Cryptographic Puzzles and Distance-bounding Protocols: Practical Tools for RFID Security

Pedro Peris-Lopez
Security Lab, Faculty of EEMCS,
Delft University of Technology,
Mekelweg 4, 2628 CD,
Delft, The Netherlands.

Julio C. Hernandez-Castro
School of Computing,
University of Portsmouth,
Buckingham Building, Lion Terrace,
Portsmouth PO1 3HE, United Kingdom

Juan M. E. Tapiador
Department of Computer Science,
University of York,
Heslington, York,
YO10 5DD, United Kingdom

Esther Palomar
Department of Computer Science,
Carlos III University of Madrid,
Avenida de la Universidad, 30,
Legans (Madrid), E-28911, Spain

Jan C.A. van der Lubbe
Security Lab, Faculty of EEMCS,
Delft University of Technology,
Mekelweg 4, 2628 CD,
Delft, The Netherlands.

Abstract—**Widespread adoption of RFID technology is slowing down because of increasing public concerns about associated security threats. This paper shows that it is possible to enhance the security of RFID systems by requiring readers to perform a computational effort test. Readers must solve a cryptographic puzzle –one of the components of the Weakly Secret Bit Commitment (WSBC) sent by tags– to obtain the static identifier of the interrogated tag. The method we present is based on a simple concept already used in such cryptographic applications as anti-spam programs or TCP SYN flooding protection, yet not in an RFID context until now. The scheme provides privacy protection while being an effective countermeasure against the indiscriminate disclosure of the whole contents of a large number of tags. Then, we scrutinize the combined use of cryptographic puzzles and distance-bounding protocols. First, a classical and straight-forward solution is presented. Secondly, we introduce a cutting-edge approach that reduces WSBC drawbacks such as key delegation whilst gaining the advantages of employing distance-bounding protocols such as certainty of the distance between a tag and reader.**

Index Terms—**RFID security, WSBC, cryptographic puzzles, distance-bounding protocols, privacy, traceability**

I. Introduction

Radio-Frequency Identification (RFID) provides a means to identify items (i.e. persons, animals or products) to which an RFID tag is attached. Specifically, an RFID system is composed of three main components: 1) Readers; 2) Tags; 3) Back-end database. Readers (transceivers) interrogate tags (transponders) to access the information stored in their memory. Afterwards, they pass this information acquired to a back-end database which uses it as a search index to locate all the information associated with the target tag. Readers and tags use the radio channel for communication, which is commonly assumed to be insecure. However, as readers and the back-end database are computationally much more powerful than tags, a secure channel is assumed (possibly using classical protocols such as SSL) for communication between these two entities.

RFID technology may be seen as the replacement for barcodes. Yet widescale of this technology is being delayed because of associated security risks [13], [24], [29]. To understand these, let us consider a simple example. Suppose that an exclusive clothing manufacturer tags all his garments. For the manufacturer, using RFID technology improves stock control, allowing him to monitor all aspects of his stock, virtually in real time, and track it efficiently. In addition, work-hours required for restocking are significantly reduced in comparison when a barcode solution is used [12], so the advantages are obvious. However, using RFID also has negative implications. An attacker –say a rival manufacturer– could also scan the state of the manufacturer's stock and acquire commercially valuable information that should have been kept secret, simply by passing close enough to the warehouse with an inexpensive RFID reader. Additionally, imagine that you buy a jacket and trousers of the same brand. An attacker could read these tags and classify you as wealthy and very attractive victim, as the brand of clothes is part of the unique identifier of a tag. Information about consumer habits is at stake. Finally, as a static identifier is often provided each time a tag is read, your tags may be easily linked to you. Afterwards, if you were scanned many times by different readers (e.g. several readers in a supermarket), your movements could be tracked and recorded.

Our contribution: In this paper, we propose a simple RFID scheme to protect privacy. Specifically, information privacy and untraceability of tags are guaranteed. In fact, the proposed scheme is a simple yet effective countermeasure against massive inventory disclosure. Moreover, it can be a useful deterrent against counterfeiting, which currently one of the main concerns for many (e.g. clothing or drug) manufacturers. We present an innovative scheme based on Weakly Secret Bit Commitment (WSBC), which requires the solution of a cryptographic puzzle. Cryptographic puzzles are a well-known technique, but this is the first time -to the best of our knowledge- that its use is proposed in the context of RFID systems. Despite the numerous advantages of this solution, in certain scenarios the issue of key delegation must be addressed to guarantee the feasibility of our proposal. Nevertheless, the combined use of cryptographic puzzles and distance-bounding protocols can overcome this drawback. Specifically, we propose a novel protocol in which the hardness of the puzzle depends on the distance measured by the tag. This approach differs from that of standard approaches in which a tag plays the role of prover instead of verifier. The solution, here named in the paper "Classical Extension", follows the standard solution, as opposed to the "Cutting-edge Extension" in which the reader and tag reverse their commonly assumed roles, verifier and prover, respectively.

The remainder of the paper is organized as follows. Section II outlines the motivation of this work. In Section III, we propose an RFID authentication protocol based on WSBC. Performance and security analysis are presented in Section IV and V respectively. In Section VI, we integrate our first protocol with distance-bounding protocols (classical and cutting-edge extension). Finally, we extract conclusions in Section VII.

II. Motivation

The need to guarantee the authenticity of the parties involved in an RFID identification process and the critical nature of the information that is at stake are encouraging the use of standard cryptography despite the severe hardware limitations of low-cost RFID tags [1]. For instance, recent secure distance-bounding protocols prove to the reader that the tag is a certain distance away from the reader by means of a time-critical (single-bit) challenge-response mechanism [4], [6], [20], [25]. Such schemes basically use a pseudo-random function and a shared secret key, known by both parties. Additionally, these schemes can be integrated into authentication schemes as suggested in [6] and recently in [26].

