Università degli Studi di Roma Tre
Dipartimento di Informatica e Automazione
Via della Vasca Navale, 79 – 00146 Roma, Italy

Efficient and Practical Authentication of PUF-based RFID Tags

Pier Francesco Cortese[1], Francesco Gemmiti[2], Bernardo Palazzi[2],
Maurizio Pizzonia[2], and Massimo Rimondini[2]

RT-DIA-150-2009 **June 2009**

(1) Data Management S.p.A.
`pcortese@consul.datamanagement.it`
(2) Dipartimento di Informatica e Automazione
Università degli Studi Roma Tre
`{gemmiti,palazzi,pizzonia,rimondin}@dia.uniroma3.it`

This work is partially supported by the Italian Ministry of Research, Grant number RBIP06BZW8, FIRB project "Advanced tracking system in intermodal freight transportation". PUF-based tags were kindly supplied by Verayo, Inc.

ABSTRACT

RFID tags are typically exposed to the risk of cloning. A promising solution to this problem is represented by the introduction of Physical Unclonable Functions (PUF) within tags. In a typical usage scenario, a trusted entity builds a database of challenge-response pairs (CRPs), usually large to improve security. This database should be kept secret but securely shared with entities in charge of the authentication. Secure distribution of such a large amount of secret data makes this approach hard to adopt in industrial and commercial contexts.

We propose a new security architecture that reduces the amount of shared secret data to a constant size, regardless of the number of generated CRPs and tags to be handled. These constant-sized data can be stored and distributed using a secure hardware token, which can be easily implemented with currently available technology. The rest of the data can be released publicly over an insecure one-way communication channel. We discuss applicative scenarios and variations of our proposal, and support it with tests performed on pre-production PUF-based tags.

1 Introduction

RFID systems allow to track items, animals, and "people" very effectively and with limited human support. This is made easy by the capability of RFID tags to be read simultaneously, without being in the line of sight, through barriers, and possibly from a distance.

Tracking by RFIDs is extremely useful for luxury goods, pharmaceuticals, and all those items whose value is considerably greater than that of the tag itself. In this context, RFID systems are typically exposed to the risk of tag "cloning", either by reproducing tag contents or by tampering with the back-end item database. Cloned tags can be used to replace original goods with counterfeit copies. Many research works address this problem by proposing dedicated authentication protocols, usually requiring additional computational power on the tags. These approaches are either unfeasible using currently available technologies and standards, or too expensive to be appealing for real applications. The alternative of using lightweight authentication protocols may potentially introduce vulnerabilities in the authentication system. Basically, with those approaches, the tradeoff between cost and security is hard to solve.

In this paper we address the problem of efficiently authenticating RFID tags. We aim at providing a solution that is both robust against counterfeiting and cost-effective. Our approach takes advantage of tags based on Physical Unclonable Functions (*PUFs*). Ideally, these tags map an arbitrary input (*challenge*) to a different output (*response*), in such a way that querying the tag with the same challenge always returns the same response, and this response is unpredictable. Real PUF-based tags are implemented by exploiting unpredictable and hard to reproduce manufacturing anomalies, and therefore usually exhibit some error on the returned response [8].

Using PUF-based tags requires an enrollment phase in which tags are queried with a large number of challenges and the responses are recorded in a table of *challenge-response pairs* (*CRPs*) which is usually very large. To authenticate a tag, a challenge is picked from the CRP table and the corresponding response from the tag is compared with the one stored in the table. Confidentiality of the CRP table is crucial for the effectiveness of this approach. Considering the size of the table, fulfilling this requirement makes authentication of PUF-based tags hard to deploy in real scenarios.

In this paper we propose two models for the authentication of PUF-based tags. The first one exploits ideal tags, while the second is suitable to be implemented with PUF-based tags available from the industry. Our models do not require to store authentication-related data into tags and are suitable to be implemented using no on-line connection. In the model that is based on real tags we require the secure distribution of a constant amount of data, regardless of the number of generated CRPs and tags to be handled. These data can be easily stored in a secure hardware token available from current technology.

We also propose applicative scenarios for the model based on real PUF-based tags. We performed experiments to check, independently from the producer of the tags, the feasibility of the adoption of those tags in our approach.

