Low-cost cryptography for privacy in RFID systems

Benoît Calmels, Sébastien Canard, Marc Girault, and Hervé Sibert

France Telecom R&D, 42, rue des Coutures, BP6243, 14066 Caen Cedex 4, France
{benoit.calmels, sebastien.canard, marc.girault,
herve.sibert}@francetelecom.com

Abstract. Massively deploying RFID systems while preserving people's privacy and data integrity is a major security challenge of the coming years. Up to now, it was commonly believed that, due to the very limited computational resources of RFID tags, only ad hoc methods could be used to address this problem. Unfortunately, not only those methods generally provide a weak level of security and practicality, but they also require to revise the synopsis of communications between the tag and the reader. In this paper, we give evidence that highly secure solutions can be used in the RFID environment, without substantially impacting the current communication protocols, by adequately choosing and combining low-cost cryptographic algorithms. The main ingredients of our basic scheme are a probabilistic (symmetric or asymmetric) encryption function, e.g. AES, and a coupon-based signature function, e.g. GPS. We also propose a dedicated method allowing the tag to authenticate the reader, which is of independent interest. On the whole, this leads to a privacy-preserving protocol well suited for RFID tags, which is very flexible in the sense that each reader can read and process all and only all the data it is authorized to.

1 Introduction

RFID (Radio-Frequency IDentification) technology appeared quite a long time ago. However, it only recently began to spread into a very wide range of applications, because of both technical improvements and dramatic cost decrease. Indeed, the price of the simplest RFID tags is no more than 5 cents per tag. Thus, RFID applications such as stocks management yield cuts in expenses that represent more than the price of *tagging* every item in the stocks.

However, widely spread RFID tags usually broadcast a unique identifier over the air whenever they are powered on. This is the case of Electronic Product Code (EPC) tags with long range used in supply chains, but also that of most short range (ISO 14443/15693) tags regardless of theoretically broader abilities. For instance, the Navigo tags used by commuters in the Paris public transportation system answer readers' requests with a unique identifier. This behavior raises many concerns on privacy, and slows down massive deployment of RFID tags. On the other hand, it is commonly believed that strong cryptographic mechanisms

Appeared in J. Domingo-Ferrer, J. Posegga, D. Schreckling (Eds.): CARDIS 2006, LNCS 3928,pp. 237–251, 2006.

cannot be embedded into RFID tags because they require too high computing resources. As a consequence, literature essentially focuses on ad hoc techniques, the security level of which (as well as the practicality) is often questionable.

The purpose of this paper is to reconcile privacy concerns with RFID technology, without restricting the range of applications the tags can be used for. Whereas, in previous work, this was usually done by adding interactions between the tag and the reader, our approach is to implement in the tag some low-cost but secure cryptographic functions. This allows us to achieve a high level of privacy, without requiring any substantial change in the synopsis of communication between the tag and the reader. This also means that the tag broadcasts the same data to any reader which, depending on the authorizations it has been given, will be able to read and process part or all of the said data.

Our paper is organized as follows: we first state the problem we address, i.e. what the privacy issues are, how they are (or not) presently dealt with, and which applications should be concerned. Then, we outline the solution we propose. In section 4, we present our basic tools, namely low cost cryptographic primitives and an authentication protocol of the reader by the tag which is of independent interest. Then we describe our basic protocol, and show in the following section how it can be used in various applications. Finally, we conclude.

2 Tags and Privacy

Privacy is a particularly big concern when millions and millions of small devices are expected to be embedded into goods and to send various information over the air about them and their holders. Many popular applications of the RFID tags require these tags to be traced, but how to proceed without threatening the privacy of the people who use or wear it ? Reconciling privacy with RFID tags is therefore a very challenging task for the coming years. To achieve this goal, we first have to define precisely the different uses of these devices and the different needs they generate in terms of privacy.

2.1 Different needs for different uses

By making remote identification possible without requiring any visual contact, RFID tags are suitable in many environments where barcodes are not. For instance, RFID technology enables the quick account of the tags surrounding a tag reader, thus providing stores and warehouses with means to manage stocks and inventories more effectively than ever.

However, depending on the context, an application needs or not uniquely identify the tags it issued. While unique identification is useful for a shop that wants not only to count, but also trace its stocks, only knowing the category of the product is for other purposes often sufficient. E.g. in an airport, customs only need to detect and/or count products submitted to restrictions, without having to trace them further. Only the nature of the product, not its serial number, is relevant here. On the other hand, checking authenticity of a product can be of

Appeared in J. Domingo-Ferrer, J. Posegga, D. Schreckling (Eds.): CARDIS 2006, LNCS 3928,pp. 237–251, 2006.

great importance, e.g. in order to thwart counterfeiting. As we can see, needs related to RFID technology deeply depend on the applications they are involved in.

