Chapter

Security and Privacy of PUF-Based RFID Systems

Ferucio Laurențiu Țiplea, Cristian Andriesei and Cristian Hristea

Abstract

The last decade has shown an increasing interest in the use of the physically unclonable function (PUF) technology in the design of radio frequency identification (RFID) systems. PUFs can bring extra security and privacy at the physical level that cannot be obtained by symmetric or asymmetric cryptography at the moment. However, many PUF-based RFID schemes proposed in recent years do not even achieve the lowest privacy level in reputable security and privacy models, such as Vaudenay's model. In contrast, the lowest privacy in this model can be achieved through standard RFID schemes that use only symmetric cryptography. The purpose of this chapter is to analyze this aspect. Thus, it is emphasized the need to use formal models in the study of the security and privacy of (PUF-based) RFID schemes. We broadly discuss the tag corruption oracle and highlight some aspects that can lead to schemes without security or privacy. We also insist on the need to formally treat the cryptographic properties of PUFs to obtain security and privacy proofs. In the end, we point out a significant benefit of using PUF technology in RFID, namely getting schemes that offer destructive privacy in Vaudenay's model.

Keywords: Radio Frequency Identification (RFID), Physically Unclonable Function (PUF), security, privacy

1. Introduction

Although the roots of the *Radio Frequency Identification* (RFID) technology can be traced back to World War II, the ancestor of modern RFID technology was introduced by Cardullo and Parks in 1973 [1] when the two proposed a passive radio transponder with memory. In recent years, RFID technology has become increasingly popular and its applicability has expanded to more and more diverse and complex domains and systems. It is worth mentioning here process automation, tracking and identification, toll collection, public transportation, national IDs and passports, medical healthcare systems, pharmaceutical systems, and so on.

From a scientific point of view, RFID has become a well-defined research field, counting more than fifteen thousand scientific papers and books indexed by IEEE, Springer, and Elsevier, and more than twenty-two thousand patents or patent applications indexed by the most essential three regional patent databases (USA, Europe, and Japan) [2]. All of these highlight a rich palette of research directions in RFID technology, such as: system implementation, design principles, chipless implementations, IoT integration, security, and so on.

An interesting aspect is that most of the RFID references cover technical aspects, applications, and protocol design, very few addressing security and privacy issues. The conclusion is that very few research papers dealing with RFID implementation or application start with security and privacy in mind. Obviously, there are RFID applications for which security and privacy are not so vital, such as human activity recognition (e.g., smart gym), environmental corrosive monitoring, soil monitoring, and so on. However, for other fields like people identification or healthcare systems [3, 4], security and privacy are crucial issues.

Attempts to improve the authentication process in RFID systems or make them resistant to physical attacks (tag corruption, for example) have led to the need to insert unclonable or tamper-evident physical objects into tags. Unclonability offers unique fingerprints to tags, while the tamper-evidence property would protect against corruption. Thus, physically unclonable functions (PUFs) [5–7] have found themself a suitable application in RFID technology and the researchers have already proposed a large spectrum of PUF-based RFID systems. However, the inclusion of PUFs in RFID systems (especially on tags) raises two key questions:

1. Are PUFs more efficient in implementation than ordinary cryptographic primitives?

2. Do PUFs provide security and privacy that standard cryptographic primitives cannot provide?

As with respect to the first question it is worth noting that an RFID implementation with strong security properties comes with increased cost for the final RFID product. This is the reason why some authors take into account the concept of *cost-effective protocol* [8]. As discussed in [9], the installation costs of current RFID solutions, not necessarily with improved hardware security, are not cheap at all, many different costs being involved when installing an RFID system (including maintenance and training).

As with respect to the second question, PUFs certainly offer security features that standard cryptographic primitives cannot provide. But if these security features are not used in a corresponding way, the result may be worse than if PUFs are not included. The lack of understanding of such issues has led many authors to propose PUF-based RFID schemes that are insecure or not at all private [10, 11] when analyzed in reputable models such as Vaudenay's security and privacy model [12, 13].

In this chapter, we want to highlight:

- The need to use PUFs in the construction of secure and private RFID schemes;

- The need to formalize the properties of PUFs to achieve provable security;

- The erroneous use of PUFs that does nothing but lead to insecure schemes and a lack of privacy.

The whole discussion is conducted on Vaudenay's security and privacy model. This model is currently considered one of the best RFID security and privacy models, offering a classification of the privacy of RFID schemes into eight classes. It is known that the strong privacy class cannot be obtained in this model, while the destructive privacy class can be obtained by using the PUF technology. This gives us an excellent example that justifies the opportunity to use PUFs in RFID technology.

2. RFID schemes and systems

An RFID system [14, 15] consists of a *reader*, a set of *tags*, and a *communication protocol* between reader and tags. The reader is a transceiver[1] that is connected through a secure channel with a back-end server, which is a powerful device that maintains a database with tag information. The reader's task is to identify *legitimate tags* (that is, tags with information stored in its database) and to reject all other incoming communication. The reader and its database are trusted entities, and the communication between them is secure. Many RFID protocols proposed so far do not make any separation between the reader and the back-end server. For this reason, the back-end server functions are considered to be taken over by the reader and, as a result, the reader is considered a powerful device not computationally restricted that can perform any cryptographic operation.

Opposite the reader, tags are small transponder[2] devices that are considered to be resource constrained. Depending on their class, they can perform only logical operations, symmetric encryption, or even public key cryptography. In practical scenarios, tags are attached to various items or carried by persons in order to facilitate some services when they are identified by readers.

The memory of a tag is typically split into *permanent* (or *internal*) and *temporary* (or *volatile*). The permanent memory stores the state values of the tag, while the temporary memory can be viewed as a set of *temporary variables* used to carry out the calculations required by the communication protocol. There are two types of temporary variables:

1. *local temporary variables*, used by tags only to do computations in a given protocol step;

2. *global temporary variables*. These get values in a given protocol step to be used in another protocol step.

From a formal point of view, an RFID scheme is defined as follows. Let $\mathcal{R}$ be a *reader identifier* and $\mathcal{T}$ be a set of *tag identifiers* whose cardinal is polynomial in some *security parameter*[3] λ. An *RFID scheme over* $(\mathcal{R}, \mathcal{T})$ [12, 13] is a triple $S = (SetupR, SetupT, Ident)$ of *probabilistic polynomial time* (PPT) *algorithms*[4], where:

1. $SetupR(\lambda)$ inputs a security parameter λ and outputs a triple (pk, sk, DB) consisting of a key pair (pk, sk) and an empty database DB. pk is public, while sk is kept secret by reader;

2. $SetupT(pk, ID)$ initializes the tag identified by ID. It outputs an initial tag state S and a tag specific secret K. The pair (ID, K) is stored in the reader's database DB;

[1] Contraction from transmitter and receiver.

[2] Contraction from transmitter and responder.

[3] A security parameter usually specifies a minimum security value, such as the minimum length of an encryption key.

[4] A probabilistic (or randomized) algorithm is an algorithm that uses uniformly random bits as an auxiliary input to guide its behavior, in the hope of achieving good performance in the "average case" over all possible choices of random bits. A polynomial time algorithm is an algorithm that runs in polynomial time with respect to the size of its input.

3. $Ident(pk; \mathcal{R}(sk, DB); ID(S))$ is an interactive protocol between the reader identified by $\mathcal{R}$ (with its private key *sk* and database *DB*) and a tag identified by *ID* (with its state *S*) in which the reader ends with an output consisting of *ID* or a special symbol $\perp$. The tag may end with no output (*unilateral authentication*), or it may end with an output consisting of *OK* or $\perp$ (*mutual authentication*).