The paper is organized as follows. In Section 2 we review the state of the art on RFID authentication. Some attacks to well-known authentication mechanisms based on PUF tags are described in Section 3. We propose two novel models for authentication in Section 4, separating the case for ideal PUF tags and real PUF tags. Practical applications of our approach are discussed in Section 5. The results of our experiments are presented in Section 6.

2 State of the Art

Security aspects have a significant practical relevance in logistics. In particular, transported goods need to be protected against counterfeit (replacement of an original product with a crafted imitation), usually achieved by cloning (duplication of a product's identifier aimed at counterfeiting or stealing the original product). Therefore, the authentication of RFID devices has attracted the interest of researchers.

Existing authentication mechanisms can provide one-way or mutual authentication between a reader and a tag, depending on the application requirements. These mechanisms usually require running a specifically designed protocol to exchange authentication information between the reader and the tag. Among the best known protocols there is the HB family of protocols (see, for example, [2, 3, 16]), which is based on the computational hardness of the problem of decoding a random linear code. Protocols that support mutual authentication between a reader and a tag are proposed in [15, 14]. In [5] the authors propose a transformation to turn a generic RFID authentication protocol into an equivalent one that has constant cost for the lookup of keys in the backend database. Counterfeit is prevented in [20] using an authentication scheme based on the RSA algorithm.

On the other hand, it has been proved [6] that RFID tags based on widely adopted technologies can be cloned. Apart from authentication mechanisms, countermeasures to this threat are often based on limiting access to information that identifies the tag, for example by protecting them with a password [10].

Resorting to authentication mechanisms has two drawbacks. First, they require some changes to standard protocols for RFID devices. Second, they require the availability of computational power (and volatile memory) on the tags. Passwords and other information used to protect tag identification information usually do not change over time, and are therefore much vulnerable.

In order to get over these shortcomings, a recent proposal [8] suggested the use of a cost-effective technology to implement clone-proof tags. The underlying idea is to encode authentication information in the form of a data transformation that is implemented using an non-reproduceable physical system. More precisely, a Physical Unclonable Function (PUF) [11] is a function that maps an arbitrary input (*challenge*) to a fixed output (*response*), such that:

1. it requires little time to be evaluated;
2. it is hard to characterize: namely, an attacker with a polynomial amount of resources and only aware of a polynomial amount of CRPs can extract a negligible amount of information about the function.

For these conditions to be guaranteed, the manufacturer of a PUF device must be unable to reconstruct the function and to produce identical PUFs with a polynomial amount of resources. Also, the PUF must be only accessible via an algorithm that is physically bound to the PUF itself, and any attempt to circumvent the algorithm will result in the destruction of the PUF.

PUFs can be implemented by exploiting the minimal random variations in a manufacturing process. These variations can stem from acoustic, optical, or electronic systems that apply unpredictable transformations to arbitrary input signals. This approach is adopted in [11, 8] to build PUF-based RFID tags based on silicon integrated circuits. The operation of such a tag is based on establishing two electrical circuits that depend on the input challenge. When traversed by a signal, the two circuits introduce a slightly (and unpredictably) different delay on that signal. A latch placed at the end of the circuits detects the one with lowest delay and returns a bit that indicates this. When queried to operate as a PUF, the tag gets the input challenge and activates these circuits multiple times in order to generate a sequence of bits (the response) as output. The PUF tags described in [8] support 128 bit challenge and responses, and are compliant with the ISO-14443 type A specification, so that they can still operate with a standard HF reader. Of course, different technologies (e.g., UHF) can be used to implement the same solution.

PUF tags can be used for authentication purposes as described in [8]. First of all, a trusted entity queries the tag with multiple challenges and collects the returned responses in a set of CRPs. When the tag needs to be authenticated, one or more challenges from this set are retrieved and submitted to the tag, which is successfully authenticated only if the obtained responses match the ones in the applicable CRPs. When queried with the same challenge, an ideal PUF tag should always return the same response. In practice, the response may differ in some of its bits, and a threshold on the number of matching bits [8] or an error correcting code [11] may be needed to verify its authenticity.

