A PUF-based Authentication Protocol to Address Ticket-Switching of RFID-tagged Items

Sjouke Mauw[1], Selwyn Piramuthu[2,3,*]

[1]Computer Science and Communications, University of Luxembourg, Luxembourg
[2]Information Systems and Operations Management, University of Florida, USA
[3]RFID European Lab, Paris, France
[*]*corresponding author: selwyn@ufl.edu*

Abstract. Ticket-switching incidents where customers switch the price tag or bar code in order to pay a lower amount for their 'purchased item' is not uncommon in retail stores. Since the item has to pass through a check-out counter before leaving the store, it has a (even if miniscule) positive probability of being identified. However, when item-level RFID tags are used in an automated check-out environment, the probability of such incidents coming to light is estimated to be almost zero. We propose an authentication protocol for this scenario using a pair of item-level RFID tags, one of which is PUF-enabled to resist cloning attacks.

Key words: RFID authentication protocol, ticket-switching, PUF

1 Introduction

Incidents where a customer switches the price tag on an expensive item with that from a relatively 'cheap' item are not new. It is generally assumed that only a small fraction of such incidents are identified. With the advent and widespread use of bar codes, this same behavior has translated to switching bar codes between items or affixing a bar code from a relatively inexpensive item on an expensive item. For example, recently (May 8, 2012), a customer at a San Francisco Bay Area Target store was caught affixing false (home-printed) bar codes (i.e., "ticket switching") to packages of LEGOs that allowed him to purchase expensive sets at substantial discounts ([20], [21]). Similar incidents have occurred in other countries where items at retail stores have been the subject of such attacks. In May 2011, a 23-year-old was caught in a Leclerc supermarket in Trélisssac, Dordogne, for replacing the labels on two €2,300 bottles of *Petrus* with €2.50 labels[23]. In a majority of such incidences, the person involved is caught by a vigilant (usually, check-out) person at the store. The increase in the number of cases where item-level RFID tags are used in retail stores (e.g., Trasluz, American Apparel) and related automation of processes such as inventory management, check-out, among others has the potential to exacerbate this situation due to reduced human interaction in the process.

Since bar codes represent information only at the class-level, it is relatively easy to get away with a switch of bar codes between two items or replace bar code on expensive item with a 'home-made' bar code corresponding to a cheaper item. However, with instance-level information stored in item-level RFID tags, it becomes somewhat difficult since 'home-made' RFID tags need to contain an extensive set of information that is read and authenticated by the store check-out system. The ticket-switcher also needs to ensure (perhaps, with an RFID-shield) that the tag does not trigger an alarm (e.g., by RFID gates at the store entrance) when it is brought into the store. Moreover, unlike *physically* switching price tags or bar codes that are affixed or printed on the item, it is relatively difficult for the ticket-switcher to deal with RFID tags since these tags could be embedded in the item. When the RFID tag is affixed on the item (possibly under a bar code sticker), it is relatively physically easy to replace it with another tag given necessary skills and resources. With sufficient effort, it is not impossible to replace RFID tag(s) in/on an expensive item with that from a 'cheap' item in the retail store. Even worse, it is not inconceivable to destroy the RFID tag(s) in/on an item and just walk away with this item in an automated check-out retail store environment. However, RFID tags can be used in combination with existing loss-prevention measures (e.g., ink tags) to ensure that this eventuality does not occur or at least significantly reduce its occurrence probability. In an automated check-out retail store setting, the damage is done once the RFID tag "ticket-switching" has occurred. There is, therefore, a need to address this vulnerability before item-level RFID tags become ubiquitous in retail settings.

There is an extensive set of literature that deals with the counterfeiting prevention, in applications that involve expensive or critical item as well as those that involve pharmaceuticals, through RFID tags. There is also a growing set of literature on tampering of RFID tags. For example, Gandino et al. [8] provide an overview of risks and defenses of tampered RFID tags. Researchers have proposed several means to address tampering RFID tags, including watermarking and the use of Physically Unclonable Functions (PUFs). In general, watermarking RFID tags involves placing unique identification information in the Object Class (OC) and/or the EPC Manager (EM) fields. The watermark information thus stored are retrieved later for physical authentication. PUFs, on the other hand, are hardware-dependent and are generated using variations that are introduced during manufacture of the RFID device.

To summarize, related challenges in a retail store check-out environment include the ability to (a) flag an item that has never been on sale at this retail store (i.e., identify 'home-made' tags or tags that are not from this retail store), (b) identify duplicates (i.e., when an item's RFID tag is cloned or when an item with this tag was already 'checked-out' from this store), and (c) recognize when an item has a false tag (i.e., mismatch between item and its tag). Among these, (a) can be addressed through item authentication, (b) can be addressed through authentication as well as records from the store's information system (e.g., inventory, check-out), and (c) can be addressed through random physical checks for matching of item and its entry in the receipt or through cloning-

prevention. Cloning or tag-impersonation, its lighter version, can sometimes be easily accomplished by communicating with the tag and capturing necessary responses and appropriately replaying them to the system. We use PUF-enabled manufacturer-placed tag as part of the mechanism to dissuade cloning attacks.

