Private Interrogation of Devices via Identification Codes*

Julien Bringer[1], Hervé Chabanne[1,2], Gérard Cohen[2], and Bruno Kindarji[1,2]

[1] Sagem Sécurité
[2] Télécom ParisTech

Abstract. Consider a device that wants to communicate with another device (for instance a contactless one). We focus on how to withstand privacy threats in such a situation, and we here describe how to query the device and then identify it, with a new identification protocol solution. The interrogation step uses the concept of identification codes introduced by Ahlswede and Dueck. We show that this probabilistic coding scheme indeed protects the device against an eavesdropper who wants to track it. In particular, when using a special class of identification codes due to Moulin and Koetter that are based on Reed-Solomon codes, we directly depend on the hardness of a cryptographic assumption known as the Polynomial Reconstruction problem. We analyse the security and privacy properties of our proposal in the privacy model for contactless devices introduced by Vaudenay at ASIACRYPT 2007. We finally explain how to apply our scheme with very low-cost devices.

Keywords: Identification, Privacy, Polynomial Reconstruction Problem.

1 Introduction

In the field of contactless communication, a verifier (often called a sensor or reader of devices) is used to identify the objects by verifying the validity of the attached contactless devices. This is the case for Radio Frequency IDentification (RFID) systems, where devices are attached to physical objects. The verification is realized through an authentication protocol between a device and the verifier. Once authenticated, the verifier manages the object and allows the owner of the object to access some service. Applications examples include in stock management application for real-time item identification and inventory tracking, e-passport applications, etc. Devices can also be part of a sensor network that gives information on the related infrastructure around a geographical zone.

In this context, a verifier has often to manage many devices at the same time in the same area. Main issues are then efficiency, security and cost, and, of course, the problem very specific to the field of contactless communication: privacy. Many schemes to handle the latter problem have been proposed so far

* This work was partially funded by the ANR T2TIT project.

R. Roy and N. Sendrier (Eds.): INDOCRYPT 2009, LNCS 5922, pp. 272–289, 2009.

(e.g. [17,20,30,28,22,19,11,7,27,24,26,21,3]; see [8] for a more exhaustive list), but finding an efficient solution enabling privacy of devices is still an active field of research.

Contactless devices are generally assumed to respond automatically to any verifier scan. We follow, in this work, an idea [23] that suggests that the verifier directly addresses the device with which it wants to communicate. To this aim, the verifier broadcasts the device identifier and then the corresponding device responds accordingly. However, the emission of the device identifier enables an eavesdropper to track it. We here look for a solution which does not require many computations and many communications efforts, while preventing an eavesdropper to be able to track a particular device. Changing the paradigm from the situation where a device initiates the protocol to a situation where the device identifies first the interrogation request enables to envisage new solutions.

We show that Identification Codes [1] perfectly fit to our needs. They were introduced by Ahlswede and Dueck to enable the identification of an element out of $\{1,\ldots,n\}$ by only conveying $\log\log n$ bits. While transmission codes enable to correct messages under some noise tolerance property – i.e. to answer the question *What is the received message?*, an identification code detects if a particular message m has been transmitted – i.e. answers the question *Is it the message m?*. We show that such a probabilistic coding scheme increases a lot the job of the eavesdropper as the same identifying bit string is not used twice except with a small probability. In particular, for the class of identification codes of [18], a reduction to the cryptographic assumption of [15] is possible.

Our introduction of Identification Codes for authenticating devices can be viewed in the more general context of challenge-response protocols. Each device has an identifier m and the prover broadcasts a challenge associated to m. Here our scheme does not rely neither on hash functions nor on a random generator on the device side. Moreover, our work shows that our solution is very efficient in terms of channel usage.

We first describe a general scheme based on these identification codes and show that our scheme satisfies good security and privacy properties by analysing it in the privacy model defined in [28]. We then explain how the scheme is suited to very low-cost devices.

Note that the problematic of this article is not limited to interrogation of low-cost devices; in fact, we focus on interrogation protocols and any independent component that communicates over a noisy broadcasting channel is a potential target (as e.g. in [4]).

2 Identification Codes

We wish to communicate mainly with contactless devices, which means that all the communications are to pass through radio waves. As a direct consequence, a message that is sent over the channel is publicly available to any eavesdropper. In a realistic model where a verifier sequentially communicates with wireless devices, it is the verifier that will initiate the communication. To that purpose,

the verifier first beckons the device with which it wants to communicate. The most efficient way for doing so is to use an identification code.

2.1 General Definition

Let $\mathcal{X}, \mathcal{Y}$ be two alphabets, and W^η a channel from $\mathcal{X}^\eta$ to $\mathcal{Y}^\eta$. W^η is defined as the probability to receive a message $y^\eta \in \mathcal{Y}^\eta$ given a transmitted message $x^\eta \in \mathcal{X}^\eta$. By extension, for a given subset $E \subset \mathcal{Y}^\eta$, $W^\eta(E|x^\eta)$ is the probability to receive a message belonging to E when x^η has been transmitted.

Definition 1 (Identification Code, [1]). *A $(\eta, N, \lambda_1, \lambda_2)$-identification code from $\mathcal{X}$ to $\mathcal{Y}$ is given by a family $\{(Q(\cdot|i), \mathcal{D}_i)\}_i$ with $i \in \{1, \ldots, N\}$ where:*

- *$Q(\cdot|i)$ is a probability distribution over $\mathcal{X}^\eta$, that encodes i,*
- *$D_i \subset \mathcal{Y}^\eta$ is the* **decoding set***,*
- *λ_1 and λ_2 are the first-kind and second-kind error rates, with*

$$\lambda_1 \geq \sum_{x^\eta \in \mathcal{X}^\eta} Q(x^\eta|i) W^\eta(\overline{D_i}|x^\eta)$$

and

$$\lambda_2 \geq \sum_{x^\eta \in \mathcal{X}^\eta} Q(x^\eta|j) W^\eta(D_i|x^\eta)$$

(where $W^\eta(D_i|x^\eta)$ is the probability to be in the decoding set D_i given a transmitted message x^η and $W^\eta(\overline{D_i}|x^\eta)$ the probability to be outside the decoding set)

for all $i, j \in \{1, \ldots, N\}$ such that $i \neq j$.

