Communication Papers of the 19th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 117–124
DOI: 10.15439/2024F5622
ISSN 2300-5963 ACSIS, Vol. 41

Analyzing the Privacy of a Healthcare RFID Authentication Protocol

Ștefana Gheorghiță
0009-0004-2925-3666
Faculty of Computer Science
"Alexandru Ioan Cuza" University of Iași, Iași, Romania
Email: gheorghitastefana@gmail.com

Anca-Maria Nica
0000-0002-3808-572X
Department of Computer Science
"Alexandru Ioan Cuza" University of Iași, Iași, Romania
Email: contact@ancamarianica.ro

***Abstract*—With the growing use of RFID systems in IoT environments, it is crucial for these systems to be highly efficient, reducing costs while also maintaining functionality. As technology evolves, adversaries' capabilities also increase, highlighting the necessity to consider all potential vulnerabilities that could be exploited, especially in terms of security and privacy. One particular case requiring attention is the use of temporary variables, which can inadvertently provide valuable information to an adversary. This scenario will be exemplified and addressed through the case of an RFID mutual authentication scheme designed for the healthcare field.**

I. Introduction

ONE essential concept that emerged in the last few years is the Internet of Things (IoT), which defines a revolutionary way of interacting with technology.

As the demand for appropriate devices that can implement IoT increased, RFID systems emerged as a potential solution due to their versatility based on specific architectures. RFID tags range from simple passive ones with minimal computational complexity (but also low production costs) to more complex, smart tags designed to perform various functions.

These considerations led to a significant increase in the number of patents and research papers on this subject [13]. One of the most exploited subdomains is represented by authentication protocols. As the number of proposed protocols is constantly growing, the technology and requirements change and it can be hard to maintain a detailed insight into the security and privacy requirements of such schemes. This led to the existence of many schemes incapable of achieving satisfactory privacy and/or security for real-life use. Key reason for this is the lack of suitable adversarial models in the analysis of the proposed protocols.

Contribution: The purpose of this paper is to draw attention to the privacy issues associated with RFID schemes and emphasize the necessity for a suitable model in analyzing security and privacy. This paper will focus on the vulnerabilities emerged from the use of global temporary variables. As such, we analyze the scheme presented in [8] using the Vaudenay model, one of the most notable security and privacy models for RFID systems. The vulnerabilities identified through this analysis will be highlighted. Additionally, we will address these problems and propose enhancements to improve the privacy of the scheme.

Paper structure: The current paper is structured in four sections. Section 1 contains the introduction. Section 2 summarizes one of the widely accepted security and privacy RFID models, the Vaudenay model. Section 3 focuses on the protocol to be analyzed. The protocol will be briefly described, followed by an analysis of potential attacks on the scheme. Finally, Section 3 will discuss enhancements aimed at improving the privacy of the scheme. The final section concludes the paper.

II. The Vaudenay Model

As RFID systems tend to be used on a larger scale, there is a need to find a balance between reducing costs in the manufacturing and utilization of tags and adhering to security requirements. Since implementing strong cryptographic algorithms would be costly and impractical for tags used at a larger scale, the focus has shifted towards finding solutions that take into account the inevitable risks to which such a system is exposed [7].

The most important attributes for an RFID system to maintain are privacy and security, as highlighted in various papers [1], [4]. Different concerns regarding these properties arise due to constraints imposed on the computational power of the tags.

From this perspective, the security and privacy model used in the analysis is crucial for defining an RFID scheme. Many models have been proposed to offer generality and suitability for simulating practical risks. Among these, two widely accepted models are the one proposed by Vaudenay in [1] and its extension for mutual authentication resulting from collaboration with Paise in [2], as well as the model proposed by Hermans et al. in [3] and its extension for multiple readers, included in [4]. The model proposed by Hermans et al. will be referred to as the HPVP model, based on the initials of its authors.

This section will conduct a summary of the Vaudenay model.

A. RFID System

An RFID system is defined by the existence of two main components:

- $\mathcal{T}$ - the set of tags (with the role of transponders), devices characterized by limited memory and computational

power; each tag has a corresponding unique identifier (ID) stored in the database of the reader; every tag stores its own state S, which may or may not include the associated ID;
- $\mathcal{R}$ - the set of readers, which have the role of transceivers (devices that both transmit and receive signals); in most cases, the focus is on the situation where there is only one reader;

As described in [1], we should also consider the three main algorithms which are necessary for an RFID system:

1) $SetupReader(1^k)$: using k (the security parameter), the pair (public key (K_P), secret key (K_S)) of the reader is generated;
2) $SetupTag_{K_P}(ID)$: for the tag with the identifier ID, the initial state is provided and stored in the memory of the tag and the corresponding secret of the tag is generated and added alongside the ID in the database of the reader if the tag is legitimate;
3) $Protocol$: the protocol between the reader and the tag, which ends with an **Output** from the reader ($\perp$ if the tag is not legitimate and ID if the tag is legitimate);

For an RFID scheme, the **Output** should be correct with overwhelming probability, meaning that for a legitimate tag the output is ID, otherwise being $\perp$.