To our knowledge, no published authentication protocol addresses the issue of "ticket-switching" item-level RFID tags in a retail setting. We attempt to fill this gap by considering watermarking-based and PUF-based RFID tags as possible contenders for this purpose. There are instances (e.g., Target stores in the U.S.) where the retail store is known to use its own bar code information that is different from that generated by the item's manufacturer. Unlike this scenario, we use *both* the manufacturer- and retailer- generated information to authenticate an item. Our rationale for two tags is two-fold: (1) a manufacturer-placed RFID tag inside the sealed item package would render it difficult to counterfeit the item as well as tamper with the tag and (2) the retailer-placed tag would allow for more retailer flexibility since this would be independent of any constraints (e.g., storage space, write-protection) in the manufacturer-placed tag. Moreover, since we are interested in RFID-tagged items, we assume the presence of a manufacturer-placed PUF-enabled RFID tag embedded in the item to prevent *cloning* and a retailer-placed passive RFID tag placed on the item. We authenticate the *simultaneous* presence of both these tags with the item of interest. Although the general idea for ensuring the simultaneous presence of two tags in the field of the reader is similar to that of *yoking proof* [11] and its variants, our protocol is structurally different and we consider a PUF-enabled manufacturer-placed RFID tag to decrease the opportunity for tampering/switching.

The rest of the paper is organized as follows: We first review a few selected related publications on watermarking and PUF-based RFID tags in the next section. We briefly discuss our system model in Section 3. We then present our PUF-based authentication protocol in Section 4 and evaluate its security properties in Section 5. We conclude the paper with a brief discussion in Section 6.

2 Related Literature

We first discuss a few related publications on watermarking RFID tags followed by those on PUF-enabled RFID tags.

2.1 Watermarking RFID Tags

Digital watermarking, a form of steganography, is a passive protection tool that helps hide information that is meant to be accessible only to authorized parties. The 'watermark' thus placed are robust against modification and are generally not encrypted - securing information about its very existence is therefore of paramount importance. The non-volatile location in an RFID tag where watermarking information is placed is known only to legitimate readers and back-end systems since these are generally not password-protected. Since not all RFID

tags have watermark information, even its very existence is known only to legitimate entities that are validated to have access to this information. When the existence and location of this information is exactly known, an adversary can easily have access to this information. There are also constraints on the length of the watermarks due to the extremely limited non-volatile space in low-cost RFID tags that are generally used in retail applications.

Potdar and Chang [16] propose TamDetect, an 8-bit watermark and a parity bit, which is a one-way hash generated from data stored in the EPC Manager (EM) and Object Class (OC) partitions. The watermark is generated by the RFID reader or RFID middleware and is embedded in the serial number partition. This can then be used to detect if data tampering has occurred in the RFID tag. Since the proposed watermarking is based on a secret function, Gandino et al. [8] claim that revelation of this function, either through an *insider attack* or otherwise, would necessitate a major modification of the system. External entities that later own this RFID-tagged entity cannot identify a tampered tag.

Noman et al. [14] develop a 32-bit watermark that they place in the reserve memory of the 32-bit *Kill* password. They claim that this can be used in applications where this *Kill* function is unused. Similar to Potdar and Chang, Noman et al. use padded values from EM and OC partitions to generate the watermark.

Yamamoto et al. [27] propose another means to watermark RFID tags. However, Gandino et al. [8] observe that this method has several drawbacks including the requirement of large memory, long transmission time for tamper checking, and its limited applicability.

Curran et al. [4] use a one-dimensional chaotic map, called the Skew Tent map, to randomly choose 6 bit positions from the OC (24-bit) field and the SN (36-bit) field to embed the watermark for the EM and the OC fields respectively.

While watermarking can be used to identify the occurrence of tampering, it is of not much help when other types of attacks (tag switching) occur.

2.2 PUF-based RFID Tags

After extensive review of existing literature on PUFs, Maes and Verbauwhede [13] conclude that PUF is not a "rigorously defined concept, but a collection of functional constructions which meet a number of naturally identifiable qualities such as uniqueness, physical unclonability and possibly tamper evidence." The implementation or use of PUFs involve the generation of a set of challenge-response pairs (CRPs) where a *challenge* to the PUF results in a *response* from the PUF. Several such CRPs are generated during the *enrollment* phase and stored in a CRP database. At a later point in time, during *verification*, the response from the PUF for a chosen challenge from this CRP database is compared with the corresponding response in the CRP database. Due to variations related to ambient conditions (e.g., temperature) as well as hardware, the response from the PUF may not exactly match that in the CRP database. However, a match within acceptable tolerance level is deemed to be sufficient.

A majority of PUF-based authentication schemes of RFID tags (e.g., [3], [5]) are not scalable since they use a pre-recorded set of challenge-response pairs

which run out after a deterministic number of authentication protocol runs. A given challenge-response pair cannot be reused since this facilitates denial-of-service attack. Another issue with PUF-generated challenge-response is that a given challenge does not result in the exact same response due to noise or hardware-related variations. This inconsistent behavior renders it difficult for the response from a PUF-enabled device to be directly used for authentication purposes. To alleviate problems associated with these issues, some researchers (e.g., [6]) use output from PUF along with *helper data* to generate keys.

Several types of PUFs have been studied by researchers including optical PUF (e.g., [15]), coating and acoustic PUFs (e.g., [22], [24]), silicon PUFs (e.g., [5]), among others. Maes and Verbauwhede [13] and Armknecht et al. [1] provide excellent overviews of PUFs.

Bolotnyy and Robins [3] propose a hardware-based approach to RFID security that relies on PUFs. They develop algorithms for key generation, tagging, and verification using PUFs and discuss the benefits of PUFs vs. hash functions for low-cost RFID tags. They note that PUFs are more resistant to side-channel attacks and physical tampering while being difficult to quantify since PUFs rely on physical characteristics that are also difficult to replicate. Bolotnyy and Robins claim that PUFs require about an order of magnitude less in terms of the number of gates required for computing hash functions. However, they require a sequence of PUF values to be stored on the already space-constrained tag.

