Open access • Journal Article • DOI:10.1007/S00779-011-0392-2

Extending ECC-based RFID authentication protocols to privacy-preserving multi-party grouping proofs — Source link

Lejla Batina, Yong Ki Lee, Stefaan Seys, Dave Singelée ...+1 more authors

Institutions: Radboud University Nijmegen, Samsung, Katholieke Universiteit Leuven

Published on: 01 Mar 2012 - Ubiquitous Computing

Topics: Authentication, Elliptic curve cryptography and Verifiable secret sharing

Personal and Ubiquitous Computing manuscript No.
(will be inserted by the editor)

Extending ECC-Based RFID Authentication Protocols to Privacy-Preserving Multi-Party Grouping Proofs

Lejla Batina · Yong Ki Lee · Stefaan Seys · Dave Singelée · Ingrid Verbauwhede

Received: October 25, 2010 / Accepted: date

Abstract Since the introduction of the concept of grouping proofs by Juels, which permit RFID tags to generate evidence that they have been scanned simultaneously, various new schemes have been proposed. Their common property is the use of symmetric-key primitives. However, it has been shown that such schemes often entail scalability, security and/or privacy problems. In this article, we extend the notion of public-key RFID authentication protocols, and propose a privacy-preserving multi-party grouping-proof protocol which relies exclusively on the use of Elliptic Curve Cryptography (ECC). It allows to generate a proof which is verifiable by a trusted verifier in an offline setting, even when readers or tags are potentially untrusted, and it is privacy-preserving in the setting of a narrow-strong attacker. We also demonstrate that our RFID grouping-proof protocol can easily be extended to use cases with more than two tags, without any additional cost for an RFID tag. To illustrate the implementation feasibility of our proposed solutions, we present a novel ECC hardware architecture designed for RFID.

Keywords RFID · Authentication · Grouping Proofs · ECC · Privacy

Lejla Batina
CS Department/Digital Security group
Radboud University Nijmegen, The Netherlands
Tel.: +31-24-3652217
E-mail: lejla@cs.ru.nl

Yong Ki Lee
Samsung Electronics Research and Development, South-Korea
E-mail: yklee93@kg21.net

Lejla Batina · Stefaan Seys · Dave Singelée · Ingrid Verbauwhede
Department of Electrical Engineering/SCD-COSIC & IBBT
University of Leuven, Belgium
E-mail: firstname.lastname@esat.kuleuven.be

1 Introduction

Radio Frequency Identification systems are rapidly expanding their applications to many areas: inventory systems, supply chains, access control, vehicle tracking, toll payments, e-ticketing, pharmaceutics, *etc*. The advantage of RFID over bar-code technology is that it does not require direct line-of-sight reading and that tags can be interrogated at greater distances. The technology also enables the automation of some control processes, which results in a significant gain in terms of time and cost.

However, due to wide spread of tags and its cheap implementations, these applications risk the security and privacy of a tag carrier. The need for privacy-preserving RFID protocols is evident even in these extremely resourceless environments. In particular, RFID tags have severe limitations with respect to area, power and energy. Nevertheless, it is generally assumed that these devices are able to perform basic cryptographic operations. Even more, nowadays even public-key protocols found their way in RFID applications [10, 17, 15].

In most RFID systems, there is a clear demand for identification of the tag and/or the reader. Recently, Juels [11] extended this notion and envisioned the concept of grouping proofs (also denoted by yoking proofs), which allows two or more tags to provide evidence that they were scanned simultaneously by a reader within its broadcast range. There are various practical scenarios where there is an explicit need for such protocols [6, 11, 24]. For example, there could be a legal requirement that certain medication should be distributed together with a brochure describing its side-effects. A technical solution to this problem is to attach RFID tags to both the medication and the brochures, and create grouping proofs when they are scanned simultaneously. The pharmacist then stores these grouping proofs as evidence, to transmit them to the government for verification. Another practical scenario is aircraft security, when a certain piece of a plane can only leave a factory accompanied by a security cap. By using grouping proofs, one can also couple a physical person via his passport to his boarding pass, or – in the military context – only enabling weaponry or equipment when an appropriate group of entities is present. Other use cases include governmental administration, to check that a specific form is enclosed with its corresponding stamp, or scenarios when one wants to generate evidence that a group of people were present at a particular location.

Various grouping proofs have been proposed in the literature. Unlike these schemes, our solutions are based on public-key cryptography and in particular on Elliptic Curve Cryptography (ECC) [14, 20]. This feature makes the protocols lightweight enough to be considered suitable even for passive tags, as will be shown later in the article.

An abridged, short version of this paper has been published in [2]. Our contributions can be summarized as follows:

- We extend the ID-transfer protocol of Lee *et al.* [15] to a secure ECC-based privacy-preserving two-party grouping-proof protocol.
- We prove the concept is easily extendable to the case with more than two tags, without any additional cost for the tags.
- We show how the complexity can be reduced in scenarios where the colluding tags attack is not relevant.

– We describe a novel architecture that is optimized on performance, showing that our solutions are suitable for RFID technology.

The remainder of this article is organized as follows. In Sect. 2 we list previous related work in the area of grouping proofs and show that all these schemes are based on symmetric-key primitives. In contrast, our solution relies exclusively on the use of ECC and extends the notion of public-key-based RFID authentication protocols, which are briefly discussed in Sect. 3. Next, in Sect. 4 we describe our assumptions and adversarial model. Our ECC-based multi-party grouping-proof protocols are given in Sect. 5 and Sect. 6. A novel architecture for an ECC processor suitable for RFID is outlined in Sect. 7. We also give performance figures for the new protocols proving that our solutions are feasible for passive RFID tags. We conclude our work in Sect. 8.

2 Related work

The idea of grouping proofs originates from Juels [11]. The motivation comes from any application that requires the combined presence of two (or more entities). His proposal for this type of identification protocol, so-called yoking proof, relies on interleaving MACs of two tags using a reading device as a communication medium. The grouping proof generated in Juels' yoking-proof protocol, is verifiable by a trusted party, even when readers are potentially untrusted.