B. Adversarial Model

The adversary is defined as an algorithm which can interact with the RFID system on the basis of the following oracles:

$CreateTag^b(ID)$**:** Based on the value of b, a legitimate ($b = 1$) or a illegitimate ($b = 0$) tag is created. This oracle calls $SetupTag$ and in the case of a legitimate tag, it is added in the central database. If the value of b is omitted, it means $b = 1$.

$Launch() \rightarrow \pi$**:** This oracle launches a new protocol session denoted by π.

$DrawTag(d) \rightarrow (vtag_1, b_1, ..., vtag_n, b_n)$**:** This oracle takes a probability distribution d as input and, based on that, creates n virtual tags, each of them being associated to a bit b, corresponding to the legitimacy of the tag. Already drawn tags cannot be drawn again (the oracle returns $\perp$). $\perp$ is also returned if the tags given as parameters do not exist. Together with this oracle, a hidden table $\mathcal{T}$ is created to store the real identity of the virtual tags: $\mathcal{T}(vtag_i) = ID_i$, where ID_i is the identifier of the real tag referenced by $vtag_i$.

$Free(vtag)$**:** This oracles frees the virtual tag $vtag$.

$SendTag(m, vtag) \rightarrow m^{'}$**:** This oracle is used to send a message m to the tag denoted by the virtual identity $vtag$ and to get its response in the form of $m^{'}$.

$SendReader(m, \pi) \rightarrow m^{'}$**:** In the protocol session π, a message m is sent to the reader and the response of it is returned.

$Execute(vtag) \rightarrow (\pi, transcript)$**:** This oracle simulates an entire protocol session by initially calling $Launch$ and then using subsequent calls of $SendReader$ and $SendTag$. It returns the pair containing the session π and the list of the successive protocol steps.

$Result(\pi) \rightarrow a$**:** This oracle is used for getting the result of the protocol session π regarding the authentication status of the tag. When the session is complete, if the tag is not considered legitimate (the output is $\perp$), it returns 0, otherwise it returns 1.

$Corrupt(vtag) \rightarrow \mathcal{S}$**:** This oracle is used for corrupting the tag referred by the virtual identity $vtag$. The returned value is the current state of the tag.

C. Adversary Classes

The Vaudenay model defines the following adversary classes:

- **weak adversary**: cannot access the $Corrupt$ oracle;
- **forward adversary**: if the adversary has used the $Corrupt$ oracle, then the only accessible oracle after that is $Corrupt$;
- **destructive adversary**: after a tag is corrupted, the tag is considered destroyed (the adversary cannot interact with it anymore);
- **strong adversary**: there are no restrictions imposed on the use of the oracles;

Additionally, **narrow adversaries** represent adversaries that do not have access to the $Result$ oracle. This notion can be combined with the ones mentioned above to construct the following classes: **narrow-weak**, **narrow-forward**, **narrow-destructive**, **narrow-strong**.

D. The Three Essential Properties

The paper [1] takes into account three cryptographic properties to be considered when analyzing an RFID scheme:

1) **Correctness**:
   A scheme ensures correctness if it outputs the correct result with overwhelming probability: if the tag with the identifier ID is legitimate, the scheme outputs ID, otherwise it outputs $\perp$.
2) **Security**:
   Definition 4 from [1] states that the security of a scheme is respected if a strong adversary has a negligible probability in determining a reader to identify an uncorrupted legitimate tag with which it has not engaged in any matching conversation. This is considered only in the case of tag authentication. When mutual authentication occurs, a tag should also only have a negligible probability of authenticating a legitimate reader with which it didn't have a corresponding conversation [2].
   Simple security refers to the same notion, but by restricting it to the situation in which the adversary cannot query the $Result$ oracle. Furthermore, the reader does not use the database for the messages it transmits and also there appear two new notions: a predicate R and a sampling algorithm S, which would simulate the $Result$ oracle. In addition, it is possible that the entry of the tag in the database to be updated by a different algorithm. For simple security to be achieved it is mandatory to have

simple tag authentication and simple reader authentication. Simple security implies security.

3) **Privacy**:
The notion of privacy is presented in **Definition 6** and **Definition 7** from [1] and it is based on the notion of *a blinder for an adversary*.
A blinder $\mathcal{B}$ for an adversary $\mathcal{A}$ represents a PPT algorithm which has access to the messages to which $\mathcal{A}$ also has access and simulates the following oracles for $\mathcal{A}$: $Launch$, $SendReader$, $SendTag$, $Result$. $\mathcal{A}^{\mathcal{B}}$ denotes a blinded adversary, adversary that does not use the oracles named above, rather those oracles are simulated by the blinder.
A trivial adversary is an adversary for which exists $\mathcal{B}$ a blinder with the property that

$$\left|\Pr[\mathcal{A}\,\text{wins}] - \Pr[\mathcal{A}^{\mathcal{B}}\,\text{wins}]\right|$$

is negligible.
P-privacy: If every adversary in class P is trivial, then a RFID scheme is considered P-private.

III. Analysis of the Shariq et al. Protocol

In their 2021 publication, Shariq et al. introduced the $SR^2\,AP\text{-}DSC$ mutual authentication protocol, specifically designed for use in the healthcare sector [8]. The authors asserted that this protocol offers robust privacy and security features. We selected this protocol for our analysis because it clearly illustrates the vulnerabilities associated with global temporary variables. Given its application in the medical field, where privacy is critically important, it is crucial to scrutinize any potential weaknesses. Highlighting these vulnerabilities is essential to ensure that the protocol meets the high privacy standards required in healthcare environments.