Other papers propose the design of protocols for the authentication of PUF tags based on public key cryptography [1] or able to support off-line authentication [18]. However, these solutions require some computational power or additional storage on the tags. The approach we propose is instead based on the bare PUF functionality. The authors of [4] show that PUFs can also be exploited to obfuscate shared keys in order to improve the resistance of a tag against physical attacks and improve their privacy.

3 Attacks to the Authentication of PUF-based Tags

In this paper we address the usage of RFID tags to prove that a certain item was not counterfeited. We now describe some attacks to which authentication schemes based on PUF-based tags are exposed. We assume that the attacker is polynomially bounded in time and space. Also, we assume that tags are embedded into the items so that attempting to detach a tag from an item results in destroying the tag (for example using techniques like those proposed in [13]). The entity that attaches tags to items is considered trusted. Under these assumptions, asserting the authenticity of an item is equivalent to asserting the authenticity of the tag itself.

RFID tags attached to goods transit through a number of untrusted operators (transport, stocking, etc.), thus being exposed to the risk of *cloning*. An attacker can use a cloned tag to replace an item with a counterfeit version. From the point of view of authentication, a cloned tag must simply produce the same authentication information as the original one, as this makes it actually impossible to distinguish a genuine tag from a cloned one.

Differently from standard RFID tags, PUF-based tags are claimed to be impossible to physically clone. However, their usage exposes to new kinds of attacks. First of all, using PUFs requires creating a table of CRPs, and care must be taken in keeping this table accessible only to the entity in charge of the authentication. In fact, if an attacker can access the CRPs generated during the enrollment, she can build a fake non-PUF tag that replies correctly to a known set of challenges. Such a "virtual" tag has the same radio interface of a standard RFID and a large amount of memory, where several CRPs can be stored. Further, if the challenges used during authentication are known (e.g., predicted or eavesdropped), the "virtual" tag can store just a small subset of the CRP table.

A similar attack can be performed if the attacker only knows the challenges in the CRP table and can physically intercept goods. In fact, the attacker can query the attached tags, collect the responses, and build fake "virtual" tags. Another possibility is that the attacker passively eavesdrops communications between a tag and a reader, thus collecting challenges and responses. This discourages re-use of challenges. Last, the attacker can impersonate the producer and authenticate a rogue freight transportation using the same security model (spoofing). The producer must therefore be somehow authenticated or trusted. Note that the usage of PUF-based tags completely defeats the physical cloning of a tag. However, since authentication is based on only one or a few challenges, the other attacks must still be considered.

Counterfeit can also be achieved by inserting "rogue" CRPs in the table. However, the integrity of this table can be checked with straightforward signing techniques in $O(n)$ time (n is the number of CRPs), or adopting an Authenticated Data Structure [17, 9] that allows to detect alterations in $O(\log n)$ time.

There is a further attack that appears really hard to avoid: when challenged by a reader, the "virtual" tag may act as a proxy, namely challenge a genuine tag (e.g., via a radio communication), get the response, and provide it back to the reader. In the rest of the paper, we assume the authentication process happens at a secure place and is protected by physical countermeasures (e.g., radio shields) against this kind of attack.

4 Novel Approaches to Authenticate PUF-based Tags

In this section we propose two different models for the authentication of PUF-based tags. The first model relies on cryptographic hash functions and is intended for use with ideal PUF-based tags which, when challenged, always reply consistently with the same value. The other one is intended for use with real PUF-based tags in which the PUF response is affected by an error. In the latter case the Hamming distance from the ideal response is a random variable with a normal distribution (see Section 6), and this makes it hard to use the hashing techniques we propose for the ideal case. Instead, with real PUF-based tags we rely on the use of symmetric cryptography.

Both models involve the following entities:

Tag. A PUF-based tag that, when queried with a challenge c, replies with a response r, computed using a function that is hard to physically clone.

Producer. An entity that generates a set of CRPs by repeatedly querying Tags.

Verifier. An entity that wants to verify the authenticity of Tag.

Producer and Verifier are both trusted and cooperate to authenticate Tag.

