PUF-Based RFID Authentication Secure and Private under Memory Leakage

Daisuke Moriyama[1], Shin'ichiro Matsuo[1] and Moti Yung[23]

[1] National Institute of Information and Communications Technology, Japan
{dmoriyam, smatsuo}@nict.go.jp
[2] Columbia University
[3] Google, inc.

Abstract. RFID tags are getting their presence noticeable and are expected to become an important tool for e-commerce, logistics, point-of-sale transactions, and so on, representing "things" and "human holding things" in transactions. Since a huge amount of tags are expected to be needed to be attached to various "objects," a low-cost tag manufacturing is necessary. Thus, it is hard to imagine they will implement costly hardware protection mechanisms (like co-processor, TPMs). Therefore, in this context memory leakage (side-channel) attacks become a critical threat. Another well known threat to RFID devices is tag tracing implying violation of privacy.
We consider physically unclonable functions (PUFs) as tamper resilient building blocks cheaper than protected hardware, and propose security against a memory leaking adversary, trying to violate security and privacy of tags (we emphasize that digitally-oriented PUFs are easy to implement and they are more likely than TPMs to be implemented in RFID chips, more so than TPMs). We then design the first provably secure and provably private RFID authentication protocol withstanding information leakage from the entire memory of the tag, and show its two properties: (1) security against man-in-th-middle attack, and (2) privacy protection against tag tracing.

1 Introduction

We are in a middle of the next electronic and information revolution, where computing electronic devices are embedded everywhere and are, at times, connected to the computing networks, as part of the advanced information society. A critical component in this development is Radio Frequency Identification (RFID) technology, which is among the basic techniques allowing wireless communication between the reader (representing the infrastructure of interacting devices/terminals) and tags installed on gadgets, essentially without any human or other mediation interaction. The technology has far reaching implications on the potential evolution of the way transactions taking place in commercial and financial settings,

and it is being used currently in sectors such as point-of-sale payments (e.g., credit cards), transportation (e.g., toll payment), logistics (e.g., barcode replacement) including inventory management, transaction closing (physical presence indication for verified delivery of physical goods or tickets), etc. Authentication is the major task of RFID technology in all these applications.

Due to the fact that RFID communication signals reach several meters, RFID tags are expected to replace barcodes in enabling better ability to track and count objects, and associate goods. They also have a role in identifying participants (smartphone of a buyer, say): tags will be associated with computers on gadgets that will perform an initial part of a transaction (e-commerce) while its fulfilment will be triggered by the proper RFID presence at the point of fulfilment. Overall, it is hard to envision e-commerce advancing (in many transactions along the system: in payments, in identification of users, in shipments, in object/goods identification, etc.) without RFID technology.

On the other hand, the currently existing RFID tags directly convey their identity, and therefore continuous usage leaks a lot of personal information about the users carrying them. Though several companies planned to use such a basic RFID tag for speedy product management in the last decade, boycott campaigns were organized to protect against tracking consumer and these companies abandoned the use of RFID tag [8, 10]. It is expected that when RFID technology gets larger share of e-commerce transactions, attacks trying to learn secrets and violate privacy will be more and more attractive in this domain.

It is quite costly to implement secure components and secure storage in particular, like Trusted Platform Module [48], Mobile Platform Module [33], etc. Since RFID tags require low manufacturing cost to be economically viable, we cannot assume these cheap tags can have such secure components and run public key cryptography. Therefore, numerous works in the literature propose "lightweight cryptography" for resource restrained devices like the tags.

Physically Unclonable Function (PUF) is an emerging security technology, whose purpose is to introduce physical variation into individual devices taking part in cryptographic protocols. In many cases, digitally-oriented PUFs are constructed by variations of electric devices caused by the manufacturing process. This phenomenon has many instantiations (arbiter PUF [29], ring oscillator PUF [15], SRAM PUF [16], butterfly PUF [15, 25], latch PUF [45], etc.). These constructions are evaluated

in terms of non-uniformity of output or by temperature variation with FPGA or ASIC implementation [16, 27].

PUFs can be viewed as a tamper resilient building block [3,15,35] and the technology is attractive to low-cost devices likes RFID tag [26, 47]. Consider the scenario where a device keeps a secret in its non-volatile memory. The secret is not directly used within the cryptographic primitives, but rather it serves as an input to the PUF implemented in the device. If a physical characteristic increases entropy, then the output is unpredictable even if a malicious adversary obtains the secret key anytime. This means that the PUF can be used as a security enhancing mechanism. The previous protocol designs [26, 47], however, assumed that the tag's secret (contained in the non-volatile memory) is fixed and is reused in many sessions (which suffices for their purposes, since they do not cover secret key leakage). In contrast, our starting point in the current investigation is the fact that in RFID cheap technology, we cannot always assume that the memory is protected. Thus, we want to cover key leakage attack where the internal secret key is compromised, and we employ a stronger model allowing the adversary to obtain the secret key, in which case (due to their fixed key leakage) the previous assumptions and design rules do not apply (for privacy), and the adversary may be able to, e.g., identify which tag is interacted with the reader after the leakage attack. Since PUFs do not require special hardware, the implementation cost is lower than mechanisms like secure hardware such as TPMs.

Our Results. In this paper, we propose a provably secure RFID authentication protocol under an adversary who can continuously access to the internal memory except the protocol execution period; this is quite a strong attack extending the capabilities of earlier adversaries. The adversary attempts (1) to impersonate the user (violate security) and (2) to trace tags (violate privacy). We show that our protocol withstands the attacks, and to the best of our knowledge this is the first such secure symmetric-key based protocol in the memory leakage case. To achieve our goal, we introduce a new variant of the indistinguishability-based security model, originally proposed by Juels and Weis [23] such that the adversary can obtain the secret key of the target tag at any time. We note, in particular, that [37] showed that in symmetric key based RFID authentication protocols, it is natural to consider an active adversary which can desynchronize the secret key shared between the tag and the reader, in the cases when a protocol supports key update mechanism. Therefore, we assume that the reader and the tag can execute the honest session before and after the challenge phase in the privacy definition (see Section 3.3 for

more details on why this assumption is needed). Though security and privacy of the canonical RFID authentication protocols are easily violated if the secret key contained in the non-volatile memory is exposed, we show that PUF is a useful building block to overcome such a leakage.

Other Related Works. PUF has been mainly used in the setting of lightweight authentication protocols. One typical design of provably secure lightweight authentication protocol, originally introduced by Hopper and Blum, and its security is shown under learning parity with noise (LPN) assumption. The works in [20, 22] proposed variants of HB where the protocol structure is based on a prover who holds the PUF trying to convince the verifier to accept an authentication protocol invocation. Their protocol assumes that the verifier holds a (software based) function which can simulate the PUF, but we remark that software simulation in this context has been called "model building attack" [44], and is considered undesirable property in the PUF setting; (our setting, anyway, is not employing this idea). Kulseng et al. proposed a PUF-based RFID authentication protocol which supports key update mechanism [28], but their protocol is vulnerable to the typical man-in-the-middle attack as shown by Kardas et al. [24]. Several other cryptographic primitives based on PUF are proposed in [1] and [4]. Armknecht et al. showed an encryption scheme (a variant of Luby-Rackoff cipher) secure against memory leakage attack [1]. In their scheme, PUF and fuzzy extractor [12] are replaced by a pseudorandom function. While this application is not interactive authentication, it sets the setting of memory leakages in a device employing PUFs. Brzuska et al. proposed PUF-based cryptographic protocols: oblivious transfer, commitment and key exchange protocol [4]. However, Rührmair and Dijk showed that their oblivious transfer protocol does not hold the hiding property [42]. Ostrovsky et al. proposed another commitment scheme on the condition that an adversary can create and access to malicious PUFs [39].