A. The Shariq et al. Protocol

1) Security Analysis: As the protocol is designed to be used in the healthcare field, it is necessary to consider that private patient data will be stored, thus requiring the modeling of a suitable adversary. With these considerations in mind, the Dolev-Yao model has been used as the baseline model for the adversary, as presented in the work [11]. In this model, the adversary is considered a legitimate user of the system, capable of being both initiator and receiver in message exchanges with a user A, while also having access to all messages passing through the network.

Considering this, the authors resorted to the security and privacy model based on indistinguishability proposed by Ouafi and Phan in paper [9]. This model is based on the Juels-Weis model [10], with certain differences, including those related to the restrictions imposed on the adversary.

In the model, communication will take place between two parties ($\mathcal{T}$ - the tag and $\mathcal{R}$ - the reader) during a session that will end with the *Accept* result for each party if it considers that the session has been conducted correctly, between the appropriate entities. We will denote by $\mathcal{S}$ the unit consisting of the server and the reader.

Two communicating entities are considered to be partners if and only if both parties provide the *Accept* response to each other, marking the completion of the session. (**Definition 1** [9])

One of the participating parties in the protocol is considered *fresh* at the end of the session if and only if: it has provided the *Accept* output (even if it has or does not have a partner) and neither the given instance nor its partner (if any) has received a $Corrupt$ query. (**Definition 2** [9])
The adversary respects the described model and has access to the following oracles:

$Execute(\mathcal{S}, \mathcal{T}, i)$**:** Defines an eavesdropping attack, in which the adversary $\mathcal{A}$ has access to the messages exchanged in the session i between two honest parties $\mathcal{S}$ and $\mathcal{T}$, also having access to the shared secrets of the two parties.

$Send(U, V, m, i)$**:** Defines an active attack which takes place in the session i and it describes the situation when $\mathcal{A}$ impersonates entity U and sends the message m to the entity V (U and V are part of different categories: one is a reader and the other one is a tag).

$Query(\mathcal{T}, m_1, m_2)$**:** The adversary queries the tag $\mathcal{T}$ with the message m_1 and receives the message m_2.

$Block(\mathcal{A})$**:** It defines a *Denial of Service* attack with the purpose of stopping the protocol execution or desynchronizing the two parties.

$Corrupt(\mathcal{T}, K)$**:** Provides access to the secret key $K^{'}$ stored in the memory of the tag to the adversary, thereby allowing the adversary the opportunity to replace that key with another one, denoted as K. This oracle is used for modelling the property of *forward privacy*.

$Test(\mathcal{T}_0, \mathcal{T}_1, i)$**:** This oracle does not refer to any of the adversary's abilities or to any real event, but defines the notion of *UPriv* (untraceable privacy), a notion based on the property of indistinguishability. If the entity has responded with *Accept* and is offered a $Test$ query, a random bit b from $\{0, 1\}$ will be chosen and the adversary will get $\mathcal{T}_b$, with the aim of discovering if they got $\mathcal{T}_0$ or $\mathcal{T}_1$. For the UPriv property to be relevant, the *Test* session should be fresh.

Definition 3 [9] describes the notion of untraceable privacy more in depth:

UPriv refers to the game $\mathcal{G}$ between an adversary $\mathcal{A}$ and different instances of tags or readers. The game contains 3 phases:

- **Learning phase**: the adversary can interact with $\mathcal{S}$ and the two randomly chosen tags ($\mathcal{T}_0$ an $\mathcal{T}_1$) and can use $Execute$, $Send$ and $Corrupt$ queries.
- **Challenge phase**: the adversary uses $Test(\mathcal{T}_0, \mathcal{T}_1, i)$ and sends it to the challenger who will choose the value $b \in \{0, 1\}$, corresponding to one of the tags. $\mathcal{A}$ can query $Execute$ and $Send$ oracles (also $Corrupt$, but without violating the freshness property) to help him guess which tag they received.
- **Guessing phase**: the adversary will present $b^{'}$, the index of the tag they think they got.

$\mathcal{A}$'s success in winning the game (and thus violating the *UPriv* property) is determined by the advantage the adversary has in distinguishing between the two tags (compared to a random choice between the two) and thus guessing the number b. According to the above, this advantage can be expressed mathematically as: $Adv_{\mathcal{A}}^{UPriv}(K) = \left|\Pr[b' = b] - \frac{1}{2}\right|$, where k is the security parameter.

2) The Authentication Protocol: The proposed protocol is intended to be used in implementing an intelligent healthcare system, in which tags are attached to both patients and medical equipment, and the reader will transmit the information obtained from the tags to the central server (trusted authority) which will store the received data and perform the necessary operations.

In the following parts, we will consider the reader and server as a unit (reader-server unit), which we will refer to more generally as the server and it will be denoted as $\mathcal{S}$.

The protocol consists of two phases: in the first phase, the system parameters are set, and in the second phase, authentication is performed.