Yet the level of security strength of a certain protocol does not depend exclusively on the cryptographic primitives used, but sometimes on whether an adversary can successfully (in time and from a given distance) break the system. It is precisely with the aims of extending the RFID identification model and providing a countermeasure against the indiscriminate disclosure of tag contents that our approach is proposed.

The idea of using cryptographic proof-of-work protocols to increase the cost of sending all emails so that sending spam becomes unprofitable [15] or to increase the cost of leaving a TCP connection half open to hamper the achievement of a connection depletion attack could also be extended to discourage misbehavior in RFID systems. In a basic (completely insecure) identification scheme, first the reader sends a $\{Request\}$ message to the tag and then the tag backscatters its static identifier $\{ID\}$ to the reader. As an alternative, we present a method based on a simple concept, as follows:

$$\begin{array}{ll} Reader \rightarrow Tag: & Request \\ Tag \rightarrow Reader: & Puzzle(ID) \end{array} \qquad (1)$$

The idea is that RFID readers which do not devote the required time and computational effort to solve the puzzle will not access any relevant identification material. Tags will generate puzzles, while readers must solve them in order to identify tags, after which process readers will possess the information previously ciphered and anonymized, i.e. the tag identifier. However, in this straightforward solution, rogue readers and honest readers have to make the same effort to solve the cryptographic puzzle. In this paper, two solutions are proposed in order to tackle this drawback. First, the back-end database could delegate to legitimate readers –after mutual authentication between both entities– part of the tag's secret key (see Section V for details). Secondly, the hardness of the puzzle could depend on the distance between the reader and the tag. Specifically, a distance-bounding protocol is employed to gain knowledge about the distance between the prover and the verifier. As legitimate readers are in close proximity and dishonest readers are often distant, the former receive much simpler puzzles than the latter, as described in Section VI-B. So, taking advantage of key delegation or distance checking mechanism, honest readers can identify a voluminous population of tags in an effective and affordable way. On the contrary, rogue readers find this issue –identification of numerous tags– to be an impossible mission.

III. A Puzzle-Based RFID Authentication Protocol

Having briefly presented an outline of the proposal in the previous section, we now describe its implementation in more depth. The scheme is illustrated in Fig. 1. $\mathcal{R}$ and $\mathcal{T}$ denote the two protocol parties, reader and tag respectively. Regarding communications, we assume that readers are connected to a back-end database through a secure channel, in contrast with the insecurity of both the forward (reader-to-tag) and backward (tag-to-reader) channels. That is, the communication channel between readers and tags is assumed to be insecure. We also assume that the ID is the information these two entities would like to exchange securely, where ID symbolizes the unique identification number of the tag. Moreover, $enc_k(x)$ is a symmetric key algorithm (e.g. the block cipher AES [16] or TEA [23]) that encrypts message x under key k. The concatenation of variables is denoted by the symbol $||$. Let $\varsigma_j = enc_k(n||ID||n||j)$ represent the cryptographic puzzle sent by $\mathcal{T}$ at the j-th protocol instance, where n is a random number. The combined use of this nonce and the encryption algorithm facilitates tag identification, providing anonymity and privacy protection, as shown in Section V.

Likewise, $\omega_j^{\pi}(k)$ represents a WSBC function, i.e. a trapdoor. A bit commitment is a means of requiring an entity to commit to a value, while keeping it hidden until revealing its value at a later point. For example, Alice generates two nonces $\{R_1, R_2\}$ and commits to a message M by computing $h(R_1||R_2||M)$ and sending $R_1, h(R_1||R_2||M)$ to Bob. When she wants to reveal M to Bob, she sends $R_2||M$. By the properties of hash function h, Bob cannot determine M and Alice cannot find a different value for M' such that $h(R_1||R_2||M) = h(R_1||R_2||M')$. WSBC functions work on the same principle, but with the noticeable difference that the

1. $\mathcal{R} \rightarrow \mathcal{T}$: $m_1 = request, n_1$
2. $\mathcal{T} \rightarrow \mathcal{R}$: $m_2 = n_2, \langle \varsigma_j, \omega_j^{\pi}(k) \rangle, \upsilon_j, \nu_j^*$
3. $\mathcal{R} \rightarrow \mathcal{T}$: $m_3 = n_4^*, \tau_j^*$ (*Optional)

where $\{n_i\}_{i=0}^{4}$ are different *nonces*
$\varsigma_j = enc_k(n_1||ID||n_1||j)$
$\omega_j^{\pi}(k) = \{k_{\pi(0)}, k_{\pi(1)}, \ldots, k_{\pi(l-1)}\}$ is a *l-bit WSBC function* and $\pi()$ is a given permutation
$\upsilon_j = h(j||n_1||ID||n_2)$
$\nu_j^* = enc_k(j||n_3||ID||n_1)$ (Optional)
and $\tau_j^* = enc_k(j||n_4||ID+1||n_3||n_1)$ (Optional)

Fig. 1. WSBC Authentication Scheme

secrecy of the bit commitment is breakable after an acceptable predefined limit in terms of time and/or computation. 2^{nd} preimage resistance and weak-preimage resistance are the general properties that a WSBC function $\omega()$ should have. Additionally, collision resistance and near-preimage resistance [27] may be required, depending on the specific application. In [35], readers may consider detailed introduction to WSBC functions. Specifically, the WSBC we suggest for the ID is simply $\langle \varsigma_j, \omega_j^{\pi}(k) \rangle$. Solutions such as time-lock puzzles, which encrypt k with the result of repeatedly squaring a value with respect to a composite module may be a very natural implementation for $\omega()$. However, we are obliged to choose a much simpler solution due to the severe restrictions of low/moderate-cost tags, which are the most suitable for these kinds of solutions. Specifically, the tag randomly selects l bits of k, and this collection of bits forms $\omega_j^{\pi}(k)$. For high-cost tags that possess superior computational capabilities, however, we find the use of strong solutions such as those outlined previously, more convenient.

