

Agence Nationale de la Recherche GIP
ANR

DWP3.1

Preliminary Design of Security Protocols

Date: 12.07.2009

Version: 2.0

CEA-LETI, Eurecom, INRIA and Orange Labs
Deliverable for RFID-AP (ANR SESUR)

Table of Contents

1	Introduction	3
2	RFID protocol features	3
2.1	Identification	3
2.2	Authentication	4
3	RFID MAC protocols	5
3.1	MAC protocols used in RFIDs	5
3.2	The ISO14443A standard MAC protocol	6
3.3	The EPC standard MAC protocol	6
3.4	Shoehorning privacy into RFID singulation protocols	10
4	Privacy-preserving RFID identification protocols	11
4.1	Information-theoretic protocols	11
4.2	Hash-based protocols	12
4.3	Key-tree based protocols	14
4.4	Protocols based on experimental crypto-primitives	16
5	RFID Authentication protocols	16
5.1	Symmetric cipher and hash-based protocols	17
5.2	Protocols based on public-key cryptography	18
5.3	HB+ and its variants	19
5.4	Physically Uncloneable Functions	19
6	Protocol properties overview	20
7	Conclusions	20

1 Introduction

There are two major goals for RFID protocols: identification and authentication. There are many security aspects regarding these two goals: privacy, secure and repeatable authentication, authentication without shared common secrets, the list is long. In this Work Package we focus on the most prominent protocols and their security aspects.

Organisation

Firstly, in Sect. 2, we detail the definitions used throughout this Work Package. Next, in Sect. 3 we describe the Media Access Control protocols used by two different RFID standards, ISO14443 and EPC Class 1 Gen. 2 and explore how they could be made private. We detail a set of prominent RFID protocols that provide private identification in Sect. 4. We then describe some of the most influential RFID authentication protocols in Sect. 5. Finally, in Sect. 6. we compare the aforementioned protocols using a feature-comparison table and in Sect. 7 we conclude this Work Package.

2 RFID protocol features

In this section we give a short description of the protocol features relevant for RFIDs. Although these features are not unique to RFIDs, their implementation is particular to the its setting, given the range of features and restrictions of RFIDs.

2.1 Identification

Identification is the act of associating an identifier in the system with an entity that we are communicating with. For instance, when a reader communicates with an EPC tag, the tag tells its EPC code, an identifier that can be looked up in the EPC code database. Identification does not make sure that the identifier sent was not faultly or maliciously sent – indeed, the EPC code sent by EPC tags can be easily sent by any malicious entity.

Identification can be Untraceable, Unlinkable, Forward-secure, and Anonymous. We now shortly describe these features:

Anonymity An identification protocol is anonymous if no adversary is able to derive the (secret) identity of the tag. An adversary is considered to be one or more people collaborating with sufficient resources such as readers, large support backends such as databases and computing resources. In the RFID litterature, the security guarantee required is 2^{80} computation, that is to say, we do not allow the adversary to have enough resources to accomplish 2^{80} operations in any reasonable amount of time.

Untraceability Suppose that an adversary is able to accumulate logs of tag-reader interactions, i.e., a (partially) successful protocol executions. Further, suppose that an adversary has accumulated a set L_1 of execution logs of a protocol between various tags and readers, and the adversary has access to the set L_2 of execution logs of the same identification protocol between one specific tag \mathcal{T}_j and various readers. The identification protocol is said to provide untraceability, if the adversary cannot decide whether some of the logs in L_1 relate to \mathcal{T}_j with a higher probability than a randomly guess.

Unlinkability Suppose that, again, an adversary has access to the set L_1 of logs of the execution of an identification protocol between various tags and readers. The protocol is said to provide unlinkability, if the adversary is unable to select any pair of logs that relate to one and the same tag – with higher probability than just guessing.

Forward secrecy Suppose that an adversary has access to the inner state of a tag \mathcal{T}_j , e.g. by physically compromising \mathcal{T}_j , reading out all its memory. Also, let us suppose that the adversary has collected a set L_1 of past execution logs of the protocol between various tags and various readers. Then, an authentication protocol is said to provide forward-secrecy, if an adversary is unable to tell which execution of the authentication protocol relates to \mathcal{T} other than with negligible advantage over a random guess.

Among these notions of privacy, the following implications can be identified [60]:

$$\text{Forward Secrecy} \Rightarrow \text{Unlinkability} \Rightarrow \text{Untraceability} \Rightarrow \text{Anonymity}$$

2.2 Authentication

To make sure that the entity sending the identifier A is indeed entity A , authentication is required. Authentication is the act of making sure that the other partner is indeed the partner it claims to be. For authentication to take place, the entity wishing to authenticate to the other party must contain a secret whose knowledge can be verified by the other party.

An authentication protocol authenticates entity A towards entity B such that B is sure that entity A is indeed who it claims to be. We say that the authentication is unidirectional when either the reader authenticates the tag, or the tag authenticates the reader. Authentication is *mutual* when both the tag authenticates the reader and the reader authenticates the tag.

Authentication requires either a challenge or tight time synchronisation to prevent so-called “replay attacks”. In a replay attack an adversary eavesdrops on the communication between reader and tag, stores the communication, and later replays the communication to impersonate a tag or a reader. Using time synchronisation in RFIDs is, however, rarely used as it would require expensive hardware on the tag, such as a battery. Using challenges is the generally accepted way to ensure *freshness* in the RFID setting. The challenge is usually sent from

the reader to the tag (for tag authentication) before further communication, and it not only ensures the freshness but can also be used to make subsequent data exchanged dependent on the challenge. For both tag and reader authentication, a challenge must be sent by both parties.

3 RFID MAC protocols

When multiple entities want to communicate at the same time they can cause interference prohibiting communication. Such an interference between two tags is called a *collision* in the RFID literature. To circumvent this the problem, there must be a Media Access Control (MAC) protocol used to control who, when and where can talk on which channel(s). This problem is common to all wireless communication protocols such as Wi-Fi, Bluetooth, and RFIDs. In this section we first outline the two MAC methods used by RFID systems, and then we describe in detail how the EPC [20] and the ISO14443A RFID protocols handle MAC. Once these protocols have been shortly described, we elaborate on how – if at all – they could be made private, a major goal of many higher-level RFID protocols.

3.1 MAC protocols used in RFIDs

There are two MAC protocols used in RFIDs, one used by the readers to avoid collisions between each other, and the other by the tags to avoid collisions between themselves. These two MAC protocols use different aspect of the communication to distinguish between entities that need to communicate at approximately the same time: the first uses the difference in physical location and angle of orientation, and the second the difference in the timing of the communication.

3.1.1 Space Division Multiple Access (SDMA)

In SDMA, the space is set up such that entities normally occupy different physical spaces. Since the communicating entities can communicate only in different areas around themselves, the spacing of the entities relative to each other solves the problem of multiple access. In RFIDs, it is relatively easy to make a reader that can only communicate in a very selective part of space around itself, thus easing the problem of interference between readers. However, RFID tags are normally made such that they occupy an evenly distributed space around themselves, thus this solution does not solve the problem of interference caused by two (or more) tags talking to the same reader.

3.1.2 Time Division Multiple Access (TDMA)

In TDMA, the time is divided such that only two entities talk at any given time. The negotiation of time slices is the most challenging part in this system. RFIDs use TDMA, as it is both cheap to implement, and suits the typical system

configuration of a higher-powered and more intelligent entity (the reader), and multiple, less sophisticated entities (the tags).

