Lightweight Privacy Preserving Authentication for RFID Using a Stream Cipher

Olivier Billet*, Jonathan Etrog**, and Henri Gilbert

Orange Labs, Issy-les-Moulineaux, France
billet@eurecom.fr, jonathan.etrog@orange-ftgroup.com,
henri.gilbert@orange-ftgroup.com

Abstract. In this paper, a privacy preserving authentication protocol for RFID that relies on a single cryptographic component, a lightweight stream cipher, is constructed. The goal is to provide a more realistic balance between forward privacy and security, resistance against denial of service attacks, and computational efficiency (in tags and readers) than existing protocols. We achieve this goal by solely relying on a stream cipher—which can be arbitrarily chosen, for instance a stream cipher design aimed at extremely lightweight hardware implementations—and we provide security proofs for our new protocol in the standard model, under the assumption that the underlying stream cipher is secure.

Keywords: RFID protocol, authentication, privacy, DoS resistance, provable security.

1 Introduction

Radio frequency identification, RFID, is a fast expanding technology that allows for the identification of items in an automated way through attached RFID tags, i.e. small low-cost devices equipped with an integrated circuit and an antenna. An RFID system typically consists of three main components: (1) a set of RFID tags, (2) readers capable of communicating with the RFID tags through their radio interface, and (3) a centralized or distributed back-end system connected to the readers through a network. The applications are numerous: automated management of the supply chain, ticketing, access control, automatic tolls, transportation, prevention of counterfeiting, pets tracking, airline luggage tracking, library management, only to name a few. Various RFID systems designed to address these different needs have varying radio and tag power supply characteristics, memory and processing capabilities, and hence costs. Unsurprisingly with such a broad range of applications and physical characteristics, the security and privacy needs for RFID systems are quite diverse:

– Though security and privacy were not felt to be important issues in some initial supply chain management applications where RFID tags were used

* Work performed while at Orange Labs.

** Partially supported by the national research project RFIDAP ANR-08-SESU-009-03.

S. Hong and T. Iwata (Eds.): FSE 2010, LNCS 6147, pp. 55–74, 2010.

as a mere replacement of bar codes and not delivered to the end consumers, the emergence of more and more RFID applications where the tags enter the life of end users (e.g. library management) has resulted in an ever increasing level of concern regarding the potential compromise of their *privacy*. The fear is that through RFID tags attached to the objects she is carrying, a person might leave electronic tracks of her moves and actions and become traceable by a malicious party equipped with a radio device.

– Applications such as ticketing or access control where owning an RFID tag materializes some rights needs to prevent the counterfeiting or impersonation of legitimate RFID tags, which can for instance result from the cloning of a legitimate tag or the replay of data previously transmitted by a legitimate tag. In order to address these *security* needs, an *authentication* mechanism that allows the system to corroborate the identity of the tag is required.

The latter need for security and the former need for privacy are often combined, for instance in the case of ticketing, public transportations, etc. However, as will be seen in the sequel, accommodating both needs for security and privacy in RFID systems using adequate cryptographic solutions is not an easy task, primarily because of strong limitations on computing and communication resources that result in strong cost constraints encountered in most RFID systems, and to a lesser extent because these two requirements are not easy to reconcile. Security and privacy in RFID systems have now become a very active research topic in cryptography, and the design of efficient algorithms and protocols suitable for such systems is a major challenge in the area of *lightweight cryptography*.

Authentication, which addresses the above-mentioned security threat, i.e. preventing the cloning or impersonation of legitimate tags, is probably the most explored topic in lightweight cryptography. Efficient authentication solutions for RFIDs are gradually emerging, even for the most constrained settings. To take into account the strong limitation of computing resources in the tags (3000 GE, is often considered as the upper limit for the area reserved to the implementation of security in low-cost RFID tags), dedicated lightweight block ciphers such as DESXL, PRESENT, and KATAN [35,11,15] have been developed. Such block ciphers can be used for authentication purposes in a traditional challenge-response protocol. Some stream ciphers with a very low hardware footprint, e.g. Grain v1 or Trivium [25,16], are also known to have the potential to lead to extremely efficient authentication solutions. On the other hand, few explicit stream cipher based authentication schemes have been proposed so far; an example is the relatively complex stream cipher based protocol from [36] which requires up to six message exchanges. Lightweight authentication protocols not based on a symmetric primitive like SQUASH [47] and the HB family of RFID schemes [34,23] represent another promising avenue of research, even though it remains a complicated task to identify practical instances from those families resisting all the partial cryptanalysis results obtained so far [42,21,22,41]. In this paper, we will use the following distinction between identification and authentication: a protocol allowing an RFID system to identify a tag, but not to corroborate this identity and thus resist cloning or impersonation attacks will be named an

identification protocol, while a protocol allowing the system to both identify a tag and corroborate this identity will be named an *authentication protocol* or equivalently an authentication scheme. If an authentication protocol additionally results in the corroboration by the tag that the RFID reader involved in the exchange is legitimate, we will call it a *mutual authentication protocol*.

Privacy preserving identification or authentication protocols for RFID have also been much researched in the recent years. It is however fair to say that these still represent a less mature area than mere authentication protocols and designing realistic lightweight protocols that take into account the constraints at both the tag and the reader side remains a very challenging problem. Following the seminal work of [30,33,4,49] definitions and formalizations of various notions of privacy have been proposed and their mutual links have been explored. Without going into the detailed definition of the various privacy notions introduced so far (a rather comprehensive typology is proposed in [49]), it is worth mentioning that a basic requirement on any private RFID identification or authentication protocol is to prevent a passive or active adversary capable of accessing the radio interface from tracing a tag—i.e. to ensure both the *anonymity* and the *unlinkability* of the exchanges of a legitimate tag. This property, named weak privacy by some authors [49], is easy to provide in a symmetric setting, for example by using a lightweight block cipher in a challenge-response protocol and trying the keys of all the tags in the system tags at the reader side in order to avoid transmitting the identity of the tag before the authentication exchange. A significantly more demanding privacy property is **forward privacy**, that is motivated by the fact that the cost of RFID tags renders any physical tamper resistance means prohibitive. In addition to the former weak privacy requirements, a forward private protocol must ensure that an adversary capable of tampering with a tag remains unable to link the data accessed in the tag to any former exchange she might have recorded. It is easy to see that the former simple example of block cipher based protocol is not forward private at all. A paradigmatic example of an RFID identification protocol providing some forward privacy is the OSK scheme [39,40] which relies on the use by the tag of two one-way hash functions.[1]

Variants of the OSK scheme turning it into a forward private authentication protocol have been proposed in [5], thus making it resistant to replay attacks.[2] It was however noticed that the OSK protocol and its authentication variants are vulnerable to Denial of Service attacks (DoS) that desynchronize a tag from the system. Furthermore, such DoS attacks compromise the forward privacy if the adversary can learn whether the identification or authentication exchanges involving a legitimate reader she has access to are successful or not (in this paper the conservative assumption that adversaries have access to this side information

[1] One hash function updates the current state of the tag at each identification while the other derives an identification value from the current internal state. The identification value received by the reader is then searched in the back-end in *hash chains* associated to each tag in the system.

[2] A time-memory trade-off speeding up the back-end computations at the expense of pre-computations was proposed for the original scheme and some of its variants [6].

is made). An alternative to the OSK family of authentication schemes named PFP was recently proposed [10] which is based on less expansive cryptographic ingredients than the one way hash functions involved in OSK (namely a pseudo-random number generator and a universal family of hash functions) and provably offers a strong form of forward privacy under the assumption that the maximum number of authentications an adversary can disturb is not too large. Its main practical drawback is the significant workload at the reader end.