2 Preliminaries

2.1 Notation

When A is a probabilistic machine or algorithm, $A(x)$ denotes the random variable of the output of A on input x. $y \xleftarrow{\mathsf{R}} A(x)$ denotes that y is randomly selected from $A(x)$ according to its distribution and $y := A(x)$ denotes that an output of deterministic algorithm $A(x)$ is assigned to y.

$A(x) \to a$ indicates the event that A outputs a on input x if a is a value. When A is a set, $y \xleftarrow{\mathsf{U}} A$ means that y is uniformly selected from A.

2.2 Fuzzy Extractor

A (d, h)-fuzzy extractor FE consists of two algorithms: key generation algorithm $\mathsf{FE.Gen}$ and reconstruction algorithm $\mathsf{FE.Rec}$. The $\mathsf{FE.Gen}$ algorithm takes as input a variable z and output a key r and helper data hd. For correctness, $\mathsf{FE.Rec}$ recovers the key r from the input variable z' and helper data hd if the hamming distance between z' and z is at most d. The fuzzy extractor satisfies security if the min-entropy of input z is at least h, r is statistically close to a uniformly random variable in $\{0,1\}^k$, even if the helper data is disclosed.

2.3 Physically Unclonable Function

A Physically Unclonable Function (PUF) is a function derived from a physical characteristic. There are many PUFs integrated in the digital circuit (arbiter PUF, ring oscillator PUF, SRAM PUF, latch PUF, etc.) and its physical properties are still under investigation (see [6, 46]). The basic purpose of these PUFs is to produce a device specific output for any input[4] like as a fingerprint. We present several required properties (common in the literature) for the PUFs.

Throughout the paper, k denotes a security parameter and f denotes a description of the PUF (e.g., arbiter PUF). f takes as input a physical characteristic x and message y, and outputs $z \xleftarrow{\mathsf{R}} f(x, y)$. x is an abstraction of the physical state and it may not be described with a mathematical expression. $f(x, \cdot)$ denotes a PUF-enabled device. Because the main purpose of PUF is to exploit the uniqueness from the internal state of the device, we distinguish the multiple PUFs by writing $f(x_1, \cdot), f(x_2, \cdot), \ldots$. We say that f is $(d, n, \ell, h, \epsilon)$-secure PUF if the following requirements hold:

1. For arbitrary inputs $y_1, y_2 \in \{0,1\}^k$, the variation from the same inputs is at most d_1 and the variation from the different outputs is at least d_2. That is, $\Pr[\mathrm{HD}(z_1, z_2) \leq d_1 \wedge \mathrm{HD}(z_1, z_3) \geq d_2 \wedge \mathrm{HD}(z_1, z_4) \geq d_2 \mid z_1 \xleftarrow{\mathsf{R}} f(x_1, y_1), z_2 \xleftarrow{\mathsf{R}} f(x_1, y_1), z_3 \xleftarrow{\mathsf{R}} f(x_1, y_2), z_4 \xleftarrow{\mathsf{R}} f(x_2, y_1)] = 1$ for any physical characteristics x_1, x_2 where HD evaluates the hamming distance.

[4] For SRAM PUF, memory addresses can be treated as the input.

2. Generate n PUFs and evaluate them for different inputs $y_1, \ldots, y_\ell \xleftarrow{\mathsf{U}} \{0,1\}^k$. The conditional min-entropy of the PUF's output variable, given the other outputs $\bar{H}_\infty(f(x_{i^*}, y_{j^*}) \mid \{f(x_i, y_j)\}_{1 \le i \le n, 1 \le j \le \ell, i \ne i^*, j \ne j^*})$ for any i^*, j^* is at least h.
3. Even if physical attacks are executed, the malicious adversary can obtain no extra information than the input-output behavior. Let $\mathcal{A}$ be an adversary who can physically access f to launch physical attacks. $\mathcal{S}$ is an algorithm which only interacts with f via oracle access. For any distinguisher $\mathcal{D}$, their output is indistinguishable such that $|\Pr[\mathcal{D}(1^k, st) \to 1 \mid st \xleftarrow{\mathsf{R}} \mathcal{A}(1^k, f(x, \cdot))] - \Pr[\mathcal{D}(1^k, st) \to 1 \mid st \xleftarrow{\mathsf{R}} \mathcal{S}^{f(x,\cdot)}(1^k)]| \le \epsilon$.

Though there are many security models for PUFs [1,2,4,15,16,39,43], it is hard to determine which model is the most suitable since the physical behavior depends on the implementation. Instead, we give requirements to provide provable security for our protocol. Because the fuzzy extractor is usually applied for the output variable from the PUF, the hamming distance between two outputs z_1 and z_2 derived by the same input must be at most d_1 and that from other outputs must be sufficiently large to avoid collision. Moreover, other outputs derived by any different device or input should not give sufficient information allowing to guess $z_{i,j}$.

The third requirement above formalizes tamper resistance properties of PUFs. Following [14,17], we describe simulation-based definition, stating that physical tampering does not provide any negative effect when compared against the black-box oracle access attack. It is known that several PUFs (excluded from our implementations) do not satisfy this property [18,36,40].

3 Security Model for PUF-based RFID Authentication Protocols

Consider an RFID reader $\mathcal{R}$ that interacts with RFID tags $\mathcal{T} := \{t_1, \ldots, t_n\}$. The reader runs a setup algorithm $\mathsf{Setup}(1^k)$ and generates public parameter pp and secret keys sk. In the authentication phase, mutual authentication is executed between $\mathcal{R}$ and a tag in $\mathcal{T}$. Finally, the parties output 1 (acceptance) or 0 (rejection) as the authentication result, respectively. A communication sequence between them is called a session, and each session is distinguished by session identifier sid which contains a series of communication message. We say that a session has a matching session if the communication messages generated by the reader and the tag are

honestly transferred until they authenticate each other. The correctness of the RFID authentication protocol is that the reader and the tag always accept the session if the session has the matching session.

3.1 Key Update Mechanism and Timing of Key Reveal

One of the major security threats corresponding to RFID tags is violating privacy via tracing the objects. If a secret key contained in the tag is fixed and the RFID-attached object is stolen by an adversary, he can learn when the tag launched sessions with the actual owner. Another threat is that the RFID tag is quite cheap and security chip is hard to implement. Many RFID authentication protocols provide key update mechanisms to minimize the above problem. We depict how to load and update the secret key from a non-volatile memory and when the adversary may be able to learn the secret key in our security model in Figure 1.

We concentrate on the passive RFID tag such that there is no internal battery and power is supplied by the reader. Thus a secret key of the tag sk_i is always loaded from the non-volatile memory to the volatile memory at the beginning of the protocol execution. When the protocol is finished and a secret key to be updated sk_{i+1} is generated, the tag saves sk_{i+1} to the non-volatile memory before leaving from the reader.

In this model, we assume that no physical attack against tags is mounted during the protocol execution. The adversary does not obtain any intermediate state in the volatile memory. Instead, the adversary can obtain all information contained in the non-volatile and volatile memory via an oracle query. Because the content stored in the non-volatile memory is not changed until the session is finished, the adversary can eventually learn the secret key used for the next session. While the adversary can access to the volatile memory between an interval of protocol executions, we can assume that no critical information is leaked from this memory because it can be erased after the protocol execution.

