

Weak, Weak-Insider, and Randomized Weak Privacy in the HPVP Model for RFID

Ferucio Laurențiu Țiplea^a

Faculty of Computer Science, “Alexandru Ioan Cuza” University of Iași, Romania

Keywords: RFID Protocol, Privacy, Indistinguishability.

Abstract: RFID schemes that provide weak privacy or similar privacy forms are useful in any domain where the adversary cannot mount a corruption attack. In addition, these schemes can be constructed using only symmetric cryptography and can provide time-efficient identification. This paper focuses on RFID schemes that provide weak privacy in the Hermans-Pashalidis-Vercauteren-Preneel (HPVP) model based on tag indistinguishability. We first show that no adversary can have a non-negligible advantage in distinguishing between keys of a pseudo-random function. We then use this result to highlight RFID schemes that provide weak, weak-insider, and randomized weak privacy in the model above.

1 INTRODUCTION

Applying *radio frequency identification* (RFID) technology requires ensuring security and privacy properties appropriate to the scope. Security in this context means unilateral or mutual authentication, while privacy can encompass a wide range of properties such as unlinkability, untraceability, anonymity, etc. The environment in which the RFID technology is deployed plays an important role because it dictates the type of adversary that must be considered when studying the security and privacy properties. For example, an RFID scheme implemented in a hypermarket does not face a strong adversary. In contrast, an RFID scheme for identifying and authenticating personal identity documents may be subject to attacks by very strong adversaries. As a result, the levels of privacy provided by an RFID scheme must be ranked according to the adversary's power.


In this context, the RFID security and privacy model proposed in (Hermans et al., 2011; Hermans et al., 2014), inspired by Vaudenay's model and hereinafter referred to as the *HPVP model*, proposes the hierarchy of privacy levels in Figure 1. However, while Vaudenay's model studies privacy by referring to a simulator (blinder), the HPVP model proposes a study based on the indistinguishability between tags. The method is appealing and in line with the study of encryption schemes by indistinguishability.

The diagram in Figure 1 shows that weak privacy is among the lowest privacy levels in the HPVP hierarchy. The difference between this privacy level and those above it is that it cannot be used in an environment where adversaries can corrupt tags. However, there are many other practical situations in which this level of privacy is sufficient, such as:

1. The RFID scheme is implemented in a populated and video-monitored environment where the adversary cannot mount a corruption attack;
2. The adversary needs specialized equipment to mount a corruption attack, and the environment that implements the RFID scheme does not allow intervention with such equipment;
3. The RFID tags identify disposable or low-value objects that do not justify a costly corruption attack.

RFID schemes dedicated to achieving weak privacy can have significant benefits compared to schemes that achieve more than weak privacy. First, RFID schemes for weak privacy generally use symmetric cryptography, while schemes for higher levels of privacy need public-key cryptography. Second, RFID schemes based on symmetric cryptography may provide an efficient identification time (logarithmic and not linear).

Unfortunately, to our knowledge, studies on RFID schemes dedicated to weak privacy are lacking. Even the papers introducing the HPVP model focus only

^a  <https://orcid.org/0000-0001-6143-3641>

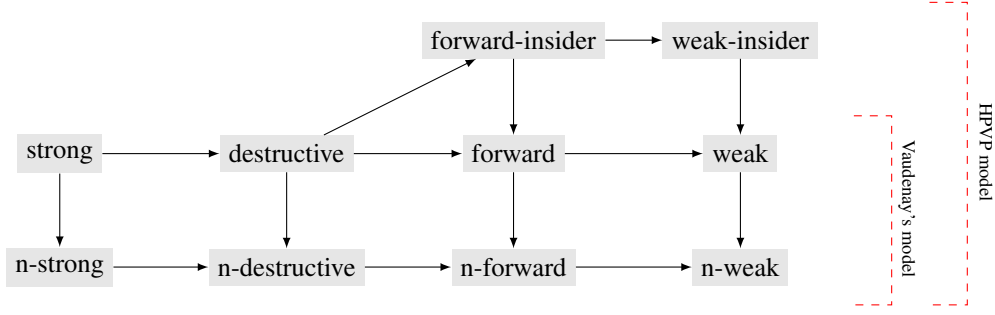


Figure 1: Privacy levels in Vaudenay's and the HPVP model: "n-p" means "narrow p" and an arrow means "implication".

on schemes offering higher privacy levels, based on public-key cryptography.

Contribution. In the context described above, our paper focuses on the weak and weak-insider privacy levels in the HPVP model, and introduces two new levels, weak-winsider and randomized weak privacy. Specifically:

- We introduce the problem of distinguishing keys of a pseudo-random function (PRF) and show that no adversary has more than a negligible advantage over it;
- We present a PRF-based RFID scheme that is weak private in the HPVP model. The proof heavily relies on the PRF key indistinguishability (discussed previously);
- We discuss weak-insider privacy for RFID schemes based on symmetric cryptography and show that this is equivalent to a weaker form of privacy, called weak-winsider privacy;
- We introduce randomized weak privacy and present a PRF-based RFID scheme that achieves this type of privacy. The proof is also based on the PRF key indistinguishability.

Our study highlights that the PRF key indistinguishability is a basic tool for studying privacy of PRF-based RFID schemes in the HPVP model.

Paper Structure. The paper is structured into five sections, the first one being the introduction. In the second section we recall basic concepts on RFID as well as the HPVP security and privacy model. The third section discusses the PRF key indistinguishability, as a central tool for studying privacy of PRF-based RFID schemes in models based on indistinguishability. The fourth section presents a PRF-based RFID scheme that achieves weak privacy in the HPVP model, shows that weak-insider privacy of RFID schemes based on symmetric cryptography is equivalent to a weaker form of weak privacy, namely

weak-winsider privacy. This result allows us to prove that the previous RFID scheme is even weak-insider private. Then, we introduce randomized weak privacy and present an RFID scheme that achieves it. The last section concludes the paper.

2 RFID SCHEMES AND SYSTEMS

We begin by recalling and fixing some standard cryptography concepts and notations (for details, the reader is referred to (Katz and Lindell, 2020)).

We use in our exposition *probabilistic polynomial time* (PPT) algorithms \mathcal{A} as defined in (Sipser, 2012) that can consult *oracles*. To specify that the algorithm \mathcal{A} can consult the oracles O_1, \dots, O_n we will write $\mathcal{A}^{O_1, \dots, O_n}$. For a set A , $a \leftarrow A$ means that a is uniformly at random chosen from A . If \mathcal{A} is a probabilistic algorithm, then $a \leftarrow \mathcal{A}$ means that a is an output of \mathcal{A} for some given input. The asymptotic approach to security makes use of security parameters, denoted by λ in our paper. A positive function $f(\lambda)$ is called *negligible* if for any positive polynomial $\text{poly}(\lambda)$ there exists n_0 such that $f(\lambda) < 1/\text{poly}(\lambda)$, for any $\lambda \geq n_0$. $f(\lambda)$ is called *overwhelming* if $1 - f(\lambda)$ is negligible.