Finally, $h(a||b)$ is a hash function whose input is the concatenation of a and b values. Specifically, $\upsilon_j = h(j||n_1||ID||n_2)$ is the pseudonym sent by the tag at the j-th identification process which mainly has the role of allowing the verification of the puzzle solution after the reader completes the operation. Generally, a pseudonym transmits the static identifier of a tag with the guarantee of keeping confidential information secret and ensuring the untraceability of tag responses [37]. We explain the steps of the protocol below:

1. $\mathcal{R} \rightarrow \mathcal{T}$: $m_1 = request, n_1$

$\mathcal{R}$ starts the protocol by sending a request message to $\mathcal{T}$ which includes a random number n_1 (Fig. 1–message m_1). This message aims to wake up the tag and energize the chip in case that those tags do not support an onboard energy source (passive and semi-passive transponders). Additionally, nonce n_1 is a necessary component to guarantee freshness of the session and is also useful as a session identifier.

2. $\mathcal{T} \rightarrow \mathcal{R}$: $m_2 = n_2, \langle \varsigma_j, \omega_j^{\pi}(k) \rangle, \upsilon_j, \nu_j^*$

Afterwards, $\mathcal{T}$ generates a new random number n_2. Then, a commitment $\langle \varsigma_j, \omega_j^{\pi}(k) \rangle$ and a pseudonym υ_j to ID are computed. All these values form message m_2 which is finally passed to $\mathcal{R}$ (Fig. 1–message m_2).

On receiving message m_2, $\mathcal{R}$ gets the cryptographic puzzle ς_j which forms part of the commitment sent by $\mathcal{T}$. The remaining part of the commitment is the output of the WSBC function $\omega_j^{\pi}(k)$, which facilitates the solution of the puzzle in a bounded time. Specifically, once $\omega_j^{\pi}(k)$ is received, l bits of the secret key k are delegated to $\mathcal{R}$. Note that $\mathcal{T}$ may provide many different $\omega_j^{\pi}(k)$ values by randomly selecting different l bits of the whole key. Finally, a brute-force process is initialized and $\mathcal{R}$ must try, on average, 2^{n-l-1} keys –n being the number of bits of key k– to decipher ς_j. Specifically, each time a new key is checked, the success of the search is verified by the use of the tag pseudonym υ_j included at the end of message m_2:

$$\upsilon_j \overset{?}{=} h(j||enc_k^{-1}(n_1||ID||n_1||j) >> p||n_2) \quad (2)$$

where $>>$ symbolizes logical shift right and p is the total bit length of n_1 and j.

If Equation 2 holds, it means that $\mathcal{R}$ has solved the puzzle correctly and knows the key and the static identifier associated with $\mathcal{T}$. Otherwise, $\mathcal{R}$ has to try with another secret key.

3. $\mathcal{R} \rightarrow \mathcal{T}_j$: $m_3 = n_4^*, \tau_j^*$ (*Optional)

Finally, $\mathcal{R}$ can prove to $\mathcal{T}$ that he knows the correct solution to the puzzle (Fig. 1–message m_3) and has then been able to acquire its private information. For that, the reader first obtains challenge n_3^* by deciphering ν_j^* and exploiting his knowledge of the secret key k. Secondly, $\mathcal{R}$ generates a random nonce n_4^* and computes τ_j^* and sends it to $\mathcal{T}$. On reception, $\mathcal{T}$ computes its local version of $\tau_j^* = enc_k(j||n_4||ID+1||n_3||n_1)$ and compares it with the received value. If verification is successful, $\mathcal{R}$ is authenticated, and the mutually authentication process is completed. We emphasize that nonces $\{n_3^*, n_4^*\}$ and messages $\{\nu_j^*, \tau_j^*\}$ are considered optional in our protocol description and are only required when mutual authentication (reader↔tag) is required by the application for which the protocol is destined.

In all the above, the underlying idea is that $\mathcal{T}$ will try to exhaust the computational resources of potential rogue readers ($\overline{\mathcal{R}}$) because it is highly likely to interact with them. If readers interact indiscriminately with tags, these transceivers would suffer computational overload should they attempt to solve all the puzzles received in parallel ($\overline{\mathcal{R}}$ has to test, on average, $2^{(n-l-1)^2}$ keys). On the contrary, if they solve the puzzles sequentially, the time consumed will be tremendous and finally they will either give up or achieve only very limited success, with potentially far fewer tags being tamper with. Honest readers would suffer the same drawback, if no additional mechanism is used. We study two alternatives to overcome this defect: 1) The use of key delegation techniques

(Section V) ; 2) The combination of cryptographic puzzles and distance-bounding protocols (Section VI-B). In the first approach, honest readers know part of the secret key of the tags before the reception of the puzzles. Equivalently, in the second approach honest readers receive much more simpler puzzles than the received by rouge readers due to its proximity to the tags. Using one of these strategies, honest readers possess a significant advantage over rogue readers, and can work with a numerous population of tags.

TABLE I
AVERAGE COMPUTATIONAL EFFORT MADE (IN 1000 EXPERIMENTS) BY EACH ALGORITHM FOR VARYING AMOUNTS OF KNOWN BITS.

$n-l$ bits	l bits	**Average time required (sec)**	
		AES-128	TEA
32	96	5495	761
28	100	544	47
24	104	15	0.22
20	108	0.01	0.01

IV. PERFORMANCE ANALYSIS

In this section, we estimate the computational effort required by readers to solve the cryptographic puzzle enclosed in the WSBCs. As RFID readers are much more powerful than tags in terms of computation and storage capability, we focus on the time consumed in each identification. Optimization of this factor is one of the main objectives for any identification system. A trade-off between security (i.e. inventory protection) and system performance is thus necessary.

