https://doi.org/10.52326/ic-ecco.2022/SEC.07

Vulnerabilities of LRSAS Protocol

Denisa-Ionela Țîflea [1], ORCID: 0000-0002-5962-4390
[1] Department of Computer Science, Faculty of Computer Science,
"Alexandru Ioan Cuza" University of Iași, Iași, România, e-mail: tifleadenisa@gmail.com

***Abstract*—The construction of secure and private RFID protocols is necessary with the increasing application of RFID technology in increasingly diverse fields. While security refers to unilateral or mutual authentication depending on the protocol, privacy is a more elaborate concept to which many studies and research have been dedicated. Unfortunately, many RFID protocols are still being developed without consistent security and privacy analysis in well-defined models, such as the Vaudenay model. In this paper, we aim to prove that a recently proposed authentication protocol, LRSAS, does not achieve any form of privacy in Vaudenays model.**

***Keywords*—RFID system, security, privacy**

I. INTRODUCTION

Radio Frequency Identification (RFID) has its origin when people started to remotely identify objects using radio technology, inventing radar. Nowadays, RFID technology market is increasing, thanks to its multitude of applications, including IoT, tracking and identification. Because its principle of working is based on data transfer, over the last couple of years there was a substantial effort to assure security and privacy over RFID schemes through protocols that can be applied to RFID systems.

A privacy model helps to define the mentioned concepts using different types of adversaries. The model that we follow in this paper is Vaudenay's model [1], being among the most influential and widely accepted security and privacy model.

If we analyze some of the protocols that were proposed in specialized literature we can identify flaws in their design that lead to vulnerabilities. Some of them are a result of not considering privacy models when designing the RFID scheme.

Contribution: In this paper, we develop an analysis of the privacy of the LRSAS protocol [2]. After inspecting the experiments and their probability results we came to the conclusion that the protocol has design flaws and doesn't achieve the class of privacy that the authors mentioned, considering Vaudenay's model.

A discussion on the possibility of fixing the protocol concludes our paper.

Paper structure: This paper is structured in six sections, the first one being the introduction. In Section 2 we present some elementary concepts about RFID schemes and systems. Section 3 consists of presenting the RFID lightweight authentication protocol, LRSAS. The main part of the paper is exposing the vulnerabilities of LRSAS in Section 4, followed by a discussion about the improvements of the protocol in Section 5. We conclude with Section 6.

II. RFID SCHEMES AND SYSTEMS

From an informal point of view, an RFID system [3], [4] consists of a *reader*, a set of *tags*, and a *communication protocol* between reader and tags. The reader is a transceiver that has associated a database that stores information about tags. Its task is to identify *legitimate tags* (that is, tags with information stored in its database) and to reject all the other incoming communication. The reader and its database are trusted entities, and the communication between them is secure. A tag is a transponder device with much more limited computation capabilities than the reader. Depending on tag, it can perform simple logic operations, symmetric key, or even public key cryptography. Each tag has a *permanent* (or *internal*) *memory* that stores the state values, and a temporary (or volatile) memory that can be viewed as a set of *volatile variables* used to carry out the necessary computations.

RFID schemes. Let $\mathcal{R}$ be a *reader identifier* and $\mathcal{T}$ be a set of *tag identifiers* whose cardinal is polynomial in some security parameter λ. An RFID *scheme over* $(\mathcal{R}, \mathcal{T})$ [1], [5] is a triple $\mathcal{S}$*=(SetupR, SetupT, Ident)* of PPT algorithms, where:

1) SetupR(λ) 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 $\mathcal{S}$ and a secret key *K*. A triple *(ID, f(S),K)* is stored in the reader's database *DB*, where *f* is a public function that extracts some information from tag's initial state $\mathcal{S}$;

3) Ident(pk; $\mathcal{R}$(sk,DB); ID($\mathcal{S}$)) 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 $\mathcal{S}$) in which the reader ends with an output consisting of ID or $\perp$. The tag may end with no output (*unilateral authentication*), or it may end with an output consisting of *OK* or $\perp$ (*mutual authentication*).

4) SetupR(λ) creates a reader identified by R and initializes it, *SetupT(pk, ID)* creates a tag $\mathcal{T}_{\text{ID}}$ for each tag identified by *ID*, initializes it with an initial tag state, and also register this tag with the reader by storing some information about it in the reader's database.