Bassil et al. [2] propose a PUF-based mutual authentication protocol. However, their protocol is vulnerable and has several errors. For example, the secret value of the tag (SVT) is generated by the tag as PUF(random number) and the secret value of the reader (SVR) is generated by the reader as PUF(SVT). However, SVT, which is a *secret* is sent in the open. Moreover, the other terms that are *encrypted* can be easily recovered by an adversary. For example, they derive $A \leftarrow SVT \oplus SVR \oplus n_1$, $B \leftarrow Rot(SVR \oplus n_2, SVT)$, and $C \leftarrow Rot(SVT \oplus SVR \oplus n_1, n_2)$ and send $A||B||C$ from reader to tag. Now, using A and C, it is easy to know n_2; knowing n_2 and SVT, it is easy to know SVR using B; knowing SVT, SVR, and A, it is easy to solve for n_1. With this knowledge, the rest of the *secret* terms (i.e., SVT_{new}, SVR_{new}) can be determined. What is not mentioned in the paper is that the reproducibility of response, $PUF(x)$, may not be reliable for a given x on multiple invocations (e.g., [3]) and this could render authentication difficult.

Rührmair et al. [18] study challenge-response pairs for a few different (e.g., standard Arbiter, Ring Oscillator of arbitrary size, XOR Arbiter, Lightweight Secure, Feed-Forward Arbiter) PUFs and show using machine learning techniques that these PUFs can be impersonated and, therefore, cloned. Their results indicate that nonlinearities and larger bit-lengths as well as optical strong PUFs add mode complexity and are difficult to clone.

Sadeghi et al. [19] propose an authentication protocol using a PUF-enabled device which was later shown by Kardas et al. [12] to be vulnerable to a *cold boot attack* [9]. Kardas et al. [12] then present a means to thwart this attack

with two keys that are consecutively generated by the same PUF-enabled device with the claim that only one of these will be revealed in a cold boot attack scenario. However, they do not present any details of their modified protocol. The destructive-private PUF-based RFID protocol presented in Sadeghi et al. [19] is also vulnerable to impersonation of the reader to the tag by an adversary. The adversary can observe a round of the protocol for a given tag of interest and note its ID, which is sent in the open when the tag is authenticated. Later, when the adversary wants to authenticate itself as a reader to this tag, a random number (i.e., a) can be sent to the tag, which replies with (b, c) and the adversary can reply to this with the tag's ID.

Kardas et al. [12] also present a distance bounding protocol based on PUFs with a fast and a slow phase and use three registers (v_1, v_2, v_3) to accomplish the fast phase. Since these three registers do not include any prover-specific information that can be replayed or used to retrieve any prover-specific information, they are easily shared by a dishonest prover with an accomplice to carry out a *terrorist fraud attack* [17].

Van Herrewege et al. [26] propose a PUF-enabled lightweight mutual authentication protocol using reverse fuzzy extractors for compact and fast implementations of secure sketches and fuzzy extractors. As opposed to the typical use of fuzzy extractors where the computationally intensive *reproduction phase* is implemented in the PUF-enabled device, they implement the *helper data generation phase* on the PUF-enabled device and move the reproduction phase to the verifier. In their mutual authentication protocol, they send the tag's identifier (i.e., ID) in the open and this can be used by an adversary to *track* the tag. The adversary can begin this protocol by sending *auth* to the tag, which will reply with its ID. The adversary does not have to continue with the rest of the protocol since the tag is now uniquely identified to be present at that location.

Based on our review of existing literature, we observe the absence of published research that specifically addresses ticket-switching. Resistance to ticket-switching requires rendering it difficult to (a) remove the price tag, bar code, or RFID tag from the 'cheap' item and (b) somehow affix/embed this (or a cloned) price tag, bar code, or RFID tag on the desired item, while ensuring that (a) the items or their packaging are not damaged and (b) their identification by the retail store 'system' (e.g., store check-out person or automated check-out system) is not compromised in the process. Although there are publications that are tangentially related such as those on yoking-proof and its variants and those that address tampering tags, we are unaware of any that directly attempts to address ticket-switching.

3 System Model

3.1 Principals

Our system comprises the following four principals: *customers, RFID-tagged items, manufacturers*, and *retailer* (Figure 1).

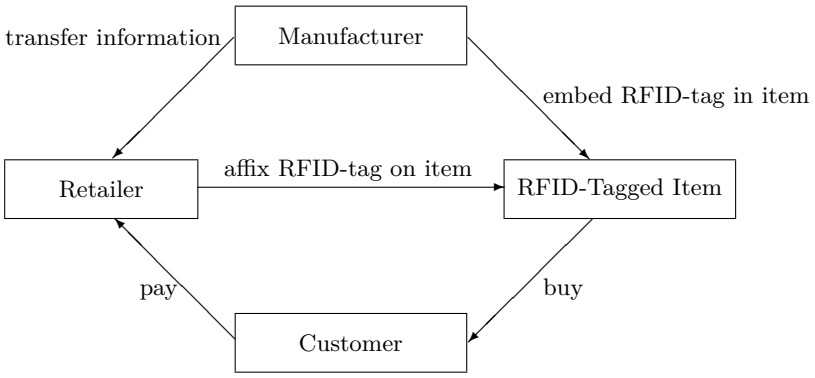

Fig. 1. The relationship among principals

The RFID-tagged item contains two RFID tags: a PUF-enabled passive tag that is embedded in the item by the manufacturer and a passive tag that is affixed on the item by the retailer. When the customer buys the item, the item is scanned (either manually by a check-out person or automatically by the automated check-out system) at check-out and both the tags are read by the reader. The unclonable PUF-enabled tag is primarily used to provide evidence of tag-tampering or cloning while the retailer-affixed tag is primarily used to facilitate in-store processes such as inventory management and automated-checkout as well as reduce shrinkage due to theft, misplacement and processing errors. Together, the two tags help authenticate the tagged item and provide proof that the item is what the honest manufacturer claims it to be and that no one has tampered with its content from the time it was packaged.