Our contribution. In this paper, we address the problem of rendering forward private authentication protocols fully practical. We show how to convert any lightweight stream cipher such as Grain or Trivium into a simple and highly efficient privacy preserving mutual authentication protocol. Our main motivation is to find a more realistic balance than existing protocols of the OSK family or the PFP protocol between forward privacy, resistance to DoS attacks, and computational efficiency of the tag and the reader. If one accepts to slightly relax the unlinkability requirements in the definition of a forward private protocol (for that purpose we introduce the notion of *almost forward private protocol* and only require our scheme to be almost forward private), we escape the dilemma between forward privacy and resistance to DoS attacks otherwise encountered in a symmetric setting. Desynchronisation can no longer occur even if there is no limitation on the maximum number of authentications an adversary can disturb, and a significant gain in complexity is achieved in the tag and the reader compared to former schemes. We provide formal proofs in the standard model that if the underlying stream cipher is secure then our mutual authentication protocol is correct, secure, and almost forward private.

We provide the definitions of the security and privacy properties required for an RFID authentication protocol and introduce the security notions needed in the subsequent proofs in Section 2. We describe our mutual authentication protocol in Section 3 and prove its security, almost privacy, and correctness in Section 4. In Section 5, we briefly discuss implementation.

2 Security and Privacy Model

In this section we introduce a simple security and privacy model inspired from [10], an adaptation of the more comprehensive typology of security and privacy models introduced by Vaudenay in [49] to the symmetric setting where (1) the internal states of the tags are initialized with independent individual secret keys and (2) these initial internal states are updated throughout the lifetime of the tags. The main differences with the security and privacy model of [10] lie in the modification of the definition of a secure protocol to address mutual authentication instead of authentication (similar to the adaptation of [49] in [43]), in the adapted definition of correctness to systems with unlimited lifetime, and in the introduction of a distinction between the notions of forward private protocol and the slightly relaxed notion of *almost forward private* protocol.

Assumptions. We denote by N the number of initialized tags of the system. During their lifetime, initialized tags enter mutual authentication exchanges with

a reader. Each exchange results in (possibly distinct) success or failure outcomes at both sides. A mutual authentication exchange involving a legitimate tag and a reader is said to be undisturbed if all messages sent by all parties are correctly transmitted and neither modified nor lost in either direction. We consider powerful active adversaries capable of tracking an individual tag during a limited time period named an *exposure period*, i.e. to identify and read the messages exchanged by this tag and its reader, to modify these messages while they are transmitted or to themselves transmit messages viewed by one side as coming from the opposite side, and finally to access the authentication success or failure information at both ends (reader and tag) at the completion of a mutual authentication exchange. In other words, we consider active adversaries capable of performing man in the middle attacks. We assume that after an exposure period of a tag, no physical characteristics of the tag nor information unrelated to mutual authentication exchanges allow an adversary to differentiate it from any of the $N-1$ other tags.

2.1 Security

We say that a mutual authentication protocol is secure if it resists impersonation attacks. An impersonation attack proceeds in two phases. During the first phase (assumed, without loss of generality, to take place during a single exposure period) an adversary interacts both with a legitimate reader and a legitimate tag $\mathcal{T}_i$ and is allowed to trigger, observe, and disturb or entirely replace up to q mutual authentication exchanges involving the tag $\mathcal{T}_i$ and the reader, and to access the outcomes of the authentication (success or failure). During the second phase, he only interacts with the reader (or with the tag $\mathcal{T}_i$, depending on which party is being impersonated) and initiates a mutual authentication exchange to impersonate the tag $\mathcal{T}_i$ (respectively the reader). The impersonation succeeds if the mutual authentication is successful and the adversary is identified as tag $\mathcal{T}_i$ (respectively a legitimate reader).

Definition 1. *A mutual authentication protocol is said to be (q, T, ϵ)-secure iff for any adversary running in time upper-bounded by T, the probability that an impersonation attack involving at most q authentication exchanges during phase 1 be successful is at most ϵ.*

2.2 Forward Privacy

Let us consider the following forward privacy experiment involving a (q, T)-privacy adversary A with a running time upper-bounded by T. During a first phase, A interacts with any two legitimate tags $\mathcal{T}_{i_0}$ and $\mathcal{T}_{i_1}$, and a legitimate reader. These interactions happen, without loss of generality, during a single exposure period of $\mathcal{T}_{i_0}$ and a single exposure period of $\mathcal{T}_{i_1}$ where the adversary is allowed to trigger, observe, and disturb at most q mutual authentication exchanges involving $\mathcal{T}_{i_0}$ and possibly the reader and at most q mutual authentication exchanges involving $\mathcal{T}_{i_1}$ and possibly the reader. During a second phase, A

again interacts with a tag $\mathcal{T}_{i_b}$ randomly selected among the two tags $\mathcal{T}_{i_0}$ and $\mathcal{T}_{i_1}$, and b is concealed to A. First, A is allowed to trigger, observe, and disturb at most q additional mutual authentication exchanges involving $\mathcal{T}_{i_b}$ and is given access to the corresponding mutual authentication outcome (success or failure). Then, A is given access to the internal state value of $\mathcal{T}_{i_b}$. Eventually, A outputs a guess b' for the value of b, and succeeds if b' is equal to b.

Definition 2. *An RFID mutual authentication protocol is said* (q, T, ϵ)*-private iff any* (q, T)*-privacy adversary in the above game has an advantage at most* ϵ*:*

$$\left|\Pr\left[A \text{ succeeds}\right] - \frac{1}{2}\right| \leq \epsilon \ .$$

We now slightly relax the above forward privacy requirements by introducing the notion of *almost forward privacy* We only require that adversaries be unable to link the internal state recovered when tampering with a tag with any mutual authentication exchanges involving the tag up to the last successful authentication exchange of the tag. This removes the constraint that adversaries be unable to link a failed mutual authentication exchange of a tag with its internal state immediately after the failed exchange. (A similar limitation of the considered privacy attacks is also encountered in the privacy notion proposed in [48].) In real life scenarios, almost forward privacy seems to be a relevant privacy notion. Let us for instance assume that tags are monthly access passes. To thwart adversaries who first collect information from tags by eavesdropping legitimate readers or using false readers, and try later on to correlate this information to (say) thrown tags, almost forward privacy is sufficient in practice. To define a (q, T)-almost private adversary A, we therefore restrict the former forward privacy experiment as follows. During a first phase, A interacts with any two legitimate tags exactly in the same way as in the first phase of the former definition. Before the second phase, an undisturbed exchange between a legitimate reader and each of the two tags $\mathcal{T}_{i_0}$ and $\mathcal{T}_{i_1}$ takes place and A is assumed not to have access to this exchange. During the second phase, A interacts with a tag $\mathcal{T}_{i_b}$ randomly selected among the two tags $\mathcal{T}_{i_0}$ and $\mathcal{T}_{i_1}$ exactly in the same way as in the former definition. A is finally given access to the internal state value of $\mathcal{T}_{i_b}$, outputs a guess b' for the value of b, and succeeds if b' is equal to b.

Definition 3. *An RFID mutual authentication protocol is said to be* (q, T, ϵ)*-almost forward private iff any* (q, T)*-almost privacy adversary in the above game has an advantage at most* ϵ*:*

$$\left|\Pr\left[A \text{ succeeds}\right] - \frac{1}{2}\right| \leq \epsilon \ .$$

2.3 Correctness

We first define the notion of correctness in a setting where the mutual authentication exchanges of legitimate tags are not disturbed by transmission errors or by an adversary. In such a setting, the protocol executions of a legitimate tag $\mathcal{T}_i$ must succeed with overwhelming probability, i.e. result in an authentication success outcome at both sides and a correct identification of $\mathcal{T}_i$ by the reader.

Definition 4. *An RFID mutual authentication protocol is ϵ-correct iff for any legitimate tag $\mathcal{T}_i$, the probability (over the initial secrets of the legitimate tags in the system and the random numbers chosen during the execution of the protocol) that an undisturbed execution of the protocol between $\mathcal{T}_i$ and a legitimate reader fails is upper-bounded by ϵ.*

We further extend the former definition of correctness by considering a setting where the mutual authentication exchanges of legitimate tags may be disturbed by a DoS adversary who succeeds if she causes the failure of a mutual authentication attempt of a legitimate tag with a legitimate reader. This allows to incorporate resistance to DoS attacks into the definition of correctness. Although we consider a unique adversary, this is not restrictive since situations where transmission errors occur and/or where mutual authentication exchanges are disturbed by a coalition of adversaries can be viewed as coming from a single adversary. We introduce limitations on the capabilities of the adversary: an adversary with a running time upper-bounded by T and able to disturb at most q mutual authentication exchanges is called a (q, T)-adversary.

