Journal of Network and Computer Applications 35 (2012) 2059–2067

Contents lists available at SciVerse ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

PUF-enhanced offline RFID security and privacy

Süleyman Kardaş [a,b,*], Serkan Çelik [a,b], Muhammet Yıldız [a,b], Albert Levi [b]

[a] TÜBİTAK BİLGEM UEKAE, Kocaeli, Turkey
[b] Sabancı University, Faculty of Engineering and Natural Sciences, İstanbul, Turkey

ARTICLE INFO

Article history:
Received 16 January 2012
Received in revised form
16 July 2012
Accepted 20 August 2012
Available online 6 September 2012

Keywords:
RFID
PUF
Security
Privacy
Compromise of reader

ABSTRACT

RFID (Radio Frequency IDentification) based communication solutions have been widely used nowadays for mobile environments such as access control for secure system, ticketing systems for transportation, and sport events. These systems usually depend on readers that are not continuously connected to a secure backend system. Thus, the readers should be able to perform their duties even in offline mode, which generally requires the management by the readers of the susceptible data. The use of RFID may cause several security and privacy issues such as traceability of tag owner, malicious eavesdropping and cloning of tags. Besides, when a reader is compromised by an adversary, the solution to resolve these issues getting worse. In order to handle these issues, several RFID authentication protocols have been recently proposed; but almost none of them provide strong privacy for the tag owner. On the other hand, several frameworks have been proposed to analyze the security and privacy but none of them consider offline RFID system.

Motivated by this need, in this paper, we first revisit Vaudenay's model, extend it by considering offline RFID system and introduce the notion of compromise reader attacks. Then, we propose an efficient RFID mutual authentication protocol. Our protocol is based on the use of physically unclonable functions (PUFs) which provide cost-efficient means to the fingerprint chips based on their physical properties. We prove that our protocol provides destructive privacy for tag owner even against reader attacks.

1. Introduction

Radio Frequency IDentification (RFID) technology is getting pervasively deployed in many daily life applications ranging from inventory management to anti-counterfeiting protection. A typical RFID system consists of three components that actively or passively interact with each other (see Fig. 1). The first component is the backend system, where is the central synchronization point for all the other components and all initialization routines take place. Moreover, the backend system is assumed to be secure against all kinds of attacks. Another component is a group known as *readers* or *interrogators* which are in the middle of the other two components. Their main role is to identify the third components, which will be discussed next, in the paper. The last component of the RFID systems is called *tags* or *labels*. There are three types of tags classified as *passive, active* and *battery assisted passive* tags. Passive tags are low-cost devices that have no internal power source and need an external signal to be invoked. On the other hand, they represent the most commonly used tag class in RFID applications. *Active tags* contain a power source (i.e., a battery) and can actively send signals to a reader for communication. The last tag family (*battery assisted passive tags*) contains a low power source but these kinds of tags still need a wake up signal as passive tags do.

There are two types of RFID structures in terms of the backend server–reader connection. First one is referred to as *central database model* but we name it as *online model*. The latter is referred to *offline model*.

In the online model, the backend system contains all the tag related information. The readers are assumed to be always connected to the backend system. Although it is between the tags and the backend system, the main duty of the reader is to query the tag and return the reply to the backend system without knowing the contents of the tag reply. It does not contain any tag specific information such as keys, IDs, and counters. A nice example is a building access system where the users have their own cards with those they enter rooms or access different facilities. Since the system is compact, all the readers have a live connection to the central database. The major shortcoming with the online model is that the readers have to be available and the secure connection between the readers and the backend system has to be continuous. This assumption is not practical in such applications requiring an intermittent access to the central database.

In offline model, the reader is connected to the central server only during synchronizations of tag information, reader information, and firmware update. Since the reader in this model is offline

* Corresponding author at: TÜBİTAK BİLGEM UEKAE, Kocaeli, Turkey. Tel.: +90 2626481701.
E-mail addresses: suleyman.kardas@tubitak.gov.tr, skardas@gmail.com (S. Kardaş), serkan.celik@tubitak.gov.tr (S. Çelik), muhamnmet.yildiz@tubitak.gov.tr (M. Yıldız), levi@sabanciuniv.edu (A. Levi).

1084-8045/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jnca.2012.08.006

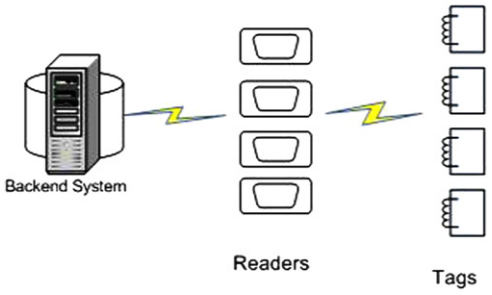

Fig. 1. A typical RFID system.

during most of its life cycle, it should be able to identify and authenticate the tags without connecting to the server. This requires the readers to have a higher resource availability and computational capacity compared to the online model. The bus transportation scenario is a good example for an offline model. As it is known, the buses are mobile and the reader in the buses are disconnected from the server during their service hours. However, they connect to the central server during its parking lot outside the working hours. The authentication mechanism used in the offline system should prevent unauthorized tag usage and it should also provide strong privacy for protecting privacy of the tag owner. On the other hand, since the readers are mobile, compromising readers is very likely to occur (Avoine et al., 2009). Such attacks make those security and privacy issues worse.

In order to handle those issues, various new authentication mechanisms that confront to ISO/IEC 9798 entity authentication standard have been recently proposed (Baudron et al., 2001; Oren and Feldhofer, 2008; Tan et al., 2008; Avoine et al., 2009). These protocols range from a simple challenge response protocol to a much complicated architecture. Research has shown that none of these protocols provide strong privacy for tag owner when a reader is compromised by an adversary.

Besides, several frameworks have been recently proposed to analyze security and privacy of RFID authentication protocols (Avoine, 2005; Vaudenay, 2007; Juels and Weis, 2009; Paise and Vaudenay, 2008; Burmester et al., 2006; Canard et al., 2010; Deng et al., 2010; Ha et al., 2008; Hermans et al., 2011; Lai et al., 2010; van Deursen et al., 2008). However, none of them considers offline reader system in which compromising a reader attack is very likely to occur.

Our contribution. In this study, we first revisit Vaudenay's adversary model and extend it to the offline RFID system. We introduce the notion of reader compromise attacks. Then, we define the notion of *privacy+* where compromise attacks on readers are considered. After that we propose a new RFID mutual authentication protocol. In our protocol, we use physically unclonable functions (*PUF*) as unique identity provider mechanisms for the tags. PUF outputs are analogous to the biometric traits in terms of uniqueness. This property provides a secure key derivation for low-cost RFID tags (Kardas et al., 2011). In our protocol, we utilize this PUF mechanism to make RFID tags strong against side-channel attacks. Finally, we prove that our protocol provides the narrow destructive privacy for tag owner. Also, we prove that our protocol satisfies narrow destructive *privacy+* in case of compromise reader attacks. To the best our knowledge, it is the first protocol which uses symmetric operations and PUF functions and satisfies these privacy properties.

The rest of the paper is organized as follows: in Section 2, we briefly discuss offline RFID system, then in Section 3 we describe the notion of PUF functions and its characteristics. Section 4 describes the notations and adversary used in privacy model. In Section 5, the extended model is described. Section 6 describes the proposed authentication protocol. In Section 7, we present the adversary capabilities and formal security analysis of the protocol. Lastly, in Section 8, we give a brief discussion and conclude the paper.

2. Background information, preliminaries and notations

In this section, we first give a brief discussion about the offline RFID system. Then, we provide the related works on the physically unclonable functions. Finally, we present the preliminaries and notations used through the paper.