3.2 The ISO14443A standard MAC protocol

The ISO14443A standard [35] uses TDMA as the MAC protocol, a protocol which it calls *singulation*, and is mandatory to use to avoid collisions. The singulation protocol is deterministic, the reader explores the Unique Identifier (UID) of the tag in a bit-by-bit tree-walking fashion. This protocol is simple to implement on the tag and is relatively time efficient if the UID is short. However, since the reader calls the different tags using their unique UID, the tags cannot remain anonymous during singulation. Furthermore, since the tag UID is repeated during the protocol by the reader, and the reader has a much greater range than the response emitted by the tag, the UID can be eavesdropped from a safe distance by a malicious reader.

An example protocol run of the tree-walking algorithm is in Fig 1. Bit-collisions occur when at least one tag sends a binary “0” and the other(s) a binary “1”. These collisions are detected using channel coding. When a collision occurs, the reader must choose which branch to follow. The reader thus explores a chosen part of the binary collision-tree during the protocol.

3.3 The EPC standard MAC protocol

The TDMA collision-avoidance protocol used by EPCglobal Class 1 Gen. 2 standard [20] is also called *singulation*, and is mandatory to use to avoid collisions. The course of this protocol is important since it may leak some information that may threaten security or privacy. EPC uses a probabilistic algorithm which does not ensure a fixed time for the singulation of all the tags in the field. It implements the principle of the management of time slots known as the ALOHA protocol.

An example run of the EPC singulation protocol is present in Fig. 2. Inventorying of the tags around the reader begins with a *Select* command from the reader. This command determines the tag population that will take part in the process. This selection is done via the “SL” or “inventoried” flag to separate tags in two populations A and B. The selection can also be done on a specified part of the tags memory (EPC, TID or User memory) with a mask which can be fully defined in the command.

When the reader has selected the subset of tags, it can launch the singulation with the *Query* command. It contains a parameter Q which will define $2Q-1$ time slots. When the tags received a *Query*, they shall pick a random value in the range $[0, 2Q-1]$ which will determine the time-slot where they will reply. Tags that pick a zero shall reply immediately a 16 bits random number (RN16) then if there is a single the reader shall acknowledge it with the *ACK* command containing the RN16. After an *ACK* command, the tag answers with its EPC code and invert its “inventoried” or “SL” (for selected) flag.

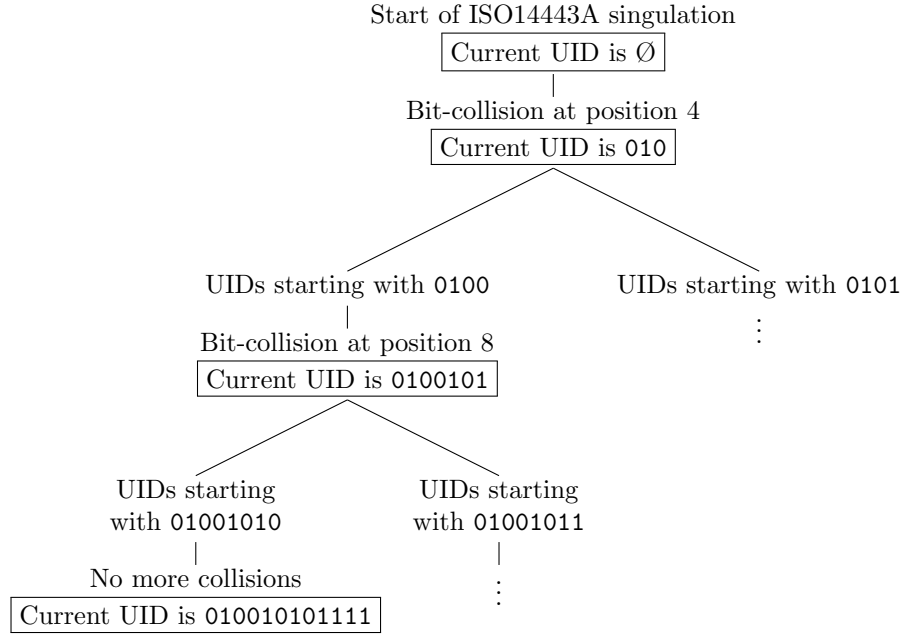


Figure 1: Example ISO14443A singulation protocol, where the UID length is 12. During singulation, there are two collisions, one at the 4th bit of the UID, and one the 8th. The found tag's UID is 010010101111. There could be many tags in the vicinity of the reader, as there might have been many more collisions in the other (unvisited) parts of the tree.

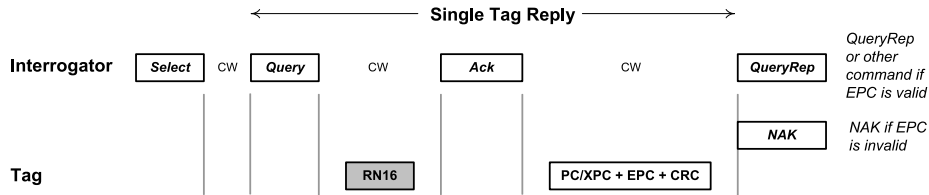


Figure 2: An example run of the EPC Class 1 Gen. 2 standard's singulation protocol. In this run, a tag randomly selects 0 as its slot-counter, and so it immediately replies to the *Select* command of the reader with a random number RN16 generated by the on-board Pseudo Random Number Generator (PRNG). There are no collisions, so the reader replies with an *ACK* command, re requests the tag to send its EPC code, and some other data (PC/XPC), plus the error-checking Cyclic Redundancy Check (CRC) code. If the EPC code is valid, the reader can further issue the *QueryRep* command. If the EPC is invalid, the reader issues a *NAK* (Not Acknowledge) command instead.

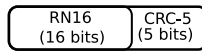
To change the time-slot after an EPC reply from a single tag, or after a collision from several tags, or even after an empty slot, the reader sends a *QueryRep* or *QueryAdjust* (to change the value of Q) that has the consequence to decrease the tags time-slot counter. The reader can also send a *NAK* command after the answer of the tags if it did not understand the EPC.

3.3.1 The *Select* command

The *Select* command is mandatory to define a subset of the tags in the field which will take part of the inventory. This subset can be determined by applying a mask in any bank of the memory of the tag (EPC, TID or user memory) with a fixed position (“pointer” parameter) and/or length (“length” parameter). Only the tags which have the value of the mask at the defined part of the memory will be activated for the inventory. The “truncate” parameter enables to limit the length of the EPC code which later will be replied by the tag.

3.3.2 The *Query* command

The *Query* command initiates and specifies an inventory round. This command lets the reader define miscellaneous parameters of the physical layer such as the data rate of the response. The main parameter is the number Q between 0 and 15 which determines the $2^Q - 1$ time slots. As soon as the *Query* command is received by the tag, it initialises its slot-counter with a random number from its on-board Pseudo-Random Number Generator (PRNG), called RN16 in the EPC standard. At each new *QueryRep* command received from the reader, the tag decrements its slot-counter. If the slot-counter has reached zero, it immediately replies with another random number, again drawn from RN16, and a parity-check code CRC-5:



The properties to which the random number generator RN16 must adhere to is defined in the EPC standard:

- Probability of a single RN16: The probability that any RN16 drawn from the RNG has value $RN16 = j$, for any j , shall be bounded by $0.8/216 < P(RN16 = j) < 1.25/216$
- Probability of simultaneously identical sequences: For a tag population of up to ten thousand tags, the probability that any two or more tags simultaneously generate the same sequence of RN16s shall be less than 0,1%, regardless of when the tags are energised (i.e. put in the field of the reader).
- Probability of predicting an RN16: An RN16 drawn from a tag RNG 10ms after the end of T_r shall not be predictable with a probability greater than 0.025% if the outcomes of prior draws from the RNG, performed under identical conditions, are known.

3.3.3 The *ACK* command

A reader sends an *ACK* to acknowledge a single tag. This command echoes the RN16 of the tag reply. If the tag receives an incorrect RN16 then it remains in its inventory mode.