As other approaches known in the literature [18], we encompass two phases. In the *enrollment* phase, Producer challenges Tag in order to generate a set of CRPs. In the *authentication* phase, the information collected by Producer is transferred to Verifier and is used by the latter to authenticate Tag. Since the set of CRPs is usually large, our proposals focus on keeping the amount of data that should be securely transferred from Producer to Verifier as small as possible.

To simplify the notation, the models are described for a single tag. Extension to the general case is straightforward. In the following subsections we describe the two models and discuss their security and strengths.

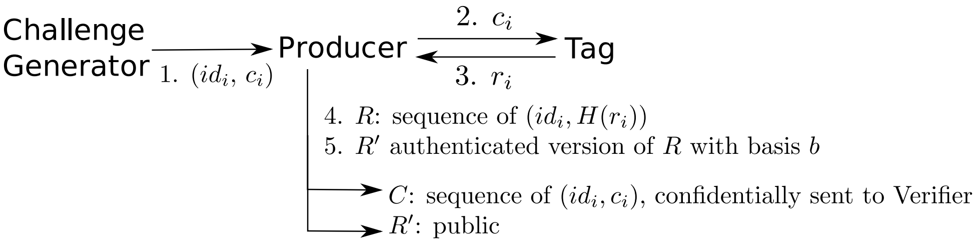

Challenge Generator
1. (id_i, c_i)
Producer
2. c_i
Tag
3. r_i
4. R: sequence of $(id_i, H(r_i))$
5. R' authenticated version of R with basis b
C: sequence of (id_i, c_i), confidentially sent to Verifier
R': public

Figure 1: Communication among entities during the enrollment of ideal Tags. The process takes place in a trusted environment.

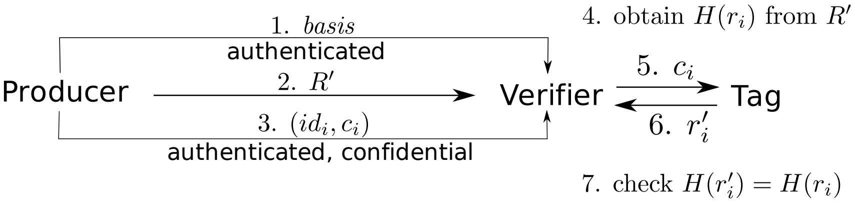

1. *basis*
authenticated
2. R'
3. (id_i, c_i)
authenticated, confidential
Producer
Verifier
4. obtain $H(r_i)$ from R'
5. c_i
Tag
6. r'_i
7. check $H(r'_i) = H(r_i)$

Figure 2: Communication among entities during the verification of ideal Tags.

4.1 Model for Ideal PUF-based Tags

In this model each CRP is represented by a triple $(id_i, c_i, H(r_i))$, where id_i is a unique identifier assigned to the triple, c_i is a challenge, r_i is Tag's response when queried with c_i, and H is a cryptographic hash function.

Fig. 1 shows the sequence of information exchanges among entities during the enrollment phase. The following operations are performed.

1. Challenges $c_1, \ldots, c_n$, and corresponding sequential IDs $id_1, \ldots, id_n$ are generated so that any c_i cannot be deduced by knowing any subset of $c_1, \ldots, c_{i-1}$. We call the sequence of pairs (id_i, c_i) *confidential sequence*, denoting it by C.

2. Producer queries Tag with each challenge c_i.

3. Tag replies with response r_i to each challenge c_i.

4. We call the sequence of pairs $(id_i, H(r_i))$ *public sequence* and denote it by R.

5. Using techniques from [17, 9], an Authenticated Data Structure (ADS) R' is computed from R, obtaining a value b (*basis*) of constant size, that represents the footprint of the entire sequence. The integrity and authenticity of b is sufficient to efficiently verify portions of an untrusted copy of R'.

The confidential sequence C, created at step 1, is arbitrarily chosen by Producer. Hence, it can be generated starting from a secret whose length does not depend on n (see Section 5). The data structure R' is meant to be public or communicated to Verifier via an untrusted channel, while C should be communicated to Verifier using a trusted channel.

Fig. 2 shows the sequence of information exchanges among entities in the verification phase. The following operations are performed.