We considered two block ciphers as the basis for cryptographic puzzles, AES-128 and TEA [11], [33]. Both were coded in C, compiled with Microsoft Visual C++, and run on an AMD ATHLON(tm)2600 2.09GHz processor, with 1GB RAM and under Windows XP SP2. Further simulations in real RFID readers may be convenient, but the results already obtained will give an adequate idea of the relative values, and are valid to perform comparisons.

A factor contributing to complexity is the cost of executing several decryptions, for testing each candidate key. Specifically, the reader receives $\langle \varsigma_j, \omega_j^{\pi}(k) \rangle$, where ς_j and $\omega_j^{\pi}(k)$ represent the cryptographic puzzle and the output of the WSBC function, respectively. The reader then starts an exhaustive search; it probes all possible keys and benefits from the knowledge of l key bits for each. The above process is repeated until the correct key is found.

We have carried out 1000 experiments for different values of $(n-l)$-bits, and also randomly varying the challenges and key used. We consider that more than 32 hidden bits would be impractical, e.g. 2^{64} is a too large number of potential keys to test; it would literally take years on average processors. Results are shown in Table I, for a key length of $n = 128$. For each case, as the l value increases, the number of candidate keys obviously decreases, so the exploration time too. For practical considerations, the particular requirements of the application in which the protocol is used will determine the choice of the l value.

V. SECURITY ANALYSIS

In this section we scrutinize the security properties of the proposed protocol. The two main objectives of our protocol are privacy protection and untraceability. Regarding privacy, the static identifier of the tag is never sent in clear on the channel. Specifically, an encrypted version $\varsigma_j = enc_k(n_1||ID||n_1||j)$ of the ID, which requires the knowledge of the secret key k for its computation is used for puzzle generation. The puzzle is accompanied by a pseudonym $\upsilon_j = h(j||n_1||ID||n_2)$ of the tag's ID which is used for puzzle verification without compromising any confidential information. Additionally, part of the secret key $\omega_j^{\pi}(k)$ is delegated from $\mathcal{T}$ to $\mathcal{R}$. This cannot be exploited by an attacker as different l bits of the key are randomly selected and employed in each iteration. Specifically, the tag randomly picks up one of the WSBCs' possible $C(n,l)$ values, where n and l are the bit lengths of the key and the WSBC function respectively. Where the $C(n,l)$ value may be considered poor in terms of security (e.g. $< 2^{32}$ for low-cost RFID tags and $< 2^{64}$ for moderate-cost tags), we recommend updating of the key as frequently as possible -see below for a detailed explanation about the updating process. Additionally, to offer traceability protection, the freshness of the exchanged messages is provided by the nonces generated by the reader and the tag and used in each sub-message generation $\{\varsigma_j, \upsilon_j, \nu_j^*, \tau_j^*\}$. An attacker cannot distinguish between the answers from different tags, thus guaranteeing users' location privacy.

Confidential information and location information are delegated to readers -even to rogue readers- once a cryptographic puzzle has been solved. Firstly, the rogue reader can acquire the private information linked to the tag (i.e. $\{ID, k\}$), which represents a data privacy invasion. Secondly, the responses of this tag can be uniquely identified using the captured information, compromising the tag holder's privacy location. We do not believe these issues pose a significant risk as the main application of our protocol is protection against the revelation of the contents of a great number of tags (e.g. a clothing manufacturer's inventory or the stock of books in a library). So an attacker can compromise the privacy of an specific tag but would fail to discover all the information associated with a group of tags. If private information is not compromised, tracking this group of tags is in vain as the attacker cannot distinguish between responses made by different tags.

One of the most important advantages of the proposed scheme is that tags do not need to be registered in the system. The key of each tag can be updated each time a tag is read because readers do not need to know this information to identify a tag. In fact, the reader identifies the tag ID and discovers its private information k after solving the puzzle. This property is very useful in scenarios where the members of the systems change continuously or where delegation is not possible. From now on, we refer to this kind of scenarios as open environments. Additionally, tags in open environments could be registered in the system by honest readers after the mutual authentication phase between readers and the back-end database. Alternatively, tags can be registered in the back-end database initially. In such closed environments, sub-message ν_j^* and message m_3 are included in the protocol where mutual authentication is necessary. The back-end database stores $\{ID, k\}$ for each tag and perhaps some additional information associated with the tagged item. In this case, once the reader solves the puzzle, it sends certain private information held by target tag to the back-end database (e.g. ID). The database checks if the information corresponds to a registered tag. If the item exists in the system, the tag is authenticated; otherwise, an error message is sent to the reader in order to abort the protocol. After completion of the tag authentication phase, the reader sends message m_3 to the tag if reader authentication is necessary. The reader proves knowledge of the private information related to the tag $\{ID, k\}$ by sending $\tau_j^* = enc_k(j||n_4||ID+1||n_3||n_1)$ as part of the m_3 message. The tag computes its local version of τ_j' and checks it against the received value. If $\tau_j \overset{?}{=} \tau_j'$, the reader is authenticated and the mutual authentication process finished.

In some scenarios it is desirable that past communications are protected even when the content of the tag is revealed (backward security property [18]). Updating of the secret information associated with the tag is necessary to achieve this objective. In open environments, the issue can be solved easily. For example, each time a tag is read, the tag's secret key is updated (i.e. $k^n = h(k^{n-1})$). In closed environments, both tags and back-end database must update the private information shared, its synchronization being crucial. After sending message m_3, the back-end database updates the private information held by the authenticated tag. Tag updating is not performed until the reader is authenticated by checking message m_3. However, certain additional precautions have to be taken in order to avoid de-synchronization attacks caused by the interception or alteration of message m_3. To avoid such attacks, the back-end database has an extra storage requirement; it must store a copy of the old and new values of all the information that is updated [8], [22] (e.g. $\{k^{n-1} = k^n, k^n = h(k^n)\}$).