Given $Q(\cdot|i)$, the **encoding set** of i is defined as the set of messages x^η for which $Q(x^\eta|i) > 0$.

Informally, an identification code is given by a set of (probabilistic) coding functions, along with (deterministic) decoding sets. The error rate λ_1 gives the probability of a false-negative, and λ_2, of a false-positive identification. We stress that the use of an identification code in our case is more interesting than using a transmission code for the following reasons:

- The efficiency in terms of information rate: the rate of such a code is defined as $R = \frac{1}{\eta} \log\log N$ and can (see [1, Theorem 1]) be made arbitrary close to the (Shannon) capacity of the channel. This means that it is possible to identify $N = 2^{2^{R\eta}}$ devices with a message of length η, with constant error rates (λ_1, λ_2). A regular **transmission** code permits only to identify $2^{R\eta}$ devices.
- The transmission of an element of D_i to identify the device i permits its identification without completely giving away the identity i. Indeed, an eavesdropper only gets the message sent $x^\eta \in Y^\eta$, not the associated index i. The use of an identification code is thus a good way to enhance privacy in the beckoning of wireless devices. This notion is formalized in Section 3.

The proof of the result stated in [1, Theorem 1] is based on a generic construction, exhibited hereafter. Let $A_1, \ldots, A_N \subset X^\eta$ be N subsets such that each A_i has cardinal n and each intersection $A_i \cap A_j$ for $i \neq j$ contains at most λn elements. The encoding distribution $Q(\cdot|i)$ is defined as the uniform distribution over A_i; in the noiseless case (the channel W^η is the identity function) the decoding sets are also the A_i's. Note that in that case the false-negative rate λ_1 is equal to 0 and the false-positive rate λ_2 is λ.

This theoretical construction gives way to multiple practical identification codes based on constant-weight codes, such as [16, 29, 6]. We focus on [18] which provides a simple though efficient identification code well suited to our application.

2.2 Moulin and Koetter Identification Codes Family

We here recall a simple construction of identification codes proposed by Moulin and Koetter [18].

The identification code detailed in [18] is based on an Error-Correcting Code C of length n, size $N = |C|$ and minimum distance d over some alphabet. For a word $c_i = (c_i^{(1)}, \ldots c_i^{(n)}) \in C$, the corresponding set A_i is the collection of all $(u, c_i^{(u)})$, for $u \in \{1, \ldots, n\}$. Note that we indeed have sets A_i of constant size n; moreover, the intersection of two different sets $A_i \cap A_j$ contains at most $n - d$ elements, which induces $\lambda_2 = \frac{n-d}{n} = 1 - \frac{d}{n}$.

A Reed-Solomon code over a finite field $A = \mathbb{F}_q$, of length $n < q - 1$, and dimension k, is the set of the evaluations of all polynomials $P \in \mathbb{F}_q[X]$ of degree less than $k-1$, over a subset $F \subset \mathbb{F}_q$ of size n ($F = \{\alpha_1, \ldots, \alpha_n\}$). In other words, for each k-tuple $(x_0, \ldots, x_{k-1}) \in \mathbb{F}_q^k$, the corresponding Reed-Solomon word is the n-tuple $(y_1, \ldots, y_n)$ where $y_i = \sum_{j=0}^{k-1} x_j \alpha_i^j$. In the sequel, we identify a source word $(x_0, \ldots, x_{k-1}) \in \mathbb{F}_q^k$ with the corresponding polynomial $P = \sum_{j=0}^{k-1} x_j X^j \in \mathbb{F}_q[X]$.

Definition 2 (Moulin-Koetter RS-Identification Codes). *Let $\mathbb{F}_q$ be a finite field of size q, $k \leq n \leq q-1$ and an evaluation domain $F = \{\alpha_1, \ldots, \alpha_n\} \in \mathbb{F}_q$. Set $A_P = \{(j, P(\alpha_j)) | \, j \in \{1, \ldots, n\}\}$ for P any polynomial on $\mathbb{F}_q$ of degree at most $k-1$.*

The Moulin-Koetter RS-Identification Codes is defined by the family of encoding and decoding sets $\{(A_P, A_P)\}_{P \in \mathbb{F}_q[X], \deg P < k}$.

This leads to a $(\log_2 n + \log_2 q, q^k, 0, \frac{k-1}{n})$-identification code from $\{0,1\}$ to $\{0,1\}$.

Using a Reed-Solomon code of dimension k, this gives $\lambda_2 = \frac{k-1}{n}$ since $d = n-k+1$ (Reed-Solomon codes are Maximum Distance Separable).

2.3 Application to Our Setting

Back to our original problem of devices interrogation, here comes a brief description of a set-up that enables the use of identification codes to initiate a protocol between a verifier and a device. A more formal description is given in Section 4.

A set of $M < q^k$ devices is constructed, and each of them is associated with a different random polynomial $p_l \in \mathbb{F}_q[X]$ of degree less than $k-1$. The memory of these devices is then filled with a set of $p_l(\alpha_j)$, for $\alpha_j \in F$, with F a public subset of $\mathbb{F}_q$, *i.e.* the devices contain the evaluation of p_l over a subset of $\mathbb{F}_q$. The verifier is given the polynomial p_l.

When the verifier wants to initiate communication with the device number l associated with the identifier p_l, it selects a random $\alpha_j \in F$ and sends $(j, p_l(\alpha_j))$ over the wireless channel. A device that receives this message checks whether the value stored in its memory at the corresponding address is equal to $p_l(\alpha_j)$, *i.e.* computes an equality test of two bit strings. If the test is successful, it replies and goes through the authentication protocol described in Section 4. Otherwise, it remains silent.

Consequently, only a legitimate verifier can interrogate a specific device. Next sections emphasize the security properties reached thanks to this principle.

3 Vaudenay's Model for Privacy

We briefly recall in this section the model for privacy, correctness and soundness described in [28]. Our main concern is interrogation of devices, but it can be easily seen as an authentication protocol, so we use almost the same model.

Following [28], we consider that provers are equipped with ContactLess Device (CLD) to identify themselves. CLDs are transponders identified by a unique Serial Number (SN). During the identification phase, a random virtual serial number (vSN) is used to address them.

An identification protocol is defined as algorithms: First to setup the system made of a verifier and several CLDs, secondly to run a protocol between CLDs and verifiers. Note that we need an authority who publishes a mathematical structure.