2.1. Offline RFID system

RFID technology is getting more popular in large-scale applications especially in mobile environments, such as ticketing system for mass transportation and sport events. These applications work with off-line RFID system which requires three components: RFID tags, readers and server. Tags are inherently mobile but they are not tamper resistant against any physical attack. Considering mobile handheld devices, the readers are regarded as mobile and they are intermittently connected to the central server. For instance, the ticket verifier of a flying agent in the site of a sport event is connected to the server only when the agent is back to the headquarter. Therefore, the readers should be able to authenticate the customers (Avoine et al., 2009) when the server is offline.

Besides, since the handheld reader is mobile, the loss or the theft of a handheld reader is a typical case of a threat for offline system. Since the privacy-friendly authentication protocol for identifying the tags is run by offline reader, there is no practical solution to renovate the privacy as soon as the readers are compromised by a malicious adversary. However, renewing all the tag information, which is impractical, can defeat this threat.

The last component is the server which hosts a centralized back-end system that manages data about the tickets and customers. Since the offline reader is not always connected to server, the detection of fraud, for example, the multiple use of tickets, is very difficult. Moreover, the firmware software or the configuration data of the reader are uploaded to the reader only at an inspection done by a maintenance personnel.

To exemplify the fear of compromise reader attacks in offline infrastructures, we consider a real-life RFID ticketing system deployed by RFIDea during a 3-day automobile race in 2009 (RFIDea, 2012). In this deployment, several mobile readers and more than 100,000 tags for tickets are used in order to reduce queues in the event and curtailing fraud. The system setup procedure works as follows. The mobile readers are the first setup by the administrator and then given to the agents in the field until the end of the event. The mobile readers store the tags' secret keys in their database which are used for authentication and identification of all spectators' and employees' badges. The agents are not mobile, whereas spectators and employees are. Thus, the offline RFID system can easily manage the mobility of all the participants during the event. In this event, contrary to the expectations of the event organizer some of the readers were stolen, so the attack so called compromise of reader is a realistic attack. With these readers, the participants are traced which violates the privacy. This case study has been already analyzed in Avoine et al. (2012).

2.2. Physically unclonable function (PUF)

A physically unclonable function (PUF) is a function, which maps a set of challenges to a set of responses based on an intractably complex physical characteristics (Naccache and

Fremanteau, 1994). The physical characteristics could be delays of gates and wires in a circuit and/or variations in the temperature and supply voltage. These physical properties enable unclonability of the PUF functions (Kardas et al., 2011). Namely, it is infeasible to construct two PUFs with the same challenge-response behavior because the control over the manufacturing process of PUF is impossible.

All PUFs are subject to environmental variations such as temperature, supply voltage and electromagnetic interference, which affect their performance. A PUF function may produce slightly different responses for the same challenge because of environmental noise. However, this can be avoided by the help of Fuzzy Extractors which consist of a secure sketch that maps similar PUF responses to the same value (Dodis et al., 2008; Yevgeniy Dodis and Smith, 2007). Moreover, Fuzzy Extractors also include a randomness extractor, which extracts full-entropy bit-strings from a partially random source (van Herrewege et al., 2012).

There are several types of PUF implementations in the literature where most of them are integrated into electronic circuits (Suh and Devadas, 2007a). The most important examples are delay-based PUFs, memory based PUFs and coating PUFs. The delay-based PUFs exploit race conditions and frequency variations in the circuits of PUF (Gassend et al., 2002; Lee et al., 2004; Öztürk et al., 2008; Maiti et al., 2010; Suh and Devadas, 2007b; Tuyls and Batina, 2006). The memory-based PUFs are based on the instability of volatile memory cells, like SRAM , flip-flops and latches (Guajardo et al., 2007; van der Leest et al., 2010; Su et al., 2008; Maes and Verbauwhede, 2008; Holcomb, 2007). The coating PUFs use capacitance of a dielectric coating applied to the chip housing the PUF (Tuyls et al., 2006).

In Tuyls and Batina (2006), PUFs are used as a secure key derivation mechanism. Instead of storing the keys, which is previously produces, in non-volatile memory, they are derived from a PUF circuit whenever needed. This approach makes hardware-based attacks impractical. In Tuyls and Batina (2006) it is stated that a PUF based system can be implemented with less than 1000 gates. Also their intrinsic structure yields resistance against tampering. When the adversary tries to evaluate a PUF or IC for instance using the probes to measure the wire delays, the characteristics of that particular PUF would change. Thus, this physical attack will not give any information to the adversary. These properties make PUFs as an attractive tool for secure key derivation mechanisms in RFID systems. Several PUF function based authentication protocols have been proposed recently (Kulseng, 2009; Ranasinghe et al., 2004; Sadeghi et al., 2010; Devadas et al., 2008).

Similar to Tuyls and Batina (2006), Sadeghi et al. (2010) describe PUF functions in order to enhance security and privacy for RFID tags. This protocol provides destructive privacy in Vaudenay's (2007) formal framework. Nevertheless, Kardas et al. (2011) assume a stronger adversarial model for PUF based RFID system, where an adversary can access to volatile memory of the tag only once. They also show that Sadeghi et al.'s (2010) protocol does not provide a narrow destructive privacy.

In this paper, we utilize the ideal PUF mechanism, which is described in Kardas et al. (2011), in our proposed offline-RFID authentication protocol. To the best of our knowledge, such a usage of PUF is the first in the literature.

2.3. Preliminaries and notations

For a set S of any cardinality, $s \in_R S$ means s is chosen uniformly random among all elements of S. $y \in \{0,1\}^{\alpha}$ means y is any natural number such that y's bit length is at most α. For the case, $\alpha = *$, there is no restriction on bit length of y, i.e. y can be any natural number. A mapping $X : \{0,1\}^{\alpha} \rightarrow \{0,1\}^{\beta}$ means that X maps elements from $\{0,1\}^{\alpha}$ to $\{0,1\}^{\beta}$. Namely, the domain of X is $\{0,1\}^{\alpha}$ and the range of X is $\{0,1\}^{\beta}$. Let C be any algorithm, then $C(a) = b$ means, on input a, the algorithm C has b as output value. Let E be some event , then $Prob(E)$ denotes the probability that the event E happens. Moreover, $\mathrm{MSB}_a\{k\}$ denotes most significant a bits of binary representation of k.

Definition 2.1 (*Physically unclonable function (PUF)*). Let $k \in \mathbb{N}$ be a security parameter such that $\beta,\theta \in \mathbb{N}$ are polynomially bounded in k. An ideal PUF function is defined as $P : \{0,1\}^{\beta} \rightarrow \{0,1\}^{\theta}$ that holds the following properties:

- Any physical search trial to investigate the structure of P results in destruction of corresponding P. Namely, after the attack, the tag having this P cannot be evaluated anymore.
- Same inputs give same output result. Namely, let $P(a_1) = b_1$ and $P(a_2) = b_2$, if $a_1 = a_2$, then $Prob[b_1 = b_2] = 1$.
- Any probabilistic polynomial time adversary can distinguish between output of a P and random value with at most negligible probability.

As it can be understood from the definition, instead of studying in real PUFs, where for the same inputs they might produce slightly different outputs, we study with an idealized version of PUFs (Tuyls and Batina, 2006; Sadeghi et al., 2010; Kardas et al., 2011) which give same output results for same inputs.

Definition 2.2 (*Hash function*). Let $k \in \mathbb{N}$ be a security parameter such that $\gamma \in \mathbb{N}$ are polynomially bounded in k. Define hash function $H : \{0,1\}^* \rightarrow \{0,1\}^{2\gamma}$. Then H has the following properties:

- For any given input $m \in \{0,1\}^*$, the time required to calculate $H(m)$ is polynomially bounded.
- Hash functions are pre-image resistant. That means, for any $c \in \{0,1\}^{2\gamma}$, it is infeasible $m \in \{0,1\}^*$ such that $H(m) = c$.
- It is infeasible to find two different inputs giving the same output.
- Any probabilistic polynomial time adversary can distinguish between output of a H and random value with at most negligible probability.