Saito and Sakurai [26] were the first to point out weaknesses in the work of Juels. They showed that the minimalist version of Juels is subject to replay attacks and they proposed a new solution using time stamps. They also generalized the concept for a group of tags and introduced the corresponding grouping proofs. However, Piramuthu [25] demonstrated that this new proof is also vulnerable to replay attacks and he proposed a modification. His ideas include adding another random variable sent from the verifier to the tags (through the reader) and the assumption that no proof should be generated without including the secret values (obtained by a one-way function) from all the tags. Bolotnyy and Robins [3] proposed a new solution for the grouping proofs and addressed the requirements on privacy. The new protocol is called anonymous yoking and each tag is supposed to compute a keyed hash function and a MAC. The main drawback of the scheme is the computational complexity on the side of the verifier being $O(n^2)$. Peris-Lopez *et al.* [23] proposed an improvement, so-called clumping proofs, that are privacy-preserving and the verification takes $O(n)$ steps. Burmester *et al.* present a security model based on the Universal Composability framework for this so-called group-scanning problem [6]. The requirements considered include privacy and forward security. As a result three grouping proofs are proposed that require only pseudo-random functions. Starting from the first one without anonymity each protocol adds a new property to the previous one, *i.e.*, anonymity and forward secrecy. Lien *et al.* [19] proposed an order-independent protocol, which should improve the efficiency and reduce the failure rates. The reason for improved efficiency is the fact that there is no requirement on predefined reading order. Finally, Leng et al. [18] proposed a variant of the grouping protocol that is actively choosing

the tag to be verified. Their solution is rather similar to the concept of an RFID search protocol.

The common property for all the schemes proposed so far is the use of symmetric-key primitives: *e.g.*, hash functions, MACs, pseudo-random functions *etc*. Such schemes are however often not scalable, and entail several security (*e.g.*, cloning attacks) and/or privacy problems (*e.g.*, it is proven that one needs public-key cryptography to achieve a certain level of privacy protection [30]). In contrast to this, we propose to rely exclusively on the use of public-key cryptography. More in particular, we show how to extend the ECC-based ID-transfer protocol proposed by Lee *et al.* [15] to a privacy-preserving multi-party grouping-proof protocol.

3 Public-key-based RFID authentication protocols

Most attempts to design RFID authentication protocols rely on the use of symmetric-key cryptography. Of the many notable designs, we mention here the HB^+ protocol of Juels and Weis [12], which was one of the first solutions proposed in the literature. The main reason why most RFID authentication protocols use symmetric-key primitives, lies in the common perception of public-key cryptography being too slow, power-hungry and too complicated for such low-cost environments.

However, recent works proved this concept to be wrong, as for example the smallest published ECC implementations [17, 10] consume less area than the candidate cryptographic hash algorithms proposed in the SHA-3 competition [29]). This has led to the introduction of public-key based RFID authentication protocols. This approach solves the scalability issues that often burden symmetric-key solutions, prevents cloning attacks and offers advanced privacy protection.

Lee *et al.* [16] proposed the EC-RAC (Elliptic Curve Based Randomized Access Control) protocol, based on the conventional public-key based authentication protocol of Schnorr [27]. However, in [5, 8], it is shown that EC-RAC is vulnerable to tracking attacks and replay attacks, and in addition [5], the randomized Schnorr protocol has been proposed. Later, the EC-RAC protocol has been gradually revised to counter the the known attacks. This resulted in the ID-transfer protocol [15], which is resistant against active impersonation attacks, and is narrow-strong privacy-preserving. This protocol will be the basic building block in the construction of our grouping proofs.

3.1 Notation

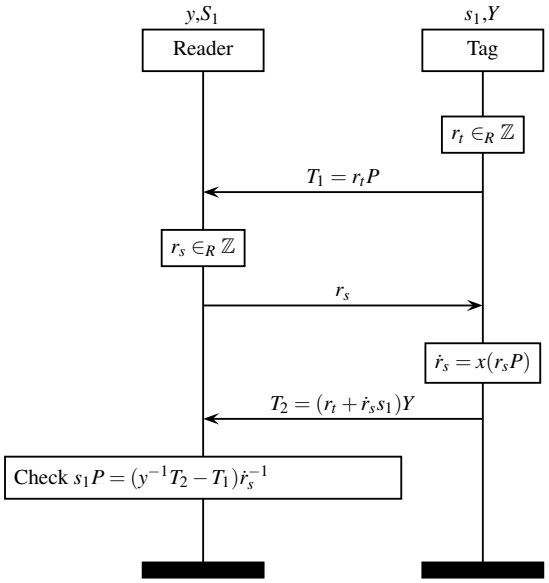
Let us first introduce the notation used in this work. We denote P as the base point on a Elliptic Curve, and y and $Y(=yP)$ are the trusted verifier's private-key and public-key pair, where yP denotes the point derived by the point multiplication operation on the Elliptic Curve group. We use the notation $x(T)$ to denote the x-coordinate of the point T on the elliptic curve, and $\dot{r}_s$ to denote the non-linear mapping $x(r_sP)$, with P the base point of the elliptic curve. The values s_t and $S_t(=s_tP)$ are tag t's private-key and public-key. One should note, although the name suggests that it can be publicly

known, that a tag should not reveal its public-key during the execution of the protocol, as this would cause tracking attacks.

3.2 ID-transfer protocol

The ID-transfer protocol [15] is shown in Fig. 1. A tag first generates a random number r_t, and computes and transmits the corresponding message T_1 to the reader. After receiving a challenge r_s from the reader, a tag first checks that it is not equal to zero or the order of the point P and then computes the response T_2 using its private-key s_1, the random number r_t, and the non-linear mapping $\dot{r}_s$ of the challenge r_s. The EC point multiplication in this operation acts as a one-way function. The response T_2 is sent to the reader. Then, the reader derives the tag's public-key $S_1 (= s_1 P)$ and checks if it is registered in the database. Note that only the reader is capable of performing this verification, as this operation requires knowledge of the private-key y.

Fig. 1 ID-transfer protocol of Lee *et al.* [15].



4 Assumptions and adversarial model

The aim of constructing a grouping proof is to enable a set of RFID tags to generate a proof that they have been scanned simultaneously by a reading device. In this setting, there are three distinct parties involved: the set of tags, the reader, and a trusted verifier. The former two will engage in a protocol run which results in the construction of the grouping proof. This proof is then verifiable by the trusted verifier in an offline setting. The verifier hence, does not need to be involved directly during the execution of the protocol.

Due to the "simultaneously scanned" requirement, the notion of time is very important, as already pointed out by Juels [11]. The correctness of this claim relies on a timeout assumption. We assume that the reader measures the round-trip-time, *i.e.* the time between sending a message to a tag and receiving the response, during the execution of the protocol. If this round-trip-time exceeds a particular threshold, the reader aborts and the protocol fails. Using a very tight threshold limits the power of an adversary, but also increases the false rejection ratio. Besides the reader, also the RFID tags are assumed to have a timeout mechanism. However, this timeout does not need to be very precise and/or small (*e.g.*, it can be in the order of seconds or even larger). Due to this timeout assumption, the protocol will always terminate. The accuracy of the grouping proofs with respect to timing, depends on the precision of the timeouts.