Setup Algorithms

- $\textsc{SetupAuthority}(1^k) \mapsto (KA_s, KA_p)$ generates the system parameters defined by an authority (KA_s stands for the private parameters and KA_p for the parameters publicly available).
- $\textsc{SetupVerifier}_{KA_p}$ initializes a verifier. It may generate a private/public set of parameters (KV_s, KV_p), associated to the verifier.
- $\textsc{SetupCLD}^b_{KA_p,KV_p}(\text{SN})$ generates the parameters of the CLD identified by SN. This algorithm outputs a couple (s, I) where s denotes the secret (if any) parameters of the CLD, I its identity within the system. It enables to initialize the internal state of the CLD, which may be updated afterwards during an execution of the protocol. If $b = 1$, it also stores the pair (I,SN) in a database which may be made available to the verifier. If $b = 0$ it is a illegitimate device.

Communication Protocol $\mathcal{P}$. Along with these setup algorithms, the identification protocol between a CLD and a verifier consists of messages sent by the two parties. Protocol instances are hereafter denoted by π.

Oracles. To formalize possible actions of an adversary, different oracles are defined to represent ways for him to interact with verifiers or CLDs, or to eavesdrop communications. The use of different oracles leads to different privacy levels.

Given a public set of parameters KV_p, the adversary has access to:

- CreateCLD^b(SN): creates a CLD with serial number SN initialized via SetupCLD^b. At this point, it is a free CLD, i.e. not yet in the system.
- DrawCLD($distr$)$\mapsto$((vSN$_1$,b_1),...,(vSN$_n$,b_n)): this oracle moves a random subset of n CLDs according to a given distribution from the set of free CLDs into the set of drawn CLDs in the system. Virtual serial numbers vSN$_i$ can be used to refer to these CLDs. If b_i is one, this indicates whether a CLD is legitimate. This oracle creates and keeps a table of correspondences $\mathcal{T}$ where $\mathcal{T}$(vSN)=SN. Adversary has no knowledge of this table $\mathcal{T}$.
- Free(vSN): moves the drawn CLD vSN to the set of free CLDs, i.e. vSN cannot be used any more to query the CLD.
- Launch $\mapsto \pi$: makes the verifier launch a new protocol instance π.
- SendVerifier$(m, \pi) \mapsto m'$: sends the message m for the protocol instance π to the verifier who may respond m'.
- SendCLD$(m', \pi) \mapsto m$: sends the message m' to the CLD who may respond m.
- Result$(\pi) \mapsto x$: when π is a complete instance of $\mathcal{P}$, it returns 1 if the verifier succeeds in identifying a CLD from π and 0 otherwise.
- Corrupt(vSN)$\mapsto S$: returns the internal state S of the CLD vSN.

Types of Adversary

- **Strong** adversary is allowed to use all of the above oracles.
- **Destructive** adversary cannot use a corrupted CLD another time.
- **Forward** adversary cannot use any oracle after one Corrupt query, i.e. destroys the system when he corrupts one CLD.
- **Weak** adversary is not allowed to use the Corrupt oracle.
- **Narrow** adversary is not allowed to use the Result oracle.

This defines 8 kinds of adversaries because a narrow adversary may also have restrictions on the use of the Corrupt oracle. For instance, an adversary can be narrow and forward, he is then denoted by narrow-forward.

Remark 1. The notion of **destructive** adversary is an intermediate notion between **strong** and **forward** adversaries. As explained in [19], **destructive** notion is different from **forward** notion only when the system enables the introduction of some correlated secrets between CLDs. This is not our case in the sequel, so we will no further distinguish these two notions.

Three security notions are defined in this model: correctness, resistance against impersonation and privacy.

Definition 3. *A scheme is* **correct** *if the identification of a legitimate CLD fails only with negligible probability.*

Resistance against Impersonation Attacks. The definition of resistance against impersonation attacks (Definition 4) deals with active adversaries. Active adversaries may impersonate verifiers and CLDs, and eavesdrop and modify communications. This property of resistance against impersonation attacks has also repercussions regarding privacy properties (cf. Lemma 1).

Definition 4. *A scheme is* **resistant against Impersonation Attacks** *if any polynomially bounded* **strong** *adversary is not identified by a verifier except with a negligible probability. Adversaries are authorized to use different devices at the same time while they communicate with the verifier. Nevertheless, the resulting protocol transcript must neither be equal to an outputted one between a legitimate CLD and the verifier nor lead to the identification of a corrupted CLD.*

Remark 2. Obviously this means that a scheme is not resistant against impersonation attacks if an adversary is able to modify on the fly outputs from a prover without affecting the identification result.

In addition to this definition, in order to mitigate replay attacks, a legitimate verifier should not output twice the same values in two complete protocol instances, except with a negligible probability.

Note that following Remark 1, the CORRUPT oracle will be useless for impersonation attacks against our scheme (as secret are not correlated between devices).

Similarly, and as in [22], we introduce the **resistance against impersonation of verifier** where an adversary should not be able to be identified as a legitimate verifier by a non-corrupted CLD except by replaying an eavesdropped transcript. This is related to the notion of verifier authentication. Note that we introduce a slight restriction in Section 5.3 as our scheme aims only at ensuring validity of the verifier against a pre-fixed CLD.

Privacy. Privacy is defined as an advantage of an adversary over the system. To formalize this, [28] proposes to challenge the adversary once with the legitimate oracles and a second time with simulated oracles. In this setting, the adversary is free to define a game and an algorithm $\mathcal{A}$ to solve his game. If the two challenges results are distinguishable, i.e. if the system cannot be simulated, then there is a privacy leakage. A game with three phases is imposed. In the first phase, $\mathcal{A}$ has access to the whole system using oracles. In a second phase, the hidden table $\mathcal{T}$ of correspondences is transmitted to $\mathcal{A}$ (note that this table is never learned by the simulator). In a third phase, $\mathcal{A}$, who is no longer allowed to use the oracles, outputs its result. A scheme is defined as **private** if for any game, all adversaries are trivial (the formal definition is given in Appendix A, Definition 7).