3. Extended RFID security and privacy model

In this section, we present an improvement to a formal specification of the RFID security and privacy model proposed by the Vaudenay in ASIACRYPT 2007. We extend it by introducing notion of compromise of reader attacks and capability of the adversaries. In our model, an offline RFID system consists of a single operator $\mathcal{I}$, a secure backend system $\mathcal{DB}$, a set of readers $\mathcal{R}_i$, and a polynomial number of tags $\mathcal{T}$. Each tag $\mathcal{T}$ is assumed to be capable of performing basic cryptographic primitives such as hashing, symmetric encryption, PUF evaluations, and random number generation. On the other hand, each reader $\mathcal{R}_i$ can perform public-key cryptography and can also handle polynomial number of authentication protocols with different tags in parallel.

The rest of the section is organized as follows. We first describe a formal way of constructing an RFID system. Then, we present the capabilities of the adversary by introducing the oracles and the adversary classes. Lastly, we introduce definitions of correctness, security and privacy notions for analyzing a privacy-preserved offline RFID authentication protocol.

3.1. System model

A privacy preserving RFID authentication scheme $\mathcal{S}$ is usually composed of a set of procedures which either describe how to setup the system, the reader and the tags, or define the studied protocol. In our model, given k is the security parameter, one way to formalize these procedures is done as follows.

- *SetupServer* $(1^k)\rightarrow(sk_{Sr},pk_{Sr},\mathcal{DB}_{Sr})$. Given security parameter k, this generates a private/public key pair (sk_{Sr},pk_{Sr}). The key pk_{Sr} is publicly released whereas the key sk_{Sr} is to be stored in the server backend. It also creates an empty database $\mathcal{DB}_{Sr}$ which will later store the identifiers and permanent keys of all tags.
- *SetupReader* $(1^k)\rightarrow(sk_{\mathcal{R}},pk_{\mathcal{R}},\mathcal{DB}_{\mathcal{R}})$. Given security parameter k, this generates a private/public key pair $(sk_{\mathcal{R}},pk_{\mathcal{R}})$. The key $pk_{\mathcal{R}}$ is publicly released whereas the key $sk_{\mathcal{R}}$ is to be stored in the reader's backend. It also creates an empty database $\mathcal{DB}_{\mathcal{R}}$ in order to store the identifiers and temporary keys of all tags. Temporary keys are derived from the permanent keys of tags, identifier of reader, and a counter which specifies time-line. Therefore, each reader may store different keys for a given tag.
- $SetupTag_{pk_{\mathcal{R}}}(ID)\rightarrow(K,S)$. This creates an instance of the tag algorithm $\mathcal{T}$. The tag specific secret K and the initial state of the tag S are computed. $\mathcal{T}_{ID}$ is initialized with S and the pair (ID,K) is to be stored in the server's database when the tag is legitimate.
- *Ident* $[\mathcal{T}_{ID}:\mathcal{S};\mathcal{R}:sk_{\mathcal{R}},\mathcal{DB}_{\mathcal{R}};*:sk_{\mathcal{R}}]\rightarrow[\mathcal{T}_{ID}:-;\mathcal{R}:out_{\mathcal{R}}]$. A 2-party interactive protocol between $\mathcal{R}$ and $\mathcal{T}_{ID}$. $\mathcal{R}$ uses the common input, $\mathcal{DB}_{\mathcal{R}}$, and $sk_{\mathcal{R}}$, produces an output equal to $\perp$ if identification failed ($\mathcal{T}_{ID}$ is not legitimate) or some ID if $\mathcal{T}_{ID}$ is legitimate. And $\mathcal{R}$ may also update the database.

3.2. Adversary model

We now have an adversary $\mathcal{A}$ which is allowed to query a set of oracles, play polynomial number of games with the tags and also interact the system. Contrary to the Vaudenay-Model, our model assumes all readers and central server are to be separate entities. Our model also assumes the server is a secure and trusted entity, but the readers can be corrupted by some malicious adversaries. Similar to the Vaudenay-Model, tag $\mathcal{T}_{ID}$ can be compromised. A tag $\mathcal{T}_{ID}$ is always either a free tag or a drawn tag. The oracles, which are executed by the adversary to interact/play with system, are described as follows.

- $CreateTag^b(ID)$: This allows $\mathcal{A}$ to create a tag $\mathcal{T}_{ID}$ with a given ID. The value of b determines whether $\mathcal{T}_{ID}$ is legitimate ($b=1$) or not ($b=0$).
- $DrawTag(dist)\rightarrow(vtag_1,b_1,\ldots,vtag_n,b_n)$: This oracle allows $\mathcal{A}$ to get access to a set of tags that has been selected according to a given probability distribution $dist$. If ID_i is legitimate, b_i is set to 1. Otherwise, it is set to 0. *DrawTag* oracle also keeps tracks of the real identifier ID_i which is associated with a temporary tag identifier $vtag_i$ (i.e., $\mathcal{T}(vtag_i)\rightarrow ID_i$). All ID values and table $\mathcal{T}$ remain unknown to $\mathcal{A}$.
- $Free(vtag)$: This moves the drawn tag with temporary identity $vtag$ back to the set of free tags, so the adversary is no longer allowed to use $vtag$ in oracle calls.
- $Launch()\rightarrow\pi$: This makes the reader to start a new instance of the protocol π. The reader can run concurrently multiple instance of the protocol with different tags but each tag can only run one session of the protocol.
- $SendReader(m,\pi)\rightarrow m'$: (resp. $SendTag(m,vtag)\rightarrow m'$): This oracles sends a message m to a protocol session π for the reader (resp. to a tag $vtag$) and receives an answer m' which will be sent to the counterpart.
- *Result* (π): If the output on the instance of protocol π is $\perp$, this oracles return 0. Otherwise, it returns 1.
- $Corrupt(vtag)\rightarrow S$: This oracle enables $\mathcal{A}$ to get access to the current internal state S of the tag with internal identity $vtag$. If $vtag$ is no longer used, it is called that the tag is *destroyed*.

To play a game, the adversary $\mathcal{A}$ first setups the RFID system, that is provided a public key. Afterwards, $\mathcal{A}$, by following some rules of the game, utilizes the oracles and produces an output. $\mathcal{A}$ may win or lose depending on the rules.

Definition 3.1 (*Adversary classes (Vaudenay, 2007)*). We define Strong as the class of the class of adversaries who are able to access to all the above oracles. Destructive is the class of adversaries who never uses $vtag$ after querying $Corrupt(vtag)$, i.e. the tag with identifier $vtag$ is destroyed. Forward is the class of adversaries where *Corrupt* queries can only be followed by *Corrupt* other queries. Weak is the class of adversaries who never use *Corrupt* queries. Narrow is the class of adversaries who never uses *Result* query.

Remark 3.2. Clearly, we have following relation: Weak $\subseteq$ Forward $\subseteq$ Destructive $\subseteq$ Strong.

In accordance with the above definitions, we now recall the definitions off security and privacy of the Vaudenay-Model and introduce notion of *privacy+*.

3.3. Security, privacy, and privacy+

In this paper, we focus on only security and privacy, so the correctness property is not discussed further. The Vaudenay correctness definition can be combined with the new privacy definition, without compatibility issues. Also, we utilize the tag authentication and privacy definitions of Vaudenay model. However, for our new privacy definition, contrary to Vaudenay model, we consider compromise of both readers and tags.

3.3.1. Security

The security in Vaudenay-Model does not consider availability and cloning attacks, but it focuses on the attacks where the adversary intends to impersonate or forge a legitimate tag. The main objective of our security property is tag authentication which is formally defined as follows.

Definition 3.3 (*Tag authentication (Vaudenay, 2007)*). A RFID system achieves tag authentication if for every strong adversary $\mathcal{A}_s$, the probability of authenticating a non-valid tag is at most negligible.

Note that tag authentication is a crucial security property and thus must be kept even against strong malicious adversaries.