We assume that the verifier is trusted by all devices in the system. Furthermore, we assume that the public-key Y of the verifier is a system parameter, known by all the devices that are involved in the construction of a grouping proof. Only the verifier knows the corresponding private-key y. Knowledge of y is a necessary requirement to check the correctness of a grouping proof. The result of a verification claim is failure (if the grouping proof was not correct), or it gives the identities of the tags that were scanned simultaneously. In this case the verifier stores the grouping proof and adds a timestamp to it. This enables temporal ordering of the grouping proofs.

The task of the reader is to coordinate the execution of the protocol with the set of tags (*i.e.*, query the tags), collect the grouping proof and forward it to the verifier. The reader does not have to check the correctness of the responses, and is not necessarily trusted by the tags and/or the verifier.

Besides operational and computational requirements, both security and privacy are important when employing the concept of grouping proofs in an RFID setting. From a security point of view, the grouping proofs must be verifiable even if the RFID tags were scanned by an adversarial reader or if the tags were compromised by the adversary. Without loss of generality, let us now assume that there are only two tags that are scanned simultaneously. To avoid an adversary impersonating tags that were not present during the execution of the protocol and/or constructing fake grouping proofs (i.e. that not reflect the correct situation), one needs to prevent the following five potential attack scenarios:

Compromised tag: One of the tags involved in the protocol, is compromised by the adversary. The reader is assumed to be non-compromised.

Man-in-the-middle attack: In this attack scenario, both tags are assumed to be non-compromised, but the reader is compromised by the adversary.

Colluding reader and tag: Both the reader and one of the tags are compromised by the adversary, the other tag is assumed to be non-compromised.

Colluding tags: In this attack scenario, both tags are compromised by the adversary. They can exchange some messages in advance (*e.g.*, via another reader), but do not know each other's private key (*e.g.*, it cannot be extracted from the tag). The reader is assumed to be non-compromised.

Replay attack performed by eavesdropper: In this attack scenario, an eavesdropping outsider scans two non-compromised tags simultaneously. At a later time, it replays the copied message-flow in the presence of a non-compromised reader to impersonate the two tags, with none of these tags being actually present.

	tag *A*	tag *B*	Reader	Eavesdropper
Compromised tag	X			
Man-in-the-middle			X	
Colluding reader and tag	X		X	
Colluding tags	X	X		
Replay attack by eavesdropper				X

Table 1 Five attack scenarios

The attack scenarios are summarized in table 1. For each of the scenarios, an *X* denotes that the corresponding entity is compromised in that particular attack scenario. Note that we do not consider the scenario where both tags and the reader are compromised by the adversary, as this would allow compromised tags to construct a valid grouping proof, even if they were not scanned simultaneously. We also do not consider the attack where a compromised reader scans two non-compromised tags, and forwards the grouping proof at a later time to the verifier (*i.e.* to have an incorrect timestamp being added to the grouping proof). To avoid this, the verifier needs to actively participate in the protocol, or one needs to incorporate the exact time in the protocol in a verifiable way (e.g., use a challenge that depends on the time, in such a way that the verifier can check this). Note that if only non-compromised readers can communicate directly to the verifier, this attack is automatically prevented. Also note that we only consider protocols on the logical level. Danev *et al.* [7] have shown that one can also identify RFID tags based on their physical-layer fingerprints. This is however outside the scope of this article.

In the design of our protocol, we also want to achieve *untraceability*, in which the (in)equality of two tags must be impossible to determine. Only the trusted verifier should be able to check which particular tags were scanned simultaneously. To evaluate the privacy of RFID systems, several theoretical models have been proposed in the literature [1,13,22,30]. We particularly focus on two characteristics of attackers from the theoretical framework of Vaudenay [30]: *wide* (or *narrow*) attackers and *strong* (or *weak*) attackers. If an attacker has access to the result of the verification of the grouping proof (accept or reject) in the verifier, he is a *wide* attacker. Otherwise he is a *narrow* attacker. If an attacker is able to extract a tag's secret and reuse it to construct a grouping proof, he is a *strong* attacker. Otherwise he is a *weak* attacker. Vaudenay demonstrated that one needs to to employ public-key cryptography to achieve strong

privacy requirements [30]. Because of this observation, we will rely on public-key cryptography to construct a narrow-strong privacy-preserving grouping-proof protocol. For efficiency reasons, we will particularly use ECC.

Note that the grouping proofs that are proposed in this article, do not prove that the tags are located in physical proximity to one another. An adversary can use multiple readers, and forward messages between these devices, to simultaneously scan tags at remote locations. Besides the large effort and cost, the effect of this attack is limited due to the timeout mechanism. It can be completely prevented by employing RFID distance bounding protocols (*e.g.*, [4,9]). This is however outside the scope of this article.

5 ECC-based grouping-proof protocol with colluding tag prevention

Starting from the ID-transfer protocol, which we briefly introduced in Sect. 3.2, we can construct a privacy-preserving ECC-based grouping-proof protocol which prevents all the five attack scenarios discussed in Sect. 4. The main idea is to intermingle runs of the ID-transfer protocol with multiple tags into a single grouping-proof protocol, which we will denote as the *CTP* protocol ("colluding tag prevention").