3.2 Security

The security requirement for RFID authentication protocols is commonly defined in the previous works. Intuitively, security requires that the reader and the tag reject the session when an active adversary modifies the communication before the verification is executed by each device. In all previous works the adversary cannot obtain any secret information about the RFID tag (via a "reveal" query). This seems like a natural assumption since the reader checks whether the response is computed by the secret

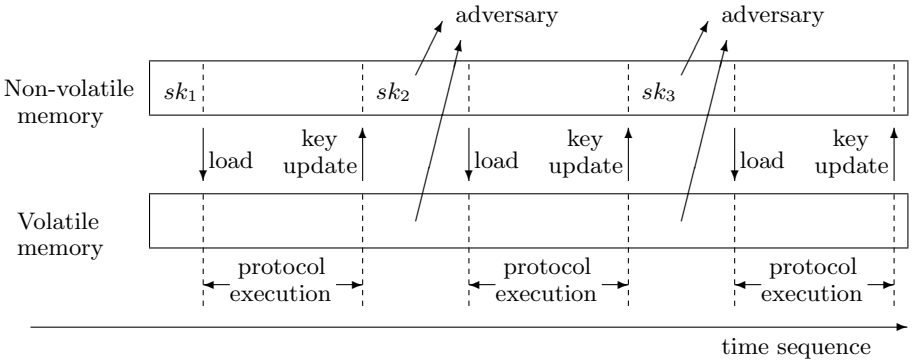

Fig. 1. Transition of the secret key and the timing of the reveal query

key which is contained in the RFID tag, In this paper, we actually assume a secure PUF (and fuzzy extractor) is implemented in the RFID tag, and we do allow the adversary to issue a "reveal" query in the security game (where it gets the content of the memory, while the PUF is by definition tamper proof as modeled above).

More formally, we consider the security game between a challenger and adversary $\mathcal{A}$ against an RFID authentication protocol Π.

$\underline{\mathsf{Exp}^{\mathsf{Sec}}_{\Pi,\mathcal{A}}(k)}$
$(pp, sk) \xleftarrow{\mathsf{R}} \mathsf{Setup}(1^k)$;
$(\mathsf{sid}^*, P) \xleftarrow{\mathsf{R}} \mathcal{A}_1^{\mathsf{Launch},\mathsf{SendReader},\mathsf{SendTag},\mathsf{Result},\mathsf{Reveal}}(pp, \mathcal{R}, \mathcal{T})$;
$b := \mathsf{Result}(\mathsf{sid}^*, P)$;
Output b

Upon receiving $(pp, \mathcal{R}, \mathcal{T})$, the adversary can issue the following oracle queries $\mathcal{O} := (\mathsf{Launch}, \mathsf{SendReader}, \mathsf{SendTag}, \mathsf{Result}, \mathsf{Reveal})$, instructed as the following:

- $\mathsf{Launch}(1^k)$: Launch the reader to start a new session.
- $\mathsf{SendReader}(m)$: Send arbitrary message m to the reader.
- $\mathsf{SendTag}(t, m)$: Send arbitrary message m to the tag $t \in \mathcal{T}$.
- $\mathsf{Result}(\mathsf{sid}, P)$: Output whether the session sid of P is accepted or not where $P = \{\mathcal{R}, \mathcal{T}\}$.
- $\mathsf{Reveal}(t)$: Output whole information contained in the memory if t is not running a session as explained in Section 3.1.

The advantage of adversary $\mathcal{A}$ against an RFID authentication protocol Π, $\mathsf{Adv}^{\mathsf{Sec}}_{\Pi,\mathcal{A}}(k)$, is defined by probability that $\mathsf{Exp}^{\mathsf{Sec}}_{\Pi,\mathcal{A}}(k)$ outputs 1 on

the condition that sid^* of P has no matching session. Recall that the adversary can learn the memory content of the RFID tag (i.e. the secret key contained in the non-volatile memory) and mount man-in-the-middle attack.

Definition 1. *An RFID authentication protocol Π is secure against man-in-the-middle attack with memory leakage if for any probabilistic polynomial time adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{Sec}}_{\Pi,\mathcal{A}}(k)$ is negligible in k (for large enough k).*

3.3 Privacy

Different from the security property, various privacy definitions are proposed even for the canonical RFID authentication protocol (to deal with tracing of tags) [7, 11, 19, 21, 23, 31, 41, 49]. A major problem has been "how to formalize a suitable privacy model for lightweight RFID authentication protocol". When a symmetric-key primitive is the main building block of the protocol, the reader shares a secret key with the tag for authentication. Thus, to minimize the influence of tag's key leakage, several key update mechanisms have been proposed in previous protocols to accommodate forward privacy. However, Ng et al. [37] showed that the de-synchronization attack is inevitable and the tag's secret key and authentication results cannot be allowed to leak at the same time in the Paise-Vaudenay privacy model [41]. Instead, we introduce a variant of the indistinguishability-based privacy model based on the Juels-Weis privacy model [23] to overcome this restriction.

In the original Juels-Weis model, the adversary chooses two RFID tags and accesses one of the two anonymously to evaluate the gap between them. Though this model allows the adversary to issue reveal queries to tags, the adversary cannot issue the reveal query to the above two tags. In contrast, in our modified model, we allow the adversary to issue the reveal query against these tags to cover backward and forward privacy. To assure that privacy still makes sense under such conditions, we add a restriction that an honest protocol execution without active adversary is launched before and after the anonymous access (i.e., the challenge phase): This is done to locally neutralize prior and future tracing compromises and allow some state update to take place before and after the challenge phase (i.e., with a little bit of lack of continued tracing by the same reader, we can achieve privacy). The proposed privacy model between the challenger and adversary $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ is then described as follows.

$\underline{\mathsf{Exp}^{\mathsf{IND}^*\text{-}b}_{\Pi,\mathcal{A}}(k)}$
$(pp, sk) \xleftarrow{\mathsf{R}} \mathsf{Setup}(1^k)$;
$(t_0^*, t_1^*, st_1) \xleftarrow{\mathsf{R}} \mathcal{A}_1^{\mathcal{O}}(pp, \mathcal{R}, \mathcal{T})$;
$b \xleftarrow{\mathsf{U}} \{0,1\}, \mathcal{T}' := \mathcal{T} \setminus \{t_0^*, t_1^*\}$;
$\pi_0 \xleftarrow{\mathsf{R}} \mathsf{Execute}(\mathcal{R}, t_0^*),\ \pi_1 \xleftarrow{\mathsf{R}} \mathsf{Execute}(\mathcal{R}, t_1^*)$;
$st_2 \xleftarrow{\mathsf{R}} \mathcal{A}_2^{\mathcal{O}}(\mathcal{R}, \mathcal{T}', \mathcal{I}(t_b^*), \pi_0, \pi_1, st_1)$:
$\pi_0' \xleftarrow{\mathsf{R}} \mathsf{Execute}(\mathcal{R}, t_0^*),\ \pi_1' \xleftarrow{\mathsf{R}} \mathsf{Execute}(\mathcal{R}, t_1^*)$;
$b' \xleftarrow{\mathsf{R}} \mathcal{A}_3^{\mathcal{O}}(\mathcal{R}, \mathcal{T}, \pi_0', \pi_1', st_2)$;
Output b'

Same as the security game, the adversary can interact with the reader and tags via oracle queries in $\mathcal{O}$. When the adversary $\mathcal{A}_1$ sends two tags (t_0^*, t_1^*) to the challenger, a random coin b is flipped and the challenger proceeds to the challenge phase such that the adversary can access to the challenge tag t_b^* anonymously. To accomplish anonymous access, $\mathcal{A}_2$ can issue the $\mathsf{SendTag}$ query with input intermediate algorithm $\mathcal{I}$ which honestly transfers the communication message between $\mathcal{A}_2$ and t_b^*. When the adversary issues $\mathsf{SendTag}(\mathcal{I}, m)$, $\mathcal{I}$ sends m to t_b^* and responds the message from t_b^*. Thus, the adversary can communicate with t_b^* without submitting tag's identity. After the challenge phase, $\mathcal{A}_3$ can continuously interact with all tags including (t_0^*, t_1^*) as $\mathcal{A}_1$.