Another important aspect regarding the usage of RFID tags is resiliency to cloning attacks. The proposed scheme can be viewed as a countermeasure against these. An attacker can clone a particular tag after solving the cryptographic puzzle sent by it. However, the success ratio of this attack is zero when the number of tags is increased because of the excess time consumed in solving all the puzzles. It may seem to readers of this paper that honest transceivers would suffer the same problem. However, honest readers (in closed environments) have access to the back-end database which can provide them with part of the key (e.g delegation of p-bits of the whole key) of those tags registered in the system. Delegation technique was first introduced by Molnar *et al.* [30] within the framework of RFID technology and was recently revised in [14], [17]. As readers and database are connected by a secure channel, exchange of part of the key does not imply any security risk and it is usually performed after mutual authentication between the two devices. Due to this secure delegation of part of the key, the space search for legitimate readers (2^{n-pl-1}) is dramatically reduced compared to that for rogue readers (2^{n-l-1}), pl being the addition of bits provided to honest readers by the database and the bits provided by the WSBC function. Honest transceivers are thus able to discover a complete inventory in a reduced amount of time ($2^{(n-pl-1)^2} << 2^{(n-l-1)^2}$).

TABLE II
PRACTICAL CONSIDERATIONS ABOUT IMPLEMENTATION

	Low-cost RFID tags	Moderate-cost RFID tags
Encryption (cryptographic puzzle)	TEA [23]	AES-128 [16]
Hash function (pseudonym)	H-PRESENT-128 [5]	SHA-1 [32]
PRNG (anonymity + WSBC function)	LAMED	Grain [21] (counter mode)
Total GE	4-7K GE	8-12K GE

Finally, we review the mandatory hardware demands for implementation of the proposed schemed. An encryption function is employed for the generation of the cryptographic puzzle. For that, we find symmetric cryptography convenient rather than asymmetric cryptography, which may be appropriate for high-cost RFID tags (e.g. e-passports [3], [31]). To facilitate verification of the correct solution to the puzzle, an anonymized version of the tag's identifier is used. Specifically, for pseudonym generation, we opt for the use of a hash function, one of the most common solutions in the literature [7], [9], [10], [28], [34], [37], [38]. Generation of random numbers is necessary to avoid traceability and replay attacks. Additionally, random numbers are also used for selection of the l bits that constitute WSBC function output. As many different primitives can be selected for these purposes, we suggest various options in Table 2. We make a rough distinction between low-cost RFID tags and moderate-cost RFID tags. To clarify this distinction, we include the number of Gates Equivalent (GE) for each of these alternatives at the bottom of the table.

VI. PROTOCOL EXTENSIONS

One of the advantages of RFID technology is that tags can be read from a distance of several meters using radio waves. In some applications, it can be useful for readers to know how far away the tags that they just read are. For example, readers in the access control system of a building would like to be assured that each tag, and therefore the person who possesses it, is no more than a few centimeters from the access control station –where the reader is located– when the tag is read.

Distance-bounding protocols solve this problem by timing the delay between sending out a challenge bit and receiving back the corresponding answer. In 1994, Brands and Chaum introduced this technique that allows one of the parties to determine an upper bound on the physical distance from the other [6]. In the RFID context, the work by Hancke and

Kuhn [19] is commonly cited and can be considered a very interesting proposal. However, this protocol is vulnerable to fraud attacks, with a success probability of $(3/4)^t$ for an adversary [26], t representing the number of exchanged bits. Additionally, the scheme is not immune to terrorist attacks [36]. We recommend reading [2], [4] which discuss the most recent advances in this area of research.

In this section, we present two original protocols that combine the use of cryptographic puzzles (authentication) and distance-bounding protocols (distance checking). According to distance-bounding protocols, our proposed protocols are inspired in the ideas suggested by Brands and Chaum. In the protocol objectives, mafia fraud attacks are thus considered and terrorist attacks are disregarded. We are currently working on certain ideas to prevent this second type of fraud too. For authentication purposes, the proposed schemes are completely inspired on the puzzle-based RFID authentication protocol introduced in Section III. So, the performance and security analysis conducted in Sections IV and V are applicable regarding to authentication.

A. Classical Extension

In this protocol, the reader/tag plays the role of verifier/prover respectively as -to the best of our knowledge- in all the RFID distance-bounding protocols in the literature. We now describe the protocol (see Fig. 2) with particular attention to its differences with our first proposal (see Fig. 1).

1) $\mathcal{R} \to \mathcal{T}$: $m_1 = request, n_1$
   $\mathcal{R}$ generates a t-bit α_j random value and starts the protocol by sending $\mathcal{T}$ a request message that includes a random number n_1.
2) $\mathcal{T} \to \mathcal{R}$: $m_2 = n_2, \langle \varsigma_j, - \rangle$
   The tag generates a t-bit s_i random value and a new random number n_2. Then, a WSBC $\langle \varsigma_j, \omega_j^{\pi}(k) \rangle$ and a pseudonym υ_j to ID are computed. Finally, the tag sends the reader the nonce n_2 and the commitment ς_j and delays, to step 4, the sending of $\omega_j^{\pi}(k)$ and υ_j.
3) $\mathcal{R}$ and $\mathcal{T}$ start a low-level distance-bounding exchange. The following steps are repeated t times (a value of $t = 30$ has been previously proposed in the literature [26]), for $i = 1, ...t$
   - $\mathcal{R}$ sends bit $\alpha_j(i)$ to $\mathcal{T}$.
   - $\mathcal{T}$ sends bit $\beta_j(i) = \alpha_j(i) \oplus s_j(i)$ to $\mathcal{R}$ immediately after the reception of $\alpha_j(i)$.
4) $\mathcal{T} \to \mathcal{R}$: $m_3 = \langle -, \omega_j^{\pi}(k) \rangle, \upsilon_j, \nu_j$
   $\mathcal{T}$ generates a new n_3 random number and computes an encryption message ν_j of the t-bit random values $\{\alpha_j || \beta_j\}$ passed over the channel during the rapid bit exchange. Then, $\mathcal{T}$ sends to $\mathcal{R}$ message m_3 which is formed by nonce n_3, the result of the WSBC function $\omega_j^{\pi}(k)$, the tag's pseudonym υ_j and ν_j.
5) 5. $\mathcal{R} \to \mathcal{T}$: $m_4 = n_4^*, \tau_j^*$ (*Optional)
   When reader authentication is required, $\mathcal{R}$ sends the nonce n_4 and the encryption message τ_j^* to $\mathcal{T}$.