3.2 Adversary Model

Based on its environment (i.e., a retail store), possible threats to the authenticity of the RFID-tagged item can come from manipulated manufacturer-placed tag or retailer-placed tag or both. Other forms of threats include any form of attack (e.g., replay attack) that would enable a dishonest 'customer' to pay less for the item than its retailer-intended price.

3.3 Assumptions

We assume honest manufacturer and retailer who are there for the long run. These manufacturers and retailers will not tamper with the tags nor switch tags, which is the subject of this study, since (1) they face penalties from regulatory agencies when the items they sell are not what they claim them to be and (2) there is no incentive for the retailer or the manufacturer to manipulate the tags.

We assume the adversary ($\mathcal{A}$) to follow the Dolev-Yao intruder model [7]. The adversary $\mathcal{A}$ has complete control over the communication between the RFID-tagged item and the retail store system whereby $\mathcal{A}$ can eavesdrop, block, modify,

and inject messages anytime from/to any entity. However, appropriate keys are required to decrypt encrypted messages and the response from PUF for a given challenge cannot be exactly determined.

We are interested in ensuring that the customer pays exactly the retailer-set price and receives the exact item as promised in the transaction. Reduction in price due to discounts (e.g., coupons) is irrelevant for this study. We also do not consider the scenario where the retailer erroneously enters a wrong price for an item in the system.

3.4 Security Properties

The proposed protocol should guarantee the following security properties:

Correctness: The customer pays exactly the retailer-set price for an item.

Traceability: If a PUF-enabled tag is switched, its origin (i.e., the item in which it was embedded by its manufacturer) can be readily identified.

Accountability: When ticket-switching occurs, it is relatively easy to determine *when* and *where* a PUF-enabled tag or the other tag was switched (i.e., when separation of the tags from an item of interest occurred). This information can be used to immediately identify and catch the ticket-switcher in the act.

4 The Proposed Authentication Protocol

We want our authentication protocol to accomplish the following:

- Provide tamper-evidence
- Provide proof that the item is what it claims to be
- Allow for the retailer to place necessary information on the item - it's relatively easier with a retailer-placed tag
- Unclonable key

Tamper-resistance is provided by the unclonable (PUF) key, which is also *unclonable*. We allow for the manufacturer and retailer to place the information they deem necessary in separate tags. The manufacturer-placed tag is embedded in the item and inaccessible from the outside (i.e., it is not placed outside the item's packaging) and the retailer-placed tag is affixed on the item's packaging. While the retailer-placed tag can be switched with ease, the manufacturer-placed tag cannot be switched without damage to at least the packaging of the item.

As mentioned in Section 1, we assume the presence of two passive RFID tags for each item - a PUF-enabled tag that is embedded in the item by the manufacturer and another tag affixed on the item by the retailer. We use some of the guidelines from a recently proposed PUF-enabled mutual authentication protocol by van Herrewege et al. [26] in developing our authentication protocol. A PUF's response is different each time it's queried with the same challenge since it

depends on both this challenge and any device- and ambient-condition- specific variations. However, a large number of PUF-based applications require reliability in its response and *fuzzy extractors* [6] are typically used along with PUFs for this purpose. *Secure sketch* in fuzzy extractor maps similar responses (i.e., based on the same *challenge*) to the same value while a *randomness extractor* extracts full-entropy bit-strings from a partially random source. Secure sketch first generates ($Gen()$) helper data (h) from the PUF's response ($h_{mi} = Gen(R'_{mi})$) to a challenge (say, C_{mi}) and this helper data is used later to recover the PUF's noisy response (R'_{mi}) from its *true response* ($R'_{mi} = Rep(R_{mi}, h)$). We follow van Herrewege et al.'s recommendation to place the computationally intensive reproduction phase, Rep(), on the verifier and the efficient helper data generation phase, Gen(), on the PUF-enabled device. For a detailed and excellent description of $Gen()$, $Rep()$, fuzzy extractor, randomness extractor, and secure sketch, the reader is referred to [6].

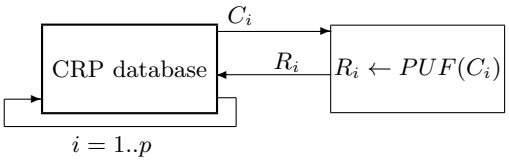

Fig. 2. PUF: Enrollment Phase

We borrow the essence of PUF-based model as presented in [26] to develop our authentication protocol. Specifically, we use the two stages from [26] as represented in Figures 2&3. Challenge-response pairs (CRPs) are generated during the enrollment stage, by repeatedly sending different challenges (C_i, where $i = 1..p$) to the tag's PUF and storing its response after 'error-correction' through helper data, and stored in the CRP database.

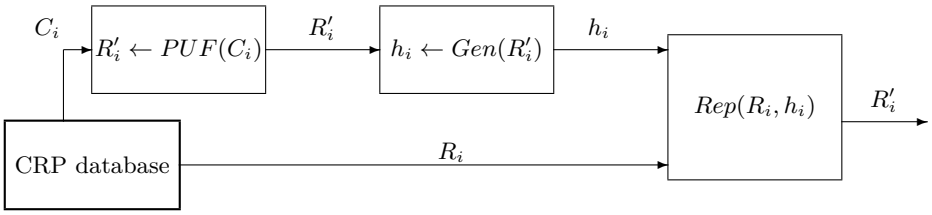

Fig. 3. PUF: Verification Phase

During the verification phase, a challenge (C_i) is chosen at random from the CRP database and sent to the tag. The tag's PUF takes this challenge as input and generates its response ($R'_i \leftarrow PUF(C_i)$). This response is then used by the tag to generate its corresponding helper ($h_i \leftarrow Gen(R'_i)$) data.