By calling $SetupR(\lambda)$ one should understand that a reader identified by $\mathcal{R}$ is created, initialized, and some public parameters of the system are also established. We simply refer to the reader such created as being $\mathcal{R}$. By calling $SetupT(pk, ID)$, a tag identified by *ID* is created, initialized, and registered with the reader by storing some information about it in *DB*. We denote this tag by $\mathcal{T}_{ID}$. The meaning of the reader's output *ID* ($\perp$) is that it authenticates (rejects) the tag. Similarly, the tag outputs *OK* ($\perp$) when it authenticates (rejects) the reader.

The *correctness* of an RFID scheme means that regardless of how the system is set up, after each complete execution of the interactive protocol between the reader and a legitimate tag, the reader outputs the tag's identity with overwhelming probability. For mutual authentication, correctness asks for one more requirement, namely that the tag outputs *OK* with overwhelming probability.

An *RFID system* is an instantiation of an RFID scheme. This is done by a trusted *operator* $\mathcal{I}$ who runs the RFID scheme over a reader identifier $\mathcal{R}$ and a set $\mathcal{T}$ of tag identifiers. In a given setting, the reader is initialized exactly once, while each tag at most once. Thus, the reader's database does not store multiple entries for the same tag. However, different settings with the same RFID scheme may initialize the reader and the tags in different ways.

We close the section by an example of a fundamental RFID scheme, namely the PRF-based RFID scheme proposed in [13]. To describe the scheme, let us assume that λ is a security parameter, $\ell_1(\lambda)$ and $\ell_2(\lambda)$ are two polynomials, and $F = (F_K)_{K \in \mathcal{K}}$ is a *pseudo-random function*[5] (PRF), where $F_K : \{0,1\}^{2\ell_1(\lambda)} \to \{0,1\}^{\ell_2(\lambda)}$ for all $K \in \mathcal{K}$.

Each tag is equipped with a random key *K* and has the capacity to compute F_K. 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* is its random key.

The protocol is given in **Figure 1** (the use of "$\leftarrow$" specifies a random selection of an element from a set). As we can see, the reader sends initially a random

	Reader (DB)		**Tag** (K)
1	$x \leftarrow \{0,1\}^{\ell_1(\lambda)}$	$\xrightarrow{x}$	
2		$\xleftarrow{y,\ z}$	$y \leftarrow \{0,1\}^{\ell_1(\lambda)}, z = F_K(x,y)$
	If $\exists (ID, K) \in DB$ s.t. $z = F_K(x,y)$ then output *ID* (tag auth.) else output $\perp$		

Figure 1.
PRF-based RFID scheme.

[5] A pseudo-random function is a collection $F = (F_K)_K$ of efficiently-computable functions with the property that no efficient algorithm can distinguish (with significant probability) between a function chosen randomly from this family and a random function (a function whose outputs are fixed at random).

$x \leftarrow \{0,1\}^{\ell_1(\lambda)}$ to the tag. On receiving it, the tag generates a random $y \leftarrow \{0,1\}^{\ell_1(\lambda)}$, computes $z = F_K(x,y)$, and answers with (y,z). The reader checks its database for a pair (ID,K) such that $z = F_K(x,y)$. If such a pair is found, it outputs ID (that is, authenticates the tag); otherwise, outputs $\perp$.

3. Security and privacy models for RFID

The design of an RFID scheme must start from consistent motivations for its usefulness and the desired security and privacy level, in a particular model of security and privacy, for the scheme to be proposed. The second desideratum requires that proofs of security and privacy accompany the proposed scheme. Ideally, the scheme designer should know in advance security and privacy models for RFID schemes and thus to offer his scheme in such a model. However, the practice shows that, although various fairly good security and privacy models have been proposed over time, many authors propose RFID schemes for which they study security and privacy in an ad hoc way without referring to the existing models. It is not surprising then that many of these schemes, analyzed in reputable models, do not reach the lowest level of security or privacy [11].

In this section, we aim to discuss one of the most critical security and privacy models for RFID, namely Vaudenay's model. We argue that this model falls into the class of gray-box models, and then make a consistent analysis of the corruption oracle in this model. The emphasis on this oracle is more than necessary, both for ordinary tags and for tags endowed with physically unclonable functions.

The discussion in this section can also be rephrased for other models that offer the corruption ability to the adversary, such as the model based on indistinguishability proposed in [16]. However, the choice of Vaudenay's model for the discussion in this chapter is a matter of the authors' scientific taste and their belief that it is one of the fundamental models for studying security and privacy properties of RFID schemes.

3.1 Security and privacy models

A *security* or *privacy model* for a cryptographic construction consists of an *attack model* and a *security* or *privacy goal*, respectively. The attack model specifies the adversary's power, while the security or privacy goal specifies the property we are interested to be achieved by the cryptographic construction. Nowadays, researchers differentiate between three attack models [17]:

1. The *black-box model*: this is the traditional model where the adversary can only observe the response of the cryptographic construction when it is queried by inputs of the adversary's choice (the adversary may know the algorithms used in the cryptographic construction);

2. The *gray-box model*: this includes the black-box model and supplementary the adversary may use side-channel information such as power consumption, electro-magnetic radiation, or timing information;

3. The *white-box model*: this has been introduced in particular for software implementation of the cryptographic constructions. In this model, the adversary is assumed to have full control over the implementation and its execution environment.

For instance, the security model IND-CCA means that the security goal is *indistinguishability* (*semantic security*) and the attack model is the *chosen ciphertext attack* [18]. The power of the adversary in this model is specified by giving him access to an *encryption* and *decryption oracles* that assists the adversary to collect a polynomial size set of (plaintext,ciphertext) pairs.

The black-box model does not depend on the software or hardware implementation, platform, and so on. In contrast to it, the gray-box model of attack exploits the algorithm/protocol implementation. For instance, the side-channel analysis that can be used with this model may take into account fluctuations in timing delays, power consumption, or emitted signals and radiation [19]. The result of such an analysis varies depending on the implementation, the platform on which it is implemented, the measuring devices. Side-channel analysis is local and not global.

3.2 Vaudenay's RFID security and privacy model

One of the most influential security and privacy model for RFID is *Vaudenay's model* [12, 13]. In this model, the adversary is a PPT algorithm that is allowed to interact with the RFID scheme. This means that the adversary may create tags to play with them as being the reader (but without having direct access to the reader's database). The adversary may also play with the reader as being any of the tags created by it. Depending on the adversary, it may or may not have access to the tags' internal memory. From a formal point of view, the adversary interacts with the RFID scheme by means of a set of oracles. Before describing these oracles, we mention that each tag in Vaudenay's model is either *free* (i.e., outside the interaction area of the adversary) or *drawn* (i.e., in the interaction area of the adversary). When a tag is created, it is free. The adversary may draw a free tag at any time and, in the end, to free it.

Now, the oracles in Vaudenay's model are the following:

1. $CreateTag^b(ID)$: When the adversary queries this oracle by ID for some bit b, the oracle calls the algorithm $SetupT(pk, ID)$ to generate a pair (K, S) and create a tag $\mathcal{T}_{ID}$ with the identifier ID and initial state S. If $b = 1$, (ID, K) is added to DB and the tag is considered *legitimate*; otherwise ($b = 0$), the tag is considered *illegitimate*. The tag thus created is considered *free*;

2. $DrawTag(\delta)$: By this oracle, the adversary is allowed to interact with free tags according to some probability distribution δ (on these tags). Therefore, this oracle chooses a number of free tags according to 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*.

