.

5.1 Protocol Description

In the proposed scheme each tag is associated a unique secret key K. This key is only known to the tag and the backend server. Each reader will communicate with a tag based on a common key K_{TR} derived with the tag's secret key from the reader's identifier. Note that the reader does not need to know K because the backend server will supply the reader with K_{TR} when the reader is created, or when a tag is added. In order to prevent the attack from Sect. 4.2, we require the reader to store $\tilde{K}_{TR}$, which is the encrypted form of K_{TR} so as to prevent the adversary from breaching privacy. We will achieve this by means of a symmetric encryption scheme that is *IND-CPA* secure. The reader will encrypt or decrypt using a reader specific key K_R which in turn will be protected from corruption by means of a PUF $K_R = P(S_R)$.

Now let us describe the protocol. The reader starts by generating a random number x and sending ID_R, x to the tag. In turn, the tag will also generate a nonce y and then prepare the reader's answer. After extracting the tag key from the PUF $K = P(S)$, the tag will compute its shared key with the reader $K_{TR} = F_K(0, 0^\alpha, ID_R)$ and then $z = F_{K_{TR}}(0, x, y)$. The tuple y, z will be sent to the reader. Using its PUF, the reader will extract its key $K_R = P(S_R)$ and assign to w a random value. For each entry in the database $(ID, \tilde{K}_{TR})$, the reader will decrypt the tag key $K_{TR} = \mathcal{D}(K_R, \tilde{K}_{TR})$ and check if the tag answer is valid $z = F_{K_{TR}}(0, x, y)$. If such an entry is found then the tag is authenticated and w becomes the reader's answer $w = F_{K_{TR}}(1, x, y)$. The tag will verify w and decide if it outputs OK (reader is authenticated) or $\perp$.

	Reader $(ID_R, S_R, DB = [(ID, \tilde{K}_{TR})])$		**Tag** (ID, S)
1	$x \leftarrow \{0,1\}^\alpha$	$\xrightarrow{ID_R, x}$	
2			$y \leftarrow \{0,1\}^\alpha$ $K = P(S)$ $K_{TR} = F_K(0, 0^\alpha, ID_R)$
		$\xleftarrow{y,\ z}$	$z = F_{K_{TR}}(0, x, y)$
3	$K_R = P(S_R)$ $w \leftarrow \{0,1\}^{2\alpha+1}$ For $(ID, \tilde{K}_{TR}) \in DB$ $K_{TR} = \mathcal{D}(K_R, \tilde{K}_{TR})$ If $z = F_{K_{TR}}(0, x, y)$ then output ID (T. auth.) $w = F_{K_{TR}}(1, x, y)$ else output $\perp$	$\xrightarrow{w}$	
4			If $w = F_{K_{TR}(1,x,y)}$ then output OK (R. auth.) else output $\perp$

Fig. 1. Proposed RFID scheme

5.2 Security and Privacy Analysis

We will now perform a security and privacy analysis of our protocol in Vaudenay's model. Due to lack of space we will only give the main idea of the proofs. For detailed security and privacy proofs the reader is referred to [3].

Theorem 1. *The RFID scheme in Fig. 1 achieves tag authentication, provided that F is a PRF and the tags are endowed with ideal PUFs.*

Proof. Let us assume that there exists an adversary A_{t-auth} that breaks this property with non-negligible probability. Then we will use A_{t-auth} to construct a PPT algorithm A_{PRF} that wins the PRF experiment against F with non-negligible probability. For simplicity we will assume there is only one reader R in the RFID system. A_{PRF} will engage in the PRF security game against a challenger $\mathcal{C}$, which will provide it with oracle access to $F_{K_{TR}}$ for some randomly chosen K_{TR} (or a random function). A_{PRF} will simulate the RFID scheme and play the role of challenger for A_{t-auth} in the tag authentication experiment. A_{PRF} will guess which tag ID will be impersonated by A_{t-auth} (this probability is polynomial) and associate this tag with the oracle from the PRF challenger (*i.e.* all queries from A_{t-auth} related to ID will be answered with the help of the oracle). Eventually A_{t-auth} will output a message (y, z). A_{PRF} will then submit $(0, x, y)$ to the PRF oracle and decide whether it is playing with F or a random

function based on the equality between z and the PRF oracle output (z should be the output of F for some key chosen by $\mathcal{C}$, $z = F_{K_{TR}}(0, x, y)$). The probability that A_{PRF} wins is the probability that A_{t-auth} wins the tag authentication game multiplied by the probability that A_{PRF} guesses the impersonated tag. This clearly contradicts the fact that F is a PRF.