In the setup phase the parameters of the system are initialized. The parameters are:

1) two large prime numbers: p (1024 bits) and q (160 bits);
2) the generator g with $g \in (1, p-1)$;
3) the shared secret key a with $a \in (0, q)$;
4) the public key v ($v = g^{-a} \bmod p$);
5) a_i - random number stored in encrypted form by the server;
6) ID_i (160 bits) - the ith tag identifier, stored in the memory of the tag and in the database of the server;

The authentication phase consists of four steps:

1) **Step 1**: $\mathcal{S} \rightarrow tag_i$: $\{C_1\}$
   In the first step, the server randomly chooses a non-zero integer C_1, which it sends to the tag with which communication is established. This tag will be denoted as tag_i.
2) **Step 2**: $tag_i \rightarrow \mathcal{S}$: $\{x_1, x_2, Auth_i\}$
   This step consists of operations performed by the tag: first, it will randomly choose two non-zero integers r_1 and r_2, and then these will be used to calculate the values that will be transmitted to the server: x_1, x_2, $Auth_i$.
   The two values x_1 and x_2 are computed as follows:
   - $x_1 = g^{r_1} \bmod p$
   - $x_2 = (r_2 \cdot v^{-r_1}) \bmod p$

   Now we will calculate e and y:
   - $e = h(r_2 \,||\, x_2 \,||\, x_1)$
   - $y = (r_2 + a_i \cdot e) \bmod q$

   The last value to be computed is $Auth_i$:
   - $Auth_i = ID_i \oplus h(r_2, C_1, e, y)$

   The tag sends $\{x_1, x_2, Auth_i\}$ to the server.
3) **Step 3**: $\mathcal{S} \rightarrow tag_i$: $\{C_2\}$
   In the third step, the authentication of the tag is performed.
   The server will carry out the necessary operations to extract the tag identifier from the received information and will search the database for the obtained value.
   Thus, we will calculate the values:
   - $S_1 = {x_1}^a \bmod p$
   - $S_2 = S_1 \bmod p$
   - $r_2 = (x_2 \cdot {S_2}^{-1}) \bmod p$
   - $e^{'} = h(r_2 \,||\, x_2 \,||\, x_1)$
   - $y^{'} = (r_2 + a_i \cdot e^{'}) \bmod q$

   Using the previously computed values, ID_i is computed:
   - $ID_i = Auth_i \oplus h(r_2, C_1, e^{'}, y^{'})$

   The resulted value will be then searched in the database and if it is found, the tag will be authenticated. Now the server computes C_2 and sends it to the tag:
   - $C_2 = h(ID_i, r_2, C_1, e^{'}, y^{'}, Auth_i)$
4) **Step 3**: Authentication of the reader/server
   The tag will compute the value $C_2^{'}$ using the known values of the variables:
   - $C_2^{'} = h(ID_i, r_2, C_1, e, y, Auth_i)$

   If $C_2^{'} = C_2$, the reader-server unit is authenticated.

Thus, through these four steps and the properties of the Schnorr Cryptosystem, mutual authentication between the server and the tag is achieved. In the **Fig. 1**, the steps of the protocol are summarized.

B. Failing to Achieve Privacy in the Vaudenay Model

The authors of the SR^2AP-DSC protocol analysed its security and privacy properties using the Ouafi-Phan privacy model and proved that their protocol achieves good security and privacy. The Vaudenay model, for which a summary was provided in the second section, represents one of the most influential models used in the analysis of RFID protocols, being a stronger model than the Ouafi-Phan model in terms of the abilities of the adversary in tag corruption.

This subsection will make an analysis of the protocol according to the Vaudenay model, proving that the scheme does not assure the necessary privacy properties.

Before moving forward, it's important to take two notes on the scheme. Firstly, the protocol exhibits linear complexity when identifying tags, as each a_i is unique for every tag and session and its value is not sent to the reader/server. Consequently, the server must iterate through all values in the database. Secondly, the protocol lacks specification regarding the update mechanism for a_i and ID_i after each session. The manner in which this update occurs is essential to the protocol, as failure to address it adequately could result in desynchronization issues.

In the article [5], there are described five cases in which an RFID protocol cannot assure privacy in the Vaudenay and HPVP models. Based on this situations, five lessons are formulated.

One scenario in which a scheme may fall is the use of global temporary variables, which means variables which are assigned at a certain step and then used at another step. Regarding this situation, **Theorem 3.1** [5] asserts that a

Reader + Server	**Message**	**Tag**
$\{g,\ p,\ q,\ v,\ a\}$		$\{ID_i,\ p,\ q,\ v,\ a_i\}$
Choose $k \in \mathbb{Z}^*$, $C_1 = k$	$\xrightarrow{\{C_1\}}$	
		Generate two integers $r_1,\ r_2$ $x_1 = g^{r_1} \bmod p,\ x_2 = (r_2 \cdot v^{-r_1}) \bmod p$ $e = h(r_2 \,\|\|\, x_2 \,\|\, x_1),\ y = (r_2 + a_i \cdot e) \bmod q$
	$\xleftarrow{\{x_1,\ x_2,\ Auth_i\}}$	$Auth_i = ID_i \oplus h(r_2, C_1, e, y)$
$S_1 = {x_1}^a \bmod p,\ S_2 = S_1 \bmod p$ $r_2 = (x_2 \cdot {S_2}^{-1}) \bmod p$ $e^{'} = h(r_2 \,\|\, x_2 \,\|\, x_1),\ y^{'} = (r_2 + a_i \cdot e^{'}) \bmod q$ $ID_i = Auth_i \oplus h(r_2, C_1, e^{'}, y^{'})$ $C_2 = h(ID_i, r_2, C_1, e^{'}, y^{'}, Auth_i)$	$\xrightarrow{\{C_2\}}$	
		$C_2^{'} = h(ID_i, r_2, C_1, e, y, Auth_i)$ If $C_2^{'} = C_2$, the reader is authenticated