As one can see, *DrawTag* provides the adversary with access to some free tags by means of temporary identifiers, and gives information on whether the tags are legitimate or not (but no other information);

3. $Free(vtag)$: By this oracle, the adversary may free the drawn tag $vtag$. The identifier $vtag$ will no longer be used. We assume that when a tag is freed, its temporary state is erased. This is a natural assumption that corresponds to the fact that the tag is no longer powered by reader;

4. *Launch*(): When the adversary queries this oracle, it means that it wants to launch a new protocol instance. Therefore, the oracle returns to it a unique identifier to be used with this protocol instance;

5. *SendReader*(m, π): By this oracle, the adversary gets 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. We emphasize that the reader's answer is conceived as the message sent to the tag by the communication channel and not as the reader's decision output (tag identity or $\perp$). Therefore, if the reader does not send anything to the tag, the output of this oracle is empty;

6. *SendTag*(m, *vtag*): This oracle 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. As in the case of the *SendReader* oracle, we emphasize that the tag's answer is conceived as the message sent to the reader by the communication channel and not as the tag's decision output (*OK* or $\perp$). Therefore, if the tag does not send anything to the reader, the output of this oracle is empty;

7. *Result*(π): By this oracle, the adversary is allowed to know the reader's decision with respect to the authentication of the tag in session π. More precisely, the oracle outputs $\perp$ if in session π the reader has not yet made a decision on tag authentication (this also includes the case when the session π does not exist), 1 if in session π the reader authenticated the tag, and 0 otherwise (this oracle is both for unilateral and mutual authentication);

8. *Corrupt*(*vtag*): This oracle 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).

It is customary to assume that the RFID tags can be corrupted to reveal not only their permanent memory but also the global temporary variables [20]. When the *Corrupt* oracle is considered in such a way, we will refer to Vaudenay's model as being *Vaudenay's model with temporary state disclosure*. We emphasize that "corruption with temporary state disclosure" means corruption of the permanent state and of the global temporary variables, but not of the local temporary variables (more details are provided in Section 3.4).

Now, the adversaries are classified into the following classes, according to the access they get to these oracles:

- *Weak adversaries*: they do not have access to the *Corrupt* oracle;

- *Forward adversaries*: if they access the *Corrupt* oracle, then they can only access the *Corrupt* oracle;

- *Destructive adversaries*: after the adversary has queried *Corrupt*(*vtag*) and obtained the corresponding information, the tag identified by *vtag* is destroyed and the temporary identifier *vtag* wil no longer be available. The database *DB* will still keep the record associated to this tag (the reader does not know the tag was destroyed). As a consequence, a new tag with the same identifier cannot be

created (in this approach, the database cannot store multiple records for the same tag identifier);

- *Strong adversaries*: there are no restrictions on the use of oracles.

If we further restrict the adversary to access the *Result* oracle, we obtain four new classes: *narrow weak*, *narrow forward*, *narrow destructive*, and *narrow strong*.

Now we are ready to introduce the *tag* and *reader authentication* properties as proposed in [12, 13], simply called the *security* of RFID schemes.

An RFID scheme has the property of *tag authentication* if no strong adversary has more than a negligible advantage in causing the reader to authenticate an uncorrupted legitimate tag in a protocol instance where the reader had no conversation with that tag to lead upon its authentication.

An RFID scheme has the property of *reader authentication* if no strong adversary has more than a negligible advantage in causing an uncorrupted legitimate tag to authenticate the reader in a protocol instance where the tag had no conversation with the reader to lead upon its authentication.

Privacy in Vaudenay's model generalizes anonymity (which means that the tag ID cannot be inferred) and untraceability (which means that the equality of two tags cannot be inferred). Thus, privacy requires that no adversary can infer non-trivial tag ID relations from the protocol messages. The information provided by a protocol is trivial when the adversary may learn it without making effective use of the protocol messages. To formalize this, Vaudenay's model introduces the concept of a *blinder* that simulates the protocol for adversary without knowing any secret information of the tags or the reader. If this simulation does not change the adversary's output compared to the case when the adversary plays with the real protocol, then the protocol achieves privacy.

A *blinder for an adversary* $\mathcal{A}$ that belongs to some class V of adversaries is a PPT algorithm $\mathcal{B}$ that:

1. simulates the *Launch*, *SendReader*, *SendTag*, and *Result* oracles for $\mathcal{A}$, without having access to the corresponding secrets;

2. passively looks at the communication between $\mathcal{A}$ and the other oracles allowed to it by the class V (that is, $\mathcal{B}$ gets exactly the same information as $\mathcal{A}$ when querying these oracles).

When the adversary $\mathcal{A}$ interacts with the RFID scheme by means of a blinder $\mathcal{B}$, we say that $\mathcal{A}$ is *blinded by* $\mathcal{B}$ and denote this by $\mathcal{A}^{\mathcal{B}}$. We emphasize that $\mathcal{A}^{\mathcal{B}}$ is allowed to query the oracles *Launch*, *SendReader*, *SendTag*, and *Result* only by means of $\mathcal{B}$; all the other oracles are queried in the standard way.

Given an adversary $\mathcal{A}$, an RFID scheme S, and a blinder $\mathcal{B}$, define the following experiment (privacy game) that a challenger sets up for $\mathcal{A}$:

Privacy experiment $RFID^{prv}_{\mathcal{A},S,\mathcal{B}}(\lambda)$
1: $b \leftarrow \{0,1\}$;
2: Set up the reader;
3: $\mathcal{A}^b$ gets the public key pk;
4: $\mathcal{A}^b$ queries the oracles;
5: $\mathcal{A}^b$ gets the secret table of the *DrawTag* oracle;
6: $\mathcal{A}^b$ outputs a bit b';
7: Return 1 if $b = b'$ and 0, otherwise,

where $\mathcal{A}^0$ stands for $\mathcal{A}$ and $\mathcal{A}^1$ stands for $\mathcal{A}^{\mathcal{B}}$.

An RFID scheme achieves privacy for a class V of adversaries if for any adversary $\mathcal{A} \in V$ there exists a blinder $\mathcal{B}$ such that $\mathcal{A}$ has a negligible advantage over 1/2 to distinguish between the *real privacy game* (the bit b is 0 in $RFID^{prv}_{\mathcal{A},S,\mathcal{B}}(\lambda)$) from the *blinded privacy game* (the bit b is 1 in $RFID^{prv}_{\mathcal{A},S,\mathcal{B}}(\lambda)$).

We thus obtain eight concepts of privacy: *strong privacy*, *narrow strong privacy*, *destructive privacy*, and so on. The diagram in **Figure 2** shows the relationship between the eight privacy concepts in Vaudenay's model in the context of unilateral authentication. In this diagram, "N-x" is a shortcut for "narrow x". An arrow from A to B means that A-privacy implies B-privacy.

3.3 Vaudenay's model is a gray-box model

Let us take one last look at Vaudenay's model to fit it into one of the three classes presented at the beginning of Section 3. The attack model associated with it falls in the class of gray-box models. Indeed, all the oracles except *Result* and *Corrupt* are specific to the black-box model because they do not output anything about the internal components of the algorithm implementation.

