Proceedings of the 19th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 177–185
DOI: 10.15439/2024F6703
ISSN 2300-5963 ACSIS, Vol. 39

On Privacy of PRF+PUF-based Authentication

Ferucio Laurenţiu Ţiplea
0000-0001-6143-3641
Faculty of Computer Science
"Alexandru Ioan Cuza" University of Iaşi
Iaşi, Romania
Email: ferucio.tiplea@uaic.ro

***Abstract*—RFID-based authentication plays a crucial role in various fields, such as e-commerce, e-learning, e-business, healthcare, cloud, IoT, etc. At the same time, there is growing interest in using physically unclonable functions (PUFs) in RFID tags to protect against key corruption of pseudo-random functions (PRFs). In this paper, we discuss the privacy properties of PRF+PUF-based RFID authentication protocols in Vaudenay's and the Hermans-Pashalidis-Vercauteren-Preneel (HPVP) models, considering two fundamental aspects: using temporary variables that might compromise privacy and using simulatable PUFs, a more realistic approach to ideal PUFs. Finally, we prove that a variant of a recently proposed RFID-based authentication protocol achieves strong privacy in the HPVP model.**

I. Introduction

AUTHENTICATION is a process by which the validity of a particular assertion regarding an entity (entity authentication) or message (message or data origin authentication) is verified. Authentication plays a crucial role in information security because the authorization of access to information and data or the permission to carry out certain activities depends on it. Over time, many authentication techniques have been proposed, such as password-based authentication, certificate-based authentication, biometric authentication, token-based authentication, voice authentication, multi-factor authentication, and so on.

In this paper, we will look at authentication as a process by which one party, usually called the *verifier*, verifies the identity of another party, usually called the *prover*, by means of a protocol that takes place between the two parties. In some situations, the authentication process can include other parties, such as a server or a trusted authority. But, as we said, our paper focuses only on the authentication protocol developed between the prover and the verifier.

The basic requirements to be satisfied by an authentication process are security and privacy. In general, *security* means that no adversary can impersonate the prover or the verifier except with negligible probability. *Privacy* properties are much more nuanced and diverse, referring to anonymity, untraceability, unlinkability, etc.

Radio frequency identification (RFID) is a wireless communication technology between two parties, usually called *tag* and *reader*, through which the reader (playing the role of verifier) tries to uniquely identify the tag (which plays the role of prover). When the identification process is also completed by authentication, we speak of *RFID* (*RFID-based*, *RFID-enabled*) *authentication*. RFID-based authentication is crucial in various fields such as e-commerce, healthcare, IoT, cloud, etc. [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18].

Over time, much effort has been dedicated to developing security and privacy models for RFID authentication. Two of the most important models are Vaudenay's [19] and the Hermans-Pashalidis-Vercauteren-Preneel (HPVP) [20], [21] models. These models treat security identically. However, the privacy properties in Vaudenay's model are stronger than those in the HPVP model. What is very important, however, is that these models propose a hierarchy of privacy properties and generic schemes for each level of privacy. The instantiation of such a scheme means nothing more than replacing the ideal cryptographic primitive, such as a pseudo-random function (PRF), pseudo-random generator (PRG), physically unclonable function (PUF), etc. with a practical construction. Suppose the said practical construction proves to be insecure at some time. In that case, it can be replaced with another practical construction, keeping the general scheme and the security and privacy results it enjoys.

Contributions and Paper Structure. One of the generic RFID schemes proposed in [19] is based on using a pseudo-random function. This scheme, generically called the *PRF-based RFID scheme*, ensures unilateral authentication and weak privacy both in Vaudenay's and the HPVP models. Since using a PRF requires a key to be stored on the tag (on a prover's device), the scheme cannot ensure privacy against adversaries who can corrupt the tag (since they will obtain the PRF key through corruption). To overcome this limitation, [22] endowed tags with PUFs for key storage. PUFs are typically assumed to be *physically unclonable* (it is infeasible to produce two PUFs that cannot be distinguished based on their challenge/response behavior), *unpredictable* (it is infeasible to predict the response to an unknown challenge), and *tamper-evident* (any attempt to physically access the PUF irreversibly changes the challenge/response behavior). In addition, PUFs are considered a less expensive alternative to non-volatile memory (NVM). The new construction based on PRF and PUF ensures unilateral authentication and destructive privacy in Vaudenay's model, a higher level than weak privacy. However, there are two main issues with this scheme:

- Extending the scheme to ensure mutual authentication

raises the issue of how to use temporary variables;
- From a practical point of view, PUFs are nondeterministic. That means PUFs must be accompanied by an auxiliary mechanism to be used by the prover and verifier in the authentication protocol.

This paper addresses these two problems. We will discuss using temporary variables in the PRF+PUF-based RFID scheme in the Vaudenay and HPVP models. Then, we will simplify a recent authentication protocol based on PRFs and simulatable PUFs [23] and discuss the level of privacy achieved by it in these two models.

II. Basics of RFID Security and Privacy Models

This section will present some basic elements of Vaudenay's and the HPVP security and privacy models. We will be brief so the reader is referred to [19], [20], [21], [24], [25] for details. We use in our exposition *probabilistic polynomial time* (PPT) algorithms $\mathcal{A}$ as defined in [25] that can consult *oracles*. For a set A, $a \leftarrow A$ means that a is uniformly at random chosen from A. If $\mathcal{A}$ is a probabilistic algorithm, then $a \leftarrow \mathcal{A}$ means that a is an output of $\mathcal{A}$ for some input.