The following lemma established by Vaudenay in [28] emphasizes the link between impersonation resistance and privacy:

Lemma 1. *A scheme secure against impersonation attacks and narrow-weak (resp. narrow-forward) private is weak (resp. forward) private.*

The proof relies on the fact that an adversary is not able to simulate any CLD if the scheme is sound. This implies that the RESULT oracle is easily simulated.

Remark 3. Our model aims at dealing with identification of multiple devices. It is therefore reasonable to amend the privacy model by stating that the SEND-CLD(m', π) oracle cannot communicate with a single CLD, but broadcasts the message m' to all the CLDs in the vicinity. Moreover, as it was shown in D'Arco *et al.* [5], no privacy is possible if the adversary can deactivate a CLD, which is possible if we allow the adversary to manipulate the CLDs one by one.

[28] proves also that narrow-strong privacy implies the use of public key cryptography and that strong privacy is impossible in this model. In the sequel we stick to symmetric cryptography, and that is why we do not analyse the narrow-strong privacy any further. Furthermore, as explained in the previous remark, we exclude from our model of threats the situation where the adversary communicates with one isolated device.

4 Our Protocol for Interrogation

Our aim is for a CLD to recognize itself into a verifier request, but authentication of the CLD toward the verifier is handled as well. That is how we set-up the system:

- SETUPAUTHORITY(1^ℓ) generates a set of parameters KA_p defining two integers η, N, two alphabets $\mathcal{X}$, $\mathcal{Y}$, and two error rates λ_1, λ_2. No private parameter is defined.
- SETUPVERIFIER$_{KA_p}$ constructs an $(\eta, N, \lambda_1, \lambda_2)$-identification code from $\mathcal{X}$ to $\mathcal{Y}$ following Definition 1, $\mathcal{IC} = \{(Q(\cdot|i), \mathcal{D}_i)\}_{i\in\{1,\ldots,N\}}$, and sets $KV_p = \mathcal{IC}$. $\mathcal{IC}$ is based on the Moulin-Koetter construction [18] (cf. Definition 2).
- SETUPCLD$_{KV_p}$(SN) first returns randomly chosen $(i, j) \in \{1, \ldots, N\}$, $i \neq j$ as the parameters of the CLD identified by SN. It then initializes the CLD with the storage of a description of the decoding set D_i of the identifier i and the description of $Q(\cdot|j)$, the encoding probability mass function for index j. It also stores (i, j, SN) in the verifier database.

A verifier and a set of devices are set-up as above and the following steps are then processed to interrogate and authenticate a specific CLD.

- The verifier, who wants to interrogate the CLD of identifier SN, recovers its identifier i in the database and encodes it via $Q(\cdot|i)$ into a message $x \in \mathcal{X}^\eta$. The verifier broadcasts the message (ACK, x), where ACK is an acknowledgement number which will help the verifier to sort the received answers when it emits simultaneously several such messages.
- Any listening CLD that receives the message (ACK, y) uses its own decoding set $D_{i_{CLD}}$ to determine whether y encodes i_{CLD}.
- If a CLD identifies y as an encoding of its identifier i_{CLD}, then it sends the message (ACK, x') to the verifier, where ACK is the incoming acknowledgement number and x' is an encoding of j_{CLD} obtained via $Q(\cdot|j_{CLD})$.

– Upon receiving this message, the verifier then checks whether the received message y' is a member of the decoding set D_j of the aimed CLD. If so, then the CLD is declared as authenticated.

Note that here x' has to be chosen in relation with the value of y so that impersonation of a CLD is not easy.

Remark 4. As a practical assumption, our interrogation protocol works as a broadcast channel and we assume that a legitimate verifier is interrogating several CLDs during the same period. Although it might look restrictive, recall that our goal is to address applications where a verifier has to manage efficiently a cloud of CLDs. More formally, we assume that a cloud of M CLDs is present in the broadcast area of the verifier and that the verifier interrogates them uniformly in a random order. In particular, an adversary is not able to *a priori* distinguish the devices without trying to exploit the content of messages exchanged.

4.1 Specifications Using Reed-Solomon Based Identification Codes

We now consider only the Moulin-Koetter setting, in particular for the security analysis in the next sections. The description is given below (see also Fig. 1).

In this setting, a set of CLDs is constructed where each of them – say CLD_l – is associated with two different random polynomial identifiers p_l, $p'_l \in \mathbb{F}_q[X]$ of degree at most $k-1$. Here p_l and p'_l are good descriptions of the associated encoding functions and the decoding sets; they are both stored on the CLD side and on the verifier database.

When the verifier wants to initiate communication with CLD_l (with identifiers p_l, p'_l), it selects a random $\alpha_j \in F \subset \mathbb{F}_q[X]$ and broadcasts $(ACK, j, p_l(\alpha_j))$ over the wireless channel. A CLD with identifiers p, p' that receives this message checks whether the polynomial p stored in its memory evaluated in α_j is equal to $p_l(\alpha_j)$. If the test is successful, it responds with the value $(ACK, p'(\alpha_j))$. Otherwise, it remains silent. The verifier authenticates the CLD if the received value $p'(\alpha_j)$ is equal to $p'_l(\alpha_j)$.

Remark 5. For privacy purposes, we do not want replay attacks to be possible at all. In order to avoid them, we add to each devices a flag bit that tells if the α_j

CLD identifiers p, p'	parameters $\mathbb{F}_q, (\alpha_1, \ldots, \alpha_n)$	Verifier (l, p_l, p'_l)
	$\xleftarrow{(ACK,\, j,\, a=p_l(\alpha_j))}$	Pick j
If $p(\alpha_j) = a$	$\xrightarrow{(ACK,\, b=p'(\alpha_j))}$	Check whether $p'_l(\alpha_j) = b$

Fig. 1. CLD identification via Moulin-Koetter identification codes

was already used or not; this bit is flipped on at the reception of $(j, p(\alpha_j))$; after that, a device no longer accepts such a message. This can be seen as coupons enabling a limited number of interrogations by a legitimate verifier.

When communicating with an isolated device, it may enable an adversary to track the device via a replay attack by listening whether the device responses. In our situation, this does not lead to a privacy threat as the adversary is only able to interrogate a cloud of devices which is continuously evolving.

5 Security Analysis

Remark first that the scheme is correct: In the Moulin-Koetter construction (cf. Section 2.2) the false-negative error rate (λ_1) is zero, thus the correct CLD will always answer and be authenticated.