The *Result* oracle facilitates non-invasive side-channel analysis. Obviously, there may be situations in which the adversary can see the final result of the reader (the reader signals non-authentication of the tag, a gate opens, etc.). But, just as well, there are situations in which the adversary cannot see the final result of the reader without use of a specialized oracle. The analysis of Vaudenay's model clearly shows that the *Result* oracle makes a big difference between protocols that ensure privacy against an adversary that has the possibility to use this oracle and protocols that ensure privacy against an adversary that does not have the possibility to use this oracle.

The *Corrupt* oracle provides the adversary with information about the internal memory of the tag. Although data stored in the internal memory of the tag (such as symmetric keys, public keys) does not depend on implementation or platform, it is internal information of the tag. The need for this oracle results from the fact that tags are devices with poor physical protection. For low-cost tags, corruption could be accomplished and thus the information stored in the permanent tag memory can be retrieved. Temporary (volatile) memory loses its data when the power is interrupted. However, the memory remanence effect may allow to recover some data. As a result, we can say that it is natural to consider the possibility of obtaining the information from the tag memory by various techniques called generically "corruption". Once this information is obtained, the analysis is a theoretical one, abstracting the implementation.

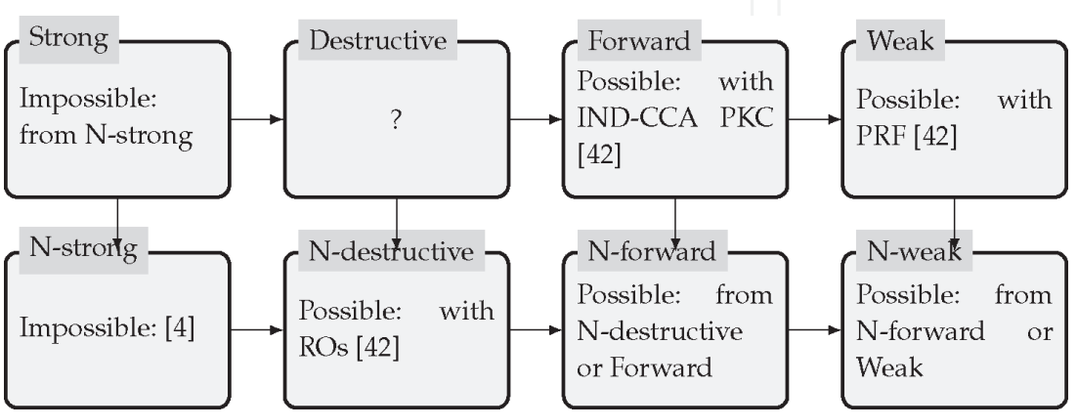

Figure 2.
Privacy and mutual authentication of RFID schemes in Vaudenay's model without temporary state disclosure (PKC stands for public-key cryptography and RO for random oracle).

As a conclusion:

1. Vaudenay's attack model falls in the category of gray-box models. It provides the adversary with general information, including a limited amount of side-channel information that does not depend on the implementation or implementation platform;

2. Side-channel analysis that is not covered by Vaudenay's model comes as an additional analysis. It depends on the implementation of the protocol, implementation platform, measuring devices, etc.

3.4 Corruption with temporary state disclosure

When Vaudenay's model was proposed [13], it was somewhat unclear whether the *Corrupt* oracle returns the full (i.e., permanent and temporary) tag state or only the permanent one. This has also remained unclear in Paisie and Vaudenay's next year paper [12] on mutual authentication. While the distinction between full and permanent state did not have a negative impact on the results already obtained in the case of unilateral authentication, it highlighted several wrong results in the case of mutual authentication [21]. Thus, one of the results in [21], namely Theorem 1, says that there is no RFID scheme that achieves both reader authentication and narrow forward privacy in Vaudenay's model with temporary state disclosure. The argument is as follows. Given a blinder $\mathcal{B}$, one may construct an adversary $\mathcal{A}_{sec}^{\mathcal{B}}$ against reader authentication so that, if the scheme is narrow forward private then $\mathcal{A}_{sec}^{\mathcal{B}}$ has non-negligible advantage to authenticate itself as a valid reader. Going inside the proof, we remark that it is crucial the *Corrupt* oracle returns the full state of a tag in order to allow an adversary to perform the test by which the tag authenticates the reader. By this test, the adversary distinguishes with non-negligible probability between the real privacy game and the blinded one.

4. Physically unclonable functions

Purely cryptographic and mathematical techniques can provide security in a black-box or partially gray-box model. As we argued in the previous section, Vaudenay's model is a gray-box model. Within this model, no RFID scheme is known, built only on symmetric or asymmetric cryptographic primitives, which would offer destructive privacy. No one has indeed proved the non-existence of such a scheme, but we firmly believe that there is no such scheme. However, if we add physical security objects to the RFID schemes, then we can obtain RFID schemes that are destructive private [22].

Physically unclonable functions (PUFs) are possible candidates that can provide physical security in that they can ensure the secure generation and storage of the cryptographic keys [5–7]. A PUF can be seen as a *physical object* that evaluates a *noisy functions*: when queried with a challenge x it generates a response y that depends on both x and its unique and specific properties which are hard to *clone*. The PUFs are noisy because their specific properties can change with the operating conditions such as supply voltage or ambient temperature. So, PUFs may return slightly different responses when queried with the same challenge multiple times.

During the last years, the PUF concept attracted the attention of the research community and industry. Many research papers and patents focusing on

implementing distinct PUF architectures, larger systems employing PUFs as separate units or protocols dedicated to PUF-based implementations were proposed.

4.1 PUF construction

In principle, PUFs can be constructed with any physical entity or structure as long as an intrinsic mismatching or nonlinear behavior, inherent to such entity when implementing multiple alike, could be exploited.

For instance, two identical transistors designed in the same technology and on the same mask will show slightly different performances after implementing their layout (real physical circuit). The main difference will be noticed for the threshold voltage, VTH in the case of CMOS process, different for both transistors. As such, a simple CMOS PUF could be obtained when implementing an array of identical transistors, this being also the first architecture reported in literature for chip identification [23] and disclosed in a patent application filed in 1989 [24]. Based on how challenges are applied to the circuit input and the great number of distinct responses (keys) that can be obtained, this particular PUF architecture is a strong PUF, at least according to PUF properties reiterated in [25]. A similar approach, yet implemented with bipolar transistors, was disclosed in a European patent application filed in 2013 [26].

Another example, even simpler, is that of a discrete electronic part, be it through-hole or surface-mount resistor or capacitor. It is well known that there are no two identical resistors or capacitors even though they have, theoretically, the same value and tolerance and are produced by the same manufacturer. Tolerance gives us valuable information about how much less or more the resistance or capacitance value is different of its nominal value. This sort of uncertainty favors PUF applications even though is not good from design perspective. And this is how the first RC PUF came into existence [27, 28].

Looking back, many PUF architectures have been proposed during the last two decades, various intrinsic properties being exploited, with many distinct classes identified [25]. This field encompasses so many implementations, technologies and design principles that two different perspectives to classify PUF architectures were used in that review. However, taking into account the scope and field of our study, i.e. RFID, we consider that the second classification (PUF tree), based on mechanism and evaluation parameter, is more relevant. In this regard, the PUF implementations fall into four classes: electronic, optical, radio frequency and magnetic PUFs. Furthermore, since RFID tags have limited chip area and (power) design constraints, it is obvious that electronic PUF architectures, known also as silicon-PUF, are of interest for our study.