Theorem 2. *The RFID scheme in Fig. 1 achieves reader authentication, provided that F is a PRF and the tags are endowed with ideal PUFs.*

Proof. The proof is similar to the one presented above. We can construct an adversary A_{PRF} that breaks the pseudo-randomness of F by using A_{r-auth}. This contradicts the hypothesis.

Theorem 3. *The RFID scheme in Fig. 1 achieves destructive privacy, provided that F is a PRF and the tags are endowed with ideal PUFs.*

Proof. We will use the sequences of games approach [22] to prove that for any destructive-private adversary A there exists a blinder B such that A's advantage of distinguishing between the real RFID scheme and the blinder is negligible. For simplicity we assume that there is a single reader in the RFID system. We define a series of games $G_0, ..., G_7$ where G_0 is the real RFID scheme and G_7 is the blinded version. In each game a probability distribution (the output of the PUF and the blinder simulated oracles *Launch, SendReader, SendTag, Result*) is replaced by another distribution indistinguishable from the replaced one. Since the adversary has a negligible advantage in distinguishing between the transition of two consecutive games, we conclude that A has a negligible advantage of distinguishing between the real RFID scheme G_0 and the blinded version G_7, *i.e.* the scheme achieve destructive privacy.

Theorem 4. *The RFID scheme in Fig. 1 achieves offline privacy, provided that $\mathcal{S}$ is $IND-CPA$ secure and the readers are endowed with ideal PUFs.*

Proof. We will use the sequences of games approach, same as above. We will define two additional games G_8, G_9 and show that the advantage of the adversary of distinguishing between the real RFID scheme and the blinder does not change when the adversary is allowed access to $CorruptReader$. We replace in G_8 the output distribution of the PUF from the reader and in G_9 the distribution of the ciphertexts of the reader encryption scheme with indistinguishable probability distributions. Since the PUFs are ideal and the encryption scheme is $IND-CPA$ secure we conclude that the scheme offers destructive privacy and offline privacy.

6 Conclusions

In this paper we have analysed the *privacy+* security notion and have proven that it is not an adequate modification of Vaudenay's privacy experiment. Therefore, RFID schemes relying on privacy-restoring mechanisms cannot use it. An attack on the accompanying protocol has been provided in this sense. Designing

a blinder-based privacy experiment that allows privacy-restoring mechanisms remains an open problem.

This paper has also presented a novel approach for providing privacy in offline RFID schemes without losing privacy when a reader is compromised. This technique is based on using PUFs on the reader together with encrypting the reader database. Following this idea, we have designed a protocol that provides destructive privacy and is immune to reader corruption attacks. To the best of our knowledge, this is the first protocol to achieve this. The protocol is proven secure and private in a slightly modified version of Vaudenay's model.

A Hash and PUF-Based RFID Scheme

	Reader $(ID_R, c_R, DB = [(ID_i, K_i)])$		**Tag** (ID, G, c_T)
1	$r_1 \leftarrow \{0,1\}^\alpha$	$\xrightarrow{ID_R, r_1, c_R}$	
2			$r_2 \leftarrow \{0,1\}^\alpha$ If $c_R \geqslant c_T$ then $S = P(G)$ $K = H(S, ID_R, c_R)$ $v_1, v_2 = H(K, r_1, r_2)$ else $v_1 \leftarrow \{0,1\}^\gamma$
		$\xleftarrow{r_2,\ v_1}$	
3	If $\exists (ID_i, K_i) \in DB$ s.t. $v_1', v_2' = H(K_i, r_1, r_2)$ $v_1' = v_1$ then output ID (T. auth.) else output $\perp$ $v_2' \leftarrow \{0,1\}^\gamma$		
		$\xrightarrow{v_2'}$	
4			If $v_2 = v_2'$ && $c_R > c_T$ then output OK (R. auth.) $c_T = c_R$ else output $\perp$

Fig. 2. RFID scheme from [5]