3.3.2. Privacy

The privacy definition of the Vaudenay-Model is very generic and, contingent to the adversary class (see Definition 3.1). It covers different notions of privacy. Vaudenay privacy experiment is described as follows. The privacy is based on the existence of a simulator $\mathcal{B}$, which is called blinder. $\mathcal{B}$ can simulate Launch, SendTag, SendReader, and Result oracles to $\mathcal{A}$. This simulator simulates any tag $\mathcal{T}_i$ or reader $\mathcal{R}_j$ without knowing real secrets. Note that, there is no interaction between $\mathcal{B}$ and $\mathcal{A}$, but inputs and corresponding output of oracles made by $\mathcal{A}$ have been seen by $\mathcal{B}$. The RFID system is secure if the probability that $\mathcal{A}$ distinguishes real RFID system from $\mathcal{B}$ is negligible. We can summarize privacy game as follows.

Let $\mathcal{P}$ is one of the adversary classes($\mathcal{P}\in\{\emptyset,$Narrow$\}\cup\{$Weak,Forward,Destructive,Strong$\}$) and $\mathcal{C}$ be challenger.

Privacy experiment $Exp^{priv}_{\mathcal{A}_\mathcal{P}}$:

1. $\mathcal{C}$ setups the system and sends 1^k, and $K_\mathcal{P}$ to $\mathcal{A}_\mathcal{P}$.
2. $\mathcal{A}_\mathcal{P}$ interacts with whole system according to her class $\mathcal{P}$.
3. $\mathcal{A}_\mathcal{P}$ analyzes system without the use of oracles.
4. $\mathcal{A}_\mathcal{P}$ receives a hidden table of DrawTag oracle.
5. If $\mathcal{A}_\mathcal{P}$ succeeds returns *true*, otherwise *false*. $Exp^{priv}_{\mathcal{A}_\mathcal{P}}$ wins $\mathcal{A}_\mathcal{P}$ returns *true*.

Definition 3.4 (*Privacy (Vaudenay, 2007)*). An RFID system $\mathcal{S}$, is said to unconditionally provide privacy class $\mathcal{P}$, if and only if for all adversaries $\mathcal{A}_{\mathcal{P}}$, $|Exp^{priv}_{\mathcal{A}_{\mathcal{P}}} - Exp^{priv}_{\mathcal{A}^{g}_{\mathcal{P}}}|$ is negligible.

3.3.3. Privacy+

Similar to RFID tags, the readers can also be corrupted by a malicious adversary because the readers in this context are mobile embedded devices, which have secure discontinuous access to the central database. In our model, we provide a new oracle for strong and destructive adversaries so as to enhance their capabilities.

Corrupt ($\mathcal{R}_i$): This oracle enables $\mathcal{A}$ to corrupt reader $\mathcal{R}_i$ and gets all internal states of that reader.

Remark 3.5. Once an adversary $\mathcal{A}$ uses (*Corrupt*($\mathcal{R}_i$)) oracle, $\mathcal{A}$ can interact tag $\mathcal{T}$ after the server's $\mathcal{DB}$ updates other reader's database and one of the updated readers run at least one successful protocol transaction with each tag $\mathcal{T}_i$ used in challenging phase of privacy game.

Considering compromise of readers, we define a new privacy notion, *privacy+*, for tag owner as follows.

Definition 3.6 (*Privacy+*). An RFID system $\mathcal{S}$ provides *privacy+* notion of $\mathcal{P}$ if $\mathcal{S}$ is still private a against an adversary $\mathcal{A}_{\mathcal{P}}$ even in the case of following conditions:

- Some of the readers are corrupted by $\mathcal{A}_P$.
- All readers except the corrupted ones are updated by the server.
- All tags have at least one successful interaction with one of the updated reader.

From Definition 3.6, it is clearly seen that once an adversary corrupts a reader in the system, she captures all the tag related information in the reader's database. Therefore, if the system does not update the remaining readers and the tags do not have successful interactions with one of the updated reader, then the adversary easily impersonates the victim reader and is able to trace any victim tag.

4. The PUF based RFID authentication protocol

In this section, we describe the authentication protocol which is composed of three phases; registration, reader update, and authentication phases.

4.1. The protocol

In this section, for a complete RFID system, we provide three phases; registration, update reader's database, and authentication.

4.1.1. Registration phase

Initially, in a stable RFID system, counter c_R and c_T are equal to each other. For each tag $\mathcal{T}_i$, Issuer $\mathcal{I}$ first setups $\mathcal{T}_i$ with a random $G_i \in \{0,1\}^{\beta}$, a unique *ID* of the tag $\mathcal{T}_i$, Id_i and the counter c_T. Then, $\mathcal{I}$ gets the secrets $S_i^1 \in \{0,1\}^{\theta}$, $S_i^2 \in \{0,1\}^{\theta}$ from $\mathcal{T}_i$ PUF evaluations. The record $\{Id_i, S_i^1, S_i^2\}$ is inserted into the central server's database $\mathcal{DB}$. After that, $\mathcal{I}$ setups each reader $\mathcal{R}_j$ in the systems with a unique *ID* of the reader $\mathcal{R}_j$, Id_j and the counter c_R. Lastly, the server starts secure communication with each reader to update their database. The update mechanism works as explained in the next subsection.

4.1.2. Update reader's database

The update protocol of reader's database is carried out during the registration phase and whenever a compromised reader is detected. The protocol works as follows. When the server starts a secure communication with the reader R_j, the server first gets Id_R, c_R from the target reader. The c_R is incremented by one. Then, for each tag $\mathcal{T}_i$ in $\mathcal{DB}$, the server computes a new record $\{Id_i, K_i^1, K_i^2\}$ where $K_i^1 = H(S_i^1, Id_R, c_R)$ and $K_i^2 = H(S_i^2, Id_R, c_R)$. Finally, the generated records and the new counter c_R are sent to $\mathcal{R}_j$ in order to update the reader's database and its counter.

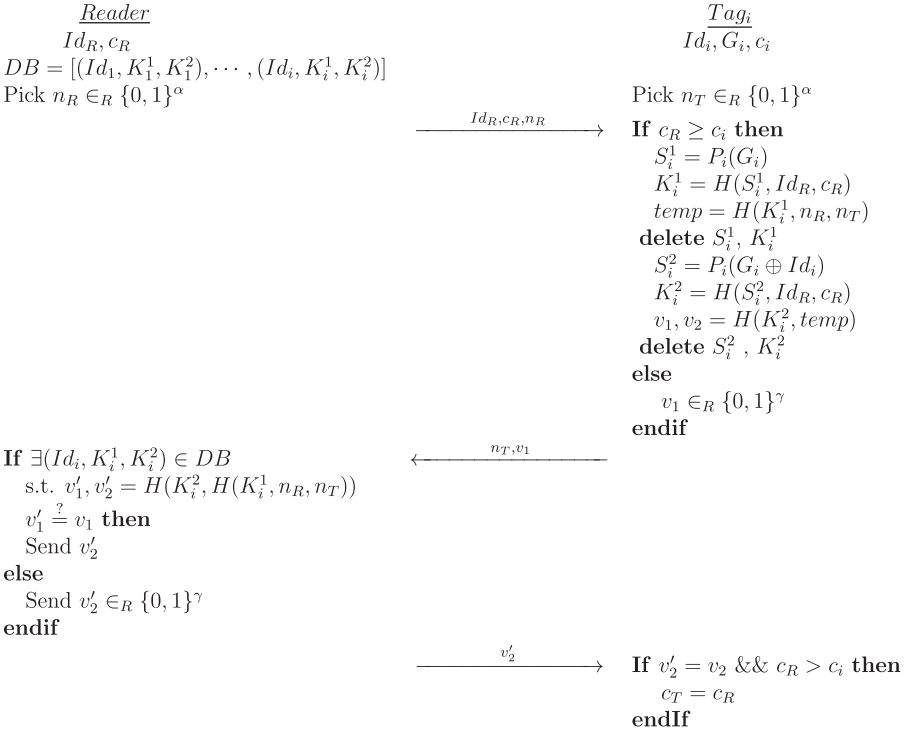

Fig. 2. The proposed authentication protocol.