Silicon-based PUFs involve conventional integrated circuit design techniques. Two essential design hints are identified regarding the implementation of a particular silicon-based PUF architecture:

1. A PUF architecture should generate at its output a unique sequence, useful either for authentication or cryptographic key generation, developed based on silicon intrinsic (physical) particularities. Therefore, no memory cells are allowed to store such a (PUF response) sequence. However, a (SRAM or DRAM) memory cell could be used to implement a PUF cell and thus generate a single bit of the PUF response because we are not interested in the binary value memorized in that cell but rather of the transition speed and delay, which are specific to that particular cell;

2. When it comes to intrinsic behavior, PUF construction starts either at transistor or system level. In the first case, it exploits certain anomalies in transistor functionality that could identify a particular circuit similar to a fingerprint, such implementation being reported in some references as analog PUF. In the second case, it uses specific differences that appear when connecting identical logic gates, as it is the case of ring oscillators or SRAM/DRAM cells array. In such implementation, the randomness property is based on intrinsic variations, at gate level, but the property is exploited and adjusted by digital designers in such manner that the spread of generated patterns (responses) is extended as much as possible. This is the reason why, such class of PUF architectures is reported in literature as digital PUF. The system-level approach favors FPGA based PUF implementations, the FPGA having all digital gates already manufactured, hence it lacks access to the transistor level. The most part of the PUF articles published during the last decade make use of FPGA. Either way, silicon PUF implementation is uniquely favored by the tolerance inherent to manufacturing process, the leading cause of device mismatching. It seems that what deteriorates the real performances of a particular silicon product, becomes quite useful for chip identification/cloning detection and key generation.

Silicon PUFs are still the most appealing ones because they occupy a very small chip area, especially when implemented in smaller technologies (¡65 nm CMOS process), therefore they can be integrated into larger electronic units and systems (such as RFID). In addition, their design and preliminary testing on FPGA development boards ensure their proof of concept reproducibility, feasibility and success, before going deeper to implement a dedicated chip. A selection of representative silicon PUF architectures reported in literature is given below (for more details the reader may consult [6, 20, 29]):

2000: Threshold voltage (TV) PUF [24];
2002: Ring oscilator (RO) PUF [30];
2004: Arbiter PUF (APUF) [31];
2007: SRAM PUF [32], LATCH PUF [33];
2008: Butterfly (B) PUF [34], D Flip-Flop (DFF) PUF [35];
2009: Power distribution (PD) PUF [36], CNN PUF [37];
2010: Super High Information Content (SHIC) PUF [38], Glitch PUF [39];
2011: Pseudo-LFSR (PL) PUF [40];
2012: Buskeeper PUF [41];
2013: Micro-electrico-mechanical system (MEMS) PUF [42];
2014: Transient effect RO (TERO) PUF [43];
2015: Dynamic random access memory (DRAM) PUF [44], SA_PUF [?];
2016: D-PUF [45];
2017: Aging-resistant Current-starved RO (ACRO) PUF [46];
2018: Cryptanalysis/Robust Multiplexer-based PUF (cMPUF/rMPUF) [47].

4.2 Cryptographic properties of PUFs and idealization

In cryptography and security we typically build a cryptographic system and prove its security under the assumption that we have used secure ingredients (building blocks) such as *collision-resistant hash functions* (CRHF), *pseudo-random generators* (PRGs), or *pseudo-random functions* (PRFs). These secure ingredients are a kind of "ground truth" of applied cryptography. "Provable security" typically starts only above the level of these secure ingredients. A proof based on

experiments and simulations may only show that the scheme is secure with respect to those experiments and simulations. A proof based on ideal primitives has a major advantage: if a cryptographic primitive is assumed ideal and later is proved (by experiments) insecure, we may change it by another one of the same type that we believe is secure. The entire scheme remains unchanged and the security analyses is moved to the cryptographic primitives.

When a cryptographic construction is deployed in practice, the secure (ideal) primitives that underlie it are replaced by algorithms for which we do not have a theoretical proof of security. Instead, these algorithms are subjected to intense scrutiny by cryptographers to see if they resist all known classes of attacks and to get evidence supporting the assumption that they are secure.

PUFs have been introduced to physically supplement specific security properties that cannot be satisfactorily obtained at the software implementation level alone. The security properties offered by PUFs can only be highlighted through experiments and simulations. To be able to apply provable security to cryptographic constructions that include PUFs, it is necessary to formalize their security properties. The major problem that arises in this context is to maintain a balance between formalization and the real physical properties. The difficulty of maintaining this balance comes from the fact that it is quite challenging to capture the behavior of a physical object through a mathematical formula that is accurate or that approximates it well enough. Without such a balance, we can reach situations such as those in which either the formalization is not useful or is too strict and has no practical equivalent. As a result, the formalization must be sufficiently realistic and, at the same time, allow its use in provable security.

Among the basic properties we want from a PUF class we mention: [left=.5cm]

Constructability – this means that it is "easy" to construct a random instance of a given PUF class;

Evaluability – this includes constructability and further requires that any random instance of a given PUF class can be easily evaluated on any random challenge;

Reproductibility – this includes evaluability and further requires that the responses resulting from evaluating the same challenge on the same PUF instance should be similar (in some distance metric) with high probability;

Uniqueness – this includes evaluability and further requires that the responses resulting from evaluating the same challenge on different PUF instances should be dissimilar (in some distance metric) with high probability;

Identifiability – this means both reproducibility and uniqueness;

Physical unclonability – this includes evaluability and further requires that it is hard to create a new PUF instance that is more alike to a given PUF instance than expressed by the uniqueness property;

Unpredictability – this means evaluability and further requires that no PPT algorithm can predict the answer of a given PUF instance for a given challenge, except with negligible probability, even if it could have previously learned the PUF's answer for a polynomial number of challenges (different from the challenge in question);

One-wayness – this includes evaluability and further requires that it is hard to invert the answer of a given PUF instance;

Tamper-evidence – this includes evaluability and further requires that it is hard to physically alter a given PUF instance without having a noticeable effect on its challenge-response behavior.

The choice of the PUF type to be included in a cryptographic system depends on the security properties we want to achieve, and which cannot be obtained through software techniques, as well as on the production costs. For example, the tamper-evidence feature can be handy for constructing destructive private RFID schemes. However, today's technological development shows that only optical [48] and coating PUFs [49] can provide this property. Besides, such PUFs have high production

costs, which requires a careful analysis of the environment of the utilization of the RFID schemes that would use such PUFs.

5. PUF-based RFID systems

PUFs have proven to be suitable for integration into RFID systems to ensure their security in gray or white box models. So far, two significant directions for the use of PUFs in RFID systems have emerged. We dedicate this section to a discussion of the two directions and the issues that arise regarding them.

5.1 Endowing RFID tags by PUFs

The vulnerability of RFID systems to corruption consists in the fact that an adversary with corruption abilities can extract the information from the tag's memory and, thus, can impersonate it or, at least, destroy the privacy property. Without having a concrete proof at the moment, the researchers' opinion is that, in Vaudenay's model but not only, destructive privacy cannot be achieved only by using symmetric or asymmetric cryptographic primitives. Storing a private key in the tag's memory is useless when the adversary has corruption capabilities and can use the information obtained through corruption. The use of a public key system in which the private key is stored on the reader side is also useless in Vaudenay's model when destructive privacy is desired.

This discussion naturally leads to the idea of using a tamper-evident mechanism embedded in the tag to help the process of identifying and authenticating it. In this context, PUFs seem to be a good choice and the newest technologies show that it is possible to embed PUFs into tags. These kind of tags, with PUFs embedded into them, will be called *PUF tags*, while the standard tags will sometimes be referred to as *ordinary tags*. A PUF-based RFID scheme is an RFID scheme with PUF tags.