5.1 Assumptions

Part of our results are directly linked to solving the problem of polynomial reconstruction (PR) [15, 14, 13, 12]:

Definition 5 ([15]). *Given n, k, t such that $n \geq t \geq 1$, $n \geq k$ and $z, y \in \mathbb{F}_q^n$, with $z_i \neq z_j$ for $i \neq j$, output all (p, I) where $p \in \mathbb{F}_q[X]$, $\deg(P) < k$, $I \subset \{1, \ldots, n\}$, $|I| \geq t$, and $\forall i \in I, p(z_i) = y_i$. Such an instance of this problem is noted $PR^z_{n,k,t}$.*

The Guruswami-Sudan algorithm [10] for the **list decoding** of Reed-Solomon codes gives a way to solve the polynomial reconstruction problem when $t \geq \sqrt{kn}$. However, no efficient solution to this problem exists when $t < \sqrt{kn}$ and it is reputed hard. If $t < k$, PR is unconditionally secure (in the information-theoretical meaning).

Based on the assumed intractability of PR, [15] derives the Decisional PR (DPR) problem which consists, given an instance y of $PR^z_{n,k,t}$ for which there exists a solution (p, I), in determining whether a given $i \in \{1, \ldots, n\}$ is in I. Thanks to the DPR assumption (hardness of the DPR problem), it is shown [15] that PR instances are pseudo-random and that they do not leak any partial information on the polynomial values.

Remark 6. In the sequel we assume that the PR and DPR problems remain hard (with respect to the security parameter ℓ) even in our setting – where the noise is generated by the other queries and responses. M will be chosen so that the DPR assumption holds when the noise is assumed to be random. To justify this choice, we can refer to [9] which explains the link between Reed-Solomon list decoding and the previous works on polynomial reconstruction in the mixture model. An algorithm to reconstruct polynomials from mixed values is designed in [2]. When considering mixed evaluations of M polynomials of degree at most $k-1$, it enables to reconstruct one of these polynomials when at least $M(k-1)$ related values are available in the mixture. In the sequel, we set M greater than $\sqrt{\frac{n}{k}}$ so that $M(k-1)$ is approximately greater than $\sqrt{nk}$, i.e. that we obtain

the same bound as for the solvability of PR instances. This algorithm is the basis – although a bit simpler – of the list decoding algorithm [10] and this fact suggests that when we get less than $M(k-1)$ values for each polynomial with M large, the problem of reconstructing one polynomial remains hard even without a perfectly random noise.

5.2 Effect of Passive Eavesdropping

When listening on the channel to the queries made by a legitimate verifier and the replies produced by legitimate CLDs, an eavesdropper sees messages of this kind: $(ACK_i, j_i, p_{l_{j_i}}(\alpha_{j_i}))$, $(ACK_i, p'_{l'_{j_i}}(\alpha_{j_i}))$ (for l'_j such that $p_{l'_j}(\alpha_j) = p_{l_j}(\alpha_j)$), for some number of i's (say $i \in \{1, \ldots, T\}$). Note that we may also have collisions on the α_j used (i.e. $j_i = j_{i'}$ may occur for some $i \neq i'$). This means that the adversary obtains a set S of several PR instances of length less or equal to n (the length of the overall code, see Section 2.2). Targeting a specific CLD, of identifier p and p', then there are at least two corresponding PR instances, $PR^{z_1}_{n_1,k,t_1}$ and $PR^{z_2}_{n_2,k,t_2}$ where p is one solution of the first one and p' a solution of the latter, among the set S of all those PR instances. One difficulty for the adversary is to sort the different messages and to deal with the collisions to extract such instances. If we assume that there is no collision (then necessarily $T \leq n$) and that the verifier queries uniformly the M CLDs (cf. Remark 4), then it implies that the adversary can recover these instances, but with $t_i \approx \frac{n_i}{M}$. So if M is greater than $\sqrt{\frac{n}{k}}$ then the PR instances are hard.

Moreover, when the number of received messages is large, the t_i's above may be greater than $\sqrt{kn}$ but the adversary has to deal with the collisions and to try all the different instances until the recovery of a solvable instance. Another strategy is to see the problem as one longer PR instance. This is related to the **list recovery problem** which is analysed in [25]. This is hard as well given some restriction on the number of eavesdropped messages. In the sequel, we assume that the list recovery problem in the mixture model is hard when $t < \sqrt{nk \times l}$ with l the maximum number of collisions per z_i.

Proposition 1. *Assume that the number M of devices simultaneously queried by the verifier is such that $\sqrt{q} \geq M \geq e\sqrt{\frac{n}{k}}$ (with $e = exp(1)$). Then a passive adversary, who eavesdrops at most T requests with $T < M^2 k$, cannot reconstruct the polynomial identifiers, except with a negligible probability.*

Proof. Assume that the adversary has eavesdropped T different requests with $T/M \geq \sqrt{kn}$, then there may exist solvable PR instances. Now he has to find these solvable instances among all possible instances. Following Remark 4 on uniformity of the queries made by a verifier, we assume that the number of different requests to each device is exactly $t = T/M$. (Due to the false-positive error rate of the underlying identification code, one request will address several additional devices and imply as many replies. In fact, as the polynomials are chosen independently and uniformly, the number of devices addressed by one query is strictly greater than 1 only if there is a collision during the evaluation

of several polynomials. The assumption $M \leq \sqrt{q}$ enables us to neglect this point, but the result is easily generalizable to the case $M > \sqrt{q}$.)

Let $M \geq \gamma\sqrt{\frac{n}{k}}$ where γ will be determined later. Note that if $T/M < k$ then it is unconditionally secure and if $T < \gamma n$ then $T/M < \sqrt{nk}$ so that the PR instances are hard. Assume that $T \geq \gamma n$, thus the number of collisions per α_j is expected to be about T/n (note that $T/M \leq n$ as each device is linked to at most n different requests). To make computation more tractable, we assume below that the number of collisions per α_j is exactly T/n.

The adversary has to reconstruct one polynomial corresponding to some part of the eavesdropped values.