The *authentication process* between a prover and a verifier requires the execution of a protocol between the two parties. When the communication between the prover and the verifier is carried out through radio waves, we say that we are dealing with an *RFID-based authentication*. In this context, the prover is seen as a resource-constrained small device, usually called a *tag*. However, the verifier called the *reader* is a computationally unrestricted powerful device that can perform any cryptographic operation. When the reader is near the tag, it transmits energy through radio waves, thus making the execution of an identification and authentication protocol possible. There are also scenarios where the tags have their own energy source.

Considerable effort has been put into the development of security and privacy models for RFID systems. Vaudenay's model [19] and the HPVP model [20], [21] are two of them, with major impact in the study of security and privacy properties of RFID systems. The HPVP model borrows the adversary model from Vaudenay's model, keeps the same approach to the security property, but treats privacy in a different way. If in Vaudenay's model privacy is based on indistinguishability between the RFID system instrumented by a challenger and the RFID system instrumented by a blinder (who does not know the secret elements in the system), the HPVP model treats privacy through indistinguishability between tags in the RFID system instrumented by a challenger. This second approach is closer to the security approach in the usual encryption systems. We will use these models throughout this work, so we recall their basic elements. First of all it is necessary to mention that the memory of a tag is typically split into *permanent* (or *internal*), used to store the state values of the tag, and *temporary* (or *volatile*), used to carry out the calculations required by the communication protocol. There are two types of temporary variables, *local*, used by tags only to do computations in a given protocol step, and *global*, that get values in a given protocol step and are used in another protocol step.

Now, we can present Vaudenay's and the HPVP adversarial model. The oracles an adversary (PPT algorithm) can query in these models are those in the table in Figure 1.

The oracle $Corrupt$ in Vaudenay's model returns only the current permanent state. We sometimes say that Vaudenay's model is *without temporary state disclosure* (TSD) [26], [27], [28], [29], [30], [31]. In *Vaudenay's model with TSD*, $Corrupt$ returns the entire state of the tag, as in the HPVP model.

The adversaries can now be classified as follows:

1) (In both models) Adversaries with no access to $CreateInsider$. These are further classified according to the way the $Corrupt$ oracle is used: *weak adversaries* (no access to $Corrupt$), *forward adversaries* (once they access the $Corrupt$ oracle, the only oracle they can access is $Corrupt$), *destructive adversaries* (the tag is destroyed after corruption), *strong adversaries* (no restrictions), *narrow adversaries* (no access to $Result$);
2) (Only in the HPVP model) Adversaries with access to $CreateInsider$. The power of a destructive or strong adversary does not increase if he is given access to the $CreateInsider$ oracle.

Security in Vaudenay's and the HPVP modesl means that no strong adversary has more than a negligible probability to make the reader authenticate an uncorrupted legitimate tag without having any tag authentication matching conversation. When the RFID scheme is with mutual authentication, besides the above requirement, it is asked that no strong adversary has more than a negligible probability to make an uncorrupted legitimate tag to authenticate the reader without having any reader authentication matching conversation.

Privacy generalizes well-known properties such as anonymity, unlinkability, untracebility, etc. It is treated differently in the two models. Vaudenay's model considers the *blinder* concept for a class $\mathcal{C}$ of adversaries, which is a PPT algorithm $\mathcal{B}$ that simulates the $Launch$, $SendReader$, $SendTag$, and $Result$ oracles for adversaries in $\mathcal{C}$, without having access to the corresponding secrets. However, $\mathcal{B}$ look passively at the communication between adversaries in $\mathcal{C}$ and the other oracles allowed to it by the class $\mathcal{C}$ (that is, $\mathcal{B}$ gets exactly the same information as any adversary in $\mathcal{C}$ when querying these oracles). The scheme is *$\mathcal{C}$-private* in Vaudenay's model if no adversary in $\mathcal{C}$ has more than a negligible advantage over $1/2$ to distinguish between protocol sessions in the real scheme from those in the scheme instrumented by some blinder.

An RFID scheme is considered *$\mathcal{C}$-private* in the HPVP model if no adversary in $\mathcal{C}$ can distinguish with more than a negligible probability over $1/2$ with which tag he communicated (the left or the right in the pairs drawn by him).

The previously defined adversary classes lead to a ranking of the privacy properties of RFID schemes as shown in the diagram in Figure 2.