The main difference from the existing indistinguishability-based privacy definitions [11, 23, 32] is that the adversary can always issue the reveal query and obtain the secret key of the tag. As we noted in Introduction, RFID tags are low-cost devices and it is hard to embed secure storage like TPMs. Thus leakage from the non-volatile memory is arguable and we consider PUF-enabled RFIDs. On the other hand, Ng et al. [37] showed that the de-synchronization attack is inevitable problem for all symmetric-key based RFID authentication protocols with key update mechanism. In particular, it is natural for these authentication protocols that the internal secret key of the tag is updated if the tag accepts the session to ensure forward secrecy. In other words, the secret key is not updated when the session is rejected. Because the adversary can send arbitrary message, it is trivial for the adversary to trace a tag when the Reveal query is issued before or after the anonymous access and the secret key of t_b^* is not updated during the anonymous access phase. Therefore, we admit a re-synchronization opportunity before and after the anonymous access. The $\mathsf{Execute}$ query is the normal protocol execution between the reader and the tag. The adversary cannot modify the

communications but the transcripts (π_0, π_1) and (π'_0, π'_1) are delivered to the adversary.

One can think that the above definition covers both forward and backward privacy. Once an honest protocol execution is finished, no one can trace the tag even if the internal information of the tag before and after the session is continuously leaked to a third party.

Finally, the advantage of the adversary in guessing the correct tag bit is evaluated as $\mathsf{Adv}^{\mathsf{IND}^*}_{\Pi,\mathcal{A}}(k) := |\Pr[\mathsf{Exp}^{\mathsf{IND}^*\text{-}0}_{\Pi,\mathcal{A}}(k) \to 1] - \Pr[\mathsf{Exp}^{\mathsf{IND}^*\text{-}1}_{\Pi,\mathcal{A}}(k) \to 1]|$.

Definition 2. *An RFID authentication protocol Π satisfies the modified indistinguishability-based privacy under memory leakage if for any probabilistic polynomial time (PPT) adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{IND}^*}_{\Pi,\mathcal{A}}(k)$ is negligible in k (for large enough k).*

The proposed privacy definition requires that the secret key contained in the RFID tag gives no information to distinguish among RFID tags. This is a novel approach, and for example, the PUF-based RFID authentication protocol in [47] does not satisfy our privacy model since the secret key of the RFID tag is always fixed.

4 How to Apply PUF in Cryptographic Protocols

It is known that even if one selects an input and evaluates the PUF multiple times, the physical circuit causes small noise and the output is not deterministically defined [15, 16, 29, 45]. Moreover, while unpredictability of its output is desirable, we cannot treat the PUF's output as a pseudorandom string. Applying a fuzzy extractor to the PUF's output is an easy solution to overcome these problems. Thus one of the major applications of the PUF combined with the fuzzy extractor is to extract a secret key from an input.

The first PUF-based RFID authentication protocol was proposed by Sadeghi, Visconti and Wachsmann [47]. However, we slightly modify their protocol in the spirit of van Herrewege et al. [50] who found that there are two typical ways to apply the fuzzy extractor in any PUF-based protocols (typical, as in the original [47], and reverse ways). We follow their idea and describe two PUF-based RFID authentication protocols and discuss their security and privacy threats.

In the typical mode, the verifier (e.g., the RFID reader) evaluates the PUF f and runs the $\mathsf{FE.Gen}$ algorithm to obtain a random key and helper data before the authentication. Upon receiving the input to the PUF and

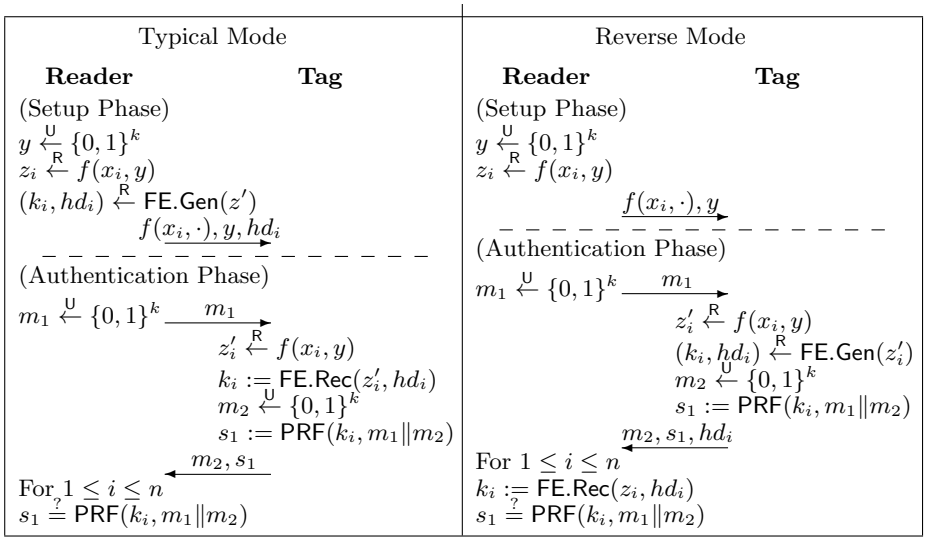

Fig. 2. Previous PUF-based RFID authentication protocols

helper data, the prover (the RFID tag) recovers the secret with the $\mathsf{FE.Rec}$ algorithm. In contrast, Herrewege et al. showed that the fuzzy extractor can be applied in an opposite fashion [50]. That is, the verifier evaluates the PUF and sends its input to the prover in the reverse mode. Then the prover computes the PUF and runs the $\mathsf{FE.Gen}$ algorithm. When the helper data is sent from the prover, the verifier reconstructs the random key with the $\mathsf{FE.Rec}$ algorithm.

In either case, the non-volatile memory contains a "superficial" secret key y and the PUF with fuzzy extractor derives the actual secret key k_i from that superficial key. When the reader evaluates the PUF and obtains input/output pair in the setup phase, k_i can be used as a secret key for any symmetric key primitives. Note that the existing RFID authentication protocols specify that the tag directly keeps k_i. Thus, the main advantage of the above PUF-based RFID authentication protocol is that the leakage of y, the secret key kept in the non-volatile memory, does not imply the total break of the tag.

However, this additional mechanism does not increase the tag's privacy right away. Even though an adversary cannot impersonate the tag under the memory leakage attack, the fixed secret key contained in the non-volatile memory leaks the tag's identity in the above protocol. Specifically, the adversary can easily break the privacy game described in Section 3.3 since the superficial secret key y is reused in many sessions and the adversary can trace the tag. Thus, we conclude that for privacy rea-

sons, we must establish a key update mechanism for PUF-based RFID authentication protocols. This is indeed, a step in the right direction and a starting point for us, since we can strengthen the security requirement, allowing the adversary to issue the reveal query at any time.

One technical problem in supporting a key update mechanism in PUF-based RFID authentication protocol is that the reader can directly handle the PUF only in the setup phase. One straightforward solution is to observe a lot of input/output pairs of the PUF and write inputs to the tag before the authentication phase. But, this method is quite inefficient from the perspective of, both, the tag and the reader. In our protocol, we employ another principle of careful chaining, where the tag securely transfers the output of the PUF which will be used in the next activation.

Another issue for PUF-based cryptographic protocols is how to transfer the helper data in a secure way. In particular, helper data hd_i is sent as a plaintext in the reverse mode as described in Figure 2. Indeed, [50] pointed out that the outsider chosen perturbation security introduced by Boyen [9] is needed in the above case. In contrast, PUF-friendly fuzzy extractors are proposed in several works [5,35] to minimize the implementation cost. Of course, there is no guarantee that these fuzzy extractors satisfy the outsider chosen perturbation security. Moreover, if we compute the fuzzy extractor multiple times which the hamming distance among inputs is sufficiently small, it may derive correlated helper data so that the adversary can trace a specific tag. Nonetheless, if we transfer the helper data in a secure way during the protocol execution, we need not rely on such highly secure fuzzy extractor.