Along with the response (R_i) stored in the CRP database for this challenge, the helper data (h_i) can be used to generated the 'noisy' output from the tag's PUF ($R'_i \leftarrow Rep(R_i, h_i)$). We use the following notations:

- s_m, s_r, s_v, s'_v: l-bit nonce
- k_m, k_r: manufacturer- and retailer- placed tag's shared secret keys
- ID_m, ID_r: (manufacturer- and retailer- placed) tag IDs
- C_{mi}: manufacturer tag's i^{th} PUF challenge, where $i = 1 \cdots p$
- R_{mi}: PUF's *true* (i.e., noise-corrected) response to C_{mi}
- R'_{mi}: PUF's (noisy) response to C_{mi}
- t_{ms}, t_{me}: challenge-response start and end times for manufacturer tag
- t_{rs}, t_{re}: challenge-response start and end times for retailer tag
- Δ_m, Δ_r: 'round-trip' times to manufacturer- & retailer- placed tags
- f_k: keyed (with key k) encryption function
- $PUF(C_{mi})$: Physically Unclonable Function with C_{mi} as input
- h_{mi}: helper data[26] for the manufacturer-placed tag
- $Rep(R_{mi}, h_{mi})$: reproduction algorithm[26]
- $Gen(R_{mi})$: helper data generation algorithm[26]
- $auth$: authentication request
- $x \leftarrow y$: assign y's value to x

The proposed authentication protocol is presented in Figure 4. We assume that the verifier has access to the CRP database which contains previously recorded challenge-response pairs, represented by the set $\{C_{mi}, R_{mi}, ID_m\}$, for all PUFs. Since there are some similarities in what we intend to accomplish - i.e., *simultaneously* authenticate the two RFID tags as well as use information from a PUF-enabled RFID tag in a lightweight protocol to render ticket-switching difficult - and the purpose of protocols presented in [11] and [26], we borrowed ideas from both these papers as well as primitives from [6] (since several of these are used in [26]). Specifically, we use concepts from yoking-proof [11], reverse-fuzzy-extractors [26], and helper data [6].

The proposed protocol comprises two main timed 'components' that authenticate the manufacturer-applied and retailer-applied tags respectively. We ensure the simultaneous presence of these two tags in the item of interest through cryptography as well as by measuring the round-trip & processing time. The removal of either of these tags from the item would render the item inaccessible from the store's information system. We do not consider this possibility since this can be considered as a denial-of-service attack, which is outside the scope of this paper. We also note that the manufacturer-placed tag cannot be switched easily since switching it involves access to the inside of the sealed item that is inside its sealed package (e.g., a manufacturer-placed PUF-enabled RFID tag in a sealed package inside a sealed LEGO box). We try to accomplish this without revealing any secure information to a resourceful adversary.

We first authenticate the manufacturer-placed PUF-enabled RFID tag. To this end, the verifier first sends an authentication request along with a freshly-generated nonce (s_v). Upon receipt, the manufacturer-placed tag generates a fresh nonce (s_m) and encrypts this along with the verifier-generated nonce (s_v)