References

1. Finkenzeller, K.: RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification, 3rd ed. Wiley Publishing (2010)
2. Vaudenay, S.: On privacy models for RFID. In: Proceedings of the Advances in Cryptology 13th International Conference on Theory and Application of Cryptology and Information Security, service. ASIACRYPT 2007, pp. 68–87. Springer-Verlag (2007)
3. Hristea, C., Tiplea, F.L.: Destructive privacy and mutual authentication in Vaudenay's RFID model. IACR Cryptol. ePrint Arch., 2019, 73 (2019). https://eprint.iacr.org/2019/073
4. Sadeghi, A.-R., Visconti, I., Wachsmann, C.: PUF-enhanced RFID security and privacy. In: Workshop on Secure Component and System Identification (SECSI), vol. 110 (2010)
5. Kardaş, S., Çelik, S., Yildiz, M., Levi, A.: PUF-enhanced offline RFID security and privacy. J. Netw. Comput. Appl. **35**(6), 2059–2067 (2012)
6. Paise, R.-I., Vaudenay, S.: Mutual authentication in RFID: Security and privacy. In: Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, ser. ASIACCS 2008, pp. 292–299. ACM, New York (2008)
7. Juels, A., Weis, S.A.: Defining strong privacy for RFID. ACM Trans. Inf. Syst. Secur. **13**(1), 1–23 (2009)
8. Hermans, J., Peeters, R., Preneel, B.: Proper RFID privacy: model and protocols. IEEE Trans. Mob. Comput. **13**(12), 2888–2902 (2014)
9. Avoine, G., Lauradoux, C., Martin, T.: When compromised readers meet RFID. In: Youm, H.Y., Yung, M. (eds.) WISA 2009. LNCS, vol. 5932, pp. 36–50. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10838-9_4
10. Garcia, F.D., van Rossum, P.: Modeling privacy for off-line RFID systems. In: Gollmann, D., Lanet, J.-L., Iguchi-Cartigny, J. (eds.) CARDIS 2010. LNCS, vol. 6035, pp. 194–208. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12510-2_14
11. Avoine, G., Coisel, I., Martin, T.: A privacy-restoring mechanism for offline RFID systems. In: Proceedings of the fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks, pp. 63-74 (2012)
12. Maes, R.: Physically Unclonable Functions: Constructions. Springer Verlag, Properties and Applications (2013)
13. Devadas, S., Suh, E., Paral, S., Sowell, R., Ziola, T., Khandelwal, V.: Design and implementation of PUF-based "unclonable" RFID ICs for anti-counterfeiting and security applications. In: 2008 IEEE International Conference on RFID, pp. 58-64. IEEE (2008)
14. Tuyls, P., Batina, L.: RFID-tags for anti-counterfeiting. In: Pointcheval, D. (ed.) CT-RSA 2006. LNCS, vol. 3860, pp. 115–131. Springer, Heidelberg (2006). https://doi.org/10.1007/11605805_8
15. Katz, J., Lindell, Y.: Introduction to Modern Cryptography. CRC press, United States (2020)
16. Sipser, M.: Introduction to the Theory of Computation. Cengage Learning (2012)
17. Delvaux, J., Gu, D., Schellekens, D., Verbauwhede, I.: Helper data algorithms for puf-based key generation: overview and analysis. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **34**(6), 889–902 (2014)

18. Chuang, K.-H., Bury, E., Degraeve, R., Kaczer, B., Linien, D., Verbauwhede, I.: A physically unclonable function with 0% ber using soft oxide breakdown in 40nm cmos. In: IEEE Asian Solid-State Circuits Conference (A-SSCC), pp. 157–160. IEEE (2018)
19. Hermans, J., Pashalidis, A., Vercauteren, F., Preneel, B.: A new RFID privacy model. In: Atluri, V., Diaz, C. (eds.) ESORICS 2011. LNCS, vol. 6879, pp. 568–587. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23822-2_31
20. Tiplea, F.L., Hristea, C.: Privacy and reader-first authentication in Vaudenay's RFID model with temporary state disclosure. In: IACR Cryptol. ePrint Arch., p. 113 (2019)
21. Ohkubo, M., Suzuki, K. and Kinoshita, S.: Cryptographic approach to privacy-friendly tags. In: RFID Privacy Workshop. vol. 82 (2003)
22. Shoup, V.: Sequences of games: a tool for taming complexity in security proofs (2004)