After an *ACK*, the tag answers with the concatenation of its PC/XPC, EPC, and finally a CRC-16 to check the transmission. The PC/XPC parameter defines the length of the EPC, the user memory (enabled or not), and if the application referred to as an EPCglobal application. The EPC code is the unique identifier of the tag which enables to track items and as a consequence it is the basis of the privacy issue in the EPC standard. The CRC-16 is used to reduce the possibility of faulty transmissions. After the transmission, the tags invert their “SL” (SeLected) or “inventoried” flags.

3.3.4 The *QueryRep* and *QueryAdjust* command

The *QueryRep* command, as shortly described previously, instructs the tags to decrement their slot counters and if the slot reached zero, to modulate an RN16 and the CRC-5 to the reader. *QueryRep* is the command used to change the time-slot after an EPC reply, a collision or an empty time slot.

The *QueryAdjust* command adjusts the parameter Q without changing any other round parameters. Tags then have to again pick a random value between 0 and $2^Q - 1$ and load it in their slot counter. Without this command, if the parameter Q was chosen by the reader to be too low (underestimating the tag population), the reader would not be able to change Q , thus rendering the identification of tags infeasible. For instance, if Q was chosen as 1, there would be exactly 1 slot, and if there were multiple tags in the vicinity of the reader, they would always collide, thus rendering the reader unable to identify either of them.

3.3.5 The *NAK* command

The *NAK* command shows to the tags a non acknowledgement of the reader. Consequently, tags have to ignore the previous command of the reader.

3.3.6 The *Access* commands

To access to the memory of a specific tag that just has been singulated or to send it a specific command such as the *Kill* command (which forever deactivates the tag), the reader asks the tag via its RN16 to pick and reply a new 16-bit random number (command *ReqRN*) called “handle”. This new handle becomes the new pointer to the tag and then the new argument of the command to access memory.

3.4 Shoehorning privacy into RFID singulation protocols

Neither of the aforementioned singulation protocols were designed with privacy in mind: the ISO14443A protocol requires the UID to be transmitted by both the tag and the reader, while the EPC standard uses a random number to carry out singulation, but once the tag is singulated, it emits its EPC code, i.e. its identifier.

Work on shoehorning authentication into RFID standards have been done by Bailey and Juels in [3]. However, instead of adding authentication, in this section we wish to include some level of privacy into the standard, as authentication could be added, as Bailey and Juels note, through a higher layer: they propose to use the ISO7816-4 standard command set to achieve this. In this section, we focus on privacy rather than authentication, thus our approach focuses instead on modifying the lower-layer protocols in such a way that they are changed the least possible, but preserve the privacy of the tag and thus its owner.

3.4.1 Adding privacy to the ISO14443A singulation protocol

A possible to achieve privacy in the ISO14443A singulation protocol is by using a randomised UID each time the protocol is run. Since normally there are no more than a couple of thousand tags in the field of the reader at any given time, the chance of randomly hitting on the same UID is minimal, as the UID is at least 32-bit long (and can be up to 128-bit long). However, in the event of two UIDs randomly choosing the same UID, since the protocol requires the UIDs to be unique, singulation would not work, and an error would ensue.

To properly protect the tag from tracing, the random number generator used to generate the UID must be of good quality. Implementing such a random number generator is expensive in terms of hardware requirements, however, it might be possible to re-use it in higher-level protocols. As privacy-preserving protocols all need a source of good random numbers, the cost of this random number generator can be spread across the two layers.

3.4.2 Adding privacy to the EPC singulation protocol

The EPC Class 1 Gen. 2 standards' singulation protocol does not require the EPC identifiers to be different, instead, it relies on the random number RN16 to select time slots and thus carry out the singulation. This considerably eases shoehorning privacy into the EPC protocol, as the only privacy-threatening part of the protocol is the sending of the EPC code (i.e. ID of the tag) by the tag after the *ACK* command of the reader. A simple way to fix this is to send a fix, all-zero EPC code by all tags after the *ACK* command, and let the higher-level private identification protocol carry out the identification.

Just as with the ISO14443A, the random number generator used during singulation must be able to generate high-quality random numbers that cannot be linked to one another. In the case of the EPC, however, it is sufficient to augment the RN16 random number generator already on the tag, thus saving space.

4 Privacy-preserving RFID identification protocols

Since people do not wish to be easily tracked, privacy is essential for any public deployment of RFID tags. For example, though the EPC standard does not have any mechanism for authentication, it provides the Kill command for privacy protection – which disables the tag. Such a drastic mode of privacy protection has, however, a negative impact on EPC tags, as they can only be used until disabled, which nowadays happens when the products are put on store shelves. In other words, as of yet, the advantages of EPC tags on products cannot be experienced by end users as they never receive working tags.

There are multiple definitions of RFID privacy. A well-known one is by Ohkubo et al. [39]. They define *Indistinguishably* and *Forward security* and then show that their scheme is indistinguishable and forward secure according to their definitions. Juels et al. define privacy in another way [38]. In this model, an RFID system is secure if an adversary has a non-negligible chance of distinguishing between two tags in a clearly defined *Privacy Experiment*. There are multiple refinements of the Juels model, e.g. one by Ouafi et al. [49], who for instance put different constraints on the adversary. A more complete but also more complex definition is by Vaudenay [60] who considers different types of attackers, namely STRONG, DESCSTRUCTIVE, FORWARD, WEAK and NARROW and then demonstrate protocols that resist attacks by certain types of attackers.

We have identified four different types of protocols that aim to protect the tag's and thus the tag owner's privacy. These are: information-theoretic protocols, hash-based protocols, key-tree protocols, and experimental protocols. In this section, we will go through each of these using example protocols for each.

4.1 Information-theoretic protocols

Information theoretic protocols simply withhold information from the attacker such that only the legitimate reader can identify the tag. The most simple way of implementing such a protocol is by Juels [36], in which tags keep a set of randomly chosen pseudonyms $\alpha_1, \alpha_2, \dots, \alpha_n$, and use them one after the other when identifying themselves to a reader. Since the reader is aware that pseudonyms are used to hide the real identity of the tag, it causes no problems for the reader to identify the tag. An eavesdropper cannot distinguish that the observed α 's belongs to the same or to different tags. However, since the tag's memory is limited, it can only store a limited n number of pseudonyms. After these have been exhausted, the tag must re-use an already used pseudonym, thereby becoming traceable.

The advantage of information-theoretic protocols is that they are guaranteed to be secure until a certain point. The problem with with them, however, is that they all loose their security after a limited number of identification attempts.

To alleviate this problem, the pseudonym-rotation protocol updates its internal pseudonyms after every authentication with a legitimate reader. This unlinks the current state of the tag from its old state, but the tag can again be traced by simply demanding identification n number of times.

4.2 Hash-based protocols

The most well-known privacy-preserving RFID protocol using hashing is the OSK protocol [39]. The protocol can be characterised as follows. The tag is loaded with initial state s_1 . At each identification query by the reader, the following happens:

1. Tag sends answer $a_i = G(s_i)$
2. Tag updates internal state: $s_{i+1} = H(s_i)$

As the reader knows the state s_i of all tags, it can simply generate $G(s_i)$ for each and find the one that matches.

The advantages of such a protocol are twofold. First, it provides privacy for an unlimited amount of identification attempts, unlike the information-theoretic protocols. Second, it is secure if the functions H and G are preimage-resistant. There are many disadvantages associated with the protocol, however. First, and foremost, the tag can simply be queried by a malicious reader for a long period of time, after which the state has evolved, let's say, m times. The backend, which has n tags, to find which tag that sent a_{i+m} , needs to iterate function H and execute G for each tag in the system exactly m times. Therefore, it takes $2mn$ operations to find the tag, which can quickly take too much time with large enough populations and and large enough m .

