

# Hierarchical ECC-Based RFID Authentication Protocol

Lejla Batina, Stefaan Seys, Dave Singelée, and Ingrid Verbauwhede

<sup>1</sup> K.U.Leuven ESAT/SCD-COSIC and IBBT  
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium  
`firstname.lastname@esat.kuleuven.be`

<sup>2</sup> Radboud University Nijmegen, CS Dept./Digital Security group  
Heyendaalseweg 135, 6525 AJ Nijmegen, The Netherlands  
`lejla@cs.ru.nl`

**Abstract.** RFID (Radio Frequency Identification) technology enables readers to scan remote RFID tags, and label the objects and people to which they are attached. Current cryptographic authentication protocols deployed in heterogeneous environments are often not compatible, or reveal too much information to the RFID readers. To tackle this problem, we introduce the concept of RFID groups and propose a hierarchical RFID authentication protocol. By using this protocol, an RFID tag can tune its identification process to the type of reader it is communicating with. Only a subset of readers can learn the identity of a particular tag, while others can only acquire information on the group to which the tag belongs. Our protocol offers impersonation resistance and is narrow-strong privacy-preserving. Furthermore, we extend the concept to multiple level of subgroups, and demonstrate the feasibility of our proposed protocols for RFID tags.

**Keywords:** RFID, Authentication, ECC, Hierarchical Groups, Privacy

## 1 Introduction

Radio Frequency Identification is a technology designed to automatically identify objects and people. RFID systems are rapidly expanding their applications to many areas: inventory systems, supply chains, access control, vehicle tracking, toll payments, e-ticketing, pharmaceuticals, *etc.* However, due to the wide spread of tags, there are potentially various security and privacy risks. Nowadays, the vast majority of the tags being used only provide an identity number (or Electronic Product Code), and neither authentication nor any kind of privacy is achieved. To tackle the identified threats, there is a clear demand for secure and privacy-preserving RFID protocols.

A large part of the RFID security research is currently focused on RFID identification protocols. The protocols differ in the cryptographic building blocks they use, their efficiency, message flows, and security and privacy properties they offer. But all of them are carried out between a tag and a reader, in which the latter learns the identity of the tag at the end of a successful protocol run.

In many real life situations a tag will not reside in one place, but will be located in different environments during its lifetime. For example, an RFID tag attached to an object will move from the manufacturer to the costumer. Throughout the supply chain, the tag will travel across several companies, and communicate to readers which are not operated by the same organization. At the various stages during the lifetime of a tag, there will be different requirements regarding the identification of the product to which the tag is attached. While it is important for the manufacturer to identify the tag (*i.e.*, to learn its exact identity), it could be sufficient for intermediate parties or the customer to only know the manufacturer of the product, or the type of product. This translates to a need for a more granular approach, in which the tag only reveals the necessary information to which that specific reader is entitled.

### 1.1 RFID groups

To realize this notion, we introduce the concept of an “RFID group”. Each tag belongs to one of these groups, and can be identified both by its unique identity and by the group to which it belongs. During the authentication process, the level of detail of the information revealed by the tag (*i.e.*, its identity or its RFID group) is determined by the reader to which it is communicating. Some readers are authorized to learn the tag’s identity, while other readers can only obtain the tag’s RFID group (or no information at all). One might notice that this concept is quite similar to the notion of anonymous credentials [3, 7, 8]. There is one important difference. When using credential systems, the prover constructs a message (*i.e.*, the credential) depending on the properties it wants to prove. In our setting, the information that is revealed depends on the reader that is participating in the protocol, and is not chosen by the tag.

Introducing the concept of RFID groups significantly improves scalability and compatibility of large RFID systems. Without using groups, all readers need a list of all tags’ keys to successfully carry out an authentication protocol. Without these keys, a reader cannot verify the authenticity of a tag. However, distributing these keys among all readers is quite cumbersome and potentially even undesirable, since the readers can be controlled by different parties. By using the notion of RFID groups, readers are destined to belong to an authentication group. Depending on the group they belong to, they will obtain a set of verification keys. Readers belonging to other groups do not have to know these keys.

We demonstrate the use of RFID groups with two practical examples. The first example is related to the supply chain we mentioned above. Suppose that in the near future many consumer goods will come with an RFID as a bar-code replacement. By employing no security or conventional authentication mechanisms, the tag will reveal its unique identity to the reader. The privacy problems resulting from employing no security has been extensively criticized (see [14] for an overview). Privacy-preserving authentication methods such as [6, 20] protect the privacy of

the user from eavesdroppers, but still reveal the unique identifier to an authorized reader. Using the concept of RFID groups presented in this paper, we can create tags that are capable of proving group membership to any reader with the correct group verification key. In this way we could, for example, manufacture tags for medicine packages that contain a unique identifier (that uniquely identifies this particular package and all its details), but also an identifier of a group that only specifies the type of drug and a third identifier that specifies the fact that this is an FDA approved drug. RFID readers in the supply chain will have access to the unique ID and thus access to all the details, the same holds for hospitals, emergency response units and any other entity that need this detailed information. Everybody else will be able to obtain a reader that only has access to the group that specifies that this drug has been FDA approved. This enables people to perform an independent check of the drug’s validity, but does not allow them to obtain any other information; thus preventing malicious individuals from obtaining details of medicines carried around by other people.