Two main points are critical to the security of our protocol. The use of commitments prevents that dishonest tags from sending their answer before reception of α_j. Specifically, as our protocol is based on WSBC, the tag first sends the reader the commitment ς_j and delays its opening – by sending $\omega_j^{\pi}(k)$ in step 4 – until completion of the distance-bounding exchange. To avoid mafia fraud attacks, two further precautions are taken. First, a fast bits exchange between the tag and the reader is performed. Note that t is the security parameter, fixed to a maximum of $1/2^t$ the probability of success for an adversary [6]. Secondly, once the distance-bounding exchange is complete, the tag sends the reader the encrypted message ν_j, which includes all the random values $\{\alpha_j, \beta_j\}$ exchanged by the two entities. This final step is crucial to offer resistance to mafia fraud attacks as suggested in [6] and partially in [26] where only the response bits are included in the final message.

Fig. 2. WSBC + Distance-Bounding Authentication Scheme (classical approach)

1. $\mathcal{R} \to \mathcal{T}$: $m_1 = request, n_1$
2. $\mathcal{T} \to \mathcal{R}$: $m_2 = n_2, \langle \varsigma_j, - \rangle$
3. Distance-bounding protocol
   For $i = 1, ...t$
   $\mathcal{R} \to \mathcal{T}$: $\alpha_j(i)$
   $\mathcal{T} \to \mathcal{R}$: $\beta_j(i) = \alpha_j(i) \oplus s_j(i)$
4. $\mathcal{T} \to \mathcal{R}$: $m_3 = \langle -, \omega_j^{\pi}(k) \rangle, \upsilon_j, \nu_j$
5. $\mathcal{R} \to \mathcal{T}$: $m_4 = n_4^*, \tau_j^*$ (*Optional)

where $\{n_i\}_{i=0}^4$ are different *nonces*
$\varsigma_j = enc_k(n_1 || ID || s_j || n_1 || j)$
$\omega_j^{\pi}(k) = \{k_{\pi(0)}, k_{\pi(1)}, \ldots, k_{\pi(l-1)}\}$ is a *l-bit WSBC function* and $\pi()$ is a given permutation
$\upsilon_j = h(j || n_1 || ID || s_j || n_2)$
$\nu_j = enc_k(j || n_3 || ID || \alpha_j || \beta_j || n_1)$
and $\tau_j^* = enc_k(j || n_4 || ID + 1 || n_3 || n_1)$ (Optional)

B. Cutting-edge Extension

In this protocol, we propose a role reversal for the reader and tag, which offers a completely new perspective. In this new scenario, the confidence of the tag (verifier) is a function of its distance from the reader (prover). The tag can therefore fix the hardness of the puzzle and thus the time/computation associated with its solution depending on distance measures. This is a very interesting possibility, as honest readers are often in close proximity and rogue readers are more distant.

Exploiting this advantage, honest readers would receive much simpler puzzles than dishonest readers. Consequently, the register of tags in the database may be omitted and transmission of part of the key to honest readers is also unnecessary. The scheme thus possesses all the advantages of open scenarios: key distribution is not required, key updating is straightforward, and synchronization between tags and the database is unnecessary, etc. The only remaining question is how tags can estimate their distance from readers. A direct approach is to measure the time between challenges and responses in a rapid bit exchange. As tags do not possess an on-chip clock, a capacitor's discharge time [36] can be enough for a rough estimate of the time (distance). A certain degree of inaccuracy regarding distance does not represent a major security risk as our main objective is to ascertain if the readers are very distant from tags.

We now describe the protocol (see Fig. 3) with particular attention to its differences with our first proposal (see Fig. 1).

1) $\mathcal{R} \rightarrow \mathcal{T}$: $m_1 = request, n_1, \gamma_j$
   $\mathcal{R}$ generates a t-bit s_j random value and commits this value by sending random number n_1 and message γ_j.
2) $\mathcal{T}$ and $\mathcal{R}$ start a low-level distance-bounding exchange. The following steps are repeated t times, for $i = 1, ...t$
   - $\mathcal{R}$ sends bit $c(i)$ to $\mathcal{T}$ to energize the tag. This step is required when we work with passive and semi-passive attacks and can be omitted with active attacks. c is a constant value that is linked neither to an specific $\mathcal{T}$ nor $\mathcal{R}$.
   - $\mathcal{T}$ sends bit $\alpha_j(i)$ to $\mathcal{R}$.
   - $\mathcal{R}$ sends bit $\beta_j(i) = \alpha_j(i) \oplus s_j(i)$ to $\mathcal{T}$ immediately after reception of $\alpha_j(i)$.
   - After completion of the rapid bit exchange, R opens the commitment of the hidden value s_j by sending $\{n_2, s_j\}$.
   - $\mathcal{T}$ can determine an upper bound on the $\{d_{rt}\}$ distance to $\mathcal{R}$ using the maximum of the delay times between sending out bit $\{\alpha_j(i)\}$ and receiving bit $\{\beta_j(i)\}$ back.
3) $\mathcal{T} \rightarrow \mathcal{R}$: $m_2 = n_3, \langle \varsigma_j, \omega_j^{\pi}(k) \rangle, \upsilon_j, \nu_j$
   The tag generates a new nonce n_3 and computes a WSBC $\langle \varsigma_j, \omega_j^{\pi}(k) \rangle$ which depends on the distance (d_{rt}) that separates the tag and the reader. Specifically, the l variable of $\omega_j^{\pi}(k)$ is conditioned by the distance $\{l = f(d_{rt})\}$. Finally, message m_2 is ended by an authentication message ν_j.
4) $\mathcal{R} \rightarrow \mathcal{T}$: $m_3 = n_5, \tau_j$
   $\mathcal{R}$ sends $\mathcal{T}$ the nonce n_5 and the encryption message τ_j which have a double purpose: 1) the tag can authenticate the reader; 2) the tag is able to check that the messages (challenges and responses) in the rapid bit exchange have not been altered by an adversary.