5 The Proposed Protocol

Setup Phase. The reader $\mathcal{R}$ selects $y_1 \xleftarrow{\mathsf{U}} \{0,1\}^k$ and inputs it to the PUF $z_1 \xleftarrow{\mathsf{R}} f(x_i, y_1)$. It computes $(r_1, hd_1) := \mathsf{FE.Gen}(z_1)$ and sends $(f(x_i, \cdot), y_1, hd_1)$ to the RFID tag t_i. The PUF $f(x_i, \cdot)$ is already implemented in the tag, so the reader computes it with the tag itself. The reader keeps $(r_1, r_{old} := r_1, t_i)$ in the database.

Authentication Phase. The reader holds database $\{(r_1, r_{old}, t_i)\}_{i \in \mathcal{T}}$ and the tag t_i keeps (y_1, hd_1) in its memory. Let $\mathcal{G} : \{0,1\}^k \times \{0,1\}^{2k} \to \{0,1\}^{6k}$ and $\mathcal{G}' : \{0,1\}^k \times \{0,1\}^{2k} \to \{0,1\}^k$ be pseudorandom functions (PRFs).

– The reader chooses nonce $m_1 \xleftarrow{\mathsf{U}} \{0,1\}^k$ and sends it to the tag.

– Upon receiving m_1, the tag runs the following steps:
  1. Compute $z'_1 \xleftarrow{\mathsf{R}} f(x_i, y_1)$.
  2. Obtain $r_1 := \mathsf{FE.Rec}(z'_1, hd_1)$.
  3. Select $m_2 \xleftarrow{\mathsf{U}} \{0,1\}^k$.
  4. Compute $(s_1, \ldots, s_6) := \mathcal{G}(r_1, m_1 \| m_2)$.
  5. Choose $y_2 \xleftarrow{\mathsf{U}} \{0,1\}^k$.
  6. Compute $z'_2 \xleftarrow{\mathsf{R}} f(x_i, y_2)$.
  7. Compute $u_1 := s_2 \oplus z'_2$ and $v_1 := \mathcal{G}'(s_3, m_2 \| u_1)$.
  8. Send (m_2, s_1, u_1, v_1) to the reader.
– When the reader $\mathcal{R}$ receives (m_2, s_1, u_1, v_1), it runs the following steps:
  1. Compute $(s'_1, \ldots, s'_6) := \mathcal{G}(r_1, m_1 \| m_2)$ and check $s'_1 = s_1$ for some $1 \leq i \leq n$. If this search fails, the reader skips the following procedure.
  2. Verify $v_1 = \mathcal{G}'(s'_3, m_2 \| u_1)$. If this verification fails, the reader skips the following procedure.
  3. Decrypt $z_2 := s'_2 \oplus u_1$.
  4. Obtain $(r_2, hd_2) \xleftarrow{\mathsf{R}} \mathsf{FE.Gen}(z_2)$.
  5. Compute $u_2 := s'_5 \oplus hd_2$ and $v_2 := \mathcal{G}'(s'_6, m_1 \| u_2)$.
  6. Send (s'_4, u_2, v_2) to the tag and update $(r_1, r_{old}) := (r_2, r_1)$.

  The reader repeats the above with r_{old} instead of r_1. If the above verifications do not hold, then the reader rejects the session and sends randomly chosen (s'_4, u_2, v_2) to the tag.
– Upon receiving (s'_4, u_2, v_2), the tag checks $s'_4 = s_4$ and $v_2 = \mathcal{G}'(s_6, m_1 \| u_2)$. If the verifications hold, u_2 is decrypted as $hd_2 := s_5 \oplus u_2$ and the tag updates (y_1, hd_1) to (y_2, hd_2). Finally, the tag erases all of the data in the volatile memory.

Intuitively, our protocol is "challenge response authentication" with PRF $\mathcal{G}$. The seed input to the function is generated by PUF f and the fuzzy extractor. If the tag does not accept any adversarial message, the tag always computes r_1 or r_{old} (when the adversary executes desynchronization attack) and the reader can authenticate the tag. Moreover, the tag generates the next input to the PUF y_2 and sends its output z'_2 to the reader in a secure way (XOR based one-time-pad and MAC with PRF $\mathcal{G}'$). If the tag authentication is accepted, the reader securely sends the next helper data hd_2 as the tag's computation.

One can imagine authenticated encryption against (u_1, v_1) and (u_2, v_2). So (u_1, u_2) is the ciphertext of the plaintext and (v_1, v_2) is the tag of the MAC. m_2 is randomly chosen by the tag and changed per session, the

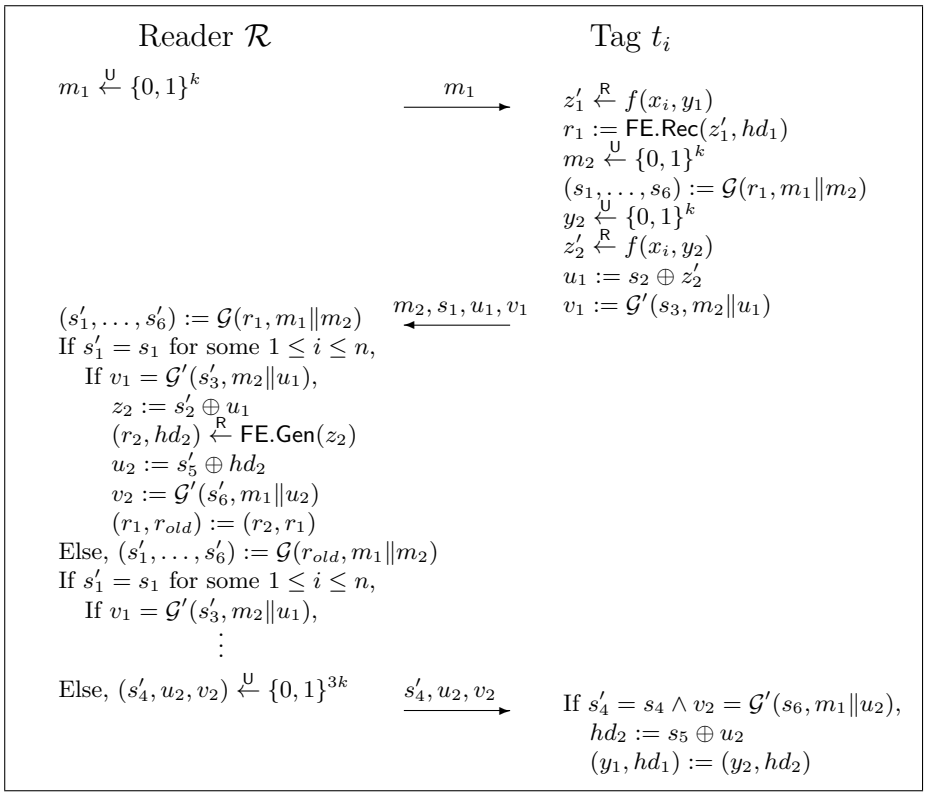

Fig. 3. The proposed PUF-based RFID authentication protocol

XOR operation is sufficient in our protocol instead of a standard symmetric key encryption algorithm. The MAC tag generation is replaced by the computation of PRF to minimize implementation cost. Obviously, when the tag holding the keys and the PUF performs, then the reader will accept the authentication (in or out of synchronization).

One may think that the exhaustive search launched by the reader is quite inefficient — in the worst case, the reader computes the checking procedure $2n$ times. However, it is inevitable to keep privacy and a standard method in RFID authentication protocols.

We note that it is hard to achieve resilience to all side-channel attacks for cryptographic protocols. Even if a secure PUF is implemented in the RFID tag, the adversary may execute side-channel attacks against other building blocks, fuzzy extractor or pseudorandom function. Though it is out of scope of our paper, several works have investigated leakage

resilience for fuzzy extractor and pseudorandom function to increase security [13, 34].