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 tag's identity with overwhelming probability. For mutual authentication RFID schemes, *correctness* means that the reader outputs tag's identity and the tag outputs *OK* with overwhelming probability.

An RFID *system* is an instantiation of an RFID scheme.

Adversaries. The two most basic security requirements for RFID schemes are *authentication* and *untraceability*. To formalize them, the concept of an *adversary model* is needed. There have been several proposal for this, such as [1], [5], [6], [7], [8], [9], [10], [11]. One of the most influential, which we follow in this paper, is *Vaudenay's model* [1], [5]. We recall below this model as in [12]. Thus, we assume first that some oracles the adversary may query share and manage a common list of tags *ListTags*, which is initially empty. This list includes exactly one entry for each tag created and active in the system. A tag entry consists of several fields with information about the tag, such as: the (permanent) identity of the tag (which is an element from $\mathcal{T}$), the temporary identity of the tag (this field may be empty saying that the tag is *free*), a bit value saying whether the tag is legitimate (the bit is one) or illegitimate (the bit is zero). When the temporary identity field is non-empty, its value uniquely identifies the tag, which is called *drawn* in this case. The adversary may only interact with drawn tags by means of their temporary identities.

The oracles an adversary may query are:

1) CreateTag^b(ID): Creates a free tag $\mathcal{T}_{\text{ID}}$ with the identifier ID by calling the algorithm *SetupT(pk, ID)* to generate a pair *(K, S)*. If *b = 1, (ID, f(S),K)* is added to *DB* and the tag is considered *legitimate;* otherwise *(b = 0)*, the tag is considered *illegitimate.* Moreover, a corresponding entry is added to *ListTags;*

2) DrawTag(δ): This oracle chooses a number of free tags according to the distribution δ, let us say *n*, and draws them. That is, *n* temporary identities $vtag_1, \ldots, vtag_n$ are generated and the corresponding tag entries in *ListTags* are filled with them. The oracle outputs $(vtag_1, b_1, \ldots, vtag_n, b_n)$, where b_i specifies whether the tag $vtag_i$ is legitimate or not;

3) Free(vtag): Removes the temporary identity *vtag* in the corresponding entry in *ListTags*, and the tag becomes free. The identifier *vtag* will no longer be used. We assume that when a tag is freed, its temporary state is erased;

4) Launch(): Launches a new protocol instance and assigns a unique identifier to it. The oracle outputs the identifier;

5) SendReader(m, π): Outputs the reader's answer when the message *m* is sent to it as part of the protocol instance π. When *m* is the empty message, abusively but suggestively denoted by $\emptyset$, this oracle outputs the first message of the protocol instance π, assuming that the reader does the first step in the protocol;

6) SendTag(m, vtag): Outputs the tag's answer when the message *m* is sent to the tag referred to by *vtag*. When *m* is the empty message, this oracle outputs the first message of the protocol instance π, assuming that the tag does the first step in the protocol;

7) Result(π): Outputs $\perp$ if in session π the reader has not yet made a decision on tag authentication (this also includes the case when the session π does not exist), 1 if in session π the reader authenticated the tag, and 0 otherwise (this oracle is both for unilateral and mutual authentication);

8) Corrupt(vtag): Outputs the current permanent (internal) state of the tag referred to by *vtag*, when the tag is not involved in any computation of any protocol step (that is, the permanent state before or after a protocol step).

We emphasize that *Corrupt* does not return snapshots of the tag's memory during its computations. When the *Corrupt* oracle returns the full state, we will refer to this model as being *Vaudenay's model with temporary state disclosure.*

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*: once they access the *Corrupt* oracle, they can only acces the *Corrupt* oracle;

- *Destructive adversaries*: after querying *Corrupt(vtag)* and obtaining the corresponding information, the tag identified by *vtag* is destroyed (marked as destroyed in *ListTags*) and the temporary identifier *vtag* will 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;
- *Strong adversaries*: there are no restrictions on the use of oracles.