RFID Scheme. An RFID scheme is typically composed of three main entities: a reader, a set of tags, and a radio frequency communication protocol between reader and tags. The reader is a powerful device not computationally restricted so it can perform any cryptographic operation. It stores tag related information in a database to which it has secure access. On the other side, tags are small devices that are considered to be resource constrained.

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/volatile variables* used to carry out the calculations required by the communication protocol.

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 $(\mathcal{R}, \mathcal{T})$ (Vaudenay, 2007; Paise and Vaudenay, 2008) is a triple of PPT algorithms $\Sigma = (\text{SetupR}, \text{SetupT}, \text{Ident})$, where:

1. $\text{SetupR}(\lambda)$ sets up the reader. It 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. $\text{SetupT}(pk, ID)$ initializes the tag identified by ID . It outputs an initial tag state S and a tag-specific secret K . The identity ID together with K is stored as a pair (ID, K) in the reader's database;
3. $\text{Ident}(pk; \mathcal{R}(sk, DB); ID(S))$ 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 \perp . The tag may end with no output (*unilateral authentication*), or it may end with an output consisting of OK or \perp (*mutual authentication*).

The *correctness* of an RFID scheme refers to the honest behavior of the reader and tag in a complete protocol session. That is, regardless of how the system is set up, after each complete and honest execution of the interactive protocol one of the two cases holds with overwhelming probability:

- If the tag is legitimate, the reader outputs tag's identity (and the tag outputs OK , in case of mutual authentication);
- If the tag is illegitimate, the reader outputs \perp .

The communication protocol is an alternating sequence of reader-to-tag and tag-to-reader communication steps in which the first step can be taken by either of them. When the reader sends a message m to the tag, we will often say that the reader queries the tag on m . When the first protocol step is taken by the tag, we will say that the tag answers to the empty query (this corresponds to the tag being powered by the reader).

An instantiation of an RFID scheme is sometimes called an *RFID system*. It is usually performed by a trusted operator who establishes a reader identifier \mathcal{R} , a set \mathcal{T} of tag identifiers, and run the RFID scheme over $(\mathcal{R}, \mathcal{T})$.

RFID Security and Privacy Model An RFID security and privacy model establishes the type of adversary to consider, how it interacts with the RFID scheme, and how the security and privacy properties of RFID schemes are defined. In this paper,

we will adopt the Hermans-Pashalidis-Verauteren-Preneel (HPVP) model (Hermans et al., 2011; Hermans et al., 2014), which is based on tag indistinguishability.

The adversary is a PPT algorithm. It interacts with the RFID scheme through a set of oracles that he can query, namely: $\text{CreateTag}()$, $\text{Launch}()$, $\text{DrawTag}()$, $\text{Free}()$, $\text{SendTag}()$, $\text{SendReader}()$, $\text{Result}()$, $\text{Corrupt}()$, $\text{CreateInsider}()$. The original approach allows for the creation of multiple readers, but for the discussion in this paper, it is sufficient to consider only a single reader \mathcal{R} . The adversary's interaction with the oracles is described below:

1. $\mathcal{T} \leftarrow \text{CreateTag}(ID)$: On input a tag identifier ID , the oracle calls $\text{SetupT}(pk, ID)$ to generate a pair (S, K) . It adds (ID, K) to DB and the tag is considered *registered* with the reader (and so it is a legitimate tag). Moreover, a distinct reference \mathcal{T} to the tag is returned. The oracle does not reject duplicate ID s;
2. $\pi \leftarrow \text{Launch}()$: Launches a new protocol instance (session), assigns a unique identifier π to it, and outputs it;
3. $vtag/\perp \leftarrow \text{DrawTag}(\mathcal{T}_0, \mathcal{T}_1)$: On input a pair of tag references \mathcal{T}_0 and \mathcal{T}_1 , the oracle outputs a virtual tag reference $vtag$ as a monotonic counter or \perp . The virtual tag reference $vtag$, when outputted, refers to either \mathcal{T}_0 or \mathcal{T}_1 , depending on the left or right indistinguishability game, respectively, where the oracle is used. The oracle maintains a table Γ of triples $(vtag', \mathcal{T}_0', \mathcal{T}_1')$. All virtual tag references point to the left or the right reference tag in such a table.

The output of this oracle must meet the following requirements:

- If one of the tags \mathcal{T}_0 or \mathcal{T}_1 is in the list of insider tags, then the output is \perp ;
- If the virtual tag references in Γ refer to the left tag reference and $(vtag', \mathcal{T}_0', \mathcal{T}_1') \in \Gamma$ for some $vtag'$ and \mathcal{T}_1' , then the output is \perp ;
- If the virtual tag references in Γ refer to the right tag reference and $(vtag', \mathcal{T}_0', \mathcal{T}_1') \in \Gamma$ for some $vtag'$ and \mathcal{T}_0' , then the output is \perp ;
- In all the other cases the output is $vtag$.

The triple $(vtag, \mathcal{T}_0, \mathcal{T}_1)$, when the output is $vtag$, is added to Γ ;

4. $\text{Free}(vtag)$: This oracle retrieves the unique triple $(vtag, \mathcal{T}_0, \mathcal{T}_1)$ from Γ , resets the tag that $vtag$ refers to, and removes $(vtag, \mathcal{T}_0, \mathcal{T}_1)$ from Γ (this means that the tags \mathcal{T}_0 and \mathcal{T}_1 are freed). Moreover, the identifier $vtag$ will no longer be used.

When a tag is reset, its temporary memory is erased, while the permanent one is preserved;

5. $m'/\perp \leftarrow \text{SendTag}(m, vtag)$: Outputs the tag's answer m' to the query m sent to the tag referred to by $vtag$, and \perp if $vtag$ is not in any triple in Γ (we draw attention that $vtag$ when in Γ , refers to one of two tags depending on the privacy game). We will consider m' as the empty message, abusively but suggestively denoted by \emptyset , to specify that the tag outputs nothing, thus marking the last step of communication;
6. $m'/\perp \leftarrow \text{SendReader}(m, \pi)$: Outputs the reader's answer m' to the query m in the protocol instance π , if π is an active instance, and \perp otherwise (π is not active or does not even exist). When m is the empty message, this oracle outputs the first message of the protocol instance π , assuming that the reader takes the first step in the protocol. When m' is the empty message, we understand that the reader does not output anything (this marks the last communication step of the protocol);
7. $1/0/\perp \leftarrow \text{Result}(\pi)$: Outputs \perp if in session π the reader has not yet made a decision on tag authentication (this also includes the case when the session π does not exist), 1 if in session π the reader authenticated the tag, and 0 otherwise. This oracle is both for unilateral and mutual authentication;
8. $S \leftarrow \text{Corrupt}(\mathcal{T})$: Outputs the current permanent and temporary state S of the tag \mathcal{T} , when the tag is not involved in any computation of any protocol step (that is, the permanent state before or after a protocol step);
9. $\mathcal{T}, S \leftarrow \text{CreateInsider}(ID)$: This oracle creates a new tag referenced by \mathcal{T} and registers it with the reader. Then, it outputs \mathcal{T} and the tag's full internal state S . The tag \mathcal{T} is then included in a list I of so-called *insider tags* and destroyed.