The first strategy for the adversary is to find a solvable PR instance in the classical meaning, i.e. without any collision. The number of possible PR instances is then expected to be $B = \left(\frac{T}{n}\right)^n$ whereas the number of solvable instances is $A = M \times \binom{T/M}{\lceil\sqrt{kn}\rceil}\left(\frac{T}{n}\right)^{n-\lceil\sqrt{kn}\rceil}$. If the ratio $\rho = \frac{A}{B}$ of the number of solvable instances over the number of all possible instances is negligible then the adversary would not find a solvable instance in polynomial time. In fact ρ is equal to

$$M\binom{T/M}{\lceil\sqrt{kn}\rceil}\left(\frac{T}{n}\right)^{-\lceil\sqrt{nk}\rceil}.$$

To approximate ρ, note $R = \frac{k}{n}$ the rate of the Reed-Solomon code as eavesdropped by the adversary. We also introduce $\theta > 1$ such as $\frac{T}{M} = \theta\sqrt{kn}$. The notations give $M = \frac{\gamma}{\sqrt{R}}$ and $\frac{T}{n} = \theta\gamma$. A good approximation of $\binom{T/M}{\lceil\sqrt{kn}\rceil}$ is, for $\theta > 2$, $2^{\frac{T}{M}h_2\left(\frac{M\sqrt{kn}}{T}\right)} = 2^{n\sqrt{R}\theta h_2(\frac{1}{\theta})}$ where h_2 is the binary entropy function. This shows that ρ can be fairly approximated by

$$\rho \approx \frac{\gamma}{\sqrt{R}}2^{n\sqrt{R}\left(\theta h_2(\frac{1}{\theta})-\log_2(\theta\gamma)\right)}.$$

Taking a closer look at the exponent, we see that $\theta h_2(\frac{1}{\theta}) - \log_2(\theta\gamma) = (\theta - 1)\log_2(\frac{\theta}{\theta-1}) - \log_2(\gamma)$ is negative only if $\gamma > \left(1 + \frac{1}{\theta-1}\right)^{\theta-1}$. As $\forall x \in \mathbb{R}^\star, \log(1 + \frac{1}{x}) < \frac{1}{x}$, we deduce that if $\gamma \geq e$, then $\theta h_2(\frac{1}{\theta}) - \log_2(\theta\gamma) < 0$. Thus, $\rho \leq M2^{-n\sqrt{R}\log_2(\frac{\gamma}{e})}$ is negligible.

This gives a negligible probability for the adversary to find a solvable instance. This conclusion can be generalized to non-constant number of collisions as soon as the j picked by the verifier is chosen uniformly and independently among the different requests.

The general strategy is to apply the list recovery technique [25] derived from the list decoding algorithm [10]. This becomes tractable as soon as T/M is greater than $\sqrt{nk \times l}$ with l the maximum number of collisions per α_j (roughly, this corresponds to solving a PR instance of length nl). Here $l = T/n$ and the condition $T/M \geq \sqrt{nkl} = \sqrt{Tk}$ is equivalent to the condition $T \geq M^2k$. Due to our hypothesis on the number of eavesdropped messages, the algorithm cannot

be applied. Finally if there exists an adversary able to reconstruct a polynomial with any other strategy, then we can exploit it to simplify the list recovery problem within the mixture model. This would contradict its difficulty when $T/M < \sqrt{nk \times l}$. □

Note that in practice, the cloud of devices is dynamic, some devices may exit or enter the cloud around a verifier, so that the difficulty for the attacker can only increase.

Following this proposition and via the DPR problem, then a passive adversary cannot distinguish the answers as soon as the same interrogation request does not appear twice. The proofs of the following results are in Appendix B.

Proposition 2. *Assume* $\sqrt{q} \geq M \geq e\sqrt{\frac{n}{k}}$ *and* $T < M^2k$. *A passive adversary cannot determine whether two requests correspond to the same CLD except if there is a collision, that happens only with probability* $1/\sqrt{n}$.

5.3 Security against Impersonation

In our protocol, a CLD replies to the verifier only if it believes that the verifier is legitimate. It is thus close to mutual authentication – although here the authentication of the verifier is only probabilistic with respect to the false-positive error rate of an identification code. It is a weaker result than general verifier authentication: a verifier cannot be impersonated in order to interrogate a pre-fixed CLD.

Proposition 3. *Assume* $\sqrt{q} \geq M \geq e\sqrt{\frac{n}{k}}$ *and* $T < M^2k$. *In our scheme, given a non-corrupted CLD, an adversary cannot impersonate a verifier to interrogate this specific CLD, without replaying an eavesdropped transcript, except with probability* $\frac{1}{q}$.

Of course, if no specific CLD is fixed, then impersonation of an interrogation towards a member of a large set of CLDs is easier. With M CLDs, the probability to reach one of them correctly is $\frac{M}{q}$.

Given this difficulty of impersonating a verifier against a chosen CLD and the uselessness of eavesdropping (cf. Proposition 1), we deduce the resistance of CLDs against impersonation attacks.

Proposition 4. *Assume* $\sqrt{q} \geq M \geq e\sqrt{\frac{n}{k}}$ *and* $T < M^2k$. *Our scheme is secure against impersonation of a CLD, i.e. an adversary will fail with probability* $1 - \frac{1}{q}$.

Replay attacks on the verifier side are not important from a security point of view as replaying a query does not give additional information to the adversary. However, they are prevented in the scheme to maintain privacy (with replay attacks, an adversary could track a device).

5.4 Privacy

Proposition 5. *If* $\sqrt{q} \geq M \geq e\sqrt{\frac{n}{k}}$ *and* $T < M^2k$, *then our scheme is weak private.*

See the proofs in Appendix B.

Moreover, even if not forward private, as the identifiers are independently chosen among devices, the corruption of one device directly affects only this device. Although, this level of privacy could seem low, it is exactly what we intended to achieve and it is important to notice that contrary to the protocols described in [28], devices do not need the use of any internal random number generator to implement the protocol.

6 Advantages for Very Low-Cost Devices

For low-cost devices, instead of storing the two polynomial identifiers p, p', we store directly the values $p(\alpha_1), \ldots, p(\alpha_n)$ and $p'(\alpha_1), \ldots, p'(\alpha_n)$ within the device. So doing, no computation is needed on the device side. Depending on the amount of memory available per device, we can also limit the number of such values by restricting ourselves to a basis of evaluation of size $L < n$, e.g. $(\alpha_1, \ldots, \alpha_L)$.