Oracles in Vaudenay's model	Oracles in the HPVP model
$CreateReader()$: Unsupported	$CreateReader()$: Creates a new reader, and a unique reference R to it is returned
$CreateTag^b(ID)$: Creates a tag with the identifier ID. When $b = 1$, the tag is considered *legitimate* and registered in the server's database; otherwise ($b = 0$) it is considered *illegitimate*. A unique reference T to the tag is returned; the tag is considered *free*	$CreateTag(ID)$: Creates a tag with the identifier ID and registers it in the server's database (that is, the oracle creates only legitimate tags). Duplicate tags with the same ID are accepted. A unique reference T to the tag is returned; the tag is considered *free*
$RegisterTag(T, R)$: Unsupported	$RegisterTag(T, R)$: Registers the tag T with the reader R
$Launch()$: Generates and outputs a new protocol session identifier π	$Launch(R)$: Generates and outputs a new protocol session identifier π with the reader R
$DrawTag(\delta)$: The oracle chooses a number of free tags according to the distribution δ, let us say n, generates n temporary identities $vtag_1, \ldots, vtag_n$, and outputs $(vtag_1, b_1, \ldots, vtag_n, b_n)$, where b_i specifies whether the tag $vtag_i$ is legitimate or not. All these tags are considered now *drawn*. The oracle maintains a list Γ of drawn tags	$DrawTag(T_0, T_1)$: Generates a fresh virtual tag reference $vtag$ that refers to either T_0 or T_1, depending on the privacy game where the oracle is queried. The triple $(vtag, T_0, T_1)$ is included in a list Γ of drawn tags, and $vtag$ is returned by the oracle. The oracle returns $\perp$ if one of the two tags is in the insider list, or one of the two tags is registered with a different set of readers than the other tag, or T_0 (T_1) is already referenced drawn
$Free(vtag)$: Resets (erases) the temporary state of the tag referenced by $vtag$ and removes it from Γ	$Free(vtag)$: Resets (erases) the temporary state of the tag referenced by $vtag$ and removes the corresponding triple from Γ
$SendTag(m, vtag)$: Outputs the tag's answer when the message m is sent to the tag referred to by $vtag$. When m is the empty message, this oracle outputs the first message of the protocol instance π, assuming that the tag does the first step in the protocol	$SendTag(m, vtag)$: Outputs the tag's answer when the message m is sent to the tag referred to by $vtag$. When m is the empty message, this oracle outputs the first message of the protocol instance π, assuming that the tag does the first step in the protocol
$SendReader(m, \pi)$: Outputs the reader's answer when the message m is sent to it as part of the protocol instance π. When m is the empty message, abusively but suggestively denoted by $\emptyset$, this oracle outputs the first message of the protocol instance π, assuming that the reader does the first step in the protocol;	$SendReader(R, m, \pi)$: Outputs the R's reader answer when the message m is sent to it as part of the protocol instance π. When m is the empty message, abusively but suggestively denoted by $\emptyset$, this oracle outputs the first message of the protocol instance π, assuming that the reader does the first step in the protocol;
Outputs $\perp$ if in session π the reader has not yet made a decision on tag authentication (this also includes the case when the session π does not exist), 1 if in session π the reader authenticated the tag, and 0 otherwise (this oracle is both for unilateral and mutual authentication)	$Result(\pi)$: Outputs $\perp$ if in session π the reader has not yet made a decision on tag authentication (this also includes the case when the session π does not exist), 1 if in session π the reader authenticated the tag, and 0 otherwise (this oracle is both for unilateral and mutual authentication)
$Corrupt(vtag)$: Outputs the current permanent (internal) state of the tag referred to by $vtag$, when the tag is not involved in any computation of any protocol step (that is, the permanent state before or after a protocol step)	$Corrupt(T)$: Outputs the current permanent and temporary state of the tag T. Remark that the corruption is with respect to a tag, not a virtual tag.
$CreateInsider$: Unsupported	$CreateInsider(ID)$: Creates a tag and returns a unique reference T to it and its full state. The tag is included in a list of insider tags.

Fig. 1. Vaudenay's and the HPVP adversarial models

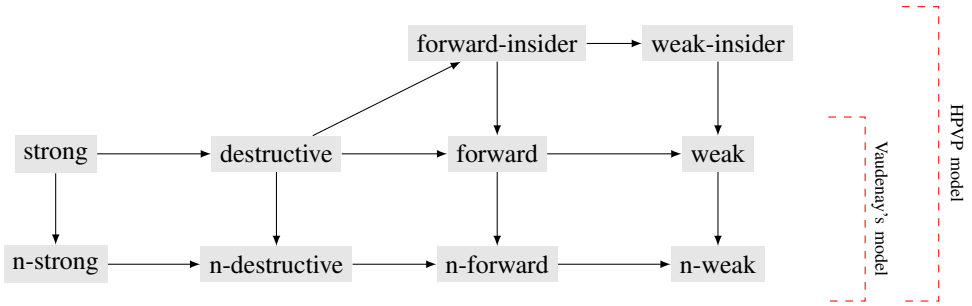

Fig. 2. Privacy levels in Vaudenay's and the HPVP model: "n-p" means "narrow p" and an arrow means "implication".

III. PUF-BASED RFID

Physically unclonable functions. One of the primary reasons that led to the development of *physically unclonable functions* (PUFs) was to find a method of protecting the secret keys against software and physical attacks [1]. A PUF can be considered a disordered physical system that can be challenged with external stimuli (challenges) to which it will react with corresponding responses. Unlike standard digital systems, the reaction of a PUF to a challenge depends on the micro- or nanoscale structural disorder of the PUF. Ideally, it is assumed that:

1) *Unclonability:* this disorder cannot be cloned or reproduced precisely, even by PUF's original manufacturer;
2) *Randomness:* the *response* r of the PUF to a *challenge* c is uniquely and uniformly at random chosen from the space of possible responses;
3) *Tamper-evident:* PUFs are tamper-evident (fully invasive attacks either damage or alter the functional behavior).

As a result, an *ideal PUF* defines a unique function P.

More realistically, a PUF is *noisy*. That is, it behaves like a non-deterministic function whose response depends on process variations, noise, environmental variables, and aging. However, it is assumed that all relevant environmental parameters are bounded, and the evaluation time of any given PUF has an upper bound. Therefore, two random evaluations of the PUF response given the same challenge might slightly vary with a Hamming distance between them bounded from above by a constant threshold. In such a case, one critical attribute of a PUF is the *reliability* of its responses, which estimates how consistently the responses can be generated against varying operating conditions.