Orthogonal to these classes, there is the class of *narrow* adversaries that do not have access to the *Result* oracle. We may now combine the narrow constraint with any of the previous constraints in order to get another four classes of adversaries, *narrow weak, narrow forward, narrow destructive,* and *narrow strong.*

Security. Now we are ready to introduce the *tag* and *reader authentication* properties as proposed in [1], [5], simply called the security of RFID shemes. First of all, we say that a tag $\mathcal{T}_{\text{ID}}$ and a protocol session π *had a matching conversation* if they exchanged well interleaved and faithfully (but maybe with some time delay) messages according to the protocol, starting with the first protocol message but not necessarily completing the protocol session. If the matching conversation leads to tag authentication, then it will be called a *tag authentication matching conversation*; if it leads to reader authentication, it will be called a *reader authentication matching conversation*.

Tag authentication property is defined by means of an experiment that a challenger sets up for a *strong* adversary $\mathcal{A}$ (after the security parameter λ is fixed). In the experiment the adversary is given the public parameters of the scheme and is allowed to query the oracles. If there has been a session in which the reader has authenticated an uncorrupted tag without a tag authentication matching conversation then the experiment returns 1 (or 0 otherwise).

The advantage of $\mathcal{A}$ in the experiment $RFID_{\mathcal{A},S}^{t\_auth}(\lambda)$ is defined as

$$Adv_{\mathcal{A},S}^{t\_auth}(\lambda) = Pr(RFID_{\mathcal{A},S}^{t\_auth}(\lambda) = 1)$$

An RFID scheme S achieves tag authentication if $Adv_{\mathcal{A},S}^{t\_auth}(\lambda)$ is negligible, for any strong adversary $\mathcal{A}$.

The experiment for reader authentication, denoted $RFID_{\mathcal{A},S}^{r\_auth}(\lambda)$, is quite similar to that above. The main difference compared to the previous experiment is that the adversary $\mathcal{A}$ tries to make some legitimate tag to authenticate the reader. As π and $\mathcal{T}_{\text{ID}}$ have no matching conversation, $\mathcal{A}$ computes at least one message that makes the tag to authenticate the reader.

An RFID scheme $\mathcal{S}$ achieves *reader authentication* if the advantage of $\mathcal{A}$, $RFID_{\mathcal{A},S}^{r\_auth}(\lambda)$, is negligible, for any strong adversary $\mathcal{A}$ ($Adv_{\mathcal{A},S}^{r\_auth}(\lambda)$ is defined as above, by using $RFID_{\mathcal{A},S}^{r\_auth}(\lambda)$ instead of $RFID_{\mathcal{A},S}^{t\_auth}(\lambda)$).

Privacy. Privacy for RFID systems [5] captures anonymity and untraceability. It basically means that an adversary cannot learn anything new from intercepting the communication between a tag and the reader. To model this, the concept of a blinder was introduced in [5].

A *blinder for an adversary* $\mathcal{A}$ that belongs to some class *V* of adversaries is a PPT algorithm $\mathcal{B}$ that simulates the *Launch*, *SendReader*, *SendTag* and *Result* oracles for $\mathcal{A}$, without having access to the corresponding secrets. Moreover, it looks passively 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 $\mathcal{A}^{\mathcal{B}}$.

Given an adversary $\mathcal{A}$, define the experiment (privacy game):

Experiment $\text{RFID}_{\mathcal{A},S}^{prv\_0}(\lambda)$

1: Set up the reader;
2: $\mathcal{A}$ gets the public key *pk*;
3: $\mathcal{A}$ queries the oracles;
4: $\mathcal{A}$ gets the secret table of the *DrawTag* oracle;
5: $\mathcal{A}$ outputs a bit *b'*;
6: Return *b'*.

In the same way, by replacing "$\mathcal{A}$" with "$\mathcal{A}^{\mathcal{B}}$" we define the experiment $RFID_{\mathcal{A},S,B}^{prv\_1}(\lambda)$. Now, the *advantage* of $\mathcal{A}$ blinded by $\mathcal{B}$ is

$$Adv_{\mathcal{A},S,B}^{prv}(\lambda) = |P(RFID_{\mathcal{A},S}^{prv\_0}(\lambda) = 1) - P(RFID_{\mathcal{A},S,B}^{prv\_1}(\lambda) = 1)|$$

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

III. LRSAS PROTOCOL

LRSAS [2] is an RFID mutual authentication protocol based on the light-weight block cipher algorithm *SKINNY*.

SKINNY [13] is a tweakable block cipher algorithm that has different block and key sizes depending on the environment of the application, having superior perfomance both in hardware implementations and in adaptability. It has 64-bit and 128-bit version for the block size *n* and *n*, *2n*, and *3n* versions for the key size *t*. The steps required for encryption are the initialization phase

and the round function, wich consists of the following operations: SubCells, AddConstants, AddRoundTweakey, ShiftRows and MixColumns. For more information on how the cryptosystem works, the reader is referred to [13].