Detection needs. Detection consists, by using a tag reader, in first finding objects that emit signals with sufficient power to reach the reader, whereever they may be hidden, and second getting some information about them. Thus, provided the level of information given by the tags is appropriate, every accounting application can be fulfilled by this procedure (see above the customs example).

However, when the level of information which is publicly available from the tags is too high, privacy concerns arise, as provided data could allow anyone to uniquely (or almost uniquely) identify each tag. Therefore, it becomes necessary to design a general scheme for RFID tags and readers, which allows tags to disclose the nature of the items they are included in, without identifying themselves uniquely to any tag reader.

Authentication needs. Another emerging application of RFID tags is control of authenticity. Manufacturers of the luxury industry have already begun to integrate RFID tags in their products, so that counterfeiting can be detected more easily. Moreover, counterfeiting is becoming more and more usual, and luxury goods are no longer the only products concerned. As a matter of fact, every well-known brand is a potential victim of counterfeiting. Despite these threats, basic RFID tags broadcasting EPC (Electronic Product Code) 64 or 96-bit numbers can be easily duplicated. As a consequence, they do not provide at the time a satisfactory solution to authentication needs.

Identification needs. Identification needs are closely related to traceability, which consumers often consider as a threat to their privacy. This is a reason why, if not the main one, spread of RFID technology is not as fast as expected. However, traceability is required by many applications (shipments, after-sales follow-up...). Thus, in order to protect consumers' privacy, a first step is to prevent tag readers with no special privilege from tracing items.

2.2 Previous work on privacy in RFID tags

Tag deactivation. Tag deactivation consists in preventing the tag from communicating once and for all. It can be, for instance, the physical destruction of the link between the tag and its antenna, or some software deactivation function. For long, tag deactivation has been the only means to deal with privacy concerns raised by RFID tags. Deactivation is supposed to take place when the consumer has bought the product containing the tag. However, this solution is not very satisfactory, because it prevents any application that could take place later from using the tag. In case of software deactivation, the tag could be reactivated later, but this raises the problem of managing rights to (de)activate the tag.

Appeared in J. Domingo-Ferrer, J. Posegga, D. Schreckling (Eds.): CARDIS 2006, LNCS 3928,pp. 237–251, 2006.

The blocker tag. In [15], Juels, Rivest and Szydlo consider the most widespread tags singulation protocol (i.e., a scheme designed to enable the detection by a reader of all the surrounding tags in its environment, and one-to-one communications between the reader and each tag). This scheme, based on tree walking, is subject to a basic attack. Starting from this attack, the authors yield the design of a privacy-enabling tag, the *blockertag*. This approach is appealing, but it implies that RFID-based architectures are not self-sufficient in order to ensure privacy. Moreover, the attack works when applied to the cheapest, first generation tags, whose "applicative" ID (that is, their serial number, which is unique and constant) is used at the network level for the singulation process. But for instance, among Philips RFID tags from the I-CODE 1 series, several models of probabilistic singulation protocols are designed, for which the blocker tag design loses its efficiency.

Applicative privacy. With privacy in mind, many papers have proposed to include cryptographic mechanisms into RFID tags. In [14], Juels and Pappu propose a re-encryption approach for tagged banknotes designed to protect privacy. However, this approach requires external computation and is hardly adaptable to tags embedded in consumer goods. In [13], Juels describes minimalist cryptographic mechanisms designed to enhance privacy in RFID technologies at a very low cost. Works of Feldhofer *et al.* promote low-cost implementations of standard algorithms, such as AES [7], or challenge-response-type symmetric mechanisms for authentication with extensions to mutual authentication [1].

Network Layer Privacy. In [3], Avoine and Oechslin show how important it is to take the network layer exchanges into consideration. Indeed, this is where the singulation protocol takes place, and, as we have seen in Section 2.2, privacy greatly depends on the way singulation is carried out. There are advances at this level, because at this layer the only requirement is that the tag reader gains the ability to distinguish temporarily between all the tags it is surrounded by. Thus, there is no requirement on the signification of the data transmitted at this level, and using secure pseudo-random generators at this level is sufficient to ensure network layer privacy. However, one should note that singulation protocols actually implemented in the tags do not take privacy into account.

3 Outline of the general scheme

As described earlier, several needs (detection, authentication, identification) are not fulfilled with the standard mechanisms. In addition, privacy issues must also be addressed.