4.1.3. Authentication

The protocol steps are summarized in Fig. 2. The detailed protocol steps are described as follows.

As soon as a tag $\mathcal{T}_i$ is in the authentication region, the reader chooses $n_R \in \{0,1\}^\alpha$ and sends it along with its Id_R and c_R to $\mathcal{T}_i$. Then, $\mathcal{T}_i$ first checks whether c_R is greater than or equal to c_i. If condition is not satisfied, $\mathcal{T}_i$ sends random bits to the reader. Otherwise, $\mathcal{T}_i$ generates a random $n_T \in \{0,1\}^\alpha$ and computes the secret value $S_i^1 = P_i(G_i)$. This is where the PUF function is used. Since P_i is specific to $\mathcal{T}_i$ and cannot be cloned, S_i^1 value can only be calculated by that tag. Session key (K_i^1) corresponding to that counter epoch (c_R) is calculated by concatenating S_i^1, Id_R and c_R and then by hashing the result. Then, a temporary hash is computed ($temp = H(K_i^1, n_R, n_T)$) and both secrets S_i^1, K_i^1 are deleted from the volatile memory. After that, the tag computes another pair of secrets by evaluating the function P_i with G_i ($S_i^2 = P_i(G_i \oplus Id_i)$) and a hash ($K_i^2 = H(S_i^2, Id_R, c_R)$). Finally, another hash is calculated over the concatenation of K_i^2 and *temp* to get the session vectors v_i and v_2. S_i^2 and K_i^2 are both deleted from the memory. The tag sends n_T and v_1 to the reader.

For each record $\{Id_i, K_i^1, K_i^2\}$ in the reader's database, the reader calculates $v_1', v_2' = H(K_i^2, H(K_i^1, n_R, n_T))$ and compares v_1' to v_1. If a match is found, then she identifies the tag and sends v_2' to $\mathcal{T}_i$. If no match is found in the database, then the reader sends random bits with bit-length of γ to $\mathcal{T}_i$. Finally, $\mathcal{T}_i$ compares v_2' that it has received from the reader to v_2. If they are equal, then the reader is genuine and the tag updates c_i if it is less than c_R. Otherwise, the tag figures out that reader is compromised.

5. Security analysis of the proposed scheme

Our proposed protocol utilizes the PUF mechanism presented in Kardas et al. (2011). This mechanism provides a secure key derivation for low-cost RFID tags so that it makes the RFID tags tamper-proof against malicious strong adversaries. We divide this section into two parts. In the first part, we state and prove some lemmas, which describe the capabilities of a strong adversary on PUF circuitry, are used in the proofs of security analysis results. In the second part, we provide security analysis of the protocol.

5.1. Security analysis tools

The following theorem and the proof are derived from Kardas et al. (2011).

Theorem 5.1. *Let S_i^1, S_i^2 be secrets of a tag $\mathcal{T}_i$ for some i in the above-mentioned protocol (see Fig.2). Assume that there is an adversary $\mathcal{A}$ with a full side-channel capability on the tag $\mathcal{T}_i$. If P_i is an ideal PUF, then $\mathcal{A}$ can only access either the secret S_i^1 or the secret S_i^2, but not both in $\mathcal{T}_i$.*

Proof (Sketch). The secret G_i and Id_i are fed into the P_i function to compute the real keys S_i^1 and S_i^2. The real keys only appear during the execution of the protocol. Notice that S_i^1 and S_i^2 never appear in the memory of $\mathcal{T}_i$ at the same time because S_i^1 is first used as an input of a one-way hash function, and then completely erased from the memory. Next, in a similar way, S_i^2 is computed by evaluating $P_i(G_i \oplus Id_i)$ and used in the hash function. Whenever $\mathcal{A}$ applies a side channel attack to $\mathcal{T}_i$, the physical characteristics of P_i will be broken and will no longer be evaluated correctly. If $\mathcal{A}$ applies side-channel attack to extract S_i^1 then the structure of P_i will be destroyed and S_i^2 cannot be computed. Similarly, if $\mathcal{A}$ applies side-channel attack to generated S_i^2 she cannot obtain S_i^1 since it is already erased. Hence, $\mathcal{A}$ can access either S_i^1 or S_i^2 but not both. □

Lemma 5.2. *Let $\mathcal{A}_d$ be destructive adversary and $\mathcal{T}_i$ be a target tag. During a protocol transcript, the advantage of $\mathcal{A}_d$'s of corrupting $\mathcal{T}_i$ before second deletion (delete S_i^2 , K_i^2) over corrupting $\mathcal{T}_i$ before first deletion (delete S_i^1 , K_i^1) is negligible.*

Proof. Let $\mathcal{A}_d$ corrupts tag $\mathcal{T}_i$ just before the first deletion, then the adversary gets the values of S^1, K^1, n_T, n_R and *temp* of the corresponding protocol run. Then, in order to beat the system in any aspect like security, privacy, the adversary has to find the values of S^2 or K^2. Thus, $\mathcal{A}_d$ has to solve a PUF function output or hash function output. Similarly, let assume $\mathcal{A}_d$ corrupts the tag just before the second deletion. Then the adversary knows the values of S^2, K^2, n_T, n_R, *temp*, v_1 and v_2 values of the corresponding protocol. Then, in order to beat the system, the adversary has to find the values of S^1 or K^1. Hence, similar to the above deduction, $\mathcal{A}_d$ has to solve a PUF function output or hash function output. Therefore, there is no real advantage difference for the adversary of corrupting a tag before first deletion and the second deletion. □

Lemma 5.3. *Let $\mathcal{A}_d$ be destructive adversary. Then $\mathcal{A}_d$'s investigating the system with many readers and tags gives him negligible advantage when it is compared with the situation that her investigating the system with just one reader and one tag.*

Proof. Before starting the proof, let us introduce some notations. Let ${}_iv_{1_j}^k$, ${}_iK_k^d$, ${}_in_{e_j}^k$ and ${}_iS^d$ be notations used at protocol description where i is tag index, k is reader index, j is protocol run index, $d \in \{1,2\}$ and $e \in \{R,T\}$. Assume that there are l readers and n tags in the system where l and n are polynomially bounded. Moreover, the number of protocol run between reader k' and tag i' is $m_{i'}^{k'}$ for $k' \in \{1,2,\ldots,l\}$ and $i' \in \{1,2,\ldots,n\}$ before corruption of tag i'. Besides, let the adversary starts a protocol run between reader k' and tag i' $p_{i'}^{k'}$ times and starts protocol run between the tag i' and himself as a replacement of reader k' $r_{i'}^{k'}$ times for $k' \in \{1,2,\ldots,l\}$ and $i' \in \{1,2,\ldots,n\}$ before corruption of tag i'. Furthermore, let the adversary starts a protocol run between himself as a replacement of tag i' $t_{i'}^{k'}$ times for $k' \in \{1,2,\ldots,l\}$ and $i' \in \{1,2,\ldots,n\}$ before corruption of tag i'. Moreover, let $m = \max_{i',k'}\{m_{i'}^{k'}\}$, $p = \max_{i',k'}\{p_{i'}^{k'}\}$, $r = \max_{i',k'}\{r_{i'}^{k'}\}$, $t = \max_{i',k'}\{t_{i'}^{k'}\}$ and let $M = m+p+r+t$. Note that M is polynomially bounded as m, p, r and t values are polynomially bounded. After $\mathcal{A}_d$' observing or corrupting the tags, $\mathcal{A}_d$ has at most $k.M.l$ ${}_iv_{1_j}^k$ values such that ${}_iv_{1_j}^k = \mathrm{MSB}_\gamma\{H({}_iK_k^2, H({}_iK_k^1, {}_in_{R_j}^k, {}_in_{T_j}^k))\}$.