There have been many iterations of the OSK scheme to alleviate these problems. One such iteration is by Avoine et al. [1], which proposes a time-memory trade-off to shorten the time it takes to find the tag. The trade-off works using Hellman's idea [33], later improved by Oechslin [46] that reduces the amount of work T needed to invert any value in a set of N outputs of a one-way function given enough memory. The trade-off can reduce the amount of work from N to $N^{2/3}$ using $N^{2/3}$ units of memory. The authors find appropriate trade-offs and present multiple examples. One such example uses $1GB$ of memory, and given $m = 2^{10}$ and $n = 2^{20}$, the time to find the tag is at most 0.0016 s, 38836 times better than without memory usage. Though this version of OSK solves one of the principle problems of OSK, a tag can still be made unreachable by the system by simply querying it more than $m \gg 2^{10}$ times.

The YA-TRAP protocol, proposed by Tsudik [59] is a another generation of hash-based private identification schemes, which, as an added feature, also provides authentication. The YA-TRAP protocol is described in Fig. 3. The main idea behind YA-TRAP is to store a timestamp on the tag that can only increase to a given t_{max} time, and which is updated after every query by the reader. Essentially, the reader sends a timestamp t_j which must be larger than the current timestamp in the tag t_i , and the tag replies with the $HMAC_{K_i}$ of

Reader \mathcal{R}_j	Tag \mathcal{T}_i
Database L : $\{\dots, (t_j, \text{HMAC}_{K_i}(t_j)), \dots\}$	Shared secrets: K_i, t_0, t_i, t_{max}
$\longrightarrow t_j$	if $(t_j < t_i)$ or $(t_j > t_{max})$ $h_j = \text{PRNG}_i^j$ else $h_j = \text{HMAC}_{K_i}(t_j)$ and update $t_i \leftarrow t_j$
check $\exists t_j$ s.t. $(t_j, h_j) \in L$	$\longleftarrow h_j$

Figure 3: The YA-TRAP protocol. On the top, the shared secrets are indicated. The protocol starts with the reader interrogating the tag with current time t_j , to which the tag responds either PRNG_i^j or $\text{HMAC}_{K_i}(t_j)$. The reader then checks using its internal database L , whether the returned value is correct or not.

the timestamp t_j , and the tag updates its timestamp to t_j . The reader simply checks if the pair (timestamp, HMAC) corresponds to a K_i in its database. This scheme is interesting, as it allows for batch operation: if many tags need to be identified and authenticated, such as envisioned by the EPC, e.g. when filling up a the stock of a supermarket, the batch of tags can be identified in $O(n)$ operations. This batch-processing is done as follows: the tags' replies, the h_j -s are collected in a hash table data structure, and when the reader goes through each element in the database, it searches the hash table for matches. Since it takes $O(1)$ to search in a hash table, it takes $O(n)$ to go through all elements of the database, finding the match for every collected h_j on the way.

However, the YA-TRAP scheme has multiple drawbacks. First of all, a denial-of-service (DoS) attack is trivial to carry out, as it suffices to send a very high t_j to the tag, e.g. $t_{max} - 1$. Also, as shown by Ouafi et al. [49], the tag can be made traceable by first making the tag think it is in the future then observing a validation check of the tag by the reader, thus winning the Privacy Experiment of Juels and Weis [38]. Subsequently, YA-TRAP was revisited and updated by Burmester et al. into YA-TRAP+ and O-TRAP [13], both of which have later been shown to be traceable by Ouafi et al. [49].

RIPP-FS by Conti et al. [17] is another hash-based protocol, which distinguishes itself by employing hash chains, originally proposed by Lamport in [40]. Each tag \mathcal{T}_i is initialised with a tag key $K_{\mathcal{T}_i}$, shared with the reader. The tag also stores the initial pair (K_0, t_0) generated by the reader, where K_0 is the last value in the tag-specific hash chain:

$$K_l = w$$

$$K_i = H(K_{i+1}) = H^{l-1}(w), i = 0, \dots, l-1$$

where w is the seed, and t_j ($j = 0, \dots, l$) is a time interval counter. Each tag requires a pseudo-random number generator, where PRNG_j denotes the j -th invocation by tag \mathcal{T}_i of its PRNG. One of the goals of RIPP-FS's design was to achieve untraceability, and offer more security guarantees than YA-TRAP and its variants. However, the untraceability properties of RIPP-FS was broken by Ouafi et al. [49] in the Juels and Weis model's proposed Privacy Experiment.

Although hash-based protocols have very useful properties, they are still unpractical for two reasons. Firstly, hash functions have been shown to be much more resource-intensive to implement on RFID tags than previously thought [21]. Secondly, with hash-based protocols it is often the case that either the number of queries allowed to the tag is limited by the protocol, or if the tag is queried too many times, the tag can be lost from the system: in YA-TRAP and RIPP-FS if the time-stamp given is too large the tag is rendered inoperative, while with OSK if the tag is queried too many times, the resulting evolution of the tag's internal state renders the tag unidentifiable and thus dysfunctional.

4.3 Key-tree based protocols

Key-tree based protocols are a family of protocols based on the original protocol by Molnar and Wagner [43]. In this protocol, each tag is a leaf of a balanced tree of depth d , with a branching factor b , thus the total number of tags in the system is b^d . At every level, each branch of the tree has an associated key. The tags know all keys on their paths from the root to the leaf. An example tree is shown in Fig. 4. When a tag wants to identify itself to the reader, it executes the protocol as described in Fig. 5 at each level of the tree from the root to its corresponding leaf.

As there are only b branches of the tree at each level, the function f only needs to be calculated on average $b/2$ times at each level, for a total of $b \cdot d/2$ executions of f on average. Since there are b^d tags in the system, this is logarithmic in the number of tags. The authors describe a trade-off between the branching factor and the depth, which is further elaborated upon by Buttyan et al. [14], where the authors calculate the depth and branching factor needed for different tag populations and timing constraints.

The key-tree approach has several features that make it exceptional in many respects. First of all, it is adaptable to any protocol: the PRF function simply needs to be replaced by the protocol in question. Secondly, if the underlying protocol has the appropriate security features (a feature that PRFs provide), it also provides authentication of the tag. Finally, if the last step is carried out, the protocol provides mutual authentication.

The drawback of using key-trees is that tags must share secrets. Since tags are usually not in a controlled environment, and they are rarely tamper-resistant, the risk of key compromise by malicious parties is not negligible. Once a tag is tampered with and its stored keys are recovered, the adversary can use the keys to undermine the privacy of other tags. For instance, if the branching factor was two, then by recovering the keys in only one tag, the anonymity group (the group of tags in which the given tag is indistinguishable from the others) of

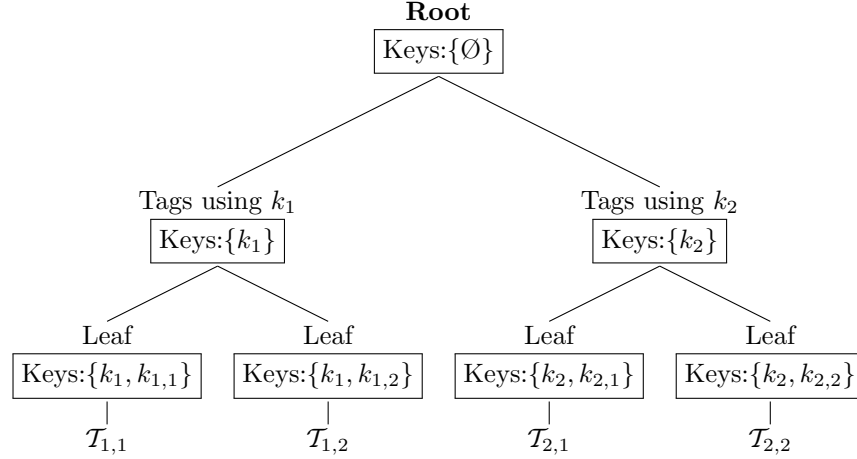


Figure 4: An example Molnar-Wagner key-tree. The branching factor b is 2 and the depth d is also 2, for a total of four tags in the system. The weakness of the system is the following: by tampering with and compromising the keys stored in tag $\mathcal{T}_{2,2}$, the privacy of tag $\mathcal{T}_{2,1}$ is compromised as there are no other tags on the branch with key k_2 . Also, the remaining two tags' anonymity set is halved, as the original anonymity set was $\{\mathcal{T}_{1,1}, \dots, \mathcal{T}_{2,2}\}$, but only $\{\mathcal{T}_{1,1}, \mathcal{T}_{1,2}\}$ remains – a drop from a set of 4 to a set of 2.