1. $\mathcal{R} \rightarrow \mathcal{T}$: $m_1 = request, n_1, \gamma_j$
2. Distance-bounding protocol
   2.0 For $i = 1, ...t$
      $\mathcal{R} \rightarrow \mathcal{T}$: $c(i)$ (Omitted with active tags)
      $\mathcal{T} \rightarrow \mathcal{R}$: $\alpha_j(i)$
      $\mathcal{R} \rightarrow \mathcal{T}$: $\beta_j(i) = \alpha_j(i) \oplus s_j(i)$
   2.1 $\mathcal{R} \rightarrow \mathcal{T}$: n_2, s_j
4. $\mathcal{T} \rightarrow \mathcal{R}$: $m_2 = n_3, \langle \varsigma_j, \omega_j^{\pi}(k) \rangle, \upsilon_j, \nu_j$
5. $\mathcal{R} \rightarrow \mathcal{T}$: $m_3 = n_5, \tau_j$
   where $\{n_i\}_{i=0}^{5}$ are different *nonces*
   $\gamma_j = h(n_1||n_2||s_j)$
   $\varsigma_j = enc_k(n_1||ID||\alpha_j||n_1||j)$
   $\omega_j^{\pi}(k) = \{k_{\pi(0)}, k_{\pi(1)}, \ldots, k_{\pi(l-1)}\}$ is a *l-bit WSBC function*, $\pi()$ is a given permutation and $l = f(d_{rt})$
   $\upsilon_j = h(j||n_1||ID||\alpha_j||n_3)$
   $\nu_j = enc_k(j||n_4||ID||n_1)$
   and $\tau_j = enc_k(j||n_5||ID + 1||\alpha_j||\beta_j||n_4||n_1)$

Fig. 3. WSBC + Distance-Bounding Authentication Scheme (cutting-edge approach)

Regarding authentication, the protocol is based on the puzzle-based RFID authentication protocol presented in Section III. For distance-checking, the protocol is inspired on the ideas suggested by Braums and Chaums and used in the classical extension scheme. We omit the security analysis due to its similarities to the analysis introduced in Sections V and VI-A respectively.

VII. Conclusions

In this paper, we explore the use of WSBCs and distance bounding-protocols as a practical and effective tool to increase the security of RFID systems. This is the first time – to the best our knowledge – that this approach has been followed to achieve this objective. Indeed, the cutting-edge extension protocol represents an additional twist on distance-bounding protocols due to the role reversal between tags and readers. That is, tags estimate their distance from readers and compute the WSBC depending on this value. The consequent advantage is that the register of tags in the database may be omitted. This provides us with an open system with many advantages such as absence of key distribution or key updating problems, which plague many other RFID security protocols.

References

[1] M. C. 0'Connor. Bridge Researchers Demo Highly Secure EPC Gen-2 RFID. RFID Journal, July 2009.

[2] G. Avoine, M. Ali Bingol, S. Kardas, C. Lauradoux, and B. Martin. A formal framework for cryptanalyzing rfid distance bounding protocols. Cryptology ePrint Archive, Report 2009/543, 2009. http://eprint.iacr.org/.

[3] G. Avoine, K. Kalach, and J.-J. Quisquater. ePassport: Securing International Contacts with Contactless Chips. In *Financial Cryptography and Data Security – FC'08*, volume 5143 of *Lecture Notes in Computer Science*, pages 141–155. Springer-Verlag, January 2008.

[4] G. Avoine and A. Tchamkerten. An efficient distance bounding RFID authentication protocol: balancing false-acceptance rate and memory requirement. In *Information Security Conference – ISC'09*, September 2009.

[5] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M. J.B. Robshaw, and Y. Seurin. Hash Functions and RFID Tags : Mind The Gap. In *Proceedings of the 10th International Workshop Cryptographic Hardware and Embedded Systems*, volume 5154 of *Lecture Notes in Computer Science*. Springer, 2008.

[6] S. Brands and D. Chaum. Distance-bounding protocols. In *Proceedings of EUROCRYPT '93: Workshop on the theory and application of cryptographic techniques on Advances in cryptology*, 1994.

[7] J.-C. Chang and H.-L. Wu. A Hybrid RFID Protocol against Tracking Attacks. Cryptology ePrint Archive, Report 2009/138, 2009.

[8] H.-Y. Chien and C.-H. Chen. Mutual Authentication Protocol for RFID Conforming to EPC Class 1 Generation 2 standards. *Computer Standards & Interfaces, Elsevier Science Publishers*, 29(2):254–259, February 2007.

[9] E. Y. Choi, S. M. Lee, and D. H. Lee. Efficient RFID Authentication Protocol for Ubiquitous Computing Environment. In *International Workshop on Security in Ubiquitous Computing Systems – SecUbiq 2005*, volume 3823 of *Lecture Notes in Computer Science*, pages 945–954. Springer-Verlag, December 2005.