An additional advantage is that the scheme can be adapted simply to work over a noisy channel by storing encoded versions – through some error-correcting code – of these values $p(\alpha_1), \ldots, p(\alpha_L)$ and $p'(\alpha_1), \ldots, p'(\alpha_L)$ and the corresponding index $1, \ldots, L$. The devices will only have to compute the distance between the received message and the stored one.

7 Practical Parameters

For real-life low-cost CLDs, we can imagine a non-volatile memory of about $2^{18} = 256$k bits. We aim at a field size $q = 2^{64}$, which permits to store $2^{12} = 4096$ fields elements in the memory, *i.e.* 2048 evaluations of the two polynomials p_l, p'_l (which implies that the length $n \leq q - 1$ of the corresponding code is $n = 2^{11}$).

With these parameters, we suggest the use of polynomials of dimension $k = 2^8$. Using such a dimension permits to define $q^k = 2^{64 \times 256}$ possible polynomials; the number M of devices needed in the cloud around a verifier has then to be greater than $e \times \sqrt{\frac{n}{k}}$, i.e. at least 8. With $M = 256$, this leads to the restriction $T < 2^{24}$, which is automatically satisfied here as $T \leq Mn = 2^{19}$.

These parameters enable 2048 interrogations of the same device without compromising the device identity - both in terms of impersonation and of weak privacy.

Remark 7. We can suppress the identification-code structure, and replace it with a random one (*i.e.* replace $p(\alpha_i), p'(\alpha_i)$ by random $\beta_i, \beta'_i \in \{0,1\}^{\log_2 q}$). However, instead of storing $k \cdot \log_2 q$ bits per device at the verifier's side, we need to store for each device the $n \cdot \log_2 q$ bits that are stored in it. With these parameters, this implies a storage space 8 times larger.

8 Conclusion

Finally, it is possible to further extend the scheme toward reaching forward privacy (equivalent to destructive privacy in this context of non-correlated identifiers): we store $L < k$ values for each identifier p, p' of degree at most $k-1$ and erase the values $p(\alpha_j)$ and $p'(\alpha_j)$ after replying to the associated query. Because we erase the values after, a corruption will not give direct access to these values and because $L < k$, it is unconditionally impossible for an adversary to recover the missing values by polynomial interpolation. Hence, the destructive privacy is fulfilled. In this case, the false-positive rate should be quite small to avoid quick waste of the coupons of the devices.

Acknowledgements. The authors thank the referees for their helpful comments.

References

1. Ahlswede, R., Dueck, G.: Identification via channels. IEEE Transactions on Information Theory 35(1), 15–29 (1989)
2. Ar, S., Lipton, R.J., Rubinfeld, R., Sudan, M.: Reconstructing algebraic functions from mixed data. SIAM J. Comput. 28(2), 487–510 (1998)
3. Bringer, J., Chabanne, H., Icart, T.: Improved Privacy of the Tree-Based Hash Protocols Using Physically Unclonable Function. In: Ostrovsky, R., De Prisco, R., Visconti, I. (eds.) SCN 2008. LNCS, vol. 5229, pp. 77–91. Springer, Heidelberg (2008)
4. Bringer, J., Chabanne, H., Icart, T.: Efficient Zero-Knowledge Identification Schemes which respect Privacy. In: ACM Symposium on Information, Computer and Communication Security – ASIACCS 2009, Sydney, Australia (March 2009)
5. Arco, P.D., Scafuro, A., Visconti, I.: Semi-destructive privacy in RFID systems. In: Workshop on RFID Security (2009)
6. Eswaran, K.: Identification via channels and constant-weight codes, `http://www.eecs.berkeley.edu/ ananth/229BSpr05/Reports/ KrishEswaran.pdf`
7. Fung, B., Al-Hussaeni, K., Cao, M.: Preserving RFID Data Privacy. In: IEEE International Conference on RFID – RFID 2009, Orlando, Florida, USA (April 2009)
8. Information Security Group. RFID security & privacy lounge, `http://www.avoine.net/rfid/`
9. Guruswami, V., Sudan, M.: Reflections on improved decoding of reed-solomon andalgebraic-geometric codes (2002)
10. Guruswami, V., Sudan, M.: Improved decoding of reed-solomon and algebraic-geometry codes. IEEE Transactions on Information Theory 45(6), 1757–1767 (1999)
11. Juels, A., Weis, S.A.: Defining strong privacy for RFID. In: PERCOMW, pp. 342–347. IEEE Computer Society, Los Alamitos (2007)
12. Kiayias, A., Yung, M.: Polynomial reconstruction based cryptography. In: Vaudenay, S., Youssef, A.M. (eds.) SAC 2001. LNCS, vol. 2259, pp. 129–133. Springer, Heidelberg (2001)