5.1 Protocol description

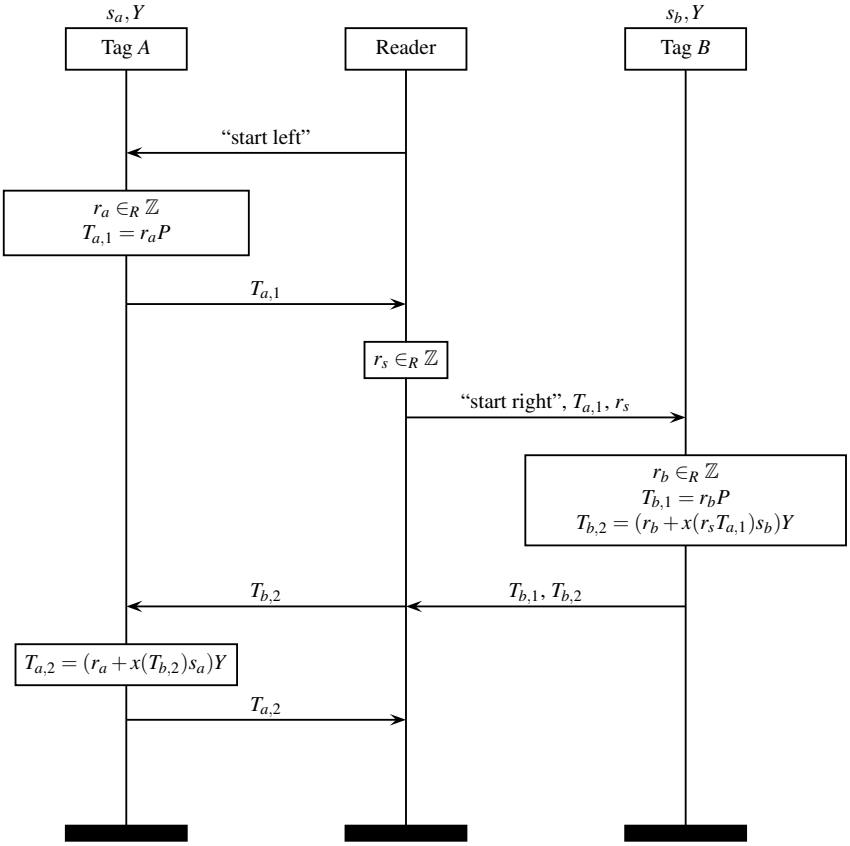
The two-party CTP protocol, which allows a pair of RFID tags (denoted by tag A and B) to prove that they have been scanned simultaneously, is shown in Fig. 2. During the entire execution of the protocol, the tags and/or the reader abort when a timeout occurs, or when they receive the EC point at infinity. The protocol works as follows. The reader first sends the messages *"start left"* and *"start right"* to indicate the role of the tags in the protocol. Next, tag A generates a random number r_a and the corresponding EC point $T_{a,1}$. This message is then forwarded to tag B. Upon reception, B will first generate a random number r_b and compute the corresponding message $T_{b,1}$. Next, it also computes the response $T_{b,2}$ using its private-key s_b, the random number r_b, the x-coordinate of the challenge $T_{a,1}$, and a random challenge r_s generated by the reader. Both $T_{b,1}$ and $T_{b,2}$ are then transmitted to the reader. In the next stage of the protocol, the reader forwards $T_{b,2}$ to tag A. This tag will then compute the response $T_{a,2}$ using its private-key s_a, the random number r_a, and the x-coordinate of the challenge $T_{b,2}$. The result is forwarded to the reader. The grouping proof, collected by the reader, consists of the following tuple:

$$(T_{a,1}, T_{a,2}, r_s, T_{b,1}, T_{b,2})$$

To verify the grouping proof constructed by tag A and B, the verifier first checks that the proof was not used before (to detect replay attacks) and then performs the following computations:

$$s_a P = (y^{-1} T_{a,2} - T_{a,1}) x(T_{b,2})^{-1}$$
$$s_b P = (y^{-1} T_{b,2} - T_{b,1}) x(r_s T_{a,1})^{-1}$$

Fig. 2 Two-party grouping-proof protocol with colluding tag prevention (CTP).



If the public keys of A and B (S_a and S_b respectively) are registered in the database of the verifier, the grouping proof is accepted and a timestamp is added.

5.2 Extension to $n > 2$ parties

The two-party CTP grouping-proof protocol shown in Fig. 2 can be easily extended to multiple tags ($n > 2$). The output of each tag is then used as input for the "next" tag in the chain, as shown in Fig. 3. This procedure is repeated until all tags are scanned. The last tag in the chain (denoted by tag Z) sends $T_{z,2}$ to tag A, which then computes its response $T_{a,2}$. The grouping proof consists of the following tuple:

$$(T_{a,1}, T_{a,2}, r_s, \ldots, T_{i,1}, T_{i,2}, \ldots, T_{z,1}, T_{z,2})$$

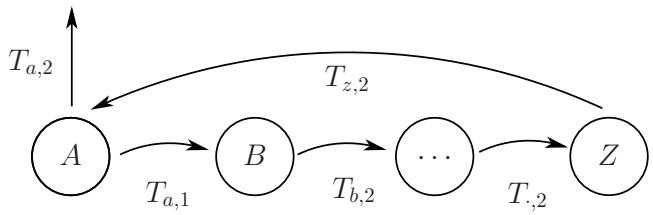


Fig. 3 Chain of grouping proofs.

To check the correctness of the grouping proof, the verifier performs similar operations as with the two-party CTP grouping-proof protocol.

As an example, let us illustrate this with a three-party grouping proof (constructed by tags *A*, *B* and *C*). In this case, the response $T_{b,2}$ is sent to tag *C*. Upon reception, *C* will generate a random number r_c and perform the following two computations:

$$T_{c,1} = r_c P$$
$$T_{c,2} = (r_c + x(T_{b,2})s_c)Y$$

The output $T_{c,1}$ and $T_{c,2}$ is sent to the reader, which forwards it to tag *A*. The latter can then compute the response $T_{a,2} = (r_a + x(T_{c,2})s_a)Y$. To check the correctness of the grouping proof, the verifier needs to perform the following computations:

$$s_a P = (y^{-1}T_{a,2} - T_{a,1})(x(T_{c,2}))^{-1}$$
$$s_b P = (y^{-1}T_{b,2} - T_{b,1})x(r_s T_{a,1})^{-1}$$
$$s_c P = (y^{-1}T_{c,2} - T_{c,1})(x(T_{b,2}))^{-1}$$

5.3 Analysis

5.3.1 Impersonation resistance

The two-party CTP grouping-proof protocol is constructed by entangling two instances of the ID-transfer protocol [15]. Due to this construction, the CTP grouping-proof protocol inherits the security properties of the ID-transfer protocol. The latter is designed to provide secure entity authentication in the setting of an active adversary, and can be shown to be equivalent to the Schnorr protocol [27] regarding impersonation resistance. One can demonstrate that to impersonate a tag in either of our attack scenarios, the adversary needs to know the private-key of that particular tag (or be able to solve the Decisional Diffie-Hellman (DDH) problem [28]).

5.3.2 "Grouping" security

In the context of grouping proofs, it is not sufficient to have impersonation resistance. It is also important to have assurance that all tags took part in the CTP grouping-proof

protocol *at some point in time*. This is realized by the interconnection of the ID-transfer protocol instances. All participating tags use an EC point that was generated by another participating tag in their execution of the ID-transfer protocol. Each of the responses computed by the tags hence depend on the output of another tag in the chain. This entanglement proofs that all tags participated in the protocol.