Fig. 1. The Shariq et al. scheme.

scheme, where the computations conducted by the tag to determine reader authentication are not based on PUFs (Physically Unclonable Functions) and the scheme uses global temporary variables, then the protocol cannot simultaneously achieve both mutual authentication and narrow-forward privacy within the Vaudenay model. This limitation arises only when the adversary has the capability to access global temporary variables, thus leading to a scenario of temporary state disclosure.

The demonstration of the theorem offers a comprehensive insight into the potential attack that can be mounted under these circumstances. We will consider $\mathcal{A}$ to be a narrow-forward adversary. The adversary will play the following privacy game:

1) $CreateTag(ID)$;
2) $(vtag, 1) \leftarrow DrawTag(distribution)$;
3) $\pi \leftarrow Launch()$;
4) $\{C_1\} \leftarrow SendReader(\emptyset, \pi)$;
5) $\{x_1,\ x_2,\ Auth_i\} \leftarrow SendTag(\{C_1\}, vtag)$;
6) $\{C_2\} \leftarrow SendReader(\{x_1, x_2, Auth_i\}, \pi)$;
7) $S \leftarrow Corrupt(vtag)$;
8) The adversary obtains all the necessary values for computing $h(ID_i, r_2, C_1, e, y, Auth_i)$ so they can decide if the reader will be authenticated by the tag or not.
9) If the adversary considers the reader to be authenticated, they return 0; otherwise, they return 1.

By corrupting the tag and considering that the temporary state is also returned when corrupting the tag, the adversary will obtain the following values: p, q, ID_i, a_i, r_2 (it is stored because it is used in two different steps), C_1. Furthermore, e, y and $Auth_i$ will be also probably present, but there is also the possibility that the tag recomputes them (if e and y are not directly stored, it means that x_1 and x_2 are or that r_1 is stored and used in the future computations). In all of these cases, the adversary has all the necessary values to compute $C_2^{'} = h(ID_i, r_2, C_1, e, y, Auth_i)$. If $C_2^{'} = C_2$, it means that the reader will be authenticated by the tag.

For the scheme to achieve narrow-forward privacy, there must exist a blinder $\mathcal{B}$ for the adversary $\mathcal{A}$ for which $\left|\Pr[\mathcal{A}\text{ wins}] - \Pr[\mathcal{A}^{\mathcal{B}}\text{ wins}]\right|$ is negligible. From the logic above, results that $\mathcal{A}$ will always output 0 in the case of a legitimate reader. On the other hand, in the blinded privacy game, because the blinder simulates $Launch$, $SendTag$ and $SendReader$, there is a high chance that this simulated reader will not be authenticated, making it obvious for the adversary which is the blinded game and which is the real one. For the adversary to not differentiate between the two games, the blinder should simulate the reader in such a manner to make it hard for the adversary to distinguish between a real reader and a fake one. This assumption would break the security of the scheme.

This means, for the scheme to still ensure narrow-forward privacy, it is necessary for the reader to not be authenticated. If reader authentication is performed, the narrow-forward privacy is lost.

As it was stated before, it can also be assumed that $Auth_i$ will also be present in the memory of the tag (as its value is also needed in the fourth step). This would offer the adversary the possibility to compare the stored $Auth_i$ with the one sent through the communication channel to differentiate between the real and the blinded privacy game. The same logic can also be applied to C_1, as it too will be saved between the steps, and it can also be used in the case when x_1 and x_2 are stored, instead of e and y.

This analysis proved that the scheme cannot achieve narrow-forward privacy and mutual authentication at the same time, the problem being the use of global temporary variables, which can be obtained by the adversary.

C. Improving the Protocol

As it was stated before, the use of global temporary variables can represent an impediment in achieving narrow-forward privacy at the same time as mutual authentication in the particularly case of temporary state disclosure in Vaudenay model, case which is plausible in the context of real-life