Reader \mathcal{R}_j	Tag \mathcal{T}_i
Generate nonce r_1	
	$\longrightarrow r_1$
	Generate nonce r_2 and calculate
	$\sigma = ID \oplus f_k(0, r_1, r_2)$
	$\longleftarrow r_2, \sigma$
find $(k, ID) \in L$ s.t.	
$ID = \sigma \oplus f_k(0, r_1, r_2)$	
<i>optional – only for mutual authentication</i>	
calculate	
$\tau = ID \oplus f_k(1, r_1, r_2)$	
	$\longrightarrow \tau$
	check $\tau \stackrel{?}{=} ID \oplus f_k(1, r_1, r_2)$

Figure 5: The Molnar-Wagner protocol, as executed at each level of the tree, starting from the root of the tree. N is the database of tags and their respective keys for the current level, and f is a Pseudo-Random Function (PRF) implemented in all tags and readers.

every tag in the system is at least halved. The paper by Karsten and Evans [45] characterises this privacy loss for key-trees with different parameters, and arrives at the conclusion that two-depth trees are the most appropriate for many RFID scenarios.

4.4 Protocols based on experimental crypto-primitives

Complexity limits of RFIDs often require protocol designers to invent new primitives, as standard primitives take up too many gate counts, or are unpractical for the RFID setting. It often happens that these new crypto-primitives are found to have weaknesses that were not anticipated by their designers. To overcome the newly discovered shortcomings, the protocols are updated by their original designers or others, and the challenge of finding weaknesses starts again. This cycle is often repeated until the protocol is sufficiently robust to withstand serious attacks.

There are multiple examples in the literature where RFID protocols were designed, published, shown to be weak against certain attacks, and then re-designed. One such example is LMAP [51], shown to be susceptible an active attack by Barasz et al. [4], then re-designed as M2AP [52], and again shown to be weak against certain attacks by Barasz et al. [5]. Another such protocol is ProbIP by Castelluccia and Soos [16], shown to be insecure by Ouafi et al. [49]. The DPM protocol by Di Pietro and Molva [53] is also part of the family of experimental protocols and it too has been shown to have some insecure features by Soos [58]. Continuing the improvement cycle of (re-)design-and-attack, the DPM protocol is has been updated by its designers and others to the Ff family of protocols [7].

Protocols that are widely different from the kind usually accepted by the cryptographic establishment could bring a big leap forward. However, from the designer's perspective, a good cryptographic background is indispensable for the creation of such protocols, otherwise well-established cryptographic techniques will be used with high success rates. From the attacker's perspective, flexibility is required to use such cryptographic techniques in a completely new environment.

5 RFID Authentication protocols

Authentication for RFIDs, though is a secondary objective, has received much attention due to the many advantages it could bring. For instance, if RFIDs could be equipped with authentication mechanism, they could be used for securing building access, or for micropayment in public transport. If EPC tags could be cheaply equipped with a means of authentication, they could be used to authenticate warranty repairs, bringing paper-less warranties for customers and less fraudulent repairs for shops.

We have identified four different types of authentication protocols for RFIDs. These are: symmetric cipher-based protocols, public-key protocols, PUF-based protocols, and LPN (Learning Parity with Noise)-based protocols.

5.1 Symmetric cipher and hash-based protocols

Given a symmetric cipher with a low hardware footprint, it is relatively easy to make a challenge-response authentication protocol. Therefore, there has been a large research effort on implementing standard crypto-primitives on hardware constrained devices.

In the area of block ciphers, a well-known result is that by Feldhofer et al. [23], implementing AES on 3500 gate equivalences. To reduce the hardware footprint to such an extent, the authors use an 8-bit architecture, calculate the round-key on-the-fly, and store the state only once (unlike FPGA-implementations). This implementation of AES takes 1032 clock cycles to encrypt a cleartext, and 1165 clock cycles to decrypt a ciphertext. Another block-cipher implementation is that of DES on 2300 gate equivalences by Poschmann et al. [54], which takes in only 144 clock cycles to encrypt a ciphertext, but cannot decrypt (which the authors argue is not of primary importance). In [54] the authors also put forward a variation of DES, DESL, that uses a serialised S-box, thereby reducing the implementation footprint to 1848 gate equivalences. Clock cycle-wise the proposed DESL implementation works the same as the authors' DES implementation. Finally, a block cipher designed specifically for RFIDs is PRESENT by Bogdanov et al. [9]. It has a candidate implementation of only 1570 gate equivalences that takes in only 32 clock cycles to encrypt a cleartext and, similarly to the DESL implementation, does not have decryption functionality.

Low hardware-footprint implementation of the SHA-1 hash function has also been attempted. However, the RFID-optimised implementation by Feldhofer and Wolkerstorfer [22] requires 10868 gate equivalences, much more than an RFID could handle. In the same paper, the authors implement MD5 and other hash functions, all of which require more than 8000 gate equivalences. From the paper it follows that most of the implementation challenge consists of reducing the number of registers needed and the number of flip-flops clocked. However, hash functions typically have a message expansion phase which require many registers to store the intermediate values, and they usually also operate on multi-byte words, requiring many flip-flops to be clocked at the same time. In contrast, AES has only a storage need of 256bits and operates on single-byte words. Therefore, as Feldhofer et al. [21] have previously pointed out, it is debatable whether currently used hash function designs such as MD5 and SHA-1 are suitable for RFIDs at all.

Another way of solving the problem of hardware constraints is to tweak the cipher itself instead of tweaking its implementation. For stream ciphers, the eStream project's low footprint hardware portfolio [2] is such an attempt. Two of its most well-studied candidates, Trivium [15] and Grain [32] could be promising for applications in RFIDs. Of the two, Trivium operates with a 287-bit state while Grain needs only 160 bits of state to operate. Though Grain uses more complex functions than Trivium, it still seems a better candidate for RFIDs since its extra functional complexity is more than offset by its much lower register usage, and consequently its much smaller hardware footprint. Grain also needs far less initialisation steps: it uses only $2 \cot 80 = 160$, clocking both

of its registers twice fully, while Trivium clocks its register set four times for $4 \cdot 288 = 1152$ initialisation steps in total. Therefore, on low hardware-footprint devices, Trivium’s initialisation takes much longer, adding additional timing overhead on the protocol it is used in.

5.2 Protocols based on public-key cryptography

There are two major types of RFID identification protocols using public-key cryptography. The first type of such schemes, introduced by Shamir [57], relies on a variation of the Rabin cryptosystem [55], but it replaces the squaring of the plaintext with an addition of the random multiple of the divisor. The second type of such schemes use a token-based approach where pre-computed tokens, *coupons* are stored on the tag. The tag, when queried, uses up these coupons to authenticate itself to the reader. The coupons are such that the tag only needs to do a limited number of operations to use them.