1. Verifier receives b by means of a communication that ensures authenticity (integrity of data and origin).

2. Verifier can access an untrusted (public, non-authenticated) copy of R'.

3. Verifier periodically receives a new pair (id_i, c_i) from Producer using a communication that ensures authenticity and confidentiality.

4. Verifier accesses R' to get the value $H(r_i)$ corresponding to id_i and uses the ADS to verify that the result is valid against the basis b.

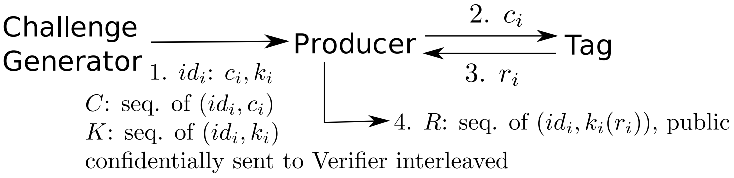

Challenge Generator
Producer
Tag
2. c_i
3. r_i
1. id_i: c_i, k_i
C: seq. of (id_i, c_i)
K: seq. of (id_i, k_i)
confidentially sent to Verifier interleaved
4. R: seq. of $(id_i, k_i(r_i))$, public

Figure 3: Communication among entities during the enrollment of real Tags. The process takes place in a trusted environment.

5. Verifier queries Tag with challenge c_i.

6. Verifier gets response r'_i from Tag.

7. Verifier checks whether $H(r'_i) = H(r_i)$. If this is the case, Tag is genuine.

8. Steps 3 to 7 are iterated as needed.

With this approach, once a challenge has been revealed to allow Verifier to perform authentication, it is not secure and should therefore not be used for other authentications. The frequency at which Verifier receives pairs (id_i, c_i) at Step 3 bounds the frequency at which authentications can be performed securely.

This model requires ideal Tags, since the results of cryptographic hash functions on noisy data are hard to compare (Step 7 of the verification).

4.2 Model for Real PUF-based Tags

We now describe an alternative model that provides a solution for real Tags where responses are affected by a noise error. In addition to supporting efficient and practical authentication, this proposal requires no on-line connection between Producer and Verifier.

The model exploits symmetric encryption (e.g., AES [7]) to communicate authentication information. In this model each CRP is represented by a 4-tuple $(id_i, c_i, k_i, k_i(r_i))$, where id_i is a unique identifier assigned to the 4-tuple, c_i is a challenge, r_i is the response of Tag when queried with c_i, and k_i is a symmetric key that encrypts r_i.

Fig. 3 shows the sequence of information exchanges among entities in the enrollment phase. In this phase the following operations are performed.

1. Challenges $c_1, \ldots, c_n$, keys $k_1, \ldots, k_n$, and sequential IDs $id_1, \ldots, id_n$ are generated. In doing so, the generator must ensure that any c_i cannot be deduced by knowing any subset of $c_1, \ldots, c_{i-1}$ and the same holds for k_i (a random number generator can be used for the purpose). The *confidential sequences* C and K are sequences of pairs (id_i, c_i) and (id_i, k_i), respectively.

2. Producer queries Tag with each challenge c_i.

3. Tag replies with response r_i to each challenge c_i.

4. The sequence of pairs $(id_i, k_i(r_i))$ is the *public sequence*, denoted by R.

Sequence R is meant to be public or communicated to Verifier via an untrusted channel. Sequences C and K are generated, one value at a time, by a trusted copy of the challenge generator at Verifier's place.

Before Verifier can authenticate Tags, Producer has to make R publicly available (and, in particular, accessible by Verifier) and provide Verifier with a trusted copy of the challenge generator (see Fig. 4(a)).

Fig. 4(b) shows the sequence of information exchanges among entities during verification. To authenticate a tag, the following operations are performed.

1. Verifier asks the challenge generator for the first unused pair (id_i, c_i).

2. Verifier gets challenge c_i from the challenge generator using a communication that ensures authenticity (integrity of data and origin) and confidentiality.