Reader + Server	**Message**	**Tag**
$\{g,\ p,\ q,\ v,\ a\}$		$\{ID_i,\ p,\ q,\ v,\ a_i,\ P,\ s_{C_2}\}$
Choose $k \in \mathbb{Z}^*$, $C_1 = k$	$\xrightarrow{\{C_1\}}$	
	$\xleftarrow{\{x_1,\ x_2,\ Auth_i\}}$	Generate two integers $r_1,\ r_2$ $K_{C_2} = P(s_{C_2})$ $x_1 = g^{r_1} \bmod p,\ x_2 = (r_2 \cdot v^{-r_1}) \bmod p$ $e = h(r_2 \,\|\| x_2 \,\|\| x_1),\ y = (r_2 + a_i \cdot e) \bmod q$ $Auth_i = ID_i \oplus h(r_2, C_1, e, y)$ $C_2^{'} = h(ID_i, r_2, C_1, e, y, Auth_i) \oplus K_{C_2}$ erase $K_{C_2},\ r_1,\ r_2,\ e,\ y,\ x_1,\ x_2,\ Auth_i$
$S_1 = {x_1}^a \bmod p,\ S_2 = S_1 \bmod p$ $r_2 = (x_2 \cdot {S_2}^{-1}) \bmod p$ $e^{'} = h(r_2 \,\|\| x_2 \,\|\| x_1),\ y^{'} = (r_2 + a_i \cdot e^{'}) \bmod q$ $ID_i = Auth_i \oplus h(r_2, C_1, e^{'}, y^{'})$ $C_2 = h(ID_i, r_2, C_1, e^{'}, y^{'}, Auth_i)$	$\xrightarrow{\{C_2\}}$	
		$K_{C_2} = P(s_{C_2})$ If $C_2^{'} \oplus K_{C_2} = C_2 \implies$ the reader is authenticated erase K_{C_2}

Fig. 2. A PUF-protected variant of the Shariq et al. scheme.

conditions. One way to overcome this problem is presented in [6].

The solution mentioned above is based on the use of *Physically Unclonable Functions* (PUFs) to protect the temporary variables. PUFs can be described as having two components: the input (challenge) and the output (response) generated for the given input, a CRP (challenge-response pair) being modelled. The response relies not only on the value of the challenge, but also on the physical attributes of the object. The most essential properties of PUFs are unpredictability, physical unclonability and tamper-evidence. One problem that may appear is the different response for the same challenge, meaning that the PUF is not reliable/robust. There exist known practical solutions to overcome this issue, but the supplementary overhead needs to also be taken into consideration [12]. Based on the above-mentioned characteristics and considering the multiple analyses conducted on their properties [16], [17], PUFs represent a suitable choice in the case of RFIDs.

In [6] it is proven that if a protocol achieves X-privacy and mutual authentication in the Vaudenay model without temporary state disclosure, it can be modified by adding PUF computations to also achieve the same properties in the Vaudenay model with temporary state disclosure.

For this purpose, ideal PUFs are used, being defined as a function $P : \{0,1\}^p \to \{0,1\}^k$ (p, k - polynomial sized values in the security parameter). This function needs to respect two conditions: it is computationally indistinguishable from random functions and also tampering with the object means that P is destroyed. Supplementary, it is also considered that, after being corrupt, the tag is destroyed (according to the tamper-evident nature) [6].

Based on this and on the vulnerabilities highlighted in the previous subsection, the scheme can be modified in the form that is presented in **Fig. 2**.

For proving that this variant is secure, **Definition 5.2** [6] can be referred to. This definition states the condition for a PUF-protected variant of a scheme to be secure for some class of adversaries. It is assumed that if the adversary cannot obtain the PUF-protected variable without corruption, then it can obtain it in the case of corruption with temporary state disclosure only with negligible probability.

In the case of Shariq et al. scheme, the $C_2^{'}$ variable is not sent through the channel, meaning that the adversary cannot access it. For computing this value, the adversary would need the value of r_2, which they cannot obtain without accessing the temporary state (the other variables can be found in the permanent state or can be computed knowing r_2). $C_2^{'}$ is protected using $C_2^{'} \oplus P(s_{C_2})$ and the other temporary variables are erased, meaning that by corrupting the tag, $C_2^{'} \oplus P(s_{C_2})$ is the only value that the adversary can obtain, value which cannot be used in gaining any advantage as the tag is also destroyed after corruption. Based on the assumed security of the PUF, it can be concluded that the scheme is secure against a narrow-forward adversary (for which the attack was constructed). **Theorem 6.1** [6] summarizes the relationship between the original scheme and the PUF-protected variant of the scheme, stating that if a scheme achieves mutual authentication and X-privacy in the Vaudenay model without temporary state disclosure, then any PUF-protected variant of the scheme will also achieve mutual authentication and X-privacy in the Vaudenay model with temporary state disclosure.

From **Theorem 6.1** [6] and using the fact that the PUFs are considered secure, the constructed PUF-protected variant of the Shariq et al. scheme achieves narrow-forward privacy in the Vaudenay model with temporary state disclosure. This happens because, by corrupting the tag, besides the variables