The correctness experiment proceeds in two phases. During the first phase the (q, T)-adversary interacts with the whole system. During the second phase, an undisturbed execution of the protocol between $\mathcal{T}_i$ and a legitimate reader occurs. The adversary succeeds if the mutual authentication protocol execution fails.

Definition 5. *An RFID mutual authentication protocol is said (q, T, ϵ)-correct iff the probability (over the initial secrets of the legitimate tags in the system, the random numbers chosen during the executions of the protocol, and the random numbers used by the adversary) that an undisturbed execution of the protocol between any tag $\mathcal{T}_i$ and a legitimate reader fails is upper-bounded by ϵ, even in the presence of a (q, T)-adversary.*

2.4 Definitions and Properties

We now introduce a few general security definitions. The starting point for the construction of our mutual authentication protocol is a stream cipher such as Grain or Trivium [25,16], that takes a secret key and a non-secret initialization value (IV) as input and produces a binary sequence (the keystream). An IV-dependent stream cipher of key length k bits and IV length n bits that produces a keystream sequence of length up to m bits can be conveniently represented as a family of functions $F = \{f_K\} : \{0,1\}^n \rightarrow \{0,1\}^m$ indexed by a key K randomly chosen from $\{0,1\}^k$; f_K thus represents the function mapping the IV to the keystream associated with key K. For such an IV-dependent stream cipher to be considered secure when producing keystreams of length at most m bits, one usually requires [9] that the associated family of functions F be a pseudo-random function (PRF). In order to formalize and quantify what we mean by a secure stream cipher, we therefore need to introduce the notion of PRF distinguisher.

Definition 6 (PRF distinguisher). *Let $F = \{f_K\} : \{0,1\}^n \rightarrow \{0,1\}^m$ be a family indexed by a key K randomly chosen from $\{0,1\}^k$. A PRF distinguisher*

for F is a probabilistic testing algorithm A modeled as a Turing Machine with a random tape and an oracle tape, that produces a binary output 0 or 1 and is able to distinguish a randomly chosen function f_K of F from a perfect random n-bit to m-bit function f^* with an advantage

$$\mathrm{Adv}_F^{PRF}(A) = \left|\Pr\left(A^{f_K} = 1\right) - \Pr\left(A^{f^*} = 1\right)\right|$$

where the probabilities are taken over K and over all the random choices of A. So we define the (q, T) PRF advantage for distinguishing the family F as

$$\mathrm{Adv}_F^{PRF}(q, T) = \max\left\{\mathrm{Adv}_F^{PRF}(A)\right\}$$

where the maximum is taken over all the possible attackers A working in time at most T and able to query an n-bit to m-bit oracle up to q times. We will call the family a PRF if this advantage smaller than a threshold for T and q suitably chosen to reflect realistic upper limits for the resources of an adversary.

Definition 7 (Secure stream cipher). *An IV-dependent stream cipher associated with a family $F = \{f_K\}$ of IV to keystream functions is (q, T, ϵ)-secure if $\mathrm{Adv}_F^{PRF}(q, T) \leq \epsilon$.*

Note that in the above definition of a PRF distinguisher, the experiment performed by a PRF distinguisher involves a single randomly chosen instance of F. In the stream cipher based construction presented in this paper, the keystream output by the stream cipher is however used to produce the key used during the next invocation of the stream cipher. Therefore the proofs of this construction require to consider testing experiments involving several instances of F instead of a single one. We address this issue by introducing the notion of *multiple oracle PRF distinguisher*. To avoid some confusions we sometimes use the name *single oracle distinguisher* to refer to the former (classical) notion of PRF distinguisher.

Definition 8 (Multiple oracle PRF distinguisher). *Let us consider a family $F = \{f_K\} : \{0,1\}^n \to \{0,1\}^m$ indexed by a key K randomly chosen from $\{0,1\}^k$. A multiple oracle PRF distinguisher for f is a probabilistic testing algorithm A distinguishing λ randomly chosen instances f_{K_i} of F from λ independent perfect random n-bit to m-bit functions f_i^* $(i = 1, \ldots, \lambda)$ with an advantage*

$$\mathrm{Adv}_F^{PRF}(A) = \left|\Pr\left(A^{(f_{K_i})_{i=1,\ldots,\lambda}} = 1\right) - \Pr\left(A^{(f_i^*)_{i=1,\ldots,\lambda}} = 1\right)\right|$$

where the probabilities are taken over K and over all the random choices of A. So we define the (λ, q, T) PRF advantage for distinguishing the family F as

$$\mathrm{Adv}_F^{PRF}(\lambda, q, T) = \max\left\{\mathrm{Adv}_F^{PRF}(A)\right\}$$

where the maximum is taken over all the possible attackers A working in time at most T and able to query up to λ n-bit to m-bit oracles up to q times each.

Theorem 1 (Link between single and multiple oracle distinguishers). *Let us denote by* $F = \{f_K\} : \{0,1\}^n \rightarrow \{0,1\}^m$ *a family of functions with* $K \in \{0,1\}^k$. *The resistance of* F *against* λ*-oracle distinguishers is related to its resistance against single-oracle distinguishers attackers via the following formula (where* T_F *is the time needed to compute one instance of* F *in one point):*

$$\mathrm{Adv}_F^{PRF}(\lambda, q, T - \lambda T_F) \leq \lambda \mathrm{Adv}_F^{PRF}(q, T)$$

Proof. See Appendix A.

Lemma 1 (PRF Product). *If* F *is indistinguishable of* F^* *in time* T, *with* q *queries and with an advantage greater than* ϵ_1 *and if* G *is indistinguishable of* G^* *in time* T, *with* q *queries and with an advantage greater than* ϵ_2 *then* $F \times G = \{f_{K_1}, g_{K_2}\} : \{0,1\}^n \rightarrow \{0,1\}^m \times \{0,1\}^m$ *is indistinguishable of* $F^* \times G^*$ *using* q *queries, in time* $T - qT_G$ *with an advantage greater than* $\epsilon_1 + \epsilon_2$.

Proof. See Appendix B.

3 A Stream Cipher Based Protocol

3.1 DoS Resistance and Privacy

To achieve resistance against DoS attacks, a natural idea is to use mutual authentication instead of one way authentication so that the tag only updates its internal state after the reader has been authenticated. However, as discussed in [10], it is not possible to aim for full DoS resistance and forward privacy in symmetric key based protocols. We thus need to find a trade-off and it seems a reasonable approach to somewhat relax the privacy requirements while keeping full DoS resistance with mutual authentication. Different protocols have been designed to achieve both DoS resistance and almost-forward privacy. An example in the OSK family is the C2 protocol [12] which reaches this goal by using cryptographic hash functions. Hash functions are unfortunately prohibitively expensive for RFID tags and have security properties (e.g. collision resistance) that are unnecessary in these applications. While the S-protocol [36] has no privacy goals, an example of a stream cipher based privacy preserving protocol is the recently proposed protocol O-FRAP and its variants [48] but it does not achieve almost-forward privacy because it stores a pseudo-random number in the tag that is transmitted during the last pass of the protocol. Due to this feature, an attacker can compare the value of the pseudo-random number found by tampering with a tag with the last pseudo-random number used by an unknown tag to immediately determine whether it is the same tag or not.