The main idea behind our solution is to integrate seamlessly privacy preservation mechanisms into the usual framework of a single response from the RFID tag to the reader. We use lightweight cryptographic tools in order to hide the information necessary for the usual applications of RFID systems. Then, on a

Appeared in J. Domingo-Ferrer, J. Posegga, D. Schreckling (Eds.): CARDIS 2006, LNCS 3928,pp. 237–251, 2006.

single request from the reader, the tag's response can be decoded to different levels of information by different RFID readers, depending on their knowledge.

For **detection** purpose, a way to meet the privacy requirements is to draw a separation line between public and private data in the unique identifier EPC (which is a serial number) of an RFID tag. The public data would consist mainly of digits that identify the product category, and the private data would, for instance, consist of the remaining digits of the serial number. Only the public data would be broadcast with no restriction.

One could be concerned by the public data being the same for two items of the same category, possibly causing collision problems during detection. This is not the case, because we are only dealing with the applicative level, and the singulation protocol at the network layer ensures the distinction between two items of the same category. Of course, in order to prevent tracking by every tag reader at this network level, one has to make sure that the singulation algorithms and responses from the tag are not flawed. This problem is considered, for instance, in [3].

Contrary to public data, private data must be encrypted. It will nevertheless fulfill the **identification** needs, since revealing the complete identifier only to authorized equipments. The main question is whether it is necessary and/or even possible to use an asymmetric or a symmetric encryption algorithm. While authentication can be publicly available, identification must remain very restricted, so here, depending on the applications that are aimed at, symmetric or asymmetric encryption can be chosen. However, for costs concerns, one may prefer to use symmetric encryption, especially if it is completed by some means to check the authenticity of the tag.

A simple way to meet **authentication** needs is to provide the tag with data authentication ability. This mechanism can be either symmetric (MACs) or asymmetric (digital signatures). However, it is necessary to define precisely the data that should be authenticated. In accordance with privacy protection, the signed data should not include constant information sufficient to authenticate uniquely the tag. Moreover, in order to prevent replay attacks, the signed data should include time-variant parameters. For security and practical reasons, asymmetric signatures should be preferred to MACs. Indeed, everyone (and consequently every reader) should be able to check the authenticity of a product, so means to check authenticity have to be publicly available. In case of a signature, certificates can be supplied to the reader by the tag, or can already be stored in the reader if they are available, for instance, in an online database.

4 Basic Tools

Before presenting our solution, we first introduce the cryptographic primitives it relies on. Then we propose a new and practical way for a RFID tag to authenticate the RFID reader before interacting with it, which is of independent interest.

Appeared in J. Domingo-Ferrer, J. Posegga, D. Schreckling (Eds.): CARDIS 2006, LNCS 3928,pp. 237–251, 2006.

4.1 Cryptographic Primitives

Our solution, sketched in Section 3, involves several cryptographic primitives: probabilistic encryption, signature and pseudo random number generation. Since a RFID tag is limited in terms of processing power and memory, we need to find out the cryptographic algorithms well suited for the RFID context. In this section, we introduce some possible algorithms for each one of these primitives.

Encryption. The result of encrypting a message M is denoted by $\texttt{Enc}_{K_e}(M)$, where Enc can be a symmetric or an asymmetric algorithm that is used with the encryption key K_e. The result of the decryption step is denoted by $\texttt{Dec}_{K_d}(C)$ where K_d is the decryption key [1] and C is the ciphertext. Note that we will need the encryption scheme be semantically secure, which (informally) means that the ciphertext does not leak any partial information whatsoever about the plaintext that can be computed in expected polynomial time. As a consequence, the encryption scheme shall be probabilistic.

As a symmetric encryption scheme, we can for example think of the Federal Standard AES (Advanced Encryption Standard [18]). Indeed Feldhofer et al. [7] have presented at CHES'04 a hardware implementation, the gate count of which is estimated to 3595. Furthermore, this implementation is expected to be improved in a near future. In the mean-time, proprietary algorithms can be used but their security is of course much less established.

As an asymmetric encryption scheme, we can think of NTRU [11], since the company NTRU Cryptosystems Inc. claims that it is implemented with around 1000 gates in their product called Genuid.

Signature. The signature of a message M will be denoted by $\texttt{Sign}_{K_s}(M)$ where K_s is the private signature key. The corresponding verification algorithm of the signature S on a message M is noted $\texttt{Ver}_{K_v}$ and the result $\texttt{Ver}_{K_v}(S, M)$ ("Yes" or "No"), where K_v is the public verification key. In the sequel, the algorithm considered is asymmetric (namely, we do not consider Message Authentication Code).