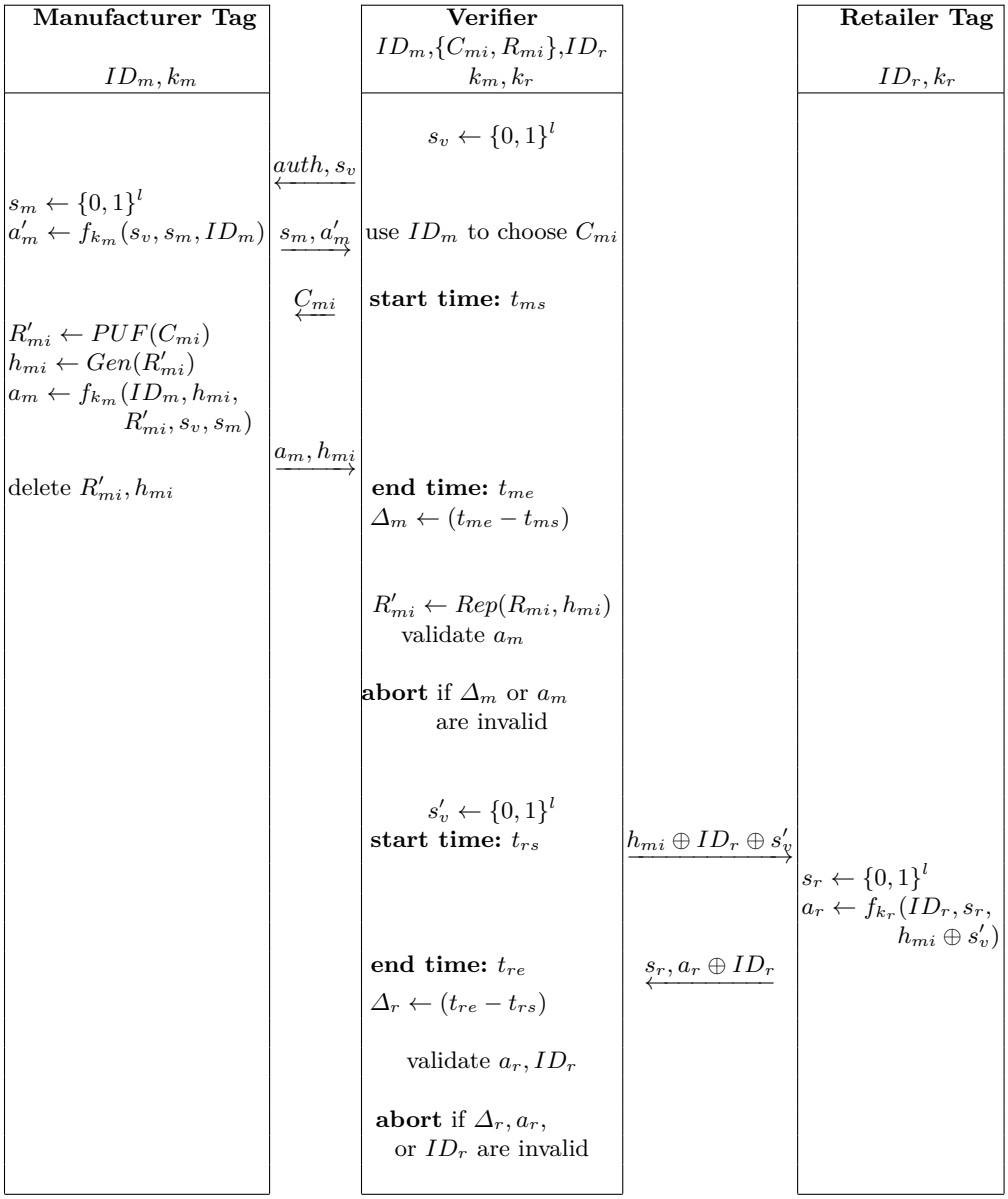

Fig. 4. The Proposed Protocol

and its ID (ID_m) and sends this encrypted value ($f_{k_m}(s_v, s_m, ID_m)$) to the verifier. The verifier decrypts the message and identifies the tag based on ID_m and randomly picks and sends a challenge (C_{mi}) for this tag from the CRP database to the manufacturer-placed tag. The verifier notes the time it takes for response from the manufacturer-placed tag. The measured response time is

then compared with the required (known) time for round-trip plus the manufacturer tag's computation time. Here, the computation time dominates since it can be multiple orders of magnitude (vs. pure round-trip time) depending on the complexity of computation and the RFID tag's processing power. When the response time from the tag is longer than expected, it more than likely signifies the tag location to be farther than expected, and authentication is aborted.

In response, the manufacturer-placed tag generates its PUF-response for the challenge and associated helper data and then computes (a_m) - a keyed encryption (using k_m) of $(ID_m, h_{mi}, R'_{mi}, s_v, s_m)$. The manufacturer-placed tag sends (h_{mi}, a_m) to the verifier. R'_{mi}, h_{mi} are deleted from the manufacturer-placed tag to prevent *cold boot* attack [9]. The verifier ensures that the response was received in reasonable time and validates the received a_m. Here, validation essentially involves verifying that $a_m = f_{k_m}(ID_m, h_{mi}, R'_{mi}, s_v, s_m)$. If at least one of these (i.e., Δ_m or a_m) is invalid, the verifier aborts the authentication protocol.

When the manufacturer-placed tag is successfully authenticated, the verifier proceeds to authenticate the associated retailer-placed tag. This part of the protocol is also timed by the verifier. Since this tag is not assumed to be PUF-enabled, this part of the protocol is structurally different. To ensure a link between the two components and their authentication sequence, the verifier sends h_{mi} from the manufacturer-placed tag to the retailer along with a freshly-generated nonce (i.e., s'_v) and the tag's unique identifier (ID_r). Upon receipt of this message, the retailer-placed tag generates a fresh nonce and then computes a keyed encryption (using k_r) of $(ID_r, h_{mi} \oplus s'_v, s_r)$. It then sends this encrypted term (a_r) along with its ID and nonce to the verifier, which then validates the 'round-trip' time as well as a_r, ID_r. Validation by the verifier involves unpacking a_r and ID_r from $a_r \oplus ID_r$ and verifying ID_r and ensuring that $a_r = f_{k_r}(ID_r, s_r, h_{mi} \oplus s'_v)$. Again, if at least one of these (i.e., a_r, ID_r, Δ_r) is invalid, the verifier aborts the protocol. Note that neither of the tags validate the verifier's message.

There are two minor concerns in the proposed authentication protocol that are fortunately easily explained, and are therefore not of any major significance specifically to the proposed authentication protocol. These include the ability of an adversary to block or even modify messages between two entities and the measurement of round-trip times.

Clearly, an adversary can readily block any message between any two entities and prevent successful authentication. In the proposed authentication protocol, an active adversary can easily modify a message that will then be accepted as valid by its recipient: $h_{mi} \oplus ID_r \oplus s'_v$ sent from verifier to retailer tag can be modified to $h_{mi} \oplus ID_r \oplus s'_v \oplus \delta$, where δ is some random number, and the retailer tag would now use $h_{mi} \oplus s'_v \oplus \delta$ instead of $h_{mi} \oplus s'_v$ to generate a_r and the authentication would fail. Instead of going through all this, an adversary can just block the message from the retailer tag to the verifier and easily send a random number as response to the verifier, which then will abort the protocol. This type of attack can be successfully mounted on *any* existing RFID authentication protocol. However, since the proposed authentication protocol does not involve

updates to any secret value during each protocol run that could potentially expose it to vulnerabilities associated with desynchronization or denial-of-service (DoS) attacks, we do not consider this eventuality.

While the round-trip times for messages between the verifier and each of the tags are individually bounded, the delay between querying these tags is not bounded. However, the verifier begins querying the retailer-placed tag if and as soon as the manufacturer-placed tag is verified and this minimal time delay to validate the response from the manufacturer-placed tag and to generate nonce s'_v would not allow for any type of attacks since (1) $(t_{rs} - t_{me})$ is negligible and therefore insufficient to, for example, detach and scan a tag separately and (2) the verifier has complete control over this in-between time period. One can question the very use of such time bounds since the separation possible (e.g., the entire length of a retail store) is relatively short for radio waves and therefore detecting location differences in such an environment necessitates extremely accurate (time-difference) measurements. However, we include this in our authentication protocol as an additional security measure. The use of round-trip times in RFID protocols is not new and it is commonly used, for example, in protocols that directly address relay attacks [10].