3.2 Our Protocol: PEPS

We present a DoS-resistant, almost-forward private mutual authentication RFID protocol accommodating any secure IV-dependent stream cipher which we call

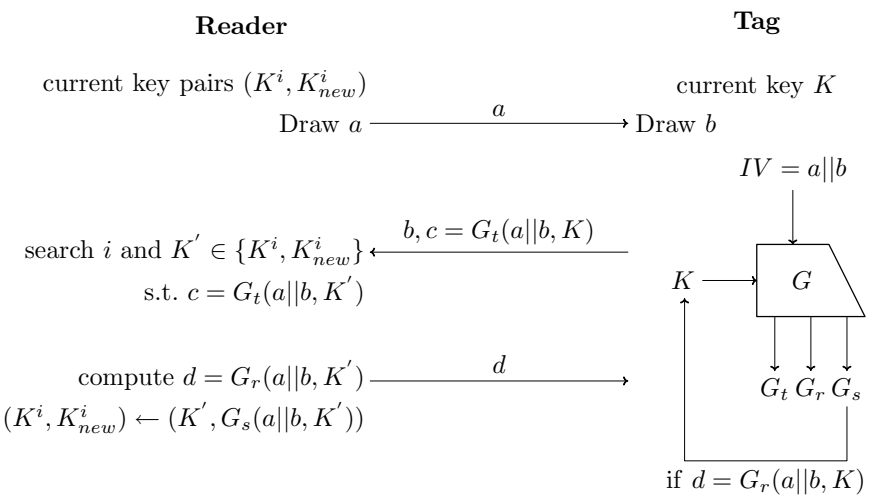

Fig. 1. Our protocol: PEPS

PEPS: a Private and Efficient Protocol based on a Stream cipher. The stream cipher is keyed with the current internal state K of the tag. The initial value of K is randomly selected at the tag initialization and known by the RFID system. The conducting idea of our design is to use the input expanding PRF G associated with the stream cipher with input values resulting from random numbers generated by the tag and the reader in order to (1) generate mutual authentication responses at both sides and (2) refresh the current internal state of the tag in a simple three-pass protocol. In order to avoid any desynchronisation due to lost messages or DoS attacks, the back-end system keeps and updates, for each active tag $\mathcal{T}_i$ of the system, a pair (K^i, K^i_{new}) of potential current keys for $\mathcal{T}_i$.

More explicitly, let us denote by K the k-bit key and by IV the n-bit IV of the stream cipher. The stream cipher is used to produce a keystream sequence $G(K, IV)$ of length $m = 2l + k$, where l represents the length of the authentication responses of our protocol, and the keystream $G(K, IV)$ is viewed as the concatenation $G_t(K, IV)||G_r(K, IV)||G_s(K, IV)$ of three subsequences of respective lengths l, l, and k. Thus G_t and G_r produce l-bit sequences while G_s produces a k-bit sequence (note that the symbols t, r, and s stand here for "tag", "reader", and "secret").

When tag $\mathcal{T}_i$ is initialized, a random initial internal value K^i_0 is drawn and installed in the tag. At the back-end side, the current pair (K^i, K^i_{new}) associated with tag $\mathcal{T}_i$ is initialized with $(K^i_0, \_)$. An execution of the mutual authentication protocol between a tag and a reader is illustrated by Figure 1. It works as follows: first the reader randomly generates an authentication challenge a of length $\frac{n}{2}$ bits and sends it to the tag. At the receipt of a, the tag (whose current key value is denoted by K) randomly generates a $\frac{n}{2}$-bit number b, derives

the value $IV = a||b$, and computes $G(K, IV)$ using the stream cipher. Then it sends $(rand_t, c)$—where $c = G_t(K, IV)$—to the reader. The reader authenticates the tag by searching a tag index i and a key $K^{'} \in \{K^i, K^i_{new}\}$ such that $G_t(K^{'}, IV) = c$. The key K' represents the conjectured internal state of the tag from the reader's "point of view". If the reader finds such an index i then the tag is considered as successfully authenticated as tag $\mathcal{T}_i$, otherwise the outcome of the authentication exchange is an authentication failure. (In the case of an authentication failure, the reader can then either terminate the exchange or send a dummy message back to the tag. This does not matter here due to the fact that we assume that adversaries have access to the positive or negative authentication outcome anyway.) If the tag has been authenticated as tag $\mathcal{T}_i$ the reader updates the current pair associated with tag $\mathcal{T}_i$ to $(K^{'}, G_s(K^{'}, IV))$, computes the reader authentication answer $d = G_r(K^{'}, IV)$ and sends it back to the tag. At the receipt of the reader's answer the tag checks whether $d = G_r(K, IV)$. If this equality holds, it replaces its current key value K by $G_s(K, IV)$: the reader is considered as successfully authenticated and this terminates the mutual authentication exchange. Otherwise it keeps its key value.

4 Security, Almost Forward Privacy, and Correctness

To prove the security and privacy properties of PEPS, we essentially need to show that any information available to the attacker defined previously behaves pseudo-randomly. We denote $G_t(K, .)||G_r(K, .) : \{0,1\}^n \to \{0,1\}^{2l}$ by f^1_K and $G_s(K, .) : \{0,1\}^n \to \{0,1\}^k$ by f^2_K. If G is a PRF, $\{f^1_K\}$ and $\{f^2_K\}$ are obviously also PRFs satisfying the same indistinguishability bounds. Figure 2 shows every possible output with known or partly chosen input an adversary can access to in a security or privacy experiment involving PEPS. In other words, attackers of the system have access to a *composed function* and we show that this function is indistinguishable of an ideal one resulting from a sequence of independent perfect random functions (with the right number of arguments for each coordinate); these independent random functions are shown in Figure 2. More formally:

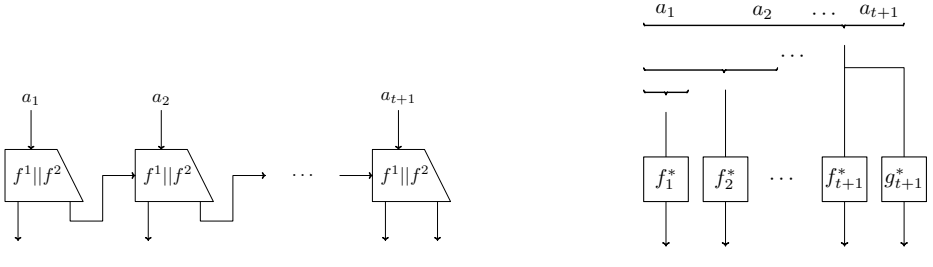

Fig. 2. On the left F_{t+1}, on the right F^*_{t+1}

Theorem 2 (Composition). *If* $F = \{f_K^1 || f_K^2 \ : \{0,1\}^n \to \{0,1\}^{2l} \times \{0,1\}^k\}$ *is a PRF with* $\mathrm{Adv}_F^{PRF}(q,T) \leq \epsilon$ *then for any integer* t

$$F_t = \Bigg\{ (a_1, \ldots, a_t) \to \Big(f_K^1(a_1), f_{f_K^2(a_1)}^1(a_2), \ldots, f^1_{f^2_{\cdots f^2_{f^2_K(a_1)}(a_2)}(a_{t-1})}(a_t), f^2_{f^2_{\cdots f^2_{f^2_K(a_1)}(a_2)}(a_{t-1})}(a_t) \Big) \Bigg\}$$

is indistinguishable from the ideal function generator

$$F_t^* = \{(a_1, \ldots, a_t) \to (f_1^*(a_1), f_2^*(a_1, a_2), \ldots, f_t^*(a_1, \ldots, a_t), g_t^*(a_1, \ldots, a_t))\}$$

where $f_i^* : \{0,1\}^{ni} \to \{0,1\}^{2l}$ *and* $g_t^* : \{0,1\}^{nt} \to \{0,1\}^k$ *are independent random functions using* q *queries, in time* $T - (t-1)qT_{PRF}$ *and with advantage greater than* $((t-1)q+1)\epsilon$.

Proof. See Appendix C.

4.1 Security of the Scheme

Theorem 3 (Security). *If* $\mathrm{Adv}_G^{PRF}(q,T) \leq \epsilon$ *then PEPS is* $(q-1, T - q(q-1)T_G, \epsilon_s)$*-secure with* $\epsilon_s = \frac{q-1}{2^{\frac{n}{2}}} + \frac{1}{2^l} + (q(q-1)+1)\epsilon$.