At CHES'04, Girault and Lefranc [10] propose a variant of GPS [8,19] that requires the prover (or the signer) to compute only one on-line addition (in $\mathbb{Z}$). From this result, we can consequently consider that a signature can be processed with less than 2000 gate equivalents using this algorithm.

Low-cost GPS necessitates that the RFID tag stores a set of coupons. A coupon is an integer that is computed beforehand and stored in the RFID tag. This computation is high-cost but can be computed by a powerful computer and sent to the RFID tag. The latter will consequently only have to send a new coupon at each new signature. As a (restrictive) consequence, the number of signatures that a RFID tag can produce is limited by the number of stored coupons, except if it is possible to refill to RFID tag with new ones. This possibility depends on the use of the RFID tag (see Section 6). Nevertheless, it is

[1] $K_d = K_e$ if the algorithm is symmetric.

Appeared in J. Domingo-Ferrer, J. Posegga, D. Schreckling (Eds.): CARDIS 2006, LNCS 3928,pp. 237–251, 2006.

possible to use hashed values for coupons, so that the hash function needs only to be implemented by the verifier [9]. Within that framework, we can consider that coupons are 32-bit long, so that several tens of coupons fit in the tag at a reasonable memory cost.

The signature scheme can also be replaced by an authentication scheme. In this case, the proof is not transferable, but this does not impact the applications of our scheme described in Section 6, as we need only convince the reader of the authenticity of the tag. Nevertheless, in the case of the authentication version of GPS, it adds one pass to the general scheme described in Section 5.2.

PRNG. The Pseudo Random Number Generator (PRNG) will be denoted by $\texttt{PRNG}(l)$ where l is the size of the output (for example in bits). This PRNG is used by the encryption scheme to ensure its probabilistic property. It can also be used to (re-)generate, during the signature phase, the random data embedded in the coupons that are used in the GPS signature scheme (see [9] for details).

We can use any PRNG that is suitable for the RFID context, that is, with a minimum of gate equivalents. It is also possible to use a block cipher algorithm (see AES in the previous section) to design a PRNG, by using standard techniques. Note that the anti-clone functions can also be turned into a PRNG. We can consequently have a PRNG in either 3600 gate equivalents or in less than 1000 gate equivalents.

4.2 Authentication of the Reader

Most authentication schemes proposed for RFID tags are dedicated to the authentication of a tag by a reader. Nevertheless, the lightweight GPS signature introduced in Section 4.1 implies the use of coupons. Thus, we have to thwart the possible denial of service attack that would consist in forcing the tag to produce many signatures, so that it uses all its coupons. This is the reason why we also propose an (optional) mechanism dedicated to the authentication of the reader. This is not a usual cryptographic authentication scheme, but rather a mechanism to distinguish between legitimate and rogue readers, based on the fact that rogue readers are unlikely to be close to the tag. We want to stress that, if it became possible to use a signature scheme that does not require coupons, then this phase could be left out.

This authentication mechanism is a kind of challenge-response scheme, and works as follows:

1. The RFID tag receives a request from a reader.
2. The RFID tag sends a challenge c to the reader (we describe below the way c is generated).
3. The reader computes $r = g(c)$ (where g is some function defined below), and sends r to the tag,
4. On reception of a value r', the RFID tag checks : $d(r', g(c)) \leq m$, where d is a distance (e.g. the Hamming distance), and m is an acceptance level.

Appeared in J. Domingo-Ferrer, J. Posegga, D. Schreckling (Eds.): CARDIS 2006, LNCS 3928,pp. 237–251, 2006.

5. The RFID tag accepts the initial request of the reader only if the previous inequality is verified.

The value m can be chosen to be equal to 0 (i.e. the RFID tag checks that $r' = g(c)$) in case data transmission already involves error-correcting codes.

The first, most interesting variant, is dedicated to tags with a short range. This is the case for ISO 14443 (proximity) tags and some ISO 15693 (vicinity) tags with a short range, this range being about a few centimeters. This is one of the most important cases, as these norms are used for contactless payment and ticketing. In this variant, c is either constant or pseudo-randomly generated by using the memory contents of the tag. It can also be picked up from a list of values, and this list can change in a deterministic way, using ideas from [13]. But the main point of this variant is that g can be chosen to be the identity function. In other words, the response is equal (or close) to the challenge ! This is made possible by the asymmetry between the emission range of the tag and the one of the reader. Indeed, an illegitimate reader is very unlikely to be in the emission range of the tag, so that it will not receive the challenge, and will consequently not be able to answer properly.