Depending on the oracles they can query, adversaries are classified into:

- *Weak-insider adversaries*: they do not have access to the *Corrupt* oracle;
- *Weak adversaries*: these are weak-insider adversaries that do not have access to the *CreateInsider* oracle;
- *Forward-insider adversaries*: if they access the *Corrupt* oracle, then they can only access the *Corrupt* oracle;
- *Forward adversaries*: these are forward-insider adversaries that do not have access to the *CreateInsider* oracle;

- *Destructive adversaries*: after the adversary has queried *Corrupt*(\mathcal{T}) and obtained the corresponding information, the tag referred by \mathcal{T} is destroyed. The database *DB* will still keep the record associated to this tag (the reader does not know that the tag was destroyed);
- *Strong adversaries*: no restrictions on the use of oracles.

Some authors (Hermans et al., 2014) refer to these classes of adversaries as being *wide* in the sense that the adversaries in these classes may consult the *Result* oracle. When an adversary is not allowed to consult the *Result* oracle, we will refer it as being *narrow*. The narrow property can be combined with any of the properties strong, destructive, forward, and weak in order to get another four classes of adversaries, *narrow weak/forward/destructive/strong*.

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

To define privacy in the HPVP model, we consider the probability experiment (privacy game) $\text{PRIV}_{\mathcal{A}, \Sigma}^b(\lambda)$ described in the table in Figure 2, where \mathcal{A} is an adversary, \mathcal{O} is the set of oracles \mathcal{A} is allowed to query, Σ is an RFID scheme, and the bit $b \in \{0, 1\}$ shows when \mathcal{A} interacts with the left-hand side tag ($b = 0$) or with the right-hand side tag ($b = 1$) of the pairs of tags drawn by him.

	$\text{PRIV}_{\mathcal{A}, \Sigma}^b(\lambda)$
1	$(pk, sk, DB) \leftarrow \text{SetupReader}(\lambda)$
2	$b' \leftarrow \mathcal{A}^{\mathcal{O}}(pk, \lambda);$
3	Return b'

Figure 2: Privacy game in the HPVP model.

An RFID scheme Σ achieves V privacy, where V is a class of adversaries, if $\text{PRIV}_{\mathcal{A}, \Sigma}(\lambda)$ is negligible, for any adversary $\mathcal{A} \in V$, where:

$$\text{PRIV}_{\mathcal{A}, \Sigma}(\lambda) = |P(\text{PRIV}_{\mathcal{A}, \Sigma}^0(\lambda) = 1) - P(\text{PRIV}_{\mathcal{A}, \Sigma}^1(\lambda) = 1)|$$

According to the type of adversary considered (class V above), the privacy classes (levels) in Figure 1 are obtained.

3 PRF KEY INDISTINGUISHABILITY

This section will present a specific property of pseudo-random functions, crucial in studying RFID schemes' privacy properties in models based on indistinguishability.

Let \mathcal{K} be an at most countable set whose elements we call *keys*. Suppose that we have defined a measure (length) of keys on this set, $|\cdot|$, which takes values in the set \mathbb{N} of positive integers. We will refer to $|K|$ as the length of the key $K \in \mathcal{K}$. For example, \mathcal{K} may be a set of binary strings, and $|\cdot|$, their length.

Given a positive integer λ , we denote $\mathcal{K}_\lambda = \{K \in \mathcal{K} \mid |K| = \lambda\}$. A \mathcal{K} -indexed family of functions is a function F that associates to each key $K \in \mathcal{K}$ a function $F_K : D_\lambda \rightarrow R_\lambda$, where $\lambda = |K|$, and D_λ and R_λ are finite sets (the same for all keys of length λ). A convenient way to define D_λ and R_λ is to use two polynomials ℓ_1 and ℓ_2 specific to the family F , and then choosing $D_\lambda = \{0,1\}^{\ell_1(\lambda)}$ and $R_\lambda = \{0,1\}^{\ell_2(\lambda)}$. We usually denote the family F by $F = (F_K)_{K \in \mathcal{K}}$ or $F = (F_\lambda)_{\lambda \in \mathbb{N}}$, where $F_\lambda = \{F_K \mid |K| = \lambda\}$. The difference in notation will be clear from the context.

We say that F is a *pseudo-random function (PRF)* if the following two properties hold:

1. F is *efficiently computable*: there is a deterministic algorithm of polynomial time complexity that can compute every function F_K defined by F ;
2. F is *indistinguishable from truly random functions*: for each PPT algorithm \mathcal{A} , its PRF -advantage against F

$$PRF_{\mathcal{A},F}(\lambda) = |P(1 \leftarrow \mathcal{A}^{F_K}(\lambda) \mid K \leftarrow \mathcal{K}_\lambda) - P(1 \leftarrow \mathcal{A}^f(\lambda) \mid f \leftarrow (D_\lambda, R_\lambda))|,$$

is negligible as a function of λ .

The two probabilities in the definition above are often associated with the probability experiments (also called *security games*) $PRF_{\mathcal{A},F}^0(\lambda)$ and $PRF_{\mathcal{A},F}^1(\lambda)$ from the table in Figure 3.

	$PRF_{\mathcal{A},F}^0(\lambda)$	$PRF_{\mathcal{A},F}^1(\lambda)$
1	$K \leftarrow \mathcal{K}_\lambda$	$f \leftarrow (D_\lambda, R_\lambda)$
2	$b \leftarrow \mathcal{A}^{F_K}(\lambda)$	$b \leftarrow \mathcal{A}^f(\lambda)$
3	Return b	Return b

Figure 3: PRF security games.

As can be seen, the function $PRF_{\mathcal{A},F}(\lambda)$ calculates the difference between the probabilities with which the adversary outputs the same bit (for example, 1) in

both probability experiments. We can write equivalently

$$PRF_{\mathcal{A},F}(\lambda) = |P(PRF_{\mathcal{A},F}^0(\lambda) = 1) - P(PRF_{\mathcal{A},F}^1(\lambda) = 1)|$$

In the above security games, the adversary queries a single function. One can show by using a hybrid argument (Mittelbach and Fischlin, 2021; Boneh and Shoup, 2023) or directly that allowing the adversary to query a polynomial number of functions in the two probability experiments does not change the concept of PRF function. In this case, the security games are shown in the table in Figure 4 (ℓ is a parameter of polynomial size with respect to λ that depends on \mathcal{A}).