Simulatable PUF. A *simulatable PUF* [1], [23] is a pair consisting of a noisy PUF and a parameterized model $SimPUF$ capable of computing a response r and its corresponding reliability confidence $conf$ in polynomial time for any given challenge c (i.e., $(r, conf) \leftarrow SimPUF(c)$), such that:

1) $SimPUF$ is constructed using one-time privileged access by an authorized party in a secure environment and subsequent acquisition of $SimPUF$ by any party is disabled;
2) if $r' \leftarrow PUF(c)$, then $P(r = r')$ is ϵ-close to 1 and $conf$ is ϵ-close to the reliability confidence of r'.

PUF tags. The (ideal) properties of PUFs mentioned above, as well as the technological progress aimed at achieving these properties, led to the proposal of security protocols that include them in various forms. Thus, we can mention protocols for oblivious transfer, bit commitment, key exchange, key generation, or authentication [1], [32], [13]. For example, [32] reviews key generators and authentication protocols based on PUFs proposed up to 2016. Among newer protocols, we mention [27], [23].

As far as we know, the first use of PUFs in RFID systems appears in [33], [34] to provide a solution to the problem of finding a private destructive RFID system in Vaudenay's model. Later, the use of PUFs in RFID systems gained momentum (see [28] for an ample discussion on this topic). The method of use is as follows. Tags are endowed with PUFs and store secret information (usually a secret key). When the tag authenticates itself to the reader, it interrogates the PUF and extracts that secret information, which will then be used in preparing the message for the reader.

Tags with PUFs embedded into them are usually called *PUF tags*. A *PUF-based RFID scheme* is an RFID scheme with PUF tags. The previously discussed adversary model is trivially extended to the case of PUF tags. We only need to discuss the $Corrupt$ and $CreateInsider$ oracles:

1) Due to PUF's tamper-evident property, no adversary with the possibility of corrupting PUF tags can obtain the secret information stored in the PUF. So, the $Corrupt(T)$ oracle returns only the full state of the tag;
2) Due to the non-clonability of PUFs, $CreateInsider(ID)$ creates a tag with the identity ID and lets the adversary simulate its PUF through a list of randomly generated pairs. This makes this oracle have a behavior similar to that of the original approach.

As a result of these, the classification and ranking of the privacy properties in Figure 2 remains the same for the case of PUF-based RFID schemes.

IV. PRF+PUF-BASED AUTHENTICATION

Obtaining an RFID mutual authentication scheme that would provide destructive privacy in Vaudenay's model was an open problem until 2010, when Sadeghi et al. [33], [34] managed to offer a solution. They started with the PRF-based RFID scheme and added PUF tags to generate PRF's

keys. Thus, corrupting the tags prevents the adversary from obtaining the PRF key. That is how the PRF+PUF paradigm was born, and it proved very useful in many constructions of authentication schemes later proposed.

In [35], a PRF+PUF-based RFID scheme was proposed, that achieves mutual authentication and destructive privacy in Vaudenay's model (without TSD). The scheme is given in Figure 3. Here, ℓ_1 and ℓ_2 are of polynomial size in the security parameter λ, and $F = (F_K)_{K \in \mathcal{K}}$ is a PRF, where F_K is from $\{0,1\}^{2\ell_1+1}$ to $\{0,1\}^{\ell_2}$, for all $K \in \mathcal{K}_\lambda = \{K \in \mathcal{K} \mid |K| = \lambda\}$. Each tag is equipped with a (unique) PUF $P : \{0,1\}^p \to \mathcal{K}_\lambda$ and has the capacity to compute F, where p is of polynomial size in λ. The internal state of the tag consists of a string $s \in \{0,1\}^p$ randomly chosen as a seed to evaluate P. The reader maintains a database DB with entries for all legitimate tags. Each entry is a vector (ID, K), where ID is the tag's identity and $K = P(s)$, where P is the tag's PUF.

The mutual authentication protocol is as follows. The reader sends initially a random x to the tag. On receiving it, the tag generates a random y, computes $K = P(s)$ and $z = F_K(0, x, y)$, erases K, and answers with (y, z). The reader checks its database for a pair (ID, K) such that $z = F_K(0, x, y)$. If such a pair is found, it outputs ID; otherwise, outputs $\perp$ and randomly chooses $K \in \mathcal{K}_\lambda$. No matter of the two cases (K is found in the database or is randomly generated), the reader computes $w = F_K(1, x, y)$ and sends it to the tag. On receiving it, the tag computes $P(s)$ and $w' = F_{P(s)}(1, x, y)$. Finally, it outputs OK or $\perp$ depending on the equality $w = w'$.

We notice that the tag erases the key K after using it in step 2. That prevents the key from being obtained through the tag's TSD-corruption. As a result, the tag must recompute the key in step 4. However, this precaution does not guarantee that the protocol achieves destructive privacy in Vaudenay's model with TSD. Let us consider the following narrow adversary $\mathcal{A}$ against the scheme:

1) $CreateTag^1(ID)$;
2) $(vtag, 1) \leftarrow DrawTag(P(ID) = 1)$;
3) $\pi \leftarrow Launch()$;
4) $x \leftarrow SendReader(\emptyset, \pi)$;
5) $y, z \leftarrow SendTag(x, vtag)$;
6) $(s, x', y') \leftarrow CorruptTag(vtag)$;
7) $If\ y = y'$ then output 0 (the real privacy game) else output 1 (the blinded privacy game).

As one can see, $\mathcal{A}$ creates a legitimate tag, draws it, runs a protocol session with the tag for the first two steps, and then corrupts the tag. If the temporary variable y' is not changed ($y' = y$), then the adversary plays the real privacy game with overwhelming probability. This is because the blinder does not know the tag's internal state and, therefore, it cannot return the value of y, except with negligible probability. A similar attack can be mounted for the case of x.