The Rabin cryptosystem-type scheme is implemented both in SQUASH [57] by Shamir and in WIPR [47] by Oren and Feldhofer. SQUASH uses multiple techniques to reduce the size of the resulting RFID implementation. For the modulus n it uses a composite Mersenne number of the form $n = 2^k - 1$ to reduce the storage cost. To reduce on-the-tag computation and the communication overhead, the tag does not need to send the whole encrypted ciphertext: a limited number of bits, say t suffice to make the scheme 2^{-t} -secure. SQUASH suggests to use a t -long window in the middle, but instead of computing all previous bits, suggests to compute only u *guard bits* before this t -long window, further reducing computational costs. After a detailed description of these techniques, the author specifies SQUASH-128 as an example proposal. SQUASH-128 uses about half the number of gates in GRAIN-128 and claims a protection against an adversary with less than 2^{64} of time and space.

The coupon-type scheme is implemented by McLoone and Robshaw [42] in their RFID-optimised implementation of the already ISO-standardised (ISO/IEC FDIS 9798-5) GPS protocol [29]. McLoone and Robshaw propose to use an elliptic curve variant of GPS due to Girault [30] and also require the reader to use Low Hamming Weight challenges, an improvement by Girault and Lefranc [31], to reduce parameter sizes. In their scheme, McLoone and Robshaw replace the modular exponentiation with a coupon and a simple integer (non-modular) calculation. They propose multiple implementations of their scheme, notably with and without a PRNG to help re-generate the random number inside the coupon. The PRNG takes about 1000 gate equivalences on the tag, but drastically reduces the coupon sizes. With the PRNG, the implementation fits on no more than an estimated 1500 gate equivalences, and 10 such reduced-sized coupons take up approximately 500 GEs, for a total of 2000 GEs.

The two types of schemes described both have their respective advantages and disadvantages. The Rabin cryptosystem-type schemes have no limit on the number of authentications, but are susceptible to attack, as demonstrated by Ouafi and Vaudenay in [50], where they authors break the security of SQUASH-0, a variation of SQUASH suggested by Shamir. On the other hand, the coupon-

type approach can be easily rendered useless by a malicious reader through the simple exhaustion of coupons – a type of Denial of Service (DoS) attack.

5.3 HB⁺ and its variants

The HB⁺ protocol by Juels and Weis [37] was a great leap forward by the RFID community towards low hardware footprint authentication protocols. The protocol uses the idea by Hopper and Blum [34], which in turn uses the Learning Parity with Noise (LPN) problem, also known as the minimal disagreement parity problem [18], to base its security on. The LPN problem is known to be NP-hard [6], though finding the solution to it has been consequently improved with ever newer algorithms. The original BKW algorithm [8] was superseded by that of Fossorier et al. [24] and then by that of Leveil and Fouque [61]. Using the newest algorithm, HB⁺'s claimed 2^{80} security is drops to around 2^{52} . On top of algorithmic advances on LPN, the protocol itself has also been shown to be susceptible to an active attack by Gilbert et al. [26].

To overcome both the active attack and the low parameter sizes offered by HB⁺, many variants emerged. The literature counts HB⁺⁺ [11], HB* [19], HB-MP [44] and HB[#] [28], all but last of which was broken by Gilbert et al. [27]. HB⁺'s newest iteration, HB[#] by Gilbert et al. seemed adequate in many respects, however, it too was shown to be susceptible to a man-in-the-middle attack by Ouafi et al. [48].

Until now, every new incarnation of the HB protocol has been broken, which indicates that making a correct variation is exceedingly hard. Furthermore, advancements in solving the LPN problem could prove fatal to not only to a specific iteration, but also to the whole concept. The parameter needs increase for every new LPN-solving algorithm and every attack method, thereby increasing the computation, communication, and storage needs of tags implementing HB-based protocols. This continuous increase could eventually make the concept so resource-intensive to implement that other protocol families would become more suitable.

5.4 Physically Uncloneable Functions

A Physically Uncloneable Function (PUF) is a function that depends on the differences introduced during manufacture to map inputs to outputs. Therefore, for the same input, different physical incarnations of the same circuit yield different outputs. For electronic circuits, the differences exploited are differences in the wire and gate-delays which are difficult to predict, measure and, most importantly, to accurately model [41]. Therefore, by using a suitable function to eliminate the differences introduced by environmental (temperature, pressure, etc.) variations, the output of the PUF can be used as a means to uniquely identify an electronic circuit.

PUFs were originally invented for optical media by Ravinkanth [56], but later they got ported to silicon by Gassend et al. [25]. They have been proposed to be used in RFIDs by Bolotnyy and Gabriel [10] as a means to authenticate the

tag. The protocol proposed is simple: the tag is queried at manufacture for random inputs, the obtained (input,output) pairs are stored, and later used to authenticate the tag. Since the attacker can neither build a sufficiently large (input, output) database due to the size of the input, nor can she model the PUF inside the tag given a number of (input, output) pairs, PUFs seems to be ideal way to authenticate a tag. Although PUFs also promise the low hardware need of only a couple of hundred gate equivalences, we know of no real-world implementation of them on RFIDs, which undermines its verifiability.

6 Protocol properties overview

In this section we provide a list of the previously mentioned protocols and their offered features in a matrix-like fashion.

The matrix of features provided by the previously mentioned protocols is in Table 1. Most of the protocols in the table have a star next to their security features, meaning that that their offered security feature(s) have been shown to be broken. This is expected in the field of computer security, as given enough time, most cryptographic and security protocols are found to have weaknesses: attacks continually get better, they never get worse. It is interesting to see that almost all RFID security protocols that relied on a new primitive, such as ProIP, SQUASH-0, DPM, etc. have had at least some of their offered security features broken. It is also evident from the feature-matrix that the protocol that offers the most features and has stood the test of time is the Molnar-Wagner key-tree protocol.

7 Conclusions

From our recap of the most influential RFID protocols and their derivatives, it is apparent that RFID protocols follow the pattern prevalent in the more general field of computer security: protocols are conceived, refined, attacked, only to be re-born again with improved design to counter the newly found security vulnerabilities. This never-ending cycle of refinement brings to light the fundamental advantages and limits of RFIDs. Based on this refined view of RFID systems, new protocols are conceived that fit the domain ever more perfectly.

There are many different goals in RFID systems: identification speed, authentication, privacy, system-wide resistance to attacks, tag cheapness, etc. Some protocols try to achieve all of these, some only a particular subset. Some of these goals have even been shown to be conflicting: Burmester et al. have shown [12] that less than linear-time (in the number of tags) private identification is not possible without either shared secrets (possibly compromising system-wide resistance) or public-key cryptography (possibly compromising tag cheapness). Even if some protocols will prove to be more useful than others over the course of time, there surely will remain many to be made use of, due to the different trade-offs for the different RFID settings.

Table 1: Overview of the previously mentioned RFID protocols' claimed offered features. Features that have been shown to be broken are clearly marked with a star. Note that HB# can be made resilient to attacks using higher parameters.