5 Security Properties and Analysis

Lemma 1. *The success probability of adversary $\mathcal{A}$ is bounded above by 2^{-2l}.*

Proof. For successful authentication, the adversary must be successful in submitting the exact information as expected by the verifier for simultaneous authentication of both the tags. Failure in either or both of these authentication stages would result in authentication failure for this item. There are two cases:
case a: The adversary can't successfully determine (h_{mi}, a_m) but knows $a_r \oplus ID_r$
case b: the adversary knows (h_{mi}, a_m) but not $a_r \oplus ID_r$

In *case a*, the probability of success is bounded above by 2^{-2l} when $\mathcal{A}$ has no information and guesses each of the $2l$ bits. Similarly, for *case b*, when $\mathcal{A}$ needs to guess all l bits, the probability of its success is bounded above by 2^{-l}. □

We first consider the security properties required of our protocol and the general setup and discuss its security properties.

Correctness: The retailer sets the price for each item that is for sale at the store and expects the customer to pay exactly that amount when an item is bought. Similarly, upon completion of the transaction, the customer expects to be charged the exact amount for which the item was offered for sale. Although the retail store may not be against the customer paying more, the customer certainly will not want to pay more. On the other hand, although the customer may not be against paying less, the retail store will not want to charge less. When a customer is charged less or more, the check-out system can be automated to raise a flag or trigger an alarm and appropriate action can then be taken.

Lemma 2. *The authentication protocol is correct if a pair of honest manufacturer and retailer tags are always successfully authenticated by a honest verifier.*

Proof. For tag authentication to succeed, both the manufacturer and the retailer tags simultaneously need to successfully authenticate themselves to the verifier.

For the manufacturer tag to successfully authenticate itself to the verifier [26], the following needs to be satisfied: $Rep(R_{mi}, Gen(R'_{mi})) = R'_{mi}$, $\forall(R_{mi}, R'_{mi})$. Note that a given challenge can detect up to d, the error correcting distance, and correct up to t ($t = \frac{(d-1)}{2}$) errors. The correctness property of *secure-sketch* [6] ensures that $Rep(R_{mi}, Gen(R'_{mi})) = R'_{mi}$ if distance$(R_{mi}, R'_{mi}) \leq t$. If the PUF generates responses of length l bits with a bit error rate of at most p, then probability(distance$(R_{mi}, R'_{mi}) \leq t) = binomial(t; l, p)$. The value of t is chosen such that this probability is very small and the *secure sketch* can recover R_{mi} from R'_{mi} with a very high probability.

For the retailer tag to successfully authenticate itself to the verifier, all it needs are h_{mi} and s'_v in addition to ID_r, s_r, k_r that it already knows. With this knowledge, it is guaranteed to successfully authenticate itself to the verifier. □

Traceability: When a tag is switched with a tag from another item at the store, the 'original' item where this tag was taken from can be readily identified when necessary since the store information system has details on the identity of the tags as well as their associated items. However, a cloned tag does not have this property since the original tag from which the clone was made may no longer be on sale at this store or could have been from a different store altogether. However, since the embedded PUF tag cannot be cloned, only the retail store-affixed tag can be replaced with a cloned tag. Moreover, replacing just one of the two tags is useless since the item cannot be switched with one of lower price.

When a certain type (say, model) of item gets targeted frequently for ticket-switching, the retail store can initiate a process for better packaging for this type of item that is somewhat more tamper-resistant.

Accountability: Since the reader(s) on the retailer shelves continually communicate with the tags and monitor their presence, it is relatively easy to identify separation of one of the tags from the other. Therefore, it is difficult to displace an item from its intended display shelf, switch tags, and fail to return the item to its original shelf or initiate check-out without the system noticing this discrepancy. When this discrepancy is noticed by the system, it can instantiate necessary events that would result in identifying and capturing the ticket-switcher.

Attacks such as side-channel attack and hardware-tampering attack are rather difficult to mount given the difficulty with which secret information from a PUF-enabled tag can be obtained without compromising the tag. Modeling attack is also difficult in the proposed protocol since we do not directly use the response from the PUF-enabled tag. Tag impersonation attack cannot be mounted due to the difficulty with which response from PUF can be predicted.

6 Discussion

RFID tags are increasingly being used to prevent counterfeiting of expensive items as well as pharmaceutical items. The assumption here is that the RFID tags thus employed cannot be tampered with nor cloned. Unfortunately, a resourceful adversary can clone or tamper a low-cost passive RFID tag with reasonable effort. Recent initiatives to thwart such incidences have used watermarking, PUFs, among others. We considered watermarking and PUFs and settled on using PUFs, due to the appropriateness of their general characteristics and beneficial properties for the addressed problem, to develop our authentication protocol. Based on necessary requirements to address ticket-switching behavior, including the ability to deter as well as recognize when ticket-switching occurs, and the requirements of retailers in being able to place as much information as is necessary on the tags, we opted to use two tags per item. This paper is an attempt at exploring the the use of PUF and yoking concepts as potential candidates as solution to the ticket-switching problem. The solution presented in this paper can and should be refined for better characteristics (e.g., less complex in terms of communication and computation, stronger security).

The proposed protocol accomplishes authentication of the tags and consequently the RFID-tagged entity by the verifier. This one-way authentication is sufficient in most retail store settings. However, there may be scenarios (e.g., when the item is in transit in a supply chain) where two-way authentication is necessary. We are currently working on extending our protocol to accommodate two-way or mutual authentication of both tag(s) and verifier.