Theorem 1. *Let* FE *be a* (d, h)*-fuzzy extractor and* $(d, n, \ell, h, \epsilon)$*-secure physically unclonable function. Assume that* $\mathcal{G}$ *and* $\mathcal{G}'$ *are secure pseudorandom functions. Then our protocol is secure against man-in-the-middle attack with memory leakage.*

Proof. The goal of the adversary $\mathcal{A}$ is for the reader or the tag to accept the session while the communication is modified by the adversary. We concentrate only on the former case, since the reader authentication is quite similar to that of the tag. We consider the following game transformations. Let S_i be the advantage that the adversary wins the game in Game i.

Game 0. This is the original game between the challenger and the adversary.

Game 1. The challenger randomly guesses the tag $t^* \xleftarrow{\mathsf{U}} \{t_1, \ldots, t_n\}$. If the adversary does not impersonate t^* to the reader, the challenger aborts the game.

Game 2. Assume that ℓ is the upper bound of the sessions that the adversary can establish in the game. For $1 \leq j \leq \ell$, we evaluate or change the variables related to the the session between the reader and t^* up to the ℓ-th session as the following.

- **Game 2-j-1.** The challenger evaluates the output from the PUF implemented in t^* at the j-th session. If the output does not have enough entropy or is correlated to the other outputs derived from the other inputs or the PUF, then the challenger aborts the game.
- **Game 2-j-2.** The output from the fuzzy extractor (r_{old}, r_1) is changed to a random variable.
- **Game 2-j-3.** The output from the PRF $\mathcal{G}(r_1, \cdot)$ is derived from a truly random function in this game.
- **Game 2-j-4.** We change the PRF $\mathcal{G}(r_{old}, \cdot)$ to a truly random function.
- **Game 2-j-5.** We change the XORed output $u_1 := s_2 \oplus z_2'$ and $u_2 := s_5' \oplus hd_2$ to randomly chosen $u_1, u_2 \xleftarrow{\mathsf{U}} \{0,1\}^k$.
- **Game 2-j-6.** The outputs from the PRFs $\mathcal{G}'(s_3, \cdot)$ and $\mathcal{G}'(s_6', \cdot)$ are derived from a truly random function in this game.

The basic strategy of the security proof is to change the communication messages corresponding to the target tag t^* to random variables. However, we must take care of the key chaining mechanism in our protocol that

updated secret keys are XORed by (s_2, s_5) which is derived by current secret key. So we must proceed the game transformation starting from the first invocation of the tag t^*, communication messages are gradually changed from Game 2-j-1 to Game 2-j-6. When these transformations are finished, we can move to the next session. We recursively apply this strategy up to the upper bound of the t^*'s activation.

If the PUF implemented in the tag generates enough entropy, the fuzzy extractor can provide variables which are statistically close to random strings. Then, this output can be applied to the PRF as the seed and the RFID reader and the tag share the common secret. So we can construct the challenge response authentication protocol with secure key update.

Lemma 1. *$S_0 = n \cdot S_1$ (where n is the number of RFID tags).*

Proof. If the adversary wins the game, there is at least one session which the reader or tag accepts the session while the communication is modified by the adversary. Since the challenger randomly selects a tag, the probability that the tag impersonated by the adversary is correctly guessed by the challenger is at least $1/n$.

Lemma 2. *$S_1 = S_{2\text{-}1\text{-}1}$ and $S_{2\text{-}(j-1)\text{-}6} = S_{2\text{-}j\text{-}1}$ for any $2 \leq j \leq \ell$ if f is a $(d, n, \ell, h, \epsilon)$-secure PUF.*

Proof. If the output from the PUF has enough min-entropy and is independent from the other outputs, there is no difference between these games. Since we now assume here the PUF has a desirable property (described in Section 2.3) that even if the input to the PUF is published, the output derived from the input keeps the sufficient min-entropy property, and therefore each output is uncorrelated [5]. Hence, even if the adversary issues the reveal query and obtains the secret key contained in the non-volatile memory (which is used to input to the PUF), there is no negative effect to proceed the game transformation.

Lemma 3. *$S_{2\text{-}j\text{-}1} = S_{2\text{-}j\text{-}2}$ for any $1 \leq j \leq \ell$ if the FE is a (d, h)-fuzzy extractor.*

Proof. Since we assumed that, always, the output from the PUF has enough min-entropy, it is clear that no adversary can distinguish these games due to the randomization property of the fuzzy extractor.

Lemma 4. *$\forall 1 \leq j \leq \ell$, $|S_{2\text{-}j\text{-}2} - S_{2\text{-}j\text{-}3}| \leq \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{G},\mathcal{B}}(k)$ where $\mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{G},\mathcal{B}}(k)$ is an advantage of $\mathcal{B}$ to break the security of the PRF $\mathcal{G}$.*

[5] That is, the challenger does not check the entropy of the output in this game.

Proof. If there is a difference between these games, we construct an algorithm $\mathcal{B}$ which breaks the security or PRF $\mathcal{G}$. $\mathcal{B}$ can access the real PRF $\mathcal{G}(r_1, \cdot)$ or truly random function RF. $\mathcal{B}$ sets up all secret keys and simulates our protocol except the n-th session. When the adversary invokes the n-th session, $\mathcal{B}$ sends $m_1 \xleftarrow{\mathsf{U}} \{0,1\}^k$ as the output of the reader. When $\mathcal{A}$ sends m_1^* to a tag t_i, $\mathcal{B}$ selects m_2 and issues $m_1^* \| m_2$ to the oracle instead of the normal computation of $\mathcal{G}$. Upon receiving $(s_1, \ldots, s_6)$, $\mathcal{B}$ continues the computation as the protocol specification and outputs (m_2, s_1, u_1, v_1) as the tag's response. When the adversary sends $(m_2^*, s_1^*, u_1^*, v_1^*)$, $\mathcal{B}$ issues $m_1 \| m_2^*$ to the oracle and obtains $(s_1', \ldots, s_6')$. These variables are used in the tag authentication.

If $\mathcal{B}$ accesses the real PRF, this simulation is equivalent to Game 2-j-2. Otherwise, the oracle query issued by $\mathcal{B}$ is completely random and this distribution is equivalent to Game 2-j-3. Thus we have $|S_{2\text{-}j\text{-}2} - S_{2\text{-}j\text{-}3}| \leq \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{G},\mathcal{B}}(k)$.

Lemma 5. $\forall 1 \leq j \leq \ell,\ |S_{2\text{-}j\text{-}3} - S_{2\text{-}j\text{-}4}| \leq \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{G},\mathcal{B}}(k)$.

Proof. We can prove this lemma as the proof for Lemma 4.

Lemma 6. $\forall 1 \leq j \leq \ell,\ S_{2\text{-}j\text{-}4} = S_{2\text{-}j\text{-}5}$.

Proof. Since the PRF $\mathcal{G}(r_1, \cdot)$ is already changed to the truly random function in Game 1-j-4, s_2 and s_5' are used as effectively one-time pad to encrypt z_2' and hd_2, respectively. Therefore this transformation is purely conceptual change and the output distributions of these games are information theoretically equivalent.

Lemma 7. $\forall 1 \leq j \leq \ell,\ |S_{2\text{-}j\text{-}5} - S_{2\text{-}j\text{-}6}| \leq 2 \cdot \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{G}',\mathcal{B}'}(k)$.