Reader + Server	**Message**	**Tag**
$\{g,\ p,\ q,\ v,\ a\}$		$\{ID_i,\ p,\ q,\ v,\ a_i,\ P,\ s_{C_2},\ s_{a_i}\}$
Choose $k \in \mathbb{Z}^*$, $C_1 = k$	$\xrightarrow{\{C_1\}}$	
		Generate two integers $r_1,\ r_2$ $K_{C_2} = P(s_{C_2})$ $K_{a_i} = P(s_{a_i})$ $x_1 = g^{r_1} \bmod p,\ x_2 = (r_2 \cdot v^{-r_1}) \bmod p$ $e = h(r_2 \,\|\, x_2 \,\|\, x_1),\ y = (r_2 + (a_i \oplus K_{a_i}) \cdot e) \bmod q$
	$\xleftarrow{\{x_1,\ x_2,\ Auth_i\}}$	$Auth_i = ID_i \oplus h(r_2, C_1, e, y)$ $C_2^{'} = h(ID_i, r_2, C_1, e, y, Auth_i) \oplus K_{C_2}$ erase $K_{C_2},\ r_1,\ r_2,\ e,\ y,\ x_1,\ x_2,\ Auth_i,\ K_{a_i}$
$S_1 = {x_1}^a \bmod p,\ S_2 = S_1 \bmod p$ $r_2 = (x_2 \cdot {S_2}^{-1}) \bmod p$ $e^{'} = h(r_2 \,\|\, x_2 \,\|\, x_1),\ y^{'} = (r_2 + a_i \cdot e^{'}) \bmod q$ $ID_i = Auth_i \oplus h(r_2, C_1, e^{'}, y^{'})$ $C_2 = h(ID_i, r_2, C_1, e^{'}, y^{'}, Auth_i)$	$\xrightarrow{\{C_2\}}$	
		$K_{C_2} = P(s_{C_2})$ If $C_2^{'} \oplus K_{C_2} = C_2 \implies$ the reader is authenticated erase K_{C_2}

Fig. 3. An Improved PUF-protected variant of the Shariq et al. scheme.

present in the permanent memory, the adversary only obtains $C_2^{'} \oplus P(s_{C_2})$, from which they cannot deduce any useful information.

The scheme still does not achieve narrow-destructive privacy (without temporary state disclosure), as the following attack can be mounted:

1) $CreateTag(ID)$;
2) $(vtag, 1) \leftarrow DrawTag(distribution)$;
3) $\pi \leftarrow Launch()$;
4) $\{C_1\} \leftarrow SendReader(\emptyset, \pi)$;
5) $\{x_1,\ x_2,\ Auth_i\} \leftarrow SendTag(\{C_1\}, vtag)$;
6) $(ID_i,\ p,\ q,\ v,\ a_i) \leftarrow Corrupt(vtag)$;
7) $\{C_2\} \leftarrow SendReader(\{x_1^{'}, x_2^{'}, Auth_i^{'}\})$;
8) If the adversary considers the reader to be authenticated, they return 0; otherwise, they return 1.

The adversary simulates the first two steps of the protocol and then corrupts the tag to obtain its permanent state.

Then, the adversary generates two random numbers r_1 and r_2 and computes $x_1^{'}$, $x_2^{'}$ and $Auth_i^{'}$ to be sent to the reader. As the adversary has access to ID_i, p, q, v, a_i, the computed $x_1^{'}$, $x_2^{'}$ and $Auth_i^{'}$ are valid. After that, the adversary can compute $C_2^{'}$ using the values obtained by corrupting the tag and the generated ones. As the response from the adversary is valid, if the adversary plays the real game, $C_2^{'} = C_2$ with overwhelming probability (if the reader would be authenticated).

In this way, if the reader is authenticated, the adversary can distinguish between the real privacy game and the blinded one (in which C_2 would be wrong with overwhelming probability). Considering this, when playing the real privacy game, the adversary will output 0 with overwhelming probability (when the reader is authenticated), while the result will be 1 with overwhelming probability, in the case of the blinded privacy game.

As a consequence, the scheme cannot achieve mutual authentication and narrow-destructive privacy, remaining limited to narrow-forward privacy.

For the scheme to also achieve narrow-destructive privacy, supplementary protection regarding the values from the permanent memory (ID_i, p, q, v and a_i) should be implemented.

In **Fig. 3**, a_i is protected using PUFs, meaning that, when the tag is corrupted, $a_i \oplus P(s_{a_i})$ is obtained (this is the stored value). As the PUFs are considered secure, the adversary cannot guess the real value of a_i, meaning that they cannot compute y and, subsequently, $Auth_i$. This type of protection could have been applied to ID_i too, depending on the specific requirements.

To ensure forward privacy, the scheme must be proven to be secure (to use the result included in **Lemma 8** from [1]). As forward privacy can be achieved using PKC, the security of the scheme depends on the security of the utilized Schnorr signature. Regarding this aspect, Schnorr signatures have been proven secure against an adaptive chosen-message attack in the Random Oracle Model, based on the complexity of the Discrete Logarithm Problem [14], [15]. Despite the numerous studies on the security of the Schnorr signature scheme, its security in the standard model has not been proven.

IV. Conclusions

As stated earlier in the paper, the disclosure of temporary state is a critical consideration for the practical implementation of RFID systems.

In the Vaudenay model, it is not explicitly stated whether the adversary also gains access to the temporary variables when corrupting the tag, but to consider this scenario as possible,

is an essential requirement given the continuous advance of technology and the increasing capabilities of adversaries.

The analysis above was conducted to highlight vulnerabilities that may arise when using global temporary variables. As exemplified in the case of the Shariq et al. protocol, any scheme which uses global temporary variables fails to achieve narrow-forward privacy and mutual authentication in the Vaudenay model with temporary state disclosure, unless these variables are protected. Based on the results from [6], PUFs represent a suitable choice for protecting temporary variables.

Furthermore, an essential element to consider is the adversarial model used in the protocol analysis. This paper also seeks to highlight the differences that may arise in the analysis when employing different security and privacy models. The Ouafi-Phan model, despite considering the adversary's corruption capability, restricts the obtained values to the secret key, thus not providing a sufficiently strong model.