The use of a timeout mechanism assures that all tags were scanned *simultaneously*, as the reader or a tag aborts the protocol if a timeout would occur.

Finally, the value r_s is randomly generated by the reader for each protocol run it initiates. The response $T_{b,2}$ will depend on this value, and cannot be computed in advance. Due to the interconnection of the ID-transfer protocol instances, the responses of the other participating tags will depend on the response $T_{b,2}$, and consequently also on the random value r_s. This makes it impossible for a set of tags to run the protocol "offline" (i.e., without the presence of the reader), recording all the generated values, and at at later time using the generated values to mislead the reader into believing that they are being scanned while in reality only one of them is present. Similarly, this also prevents an eavesdropping outsider to scan a set of non-compromised tags, and replay the flows at a later time to an authorized reader while non of the scanned tags is present. As already discussed in this article, we do not consider the attack where a reader forwards the grouping proof to the verifier with a large delay.

There is however one particular attack scenario where these mechanisms are not sufficient. When closely observing fig. 2, one notices that tag A both starts and ends the protocol. Because of this, tag A and the reader can collude to generate a valid proof with any set of victim tags. In order to do this, tag A and the reader ignore the timeout mechanism during the run of the protocol. The attack starts by tag A generating $T_{a,1}$ and giving this value to the reader. Next, at a later time, the protocol is carried out with all the victim tags, but without tag A being present. Finally, again at some time later, the reader finishes the protocol with tag A. To prevent this attack, one has to extend the protocol by performing a second round of the protocol in *reverse order*, using the final output of tag A ($T_{a,2}$) as initial input for first tag in the reverse loop. Since this extension increases the computational complexity, we will not discuss it further in the rest of this article. Note that this security problem does not occur when tag B (or any of the other tags) and the reader collude.

5.3.3 Privacy analysis

The same argumentation as above can be used to demonstrate the privacy properties of the CTP grouping-proof protocol. It was a specific design feature of our protocols that only the trusted verifier can check the correctness of the grouping proofs, and hence obtain any information on the tags that have been scanned simultaneously. Since the ID-transfer protocol offers privacy protection against a narrow-strong adversary, untraceability can even be guaranteed if the challenges of the ID-transfer protocol are controlled by the adversary. The responses computed by the tags do not leak any information about the tags' identities to any third party that does not know the private key y. As a direct consequence, one can demonstrate that our CTP grouping-proof protocol inherits the privacy properties of the ID-transfer protocol, and is hence also narrow-strong privacy-preserving.

5.3.4 Cost analysis

The protocol uses the following operations: modular multiplication, addition, and point multiplication on an elliptic curve, of which the latter is the most complex operation and its use hence needs to be minimized.

In our protocol, each tag i has to perform two EC point multiplications to create the output $T_{i,1}$ and $T_{i,2}$. The workload of a tag is independent of the number of tags n involved in the protocol. Another interesting observation is that an n-party grouping proof exactly contains $2n$ EC points. The bitlength of the grouping proof is thus linearly dependent on the number of tags n. Note however that there is a practical upper limit on the number of tags n that can be scanned simultaneously. If n is very large, a timeout could occur in tag A before the protocol has terminated. This should be taken into account when deploying the grouping-proof protocol.

6 Grouping-proof protocol without CTP

It is interesting to note that one can reduce the complexity of the CTP grouping-proof protocol in scenarios where the last two attack scenarios described in Sect. 4 (colluding tags and replay attacks where an eavesdropper impersonates tags which it has scanned before) are not relevant. In this case, one does need to check that the tags are "online" during the run of the protocol and there is hence no longer a need for a fresh random value r_s. Instead of the reader generating the random challenge r_s, one can replace it by the value 1. Note that since this value is fixed, it should no longer be sent to tag B. By performing this modification, the efficiency can be increased. The resulting protocol still prevents the first three attack scenarios described in Sect. 4, and all other security and privacy properties are similar to these of the CTP grouping-proof protocol.

6.1 Protocol description

The reduced two-party grouping-proof protocol without CTP is shown in Fig. 4. It works similar to the CTP grouping-proof protocol described in Sect. 5.1. The result of the protocol is a grouping proof that consists of the following tuple:

$$(T_{a,1}, T_{a,2}, T_{b,1}, T_{b,2})$$

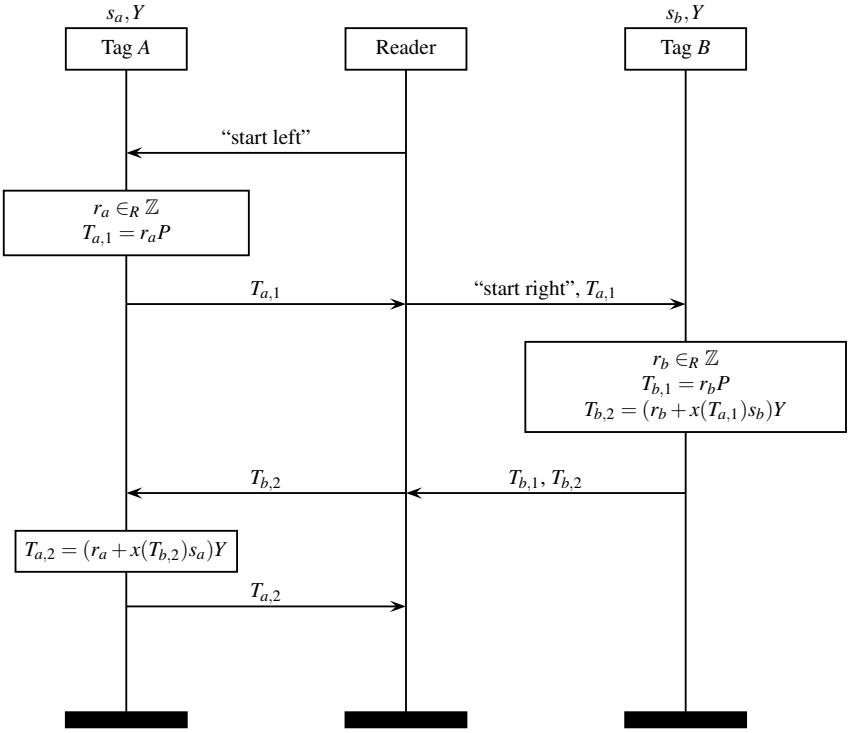
To verify the grouping proof constructed by tag A and B, the verifier first checks if the proof was not already sent before (this would indicate that a replay attack has taken place). Next, it performs the following computations:

$$s_a P = (y^{-1} T_{a,2} - T_{a,1})(x(T_{b,2}))^{-1}$$
$$s_b P = (y^{-1} T_{b,2} - T_{b,1})(x(T_{a,1}))^{-1}$$

If the public keys of A and B (S_a and S_b respectively) are registered in the database of the verifier, the grouping proof is accepted and a timestamp is added.