A $(q-1,T)-$attacker A against PEPS succeeds if it is authenticated as the legitimate tag by the legitimate reader or as the legitimate reader by the legitimate tag. We will denote the different internal states of the tag through the different updates during the experiment by $K_i, 1 \leq i \leq I$ and A's queries by x_l. In the first phase the attacker A collects some $(G_t(K_i, x_l), G_r(K_i, x_l))$. Then in the second one it has to guess $G_t(K_I, x)$ or $G_r(K_I, x)$ when it is challenged by a legitimate part. We use A to construct a distinguisher B for F_q. We still use the notation $f_K^1(x) = G_t(K,x)||G_r(K,x)$ and $f_K^2(x) = G_s(K,x)$. B works as follows: it simulates a tag and a reader to answer A's queries using its oracle $F = (f_1, f_2, \ldots, f_q, g_q)$. The f_i are used to simulate both the tag's and reader's behavior, for example to simulate the tag, B generates a random r and computes some $f_i(((x_j)||(r_j))_{j \leq i})$ as answers or verifies an equality to know if the state should be updated, in this case B keeps in memory the challenge a which provokes the update, adds it to the list a_i of its memories challenges and will use the next coordinate with the tuple $((a_i)_i)$ for the first arguments for the sequel of the simulation. Finally if in the second phase the value computed by A matches the correct value as verified by B using an additional query then B outputs '1' (i.e. it guesses $F \in F_q$), otherwise it outputs '0' (i.e. it guesses $F \in F_q^*$). Clearly B uses q queries and runs in the same time as A and since $Adv_G^{PRF}(q,T) \leq \epsilon$, Theorem 2 upper bounds B's advantage by $((t-1)q+1)\epsilon$ with $t = q$. So A's success probability is upper bounded by $(q(q-1)+1)\epsilon + \epsilon_r$ where ϵ_r is the maximum of the success probability of a $(q-1, T - q(q-1)T_G)$-attacker in the case where he has access to F_q^*. In this case it is obvious that the best strategy

for an attacker is to make the state of its target constant and to make the same challenges each time, hoping that the last challenge will be equal to one of the previous ones. So in the random case an attacker has a probability of success upper bounded by $\frac{q-1}{2^{\frac{n}{2}}} + \frac{1}{2^l}$ where the first term corresponds to the case where the last challenge is equal to one of the previous ones and the second term is the probability of a random guess of an unknown challenge. This ends the proof.

In order to allow the reuse of this security result in the proof of correctness, hereafter, we introduce an extended notion of security. The adversary is now considered successful if it manages to be successfully authenticated as one of the legitimate tags of the system or as a legitimate reader. We have the following:

Theorem 4. *If* $\mathrm{Adv}_G^{PRF}(q,T) \leq \epsilon$ *then in a system with* N *tags and a legitimate reader PEPS is* $(q-1, T-q(q-1+N)T_G, \epsilon_S)$*-secure (under the extended security notion introduced above) with* $\epsilon_S = \frac{q-1}{2^{\frac{n}{2}}} + \frac{N}{2^l} + N(q(q-1)+1)\epsilon$.

Proof. A $(q-1,T)-$attacker A which interacts with different tags has access to different instances of F_q so we use it to derive a multiple oracle distinguisher against F_q with N instances (B needs to ask each instance corresponding to each tag in the system to simulate the reader), using at most q queries to each and working in time $T - q(q-1+N)T_G$ with a similar process than in the proof of Theorem 3. As $T_{F_q} = qT_F$, Theorem 1 upper bounds B's advantage by $N(q(q-1)+1)\epsilon$ and a similar proof to the one of Theorem 3 upper bounds the advantage of an attacker running in time $T - q(q-1+N)T_G$ against the system in the case where the oracles are random by $\frac{q-1}{2^{\frac{n}{2}}} + \frac{N}{2^l}$, where the first term corresponds to the probability that the last challenge from the target matches a previous one and the second term is the probability of a random guess of an unknown challenge among the N tags.

4.2 Almost Forward Privacy of the Scheme

Theorem 5 (Almost forward privacy). *If* $\mathrm{Adv}_G^{PRF}(2q+1,T) \leq \epsilon$ *then PEPS is* (q, T', ϵ_f)*-almost forward private with* $T' = T - q(2q+1)T_G$ *and* $\epsilon_f = \frac{q}{2^{\frac{n}{2}-1}} + \frac{1}{2^{l-1}} + 2(q(q+1)+1)\epsilon + (2q+1)((2q(2q+1)+1) + ((2q+1)(q-1)+1))\epsilon$.

Proof. We consider a (q,T')-privacy attacker A with advantage ϵ_a and we need to prove that $\epsilon_a \leq \epsilon_f$. As this is obvious if $\epsilon_a \leq 2\epsilon_s = \frac{q}{2^{\frac{n}{2}-1}} + \frac{1}{2^{l-1}} + 2(q(q+1)+1)\epsilon$ we assume that $\epsilon_a \geq 2\epsilon_s$. We are using the notation $F_{i,j}^* = F_i^* \times F_j^*$ and $F_{i,j} = F_i \times F_j$. We will denote by α the number of updates of the state of the tag $\mathcal{T}_b$ during the almost forward privacy experiment conducted by A. We use A to derive a distinguisher B between $\{F_{i,q}\}$ and $\{F_{i,q}^*\}$ for a randomly chosen integer i such that $1 \leq i \leq 2q+1$. B uses its oracle to simulate the system to A as previously, B works in the same time than A and uses $2q+1$ queries. If $\alpha \neq i$ (which means that B cannot answer correctly when A asks the internal state) then B aborts the simulation and returns a random guess. We note $\mathcal{Q}$ the event that $\alpha = i$ and $\mathcal{S}$ the event that the undisturbed execution of the protocol between $\mathcal{T}_b$ and the reader has been successful. In the case where we have both $\mathcal{S}$

and $\mathcal{Q}$, then B perfectly simulates the behavior of the system so its probability of success is exactly that of A. In the sub-case where B's oracle is in $F^*_{\alpha,q}$, the oracles used in phase 2 are independent random functions that are independent of the random functions used in phase 1, and so the probability of success of A is exactly $\frac{1}{2}$. By Lemma 1, $((2q(2q+1)+1)+((2q+1)(q-1)+1))\epsilon$ upper-bounds the advantage of B. Since the probability that the undisturbed execution of the protocol at the beginning of phase 2 does not succeed is upper-bounded by the probability that a (q, T')-attacker against the security of the scheme succeeds in phase 1, Theorem 3 shows that it is upper bounded by $\frac{q}{2^{\frac{n}{2}}} + \frac{1}{2^t} + (q(q+1)+1)\epsilon$. Now we have $\Pr\left[B^f = 1\right] = \frac{1}{2q+1}\Pr\left[B^f = 1|Q\right] + (1 - \frac{1}{2q+1})\frac{1}{2}$ together with $\Pr\left[B^f = 1|Q\right] = \Pr[S]\Pr\left[B^f = 1|S, Q\right] + \Pr\left[\neg S\right]\Pr\left[B^f = 1|\neg S, Q\right]$ which implies $\Pr\left[B^{f_K \in F_{i,q}} = 1|Q\right] \geq \epsilon_a + \frac{1}{2} - \epsilon_s$ and $\Pr\left[B^{f^* \in F^*_{i,q}} = 1|Q\right] \leq \frac{1}{2} + \epsilon_s$. Therefore we have $\left|\Pr\left[B^{f_K \in F_{i,q}} = 1\right] - \Pr\left[B^{f^* \in F^*_{i,q}} = 1\right]\right| \geq \frac{1}{2q+1}(\epsilon_a - 2\epsilon_s)$ so $\epsilon_a \leq (2q+1)\mathrm{Adv}^{PRF}_{F_{i,q}}(B) + 2\epsilon_s$ which concludes the proof.

4.3 Correctness of the Scheme

Theorem 6 (Correctness). *If* $\mathrm{Adv}^{PRF}_G(q, T) \leq \epsilon$ *with* $T \geq NT_c + (1+2N)T_G$ *then PEPS is* $(q-1, T - q(q-1+N)T_G, \epsilon_c)$*-correct where* T_c *denotes the time needed to store the answer of one oracle and* $\epsilon_c = 2N\epsilon + \frac{N(N-1)}{2^l} + \frac{q-1}{2^{\frac{n}{2}}} + \frac{N}{2^l} + N(q(q-1)+1)\epsilon$.