This should also work when the emission range is about one meter, because it is quite likely that an attacker aiming at making a denial of service on tags will be quite far from its potential victims. Moreover, considering that tags are moving with the persons they are carried by, it is even more unlikely that an attacker will succeed more than once to send a request that will be answered by the tag, because of this mechanism.

However, we have to take into consideration the fact that, in some places (for instance, places where luggage is gathered in an airport), this attack could be effective even if this variant is implemented. In such a case, it is recommended to use this challenge-response protocol in a classical way, i.e. by making g depend on a shared secret key, known only by the tag and authorized readers.

5 General Scheme

In this section, we introduce the general scheme of our solution on the tag's side. As outlined in Section 3, it fits seamlessly into the usual synopsis of communication between a tag and a reader. Essentially, the scheme consists in encrypting some data, and signing this ciphertext together with some public plaintext. Thus, the transmitted data do not depend on the identity of the reader.

The scalability of this construction is the essence of our scheme, since it enables privacy protection together with the applications introduced in Section 2.1. Indeed, on a single request from the reader, the tag's response can be decoded to different levels of information depending on the knowledge of the reader, thus enabling (or preventing) each application in a scalable way. We will introduce the relations between the knowledge of the reader and the possible applications/use cases in Section 6.

Appeared in J. Domingo-Ferrer, J. Posegga, D. Schreckling (Eds.): CARDIS 2006, LNCS 3928,pp. 237–251, 2006.

5.1 Data elements in the RFID Tag.

Our solution requires the RFID tag to store some values such as an identifier and, since it also performs some cryptographic operations, to store cryptographic keys. Let us introduce our notations of the data elements stored in the tag.

- Id_p: this is the public part of the identifier of the RFID tag. It can be, for example, a part of the EPC number. This public part contains general characteristics of the RFID tag, not precise enough to uniquely identify the item carrying this RFID tag. This part of the identifier has a size denoted by l_p.
- Id_s: this is the secret part of the identifier of the RFID tag. It is typically the remaining part of the EPC number (64 or 96 bits long). This data element must be considered as sensitive because it is unique and, consequently, fully identifies a particular RFID tag. If not secret, it could be used to trace the user of this RFID tag or to know where it comes from. This identifier part has a size denoted by l_s (and consequently $l_p + l_s$ will typically be the length of the entire EPC number, that is 64 or 96 bits).
- K_s: this is the signature private key that permits the RFID tag to use the algorithm `Sign` in order to sign a message. If we use the GPS algorithm, this key can be of size 160 bits. The corresponding public key K_v that will be used by the reader, is linked (in a way or another) to Id_p. The pair (K_v, K_s) can be either:
  - certified [2] by an authorized certification authority, or
  - written in a secured database (locally saved or not) that is used by the verifier.

  Consequently, all RFID tags that belong to the same group of public identifier Id_p have the same private signature key.
- K_e: this is the encryption key used by the RFID tag in the `Enc` algorithm to encrypt messages. This key can be secret (if the algorithm is symmetric) or public (if the algorithm is asymmetric). Using AES, the size of this key can be equal to 128 bits. Using the anti-clones functions, we need to store 320 bits for this key. Finally, if one wants to use NTRU, the key size depends on the security level : 1000 bits for NTRU-167, 2000 bits for NTRU-263 and 4000 bits for NTRU-503 [3]. The corresponding decryption key K_d will be used by the tag reader. Note that the pair (K_e, K_d) should be the same for a given Id_p.

5.2 The General Scheme Embedded in the RFID Tag.

The general scheme presented in this section is to be embedded into the RFID tag. This general scheme is called each time the RFID tag is requested by a

[2] This solution requires the RFID tag to store the certificate, which may conflict with memory limitations due to cost target.

[3] In terms of security, NTRU-503 is claimed to be equivalent to RSA-4096. However, the size of the key would be too high for storage in RFID tags.

Appeared in J. Domingo-Ferrer, J. Posegga, D. Schreckling (Eds.): CARDIS 2006, LNCS 3928,pp. 237–251, 2006.

RFID reader, whatever the RFID reader may be. The request from a RFID reader includes a challenge sent to the tag. This challenge can for example be a random value generated by the RFID reader, the date of the request, or simply the request sent at the network layer (see Figure 1 for details). Note that this challenge-response protocol is consistent with the communication protocol between a tag reader and an RFID tag [4].