	$PRF_{\mathcal{A},F}^{*,0}(\lambda)$	$PRF_{\mathcal{A},F}^{*,1}(\lambda)$
1	$\ell(\lambda) \leftarrow \mathcal{A}(\lambda)$	$\ell(\lambda) \leftarrow \mathcal{A}(\lambda)$
2	$K_1, \dots, K_\ell \leftarrow \mathcal{K}_\lambda$	$f_1, \dots, f_\ell \leftarrow (D_\lambda, R_\lambda)$
3	$b \leftarrow \mathcal{A}^{F_{K_1}, \dots, F_{K_\ell}}(\lambda)$	$b \leftarrow \mathcal{A}^{f_1, \dots, f_\ell}(\lambda)$
4	Return b	Return b

Figure 4: PRF^* security games.

Then, F is a PRF if and only if the PRF^* -advantage of \mathcal{A} against F ,

$$PRF_{\mathcal{A},F}^*(\lambda) = |P(PRF_{\mathcal{A},F}^{*,0}(\lambda) = 1) - P(PRF_{\mathcal{A},F}^{*,1}(\lambda) = 1)|,$$

is negligible.

A specific property of PRFs is that no adversary can distinguish between two keys of a PRF based on (having) a value computed with one of them, except with negligible probability. To define this property precisely, we consider the probability experiments in the table in Figure 5, where F is a \mathcal{K} -indexed family of functions, \mathcal{A} is a PPT algorithm, and ℓ is a parameter of polynomial size in λ that depends on \mathcal{A} .

We define now the *key distinguishing advantage (KD-advantage)* of \mathcal{A} against F by

$$KD_{\mathcal{A},F}(\lambda) = |P(KD_{\mathcal{A},F}^0(\lambda) = 1) - P(KD_{\mathcal{A},F}^1(\lambda) = 1)|.$$

Theorem 3.1. *Let F be a \mathcal{K} -indexed family of functions. If F is a PRF , then the KD -advantage of any adversary \mathcal{A} against F is negligible.*

Proof. Let F be a PRF and \mathcal{A} an adversary that requests $\ell = \ell(\lambda)$ key pairs to query in the KD security games against F . We will shown that $KD_{\mathcal{A},F}(\lambda)$ is a negligible function. The triangle inequality for absolute values leads to:

	$KD_{\mathcal{A},F}^0(\lambda)$	$KD_{\mathcal{A},F}^1(\lambda)$
1	$\ell(\lambda) \leftarrow \mathcal{A}(\lambda)$	$\ell(\lambda) \leftarrow \mathcal{A}(\lambda)$
2	$(K_0^1, K_1^1), \dots, (K_0^\ell, K_1^\ell) \leftarrow \mathcal{K}_\lambda^2$	$(K_0^1, K_1^1), \dots, (K_0^\ell, K_1^\ell) \leftarrow \mathcal{K}_\lambda^2$
3	$b \leftarrow \mathcal{A}^{F_{K_0^1}, \dots, F_{K_0^\ell}}(\lambda)$	$b \leftarrow \mathcal{A}^{F_{K_1^1}, \dots, F_{K_1^\ell}}(\lambda)$
4	Return b	Return b

Figure 5: KD security games.

$$\begin{aligned}
& KD_{\mathcal{A},F}(\lambda) = \\
& = |P(1 \leftarrow \mathcal{A}^{F_{K_0^1}, \dots, F_{K_0^\ell}}(\lambda) | (K_0^i, K_1^i) \leftarrow_{i=1}^\ell \mathcal{K}_\lambda^2) - \\
& \quad P(1 \leftarrow \mathcal{A}^{F_{K_1^1}, \dots, F_{K_1^\ell}}(\lambda) | (K_0^i, K_1^i) \leftarrow_{i=1}^\ell \mathcal{K}_\lambda^2)| \\
& \quad |P(1 \leftarrow \mathcal{A}^{F_{K_0^1}, \dots, F_{K_0^\ell}}(\lambda) | K_0^1, \dots, K_0^\ell \leftarrow \mathcal{K}_\lambda) - \\
& \quad P(1 \leftarrow \mathcal{A}^{F_{K_1^1}, \dots, F_{K_1^\ell}}(\lambda) | K_1^1, \dots, K_1^\ell \leftarrow \mathcal{K}_\lambda)| \\
& \leq |P(1 \leftarrow \mathcal{A}^{F_{K_0^1}, \dots, F_{K_0^\ell}}(\lambda) | K_0^1, \dots, K_0^\ell \leftarrow \mathcal{K}_\lambda) - \\
& \quad P(1 \leftarrow \mathcal{A}^{f_1, \dots, f_\ell}(\lambda) | f_1, \dots, f_\ell \leftarrow (D_\lambda, R_\lambda))| + \\
& \quad |P(1 \leftarrow \mathcal{A}^{f_1, \dots, f_\ell}(\lambda) | f_1, \dots, f_\ell \leftarrow (D_\lambda, R_\lambda)) - \\
& \quad P(1 \leftarrow \mathcal{A}^{F_{K_1^1}, \dots, F_{K_1^\ell}}(\lambda) | K_1^1, \dots, K_1^\ell \leftarrow \mathcal{K}_\lambda)|
\end{aligned}$$

Since F is a PRF , both absolute values in the last term of the above inequality are negligible, which shows that $KD_{\mathcal{A},F}(\lambda)$ is negligible. \square

In the key distinguishability security games for a family of functions F (Figure 5), the challenger randomly chooses several key pairs, and the adversary interrogates (the functions indexed by) the first or second component of these pairs (depending on the security game he is playing). If F is a PRF , the two sequences of values obtained by the adversary through interrogation are indistinguishable (Theorem 3.1).

We now consider a stronger security game for distinguishing keys of a family of functions F . In this game, the challenger randomly generates several keys, but the adversary is the one who pairs them as he wishes to interrogate them. The significant difference from the previous game is that, by pairing the keys, the adversary can create patterns by which he can distinguish between the two sequences of values he obtains through interrogation (as we said above). Since the keys are randomly generated and unknown to the adversary, the pattern that the adversary can create is based on repeating some keys on the first or second component of the pairs so that, upon querying, one of the two sequences contains repetitions of the same value.

For example, suppose the adversary requests to query (K_0, K_1) and (K_0, K_2) on the same input x and obtains the same value y . In that case, he knows with overwhelming probability that he is querying the first component of the key pairs (if the values obtained by

querying are distinct, then he knows with overwhelming probability that he is querying the second component of the key pairs). As a result, the adversary can distinguish between the keys of the function family F .

If the adversary requests to query (K_0, K_1) and then again (K_0, K_1) on the same input x , he will not be able to draw any clear conclusion about the component that was evaluated unless he knows, for instance, $F_{K_0}(x)$ (the adversary can obtain this value by querying the pair (K_0, K_0)).