As a conclusion, the scheme is not even narrow forward private in Vaudenay's model with TSD.

In the HPVP model, the adversary cannot corrupt virtual tags; it can only corrupt physical tags. In addition, when a tag is released, its state is reset, which means that the adversary cannot obtain the values of the temporary variables after a protocol execution. As a result, we can simplify the protocol in Figure 3 without exposing it to corruption, as shown in Figure 4. Moreover, this new protocol ensures mutual authentication and strong privacy in the HPVP model (we will show this in a more general framework in the next section).

V. PRF+SimPUF-based Authentication

The use of a non-deterministic PUF P on a tag raises the problem of selecting the answer to a challenge c. However, having stored on the reader a SimPUF P' associated with P, there are procedures that can decide in polynomial time the answer of P on c. Such a procedure is TREVERSE proposed in [23]. This uses P' for possible responses of P to c, and the correct selection is made based on a pseudo-random function F. The authentication protocol is the one in Figure 5. The server initiates the protocol by sending a challenge c to the tag. The tag queries its (non-deterministic) PUF, obtains $r \leftarrow P(c)$, and responds with $(x_1, y = F_r(x_1))$, where x_1 is a random value and F is a pseudo-random function shared by the server and tag. When the server receives the tag response, it uses P', the model of the PUF P, and the TRESERVE function to determine r^t, the possible response of P. The check is done by "$y = F_{r^t}(x_1)$". If such a value is found, the tag is authenticated and announced. Otherwise, the protocol is aborted. In the case of tag authentication, it sends the server a random value x_2 and receives $z = F_{r^t}(x_2)$. The value z is checked against $F_r(x_2)$. If the values match, the tag authenticates the server.

In [23], the authors presented a security analysis of the protocol in Figure 5. The adversarial model used is the following:

- The adversary can eavesdrop on the communication channel;
- The adversary can arbitrarily apply challenges via the publicly accessible interface to observe the tag's response.

A supplementary assumption states that $SimPUF$ enrollment is performed by the server in a secure environment using one-time privileged access, and such access is prohibited afterward.

Under these, the security analysis in [23] focused on brute force, replay, modeling, and physical attacks. However, from the protocol's privacy point of view, [23] did not conduct any study. We note that the protocol uses r as a global temporary variable. Then, similarly to the previous section, an adversary can mount the following attack: corrupt the tag, get r, and verify the equality "$z = F_r(x_2)$". If it holds, the adversary plays the real privacy game; otherwise, the adversary plays the blinded privacy game. Therefore, the protocol cannot simultaneously ensure mutual authentication and narrow forward privacy in Vaudenay's model.

Concerning privacy in the HPVP model, we will show below that the protocol achieves strong privacy. First, we

	Reader (DB)		**Tag** (P, s)
1	$x \leftarrow \{0,1\}^{\ell_1}$	$\xrightarrow{x}$	
2		$\xleftarrow{y,\ z}$	$y \leftarrow \{0,1\}^{\ell_1}$, $K = P(s)$ $z = F_K(0, x, y)$, erase K
3	If $\exists (ID, K) \in DB$ s.t. $z = F_K(0, x, y)$ then output ID (tag auth.) else output $\perp$, $K \leftarrow \mathcal{K}_\lambda$; $w = F_K(1, x, y)$	$\xrightarrow{w}$	
4			$K = P(s)$, $w' = F_K(1, x, y)$, erase K If $w = w'$ then output OK else output $\perp$

Fig. 3. PRF+PUF-based RFID scheme

	Reader (DB)		**Tag** (P, s)
1	$x \leftarrow \{0,1\}^{\ell_1}$	$\xrightarrow{x}$	
2		$\xleftarrow{y,\ z}$	$y \leftarrow \{0,1\}^{\ell_1}$, $K = P(s)$ $z = F_K(0, x, y)$
3	If $\exists (ID, K) \in DB$ s.t. $z = F_K(0, x, y)$ then output ID (tag auth.) else output $\perp$, $K \leftarrow \mathcal{K}_\lambda$; $w = F_K(1, x, y)$	$\xrightarrow{w}$	
4			$w' = F_K(1, x, y)$, erase K If $w = w'$ then output OK else output $\perp$

Fig. 4. A simplified variant of the PRF+PUF-based RFID scheme

simplify the protocol by eliminating unnecessary steps without changing its functionality. Figure 6 presents this new protocol.

Theorem 5.1: The mutual authentication scheme in Figure 6 provides strong privacy in the HPVP model, provided that P behaves randomly and F is a PRF.

Proof: Let Σ be the scheme in Figure 6. Assume that Σ is not strong private in the HPVP model, and let $\mathcal{A}$ be a strong adversary that can break Σ's privacy. We will show that there is an adversary $\mathcal{B}$ that has a non-negligible advantage in the pseudo-randomness game with F. Let $\mathcal{C}$ be a challenger for the pseudo-randomness game with F.