3. Verifier queries Tag with challenge c_i.

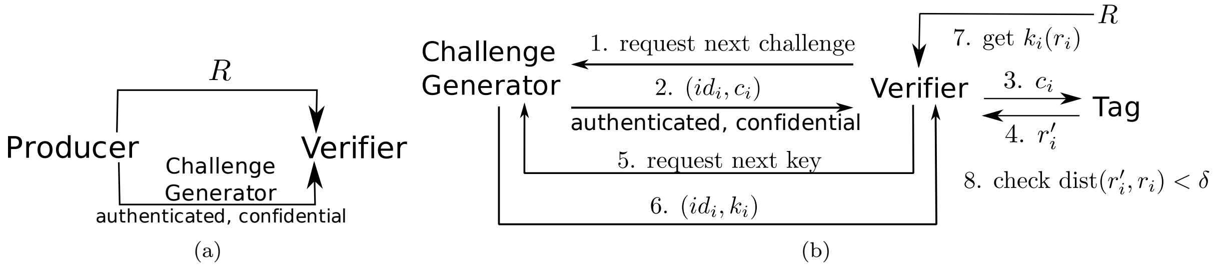


Figure 4: Information exchange before (a) and during (b) the verification of real Tags.

4. Verifier gets response r'_i from Tag.

5. Verifier asks the challenge generator for the next pair (id_i, k_i)

6. Verifier gets pair (id_i, k_i) from the challenge generator by means of an untrusted channel.

7. Verifier uses id_i to search R for the value $k_i(r_i)$ and deciphers it, obtaining r_i.

8. Verifier checks whether $\mathbf{dist}(r'_i, r_i) < \delta$, where **dist** is the Hamming distance and δ is a threshold. If this inequality holds, then Tag is genuine.

Pair (id_i, k_i) can be transferred over an untrusted channel because at step 4 Verifier has already collected all the responses it needs to authenticate Tag.

A possible variation of this model is to assign c_{i+1} the function of k_i. In this way there is only one sequence to manage. On the other hand, we introduce a synchronization among information used for different authentications. This synchronization can only be relaxed by wasting some challenges.

4.3 Security Analysis and Strengths of the Two Models

Both the models we propose are robust against cloning attacks. In fact, in order to clone the behavior of Tag during the authentication phase (see Section 3), an attacker needs to know in advance a response (for example by physically accessing Tag) and either the corresponding challenge or the time at which this challenge will be used to authenticate Tag. All these situations are hard for the attacker, because i) responses in the public sequence are protected either by a cryptographic hash function or by encryption, ii) future challenges are unpredictable, confidential before being released, and will never be used again for authentication. In both models we also rely on the fact that Verifier, after receiving a challenge, securely stores it before its use. We assume Verifier cannot be tampered with.

With respect to privacy issues, our proposal does not introduce any further drawbacks with respect to standard RFID tag systems.

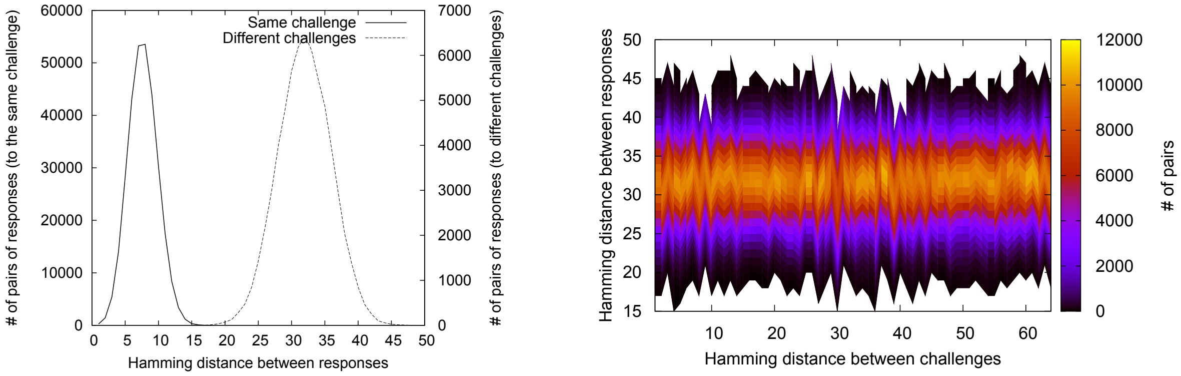
The model for ideal Tags has several strengths: it requires no encryption, the communication channel between Producer and Verifier can be one-way, and there is no secret that, if revealed alone, grants access to the whole database of CRPs. We stress that only Verifier can know the responses, by directly querying Tags. Unfortunately, this model is unfeasible with real Tags.