Proof. We can think that the seed input to the PRF $\mathcal{G}'$ is changed to the random variable from the previous games. Consider an algorithm $\mathcal{B}$ which interacts with PRF $\mathcal{G}'(s_3', \cdot)$ or random function RF. As in the proof for Lemma 4, $\mathcal{B}$ simulates the protocol as the challenger up to the n-th session. $\mathcal{B}$ generates (m_2, u_1) and issues $m_2 \| u_1$ to the oracle. $\mathcal{B}$ generates the other variables as Game 5 and sends (m_2, s_1, u_1, v_1) as the tag's output after it obtains v_1 from the oracle. If the reader receives $(m_2^*, s_1^*, u_1^*, v_1^*)$, $\mathcal{B}$ checks that $(m_2^*, s_1^*) = (m_2, s_1)$. If so, $\mathcal{B}$ issues $m_2^* \| u_1^*$ to the oracle to check whether its response is identical to v_1^*.

If $\mathcal{B}$ accesses the real PRF, this simulation is equivalent to Game 2-j-5. Otherwise, $\mathcal{B}$'s simulation is identical to Game 2-j-6. Thus the difference between these games are bounded by the security of PRF $\mathcal{G}'$. Similarly, we evaluate the gap between $\mathcal{G}'(s_6', \cdot)$ and RF.

When we transform Game 0 to Game 2-ℓ-6, there is no advantage against the adversary to impersonate the tag. To accomplish man-in-the-middle attack, the adversary must modify (m_2, s_1, u_1, v_1) given from the tag. Consider this tuple as (m_2, s_1) and (u_1, v_1). When the adversary modifies m_2, the probability that the adversary wins the security game is negligible since s_1 is chosen from the truly random function. If m_2 is not changed, the reader only accepts s_1 since it is deterministically defined by m_1 chosen by the reader and m_2. The first verification is passed only when the adversary reuses (m_2, s_1), but v_1 is also derived from another random function. Thus the adversary cannot guess it and any modified message is rejected except with negligible probability. The same argument also applies to the reader authentication, because the tag checks the reader with the outputs from $\mathcal{G}$ and $\mathcal{G}'$.

Finally, we have

$$\mathsf{Adv}^{\mathsf{Sec}}_{\Pi,\mathcal{A}}(k) \leq \frac{1}{2\ell n} \cdot \left(\mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{G},\mathcal{B}}(k) + \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{G}',\mathcal{B}'}(k)\right)$$

if the PUF and fuzzy extractor holds properties described in Section 2.

□

Theorem 2. *Let* FE *be a* (d, h)*-fuzzy extractor and* $(d, n, \ell, h, \epsilon)$*-secure physically unclonable function. Assume that* $\mathcal{G}$ *and* $\mathcal{G}'$ *are secure pseudo-random functions. Then our protocol satisfies the modified indistinguishability-based privacy under memory leakage (described in Section 3).*

Proof. The proof we provide here is similar to that for Theorem 1. However, we remark that it is important to assume that our protocol satisfies security first for privacy to hold. The reason is that if the security is broken and a malicious adversary successfully impersonates tag t_0^*, the reader will update the secret key that is not derived by the tag any more. So the reader does not accept this tag after the attack and the adversary easily distinguishes the tag in the privacy game. Even if the adversary honestly transmits the communication message between $\mathcal{I}(t_0^*)$ and the reader in the challenge phase, the authentication result is always 0 and the adversary can realize which tag is selected as the challenge tag.

Based on the game transformation described in the proof of Theorem 1, we modify Game 1 such that the challenger guesses two tags which will be chosen by the adversary in the privacy game. This probability that is at least $1/n^2$, then, we can continue the game transformation. After that, the game transformation described in Game 2 is applied to the sessions related to t_0^* and t_1^*. Then the communication message (m_2, s_1, u_1, v_1) and

(s'_4, u_2, v_2) are changed to random variables. Even if the adversary can obtain the secret key of the tag within the privacy game, input to the PUF and helper data used in the challenge phase are independent from choices in the other phases. The re-synchronization allows this separation and new values are always random. Therefore, there is no information against which the adversary can distinguish the challenge tag in the privacy game, and we get:

$$\mathsf{Adv}^{\mathsf{IND}^*}_{\Pi,\mathcal{A}}(k) \leq \mathsf{Adv}^{\mathsf{Sec}}_{\Pi,\mathcal{A}'}(k) + \frac{1}{4\ell n^2} \cdot \left(\mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{G},\mathcal{B}}(k) + \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{G}',\mathcal{B}'}(k)\right)$$

for some algorithm $(\mathcal{A}', \mathcal{B}, \mathcal{B}')$ derived from the games. □

6 Conclusion

We considered security and privacy of RFID tags, and proposed a provably secure and private PUF-based RFID authentication protocol in the case that the adversary gets contents of memories. We investigated a new variant of the indistinguishability-based privacy model for RFID authentication protocol where the adversary can obtain the information contained in the tag's non-volatile memory. Our protocol is resilient to this memory leakage attack because the PUF can, in effect, serve as the secure component of the RFID tag, in a way which is sufficient to foil man-in-the-middle attack and tracing. Conceptually, the work has shown that an infrastructure reading RFID tags can be made robust to leakages given the adoption of PUFs of tags; this opens an avenue of design possibilities for RFID devices which will increase the robustness of RFID authentication and the major applications they span.

References

1. Armknecht, F., Maes, R., Sadeghi, A.R., Sunar, B., Tuyls, P.: Memory leakage-resilient encryption based on physically unclonable functions. In: ASIACRYPT 2009. LNCS, vol. 5912, pp. 685–702. Springer Heidelberg (2009)
2. Armknecht, F., Maes, R., Sadeghi, A.R., Standaert, F-X., Wachsmann, C.: A Formal Foundation for the Security Features of Physical Functions. In: IEEE S&P 2011, pp. 297–412. IEEE (2011)
3. Billet, O., Etrog, J., Gilbert, H.: Lightweight privacy preserving authentication for RFID using a stream cipher. In: FSE 2010. LNCS, vol. 6147, pp. 55–74. Springer Heidelberg (2010)
4. Brzuska, C., Fischlin, M., Schröder, H., Katzenbeisser, S.: Physically uncloneable functions in the universal composition framework. In: CRYPTO 2011. LNCS, vol. 6841, pp. 51–70. Springer Heidelberg (2011)