These considerations aim to emphasize the importance of adopting a suitable adversarial model in protocol analysis, one that accurately reflects the practical requirements of real-life RFID systems. This was illustrated through an examination of a scheme designed for use in the healthcare field, where privacy is a critical attribute.

ACKNOWLEDGMENT

Our sincere thanks to Prof. Dr. Ţiplea for his valuable suggestions, constant support, and insightful contributions to our research.

REFERENCES

[1] S. Vaudenay, "On Privacy Models for RFID," in *Advances in Cryptology – ASIACRYPT 2007*, K. Kurosawa (Ed.), Berlin, Heidelberg: Springer, 2007, pp. 68–87, http://dx.doi.org//10.1007/978-3-540-76900-2_5.

[2] R.-I. Paise and S. Vaudenay, "Mutual Authentication in RFID: Security and Privacy," *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, ASIACCS '08*, March 2008, http://dx.doi.org/10.1145/1368310.1368352.

[3] J. Hermans, A. Pashalidis, F. Vercauteren, and B. Preneel, "A New RFID Privacy Model," in *Computer Security – ESORICS 2011*, V. Atluri and C. Diaz (Eds.), Berlin, Heidelberg: Springer, 2011, pp. 568–587, https://doi.org/10.1007/978-3-642-23822-2_31.

[4] J. Hermans, R. Peeters, and B. Preneel, "Proper RFID privacy: Model and protocols," *IEEE Transactions on Mobile Computing*, vol. 13, no. 12, pp. 2888–2902, December 2014, http://dx.doi.org/10.1109/TMC.2014.2314127.

[5] F. L. Tiplea, "Lessons to be Learned for a Good Design of Private RFID Schemes," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 4, pp. 2384–2395, 2022, http://dx.doi.org/10.1109/TDSC.2021.3055808. http://dx.doi.org/10.1109/TMC.2014.2314127.

[6] F. L. Tiplea and C. Hristea, "PUF Protected Variables: A Solution to RFID Security and Privacy Under Corruption With Temporary State Disclosure," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 999–1013, 2021, http://dx.doi.org/10.1109/TIFS.2020.3027147.

[7] S. Weis, S. Sarma, R. Rivest, and D. Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems," *Lect. Note. Comput. Sci.*, vol. 2802, April 2003, http://dx.doi.org/10.1007/978-3-540-39881-3_18.

[8] M. Shariq, K. Singh, M. Y. Bajuri, A. Pantelous, A. Ahmadian, and M. Salimi, "A Secure and Reliable RFID Authentication Protocol using Schnorr Digital Cryptosystem for IoT-enabled Healthcare in COVID-19 Scenario," *Sustainable Cities and Society*, vol. 75, pp. 103354, September 2021, http://dx.doi.org/10.1016/j.scs.2021.103354.

[9] K. Ouafi and R. C.-W. Phan, "Privacy of Recent RFID Authentication Protocols," in *Information Security Practice and Experience. ISPEC 2008*, Springer, Berlin, 2008, pp. 263-277, https://doi.org/10.1007/978-3-540-79104-1_19.

[10] A. Juels and S. A. Weis, "Defining Strong Privacy for RFID," *Cryptology ePrint Archive, Paper 2006/137*, 2006. Available: https://eprint.iacr.org/2006/137

[11] D. Dolev and A. Yao, "On the Security of Public Key Protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198-208, 1983, https://doi.org/10.1109/TIT.1983.1056650.

[12] Y. Gao, S. Al-Sarawi, and D. Abbott, "Physical unclonable functions," *Nature Electronics*, vol. 3, no. 2, pp. 81-91, February 2020, http://dx.doi.org/10.1038/s41928-020-0372-5.

[13] F. Costa, S. Genovesi, M. Borgese, A. Michel, F. A. Dicandia, and G. Manara, "A Review of RFID Sensors, the New Frontier of Internet of Things," *Sensors*, vol. 21, no. 9, article no. 3138, 2021, http://dx.doi.org/10.3390/s21093138. Available: https://www.mdpi.com/1424-8220/21/9/3138.

[14] D. Pointcheval and J. Stern, "Security Proofs for Signature Schemes," in *EUROCRYPT 1996*, vol. 1070, pp. 387-398, 1996, http://dx.doi.org/10.1007/3-540-68339-9_33.

[15] D. Pointcheval and J. Stern, "Security Arguments for Digital Signatures and Blind Signatures," *Journal of Cryptology*, vol. 13, pp. 361-396, 2000, https://doi.org/10.1007/s001450010003.

[16] B. Halak, "Physically Unclonable Functions: Design Principles and Evaluation Metrics," in *Physically Unclonable Functions*, Springer, Cham, 2018, https://doi.org/10.1007/978-3-319-76804-5_2.

[17] D. Yamamoto, M. Takenaka, K. Sakiyama, and N. Torii, "Security Evaluation of Bistable Ring PUFs on FPGAs Using Differential and Linear Analysis," *2014 Federated Conference on Computer Science and Information Systems*, Warsaw, Poland, 2014, pp. 911-918, http://dx.doi.org/10.15439/2014F122.