The strengths of the model for real Tags are: feasibility with Tags manufactured with currently available technologies [8], one-way exchange of information from Producer to Verifier before the authentication, and a constant amount of confidential information transferred from Producer to Verifier during the authentication. On the other hand, this model relies on encryption techniques and requires two-way communication during authentication.

5 Applicative Scenarios

In typical applicative scenarios it is likely to have many distinct verifiers and, of course, a large number of tags. Our models can be effectively applied within such contexts.

The role of Producer may be assumed by the Tags manufacturer or by another entity, e.g., a wholesale trader, a trademark manufacturer, or a reseller. Whichever the scenario, Producer must be considered



(a) Distribution of Hamming distances between responses.

(b) Correlation between the Hamming distance of challenges and responses.

Figure 5: How the response of a tag changes when queried with different sets of challenges.

trusted by all Verifiers. The enrollment requires the physical presence of the Tags, and hence must be completed when the Tags are still at the Producer's place. The costs of this process, which is proportional to the number of generated CRPs, should be accounted as the cost of the Tags. Along a supply chain there may be several entities playing the role of Producer. Different producers could generate their own sets of CRPs, thus supporting authentication by different Verifiers. Interestingly, if challenges are picked randomly, Producers and Verifiers may interleave.

In order to reduce communication and storage costs of secret data, it is possible and economically convenient to share the same confidential sequence for a large number of Tags. Ideally, all Tags released by Producer can be associated with the same confidential sequence. On the other hand, public sequences should be different for each Tag.

The confidential sequence can be generated by Producer without actually storing it, for example hashing a secret and a sequential number. With current technologies, these information can be embedded into a hardware token that has the same level of security of Producer, in a way similar to One Time Password (OTP) tokens [12]. This approach can be applied, with slight variations, both to ideal Tags and to real ones. If each Verifier receives a token that implements a different challenge generator, it does not need to trust the other verifiers and hacking a token only compromises the challenges generated by that token. To avoid re-use of a compromised token, a system of token identification and blacklist can be easily designed.

Another possible variation is to generate confidential sequences on the basis of a secret and a timestamp, the latter obtained from a *world master clock*. In this scenario, challenges are supposed to be used within a limited time window If this window is short enough, the delay between the last obtained challenge and the authentication is small enough to discourage attackers from performing replay attacks. Therefore, the duration of the time window is a tradeoff for the level of security. Depending on the application, the frequency at which challenges are published can be tuned, so that the right tradeoff between security and CRP generation cost is reached.

6 Experimental evaluation

In order to evaluate the applicability of the model described in Section 4.2, we performed experiments with a sample set of pre-production PUF-based tags VeraX512H capable of managing 64 bit challenges and responses, using a SkyeTek SR70 reader. The equipment has been kindly provided by Verayo [19].

The first experiment was aimed at determining the time needed to generate CRPs. We measured the time required to query PUF tags with 80 randomly chosen challenges and to collect the responses. Each query was repeated 1000 times. We observed that a single CRP can be generated in about 17ms in 96% of cases, anyway never more than 28ms. This time is not affected by the number of generated CRPs. Hence, generating a large set of CRPs for a tag can be done in a few hours (more than 200,000 CRPs per hour). Since each tag has a unique identifier, we argue that the process of generating CRPs could be further sped up by querying multiple tags in parallel, possibly using multiple readers.