How PUF tags can be built can be very important in terms of tag corruption. This aspect will be touched on in the next section.

Two significant directions have emerged on the authentication protocol of PUF-based RFID schemes. The first direction treats PUFs as fingerprints [50–54]. This approach requires an initial configuration phase in which a PUF model or a large set of PUF challenge-and-response (CR) pairs is pre-computed and stored in the reader's database. To identify a PUF tag, the reader queries it by some challenge, the tag evaluates its PUF on the challenge, and then the reader compares the tag's response with the pre-computed response it already has stored in the database. There are several variants of this scenario, but regardless of these, special attention must be paid to the modeling attacks of PUFs [55]. This is because the adversary might get sufficient CR pairs in order to simulate the tag's PUF. Anyway, the authors of this paper are not aware of any PUF-based RFID schemes based on this approach, and that would provide destructive privacy in Vaudenay's model. Moreover, we believe that it is not possible to achieve this level of privacy through this approach because the set of CR pairs is generally polynomial in size. Then, a strong enough adversary may run the authentication protocol with a tag until it exhausts all CR pairs stored in the database. In such a situation, either a CR pair will be reused, or a reset mechanism has to be used. Regardless of the case, the privacy property might be compromised.

A second direction for the authentication protocol of PUF-based RFID schemes starts from the idea of using PUFs as cryptographic key generators or as storage methods [10, 22, 56, 57]. That is, the tag evaluates its PUF only to generate or extract a cryptographic key. Thus, the PUF is evaluated for a minimum number of challenges. This fact eliminates the shortcoming that the adversary can model the

PUF, but if the PUF is noisy, then an additional overhead may be incurred by using fuzzy extractors. Assuming PUFs are tamper-evident, this second approach produces schemes that achieve destructive privacy in Vaudenay's model (please see Section 5.3).

5.2 Tag corruption and PUFs

In order to adapt Vaudenay's model (with or without temporary state disclosure) to PUF-based RFID schemes, we have to clarify what corruption means in this case. At least two main scenarios are possible:

1. By corrupting a PUF tag, the adversary gets the state of the tag, according to the type of the attack model (with or without temporary state disclosure). Besides, the tag is destroyed, but its PUF can still be evaluated. This variant does not show significant differences compared to the case of corruption of ordinary tags, because the PUF of the tag can now be seen as a public function that the adversary can evaluate as he wishes;

2. By corrupting a PUF tag, the adversary gets the state of the tag, according to the type of the attack model (with or without temporary state disclosure). Besides, the tag and its PUF are destroyed (in this case, the PUF cannot anymore be evaluated).

The second scenario is the most significant. Within it, the PUF tag is seen as a tamper-evident device (circuit), such as a tamper-evident PUF [58, 59]. Working in this scenario, Theorem 1 in [21], at least in its present form, cannot be applied to PUF-based RFID schemes. This leaves open the invitation to PUF-based design RFID schemes that achieve mutual authentication and higher privacy levels than narrow forward in Vaudenay's model with temporary state disclosure. As we have already said, such schemes cannot be based on ordinary tags. A good choice is to use PUF tags, as it was done in [10, 22, 56, 57, 60]. However, the use of PUF tags does not mean that the schemes are immune to corrupting adversaries. This is because an adversary might not need the entire tag state to attack the scheme. An example in this sense is provided in [10] where it was shown that the RFID schemes proposed in [56, 57] do not achieve mutual authentication and (narrow) destructive privacy in Vaudenay's model with temporary state disclosure, as it was claimed by authors, although they use PUF tags. The proof exploits the fact that these schemes use volatile variables to carry values between protocol steps.

As we have seen, the corruption attack in Vaudenay's model may provide the adversary with the full state of the tag. However, this state does not include the values of the local temporary variables. The varied range of side-channel attacks includes other types of attacks, such as those called cold-boot attacks, through which the tag's memory can be frozen. Thus the adversary can obtain the value of the local variables at a given time. This type of attack has also been discussed in RFID-oriented papers, such as [56, 57, 61]. We are not aware of any formal treatment of this scenario in Vaudenay's model. To implement it in Vaudenay's model, the *Corrupt* oracle should be changed to return snapshots of the tag's state during its computation (recall that the standard *Corrupt* oracle returns the tag's state before or after a protocol step). A formal and complete treatment of such a corruption seems hard to reach; on the other side, such a corruption is very strong and probably no PUF-based RFID scheme may achieve a privacy level higher than (narrow) weak under such a corruption. However, special cases may be relevant. One of them is the cold boot attack mentioned

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

Figure 3.
PRF- and PUF-based RFID scheme that achieves destructive privacy and mutual authentication

above [56, 57, 61]. To defeat it, a PUF double evaluation technique was proposed in [61], which consists of two evaluations in a row of the same PUF. If the attack is applied immediately after the first PUF evaluation, the second PUF evaluation is lost, and vice-versa. This technique was implemented in [56, 57] too. Unfortunately, the authors did not pay much attention to the temporary variables, which made their schemes not to achieve even the narrow forward privacy level [10].

5.3 Destructive privacy by PUF-based RFID schemes

When the Vaudenay [12, 13] model was proposed, finding an RFID scheme to provide destructive privacy remained an open issue (please see the diagram in **Figure 2**). This problem was later solved by a PUF-based RFID scheme [22, 60]. The scheme, which provides unilateral authentication, is obtained from the PRF-based RFID scheme presented in Section 2, adding tamper-evident PUFs to tags to generate the key K. If the adversary corrupts the tag, its PUF is destroyed and cannot be evaluated. Thus, the adversary cannot get the key K. The scheme was extended later to ensure mutual authentication [10]. We present it in **Figure 3**. As one can see, the main difference between the scheme in **Figure 1** and this new one is that the domain of the PRF function F is extended with one more bit and the tag is endowed with a tamper-evident PUF P and a seed s for it. Whenever the tag needs to evaluate its PRF, it first computes the key $K = P(s)$ and then uses it. It has to be understood that after using it, the variable K is erased. If the adversary corrupts the tag, the seed s he gets is useless because the PUF can no longer be evaluated (please see [10] for details regarding the security and privacy proofs).

As corruption with temporary state disclosure is a real threat in practice, the most natural question is how to extend the above schemes, or how to design new ones, secure and private in Vaudenay's model under such a corruption. It is clear that ordinary tags (i.e., tags that only implement cryptographic primitives) do not help if one wants to achieve both mutual authentication and privacy (Theorem 1 in [21]). Endowing tags with PUFs is a potential solution but it is not a guarantee. It turns out that the subtlety is how to use temporary variables. This has been missed in some recently proposed RFID schemes [56, 57], which made these schemes not to achieve the privacy level claimed by authors [10]. It seems that the use of temporary variables in connection with mutual authentication and privacy is not really very well understood, especially under corruption with temporary state disclosure.

6. Conclusions

The significant impact of PUF technology in the construction of RFID systems is demonstrated by the great diversity of scientific articles and patents proposed in the last decade. The use of PUFs in the construction of RFID schemes can bring extra security and privacy at the physical level that cannot be obtained by symmetric and asymmetric cryptography at the moment. However, this requires an adequate understanding and analysis of security and privacy models for RFID to consider PUFs only if existing standard techniques cannot lead to the desired security and privacy level. Unfortunately, the literature shows us enough PUF-based RFID schemes proposed in recent years that do not even reach the weak privacy level in Vaudenay's model. In contrast, weak privacy in this model can be achieved through standard RFID schemes that use only symmetric cryptography. This fact clearly shows that a sustained effort is needed to consolidate the understanding of the concept of security and privacy model and adapt it accordingly to PUF technology.