The two-party grouping-proof protocol without CTP can be easily extended to multiple tags ($n > 2$) by using the same principle as described in Sect. 5.2.

Fig. 4 Two-party grouping-proof protocol without CTP.



6.2 Analysis

6.2.1 Impersonation resistance

As discussed in Sect. 5.3, the CTP grouping-proof protocol is designed to provide secure entity authentication in the setting of an active adversary, and is equivalent to the Schnorr protocol [27] regarding impersonation resistance. These security properties are independent of the choice of the value r_s. By selecting the value 1, the CTP protocol can be transformed to the reduced grouping-proof protocol without CTP, and consequently the latter hence also offers impersonation resistance.

6.2.2 "Grouping" security

It is rather trivial to see that the protocol shown in Fig. 4 does not offer resistance to colluding tags, or to an eavesdropper that impersonates a set of tags which it has

scanned before. By fixing the value r_s to 1, all input needed to compute the tags's responses depends on the tags' data (*i.e.* random numbers, private keys, and challenges generated by other tags), and not on any input from the reader. It is hence impossible to determine if the grouping proof collected by a non-compromised reader is fresh data, or is replayed from an earlier protocol flow.

The grouping-proof protocol without CTP still provides resistance to compromised tags, compromised readers, or a colluding tag and reader. In each of these attack scenarios, it is impossible to impersonate a non-compromised tag, since its private key is needed to compute the response. The impersonation resistance combined with the timeout mechanism hence guarantees the "grouping" security.

6.2.3 Privacy analysis

The privacy properties of the CTP protocol are independent of the choice of the value r_s. The grouping-proof protocol without CTP is hence also narrow-strong privacy-preserving.

6.2.4 Cost analysis

By fixing the value r_s to 1, tag B does not need to compute the EC point multiplication $r_s T_{a,1}$. The number of EC point multiplications is reduced to 2 (compared to 3 in the case of the CTP grouping-proof protocol. All other observations regarding the cost of the protocol remain valid.

7 Implementation

In order to show the feasibility of the proposed protocols for RFID tags, we analyze a hardware implementation of our solutions. The EC processor we present in this article has a novel architecture that features the most compact and at the same time the fastest solution when compared to previous work.

7.1 Overall architecture

The overall architecture is shown in Fig. 5. The processor is composed of a micro controller, a bus manager and an EC processor (ECP). It is connected with a front-end module, a random number generater (RNG), ROM and RAM as shown in the overall architecture (Fig. 5). The solid arrows are for data exchange, the dash arrows are for addressing, and control signals are omitted in this picture. The ROM stores program codes and data. The program is executed by the micro controller and the data may include a tag's private key, the server's public key and system parameters. The program is basically a grouping proof for a tag or an authentication protocol. The micro controller is able to perform general modular arithmetic operations (additions and multiplications) in a byte-serial fashion. It also gives commands for the execution of the ECP via the bus manager. The ECP loads a value k and an EC point P from

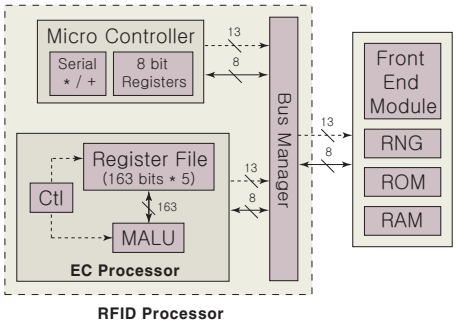


Fig. 5 RFID Processor Architecture.

ROM or RAM and executes the EC scalar multiplication kP. After finishing the scalar multiplication, it stores the results in RAM.

7.2 New design characteristics

The new ECP architecture is similar to the one presented in [17]. Further optimizations are performed in the register file and the Modular ALU (MALU). The EC processor presented in [17] uses a MALU which performs modular addition and multiplications, and it reuses the logic of modular multiplications for modular squaring operations. On the other hand, the new MALU we designed includes a specialized squarer logic. Since the modular squaring can be completed in one cycle on a dedicated squarer, the performance can be substantially increased with an overhead of the square logic. Moreover, in the new architecture the size of register file is reduced to 5×163 bits from 6×163 bits as we are using ECC over $GF(2^{163})$. This reduction is possible since the specialized squarer requires only one operand as input. As a result, the overall circuit area can be reduced even further after including the squarer in the MALU while achieving a much higher performance.

Here we give more details on the new MALU architecture and elaborate on differences with previous works. In this work, instead of designing and combining a squarer with the MALU in a straightforward way, we merge the squarer with the original MALU to minimize the hardware cost.

The squaring formula over the extended binary field of $GF(2^{163})$, where the used irreducible polynomial is $r(x) = x^{163} + x^7 + x^6 + x^3 + 1$, can be derived as follows:

$$A^2 = \left(\sum_{i=0}^{162} a_i x^i \right)^2 = T_1 + T_2 + T_3 \tag{1}$$

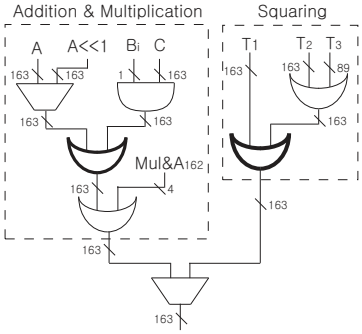


Fig. 6 MALU before integration.

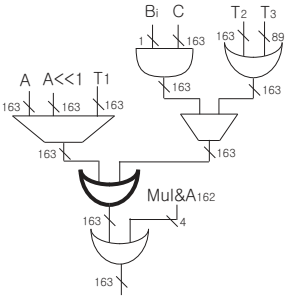


Fig. 7 MALU after integration.

where

$$T_1 = \sum_{i=0}^{81} a_i x^{2i} + \sum_{i=0}^{80} a_{i+82} x^{2i+1},$$

$$T_2 = \sum_{i=0}^{77} a_{i+82}(x^{2i+8} + x^{2i+7}) + a_{160}(x^6 + x^4 + x^3 + x + 1) + a_{161}(x^5 + x^2),$$

$$T_3 = \sum_{i=0}^{79} a_{i+82} x^{2i+4} + a_{160} x^8 + a_{161}(x^{10} + x^8 + x^6 + x^3) + a_{162}(x^{12} + x^{10} + x^5 + x).$$

Note that T_1 and T_2 are whole 163-bit words and T_3 has 89 terms, and that T_1, T_2 and T_3 can be derived by only wiring, without any logic operation. As we perform the operation as $A^2 = (T_2 + T_3) + T_1$, the first addition requires an 89-bit XOR array and the second addition requires a 163-bit XOR array.