By Lemma 5.2, let assume that all tags are corrupted before the second deletion. Let us fix tag T_y and reader R_z. In order to prove the lemma, we have the figure out how much advantage $\mathcal{A}_d$ gets to guess the value of ${}_yv_{1_{m_y^z+1}}^z$ by observing, creating or corrupting all protocol runs except all protocol runs between (T_y, R_z) pair and T_y and himself as a replacement of R_z and reader R_z and himself as a replacement of T_y. Now, let us take a pair $(u,w) \neq (y,z)$. There are two cases to consider. First of all, let $u=y$ and $w \neq z$. Then if the adversary finds the value of S_u^1, then the adversary can calculate the value of ${}_uK_w^1$. Otherwise, the adversary has to find relation the among keys or S values or resulting v_1 values. The maximum success probability is $M(l-1)(1/2^{\theta-1} + 1/2^{4\gamma} + 1/2^{2\gamma} + 1/2^{\gamma})$. Let C denotes this probability. As a second case, if $u \neq y$, then $\mathcal{A}_d$ again has to find relation the among keys or S values or resulting v_1 values. However, in this case, the maximum success probability is

$Ml(n-1)((1/2^{\theta}+(1/2^{2\theta}+1/2^{2\gamma}+1/2^{\gamma}+1/2^{2\gamma}\max\{1/2^{2\gamma},1/2^{\theta}\})$ Let D denotes this probability and let $\beta=\max\{\theta,\gamma\}$. Then $C+D\leq M(\ln-1)(1/2^{\beta-2})$. Since n, l and M are polynomially bounded and the value of β is sufficiently large, the maximum total advantage is negligible. □

In the next section, these theorem and lemmas will be used in the proof of security and privacy analysis of the proposed protocol.

5.2. Security and privacy analysis

In this section, we first prove that our protocol achieves tag authentication and destructive privacy. Then, we also prove that our protocol satisfies reader authentication and destructive *privacy+*.

Theorem 5.4. *The RFID protocol demonstrated in Fig.2 achieves tag authentication if H is a hash function (Definition2.2).*

Proof. Assume to the contrary, the protocol described in Fig. 2 does not achieve tag authentication. That means, the adversary $\mathcal{A}_s$ behaves like a legitimate tag to a legitimate reader with non-negligible probability. By Lemma 5.3, let us assume that there are only one legitimate reader R and one tag T in the system and for simplicity, R is not updated throughout the proof. By the argument above, the strong adversary $\mathcal{A}_s$ does not need to apply *CreateTag*, *DrawTag* and *Free* oracles. Let $\mathcal{A}_s$ observed protocol runs between the reader and the tag m times. Moreover, let $\mathcal{A}_s$ uses *SendReader*(π) oracle p times to start protocol run between the reader and the tag and uses *SendTag* oracle r times to start protocol run between himself and the tag. Here, the values of m, p and r are polynomially bounded. Note that, $\mathcal{A}_s$ can use *Corrupt* oracle at most one time as the tag T has PUF function inside. However, we assume that $\mathcal{A}_s$ applies this oracle exactly one time as this assumption increases his chances to win the game.

Let the adversary has chance to impersonate the corresponding tag k times, where k is polynomially bounded. In order to achieve the impersonation, at each round $\mathcal{A}_s$ creates u_i triple $(S^2,K^2,v_{1_i})_j$, where $i\in\{1,\ldots,k\}$, $j\in\{1,\ldots,u_i\}$ and each u_i is polynomially bounded. Note that, if the space of PUF is smaller than the space of hash function, these triples are created on guesses of $\mathcal{A}_s$ on the values of S^2s. Otherwise, they are created on guesses of $\mathcal{A}_s$ on the values of K^2s. Since the hash function is pre-image resistant, guesses are not made on the third component. The adversary checks whether they are true or not at each triple at each impersonation trial based on the protocol transcripts that have been reached so far. If the adversary could not find any match at the end of calculations, then the adversary just guesses the value of v_{1_i}.

Let $M=m+p+r$ and $U=\max\{u_1,u_2,\ldots,u_k\}$ and so M and U are polynomially bounded. Moreover, let $\beta=\max\{\theta,2\gamma\}$. By Lemma 5.3, let us assume that corruption made before the first deletion. Note that, if the value of n_R sent by the reader at each impersonation trial is one of those n_R values which is used at previous protocol runs, then the success probability of destroying tag authentication is 1 by choosing corresponding n_T value. However, the probability of realization of this scenario is at most $1-(1-M/2^{\alpha})^k$. Otherwise, the probability of $\mathcal{A}_s$'s generating correct value of v_1 in at least one impersonation trial is at most $2-\prod_{j=0}^{kU-1}(1-1/(2^{\beta}-j))+(1-1/2^{\gamma})^k$. In order to see the total probability is minimum, let us use $ln(1-x)\approx -x$ for small x values. Then the success probability is at most $3-e^{-Mk/2^{\alpha}}-e^{-kU/(2^{\beta}-kU)}-e^{-k/2^{\gamma}}$. By contradiction assumption, this probability is non-negligible, so at least one of the values of M, U and k is non-negligible. However, this contradicts with the fact that M, U and k are polynomially bounded. □

Theorem 5.5. *The RFID protocol demonstrated in Fig.2 achieves destructive privacy if the protocol achieves tag authentication, P is a PUF (Definition2.1) and H is hash function (Definition2.2).*

Proof. Assume to the contrary, the system does not achieve destructive privacy property. That means, there is a destructive adversary $\mathcal{A}_d$, who can distinguish between the real RFID system and the system which is simulated by a blinder $\mathcal{B}$ with non-negligible probability. Note that, $\mathcal{B}$ simulates *Launch*, *SendTag*, *SendReader* and *Result* oracles without knowing the tag and the reader secrets.

More formally, let there exists an oracle $\mathcal{O}^{dest}$ such that $\mathcal{A}_d$ plays the following game with this oracle. $\mathcal{O}^{dest}$ chooses a number $b\in_R\{0,1\}$, if $b=1$, real RFID system is used, otherwise $\mathcal{B}$ simulates the system. $\mathcal{A}_d$ watches the system for polynomially bounded number of times and the adversary is allowed to use corrupt oracle as well. At the end, $\mathcal{A}_d$ guesses a number b'. If $|Prob(b=b')|=\frac{1}{2}+a$ where a is non-negligible, $\mathcal{A}_d$ wins the game, else the adversary loses. Note that, by contradiction assumption, $\mathcal{A}_d$ wins the game.

Let start with how $\mathcal{B}$ evaluates oracles:

- **Launch()**: Evaluated in a trivial way.
- **SendTag**($Id_R,c_R,n_R,vtag$): The output is $n_T\in_R\{0,1\}^{\alpha}$, $v_1\in_R\{0,1\}^{\gamma}$.
- **SendReader**(π): The output is $n_R\in_R\{0,1\}^{\alpha}$ and the real values of Id_R and c_R.
- **SendReader**((n_T,v_1), π): The output is $v_2\in_R\{0,1\}^{\gamma}$.
- **SendTag**(v_2): Returns no output.
- **Result**(π): If π is generated by *Launch* oracle and the protocol transcript is generated by *SendTag* and *SendReader* oracles, the output is 1. If one of the conditions does not hold, then the output is 0.

By Lemma 5.3, let us assume that there are only one legitimate reader R and one tag T in the system and for simplicity, R is not updated throughout the proof. Let the system is run n_1 times only by real RFID system or the blinder according to b value the oracle $\mathcal{O}^{dest}$ chooses and let at n_1th run, $\mathcal{A}_d$ applies *Corrupt* oracle to the tag T. By Lemma 5.2, let assume that corruption is applied before second deletion. Thus, $\mathcal{A}_d$ have the knowledge of $\{(n_R^1,n_T^1,v_1^1,v_2^1)$, $(n_R^2,n_T^2,v_1^2,v_2^2),\ldots,(n_R^{n_1},n_T^{n_1},v_1^{n_1},v_2^{n_1})\}$ and S^2, K^2, $temp^{n_1}$.