References

1. Armknecht, F., R. Maes, A. Sadeghi, F. Standaert, and C. Wachsmann. 2011. A Formal Foundation for the Security Features of Physical Functions. *Proceedings of the IEEE Symposium on Security and Privacy*, 397-412.
2. Bassil, R., W. El-Beaino, W. Itani, A. Kayssi, A. Chehab. 2012. PUMAP: A PUF-Based Ultra-Lightweight Mutual-Authentication RFID Protocol. *International Journal of RFID Security and Cryptography*, 1(1/2), 58-66.
3. Bolotnyy, L., G. Robins. 2007. Physically Unclonable Function-based Security and Privacy in RFID Systems. *Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications (PERCOM)*, 211-220.
4. Curran, K., T. Lunney, A.N.M. Noman. 2011. Tamper Detection for Low Cost RFID Tags; Using Watermarking with Chaotic Mapping. *International Journal of Engineering and Technology*, 1(1), 27-32.
5. Devadas, S., E. Suh. S. Paral, R. Sowell, T. Ziola, V. Khandelwal. 2008. Design and Implementation of PUF-Based "Unclonable" RFID ICs for Anti-Counterfeiting and Security Applications. *Proceedings of the IEEE Int'l Conf. on RFID*, 58-64.
6. Dodis, Y., R. Ostrovsky, L. Reyzin, A. Smith. 2008. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *SIAM Journal of Computing*, 38(1), 97-139.
7. Dolev, D., A.C.-C. Yao. 1983. On the Security of Public Key Protocols. *IEEE Transactions on Information Theory.* 29(2), 198-207.

8. Gandino, F., B. Montrucchio, M. Rebaudengo. 2010. Tampering in RFID: A Survey on Risks and Defenses. *Mobile Networks and Applications*, 15(4), 502-516.
9. Halderman, J.A., S.D. Schoen, N. Heninger, W. Clarkson, W. Paul, J.A. Calandrino, A.J. Feldman, J. Appelbaum, E.W. Felten. 2009. Lest We Remember: Cold-boot Attacks on Encryption Keys. *Communications of the ACM* 52, 91-98.
10. Hancke, G.P., M.G. Kuhn. 2005. An RFID Distance-bounding Protocol. *Proceedings of IEEE/CreateNet SecureComm*, 67-73.
11. Juels, A. 2004. "Yoking-Proofs" for RFID Tags. *Proceedings of the International Workshop on Pervasive Computing and Communication Security*, 138-143.
12. Karda, S., M.S. Kiraz, M.A. Bingöl, H. Demirci, 2011. A Novel RFID Distance Bounding Protocol Based on Physically Unclonable Functions. *Proceedings of RFIDSec*, LNCS 7055, 78-93.
13. Maes, R., I. Verbauwhede. 2010. Physically Unclonable Functions: a Study on the State of the Art and Future Research Directions. *Towards Hardware-Intrinsic Security and Cryptology*, D. Naccache & A. Sadeghi (eds.), Springer, 3-38.
14. Noman, A.N.M., K. Curran, T. Lunney. 2010. A Watermarking Based Tamper Detection Solution for RFID Tags. *Proceedings of the International Conf. on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, 98-101.
15. Pappu, S.R. 2001. Physical One-Way Functions. *PhD Thesis*, MIT.
16. Potdar, V., E. Chang. 2006. Tamper Detection in RFID Tags using Fragile Watermarking. *Proceedings of the IEEE International Conference on Industrial Technology (ICIT)*, 2846-2852.
17. Reid, J., J. M. Gonzalez Nieto, T. Tang, and B. Senadji. 2007. Detecting Relay Attacks with Timing-based Protocols. *Proceedings of ASIACCS*, 204-213.
18. Rührmair, U., F. Sehnke, J. Sölter, G. Dror, S. Devadas, J. Schmidhuber. 2010. Modeling Attacks on Physical Unclonable Functions. *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS)*, 237-249.
19. Sadeghi, A.-R., I. Visconti, C. Wachsmann. 2010. PUF-Enhanced RFID Security and Privacy. *Secure Component and System Identification - SECSI*.
20. Schneier, B. 2008. Schneier on Security - UPC Switching Scam. http://www.schneier.com/blog/archives/2008/10/upc_switching_s.html
21. Schneier, B. 2012. Schneier on Security - Bar Code Switching. http://www.schneier.com/blog/archives/2012/05/bar_code_switch.html
22. Skoric, B. P. Tuyls, W. Ophey. 2005. Robust Key Extraction from Physical Unclonable Functions. *Proceedings of the Conference on Applied Cryptography and Network Security*, LNCS 3531.
23. Agence France Press. 2011. http://www.intothewine.fr/tags/trelissac-vin
24. Tuyls, P., B. Skoric, S. Stallinga, A. Akkermans, W. Ophey. 2005. Information Theoretical Security Analysis of Physical Unclonable Functions. *Proceedings of the Conference on Financial Cryptography and Data Security*, LNCS 3570.
25. Tuyls, P., L. Batina. 2006. RFID-Tags for Anti-Counterfeiting. *The Cryptographers' Track at the RSA Conference (CT-RSA)*, LNCS 3860.
26. Van Herrewege, A., S. Katzenbeisser, R. Maes, R. Peeters, A. Sadeghi, and I. Verbauwhede, C. Wachsmann. 2012. Reverse Fuzzy Extractors: Enabling Lightweight Mutual Authentication for PUF-enabled RFIDs. *Proceedings of the 16th International Conference on Financial Cryptography and Data Security*, LNCS, Springer.
27. Yamamoto, A., S. Suzuki, H. Hada, J. Mitsugi, J., F. Teraoka, O. Nakamura. 2008. A Tamper Detection Method for RFID Tag Data. *IEEE International Conference on RFID*, 51-57.