In a second experiment we observed how the response of a single tag changes when queried with different challenges. More precisely, we first queried a tag 800 times with the same challenge and computed the Hamming distances between all possible pairs of returned responses. The distribution of the resulting distances is shown in the left part of Fig. 5(a) with solid stroke. The figure shows that the distances have a normal distribution centered on 7, and for almost all pairs the distance is less than 17 bits. This confirms the results from [8]. We then queried the tag with a set of 6400 pairs of challenges, each consisting of a random challenge c and another challenge at a fixed Hamming distance from c. We considered all the distance values between 1 and 64, generated 100 pairs for each distance, and submitted each pair to the tag 10 times, collecting 64000 pairs of responses. The distribution of the Hamming distances of these pairs of responses is displayed in the right part of Fig. 5(a) with dashed stroke. The plot shows that the distances have a normal distribution centered on 32, as theoretically expected for distances of random numbers of 64 bits, and for almost all pairs the distance is more than 17 bits. We conclude that 17 bits is a good threshold to authenticate a PUF tag without the risk of false positives. This value is also suggested by the producer of the tags [8].

In order to further verify the applicability of PUF tags for authentication, we reused the data set from the previous experiment to check for potential correlation between the distances of pairs of challenges and the distances of the corresponding responses. The plot in Fig. 5(b) illustrates the results. The X axis represents Hamming distances of each pair of challenges, the Y axis represents Hamming distances of the corresponding pair of responses, and the colors show the density (number of pairs of challenges and pairs of responses) in that point. It can be easily seen that there is no evident correlation between the two distances.

References

[1] L. Batina, J. Guajardo, T. Kerins, N. Mentens, P. Tuyls, and I. Verbauwhede. Public-key cryptography for RFID-tags. In *PerSec 2007*, pages 217–222, 2007.

[2] M. Blum and N. J. Hopper. A secure human-computer authentication scheme. Technical Report CMU-CS-00-139, School of Computer Science, CMU, 2000.

[3] J. Bringer, H. Chabanne, and Dottax E. HB^{++}: a lightweight authentication protocol secure against some attacks. In *SecPerU 2006*, 2006.

[4] J. Bringer, H. Chabanne, and T. Icart. Improved privacy of the tree-based hash protocols using physically unclonable function. In *SCN'08*, LNCS 5229. 77–91.

[5] M. Burmester, B. de Medeiros, and R. Motta. Robust, anonymous RFID authentication with constant key-lookup. Cryptology ePrint Archive, Report 2007/402.

[6] N. T. Courtois. The dark side of security by obscurity and cloning MiFare classic rail and building passes anywhere, anytime. Cryptology ePrint, Report 2009/137.

[7] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard (Information Security and Cryptography)*. 1 edition, 2002.

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

[9] Giuseppe Di Battista and Bernardo Palazzi. Authenticated relational tables and authenticated skip lists. In *DBSec*, pages 31–46, 2007.

[10] EPCglobal. Epc Radio-Frequency Identity Protocols – Class-1 Generation-2 UHF RFID – Protocol for communications at 860MHz-960MHz, 2008.

[11] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. Silicon physical random functions. In *Computer and Communication Security Conf.*, pages 148–160, 2002.

[12] N. Haller, C. Metz, P. Nesser, and M. Straw. A One-Time Password System. RFC 2289 (Standard), February 1998.

[13] Günter Karjoth and Paul A. Moskowitz. Disabling RFID tags with visible confirmation: clipped tags are silenced. In *WPES '05*, pages 27–30, 2005.

[14] D. Konidala, Z. Kim, and K. Kim. A simple and cost-effective RFID tag-reader mutual authentication scheme. In *Conf. on RFID Security*, pages 141–152, 2007.

[15] J. Lee and Y. Yeom. Efficient RFID authentication protocols based on pseudorandom sequence generators. Cryptology ePrint Archive, Report 2008/343.

[16] K. Ouafi, R. Overbeck, and S. Vaudenay. On the security of HB# against a man-in-the-middle attack. In *ASIACRYPT*, 2008.

[17] Roberto Tamassia. Authenticated data structures. In *ESA*, pages 2–5, 2003.

[18] P. Tuyls and L. Batina. RFID-tags for anti-counterfeiting. In *CT-RSA 06*. 13–17.

[19] Verayo, Inc. PUF RFID. http://www.verayo.com/.

[20] A. Wallstabe and H. Pohl. Implementing high-level counterfeit security using RFID and PKI. In *RFID SysTech 2007*.