There are five cases to consider. First two cases are $\mathcal{A}_d$'s determining the value of S^1 or K^1. The probability of these happening is $1/2^{\theta}$ and $1/2^{2\gamma}$, respectively. The third case is $\mathcal{A}_d$'s determining value of temp at least one protocol run. The probability of this case is $1-(1-1/2^{2\gamma})^{n_1}$. The fourth possibility is $\mathcal{A}_d$'s determining value of v_1 at least one protocol run. The probability of this case is $1-(1-1/2^{\gamma})^{n_1}$. The last case is $\mathcal{A}_d$'s determining value of v_2 being random.

By contradiction assumption, as $\mathcal{A}_d$ wins the game against the oracle, then one of the four probabilities above is non-negligible or realization of the last case is non-negligible. However, with sufficiently large θ and γ values, the four possibilities listed above are negligible. Thus, by assumption, the probability of $\mathcal{A}_d$'s determining v_2 value being random is non-negligible. However, this statement contradicts with Theorem 5.4, i.e. contradiction to tag authentication. Thus, proposed protocol satisfies destructive privacy property. □

Theorem 5.6. *The RFID protocol demonstrated in Fig.2 achieves reader authentication if H is a hash function (Definition2.2).*

Proof. By Lemma 5.3, without loss of generality, there are one reader R and one tag T in the system. Let before $\mathcal{A}_d$ starts a protocol run with T, $\mathcal{A}_d$ observed previous p run of tag T and R. As a result of the observations, $\mathcal{A}_d$ gets the following protocol transcripts $(n_{R_1},Id_R,c_R,n_{T_1},v_{1_1},v_{2_1}),\ldots,(n_{R_p},Id_R,\ c_R,n_{T_p},\ v_{1_p},v_{2_p})$. Note that, $\mathcal{A}_d$'s aim is to impersonate the reader R by convincing T. The most logical move for $\mathcal{A}_d$ is choosing one of the values of $n_{R_1},n_{R_2},\ldots,n_{R_p}$ as n_R value. W.l.o.g., let $\mathcal{A}_d$ sends n_{R_1},Id_R,c_R to tag T. There are two cases to consider. First of all, if T responds with n_{T_1},v_{1_1}, then the probability that the adversary returns the correct value of v_2 is 1. If this is not the case, then there are two cases, which are $\mathcal{A}_d$'s calculating the value of v_2 or guess the value of v_2. For the first case, $\mathcal{A}_d$ has to now at least one of the values of (S^1,S^2), (S^1,K^2), (K^1,S^2) and (K^1,K^2). The corresponding probabilities are $1/2^{2\theta}$, $1/2^{\theta+2\gamma}$, $1/2^{\theta+2\gamma}$, $1/2^{4\gamma}$. Let $q=\max\{1/2^{2\theta},1/2^{\theta+2\gamma},1/2^{4\gamma}\}$. For the second case, $\mathcal{A}_d$ guess the value of v_2 with possibility $1/2^{\gamma}$. Thus, the probability that $\mathcal{A}_d$'s convincing the tag T is $1/2^{\alpha}+(2^{\alpha}-1)/2^{\alpha}\max\{m,1/2^{\gamma}\}$. Note that the probability given above negligible provided that α, γ and θ are large enough. □

Theorem 5.7. *The RFID protocol illustrated in Fig.2 provides destructive privacy+ if the protocol achieves tag authentication, P is a PUF (Definition2.1) and H is hash function (Definition 2.2).*

Proof. Assume that a reader R_C is compromised. Then the adversary $\mathcal{A}_{R_C}$ gets the information $(Id_1,K_1^1,K_2^1,\ldots,Id_n,K_n^1,K_n^1)$ of tags T_i for $i=1,2,\ldots,n$, where n is polynomially bounded. Due to the assumption at Remark 3.5, after DB updates all other reader, the value of c_R changes. Moreover, as the value of c_R changed, then the values of $K_i^1=H(S_i^1,Id_R,c_R)$ and $K_i^2=H(S_i^2,Id_R,c_R)$ for $i=1,\ldots,n$ are changed. Note that, the adversary $\mathcal{A}_{R_C}$ does not have the values of S_i^1,S_i^2 for $i=1,\ldots,n$ due to the pre-image resistance property of hash function. Thus, from previous knowledge of $(Id_1,K_1^1,K_2^1,\ldots,Id_n,K_n^1,K_n^1)$, $\mathcal{A}_{R_C}$ cannot calculate new K_i^1,K_i^2 values for $i=1,\ldots,n$. Therefore, the only legitimate information that $\mathcal{A}_{R_C}$ has after system re-setup is Id's of all tags. Therefore, by Theorems 5.4 and 5.5, the system is private against the adversary $\mathcal{A}_{R_C}$. Hence, the RFID system provides destructive *privacy+*. □

5.3. Security and privacy and performance comparisons

Considering memory storage for tag identifiers or keys and other information, our protocol requires 3β-bit (Id, G, and c) memory in tag side where β is at most the length of a hash output. Contrary to tags, server has no limited resource, so we do not concern on the server-side memory usage. In terms of computational cost, our protocol requires at most four hash computation and two PUF evaluations overhead at the tag side. On the other hand, the computational complexity at the server side at most $\mathcal{O}(n)$, where n is the number of tags in the system.

Table 1 summarizes the comparison of our protocol with other protocols, where n is the number of tags in the system. Our protocol and Avoine et al. (2009) have reader authentication whereas only our protocol provides destructive privacy, and destructive *privacy+*. While considering computational complexity at the server side, the complexity required for each scheme is roughly proportional to the number of tags in the system.

Table 1
Security and privacy and performance comparisons.

Protocol	Tan et al. (2008)	Avoine et al. (2009)	Our protocol
Reader authentication	+	+	+
Privacy	–	Weak	Destructive
Privacy+	–	Weak+	Destructive+
Crypto primitive	Hash	Hash	Hash and PUF
Reader complexity	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$

6. Conclusion

In this paper, we first extends Vaudenay's (2007) adversarial model for offline RFID system and introduce the notion of compromise reader attacks. We define the notion of *privacy+* and the game behind this privacy notion. Then, we propose a RFID mutual authentication protocol based on PUF functions. We prove that our protocol achieves destructive privacy for tag owner. To the best our knowledge, it is the first protocol which utilizes only symmetric cryptographic primitives and PUF functions and provides destructive *privacy+* even in case of compromising reader attacks. Our protocol can be efficiently implemented in low-cost RFID tags because the tags need only lows cost cryptographic primitives such as hash and PUF functions.

References

Avoine G. Adversarial model for radio frequency identification. Tech. rep. Swiss Federal Institute of Technology (EPFL), Security and Cryptography Laboratory (LASEC); 2005.

Avoine G, Lauradoux C, Martin T. When compromised readers meet RFID. In: Youm H, Yung M, editors. Workshop on information security applications (WISA'09). Lecture notes in computer science, vol. 5932. Busan, Korea: Springer; 2009. p. 36–50.

Avoine G, Coisel I, Martin T. A privacy-restoring mechanism for offline RFID systems. In: Proceedings of the 5th ACM conference on security and privacy in wireless and mobile networks (WISEC '12). New York, NY, USA: ACM; 2012. p. 63–4.

Baudron O, Boudot F, Bourel P, Bresson E, Corbel J, Frisch L, et al. GPS—an asymmetric identification scheme for on the fly authentication of low cost smart cards. 2001.

Burmester M, Le Tv, Medeiros Bd. Provably secure ubiquitous systems: universally composable RFID authentication protocols. In: IEEE conference on security and privacy for emerging areas in communication networks (secureComm 2006). Baltimore, MD, USA: IEEE Computer Society; 2006. p. 1–10.

Canard S, Coisel I, Etrog J, Girault M. Privacy-preserving RFID systems: model and constructions. Cryptology ePrint archive, report 2010/405; 2010.