As a result, the pairs that leak information to the adversary are of the form (K_0, K_1) and (K'_0, K'_1) , where $K_0 = K'_0$ and $K_1 \neq K'_1$, or $K_0 \neq K'_0$ and $K_1 = K'_1$. The pair (K_0, K_0) leaks information to the adversary only if it is accompanied by another pair that has K_0 on the first or second component, while the other component is different from K_0 ; this case actually falls into the first case mentioned above.

The discussion above leads us to consider a stronger probability experiment for key distinguishability in which the adversary can pair keys as he wants. We denote this game by $SKD_{\mathcal{A},F}$ and define it as in the table in Figure 6.

Similarly, using the second component of the key pairs, the security game $SKD_{\mathcal{A},F}^1(\lambda)$ is defined. We define now the *strong key distinguishing advantage* (SKD -advantage) of \mathcal{A} against F by

$$\begin{aligned}
SKD_{\mathcal{A},F}(\lambda) &= |P(SKD_{\mathcal{A},F}^0(\lambda) = 1) \\
&\quad - P(SKD_{\mathcal{A},F}^1(\lambda) = 1)|.
\end{aligned}$$

Theorem 3.2. *Let F be a \mathcal{K} -indexed family of functions. If F is a PRF , then the SKD -advantage of any adversary \mathcal{A} against F is negligible.*

Proof. Let F be a PRF and \mathcal{A} an adversary. According to Theorem 3.1, there exists a negligible function $\mu(\lambda)$ such that $KD_{\mathcal{A},F}(\lambda) \leq \mu(\lambda)$.

By a sequence of security games we will show that $SKD_{\mathcal{A},F}(\lambda)$ is a negligible function. For readability, we denote by $G_0(\lambda)$ the security game $SKD_{\mathcal{A},F}^0(\lambda)$ and assume the adversary requests $\ell(\lambda)$ queries (see the security game in the table from Figure 6). For $0 < t \leq \ell(\lambda)$, we recursively define the security game $G_t(\lambda)$ as obtained by modifying the security game $G_{t-1}(\lambda)$ as follows:

	$SKD_{\mathcal{A},F}^0(\lambda)$
1	\mathcal{A} asks the challenger to randomly generate p keys, where p is a polynomial parameter in λ ;
2	The challenger draws uniformly at random and independent of each other p keys K_1, \dots, K_p , which he will identify by their index;
3	\mathcal{A} repeats the following procedure ℓ times, where ℓ is a polynomial parameter in λ :
3.1	\mathcal{A} adaptively chooses a pair of indices (i, j) and an input x not previously used before with different pairs of indices;
3.2	\mathcal{A} sends (i, j) and x to the challenger and receives back $F_{K_i}(x)$;
4	\mathcal{A} outputs a bit b ;
5	Return b .

 Figure 6: SKD^0 security game.

- at the t -th query made by \mathcal{A} , the answer given by the challenger contains the evaluation of the function defined by the second component of the key pair and not by the first. The other queries remain unchanged (as they are in the game $G_{t-1}(\lambda)$). Clearly, the requirement from (3.1) in Figure 6 remains fulfilled.

According to Theorem 3.1, for any $0 < t \leq \ell(\lambda)$,

$$|P(G_t(\lambda) = 1) - P(G_{t-1}(\lambda) = 1)| \leq \mu(\lambda),$$

Moreover, $G_{\ell(\lambda)}(\lambda) = SKD_{\mathcal{A},F}^1(\lambda)$.

The triangle inequality for absolute values leads then to:

$$\begin{aligned} SKD_{\mathcal{A},F}(\lambda) &= \\ &= |P(SK D_{\mathcal{A},F}^0(\lambda) = 1) - P(SK D_{\mathcal{A},F}^1(\lambda) = 1)| \\ &= |P(G_t(\lambda) = 1) - P(G_{\ell(\lambda)}(\lambda) = 1)| \\ &\leq \sum_{t=1}^{\ell(\lambda)} |P(G_t(\lambda) = 1) - P(G_{t-1}(\lambda) = 1)| \\ &\leq \ell(\lambda)\mu(\lambda). \end{aligned}$$

As ℓ is a polynomial and μ is negligible, $\ell(\lambda)\mu(\lambda)$ is a negligible function. \square

Remark 3.1. We believe that the SKD security games can further be strengthened relative to the values chosen by the adversary for which he requests the evaluation of a function (item (3.1) in the security game in Figure 6). However, for our purpose, the already chosen variant is sufficient.

4 WEAK, WEAK-INSIDER, AND RANDOMIZED WEAK PRIVACY

There are many practical situations in which the RFID system is not subject to a corruption attack, such as:

1. The RFID system is implemented in a populated and video-monitored environment;
2. The adversary needs specialized equipment to mount the corruption attack, and the environment that implements the RFID system does not allow intervention with such equipment;
3. The RFID tags identify disposable or low-value objects to justify a corruption attack.

In such situations, weak privacy is sufficient for the RFID system. Except for the fact that “weak privacy” does not ensure privacy when the adversary uses corruption, weak privacy is not really that “weak”. In addition, protocols that ensure weak privacy can only use symmetric cryptography, and in some cases can provide efficient-time identification.

In this section, we will discuss three forms of weak privacy in the HPVP model. We explicitly mention from the beginning that the literature on the HPVP model does not present any protocol of this type. Therefore, our paper is the first to present protocols and proofs of their weak privacy in the HPVP model.

To better understand what follows, we will briefly discuss the cryptographic ingredients used in constructing RFID schemes (protocols). Thus, RFID protocols can use symmetric cryptography, public key cryptography, and, more recently, physically unclon-

able functions (PUFs) for security and privacy reasons. Among these, symmetric cryptography is vital in constructing RFID protocols because it is much more efficient to implement than public key cryptography. Physically unclonable functions have received special attention recently and offer unique advantages when combined with symmetric cryptography. In this paper, we will only consider the case of symmetric cryptography, which we will assume includes:

1. *Invertible algorithms*: algorithms O for which there exists a deterministic polynomial time algorithm O^{-1} that inverts O in the sense that $x = O^{-1}(K, y)$ for all $y \leftarrow O(K, x)$;
2. *One-way algorithms*: algorithms O for which there exists a deterministic polynomial time algorithm O^v that verifies the output of O in the sense that

$$O^v(K, x, y) = \begin{cases} 1, & \text{if } y \leftarrow O(K, x) \\ 0, & \text{otherwise;} \end{cases}$$

3. *Random number generators*: algorithms G of polynomial time complexity that might depend on some secret key and are used to generate random numbers.

Symmetric-key encryption and decryption algorithms fall in the first category, while hash functions, message authentication codes (MACs), and pseudo-random functions (PRFs) fall in the second category. Pseudo-random generators (PRGs) may be thought as deterministic random number generators that depend on some secret key or seed. All the operations with integers, such as addition, multiplication, XOR and so on, can also be viewed as being performed by algorithms as those above (except that they do not depend on any secret key).