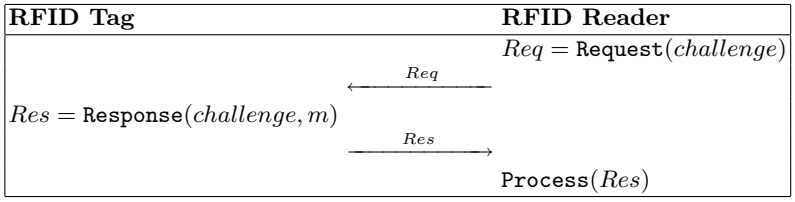

Fig. 1. General Scheme: Interactions

The generation of the response by a RFID tag is then computed as follows:

1. The RFID tag first encrypts its secret identifier Id_s plus some optional extra data m_s with the probabilist encryption scheme.

$$C = \texttt{Enc}_{K_e}(Id_s \| m_s)$$

   where the symbol $a\|b$ denotes the concatenation of a and b.
2. The RFID tag then signs a message which consists of the ciphertext C, the public identifier Id_p, the *challenge* and some optional extra public data m_p.

$$S = \texttt{Sign}_{K_s}(C \| Id_p \| challenge \| m_p).$$

3. The RFID tag finally generates the response formed as Id_p, the ciphertext C, the signature S and some optional extra data p. The data p at least includes the extra public data m_p.

$$Res = Id_p \| C \| S \| p.$$

After receiving the response of the RFID tag, the RFID reader has to process it. The way a RFID reader proceeds depends on the cryptographic keys it holds: this leads to various applications that are detailed in the next section.

5.3 Security Arguments

The lightness of this security part is due to the fact that our scheme fits into the standard RFID model, in which a tag issues a single response to the request

[4] Since an RFID tag never sends a message without a request.

Appeared in J. Domingo-Ferrer, J. Posegga, D. Schreckling (Eds.): CARDIS 2006, LNCS 3928,pp. 237–251, 2006.

of a reader. The response of the tag, except for the signature part, is the same for every tag, be it legitimate or not. For the signature part, the tag includes in the signed message the challenge sent by the reader, which is a classical technique used to turn signature into interactive proof of knowledge, thus providing authentication. Thus, it turns out that our scheme does not require a particular security model and proof, the security of the scheme being essentially that of the cryptographic primitives used.

There may be some security concerns due to the fact that the private signature key should be shared between (possibly very) many tags. A way to improve this is to use group signatures [6,2] with revocability, since there is the same problem in [5]. This would require a group signature scheme that is low-cost on the side of the signer. Thus, we think that research in this direction is definitely one of the biggest needs to concile security and privacy preservation with the massive deployment of pervasive, low-cost devices.

6 Applications

In our paper, we consider that there are various types of RFID readers, depending on the keys (secret or "public") they hold. We will consequently consider three types of RFID readers:

1. The RFID reader which holds no cryptographic key related to our scheme. We call it a "Detection reader".
2. The RFID reader which holds the verification public key that allows to verify a signature produced by a RFID tag. We call it an "Authentication reader".
3. The RFID reader which holds the decryption secret key that allows to decrypt the ciphertext generated by a RFID tag. We call it an "Identification reader".

Let us now study in details these three types of RFID readers.

6.1 Detection

An RFID reader that holds no cryptographic functionality related to the scheme described in section 4 can only read the public data sent by the reader without being able to verify them: it can consequently only detect the presence of an RFID tag. This is why we call it a "Detection reader".

Reader Side. The only data this kind of reader is able to manage are Id_p and p. In particular, the reader is not able to derive any benefit from the signature S. The reader is also unable to decrypt the ciphered data C, ensuring the privacy of the tag (and of its owner), and preventing traceability.

Appeared in J. Domingo-Ferrer, J. Posegga, D. Schreckling (Eds.): CARDIS 2006, LNCS 3928,pp. 237–251, 2006.

Use case. Though enjoying no privileges, a "Detection reader" can be useful in different situations:

- it is able to detect hidden items. For instance, it can be used in **theft detection**. In such scenarii, the articles hidden by the robber are tagged. Any reader is then able to detect tags that should not pass through a portal for instance. No authentication of the tag is required for this, and "after purchase" privacy issues are solved since unprivileged reader are not able to trace the tag.
- it can be used to **count** and **control purposes**. In a trusted environment, for instance in a warehouse where the access is controlled (and theft is not considered an issue), a Detection reader is able to count tags for inventory purpose even if it is not able to differentiate two objects[5] with the same public identifier Id_p. In an untrusted environment it can be used to detect an overtaking, for instance for customs purpose. If Id_p reveals a type (alcohol) or a brand, customs are able to easily count tags revealing a minimum of the quantity really carried.
- It can be used to **facilitate** the **search** of lost objects, since the technology does not require a visual contact and has a limited range.