$\mathcal{B}$ will simulate Σ (will be the challenger) in the privacy game that A plays with Σ. So, $\mathcal{B}$ will have to simulate the oracles for $\mathcal{A}$. $\mathcal{B}$ does not know the secret parameters of the scheme but will want the simulation it performs to be indistinguishable from the real privacy game between $\mathcal{A}$ and Σ. We will show below how the oracles are simulated:

1) $\mathcal{B}$ keeps a list $\mathcal{R}$ of readers that will be created by adversary, and a list of tags $\mathcal{T}_R$ registered with each reader $R \in \mathcal{R}$. Initially, these lists are empty;
2) $\mathcal{B}$ keeps a list $\mathcal{T}$ of tags that will be created in the system in the order in which they are created. Each tag receives a fresh reference. We recall that the HPVP model allows the creation of several tags with the same identity. The corrupted (insider) tags will be stored in a separate list $c\mathcal{T}$ ($i\mathcal{T}$), initially empty;
3) $\mathcal{B}$ will simulate the tag T's PUF as a list of challenge-response pairs. Initially, this list, denoted $P(T)$, is empty. When evaluating the PUF on c, $\mathcal{B}$ looks in $P(T)$ a pair (c, r), for some r. If such a pair is found, r will be returned as the value of P on c; otherwise, $\mathcal{B}$ generates a random value r, returns it, and includes (c, r) in $P(T)$;
4) $\mathcal{B}$ keeps a list Γ of active triples $(vtag, T_0, T_1)$ as specified in the oracle $DrawnTag$. The oracle $Free(vtag)$ removes $(vtag, T_0, T_1)$ from Γ. Remark that Γ can contain at most one triple with $vtag$ in the first position;
5) $\mathcal{B}$ will keep a list $\mathcal{Q}$ of $(query, ext\_answer)$ pairs, where $query$ is a query of $\mathcal{A}$ and ext_answer is a possibly detailed information from which the answer to the query is extracted;
6) $CreateReader()$: $\mathcal{B}$ generates a unique reader reference R, answers to $\mathcal{A}$ with R, and includes R in $\mathcal{R}$. Moreover, $(CreateReader(), R)$ is included in $\mathcal{Q}$;
7) $CreateTag(ID)$: $\mathcal{B}$ generates a fresh tag reference T, associates it with ID, initializes $P(T)$ by the empty list, includes the pair (T, ID) in $\mathcal{T}$, and answers to $\mathcal{A}$ with T. Moreover, $(CreateTag(ID), T)$ is included in $\mathcal{Q}$;
8) $RegisterTag(T, R)$: $\mathcal{B}$ includes T in the list $\mathcal{T}_R$ and $(RegisterTag(T, R), \emptyset)$ in $\mathcal{Q}$;

	Server (Reader) (SimPUF P', PRF F)		**Prover (Tag)** (PUF P, PRF F)
1	$c \leftarrow \{0,1\}^\ell$	$\xrightarrow{c}$	
2		$\xleftarrow{x_1,\ y}$	$x_1 \leftarrow \{0,1\}^m$, $r \leftarrow P(c)$ $y := F_r(x_1)$
3	If $fail \leftarrow TREVERSE(c, P', x_1, y)$ then *abort* else let r^t be its output (i.e., $y = F_{r^t}(x_1)$) *authenticate tag*	$\xrightarrow{auth}$	
4		$\xleftarrow{x_2}$	$x_2 \leftarrow \{0,1\}^m$
5	$z := F_{r^t}(x_2)$	$\xrightarrow{z}$	
6			if $z = F_r(x_2)$ then *auth. server* else *abort*

Fig. 5. $PRF + SimPUF$-based authentication scheme in [23]

	Server (Reader) (SimPUF P', PRF F)		**Prover (Tag)** (PUF P, PRF F)
1	$c \leftarrow \{0,1\}^\ell$	$\xrightarrow{c}$	
2		$\xleftarrow{x,\ y}$	$r \leftarrow P(c)$, $x_1 \leftarrow \{0,1\}^m$ $y := F_r(x, 0)$
3	If $fail \leftarrow TREVERSE(c, P', x, y)$ then *abort* else let r^t be its output (i.e., $y = F_{r^t}(x, 0)$) *authenticate tag* $z := F_{r^t}(x, 1)$	$\xrightarrow{z}$	
4			if $z = F_r(x, 1)$ then *auth. server* else *abort*

Fig. 6. $PRF + SimPUF$-based strong private authentication scheme in the HPVP model

9) $Launch(R)$: $\mathcal{B}$ generates a fresh session identifier π, returns it to $\mathcal{A}$, and includes $(Launch(R), (R, \pi))$ in $\mathcal{Q}$;
10) $DrawTag(T_0, T_1)$: $\mathcal{B}$ checks if the constraints of the $DrawTag$ oracle are satisfied. If not, the answer is $\perp$. Otherwise, $\mathcal{B}$ generates a fresh virtual tag reference $vtag$, includes $(vtag, T_0, T_1)$ in Γ, and answers with $vtag$. In $\mathcal{Q}$ the pair $(DrawTag(T_0, T_1), \perp/vtag)$ is included, where $\perp/vatg$ is for the first/second case, resp.;
11) $Free(vtag)$: the triple whose first component is $vtag$ is removed from Γ (if it is in Γ). In this case, the pair $(Free(vtag), \emptyset)$ is included in $\mathcal{Q}$;
12) $SendTag(c, vtag)$: $\mathcal{B}$ extracts from Γ the triple whose first component is $vtag$. If no such triple exists, the answer is $\perp$. Otherwise, let $(vtag, T_0, T_1)$ be this triple. $\mathcal{B}$ searches each list $P(T_0)$ and $P(T_1)$ for a pair with c in the first position. If one of the lists does not contain such a pair, $\mathcal{B}$ generates a random r and includes (c, r) in that list. Now suppose that $(c, r_0) \in P(T_0)$ and $(c, r_1) \in P(T_1)$. $\mathcal{B}$ randomly generates x, queries $\mathcal{C}$ with $((r_0, x, 0), (r_1, x, 0))$ and returns (x, y). In $\mathcal{Q}$, $(SendTag(c, vtag), \perp/(c, r_0, r_1, x, y))$ is included depending on one of the two cases above;
13) $SendReader(R, (x, y), \pi)$: Since x is generated randomly at each query of a tag, and y is calculated from x through a PRF function, x and y can be found in at most one tuple (c, r_0, r_1, x, y) previously computed by $\mathcal{B}$ when R queried some tag by c. In addition, x and y can only appear independently with negligible probability. As a result, if $\mathcal{B}$ does not find in $\mathcal{Q}$ a tuple like the one above, it responds with $\perp$. Otherwise, it extracts the only tuple (c, r_0, r_1, x, y), queries $\mathcal{C}$ with $((r_0, x, 1), (r_1, x, 1))$, and returns the answer z of $\mathcal{C}$. In $\mathcal{Q}$, the pair
$$(SendReader(R, (x, y), \pi), \perp/(c, r_0, r_1, x, y, z))$$
is included depending on one of the two cases above;
14) $Result(\pi)$: Having the entire history of the privacy game up to the moment of this query, $\mathcal{B}$ can answer faithfully whether the tag was authenticated by the reader or not, or there is another case outside of these. In $\mathcal{Q}$, it will include $(Result(\pi), 1/0/\perp)$, depending on