A lightweight protocol is a category of RFID systems protocols, among mature protocols, simple protocols and ultra-lightweight protocols. Being in a balance of costs and assuring security and privacy, lightweight protocols are usually chosen over other types of protocols. In this setup, a tag has support for simple functions (cyclic redundancy code check) and pseudo-random number generators. Unlike simple and mature protocols, it cannot compute one-way hash functions.

In this protocol, the following version of SKINNY algorithm is adopted: the block has 128 bits, key size is 128 bits and the encryption round is 40 times.

The authors use the following notations to describe the protocol:

- $\mathcal{R}$: reader,
- $\mathcal{T}$: tag,
- *ID*: tag identifier,
- *FID*: encryption of *ID*,
- *K*: key shared by $\mathcal{T}$ and $\mathcal{R}$,
- *r*: random number, generated by $\mathcal{R}$,
- $\oplus$: XOR operation,
- *En(X)*: Encryption using SKINNY.

The protocol is composed of four phases in order to achieve mutual authentication: *initialization* phase, *tag indentification* phase, *mutual authentication* phase and *update* phase.

Before describing the above phases, note that a tag keeps in its permanent memory its *ID* (96 bits), *FID* (96 bits) and the shared key *K* (128 bits), the last two being updated after each authentication, where *FID* is the ciphertext obtained by encrypting *ID* using SKINNY.

Initialization phase: The database *DB* will store two entries for every tag identity *ID*: $\{ID, FID^{old}, K^{old}\}$ and $\{ID, FID^{new}, K^{new}\}$, values that are obtained by reader in the communication with the tag in the previous and current protocol instances.

Tag identification phase: After the reader sends a request, the tag responds with its identifier, FID^{new}. If the reader recognizes the message as the FID^{new} as it is in the database *DB*, the authentication phase starts. If the message received is FID^{old} then the pair (FID^{old}, K^{old}) is used for authentication and the tag may be subjected to desyncronization attack.

Mutual authentication phase: The reader generates *r*, computes M_1 and M_2, then sends $M_1 \| M_2$ to tag.

$$M_1 = FID \oplus r$$
$$M_2 = En(FID \oplus ID \oplus r)$$

The tag computes *r*' and M_2'. If M_2 and M_2' are equal, the reader is authenticated, else, the authentication ends.

$$r' = M_1 \oplus FID$$
$$M_2' = En(M_2' \oplus r')$$

After that, the tag calculates M_3' and transmits it to the reader.

$$M_3' = En(M_2' \oplus r')$$

After the reader receives M_3', it will compute M_3. If M_3 and M_3' are the same, the tag is authenticated. Otherwise, the authentication stops.

$$M_3 = En(M_2 \oplus r)$$

Update phase: After the mutual authentication, the update phase starts with two options depending on the pair that the reader is using to authenticate the tag:

- if the reader uses (FID^{old},K^{old}), then the database *DB* will not modify the pseudonym and shared key;
- if the reader uses (FID^{new}, K^{new}), the database *DB* will be updated ($FID^{old} = FID^{new}$, $K^{old} = K^{new}$, $FID^{new} = M_1$), where K^{new} is computed through the key schedule module of SKINNY.

If the tag receives the *OK* message, it will update both its own FID^{new} with M_1, and the K^{new}.