6.2 Authentication

An RFID reader that holds the verification public key can check the validity of what is sent by the RFID tag. By verifying the signature, this type of RFID reader can authenticate the RFID tag. This is why we call it an "Authentication reader".

Reader side. The reader is able to manage Id_p and p, but is also able to verify the signature S, certifying the public data. It is unable to understand the ciphered data, ensuring privacy protection w.r.t. this kind of reader.

To check the signature S, the reader retrieves the public key K_v from the public identifier Id_p. The key can be stored locally in the reader or might be extracted from a database, depending on the use case. The RFID tag can also send the corresponding certificate linked to Id_p, which contains the verification public key.

Use case. The cryptographic functionalities of the reader allow it to perform operations requiring trust:

- It can be used for count and **control purposes**, with the guarantee that the final count matches the reality (e.g. for inventory purpose)

[5] As discussed earlier, to do so, the discovering protocol at the network layer must be designed with this requirement in mind. The classical "tree parsing protocol" for instance would not allow the reader to properly count tags with the same Id_p.

Appeared in J. Domingo-Ferrer, J. Posegga, D. Schreckling (Eds.): CARDIS 2006, LNCS 3928,pp. 237–251, 2006.

– The signature can be used to **prevent counterfeits**: the tags can prove their characteristics such as the brand (for luxury products), or their origin [6] (passport, identification cards), etc.
– Other services such as **traceability services** for mail and delivery can use the signature to ensure the authenticity of the product they are following up [7].

6.3 Identification

An RFID reader that holds the decryption secret key can decrypt what is sent by the RFID tag, and more particularly the secret identifier Id_s of the RFID tag. This is why we call it an "Identification reader".

Reader side. The reader is able to manage Id_p and p, but is also able to decrypt the data M in order to gain access to the secret identity Id_s and m_s.

The reader retrieves the secret key K_d thanks to the public identifier. To do so, several methods can be chosen depending on the use case. This key can be locally stored in the tag reader for instance. It can also be stored in a remote database. Then, the reader needs to authenticate itself to the remote server, and ask the required key to the server. The tag reader can also simply relay the messages to the server which will find the right key and perform the decryption.

The problem of retrieving a symmetric key to decrypt the response of a tag was adressed in [4,17]. In our scheme, each tag having the same public identifier should use the same encryption key, thus retrieving this key is straightforward in this case. However, one may want to diversify encryption keys for security concerns. Then, the tags having the same public identifier should be divided into several subgroups sharing the same encryption key, in order to increase security (and rely less on tamper-resistance of the tags) while keeping low table lookup costs to retrieve the key, using for instance a hash-based approach.

Use case. The encryption ensures the traceability w.r.t. the readers which do not hold the decryption secret. Conversely, being able to decrypt the ciphered data allows the reader to precisely identify a product. This property can then be used to:

– **identify** the owner of a **lost or stolen product**. After the purchase of a product, a database managed by the manufacturer (for instance) can link the Id_s to the identity of the customer. If a lost or stolen product is found, police services can contact the manufacturer and ask him to reveal the secret identity. To do so, the reader can relay the exchanges with the tags to the manufacturer to perform an on-line decryption.

[6] Such information can be considered as sensitive and private, so the signature could be used to validate ciphered data, and the "Authentication reader" used in conjunction with an "Identification reader".

[7] Again, to be able to identify precisely an item, such a reader should be used in conjunction with an "Identification reader".

Appeared in J. Domingo-Ferrer, J. Posegga, D. Schreckling (Eds.): CARDIS 2006, LNCS 3928,pp. 237–251, 2006.

– easily identify a product serial number for **after-sale services**. The manufacturer can get access to the whole identification code in order to know if a product is concerned by e.g. a specific problem. Then a manufacturer can call back a faulty sub-series, after rapidly and cheaply identifying them.

7 Conclusion

We have presented a protocol which allows to reconcile privacy with RFID technology. By making an optimal choice of algorithms, the cost of our solution in terms of gate equivalents is less than 5000. As stated by Juels and Weis [16], the total number of gates in supply chain RFID tags is usually between 5000 and 10000. Among these, about 2000 can currently be dedicated to security functions. Moreover, short range tags can include several times this number of gates. As a consequence, our solution suits present (or available in a very near future) hardware abilities, and we think the additional cost of such a solution is justified both by continuously decreasing costs, due to production increases and strengthening market competition, and by the functionalities it provides. In any case, we wish to emphasize that tags need to be equipped with cryptographic functions in order to enhance privacy protections in RFID systems. This should encourage the research activity in (ultra-)low cost cryptography.