the case;

15) $Corrupt(T)$: The tag T has no permanent variables, its only global temporary variables being c and x. Through corruption its PUF is destroyed. As a result, if the tag is not in $\mathcal{T}$, $\mathcal{B}$ has nothing to return to $\mathcal{A}$. Otherwise, it returns c and x, which $\mathcal{A}$ has learned from previous communications anyway. $\mathcal{B}$ moves then T form $\mathcal{T}$ into the list $c\mathcal{T}$ of corrupted tags. The pair $(Corrupt(T), \emptyset/(c, x))$ is included in $\mathcal{Q}$ (depending on the case);
16) $CreateInsider(ID)$: $\mathcal{B}$ creates a new tag reference T, associates it with ID, returns T to $\mathcal{A}$, and includes (T, ID) in $i\mathcal{T}$. Moreover, $(CreateInsider(ID), T)$ is included in $\mathcal{Q}$.

It is as clear as possible that the probability with which $\mathcal{B}$ guesses to which component, left or right, $\mathcal{C}$ applied the function F is precisely the probability with which $\mathcal{A}$ guesses with which tag, left or right, played the privacy game for the Σ scheme. Therefore, the assumption that the protocol is not strongly private will contradict the pseudo-randomness of F. So, the protocol must be strongly private. ■

It is interesting to compare the protocol in Figure 6 with the one in Figure 4. The differences are only from the point of view of how the PUF is viewed, deterministic or non-deterministic.

VI. Conclusions

The privacy properties offered by RFID protocols are mainly studied using ad hoc techniques. That makes it that when we study privacy in reputable privacy models, such as the Vaudenay or HPVP model, we find that many RFID protocols do not ensure privacy at all [29], [30], [36], [37].

The present work makes a short foray into the PRF+PUF paradigm used in the last 15 years to construct RFID protocols. The emphasis falls on the use of temporary variables in the construction of such protocols, as well as on the difference in approach to PUFs as deterministic devices (such as the protocol in Figure 4) or non-deterministic (such as the protocol in Figure 6). The privacy study is conducted in each case using Vaudenay's and HPVP models.

The fact that the protocol in Figure 6 does not ensure narrow forward privacy in Vaudenay's model but ensures strong privacy in the HPVP model shows that the two models significantly differ in approach when we go beyond the forward level of privacy.

References

[1] U. Rührmair and M. van Dijk, "PUFs in security protocols: Attack models and security evaluations," in *2013 IEEE Symposium on Security and Privacy*, 2013, pp. 286–300.

[2] W. Xie, L. Xie, C. Zhang, Q. Zhang, and C. Tang, "Cloud-based RFID authentication," in *2013 IEEE International Conference on RFID (RFID)*, 2013, pp. 168–175.

[3] Z. Zhao, "A secure RFID authentication protocol for healthcare environments using elliptic curve cryptosystem," *J. Medical Syst.*, vol. 38, no. 5, p. 46, 2014.

[4] C. Jin, C. Xu, X. Zhang, and J. Zhao, "A secure RFID mutual authentication protocol for healthcare environments using elliptic curve cryptography," *J. Medical Syst.*, vol. 39, no. 3, p. 24, 2015.

[5] H. Xiao, A. A. Alshehri, and B. Christianson, "A cloud-based RFID authentication protocol with insecure communication channels," in *2016 IEEE Trustcom/BigDataSE/ISPA*, 2016, pp. 332–339.

[6] H. Xu, J. Ding, P. Li, F. Zhu, and R. Wang, "A lightweight RFID mutual authentication protocol based on physical unclonable function," *Sensors*, vol. 18, no. 3, 2018.

[7] M. Safkhani, Y. Bendavid, S. Rostampour, and N. Bagheri, "On designing lightweight RFID security protocols for medical IoT," Cryptology ePrint Archive, Paper 2019/851, 2019.

[8] W. Liang, S. Xie, J. Long, K.-C. Li, D. Zhang, and K. Li, "A double PUF-based RFID identity authentication protocol in service-centric internet of things environments," *Information Sciences*, vol. 503, pp. 129–147, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025519305857

[9] F. Zhu, P. Li, H. Xu, and R. Wang, "A lightweight RFID mutual authentication protocol with PUF," *Sensors*, vol. 19, no. 13, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/13/2957

[10] K. Fan, Q. Luo, K. Zhang, and Y. Yang, "Cloud-based lightweight secure RFID mutual authentication protocol in IoT," *Information Sciences*, vol. 527, pp. 329–340, 2020.