In this chapter, we highlighted the aspects mentioned above and emphasized the need to use formal models in the study of security and privacy properties of (PUF-based) RFID schemes. Achieving the level of destructive privacy in Vaudenay's model through PUF-based RFID schemes clearly shows us the potential of using PUF technology in the construction of RFID systems. Even if the security and privacy proofs on PUF-based RFID schemes make use of ideal PUFs, this is not a negative aspect as long as there is practically reasonable support for idealization, and this is in the trend of technology evolution.

Authors contribution

This chapter (structure and content) was proposed by F.L. Țiplea, who also supervised its complete realization. Section 4.1 was prepared by C. Andriesei, as well as the second and third paragraphs of the introductory section. All the other sections of the chapter were prepared in an equal contribution by F.L. Țiplea and C. Hristea. All authors have read and agreed to the published version of the manuscript.

Author details

Ferucio Laurențiu Țiplea[1,3*], Cristian Andriesei[2,4] and Cristian Hristea[3]

1 Alexandru Ioan Cuza University of Iasi, Iasi, Romania

2 Gheorghe Asachi Technical University, Iasi, Romania

3 Simion Stoilow Institute of Mathematics of the Romanian Academy, Bucharest, Romania

4 SC AT&C Technology SRL, Iasi, Romania

*Address all correspondence to: fltiplea@gmail.com

References

[1] Marion Cardullo and William Parks. Transponder apparatus and system, Jan 1973. US Patent 3713148

[2] FPO. Free Patents Online, 2020. http://freepatentsonline.com

[3] Haddara M, Staaby A. RFID applications and adoptions in healthcare: A review on patient safety. *Procedia computer science*. 2018;**138**: 80-88

[4] Antti Lahtela. A short overview of the RFID technology in healthcare. In *2009 Fourth International Conference on Systems and Networks Communications*, pages 165–169. IEEE, 2009

[5] Halak B. *Physically Unclonable Functions: From Basic Design Principles to Advanced Hardware Security Applications*. Springer International Publishing; 2019

[6] R Maes. Physically Unclonable Functions: Constructions, Properties and Applications. Springer-Verlag Berlin Heidelberg; 2013

[7] Christian Wachsmann and Ahmad-Reza Sadeghi. *Physically Unclonable Functions (PUFs): Applications, Models, and Future Directions*. Number 12 in Synthesis Lectures on Information Security, Privacy, & Trust. Morgan & Claypool Publishers, Dec 2014

[8] Jones EC, Chung CA. *RFID and Auto-ID in Planning and Logistics: A Practical Guide for Military UID Applications*. CRC Press; 2016

[9] Ustundag A. *The Value of RFID: Benefits vs*. Costs. Springer-Verlag, London; 2013

[10] Cristian Hristea and Ferucio Laurențiu Țiplea. Destructive privacy and mutual authentication in Vaudenay's RFID model. Cryptology ePrint Archive, Report 2019/073, 2019. https://eprint.iacr.org/2019/073

[11] Cristian Hristea and Ferucio Laurențiu Țiplea. Privacy of stateful RFID systems with constant tag identifiers. *IEEE Transactions on Information Forensics and Security*, 15: 1920–1934, Nov 2019.

[12] Radu-Ioan Paise and Serge Vaudenay. Mutual authentication in RFID: Security and privacy. In *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security*, ASIACCS '08, pages 292–299, New York, NY, USA, 2008. ACM

[13] 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. pages 68-87

[14] Klaus Finkenzeller. RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification. Wiley Publishing, 3rd edition, 2010

[15] Yingjiu Li H. Robert Deng, and Elisa Bertino. *RFID Security and Privacy*. Synthesis Lectures on Information Security, Privacy, and Trust. In: Morgan & Claypool Publishers. 2013

[16] Hermans J, Peeters R, Preneel B. Proper RFID privacy: Model and protocols. *IEEE Transactions on Mobile Computing*. Dec 2014;**13**(12): 2888-2902

[17] Pascal Sasdrich, Amir Moradi, and Tim Güneysu. White-box cryptography in the gray box. In *Revised Selected Papers of the 23rd International Conference on Fast Software Encryption - Volume 9783*, FSE 2016, pages 185–203, Berlin, Heidelberg, 2016. Springer-Verlag

[18] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman & Hall/CRC, 2nd edition, 2014

[19] Peeters E. *Advanced DPA Theory and Practice: Towards the Security Limits of Secure Embedded Circuits*. Incorporated: Springer Publishing Company; 2013

[20] M. Al-Haidary and Q. Nasir. Physically unclonable functions (PUFs): A systematic literature review. In *2019 Advances in Science and Engineering Technology International Conferences (ASET)*, pages 1–6, 2019

[21] Armknecht F, Sadeghi A-R, Scafuro A, Visconti I, Wachsmann C. Impossibility results for RFID privacy notions. In: Gavrilova ML, Tan CJK, Moreno ED, editors. *Transactions on Computational Science XI*. Berlin, Heidelberg: Springer-Verlag; 2010. pp. 39-63

[22] Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. PUF-enhanced RFID security and privacy. In *Workshop on secure component and system identification (SECSI)*, volume 110, 2010

[23] K. Lofstrom, W. R. Daasch, and D. Taylor. IC identification circuit using device mismatch. In *2000 IEEE International Solid-State Circuits Conference. Digest of Technical Papers (Cat. No.00CH37056)*, pages 372–373, 2000

[24] K. Lofstrom. System for providing an integrated circuit with a unique identification. US patent no. 6161213, Dec 2000

[25] McGrath T, Bagci IE, Wang ZM, Roedig U, Young RJ. A PUF taxonomy. *Applied Physics Reviews*. 2019;**6**(1): 011303

[26] Vanhoucke T and Nguyen V. A PUF method using and circuit having an array of bipolar transistors. European patent no. 2833287A1, July 2013

[27] Lee S, Oh M-K, Kang Y, Choi D. Design of resistor-capacitor physically unclonable function for resource-constrained IoT devices. *Sensors*. 2020;**20** (2), pages 326-337

[28] Sangjae Lee, Mi-Kyung Oh, Yousung Kang, and Dooho Choi. RC PUF: A low-cost and an easy-to-design PUF for resource-constrained IoT devices. In Ilsun You, editor, *Information Security Applications*, pages 275–285, Cham, 2020. Springer International Publishing

[29] Herder C, Yu M, Koushanfar F, Devadas S. Physical unclonable functions and applications: A tutorial. *Proceedings of the IEEE*. 2014;**102**(8): 1126-1141

[30] Blaise Gassend, Dwaine E. Clarke, Marten van Dijk, and Srinivas Devadas. Silicon physical random functions. In Vijayalakshmi Atluri, editor, *ACM Conference on Computer and Communications Security*, pages 148–160. ACM, 2002

[31] Jae W. Lee, Daihyun Lim, Blaise Gassend, G. Edward Suh, Marten van Dijk, and Srinivas Devadas. A technique to build a secret key in integrated circuits for identification and authentication applications. In *2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No.04CH37525)*, pages 176–179, 2004