Protocol	Unlinkable ident.	Untraceable ident.	Tag auth.	Reader auth.
ISO14443A coll.-avoidance [35]	No	No	No	No
EPC coll.-avoidance [20]	No	No	No	No
Pseudonym-rotation [36]	Yes*	Yes*	No	No
ProbIP [16]	Yes*	Yes*	No	No
OSK [39]	Yes*	Yes*	Yes	No
YA-TRAP [59]	Yes*	Yes*	Yes	No
YA-TRAP+ [13]	Yes*	Yes*	Yes	No
O-TRAP [13]	Yes*	Yes*	Yes	No
RIPP-FS [17]	Yes*	Yes*	Yes	Yes
Molnar-Wagner [43]	Yes	Yes	Yes	Yes
DPM [53]	Yes*	Yes*	Yes	Yes
SQUASH-0 [57]	No	No	Yes*	No
WIPR [47]	No	No	Yes	No
HB+ [37]	No	No	Yes*	No
HB# [28]	No	No	Yes*	No
PUF [10]	No	No	Yes	No

References

- [1] AVOINE, G., AND OECHSLIN, P. A Scalable and Provably Secure Hash-Based RFID Protocol. In *The 2nd IEEE International Workshop on Pervasive Computing and Communication Security - PerSec 2005* (2005), pp. 110–114.
- [2] BABBAGE, S., CANNIERE, C. D., CANTEAUT, A., CID, C., GILBERT, H., JOHANSSON, T., PAAR, C., PARKER, M., PRENEEL, B., RIJMEN, V., ROBshaw, M., AND WU, H. The eSTREAM portfolio. Tech. rep., eStream Project, September 2008.
- [3] BAILEY, D., AND JUELS, A. Shoehorning Security into the EPC Standard. In *International Conference on Security in Communication Networks – SCN 2006* (Maiori, Italy, September 2006), R. De Prisco and M. Yung, Eds., vol. 4116 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 303–320.

- [4] BÁRASZ, M., BOROS, B., LIGETI, P., LÓJA, K., AND NAGY, D. Breaking LMAP. In *Conference on RFID Security – RFIDSec’07* (Malaga, Spain, July 2007), pp. 69–78.
- [5] BÁRASZ, M., BOROS, B., LIGETI, P., LÓJA, K., AND NAGY, D. A. Passive attack against the M2AP mutual authentication protocol for RFID tags. In *RFID 2007 – The First International EURASIP Workshop on RFID Technology* (September 2007).
- [6] BERLEKAMP, E., MCELIECE, R., AND VAN TILBORG, H. On the inherent intractability of certain coding problems (corresp.). *Information Theory, IEEE Transactions on* 24, 3 (May 1978), 384–386.
- [7] BLASS, E.-O., KURMUS, A., MOLVA, R., NOUBIR, G., AND SHIKFA, A. The Ff-Family of Protocols for RFID-Privacy and Authentication. In *Workshop on RFID Security – RFIDSec’09* (Leuven, Belgium, July 2009).
- [8] BLUM, A., KALAI, A., AND WASSERMAN, H. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM* 50, 4 (2003), 506–519.
- [9] BOGDANOV, A., KNUDSEN, L. R., LEANDER, G., PAAR, C., POSCHMANN, A., ROBshaw, M. J., SEURIN, Y., AND VIKKELSOE, C. PRESENT: An ultra-lightweight block cipher. In *Workshop on Cryptographic Hardware and Embedded Systems – CHES 2007* (Vienna, Austria, September 2007), P. Paillier and I. Verbauwhede, Eds., vol. 4727 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 450–466.
- [10] BOLOTNYY, L., AND ROBINS, G. Physically unclonable function-based security and privacy in RFID systems. In *PerCom 2007* (March 2007), IEEE, pp. 211–220.
- [11] BRINGER, J., CHABANNE, H., AND DOTTA, E. HB⁺⁺: a lightweight authentication protocol secure against some attacks. In *Security, Privacy and Trust in Pervasive and Ubiquitous Computing, 2006 – SecPerU 2006* (June 2006), pp. 28–33.
- [12] BURMESTER, M., DE MEDEIROS, B., AND MOTTA, R. Robust, anonymous RFID authentication with constant key-lookup. In *ASIACCS (2008)*, M. Abe and V. D. Gligor, Eds., ACM, pp. 283–291.
- [13] BURMESTER, M., LE, T. V., AND MEDEIROS, B. D. Provably secure ubiquitous systems: Universally composable RFID authentication protocols. In *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm ’06* (Baltimore, Maryland, USA, August-September 2006), IEEE.
- [14] BUTTYÁN, L., HOLCZER, T., AND VAJDA, I. Optimal key-trees for tree-based private authentication. In *Workshop on Privacy Enhancing*

- Technologies – PET 2006* (Cambridge, United Kingdom, June 2007), pp. 332–350.
- [15] CANNIÈRE, C. D. Trivium: A stream cipher construction inspired by block cipher design principles. In *ISC (2006)*, S. K. Katsikas and et al, Eds., vol. 4176 of *LNCS*, Springer, pp. 171–186.
 - [16] CASTELLUCCIA, C., AND SOOS, M. Secret shuffling: A novel approach to RFID private identification. In *RFIDSec'07 (July 2007)*, pp. 169–180.
 - [17] CONTI, M., PIETRO, R. D., MANCINI, L. V., AND SPOGNARDI, A. RIPPFS: an RFID Identification, Privacy Preserving Protocol with Forward Secrecy. In *International Workshop on Pervasive Computing and Communication Security – PerSec '07* (New York City, New York, USA, March 2007), IEEE, IEEE Computer Society Press, pp. 229–234.
 - [18] CRAWFORD, J. M., KEARNS, M. J., AND SHAPIRE, R. E. The minimal disagreement parity problem as a hard satisfiability problem. Tech. rep., Computational Intelligence Research Laboratory and AT&T Bell Labs, February 1994.
 - [19] DUC, D., AND KIM, K. Securing HB⁺ against GRS man-in-the-middle attack. *Institute of Electronics, Information and Communication Engineers, Symposium on Cryptography and Information, Security* (2007).
 - [20] EPCGLOBAL. 13.56 MHz ISM band class 1 radio frequency identification tag interface specification (2003). Tech. rep., Auto-ID center, MIT, February 2003.
 - [21] FELDHOFFER, M., AND RECHBERGER, C. A case against currently used hash functions in RFID protocols. In *OTM Workshops (1)* (2006), R. Meersman, Z. Tari, and P. Herrero, Eds., vol. 4277 of *Lecture Notes in Computer Science*, Springer, pp. 372–381.
 - [22] FELDHOFFER, M., AND WOLKERSTORFER, J. Strong crypto for RFID tags - a comparison of low-power hardware implementations. *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on* (May 2007), 1839–1842.
 - [23] FELDHOFFER, M., WOLKERSTORFER, J., AND RIJMEN, V. AES implementation on a grain of sand. In *Information Security* (2005), IEEE, pp. 13–20.
 - [24] FOSSORIER, M. P. C., MIHALJEVIĆ, M. J., IMAI, H., CUI, Y., AND MATSUURA, K. A novel algorithm for solving the LPN problem and its application to security evaluation of the HB protocol for RFID authentication. In *INDOCRYPT* (2006), R. Barua and T. Lange, Eds., vol. 4329 of *Lecture Notes in Computer Science*, Springer, pp. 48–62.