Proof. We denote the current key pair for tag $\mathcal{T}_i$ by (K^i_1, K^i_2) instead of (K^i, K^i_{new}) to simplify the notation of the proof. The failure of the final authentication can only come from two scenarii: the attacker has provoked an undesired update during the first phase or a collision occurs during the final authentication and provokes the incorrect identification of $\mathcal{T}_i$ as another tag. As the first event has a probability bounded by Theorem 4 we only need to upper bound the probability of a collision. To upper bound the probability of a collision between a given $G_t(K^i, x)$ and $G_t(K^{i'}_b, x)$ we construct a multiple oracle distinguisher B for F_2 which queries each of its N instances f_i of its oracle with one random query x and compare for each i the values $f_i(x)$. If B finds a collision between the first l bits of the first coordinate of f_{i_0} with the first l bits of the first or the second coordinate of another f_i then it guesses F_2 otherwise it guesses a truly random function generator. B works in time $NT_c \leq T - (1+2N)T_G$. As previously $\mathrm{Adv}^{PRF}_{F_2}(B) \leq 2N\epsilon$. For a truly random function generator the probability of a collision is upper bounded by $\frac{N(N-1)}{2^l}$ so the probability of the collision in the case of F_2 is upper bounded by $\frac{N(N-1)}{2^l} + 2N\epsilon$. We add these probabilities with the probability of an impersonation attack of a $(q-1, T - q(q-1+N)T_G)$-attacker in the whole system by the attacker to find ϵ_c.

5 Efficient Implementation of PEPS

The eSTREAM project [18] has lead to the design of several stream ciphers which offers very lightweight hardware implementation [24]. The hardware foot-

print of some implementations of Grain and Trivium (two stream ciphers with a hardware profile selected in the eSTREAM portfolio) is quite low: Grain uses 1294 GE, Trivium 2580 GE, both conjecturing an 80-bit security. It is also possible to use stream ciphers offering some provable security arguments and efficient implementations. An example is QUAD [8] also conjecturing 80-bit security for instances of the algorithm requiring less than 3000 GE to implement [1].

Also, note that a very interesting feature of our design is that it allows for an easy bit by bit processing. Therefore, when the key and IV setup of the underlying stream cipher also loads the key and the IV bit by bit, it is possible to implement the protocol inside the tag with just a few additional GE for the storage of the next key K. To see this, note that once the key is loaded, the tag can load its seed b at the same time as it outputs it. Then, it switches to keystream production mode and outputs c bit by bit. Finally, as it inputs d, it checks it by comparing it bit by bit to the keystream bits it produces. Eventually, if d was correct, it accumulates the next key in a buffer. In the case of Grain with an 80-bit secret key, this strategy only increases the size by about 4×80 GE, leading to an overall implementation of size about 1700 GE.

6 Conclusion

We presented an RFID protocol that provably achieves both DoS-resistance and a very strong form of privacy, close to the notion of forward privacy. Our protocol can be instantiated with any secure stream cipher, and choosing a stream cipher that admits a very low hardware complexity demonstrated that our protocol is also suitable for the highly constrained setting of RFID systems.

References

1. Arditti, D., Berbain, C., Billet, O., Gilbert, H.: Compact FPGA implementations of QUAD. In: Bao, F., Miller, S. (eds.) ASIACCS 2007. ACM, New York (2007)
2. Auto-ID Center. 860MHz 960MHz Class I RFID Tag Radio Frequency & Logical Communication Interface Spec., v1.0.0. RR MIT-AUTOID-TR-007 (2002)
3. Avoine, G.: Privacy Issues in RFID Banknote Protection Schemes. In: Quisquater, J.-J., Paradinas, P., Deswarte, Y., Abou El Kadam, A. (eds.) CARDIS 2004, pp. 33–48. Kluwer, Dordrecht (2004)
4. Avoine, G.: Adversarial model for radio frequency identification. Cryptology ePrint Archive, Report 2005/049 (2005), `http://eprint.iacr.org/`
5. Avoine, G., Dysli, E., Oechslin, P.: Reducing Time Complexity in RFID Systems. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 291–306. Springer, Heidelberg (2006)
6. Avoine, G., Oechslin, P.: A Scalable and Provably Secure Hash Based RFID Protocol. In: PerSec 2005. IEEE Computer Society Press, Los Alamitos (2005)
7. Avoine, G., Oechslin, P.: RFID traceability: A multilayer problem. In: Patrick, A., Yung, M. (eds.) FC 2005. LNCS, vol. 3570, pp. 125–140. Springer, Heidelberg (2005)
8. Berbain, C., Gilbert, H., Patarin, J.: QUAD: A practical stream cipher with provable security. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 109–128. Springer, Heidelberg (2006)

9. Berbain, C., Gilbert, H.: On the security of IV dependent stream ciphers. In: Goos, G., Hartmanis, J., van Leeuwen, J. (eds.) FSE 2007. LNCS, vol. 4593, pp. 254–273. Springer, Heidelberg (2007)
10. Berbain, C., Billet, O., Etrog, J., Gilbert, H.: An Efficient Forward-Private RFID Protocol. In: ACM CCS 2009 (2009)
11. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An Ultra-Lightweight Block Cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
12. Canard, S., Coisel, I.: Data Synchronization in Privacy-Preserving RFID Authentication Schemes. In: Conference on RFID Security (2008)
13. CASPIAN, `http://www.spychips.com`
14. Damgård, I., Østergaard, M.: RFID Security: Tradeoffs between Security and Efficiency. Cryptology ePrint Archive, Report 2006/234 (2006)
15. De Cannière, C., Dunkelman, O., Knežević, M.: KATAN and KTANTAN—A Family of Small and Efficient Hardware-Oriented Block Ciphers. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 272–288. Springer, Heidelberg (2009)
16. De Cannière, C., Preneel, B.: Trivium. In: Robshaw, M.J.B., Billet, O. (eds.) New Stream Cipher Designs: The eSTREAM Finalists. LNCS, vol. 4986, pp. 244–266. Springer, Heidelberg (2008)
17. Dimitriou, T.: A lightweight RFID protocol to protect against traceability and cloning attacks. In: SECURECOMM 2005. IEEE Computer Society, Los Alamitos (2005)
18. ECRYPT. The eSTREAM Project (2008), `http://www.ecrypt.eu.org/stream/`
19. Electronic Product Code Global Inc., `http://www.epcglobalinc.com`
20. Feldhofer, M., Rechberger, C.: A case against currently used hash functions in RFID protocols. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006. LNCS, vol. 4275. Springer, Heidelberg (2006)
21. Gilbert, H., Robshaw, M., Sibert, H.: An active attack against HB^+—a provably secure lightweight authentication protocol. IEE Electronic Letters 41, 1169–1170; See also Cryptology ePrint Archive, Report 2005/237, `http://eprint.iacr.org`
22. Gilbert, H., Robshaw, M., Seurin, Y.: Good variants of HB^+ are hard to find. In: Tsudik, G. (ed.) FC 2008. LNCS, vol. 5143, pp. 156–170. Springer, Heidelberg (2008)
23. Gilbert, H., Robshaw, M., Seurin, Y.: $HB^{\#}$: Increasing the Security and Efficiency of HB. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 361–378. Springer, Heidelberg (2008)
24. Good, T., Benaissa, M.: Asic hardware performance. In: Robshaw, M.J.B., Billet, O. (eds.) New Stream Cipher Designs. LNCS, vol. 4986, pp. 267–293. Springer, Heidelberg (2008)
25. Hell, M., Johansson, T., Meier, W.: Grain—A Stream Cipher for Constrained Environments. In: Robshaw, M., Billet, O. (eds.) New Stream Cipher Designs: The eSTREAM Finalists. LNCS, vol. 4986, pp. 179–190. Springer, Heidelberg (2008)
26. Hellman, M.: A Cryptanalytic Time-Memory Trade-Off. IEEE Transactions on Information Theory 26(4), 401–406 (1980)
27. Hennig, J.E., Ladkin, P.B., Sieker, B.: Privacy Enhancing Technology Concepts for RFID Technology Scrutinised. RVS-RR-04-02, Univ. of Bielefeld (2004)
28. Henrici, D., Muller, P.: Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers. In: Pervasive Computing and Communications Workshops (2004)