13. Kiayias, A., Yung, M.: Cryptographic hardness based on the decoding of reed-solomon codes. In: Widmayer, P., Triguero, F., Morales, R., Hennessy, M., Eidenbenz, S., Conejo, R. (eds.) ICALP 2002. LNCS, vol. 2380, pp. 232–243. Springer, Heidelberg (2002)
14. Kiayias, A., Yung, M.: Cryptographic hardness based on the decoding of reed-solomon codes with applications. In: Electronic Colloquium on Computational Complexity (ECCC), vol. 017 (2002)
15. Kiayias, A., Yung, M.: Cryptographic hardness based on the decoding of reed-solomon codes. IEEE Transactions on Information Theory 54(6), 2752–2769 (2008)
16. Kurosawa, K., Yoshida, T.: Strongly universal hashing and identification codes via channels. IEEE Transactions on Information Theory 45(6), 2091–2095 (1999)
17. Molnar, D., Wagner, D.: Privacy and security in library RFID: issues, practices, and architectures. In: CCS, pp. 210–219. ACM, New York (2004)
18. Moulin, P., Koetter, R.: A framework for the design of good watermark identification codes. In: Delp III, E.J., Wong, P.W. (eds.) SPIE, vol. 6072, p. 60721H. SPIE, San Jose (2006)
19. Ng, C.Y., Susilo, W., Mu, Y., Safavi-Naini, R.: RFID privacy models revisited. In: Jajodia, S., López, J. (eds.) ESORICS 2008. LNCS, vol. 5283, pp. 251–266. Springer, Heidelberg (2008)
20. Ohkubo, M., Suzuki, K., Kinoshita, S.: RFID privacy issues and technical challenges 48(9), 66–71 (2005)
21. Ouafi, K., Phan, R.C.-W.: Traceable Privacy of Recent Provably-Secure RFID Protocols. In: Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 479–489. Springer, Heidelberg (2008)
22. Paise, R.-I., Vaudenay, S.: Mutual authentication in RFID: security and privacy. In: Abe, M., Gligor, V.D. (eds.) ASIACCS, pp. 292–299. ACM, New York (2008)
23. PEARS. Privacy Ensuring Affordable RFID System. European Project
24. Rieback, M.R.: Security and Privacy of Radio Frequency Identification. PhD thesis, Vrije Universiteit, Amsterdam, The Netherlands (2008)
25. Rudra, A.: List Decoding and Property Testing of Error Correcting Codes. PhD thesis, University of Washington (2007)
26. Sadeghi, A.-R., Visconti, I., Wachsmann, C.: User Privacy in Transport Systems Based on RFID E-Tickets. In: Workshop on Privacy in Location-Based Applications – PILBA 2008, Malaga, Spain (October 2008)
27. Spiekermann, S., Evdokimov, S.: Privacy Enhancing Technologies for RFID - A Critical Investigation of State of the Art Research. In: IEEE Privacy and Security (2009)
28. Vaudenay, S.: On privacy models for RFID. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 68–87. Springer, Heidelberg (2007)
29. Verdu, S., Wei, V.K.: Explicit construction of optimal constant-weight codes for identification via channels. IEEE Transactions on Information Theory 39(1), 30–36 (1993)
30. Weis, S.A., Sarma, S.E., Rivest, R.L., Engels, D.W.: Security and privacy aspects of low-cost radio frequency identification systems. In: Hutter, D., Müller, G., Stephan, W., Ullmann, M. (eds.) Security in Pervasive Computing. LNCS, vol. 2802, pp. 201–212. Springer, Heidelberg (2004)

A Formal Definition of Privacy

The definition given in [28] follows.

Definition 6. *A* **blinded** *adversary uses simulated oracles instead of the oracles* Launch, SendVerifier, SendCLD *and* Result. *Simulations are made using an algorithm called a* **blinder** *denoted* $\mathcal{B}$.

To simulate oracles, a blinder has access neither to the provers secrets nor to the secret parameters KV_s. We denote $\mathcal{A}^{\mathcal{O}}$ the algorithm $\mathcal{A}$ when executed using legitimate oracles and $\mathcal{A}^{\mathcal{B}}$ the algorithm $\mathcal{A}$ when executed using the blinder.

Definition 7. *An adversary is* **trivial** *if there exists a blinder* $\mathcal{B}$ *such that the difference*

$$\left|\Pr\left[\mathcal{A}^{\mathcal{O}} \ wins\right] - \Pr\left[\mathcal{A}^{\mathcal{B}} \ wins\right]\right|$$

is negligible.

Hence, to prove privacy, it suffices to prove that an adversary cannot distinguish between the outputs of the blinder $\mathcal{B}$ and outputs made by legitimate oracles. As stated in [28], this definition of privacy is more general than anonymity and untraceability. To the different kinds of adversaries enumerated above correspond accordingly as many notions of privacy.

Note that Corrupt queries are considered to always leak information on the CLDs' identity. For instance, an adversary can systematically open CLDs in order to track them. In this model, such an adversary is considered as a trivial one because a blinded adversary will succeed in the same way, as the Corrupt oracle is not simulated. Strong privacy is defined only to ensure that CLDs cannot be tracked using their outputs even if their secrets are known.

B Security Proofs

B.1 Security against Impersonation

Proposition 3. *Assume* $\sqrt{q} \geq M \geq e\sqrt{\frac{n}{k}}$ *and* $T < M^2k$. *In our scheme, given a non-corrupted CLD, an adversary cannot impersonate a verifier to interrogate this specific CLD, without replaying an eavesdropped transcript, except with probability* $\frac{1}{q}$.

Proof. To interrogate a CLD, the only useful information for an adversary are the requests made by the verifier. Proposition 1 implies that this does not give an efficient solution to the adversary for obtaining information on one identifier.

Hence, the remaining solution to interrogate a CLD is to try at random to initiate a communication without prior knowledge of its identifier. The question is what is the probability to succeed out of a random couple (j, a)? If a specific CLD with identifier p is targeted, this probability is equal to $\Pr\left[p(\alpha_j) = a\right] = \frac{1}{q}$. □

Proposition 4. *Assume* $\sqrt{q} \geq M \geq e\sqrt{\frac{n}{k}}$ *and* $T < M^2k$*. Our scheme is secure against impersonation of a CLD, i.e. an adversary will fail with probability* $1-\frac{1}{q}$*.*

Proof. As stated in the previous proposition, impersonation of a verifier is not possible except with probability $\frac{1}{q}$ and an adversary would need to succeed at least k times to reconstruct the p' polynomial of a CLD. Moreover, eavesdropping the devices responses does not give a solution to reconstruct an identifier or to obtain information on an identifier, as stated in Proposition 1. Furthermore corruption is not useful here as identifiers are not correlated between CLDs (following Definition 4, the adversary is not allowed to impersonate a corrupted CLD). The best choice for an adversary is thus to try at random. □

B.2 Privacy

Proposition 5. *Assume* $\sqrt{q} \geq M \geq e\sqrt{\frac{n}{k}}$ *and* $T < M^2k$*, then our scheme is weak private.*

Proof. We first prove the narrow-weak privacy; then, Lemma 1 together with Proposition 4 enables us to conclude. It is clear that all oracles are easy to simulate except SendCLD and SendVerifier (Result is not simulated in the narrow case). Concerning the latter, SendVerifier is used to generate an interrogation request; it is simulated simply by sending a random value. As PR instances are not distinguishable from random sequences (cf. [15]), an adversary cannot distinguish the requests from non-simulated ones.

Concerning SendCLD, the simulator needs to simulate the output of a CLD. For this, it can answer only on average to one request over M with a random value. As the adversary cannot impersonate a verifier, he cannot determine if a CLD is answering when beckoned or not. He cannot either distinguish the answered values from PR instances as above. □