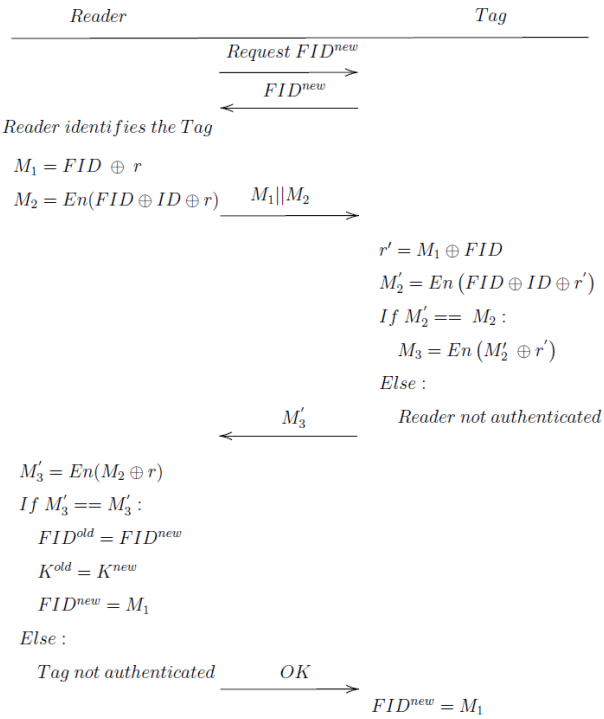

Figure 1. The authentication process of LRSAS

The authors claim that the LRSAS protocol achieves forward security, being protected against impersonation attack, track attack, desyncronization attack and denial of service.

IV. VULNERABILITIES OF LRSAS

In [14] the authors proved that not including a random piece specific to the tag when sending the tag identifier (in our case FID^{new}) to the reader compromises the privacy of the RFID scheme.

Assume that we can determine in polynomial time if two tags contain or not the same tag identifier for the LRSAS RFID scheme.

Consider a narrow weak adversary $\mathcal{A}$ that plays the following privacy game with the LRSAS protocol denoted Σ:

$$\underline{RFID_{\mathcal{A},\Sigma}^{prv\_0}(\lambda)}$$

1) $CreateTag^1(ID_1)$
2) $CreateTag^1(ID_2)$
3) $(vtag_1, 1) \leftarrow DrawTag(uniform\_distribution)$
4) $\pi_1 \leftarrow Launch()$
5) $request \leftarrow SendReader(\emptyset, \pi_1)$
6) $FID_1^{new} \leftarrow SendTag(request, vtag_1)$
7) $Free(vtag_1)$
8) $(vtag_2, 1) \leftarrow DrawTag(uniform\_distribution)$
9) $\pi_2 \leftarrow Launch()$
10) $request' \leftarrow SendReader(\emptyset, \pi_2)$
11) $FID_2^{new} \leftarrow SendTag(request, vtag_2)$
12) $Free(vtag_2)$
13) $\mathcal{A}\ receives\ the\ association\ table\ \Gamma\ of\ DrawTag$
14) $If\ (\Gamma(vtag_1) = \Gamma(vtag_2) \wedge TI(vtag\_1) = TI(vtag\_2))$
$or\big(\Gamma(vtag_1) \neq \Gamma(vtag_2) \wedge TI(vtag_1) \neq TI(vtag_2)\big)$
$then\ return\ 0\ else\ return\ 1,$

where *TI(vtag)* represents the *vtag* tag's identifier determined in the present tag's state.

The output of $RFID_{\mathcal{A},\Sigma}^{prv\_0}(\lambda)$ is 0 with overwhelming probability (respectively, 1 with negligible probability) because, given that the adversary $\mathcal{A}$ queries $vtag_1$ and $vtag_2$ on the same request, we have 2 scenarios:

1) If $vtag_1$ and $vtag_2$ refer to the same tag then FID_1^{new} and FID_2^{new} contain the same tag identifier with overwhelming probability;

2) If $vtag_1$ and $vtag_2$ do not refer to the same tag then FID_1^{new} and FID_2^{new} do not contain the same tag id with overwhelming probability.

Now we consider a blinder $\mathcal{B}$ arbitrarily chosen and the blinded privacy game $RFID_{\mathcal{A},\Sigma,B}^{prv\_1}(\lambda)$ described as $RFID_{\mathcal{A},\Sigma}^{prv\_0}(\lambda)$ but the *Launch*, *SendReader and SendTag* oracles responses are reproduced by $\mathcal{B}$. Remark that $\Gamma(vtag_1) = \Gamma(vtag_2)$ and $\Gamma(vtag\_1) \neq \Gamma(vtag\_2)$ occur with probability ½.

Next, considering that the blinder $\mathcal{B}$ can output an accurate response to $vtag_1$, it has only ½ chances to give an accurate response $vtag_2$.

Thus, because there is a significant difference between the probability of an accurate response given by the real privacy game, respectively by the blinded privacy game, the adversary $\mathcal{A}$ can determine what game it was playing.

Considering that $\mathcal{A}$ is the most limited type of adversary (narrow weak), we can deduce that the LRSAS protocol does not achieve any type of privacy in Vaudenay's model.