Based on the derived equation, we can implement the squarer by implementing two XOR arrays as shown in Fig. 7. Actually, we integrate two bold XOR arrays shown in Fig. 6. As a result of integrating the squarer, a 163-bit XOR array and a two-input 163-bit MUX are replaced with a three-input 163-bit MUX. In Table 2, the new MALU is compared with the previous versions in terms of gate area, whose measurements are performed by synthesizing with a UMC standard CMOS library. The cost for the merged squarer is 558 gates.

Table 2 Comparison of MALUs of the digit size 1.

Operations	Gate Area ($d = 1$)	Comment
+, ×	913 Gates	No squarer
+, ×, ^2	1,636 Gates	With a squarer not sharing XOR array
+, ×, ^2	1,471 Gates	With a squarer sharing XOR array

Although in the proposed MALU, the word size and the irreducible polynomials are fixed, this designing method can be applied to any arbitrary word size and irreducible polynomial.

The MALU operation can be described as the following Eq. (2).

$$\begin{aligned} A(x) &= A^2(x) \bmod P(x) && \text{if } cmd = 2 \\ A(x) &= B(x) \cdot C(x) \bmod P(x) && \text{if } cmd = 1, \\ A(x) &= A(x) + C(x) \bmod P(x) && \text{if } cmd = 0, \end{aligned} \tag{2}$$

where $A(x) = \Sigma a_i x^i$, $B(x) = \Sigma b_i x^i$, $C(x) = \Sigma c_i x^i$ and $P(x) = x^{163} + x^7 + x^6 + x^3 + 1$.

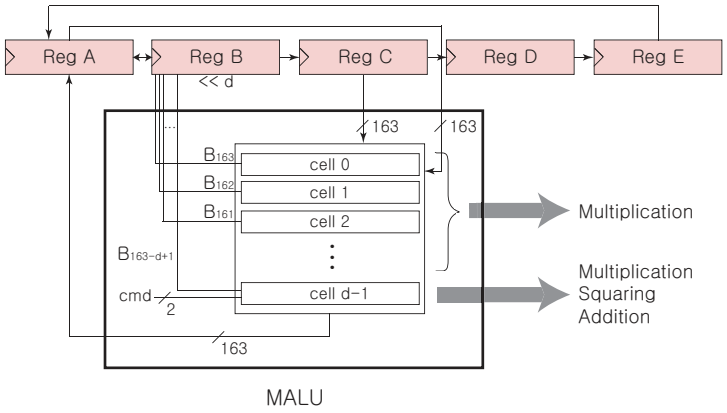


Fig. 8 MALU architecture with register file.

The architecture of MALU with the required registers is shown in Fig. 8. Here the registers in the MALU are combined with the external registers to reduce the number of registers. At the completion of each operation, only register RegA is updated while registers RegB and RegC hold the same data as at the beginning of the operations (we make the shift of d-bits of RegB a circular shift so the value goes back to the original after finishing a multiplication). Therefore, RegB and RegC can be used not only to store field operands but also to store some intermediate values.

7.3 Performance evaluation

The performance comparison is made with the work in [17] for the digit size of 4 in the MALU for both architectures. This work achieves about 24% better performance with a smaller circuit area, and the energy consumption is much smaller. In particular, the size of our ECP processor is estimated to 14,566 kgates. We used a $0.13\mu m$ CMOS technology, and the gate area does not include RNG, ROM and RAM which are required to store or run programmed protocols. The area specifies a complete EC processor with required registers. The required number of cycles for scalar multiplication is 78 544. Assuming an operating frequency of $700KHz$ expected power consumption is around $11.33\mu W$ per point multiplication, which is a promising figure for the targeted applications. Table 3 gives the performance results for the two proposed protocols.

Moreover, this work includes the coordinate conversion to affine-coordinates from Z-coordinates while the work of [17] gives output in Z-coordinates. This is very important as it was shown that representations with projective coordinates imply some weaknesses with respect to side-channel security [21].

Table 3 Performance results of our protocols.

Protocols	# PM/tag †	Cycles	Time (*ms*)
Grouping proof without CTP (Fig. 4)	2	157 088	224
Grouping proof with CTP (Fig. 2)	3	235 632	295

(† PM denotes the number of point multiplications.)

8 Conclusions

Various grouping-proof protocols have been proposed in the literature to enable multiple tags to generate a proof that they were scanned (virtually) simultaneously. The common property for all the schemes proposed so far is the use of symmetric-key primitives. However, this often results into scalability issues, and several security and/or privacy problems. In this article, we have shown that the ID-transfer protocol of Lee *et al.* can be extended to an efficient multi-party privacy-preserving grouping-proof protocol for RFID that is based solely on ECC. The only complex operations required from the tags are the generation of a random number and two EC point multiplications. Next to this basic protocol we show how to extend the protocol to multiple ($n > 2$) tags and how to reduce the complexity in the scenario when tags cannot collude or when an adversarial reader cannot impersonate a set of tags it has scanned before.

In addition, we presented a hardware architecture that can realize the proposed grouping-proof protocols. The performance results show the feasibility of the protocols even for a passive tag and outperforms other EC-based protocols proposed in the literature.

Acknowledgements This work was supported in part by the IAP Programme P6/26 BCRYPT of the Belgian State, by the Concerted Research Action (GOA) TENSE of the Flemish Government, by FWO project G.0300.07, by the European Commission under contract number ICT-2007-216676 ECRYPT NoE phase II, and by the K.U. Leuven-BOF (OT/06/40).