What is the maximum level of privacy an RFID protocol can achieve in the HPVP model using only symmetric cryptography? For Vaudenay's model, (Tiplea, 2022b) proved this is "weak privacy". A similar result for the HPVP model is missing, but we firmly believe that the above result also holds in this case. The papers (Hermans et al., 2011; Hermans et al., 2014) that introduce the HPVP model do not discuss the case of RFID protocols based on symmetric cryptography. We are unaware of any study dedicated to these protocols in the HPVP model. In this context, we present below RFID protocols that ensure different types of weak privacy in the HPVP model. The key tool in our study is Theorem 3.2 obtained in the previous section, which functions as a general methodology for establishing privacy properties in the HPVP model for PRF-based RFID protocols.

4.1 Weak Privacy

This section shows that the PRF-based RFID protocol in (Vaudenay, 2007), which is weak private in Vaudenay's model, is also weak private in the HPVP model. Again, we mention that, to our knowledge, there is no proof of this fact yet. One cannot adapt the proof in Vaudenay's model to the HPVP model because it relies on constructing a *blinder*. In contrast, the privacy properties in the HPVP model rely on tag indistinguishability. Moreover, tag indistinguishability cannot be directly related to the specific property of pseudo-random functions. In other words, the assumption that the RFID protocol does not provide tag indistinguishability does not directly contradict the fact that it relies on a pseudo-random function. However, as we will see, it contradicts the key indistinguishability property of PRFs.

Let us recall now the PRF-based RFID scheme from (Vaudenay, 2007). The scheme, pictorially represented in Figure 7, is based on a PRF $F = (F_K)_K$, where F_K is a function from $\{0, 1\}^{\ell_1(\lambda)}$ to $\{0, 1\}^{\ell_2(\lambda)}$ for all $K \in \mathcal{K}_\lambda$ and some polynomials $\ell_1(\lambda)$ and $\ell_2(\lambda)$. Each tag is equipped with a pair (ID, K) , where ID is the tag's identity and K is a key for F , uniformly at random chosen and independent from the other tags.

Theorem 4.1. *The PRF-based RFID scheme assures weak privacy in the HPVP model.*

Proof. (sketch) Assume that the PRF-based RFID scheme, denoted Σ , does not assure weak privacy in the HPVP model. Then, there exists a weak adversary \mathcal{A} that has a non-negligible advantage against Σ .

Define a PRF adversary \mathcal{A}' that breaks the strong key indistinguishability property of F with non-negligible probability. First, assume that \mathcal{C} is a challenger for F that initiates the $SKD_{\mathcal{A}', F}^b$ game of \mathcal{A}' against F , where $b \in \{0, 1\}$. The basic steps are as follows:

1. \mathcal{A}' simulates the scheme Σ for \mathcal{A} :
 - (a) When \mathcal{A} asks for tag creation with identity ID , \mathcal{A}' will ask \mathcal{C} to generate a key and assign the identity ID to it. A specification \mathcal{T}_{ID} is returned to \mathcal{A} ;
 - (b) When \mathcal{A} draws $(\mathcal{T}_{ID_1}, \mathcal{T}_{ID_2})$ and sends x for interrogation, \mathcal{A}' generates a random y and sends (ID_1, ID_2) , x , and y to \mathcal{C} . Then, \mathcal{C} generates at random y , computes $z = F_{ID_b}(x, y)$, and returns the result (y, z) to \mathcal{A}' . In turn, \mathcal{A}' returns (y, z) to \mathcal{A} ;
 - (c) \mathcal{A} queries $Result(\pi)$: If the queries made by \mathcal{A} within the protocol session π preserve the order of the steps in the RFID scheme (possibly

	Reader (F, DB)	Tag (ID, K)
1	$x \leftarrow \{0, 1\}^{\ell_1(\lambda)}$	\xrightarrow{x}
2		$\xleftarrow{y, z} \quad y \leftarrow \{0, 1\}^{\ell_2(\lambda)}, z = F_K(x, y)$
3	If $\exists (ID, K) \in DB$ s.t. $z = F_K(x, y)$ then output ID else output \perp	

Figure 7: PRF-based RFID scheme.

with a delay), and the messages have not been changed by \mathcal{A} , then \mathcal{A}' will return 1. If the order of the steps is preserved but the messages have been changed by \mathcal{A} , then \mathcal{A}' will return 0. In all the other cases, \mathcal{A}' will return \perp ;

2. It is easy to check that

$$|P(PRIV_{\mathcal{A}, \Sigma}^b(\lambda) = 1) - P(SKD_{\mathcal{A}', F}^b(\lambda) = 1)|$$

is negligible for any b . As a conclusion,

$$|SKD_{\mathcal{A}', F}(\lambda) - PRIV_{\mathcal{A}, \Sigma}(\lambda)|$$

is negligible.

Therefore, if $PRIV_{\mathcal{A}, \Sigma}(\lambda)$ is non-negligible, then $SKD_{\mathcal{A}', F}(\lambda)$ is non-negligible, contradicting Theorem 3.2. \square

4.2 Weak-Insider Privacy

From design to utilization in practice, an RFID system goes through several stages, such as physical manufacturing, software implementation of the protocols, deployment, and maintenance. Each party responsible for these stages can maliciously intervene in the system, inserting tags into the database that the adversary can control by knowing their internal state. Such tags are called *insider tags*, and the HPVP model takes them into consideration for studying the security and privacy of RFID systems.

The use of insider tags by adversaries usually goes along the following line (for examples, please see (van Deursen and Radomirović, 2012)):

1. The adversary queries let us say two pairs of tags, gets the responses c and c' , and wants to find out if the responses satisfy a certain property ϕ (for instance, if c and c' come from the same tag);
2. The adversary uses an insider tag, prepares a valid message m for the reader, and “adds” $\phi(c, c')$ to it getting a new message m' . The trick is for the reader to accept m' only if c and c' verify ϕ , except with negligible probability. Then, by consulting the oracle *Result*, the adversary will know whether or not the property in question is satisfied by c and c' .

The insider tag-based attack is relevant only for the forward and weak privacy levels. Destructive (and therefore strong) adversaries can simulate insider tags by creating legitimate tags and then corrupting them.

What does a weak-insider private RFID scheme look like? To our knowledge, no such scheme has been proposed in the literature. Existing RFID schemes resistant to insider tag attacks are based on public-key cryptography and ensure forward-insider privacy.

In this section, we will show that the *PRF*-based RFID scheme discussed in the previous section is indeed weak-insider private. We will start with an important observation about *SKC*-based RFID schemes. For such schemes, verifying a property between the responses of two queries can be much simpler because this type of cryptography uses deterministic algorithms. As a result, it is no longer necessary to record the insider tag in the database in such a case. From a practical point of view, this means that the RFID system manufacturer can keep a few tags for later use without them being recorded in the database.