[11] L. Xiao, H. Xu, F. Zhu, R. Wang, and P. Li, "SKINNY-based RFID lightweight authentication protocol," *Sensors*, vol. 20, no. 5, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/5/1366

[12] F. Zhu, P. Li, H. Xu, and R. Wang, "A novel lightweight authentication scheme for RFID-based healthcare systems," *Sensors*, vol. 20, no. 17, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/17/4846

[13] P. Gope and B. Sikdar, "A comparative study of design paradigms for PUF-based security protocols for iot devices: Current progress, challenges, and future expectation," *Computer*, vol. 54, no. 11, pp. 36–46, 2021.

[14] M. Shariq, K. Singh, M. Y. Bajuri, A. A. Pantelous, A. Ahmadian, and M. Salimi, "A secure and reliable RFID authentication protocol using digital Schnorr cryptosystem for IoT-enabled healthcare in COVID-19 scenario," *Sustainable Cities and Society*, vol. 75, p. 103354, 2021.

[15] V. Kumar, R. Kumar, S. Jangirala, S. Kumari, S. Kumar, and C.-M. Chen, "An enhanced RFID-based authentication protocol using PUF for vehicular cloud computing," *Security and Communication Networks*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251239918

[16] M. Adeli, N. Bagheri, S. Sadeghi, and S. Kumari, "χperbp: a cloud-based lightweight mutual authentication protocol," *Peer Peer Netw. Appl.*, vol. 16, no. 4, pp. 1785–1802, 2023.

[17] A. Kumar, K. Singh, M. Shariq, C. Lal, M. Conti, R. Amin, and S. A. Chaudhry, "An efficient and reliable ultralightweight RFID authentication scheme for healthcare systems," *Computer Communications*, vol. 205, pp. 147–157, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366423001329

[18] Y. Wang, R. Liu, T. Gao, F. Shu, X. Lei, G. Gui, and J. Wang, "A novel RFID authentication protocol based on a block-order-modulus variable matrix encryption algorithm," 2023.

[19] S. Vaudenay, "On privacy models for RFID," in *Proceedings of the Advances in Crypotology 13th International Conference on Theory and Application of Cryptology and Information Security*, ser. ASIACRYPT'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 68–87.

[20] J. Hermans, F. Pashalidis, Andreasand Vercauteren, and B. Preneel, "A new RFID privacy model," in *Computer Security – ESORICS 2011*, V. Atluri and C. Diaz, Eds. Berlin, Heidelberg: Springer Verlag, 2011, pp. 568–587.

[21] 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, Dec 2014.

[22] F. Armknecht, A.-R. Sadeghi, I. Visconti, and C. Wachsmann, "On RFID privacy with mutual authentication and tag corruption," in *Proceedings of the 8th International Conference on Applied Cryptography and Network Security*, ser. ACNS'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 493–510.

[23] Y. Gao, M. van Dijk, L. Xu, W. Yang, S. Nepal, and D. C. Ranasinghe, "TREVERSE: TRial-and-Error lightweight secure ReVERSE authentication with simulatable PUFs," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 1, pp. 419–437, 2022.

[24] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, 3rd ed. Chapman & Hall/CRC, 2020.

[25] M. Sipser, *Introduction to the Theory of Computation*. Cengage Learning, 2012.
[26] F. L. Ţiplea and C. Hristea, "Privacy and reader-first authentication in Vaudenay's RFID model with temporary state disclosure," Cryptology ePrint Archive, Report 2019/113, 2019, https://eprint.iacr.org/2019/113.
[27] F. L. Ţiplea 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.
[28] F. L. Ţiplea, C. Andriesei, and C. Hristea, "Security and privacy of PUF-based RFID systems," in *Cryptography - Recent Advances and Future Developments*. IntechOpen, 2021, iSBN 978-1-83962-566-4.
[29] F. L. Ţiplea, "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.
[30] F. L. Ţiplea, "Narrow privacy and desynchronization in Vaudenay's RFID model," *International Journal of Information Security*, vol. 22, pp. 563–575, June 2022.
[31] F. L. Ţiplea, C. Hristea, and R. Bulai, "Privacy and reader-first authentication in Vaudenay's RFID model with temporary state disclosure," *Comput. Sci. J. Moldova*, vol. 30, no. 3, pp. 335–359, 2022.
[32] J. Delvaux, "Security analysis of PUF-based key generation and entity authentication," 2017.
[33] A.-R. Sadeghi, I. Visconti, and C. Wachsmann, "PUF-enhanced RFID security and privacy," in *Workshop on secure component and system identification (SECSI)*, vol. 110, 2010.
[34] ——, *Enhancing RFID Security and Privacy by Physically Unclonable Functions*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 281–305.
[35] C. Hristea and F. L. Ţiplea, "Destructive privacy and mutual authentication in Vaudenay's RFID model," Cryptology ePrint Archive, Report 2019/073, 2019.
[36] F. L. Ţiplea, "On privacy of RFID-based authentication protocols," in *Proceedings of the 21st International Conference on Security and Cryptography - SECRYPT*, INSTICC. SciTePress, 2024, pp. 128–139.
[37] F. L. Ţiplea, "Security and privacy requirements for RFID schemes in healthcare: Case studies, solutions, and challenges," *Procedia Computer Science*, 2024, 28th International Conference on Knowledge Based and Intelligent Information and Engineering Sytems (KES 2024).