- [25] GASSEND, B., CLARKE, D., VAN DIJK, M., AND DEVADAS, S. Controlled physical random functions. In *Proceedings of the 18th Annual Computer Security Applications Conference – ACSAC '02* (2002), IEEE.
- [26] GILBERT, H., ROBSHAW, M., AND SIBERT, H. An active attack against HB^+ - a provably secure lightweight authentication protocol. In *IEE Electronic Letters* 41, 21 (2005), pp. 1169–1170.
- [27] GILBERT, H., ROBSHAW, M. J., AND SEURIN, Y. Good variants of HB^+ are hard to find. In *Financial Cryptography* (January 2008), Springer.
- [28] GILBERT, H., ROBSHAW, M. J. B., AND SEURIN, Y. $HB^\#$: Increasing the security and efficiency of HB^+ . In *Advances in Cryptology – EUROCRYPT '08* (2008), N. P. Smart, Ed., vol. 4965 of *Lecture Notes in Computer Science*, Springer, pp. 361–378.
- [29] GIRAULT, M. Self-certified public keys. In *Advances in Cryptology – EUROCRYPT '91* (1991), pp. 490–497.
- [30] GIRAULT, M. Low-size coupons for low-cost IC cards. In *CARDIS* (2000), J. Domingo-Ferrer, D. Chan, and A. Watson, Eds., vol. 180 of *IFIP Conference Proceedings*, Kluwer, pp. 39–50.
- [31] GIRAULT, M., AND LEFRANC, D. Public key authentication with one (online) single addition. In *Cryptographic Hardware and Embedded Systems - CHES 2004* (2004), vol. 3156/2004 of *Lecture Notes in Computer Science*, pp. 967–984.
- [32] HELL, M., JOHANSSON, T., AND MEIER, W. Grain - a stream cipher for constrained environments. In *Proceeding of the Workshop on RFID and Lightweight Crypto* (July 2005), pp. 114–125.
- [33] HELLMAN, M. E. A cryptanalytic time-memory trade off. In *IEEE Transactions on Information Theory* (1980), vol. IT-26/4, pp. 401–406.
- [34] HOPPER, N. J., AND BLUM, M. Secure human identification protocols. In *ASIACRYPT '01: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security* (London, UK, 2001), Springer-Verlag, pp. 52–66.
- [35] ISO/IEC. 14443-3 – Identification cards – Contactless integrated circuit(s) cards – Proximity cards – Part 3: Initialization and anticollision, 2001, Stage: 90.92 – 2007-12-11.
- [36] JUELS, A. Minimalist cryptography for low-cost RFID tags. In *International Conference on Security in Communication Networks – SCN 2004* (Amalfi, Italia, September 2004), C. Blundo and S. Cimato, Eds., vol. 3352 of *LNCS*, Springer-Verlag, pp. 149–164.

- [37] JUELS, A., AND WEIS, S. Authenticating pervasive devices with human protocols. In *Advances in Cryptology – CRYPTO’05* (Santa Barbara, California, USA, August 2005), V. Shoup, Ed., vol. 3126 of *LNCS*, IACR, Springer-Verlag, pp. 293–308.
- [38] JUELS, A., AND WEIS, S. Defining Strong Privacy for RFID. In *International Conference on Pervasive Computing and Communications – PerCom 2007* (New York City, New York, USA, March 2007), IEEE, IEEE Computer Society Press, pp. 342–347.
- [39] KOUTAROU, M. O., SUZUKI, K., AND KINOSHITA, S. Cryptographic approach to ”privacy-friendly” tags. In *RFID Privacy Workshop* (MIT, Massachusetts, USA, November 2003).
- [40] LAMPORT, L. Password authentication with insecure communication. *Commun. ACM* 24, 11 (1981), 770–772.
- [41] LIM, D., LEE, J. W., GASSEND, B., SUH, G. E., VAN DIJK, M., AND DEVADAS, S. Extracting secret keys from integrated circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* (2005), 1200–1205.
- [42] MCLOONE, M., AND ROBSHAW, M. J. B. Public key cryptography and RFID tags. In *CT-RSA* (2007), M. Abe, Ed., vol. 4377 of *Lecture Notes in Computer Science*, Springer, pp. 372–384.
- [43] MOLNAR, D., AND WAGNER, D. Privacy and security in library RFID: issues, practices, and architectures. In *CCS ’04: Proceedings of the 11th ACM conference on Computer and communications security* (New York, NY, USA, 2004), ACM Press, pp. 210–219.
- [44] MUNILLA, J., AND PEINADO, A. HB-MP: A further step in the hb-family of lightweight authentication protocols. *Comput. Netw.* 51, 9 (2007), 2262–2267.
- [45] NOHL, K., AND EVANS, D. Hiding in Groups: On the Expressiveness of Privacy Distributions. In *Proceedings of The Ifip Tc 11 23rd International Information Security Conference – SEC 2008* (Milan, Italia, September 2008), vol. 278 of *Lecture Notes in Computer Science*, Springer, pp. 1–15.
- [46] OECHSLIN, P. Making a faster cryptanalytic time-memory trade-off. In *Advances in Cryptology – CRYPTO 2003* (2003), vol. 2729 of *Lecture Notes in Computer Science*, Springer, pp. 617–630.
- [47] OREN, Y., AND FELDHOFFER, M. WIPR - a public key implementation on two grains of sand. In *Workshop on RFID Security 2008* (2008), S. Dominikus, Ed., pp. 15 – 27.
- [48] OUAFI, K., OVERBECK, R., AND VAUDENAY, S. On the security of HB# against a man-in-the-middle attack. In *Advances in Cryptology – Asiacrypt 2008* (Melbourne, Australia, December 2008), vol. 5350 of *Lecture Notes in Computer Science*, Springer, pp. 108–124.

- [49] OUAFI, K., AND PHAN, R. C.-W. Privacy of Recent RFID Authentication Protocols. In *Information Security Practice and Experience, 4th International Conference, ISPEC 2008* (Berlin, 2008), Lecture Notes in Computer Science, Springer, pp. 263–277.
- [50] OUAFI, K., AND VAUDENAY, S. Smashing SQUASH-0. In *Advances in Cryptology – EUROCRYPT ’09* (April 2009), vol. 5479 of *LNCS*, IACR, pp. 300–312.
- [51] PERIS-LOPEZ, P., HERNANDEZ-CASTRO, J. C., ESTEVEZ-TAPIADOR, J., AND RIBAGORDA, A. LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags. In *Proceedings of RFIDSec’06* (Graz, Austria, July 2006), Ecrypt.
- [52] PERIS-LOPEZ, P., HERNANDEZ-CASTRO, J. C., ESTEVEZ-TAPIADOR, J., AND RIBAGORDA, A. M2AP: A minimalist mutual-authentication protocol for low-cost RFID tags. In *International Conference on Ubiquitous Intelligence and Computing – UIC’06* (September 2006), vol. 4159 of *LNCS*, Springer-Verlag, pp. 912–923.
- [53] PIETRO, R. D., AND MOLVA, R. Information confinement, privacy, and security in RFID systems. In *Proceedings of the 12th European Symposium On Research In Computer Security* (September 2007), pp. 187–202.
- [54] POSCHMANN, A., LE, G., SCHRAMM, K., AND PAAR, C. A family of light-weight block ciphers based on DES suited for RFID applications. In *Proceedings of FSE 2007, LNCS* (2006), Springer-Verlag.
- [55] RABIN, M. O. Digitalized signatures and public-key functions as intractable as factorization. Tech. rep., Massachusetts Institute of Technology, Cambridge, MA, USA, 1979.
- [56] RAVINKANTH, P. Physical one-way functions. Tech. rep., MIT, 2001. Ph.D. Thesis.
- [57] SHAMIR, A. SQUASH - a new MAC with provable security properties for highly constrained devices such as RFID tags. In *FSE* (2008), K. Nyberg, Ed., vol. 5086 of *Lecture Notes in Computer Science*, Springer, pp. 144–157.
- [58] SOOS, M. Analysing the Molva and Di Pietro Private RFID Authentication Scheme. In *Workshop on RFID Security – RFIDSec’08* (Budapest, Hungary, July 2008).
- [59] TSUDIK, G. YA-TRAP: Yet another trivial RFID authentication protocol. In *International Conference on Pervasive Computing and Communications – PerCom 2006* (Pisa, Italy, March 2006), IEEE, IEEE Computer Society Press, pp. 640–643.

- [60] VAUDENAY, S. On privacy models for RFID. In *Advances in Cryptology – Asiacrypt 2007* (Kuching, Malaysia, December 2007), vol. 4833 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 68–87.
- [61] ÉRIC LEVIEIL, AND FOUQUE, P.-A. An improved LPN algorithm. In *Security and Cryptography for Networks – SCN (2006)*, R. D. Prisco and M. Yung, Eds., vol. 4116 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 348–359.