Daniel E. Holcomb, Wayne P. Burleson, and Kevin Fu. Initial SRAM state as a fingerprint and source of true random numbers for RFID tags. In: Conference on RFID security (RFIDSec); 2007.

Deng RH, Li Y, Yung M, Zhao Y. A new framework for RFID privacy. In: Gritzalis D, Preneel B, Theoharidou M, editors. 15th European symposium on research in computer security (ESORICS 2010). Lecture notes in computer science, vol. 6345. Athens, Greece: Springer; 2010. p. 1–18.

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: IEEE international conference on RFID, 2008; 2008. p. 58–64.

Dodis Y, Ostrovsky R, Reyzin L, Smith A. Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. SIAM Journal on Computing 2008;38(1):97–139 ISSN: 0097-5397.

Gassend B, Clarke D, van Dijk M, Devadas S. Controlled physical random functions. In: Proceedings of the 18th annual computer security applications conference, 2002; 2002. p. 149–60, ISSN: 1063-9527, http://dx.doi.org/10.1109/CSAC.2002.1176287.

Guajardo J, Kumar SS, Schrijen G-J, Tuyls P. FPGA intrinsic PUFs and their use for IP protection. In: Proceedings of the 9th international workshop on cryptographic hardware and embedded systems (CHES '07). Berlin/Heidelberg: Springer-Verlag; 2007. p. 63–0, ISBN: 978-3-540-74734-5, http://dx.doi.org/10.1007/978-3-540-74735-2_5, URL 〈http://dx.doi.org/10.1007/978-3-540-74735-25〉.

Ha J, Moon S, Zhou J, Ha J. A new formal proof model for RFID location privacy. In: Jajodia S, Lopez J, editors. 13th European symposium on research in computer security (ESORICS 2008). Lecture notes in computer science, vol. 5283. Malaga, Spain: Springer; 2008. p. 267–81.

Hermans J, Pashalidis A, Vercauteren F, Preneel B. A new RFID privacy model. In: 16th European symposium on research in computer security (ESORICS 2011). Lecture notes in computer science. Leuven, Belgium: Springer; 2011.

Juels A, Weis SA. Defining strong privacy for RFID, ACM Transactions on Information and System Security (TISSEC). October 2009; 13(1):1–23. 〈http://dx.doi.org/10.1145/1609956.1609963〉.

Kardas S, Kiraz MS, Bingol MA, Demirci HA. Novel RFID distance bounding protocol based on physically unclonable functions. In: RFID. Security and Privacy,

Lecture Notes in Computer Science, vol. 7055. Berlin, Heidelberg: Springer; 2012. p. 78–93.
Kulseng L. Lightweight mutual authentication, ownership transfer, and secure search protocols for RFID systems. Master's thesis. Electrical & Computer Engineering Department, Iowa State University; 2009.
Lai J, Deng RH, Li Y. Revisiting unpredictability-based RFID privacy models. In: Zhou J, Yung M, editors. Proceedings of the 8th international conference on applied cryptography and network security (ACNS 2010). Lecture notes in computer science, vol. 6123. Beijing, China: Springer; 2010. p. 475–92.
Lee J, Lim D, Gassend B, Suh G, van Dijk M, Devadas S. A technique to build a secret key in integrated circuits for identification and authentication applications. In: Symposium on VLSI circuits, 2004. Digest of technical papers; 2004. p. 176–9, http://dx.doi.org/10.1109/VLSIC.2004.1346548.
Maes PTR, Verbauwhede I. Intrinsic PUFs from Flip-flops on reconfigurable devices. In: 3rd Benelux workshop on information and system security (WISSec 2008); 2008.
Maiti A, Casarona J, McHale L, Schaumont P. A large scale characterization of RO-PUF. In: HOST; 2010. p. 94–9.
Naccache D, Fremanteau P. Unforgeable identification device, identification device reader and method of identification, Patent-EP0583709 (1994).
Oren Y, Feldhofer M. WIPR—a public key implementation on two grains of sand. In: Dominikus S, editor. Workshop on RFID security 2008; 2008. p. 15–27.
Öztürk E, Hammouri G, Sunar B. Towards robust low cost authentication for pervasive devices. In: Proceedings of the 2008 6th annual IEEE international conference on pervasive computing and communications (PERCOM '08). Washington, DC, USA: IEEE Computer Society; 2008. p. 170–8, ISBN: 978-0-7695-3113-7.
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 (ASIACCS '08). New York, NY, USA: ACM; 2008. p. 292–9, ISBN: 978-1-59593-979-1.
Ranasinghe DC, Engels DW, Cole PH. Security and privacy: modest proposals for low-cost RFID systems. In: Proceedings of the auto-ID labs research workshop systems; 2004.
RFIDea, Engineering & applications in electronic traceability; 2012 〈http://www.rfidea.com〉.
Sadeghi A-R, Visconti I, Wachsmann C. PUF-enhanced RFID security and privacy. In: Secure component and system identification (SECSI'10), Cologne, Germany; 2010.
Su Y, Holleman J, Otis BP. A digital 1.6 pJ/bit chip identification circuit using process variations. IEEE Journal of Solid-State Circuits 2008;43(1):69–77.
Suh G, Devadas S. Physical unclonable functions for device authentication and secret key generation. In: 44th design automation conference, 2007 (DAC '07). ACM/IEEE; 2007. p. 9–14, ISSN: 0738-100X.
Suh GE, Devadas S. Physical unclonable functions for device authentication and secret key generation. In: DAC '07: proceedings of the 44th annual design automation conference. New York, NY, USA: ACM; 2007. p. 9–14, ISBN: 978-1-59593-627-1.
Tan CC, Sheng B, Li Q. Secure and serverless RFID authentication and search protocols. IEEE Transactions on Wireless Communications 2008;7(4):1400–7.
Tuyls P, Batina L. RFID-tags for anti-counterfeiting. In: Topics in cryptology (CT-RSA 2006). Lecture notes in computer science, vol. 3860; 2006. p. 115–31.
Tuyls P, Schrijen G-J, Škorić B, van Geloven J, Verhaegh N, Wolters R. Read-proof hardware from protective coatings. In: Proceedings of the 8th international conference on cryptographic hardware and embedded systems (CHES'06). Berlin/Heidelberg: Springer-Verlag; 2006. p. 369–83, ISBN: 3-540-46559-6, 978-3-540-46559-1, http://dx.doi.org/10.1007/11894063_29.
van der Leest V, Schrijen G-J, Handschuh H, Tuyls P. Hardware intrinsic security from D flip-flops. In: Proceedings of the 5th ACM workshop on scalable trusted computing (STC '10). New York, NY, USA: ACM; 2010. p. 53–62, ISBN: 978-1-4503-0095-7, http://dx.doi.org/10.1145/1867635.1867644.
van Deursen T, Mauw S, Radomirović S. Untraceability of RFID protocols. In: Onieva JA, Sauveron D, Chaumette S, Gollmann D, Markantonakis C, editors. Workshop on information security theory and practice (WISTP'08). Lecture notes in computer science, vol. 5019. Sevilla, Spain: Springer; 2008. p. 1–15.
van Herrewege A, Katzenbeisser S, Maes R, Peeters R, Sadeghi A-R, Verbauwhede I, et al. Reverse fuzzy extractors: enabling lightweight mutual authentication for PUF-enabled RFIDs. In: Financial cryptography (FC 2012). Lecture notes in computer science. Springer; 2012.
Vaudenay S. On privacy models for RFID. In: Proceedings of the advances in crypotology 13th international conference on theory and application of cryptology and information security (ASIACRYPT '07). Berlin/Heidelberg: Springer-Verlag; 2007. p. 68–87, ISBN: 3-540-76899-8, 978-3-540-76899-9.
Yevgeniy Dodis LR, Smith A. Fuzzy extractors. In: Security with noisy data. Springer-Verlag; 2007.