[10] M. Conti, R. D. Pietro, L. V. Mancini, and A. Spognardi. RIPP-FS: an RFID Identification, Privacy Preserving Protocol with Forward Secrecy. In *International Workshop on Pervasive Computing and Communication Security – PerSec 2007*, pages 229–234. IEEE Computer Society Press, March 2007.

[11] J. Daemen and V. Rijmen. *The Design of Rijndael: AES The Advanced Encryption Standard*. Springer-Verlag, 2002.

[12] Avery Dennison. American Apparel RFID Case Study. http://www.ibmd.averydennison.com/solutions/documents/v05-11AmericanApparelTriEng005_090701FINAL.pdf, 2009.

[13] T. van Deursen and S. Radomirovic. Attacks on rfid protocols. Cryptology ePrint Archive, Report 2008/310, 2008. http://eprint.iacr.org/.

[14] T. Dimitriou. RFID-DOT: RFID Delegation and Ownership Transfer made simple. In *4th International Conference on Security and Privacy for Communication Networks – SecureComm 2008*, September 2008.

[15] C. Dwork and M. Naor. Pricing via processing or combatting junk mail. In *Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, pages 139–147. Springer-Verlag, 1992.

[16] M. Feldhofer, J. Wolkerstorfer, and V. Rijmen. AES Implementation on a Grain of Sand. *IEE Proceedings - Information Security*, 152(1):13–20, 2005.

[17] S. Fouladgar and H. Afifi. An Efficient Delegation and Transfer of Ownership Protocol for RFID Tags. In *First International EURASIP Workshop on RFID Technology*, September 2007.

[18] F. D. Garcia and P. van Rossum. Modeling Privacy for Off-line RFID Systems. In *Workshop on RFID Security – RFIDSec'09*, July 2009.

[19] G. Hancke and M. Kuhn. An RFID Distance Bounding Protocol. In *Proceedings of Conference on Security and Privacy for Emerging Areas in Communication Networks*, pages 67–73. IEEE Computer Society, 2005.

[20] G. P. Hancke and M. G. Kuhn. An rfid distance bounding protocol. In *Proceedings of the 1st Int. Conference on Security and Privacy for Emerging Areas in Communications Networks*, 2005.

[21] M. Hell, T. Johansson, A. Maximov, and W. Meier. A Stream Cipher Proposal: Grain-128. pages 1614–1618, 2006.

[22] D. Henrici and P. Müller. Hash-Based Enhancement of Location Privacy for Radio-Frequency Identification Devices Using Varying Identifiers. In *International Workshop on Pervasive Computing and Communication Security – PerSec 2004*, pages 149–153. IEEE, IEEE Computer Society, March 2004.

[23] P. Israsena. Securing Ubiquitous and Low-cost RFID Using Tiny Encryption Algorithm. In *Proceedings of International Symposium on Wireless Pervasive Computing*, 2006.

[24] A Juels. RFID security and privacy: A research survey. Manuscript, 2005.

[25] C. H. Kim and G. Avoine. RFID distance bounding protocol with mixed challenges to prevent relay attacks. Cryptology ePrint Archive, Report 2009/310, 2009.

[26] C H. Kim, G. Avoine, F. Koeune, F.-X. Standaert, and O. Pereira. The Swiss-Knife RFID Distance Bounding Protocol. In *Proceedings of International Conference on Information Security and Cryptology – ICISC*, LNCS. Springer-Verlag, 2008.

[27] L Knudsen and F. Muller. Some attacks against a double length hash proposal. In *ASIACRYPT*, volume 3788 of *Lecture Notes in Computer Science*, pages 462–473. Springer-Verlag, 2005.

[28] S. Lee, T. Asano, and K. Kim. RFID Mutual Authentication Scheme based on Synchronized Secret Information. In *Symposium on Cryptography and Information Security*, January 2006.

[29] A. Mitrokotsa, M. R. Rieback, and A. S. Tanenbaum. Classification of RFID Attacks. In *Proceedings of the 2nd International Workshop on RFID Technology*, 2008.

[30] D. Molnar, A. Soppera, and D. Wagner. A Scalable, Delegatable Pseudonym Protocol Enabling Ownership Transfer of RFID Tags. In *Selected Areas in Cryptography – SAC 2005*, volume 3897 of *Lecture Notes in Computer Science*, pages 276–290. Springer-Verlag, August 2005.

[31] R. Nithyanand. The Evolution of Cryptographic Protocols in Electronic Passports. Cryptology ePrint Archive, Report 2009/200, 2009. http://eprint.iacr.org/.

[32] M. O'Neill (McLoone). Low-Cost SHA-1 Hash Function Architecture for RFID Tags. In *Proceedings of Conference on RFID Security*, 2008.

[33] M.D. Russell. Tinyness: An overview of tea and related ciphers, February 2004. http://www-users.cs.york.ac.uk/ matthew/TEA/TEA.html.

[34] B. Song and C. J. Mitchell. RFID Authentication Protocol for Low-cost Tags. In *ACM Conference on Wireless Network Security, WiSec'08*, pages 140–147. ACM Press, April 2008.

[35] P. Syverson. Weakly secret bit commitment: Applications to lotteries and fair exchange. In *Proceedings of the 11th IEEE Computer Security Foundations Workshop*, pages 2–13, 1998.

[36] Y.-J. Tu and S. Piramuthu. RFID Distance Bounding Protocols. In *Proceedings of The First International EURASIP Workshop on RFID Technology*, 2007.

[37] S. Weis, S. Sarma, R. Rivest, and D. Engels. Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In *International Conference on Security in Pervasive Computing – SPC 2003*, volume 2802 of *Lecture Notes in Computer Science*, pages 454–469. Springer-Verlag, March 2003.

[38] J. Yang, J. Park, H. Lee, K. Ren, and K. Kim. Mutual Authentication Protocol for Low-Cost RFID. Handout of the Ecrypt Workshop on RFID and Lightweight Crypto, July 2005.