References

1. M. Aigner and M. Feldhofer. Secure symmetric authentication for rfid tags. In *Telecommunication and Mobile Computing – TCMC 2005*, Graz, Austria, March 2005.
2. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In T. Okamoto, editor, *Advances in Cryptology - Asiacrypt '00*, volume 1976 of *Lecture Notes in Computer Science*, pages 255–270. Springer-Verlag, 2000.
3. G. Avoine and P. Oechslin. RFID Traceability: A Multilayer Problem. In *Financial Cryptography 2005*, Lecture Notes in Computer Science. Springer-Verlag, 2005.
4. Gildas Avoine and Philippe Oechslin. A scalable and provably secure hash based RFID protocol. In *International Workshop on Pervasive Computing and Communication Security – PerSec 2005*, pages 110–114, Kauai Island, Hawaii, USA, March 2005. IEEE, IEEE Computer Society Press.
5. S. Canard and M. Girault. Implementing group signatures schemes with smart cards. In *Smart Card Research and Advanced Applications V - Cardis 2002*. Kluwer, 2002.
6. David Chaum and Eugène van Heyst. Group signatures. In *EUROCRYPT*, pages 257–265, 1991.
7. M. Feldhofer, S. Dominikux, and J. Wolkerstorfer. Strong Authentication for RFID Systems Using the AES Algorithm. In Joye and Quisquater [12], pages 357–370.
8. M. Girault. Self-Certified Public Keys. In D. W. Davies, editor, *Advances in Cryptology - Eurocrypt '91*, volume 547 of *Lecture Notes in Computer Science*, pages 490–497. Springer-Verlag, 1991.

Appeared in J. Domingo-Ferrer, J. Posegga, D. Schreckling (Eds.): CARDIS 2006, LNCS 3928,pp. 237–251, 2006.

9. M. Girault. Low-Size Coupons for Low-Cost IC Cards. In J. Domingo-Ferrer, D. Chan, and A. Watson, editors, *Cardis 2000*, volume 180 of *IFIP Conference Proceedings*, pages 39–50. Kluwer Academic Publishers, 2000.
10. M. Girault and D. Lefranc. Public Key Authentication with one Single (on-line) Addition. In Joye and Quisquater [12], pages 413–427.
11. J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem. In *The 3rd International Symposium ANTS-III*, volume 1426 of *Lecture Notes in Computer Science*, pages 267–288, 1998.
12. M. Joye and J. J. Quisquater, editors. *CHES 2004*, volume 3156 of *Lecture Notes in Computer Science*. Springer-Verlag, 2004.
13. A. Juels. Minimalist Cryptography for Low-Cost RFID Tags, 2003.
14. A. Juels and R. Pappu. Squealing Euros: Privacy Protection in RFID-Enabled Banknotes. In R. N. Wright, editor, *Financial Cryptography 2003*, volume 2742 of *Lecture Notes in Computer Science*, pages 103–121. Springer-Verlag, 2003.
15. A. Juels, R. L. Rivest, and M. Szydlo. The blocker tag: selective blocking of RFID tags for consumer privacy. In *10th ACM conference on Computer and communications security*, pages 103–111. ACM Press, 2003.
16. Ari Juels and Stephen Weis. Authenticating pervasive devices with human protocols. In V. Shoup, editor, *Advances in Cryptology - Crypto 05*, Lecture Notes in Computer Science. Springer-Verlag, 2005.
17. David Molnar and David Wagner. Privacy and security in library RFID: Issues, practices, and architectures. In Birgit Pfitzmann and Peng Liu, editors, *Conference on Computer and Communications Security – ACM CCS*, pages 210–219, Washington, DC, USA, October 2004. ACM, ACM Press.
18. National Institute of Standards and Technology (NIST). FIPS-197: Advanced Encryption Standard. November 2001.
19. G. Poupard and J. Stern. Security Analysis of a Practical "on the fly" Authentication and Signature Generation. In K. Nyberg, editor, *Advances in Cryptology - Eurocrypt '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 422–436. Springer-Verlag, 1998.

Appeared in J. Domingo-Ferrer, J. Posegga, D. Schreckling (Eds.): CARDIS 2006, LNCS 3928,pp. 237–251, 2006.