5. Bösch, C., Guajardo, J., Sadeghi, A.R., Shokrollahi, J., Tuyls, P.: Efficient helper data key extractor on FPGAs. In: CHES 2008. LNCS, vol. 5154, pp. 181–197. Springer Heidelberg (2008)
6. Böhm, C., Hofer, M.: Physical unclonable functions in theory and practice. Springer, Heidelberg (2013)
7. Burmester, M., Le, T.V., Medeiros, B.D., Tsudik, G.: Universally composable RFID identification and authentication protocols. TISSEC 2009 12(4) pp. 1–33. ACM (2009)
8. Boycott benetton no RFID tracking chips in clothing!. http://www.boycottbenetton.com/.
9. Boyen, X.: Reusable cryptographic fuzzy extractors. In: ACMCCS 2004. pp. 82–91. ACM (2004)
10. Consumers against supermarket privacy invasion and numbering (CASPIAN): Anti-RFID Campaign webpage available at http://www.spychips.com/.
11. Deng, R.H., Li, Y., Yung, M., Zhao, Y.: A new framework for RFID privacy. In: ESORICS 2010. LNCS, vol. 6345, pp. 1–18. Springer Heidelberg (2010)
12. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. In: SIAM J. Comput. 38 (1). pp. 97–139. SIAM (2008)
13. Dodis, Y., Pietrzak, K.: Leakage-Resilient Pseudorandom Functions and Side-Channel Attacks on Feistel Networks. In: CRYPTO 2010, LNCS, vol. 6223. pp. 21–40. Springer Heidelberg (2010)
14. Faust, S., Pietrzak, K., Venturi, D.: Tamper-Proof Circuits: How to Trade Leakage for Tamper-Resilience. In: ICALP 2011. LNCS, vol. 6755, pp. 391–402. Springer Heidelberg (2011)
15. Gassend, B., Clarke, D., Dijk, M., Devadas, S.: Silicon physical random functions. In: ACMCCS 2002. pp. 148–160. ACM (2002)
16. Guajardo, J., Kumar, S., Schrijen, G.J., Tuyls, P.: FPGA intrinsic PUFs and their use for IP protection. In: CHES 2007. LNCS, vol. 4727, pp. 63–80. Springer Heidelberg (2007)
17. Gennaro, R., Lysyanskaya, A., Malkin, T., Micali, S., Rabin, T.: Algorithmic Tamper-Proof (ATP) Security: Theoretical Foundations for Security Against Hardware Tampering. In: TCC 2004. LNCS, vol. 2951, pp. 258–277. Springer Heidelberg (2004)
18. Helfmeier, C., Boit, C., Nedospasov, D., Seifert, J.-P.: Cloning Physically Unclonable Functions. In: HOST 2013, pp. 1-6. IEEE (2013)
19. Ha, J., Moon, S., Zhou, J., Ha, J.: A new formal proof model for RFID location privacy. In: ESORICS 2008. LNCS, vol. 5283, pp. 267–281. Springer Heidelberg (2008)
20. Hammouri, G., Öztürk, E., Birand, B., Sunar, B.: Unclonable lightweight authentication scheme. In: ICICS 2008. LNCS, vol. 5308, pp. 33–48. Springer Heidelberg (2008)
21. Hermans, J., Pashalidis, A., Vercauteren, F., Preneel, B.: A new RFID privacy model. In: ESORICS 2011. LNCS, vol. 6879, pp. 568?587. Springer Heidelberg (2011)
22. Hammouri, G., Sunar, B.: PUF-HB : a tamper-resilient HB based authentication protocol. In: ACNS 2008. LNCS, vol. 5037, pp. 346–365. Springer Heidelberg (2008)
23. Juels, A., Weis, S.A.: Defining strong privacy for RFID. ACM TISSEC 13(1) (2009)
24. Kardas, S., Akgün, M., Kiraz, M.S., Demirci, H.: Cryptanalysis of lightweight mutual authentication and ownership transfer for RFID systems. In: LightSec 2011, pp. 20–25. IEEE (2011)

25. Jumar, S., Guajardo, J., Maes, R., Schrijen, G.J., Tuyls , P.: The butterfly PUF protecting IP on every FPGA. In: HOST 2008. pp. 67–70. IEEE (2008)
26. Kardas, S., Kiraz, M.S., Bingöl, M.A., Demirci, H.: A novel RFID distance bounding protocol based on physically unclonable functions. In: RFIDSec 2011. LNCS, vol. 7055, pp. 78–93. Springer, Heidelberg (2011)
27. Katzenbeisser, S., Kocabas, "U., Rozic, V., Sadeghi, A.R., Verbauwhede, I., Wachsmann, C.: PUFs: myth, fact or busted? A security evaluation of physically unclonable functions (PUFs) poured in silicon. In: CHES 2012. LNCS, vol. 7428, pp. 283–301. Springer, Heidelberg (2012)
28. Kulseng, L., Yu, Z., Wei, Y., Guan, Y.: Lightweight mutual authentication and ownership transfer for RFID systems. In: INFOCOM 2010. pp. 1–5. IEEE (2010)
29. Lee, J.W., Lim D., Gassend, B., Suh, G.E., Dijk, M. Devadas, S.: A technique to build a secret key in integrated circuits for identification and authentication applications. In: VLSI Circuits 2004. pp. 176–179. IEEE (2004)
30. Leest, V., Preneel, B., Sluls E.: Soft decision error correction for compact memory-based PUFs using a single enrollment. In: CHES 2012. LNCS, vol. 7428, pp. 268–282. Springer, Heidelberg (2012)
31. Ma, C., Li, Y., Deng, R.H., Li, T.: RFID privacy: Relation between two notions, minimal condition, and efficient construction. In: ACMCCS 2009, pp. 54–65. ACM (2009)
32. Moriyama, D., Ohkubo, M., Matsuo, S.: Relations among notions of privacy for RFID authentication protocols. In: Foresti, S., Yung, M., Martinelli, F. (eds.) ESORICS 2012, LNCS, vol. 7459, pp. 661–678. Springer Heidelberg (2012)
33. Mobile Trusted Module (MTM) Specifications, May 2009.
34. Medwed, M., Standaert, F.-X.: Extractors against side-channel attacks: weak or strong?. Journal of Cryptographic Engineering. 1(3), 231–241. Springer, Heidelberg (2011)
35. Maes, R., Tuyls, P., Verbauwhede, I.: Low-overhead implementation of a soft decision helper data algorithm for SRAM PUFs. In: CHES 2009. LNCS, vol. 5747, pp. 332–347. Springer, Heidelberg (2009)
36. Nedospasov, D., Seifert, J.-P., Helfmeier, C., Boit, C.: Invasive PUF Analysis. In: FDTC 2013. pp. 30–38. IEEE (2013)
37. Ng, C.Y., Susilo, W., Mu, Y., Safavi-Naini, R.: New privacy results on synchronized RFID authentication protocols against tag tracing. In: ESORICS 2009. LNCS, vol. 5789, pp. 321–336. Springer, Heidelberg (2009)
38. Ouafi, K., Phan, R.C.W.: Traceable privacy of recent provably-secure RFID protocols. In: ACNS 2008. LNCS, vol. 5037, pp. 479–489. Springer Heidelberg (2008)
39. Ostrovsky, R., Scafuro, A., Visconti, I., Wadia, A.: Universally composable secure computation with (malicious) physically uncloneable functions. In: EUROCRYPT 2013. LNCS, vol. 7881, pp. 702–718. Springer Heidelberg (2013)
40. Oren, Y., Sadeghi, A.-R., Wachsmann, C.: On the Effectiveness of the Remanence Decay Side-Channel to Clone Memory-Based PUFs. In: CHES 2013. LNCS, vol. 8086, pp. 107–125. Springer Heidelberg (2013)
41. Paise, R.I., Vaudenay, S.: Mutual authentication in RFID. In: ASIACCS 2008. pp. 292–299. ACM (2008)
42. Rührmair, U., Dijk, M.: Practical security analysis of PUF-based two-player protocols. In: CHES 2012. LNCS, vol. 7428, pp. 251–267. Springer, Heidelberg (2012)
43. Rührmair, U., Sölter, J., Sehnke, F.: On the Foundations of Physical Unclonable Functions. ePrint Archive, 2009/277 (2009).

44. Rührmair, U., Sehnke, F., Sölter, J., Dror, G., Devadas, S., Schmidhuber, J.: Modeling attacks on physical unclonable functions. In: ACMCCS 2010. pp. 237–249. ACM (2010)
45. Su, Y., Holleman, J., Otis, B.: A digital 1.6 pJ/bit chip identification circuit using process variations. J. Solid-State Circuits 32(1), pp. 69–77. IEEE (2007)
46. Sadeghi, A.R., Nacchache, D (eds).: Towards hardware-intrinsic security. Springer, Heidelberg (2010)
47. Sadeghi, A.R., Visconti, I., Wachsmann, C.: PUF-enhanced RFID security and privacy. In: SECSI 2010. (2010)
48. TPM Main Specification, Version 1.2, February 2005.
49. Vaudenay, S.: On privacy models for RFID. In: ASIACRYPT 2007. LNCS, vol. 4833, pp. 68–87. Springer, Heidelberg (2007)
50. Van Herrewege, A., Katzenbeisser, S., Maes, R., Peeters, R., Sadeghi, A.R., Verbauwhede, I., Wachsmann, C.: Reverse fuzzy extractors: enabling lightweight mutual authentication for PUF-enabled RFIDs. In: FC 2012. LNCS, vol. 7397, pp. 374–389. Springer Heidelberg (2012)