[32] Jorge Guajardo, Sandeep S. Kumar, Geert-Jan Schrijen, and Pim Tuyls. FPGA intrinsic PUFs and their use for IP protection. In *Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems*, CHES 07, pages 63–80, Berlin, Heidelberg, 2007. Springer-Verlag

[33] Y. Su, J. Holleman, and B. Otis. A 1.6pJ/bit 96% Stable Chip-ID Generating Circuit using Process Variations. In *2007 IEEE International*

Solid-State Circuits Conference. Digest of Technical Papers, pages 406–611, 2007

[34] S. S. Kumar, J. Guajardo, R. Maes, G. Schrijen, and P. Tuyls. Extended abstract: The butterfly PUF protecting IP on every FPGA. In *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*, pages 67–70, 2008

[35] Maes R, Tuyls P, Verbauwhede I. Intrinsic PUFs from flip-flops on reconfigurable devices. In: *3rd Benelux workshop on information and system security (WISSec 2008)*. 2008

[36] Ryan Helinski, Dhruva Acharyya, and Jim Plusquellic. A physical unclonable function defined using power distribution system equivalent resistance variations. In *2009 46th ACM/IEEE Design Automation Conference*, pages 676–681. IEEE, 2009

[37] Csaba G, Xueming J, Chen Q, Porod W, Schmidhuber J, Schlichtmann U, et al. On-chip electric waves: An analog circuit approach to physical uncloneable functions. *IACR Cryptology ePrint Archive*. 2009;**2009**: 246

[38] Ulrich Ruehrmair, Christian Jaeger, Christian Hilgers, Michael Algasinger, Gyoergy Csaba, and Martin Stutzmann. Security applications of diodes with unique current-voltage characteristics. In *Proceedings of the 14th International Conference on Financial Cryptography and Data Security*, FC'10, page 328–335, Berlin, Heidelberg, 2010. Springer-Verlag

[39] Daisuke Suzuki and Koichi Shimizu. The Glitch PUF: A new delay-PUF architecture exploiting glitch shapes. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 366–382. Springer, 2010

[40] Yohei Hori, Hyunho Kang, Toshihiro Katashita, and Akashi Satoh. Pseudo-LFSR PUF: A compact, efficient and reliable physical unclonable function. In *2011 International Conference on Reconfigurable Computing and FPGAs*, pages 223–228. IEEE, 2011

[41] Peter Simons, Erik van der Sluis, and Vincent van der Leest. Buskeeper PUFs, a promising alternative to D flip-flop PUFs. In *2012 IEEE International Symposium on Hardware-Oriented Security and Trust*, pages 7–12. IEEE, 2012

[42] Patrick Koeberl, Ünal Kocabas, and Ahmad-Reza Sadeghi. Memristor PUFs: a new generation of memory-based physically unclonable functions. In *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 428–431. IEEE, 2013

[43] Bossuet L, Ngo XT, Cherif Z, Fischer V. A PUF based on a transient effect ring oscillator and insensitive to locking phenomenon. *IEEE Trans. Emerg. Top. Comput.* 2014;**2**(1):30-36

[44] Fatemeh Tehranipoor, Nima Karimian, Kan Xiao, and John Chandy. DRAM based intrinsic physical unclonable functions for system level security. In *Proceedings of the 25th edition on Great Lakes Symposium on VLSI*, pages 15–20, 2015

[45] S. Sutar, A. Raha, and V. Raghunathan. D-PUF: An intrinsically reconfigurable DRAM PUF for device authentication in embedded systems. In *2016 International Conference on Compliers, Architectures, and Sythesis of Embedded Systems (CASES)*, pages 1–10, 2016

[46] Liu CQ, Cao Y, Chang CH. ACRO-PUF: A low-power, reliable and aging-resilient current starved inverter-based ring oscillator physical unclonable function. *IEEE Transactions on Circuits and Systems I: Regular Papers*. 2017; **64**(12):3138-3149

[47] Sahoo DP, Mukhopadhyay D, Chakraborty RS, Nguyen PH. A multiplexer-based arbiter PUF composition with enhanced reliability and security. *IEEE Transactions on Computers*. 2018;**67**(3):403-417

[48] Pappu Srinivasa Ravikanth. Physical One-Way Functions. PhD thesis, USA, 2001

[49] Pim Tuyls, Geert-Jan Schrijen, Boris Škorić, Jan van Geloven, Nynke Verhaegh, and Rob Wolters. Read-proof hardware from protective coatings. In *Proceedings of the 8th International Conference on Cryptographic Hardware and Embedded Systems*, CHES'06, pages 369–383, Berlin, Heidelberg, 2006. Springer-Verlag

[50] Pier Francesco Cortese, Francesco Gemmiti, Bernardo Palazzi, Maurizio Pizzonia, and Massimo Rimondini. Efficient and practical authentication of PUF-based RFID tags in supply chains. In *2010 IEEE International Conference on RFID-Technology and Applications*, pages 182–188. IEEE, 2010

[51] Devadas S, Suh E, Paral S, Sowell R, Ziola T, Khandelwal V. Design and implementation of PUF-based unclonable RFID ICs for anti-counterfeiting and security applications. In: *2008 IEEE international conference on RFID*, pages 58–64. IEEE. 2008

[52] Gope P, Lee J, Quek TQS. Lightweight and practical anonymous authentication protocol for RFID systems using physically unclonable functions. *IEEE Transactions on Information Forensics and Security*. Nov 2018;**13**(11):2831-2843

[53] Öztürk E, Hammouri G, Sunar B. Towards robust low cost authentication for pervasive devices. In: *2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 170–178. IEEE. 2008

[54] Anthony Van Herrewege, Stefan Katzenbeisser, Roel Maes, Roel Peeters, Ahmad-Reza Sadeghi, Ingrid Verbauwhede, and Christian Wachsmann. Reverse fuzzy extractors: Enabling lightweight mutual authentication for PUF-enabled RFIDs. In *International Conference on Financial Cryptography and Data Security*, pages 374–389. Springer, 2012

[55] Ulrich Rührmair, Frank Sehnke, Jan Sölter, Gideon Dror, Srinivas Devadas, and Jürgen Schmidhuber. Modeling attacks on physical unclonable functions. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, pages 237–249, New York, NY, USA, 2010. Association for Computing Machinery

[56] Mete Akgün and M. Ufuk Çaglayan. Providing destructive privacy and scalability in RFID systems using PUFs. *Ad Hoc Netw.*, 32(C):32–42, September 2015

[57] Süleyman Kardas, Serkan Çelik, Muhammet Yildiz, and Albert Levi. PUF-enhanced offline RFID security and privacy. *J. Netw. Comput. Appl.*, 35 (6):2059–2067, November 2012

[58] Dmitry Nedospasov, Jean-Pierre Seifert, Clemens Helfmeier, and Christian Boit. Invasive PUF analysis. In *Proceedings of the 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography*, FDTC '13, pages 30–38, USA, 2013. IEEE Computer Society

[59] Pim Tuyls and Lejla Batina. RFID-tags for anti-counterfeiting. In *Cryptographers' Track at the RSA Conference*, pages 115–131. Springer, 2006

[60] Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. *Enhancing RFID Security and Privacy by Physically Unclonable Functions*, pages 281–305. Springer Berlin Heidelberg, 2010

[61] Süleyman Kardas, Mehmet Sabir Kiraz, Muhammed Ali Bingöl, and Hüseyin Demirci. A novel RFID distance bounding protocol based on physically unclonable functions. In Ari Juels and Christof Paar, editors, *RFID. Security and Privacy*, pages 78–93, 2012. Springer Berlin Heidelberg