29. International Organisation for Standardisation, http://www.iso.org
30. Juels, A.: Minimalist Cryptography for Low-Cost RFID Tags. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 149–164. Springer, Heidelberg (2005)
31. Juels, A., Pappu, R.: Squealing Euros: Privacy Protection in RFID-Enabled Banknotes. In: Wright, R.N. (ed.) FC 2003. LNCS, vol. 2742, pp. 103–121. Springer, Heidelberg (2003)
32. Juels, A., Rivest, R., Szydlo, M.: The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy. In: Atluri, V. (ed.) ACM CCS (2003)
33. Juels, A., Weis, S.: Defining strong privacy for RFID. ePrint, Report 2006/137
34. Juels, A., Weis, S.A.: Authenticating Pervasive Devices With Human Protocols. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 293–308. Springer, Heidelberg (2005)
35. Leander, G., Paar, C., Poschmann, A., Schramm, K.: A Family of Lightweight Block Ciphers Based on DES Suited for RFID Applications. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 196–210. Springer, Heidelberg (2007)
36. Lee, J., Yeom, Y.: Efficient RFID Authentication Protocols Based on Pseudorandom Sequence Generators. Cryptology ePrint Archive, Report 2008/343
37. Molnar, D., Wagner, D.: Privacy and security in library RFID: Issues, practices, and architectures. In: Pfitzmann, B., Liu, P. (eds.) ACM CCS 2004, pp. 210–219 (2004)
38. Oechslin, P.: Making a faster cryptanalytic time-memory trade-off. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 617–630. Springer, Heidelberg (2003)
39. Ohkubo, M., Suzuki, K., Kinoshita, S.: Cryptographic Approach to "Privacy-Friendly" Tags. In: RFID Privacy Workshop (2003)
40. Ohkubo, M., Suzuki, K., Kinoshita, S.: Efficient hash-chain based RFID privacy protection scheme. In: Ubiquitous Computing—Privacy Workshop (2004)
41. Ouafi, K., Overbeck, R., Vaudenay, S.: On the Security of HB# against a Man-in-the-Middle Attack. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 108–124. Springer, Heidelberg (2008)
42. Ouafi, K., Vaudenay, S.: Smashing SQUASH-0. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 300–312. Springer, Heidelberg (2009)
43. Paise, R.-I., Vaudenay, S.: Mutual Authentication in RFID: security and privacy. In: Abe, M., Gligor, V.D. (eds.) ASIACCS 2008, pp. 292–299. ACM, New York (2008)
44. Robshaw, M., Billet, O. (eds.): New Stream Cipher Designs: The eSTREAM Finalists. LNCS, vol. 4986. Springer, Heidelberg (2008)
45. Stop RFID, http://www.stoprfid.de/en/
46. Sarma, S., Weis, S., Engels, D.: RFID Systems and Security and Privacy Implications. In: Kaliski, B., Koç, C., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 454–469. Springer, Heidelberg (2002)
47. Shamir, A.: SQUASH—a New MAC With Provable Security Properties for Highly Constrained Devices Such As RFID Tags. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 144–157. Springer, Heidelberg (2008)
48. van Le, T., Burmester, M., de Medeiros, B.: Universally composable and forward-secure RFID authentication and authenticated key exchange. In: Bao, F., Miller, S. (eds.) ASIACCS 2007, pp. 242–252. ACM press, New York (2007)
49. Vaudenay, S.: On privacy models for RFID. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 68–87. Springer, Heidelberg (2007)

50. Weis, S., Sarma, S., Rivest, R., Engels, D.: Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In: Hutter, D., Müller, G., Stephan, W., Ullmann, M. (eds.) SPC 2003. LNCS. Springer, Heidelberg (2003)
51. Wolkerstorfer, J., Dominikus, S., Feldhofer, M.: Strong authentication for RFID systems using the AES algorithm. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 357–370. Springer, Heidelberg (2004)

A Proof of Theorem 1

We show that, if for a given $\lambda \geq 1$ there exists a multiple oracle distinguisher A for $\{f_K\}$ using λ oracles and asking at most q queries to each, of computing time lower than $T' = T - \lambda T_{PRF}$, and of advantage of at least ϵ, then there is a single oracle distinguisher B able to distinguish $\{f_K\}$ with an advantage of at least $\frac{\epsilon}{\lambda}$, asking at most q queries and of computing time lower than $T' + \lambda T_{PRF}$. We use a classical proof technique relying on an hybrid argument. For $0 \leq i \leq \lambda$, $K_1, \ldots, K_i$ denote randomly chosen values of $\{0,1\}^k$ at random if $i \geq 1$ and the empty list if $i = 0$, $f_1, \ldots, f_{\lambda - i}$ random functions if $i \leq \lambda - 1$ and the empty list if $i = \lambda$. Let x_j^i $(1 \leq j \leq q)$ denote the challenges of $\{0,1\}^n$ chosen by A (the challenges (x_j^i) with $1 \leq j \leq q$ are given to the i$^{\text{th}}$ oracle). Let X_i be the following λm-bit random vector: $\big((f_{K_1}(x_j^1))_{j=1,\ldots,q}, \ \ldots, \ (f_{K_i}(x_j^i))_{j=1,\ldots,q}, (f_1(x_j^{i+1}))_{j=1,\ldots,q}, \ \ldots, \ (f_{\lambda - i}(x_j^\lambda))_{j=1,\ldots,q} \big)$.

We use the conventions that $(f_l(x_j^l))_{j=1,\ldots,q;l=1,\ldots,\lambda-i}$ represents the empty string for $i = \lambda$ and $(f_{K_l}(x_j^l))_{j=1,\ldots,q;l=1,\ldots,i}$ represents the empty string for $i = 0$. We see that X_0 is the random vector obtained by A when the oracles are random functions, X_λ is the vector obtained by A when the oracles are chosen from $\{f_K\}$, and the X_i are intermediate between X_0 and X_λ. Let p_i denote the probability that A accepts while receiving a vector X_i to his challenges. The hypothesis about algorithm A is $|p_0 - p_\lambda| \geq \epsilon$.

Algorithm B works as follows: given an oracle f, it randomly selects an integer i_0 such that $1 \leq i_0 \leq \lambda$ and $i_0 - 1$ random values $K_1, \ldots, K_{i_0 - 1}$. When receiving the challenges (x_j^i) from A with $1 \leq i \leq i_0 - 1$ he returns to A the values $f_{K_i}(x_j^i)$. When receiving the challenges $(x_j^{i_0})$ from A, B sends them to its oracle f and forwards the answers $f(x_j^{i_0})$ to A. When receiving the challenges (x_j^i) from A with $i_0 + 1 \leq i \leq \lambda$ he chooses $q(\lambda - i_0)$ random values r_l^k (with $\lambda - i_0 \leq k \leq \lambda$ and $1 \leq l \leq q$) such that if $x_{o_1}^k = x_{o_2}^k$ then $r_{o_1}^k = r_{o_2}^k$. These random values are used to simulate $\lambda - i_0$ random functions $f_1, \ldots, f_{\lambda - i_0}$ to A. To summarize, B constructs and sends to A the λm-bit vector Y_i defined as: $\big(f_{K_1}(x_1^1), \ldots, f_{K_1}(x_q^1), \ldots, f_{K_{i-1}}(x_q^{i-1}), f(x_1^i), \ldots, f(x_q^i), f_1(x_1^{i+1}), \ldots, f_{\lambda - i}(x_q^\lambda)\big)$. If the oracle given to B is a perfect random function then the vector Y_i is distributed in the same way as X_{i-1}. On the other hand, if the oracle given to B belongs to $\{f_K\}$ then the vector is is distributed in the same way as X_i.