V. IMPROVEMENTS FOR LRSAS

When a protocol presents vulnerabilities, an analysis over the causes may help to determine if we can improve the protocol in order to increase or re-establish the privacy class that it originally claimed to have.

As shown in the previous section, the privacy of the LRSAS protocol is compromised due to non-including a random piece specific to the tag when sending the tag identifier, this step being the first one in the reader-tag communication. Also, this is the only step used for tag identification.

Although some weaknesses can be fixed in faulty protocols, the ones that compromise privacy are difficult to correct, considering that, in our case, LRAS does not achieve any form of privacy.

Because LRSAS doesn't have an omission/oversight/ negligence error, but a design flaw, we cannot improve the protocol. If one would try to increase the privacy of the scheme, it would result in developing another protocol due to LRSAS construction logic.

VI. CONCLUSIONS

With the RFID technology continuously expanding in many fields of our day-to-day life, we need to be sure that our data is protected and existing RFID schemes used for private information have a certain amount of security and privacy.

Some designers of RFID schemes build their privacy analysis without considering well-established models, misclassifying their protocols into another class of security or privacy.

By revealing the vulnerabilities that a protocol has we bring contribution to the scientific community and highlight a design flaw that we should avoid.

In this paper we recalled the concepts that are needed to understand the security and privacy of RFID shemes, then we presented the LRSAS protocol and the cryptosystem that it's using for encrypting the transmitted

data and proved that this protocol doesn't achieve any kind of privacy in Vaudenay's model due to a design error that it cannot be fixed.

VII. Acknowledgment

I want to thank Prof. dr. Ferucio Laurențiu Țiplea from the "Alexandru Ioan Cuza" University of Iași for pointing out the LRSAS vulnerability, for his suggestion to write this paper and for the feedback he gave me during its writing.

References

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

[2] L. Xiao, H. Xu, F. Zhu, R. Wang, and P. Li, "Skinny-based RFID lightweight autentification protocol," vol. 20, no. 5, 2020.

[3] K. Finkenzeller, "RFID *Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*", 3rd ed. Wiley Publishing, 2010.

[4] Y. Li, H. R. Deng, and E. Bertino, "*RFID Security and Privacy*," ser. Synthesis Lectures on Information Security, Privacy, and Trust. Morgan & Claypool Publishers, 2013.

[5] R.-I Paise and S. Vaudenay, "Mutual authentication in RFID: Security and privacy," in *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security* , ser. ASIACCS '08. New York, NY, USA: ACM, 2008, pp.292-299.

[6] A. Juels and S.A. Weis, "Defining strong privacy for RFID," *ACM Trans. Inf. Syst. Secur*., vol. 13, no.1, pp. 7:1-7:23, Nov. 2009.

[7] S. Canard, I. Coisel, J. Etrog, and M. Girault, "Privacy-preventing RFID systems: Model and constructions," https://eprint.iacr.org/2010/405.pdf, 2010.

[8] R. H. Deng, Y. Li. Yung, and Y. Zhao, "A new framework for RFID privacy," in *Proceedings of the 15th Euorpean Conference on Research in Computer Security*, ser. ESORICS'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 1-18.

[9] J.-M. Bohli and A. Pashalidis, "Relations among privacy notions," *ACM Trans, Inf. Syst. Secur*., vol. 14, no.1, pp. 4:1-4:24, Jun. 2011.

[10] 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 Berlin Heidelberg, 2011, pp. 568-587.

[11] J. Hermans, R. Peeters, and B. Preneel, "Proper RFID privacy: Model and protocols," *IEEE Transactions on Mobile Computing*, vol. no. 12, pp. 2888-2902, Dec. 2014.

[12] C. Hristea and F. L. Țiplea, "Destructive privacy and mutual authentication in Vaudenay's RFID model," Cryptology ePrint Archive, Report 2019/073, 2019, https://eprint.iacr.org/2019/073.

[13] C. Beierle, J. Jean, S. Kölbl , G. Leander, A. Morandi, T. Peyrin, Y. Sasaki, P. Sasdrich, and S. M. Slim, "The skinny family of block ciphers and its low-latency variant mantis," 08 2016, pp. 123-153.

[14] F. L. Țiplea, "Lessons to be learned for a good design of private RFID schemes," *IEEE Transactions on Dependable and Secure Computing*, pp. 1-1, 2021.