To model this case, we introduce a new oracle, *CreateWInsider*, which creates a *weak insider* (*winsider*) tag. Its creation means launching *SetupTag(pk, ID)* to generate (S, K) and then give S to the adversary. The tag will not be created and, therefore, will not be registered. It will not be possible to draw it, so it will not be possible to communicate with it. The reader will never be able to identify it, but its state can be used by the adversary, possibly in combination with other states. As a result, a *winsider* tag is an insider tag that is not registered in the database. Replacing the *CreateInsider* oracle with *CreateWInsider* we obtain the concept of *weak-winsider adversary* and *weak-winsider privacy* (see Section 2).

To make our result as general as possible, we will assume that the RFID protocols in the theorem below have at most three communication steps:

1. The reader sends a message m_1 to the tag;
2. On receiving the message m_1 , the tag computes and sends back its answer m_2 ;
3. On receiving the message m_2 , the reader runs the

identification and/or authentication procedure and sends back to the tag its answer m_3 .

The first two steps are usually used in the case of unilateral authentication. The third step is used to allow the tag to authenticate the reader.

Theorem 4.2. *Any weak-winsider private SKC-based RFID scheme is also weak-insider private in the HPVP model.*

Proof. Assume that Σ is an weak-winsider private SKC-based RFID scheme that is not weak-insider private in the HPVP model. Let \mathcal{A} be a weak-insider adversary against Σ . We define a weak-winsider adversary \mathcal{A}' that has the same advantage against Σ as \mathcal{A} , leading thus to a contradiction (because Σ is weak private).

The main idea is the next one. The adversary \mathcal{A}' runs and monitors \mathcal{A} to play with Σ :

- If \mathcal{A} does not use any insider tag, then \mathcal{A}' does not do anything;
- If \mathcal{A} queries *CreateInsider*, \mathcal{A}' will create an insider tag for \mathcal{A} and gives him the internal state. Moreover, \mathcal{A}' will keep track of this tag (state updates and involvements in the protocol sessions);
- Insider tags cannot be drawn and so they cannot be queried by means of *SendTag*. However, their internal states can be used by \mathcal{A} to query *SendReader*(m, π), where m is a valid message built from an insider tag information and according to the messages of the protocol session π . The adversary \mathcal{A}' can check if m fulfills this property because Σ uses only SKC and the winsider tags were created by \mathcal{A}' . Therefore, in such a case, \mathcal{A}' will simulate the reader's answer for \mathcal{A} and also know how to answer queries addressed to the oracle *Result*.

In this way, \mathcal{A}' will play with Σ as \mathcal{A} does, but without using any insider tag. In other words, \mathcal{A}' is a weak-winsider adversary. Moreover, its advantage against Σ is exactly the advantage of \mathcal{A} against Σ . Hence, the assumption that \mathcal{A} has a non-negligible advantage against Σ contradicts the weak-winsider privacy of Σ . \square

Remark 4.1. *If the adversary can create himself winsider tags in an RFID scheme, then the scheme is weak private if and only if it is weak-winsider private. The PRF-based scheme in the previous section fits this case because the function F is public, and so is its key space; the adversary can create insider tags himself.*

Corollary 4.1. *The PRF-based RFID scheme in Section 4.1 is weak-insider private.*

Proof. From Theorem 4.2 and Remark 4.1. \square

4.3 Randomized Weak Privacy

Randomized weak (r-weak) privacy was introduced in (Hrístea and Tiplea, 2020) as a weaker form of weak privacy in Vaudenay's model. It helps identify entities where tracing is unimportant or cannot be done appropriately due to very high mobility or crowding in which the entity is. For details, please see (Hrístea and Tiplea, 2020).

In the HPVP model, r-weak privacy can simply be obtained by replacing the oracle *Free*($vtag$) with the oracle *r-Free*($vtag$) defined as follows:

r-Free($vtag$): the freed tag is re-randomized.

By tag re-randomization, we mean that after the tag is freed, some of the parameters that were set randomly are replaced by others, also chosen randomly, so that the tag's response to a new communication with the adversary is indistinguishable from the responses from previous communications. For example, if the tag computes $F_K(x)$, where F is a PRF and x is a random secret parameter, re-randomization can be achieved by randomly choosing a new parameter x or a new key K .

The RFID scheme in Figure 8, proposed in (Hrístea and Tiplea, 2020), is r-weak private in the Vaudenay model. We will show that it is r-weak private in the HPVP model as well. First, let us discuss the scheme a little. The tag, which takes the first step in the protocol, computes $z = F_K(0, 0, x)$, where $F = (F_K)_K$ is a pseudo-random function, and sends it to the reader. The reader checks its database for a triple (ID, K, x) such that $z = F_K(0, 0, x)$ or $z = F_K(0, 0, x + 1)$. The reason is that at most one step of desynchronization may occur between reader and tag. When the reader finds the right value, resynchronizes with the tag and prepares the answer w . The tag checks the value w received from reader, takes a decision, updates x if necessary, and prepares the answer for reader. On receiving the tag's answer, the reader checks it, takes a decision, and updates x . If the reader does not update x (because it rejects the tag), then it will do so in the second step of the next protocol session (with the same tag). Therefore, the desynchronization between reader and tag is at most one step.

It is straightforward to check the correctness of the scheme in Figure 8. We remark that it does not use temporary variables, and the tag only needs to compute F . Moreover, the scheme allows for a quite efficient search procedure in the reader's database. Two ordered sets of indices are used: the first one with indices of the form $F_K(0, 0, x)$, and the second one with

	Reader (DB, F)	Tag (K, x)
1		\xleftarrow{z} $z = F_K(0, 0, x)$
2	If $\exists((ID, K, x) \in DB, i \in \{0, 1\})$ such that $z = F_K(0, 0, x + i)$ then $x = x + i, w = F_K(0, 1, x + i)$ else $w \leftarrow \{0, 1\}^{\ell_2}$	\xrightarrow{w}
3		$w' = F_K(0, 1, x)$ If $w = w'$ then output $OK, x = x + 1, w' = F_K(1, 1, x)$ else output $\perp, w' = F_K(1, 0, x)$
4	If $w' = F_K(1, 1, x + 1)$ then output $ID, x = x + 1$ else output \perp	$\xleftarrow{w'}$

Figure 8: Stateful PRF-based RFID scheme.

indices of the form $F_K(0, 0, x + 1)$. Both $F_K(0, 0, x)$ and $F_K(0, 0, x + 1)$ will point to the same database record (ID, K, x) . When z is received by reader, the first set of indices is searched for it; if not found, then the second set is searched for it. Therefore, it takes $O(\log n)$ time to search for a tag, where n is the size of the databases. When the reader authenticates the tag, x is updated both in the database and in the indices sets. Moreover, the indices sets must be resorted. This can be simply done in $O(\log n)$ time because the old indices have to be removed and the new ones have to be reinserted in the right position.