To distinguish $\{f_K\}$ from a perfect random function generator, B calls A with input Y_i and outputs A's output. Since $\big|\Pr_f\big[B(f) = 1\big] - \Pr_{f_K}\big[B(f_K) = 1\big]\big|$ can be written as

$$\left| \frac{1}{\lambda} \sum_{i=1}^{\lambda} p_{i-1} - \frac{1}{\lambda} \sum_{i=1}^{\lambda} p_i \right| = \frac{1}{\lambda} |p_0 - p_\lambda| \geq \frac{\epsilon}{\lambda},$$

B distinguishes $\{f_K\}$ from a perfect random function generator with probability at least $\frac{\epsilon}{\lambda}$ in time at most $T' + \lambda T_{PRF}$.

B Proof of Lemma 1

To upper bound the advantage of a $(q, T - qT_G)$-adversary A, we consider the intermediate situation where the oracle function is (f^*, g_K) and f^* is a random function. The triangular inequality gives: $\mathrm{Adv}^{PRF}_{F \times G}(A) \leq |\Pr(A^{f_K, g_K} = 1) - \Pr(A^{f^*, g_K} = 1)| + |\Pr(A^{f^*, g_K} = 1) - \Pr(A^{f^*, g^*} = 1)|$. We bound each term by expressing it as the advantage of a distinguisher against F or G. For the first term, we consider a single oracle distinguisher B against F constructed as follows: first it chooses a random K, then having access to an oracle f it answers the challenges x_i of A with $(f(x_i), g_K(x_i))$. Clearly $\Pr(B^{f^*} = 1) = \Pr(A^{f^*, g_K} = 1)$ and $\Pr(B^{f_K} = 1) = \Pr(A^{f_K, g_K} = 1)$ and as B works in the same time as A plus q computations of g_K, the first term is bounded by ϵ_1. For the second term, we consider a single oracle distinguisher C against G constructed as follows: having access to an oracle g it answers the queries x_i of A with $(y_i, g(x_i))$ where y_i are random values simulating a random function and so $x_i = x_j \Rightarrow y_i = y_j$. Clearly $\Pr(B^{g^*} = 1) = \Pr(A^{f^*, g^*} = 1)$ and $\Pr(B^{g_K} = 1) = \Pr(A^{f^*, g_K} = 1)$ and as B runs in the same time as A, the second term is bounded by ϵ_2.

C Proof of Theorem 2

We prove by induction. The case $t = 1$ is trivial. To establish the property at rank t we consider the intermediate situation where the oracle function is $(a_1, \ldots, a_{t+1}) \to (f_1^*(a_1), \ldots, f_t^*(a_1, \ldots, a_t), f^1_{g_t^*(a_1, \ldots, a_t)}(a_{t+1}), f^2_{g_t^*(a_1, \ldots, a_t)}(a_{t+1}))$ where the f_i^* and g_t^* are independent random functions (see Figure 3). Let A

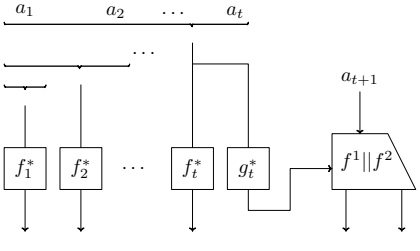

Fig. 3. Intermediate setting

be a single oracle distinguisher against F_{t+1} using q queries and working in time $T' = T - tqT_{PRF}$. Its advantage is upper-bounded by the triangular inequality:

$$\begin{aligned}\mathrm{Adv}^{PRF}_{F_{t+1}}(A) \leq |\Pr(A^{f^1_K(.),f^1_{f^2_K(.)}(.),\ldots,f^1_{f^2\cdots f^2_{f^2_K(.)}(.)\cdots(.)}(.),f^2_{f^2\cdots f^2_{f^2_K(.)}(.)\cdots(.)}(.)} = 1)\\ -\Pr(A^{f^*_1(),f^*_2(),\ldots,f^*_t(),f^1_{g^*_t(.,\ldots,.)}(.),f^2_{g^*_t(.,\ldots,.)}(.)} = 1)|\\ +|\Pr(A^{f^*_1(),f^*_2(),\ldots,f^*_t(),f^1_{g^*_t(.,\ldots,.)}(.),f^2_{g^*_t(.,\ldots,.)}(.)} = 1) - \Pr(A^{(f^*_1(),\ldots,f^*_t(),f^*_{t+1}(),g^*_{t+1}())} = 1)|\end{aligned}$$

To bound the first term by $((t-1)q+1)\epsilon$ we notice that it is the advantage of the single oracle distinguisher B against F_t constructed as follows. B's oracle is a $(t+1)$-tuple of functions denoted by $g = (g^1, \ldots, g^{t+1})$ (where g^i for $1 \leq i \leq t$ has i arguments and g^{t+1} has t arguments). B launches A and answers any oracle query $(a_1, \ldots, a_{t+1})$ of A by returning the value

$$(g^1(a_1), \ldots, g^t(a_1, \ldots, a_t), f^1_{g^{t+1}(a_1,\ldots,a_t)}(a_{t+1}), f^2_{g^{t+1}(a_1,\ldots,a_t)}(a_{t+1})).$$

Finally B outputs A's output. Clearly B works in time $T' + qT_{PRF}$ and its advantage is exactly equal to the first term. The induction hypothesis on the indistinguishability of F_t provides the claimed upper bound.

To bound the second term by $q\epsilon$ we show that it is the advantage of the multiple oracle distinguisher C against F build as follows. Each of the $\lambda = q$ oracles queried by C is a pair $h_l = (h^1_l, h^2_l)$ of single argument functions. C launches A and answers any of the q queries $((x^j_i)_{1\leq j\leq t}, y_i)$ of A as follows: first it chooses an $I \in \{1, \ldots, \lambda\}$ as a random function of $(x^j_i)_{1\leq j\leq t}$, then chooses a qt-tuple $(r^j_i)_{(1\leq i\leq q, 1\leq j\leq t)}$ of values so that for all $1 \leq i \leq q, 1 \leq j \leq t$, r^j_i is a random function of $(x^{j'}_i)_{1\leq j'\leq j}$, and finally returns to A the value $((r^j_i)_{1\leq j\leq n}, h^1_I(y_i), h^2_I(y_i))$ and outputs A's output. C works in the same time as A.

When C's oracles are chosen into F as it is equivalent to choose a key among q random keys $K_1, \ldots, K_q$ by selecting a random function of $(a_1, \ldots, a_t)$ as index in $\{1, \ldots, q\}$ and to choose as a key $g^*(a_1, \ldots, a_t) \in \{0,1\}^k$ where g^* is a perfect random function we have

$$\Pr(C^{(f^1_{K_i}, f^2_{K_i})_{i=1,\ldots,q}} = 1) = \Pr(A^{f^*_1(),f^*_2(),\ldots,f^*_t(),f^1_{f^*_{t+1}(.,\ldots,.)}(.),f^2_{f^*_{t+1}(.,\ldots,.)}(.)} = 1).$$

When C's oracles are perfect random functions, as it is equivalent (as long as at most q distinct t tuples $(a_1, \ldots, a_t)$ are considered) to choose a perfect random function of $(a_1, \ldots, a_{t+1})$ and to choose a perfect random function of a_{t+1} among q parametrized by a perfect random function of $(a_1, \ldots, a_t)$, we have

$$\Pr(C^{(h^*_i)_{i=1,\ldots,q}} = 1) = \Pr(A^{(f^*_1(),\ldots,f^*_t(),f^*_{t+1}(),g^*_{t+1}())} = 1).$$

The inequality $\mathrm{Adv}^{PRF}_F(C) \leq \mathrm{Adv}^{PRF}_F(q, q, T - tqT_{PRF}) \leq q\mathrm{Adv}^{PRF}_F(q, T - (t-1)qT_F) \leq q\mathrm{Adv}^{PRF}_F(q, T)$ of Theorem 1 gives the upper bound of the second term and concludes the proof.