References

1. Avoine, G.: Adversarial Model for Radio Frequency Identification. Cryptology ePrint Archive, Report 2005/049 (2005). `http://eprint.iacr.org/`
2. Batina, L., Lee, Y., Seys, S., Singelée, D., Verbauwhede, I.: Short Paper: Privacy-Preserving ECC-based Grouping Proofs for RFID. In: To appear in the proceedings of the 13th International Conference on Information Security, Lecture Notes in Computer Science. Springer-Verlag (2010)

3. Bolotnyy, L., Robins, G.: Generalized "Yoking-Proofs" for a Group of RFID Tags. Annual International Conference on Mobile and Ubiquitous Systems (MOBIQUITOUS '06) pp. 1–4 (2006)
4. Brands, S., Chaum, D.: Distance-Bounding Protocols. In: Advances in Cryptology (EUROCRYPT '93), *Lecture Notes in Computer Science*, vol. 765, pp. 344–359. Springer-Verlag (1994)
5. Bringer, J., Chabanne, H., Icart, T.: Cryptanalysis of EC-RAC, a RFID Identification Protocol. In: International Conference on Cryptology and Network Security - CANS'08, Lecture Notes in Computer Science. Springer-Verlag (2008)
6. Burmester, M., de Medeiros, B., Motta, R.: Provably Secure Grouping-Proofs for RFID Tags. In: G. Grimaud, F.X. Standaert (eds.) Proceedings of the 8th IFIP WG 8.8/11.2 International Conference on Smart Card Research and Advanced Applications (CARDIS '08), *Lecture Notes in Computer Science*, vol. 5189, pp. 176–190. Springer (2008)
7. Danev, B., Benjamin, T., Čapkun, S.: Physical-layer Identification of RFID Devices. In: Proceedings of the 18th USENIX Security Symposium (USENIX Security '09), pp. 125–136. USENIX (2009)
8. Deursen, T., Radomirović, S.: Attacks on RFID Protocols. In: Cryptology ePrint Archive: listing for 2008 (2008/310) (2008)
9. Hancke, G.P., Kuhn, M.G.: An RFID Distance Bounding Protocol. In: Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM '05), pp. 67–73. IEEE Computer Society (2005)
10. Hein, D., Wolkerstorfer, J., Felber, N.: ECC is Ready for RFID - A Proof in Silicon. In: R. Avanzi, L. Keliher, F. Sica (eds.) Selected Areas in Cryptography, vol. 5381, pp. 401–413 (2009)
11. Juels, A.: "Yoking-Proofs" for RFID Tags. In: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops (PERCOMW '04), pp. 138–143. IEEE Computer Society (2004)
12. Juels, A., Weis, S.A.: Authenticating pervasive devices with human protocols. In: In Proc. of CRYPTO'05, volume 3126 of LNCS, pp. 293–308. IACR, SpringerVerlag (2005)
13. Juels, A., Weis, S.A.: Defining Strong Privacy for RFID. Cryptology ePrint Archive, Report 2006/137 (2006). `http://eprint.iacr.org/`
14. Koblitz, N.: Elliptic Curve Cryptosystem. Math. Comp. **48**, 203–209 (1987)
15. Lee, Y.K., Batina, L., Singelée, D., Verbauwhede, I.: Low-Cost Untraceable Authentication Protocols for RFID (extended version). In: S. Wetzel, C.N. Rotaru, F. Stajano (eds.) Proceedings of the 3rd ACM Conference on Wireless Network Security (WiSec '10), pp. 55–64. ACM (2010)
16. Lee, Y.K., Batina, L., Verbauwhede, I.: EC-RAC (ECDLP Based Randomized Access Control): Provably Secure RFID authentication protocol. In: IEEE International Conference on RFID, pp. 97–104. IEEE (2008)
17. Lee, Y.K., Sakiyama, K., Batina, L., Verbauwhede, I.: Elliptic Curve Based Security Processor for RFID. IEEE Transactions on Computer **57**(11), 1514–1527 (November 2008)
18. Leng, X., Lien, Y., Mayes, K., Markantonakis, K., Chiu, J.H.: Select-Response Grouping Proof for RFID Tags. In: Proceedings of First Asian Conference on Intelligent Information and Database Systems - ACIIDS 2009, pp. 73–77 (2009)
19. Lien, Y., Leng, X., Mayes, K., Chiu, J.H.: Reading order independent grouping proof for RFID tags. In: Proceedings of IEEE International Conference on Intelligence and Security Informatics, pp. 128–136 (2008)
20. Miller, V.: Use of Elliptic Curves in Cryptography. In: Advances in Cryptology (CRYPTO '85), *Lecture Notes in Computer Science*, vol. 218, pp. 417–426. Springer-Verlag (1986)
21. Naccache, D., Smart, N.P., Stern, J.: Projective coordinates leak. In: C. Cachin, J. Camenisch (eds.) Advances in Cryptology (EUROCRYPT '04), *Lecture Notes in Computer Science*, vol. 3027, pp. 257–267. Springer (2004)
22. Ng, C.Y., Susilo, W., Mu, Y., Safavi-Naini, R.: RFID Privacy Models Revisited. In: European Symposium on Research in Computer Security (ESORICS'08), *Lecture Notes in Computer Science*, vol. 5283, pp. 251–266. Springer-Verlag (2008)
23. Peris-Lopez, P., Hernandez-Castro, J., Estevez-Tapiador, J., Ribagorda, A.: Solving the Simultaneous Scanning Problem Anonymously: Clumping Proofs for RFID Tags. In: In the Proceedings of the 3rd International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU '07). IEEE Computer Society Press (2007)
24. Peris-Lopez, P., Orfila, A., Hernandez-Castro, J., van der Lubbe, J.: Flaws on RFID Grouping-Proofs. Guidelines for Future Sound Protocols. Journal of Network and Computer Applications (2010)
25. Piramuthu, S.: On Existence Proofs for Multiple RFID Tags. In: Proceedings of the 2nd International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU '06), pp. 317–320. IEEE, IEEE Computer Society Press (2006)

26. Saito, J., Sakurai, K.: Grouping Proof for RFID Tags. In: 19th International Conference on Advanced Information Networking and Applications (AINA '05), pp. 621–624. IEEE Computer Society (2005)
27. Schnorr, C.P.: Efficient Identification and Signatures for Smart Cards. In: G. Brassard (ed.) Advances in Cryptology (CRYPTO '89), *Lecture Notes in Computer Science*, vol. 435, pp. 239–252. Springer-Verlag (1989)
28. Smart, N.: Cryptography, An Introduction, 3th edn. McGraw-Hill (2002). 433 pages
29. of Standards, N.N.I., Technology: Cryptographic Hash Algorithm Competition. `http://csrc.nist.gov/groups/ST/hash/sha-3/index.html`
30. Vaudenay, S.: On privacy models for RFID. In: Advances in Cryptology (ASIACRYPT'07), *Lecture Notes in Computer Science*, vol. 4833, pp. 68–87. Springer-Verlag (2007)