A more efficient search can be performed by using hash indices as in (Alomair et al., 2012), where constant time is claimed for search. Moreover, the technique in (Alomair et al., 2012) applied to our scheme works much better than for the scheme in (Alomair et al., 2012). This is because the reader-tag desynchronization in our scheme is at most one step, while in (Alomair et al., 2012) it is bounded by some polynomial $c(\lambda)$ in the security parameter λ . This fact leads to $c(\lambda)$ sets of indices in (Alomair et al., 2012), while in our case we have only two.

Of course, the scheme is not weak private in the HPVP model because the tag identifier (z) is not randomized (Tiplea, 2022a). However, we have the following result.

Theorem 4.3. *The RFID scheme in Figure 8 assures r -weak privacy in the HPVP model.*

Proof. (sketch) Assume that the PRF-based RFID scheme, denoted Σ , does not assure r -weak privacy in the HPVP model. Then, there exists a weak adversary \mathcal{A} that has a non-negligible advantage against Σ .

Define a PRF adversary \mathcal{A}' that breaks the strong key indistinguishability property of F with non-negligible probability. First, assume that \mathcal{C} is a chal-

lenger for F that initiates the $SKD_{\mathcal{A}', F}^b$ game of \mathcal{A}' against F , where $b \in \{0, 1\}$. The basic steps are as follows:

1. \mathcal{A}' simulates the scheme Σ for \mathcal{A} :
 - (a) When \mathcal{A} asks for tag creation with identity ID , \mathcal{A}' will ask \mathcal{C} to generate a key and assign the identity ID to it. The internal state of the tag must also contain a counter x that is incremented after each completed session. However, working in the randomized model, this counter is randomized every time the tag is freed. As a result, \mathcal{A}' doesn't need to generate a random x to be recorded in the internal state of the tag along with the key generated by \mathcal{C} (please see below the query process). A specification \mathcal{T}_{ID} is returned to \mathcal{A} ;
 - (b) When \mathcal{A} draws $(\mathcal{T}_{ID_1}, \mathcal{T}_{ID_2})$ and asks for interrogation, \mathcal{A}' generates a random x and sends (ID_1, ID_2) and x to \mathcal{C} . Then, \mathcal{C} computes $z = F_{ID_b}(0, 0, x)$, and returns the result z to \mathcal{A}' . In turn, \mathcal{A}' returns z to \mathcal{A} ;
 - (c) To simulate the reader's answer w , \mathcal{A}' either generate a random value w or queries \mathcal{C} with the same pair of tags (ID_1, ID_2) and $(0, 1, x)$, depending on the case (please see the above item and the step 2 in the RFID scheme). Similarly, to simulate the tag's answer w' , \mathcal{A}' queries \mathcal{C} with the same pair of tags (ID_1, ID_2) and $(1, 1, x)$ or $(1, 0, x)$, depending on the case (please see the above item and the step 3 in the RFID scheme);
 - (d) $Result(\pi)$ is simulated similar to that in the proof of Theorem 4.1.
2. It is easy to check that

$$|P(PRIV_{\mathcal{A}, \Sigma}^b(\lambda) = 1) - P(SKD_{\mathcal{A}', F}^b(\lambda) = 1)|$$

is negligible for any b . As a conclusion,

$$|SKD_{\mathcal{A}',F}(\lambda) - PRIV_{\mathcal{A},\Sigma}(\lambda)|$$

is negligible.

Therefore, if $PRIV_{\mathcal{A},\Sigma}(\lambda)$ is non-negligible, then $SKD_{\mathcal{A}',F}(\lambda)$ is non-negligible, contradicting Theorem 3.2. \square

5 CONCLUSIONS

In recent years, many studies have been devoted to RFID schemes that provide a high degree of privacy in Vaudenay's or the HPVP model, such as forward-insider, destructive, or strong privacy. These schemes are based on public-key cryptography or combinations of symmetric cryptography with physically unclonable functions (PUFs). The use of public-key cryptography is expensive. Even though PUFs are considered light physical devices, their use in the construction of RFID schemes requires error correction methods, which makes their practical use difficult.

In contrast, using symmetric cryptography, RFID schemes that provide weak privacy or similar forms can be built. It is true, however, that such schemes cannot offer privacy against adversaries with the ability to corrupt tags (Tiplea, 2022b). However, there are many practical situations in which it is not necessary to use RFID schemes that are resistant to corruption attacks. This paper focuses on schemes that provide various forms of weak privacy in the HPVP model. In addition to highlighting such schemes, a central objective of this paper is to establish a fundamental property of pseudo-random functions (PRFs) through which one can show that a given RFID scheme achieves weak privacy. Thus, we believe PRF key indistinguishability is the defining property for PRF-based RFID schemes in the HPVP model (or any other model based on tag indistinguishability).

REFERENCES

- Alomair, B., Clark, A., Cuéllar, J., and Poovendran, R. (2012). Scalable RFID systems: A privacy-preserving protocol with constant-time identification. *IEEE Transactions on Parallel and Distributed Systems*, 3(8):1536–1550.
- Boneh, D. and Shoup, V. (2023). *A Graduate Course in Applied Cryptography*. Stanford University, California, USA.
- Tiplea, F. L. (2022b). Narrow privacy and desynchronization in Vaudenay's RFID model. *International Journal of Information Security*, 22:563–575.
- Hermans, J., Pashalidis, A., Vercauteren, F., and Preneel, B. (2011). A new RFID privacy model. In Atluri, V. and Diaz, C., editors, *Computer Security – ESORICS 2011*, pages 568–587, Berlin, Heidelberg. Springer Verlag.
- Hermans, J., Peeters, R., and Preneel, B. (2014). Proper RFID privacy: Model and protocols. *IEEE Transactions on Mobile Computing*, 13(12):2888–2902.
- Hristea, C. and Tiplea, F. L. (2020). Privacy of stateful RFID systems with constant tag identifiers. *IEEE Transactions on Information Forensics and Security*, 15:1920–1934.
- Katz, J. and Lindell, Y. (2020). *Introduction to Modern Cryptography*. Chapman & Hall/CRC, 3rd edition.
- Mittelbach, A. and Fischlin, M. (2021). *The Theory of Hash Functions and Random Oracles - An Approach to Modern Cryptography*. Information Security and Cryptography. Springer.
- Paise, R.-I. and Vaudenay, S. (2008). 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. ACM.
- Sipser, M. (2012). *Introduction to the Theory of Computation*. Cengage Learning.
- van Deursen, T. and Radomirović, S. (2012). Insider attacks and privacy of RFID protocols. In Petkova-Nikova, S., Pashalidis, A., and Pernul, G., editors, *Public Key Infrastructures, Services and Applications*, pages 91–105, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Vaudenay, S. (2007). On privacy models for RFID. In *Proceedings of the Advances in Cryptology 13th International Conference on Theory and Application of Cryptology and Information Security*, ASIACRYPT'07, pages 68–87, Berlin, Heidelberg. Springer-Verlag.
- Tiplea, F. L. (2022a). Lessons to be learned for a good design of private RFID schemes. *IEEE Transactions on Dependable and Secure Computing*, 19(4):2384–2395.