A second example is access control. Assume a large corporate building is protected with an access control system based on RFID. Further suppose that each employee is part of one “access group” that allows them access to a set of hallways or rooms within the building. Using RFID groups, their RFID card could contain the identities of the group they belong to and their unique employee number. Instead of always providing this unique employee number to any reader in the building (as is the case now), readers will only obtain information on the access group of the user. Once inside the building, the user can use the same RFID tag to log in to his terminal using the unique employee number of the card. This allows fine grained access control (using multiple groups), but still protects the privacy of the employee (as readers will only obtain the access group and not the unique ID of the employee).

## 1.2 Contributions and outline

In this paper, we propose a hierarchical, secure, privacy-preserving RFID authentication protocol, which incorporates the concept of RFID groups. Depending on the keys used during the verification process, the reader will learn the necessary information to which it is entitled to. This can be the identity of the tag or the group to which the tag belongs. We prove that the protocol is narrow-strong privacy-preserving and is resistant to impersonation attacks. It is exclusively based on ECC (Elliptic Curve Cryptography) [19, 23] and can be easily extended to the case with  $n$  levels in the group hierarchy. Moreover, we present the performance results of our protocol on an ECC coprocessor, to show that the protocols are feasible for RFID tags.

The remainder of the paper is organized as follows. In Sect. 2, RFID authentication protocols are reviewed. In Sect. 3 we describe the setting of hierarchical RFID groups. Next, we present our basic hierarchical RFID

authentication protocol in Sect. 4, and show that it can be easily extended to the setting where there are multiple levels of RFID subgroups. The security and privacy properties of the protocol are discussed in Sect. 5. The performance results of our protocol are outlined in Sect. 6. We conclude our work in Sect. 7.

## 2 Related work

To solve the security and privacy issues posed by RFID technology, various RFID authentication protocols have been proposed in the literature. So far, most schemes rely exclusively on symmetric-key cryptography. One of the first was the work of Feldhofer *et al* [12] that proposed a challenge-response protocol based on the AES block-cipher. The implementation consumes a chip area of 3,595 gates and has a current consumption of  $8.15 \mu A$  at a frequency of  $100 kHz$ . Juels and Weis proposed the HB+ protocol [18], which was designed as an efficient solution, as it even can be implemented on tags of 5-10 cents, and offers protection against active adversaries. Later other variants of HB followed. However, it is shown that these are vulnerable to various security flaws. For example, Gilbert *et al.* [15] presented a man-in-the-middle attack that uses failed authentications to extract the HB+ key. As a fix, a new protocol called HB++ from Bringer *et al.* [5] was proposed. HB++ is claimed to be secure against man-in-the-middle attacks but it requires additional secret key material and a universal hash function to detect the attacks. In the follow-up work Bringer and Chabanne [4] proposed a new HB+ variant (so-called Trusted-HB) using special linear feedback shift register (LFSR) constructions. However Frumkin and Shamir [13] discovered several weaknesses of Trusted-HB. Various other symmetric-key based authentication protocols have been proposed for RFID, each having specific security and privacy properties. However, since these protocols are not the main focus of the paper, we will not discuss them further.

The main reason why most work focused on symmetric-key solutions lies in the common perception of public-key cryptography being too slow, complex and power-hungry for RFID. However, recent publications on compact and efficient Elliptic Curve Cryptography (ECC) implementations challenge this assumption [16, 20, 22, 25]. Using public-key protocols solves the scalability issues that often burden symmetric-key based solutions and can offer strong privacy protection [27]. One of the first ECC based authentication protocols is the EC-RAC (Elliptic Curve Based Randomized Access Control) protocol that has been proposed to address tracking attacks. However, in [6, 9–11], it is shown that EC-RAC is vulnerable to various man-in-the-middle and replay attacks. As a result, the EC-RAC protocol has been gradually revised in [20, 21] to tackle the known attacks and offer narrow-strong privacy. Furthermore, Bringer, Chabanne and Icart proposed the randomized Schnorr protocol [6] (an extension of the basic Schnorr protocol [26]) as an efficient alternative that is also

narrow-strong privacy-preserving. The hierarchical RFID authentication protocol we propose in this paper is inspired by this protocol.

### 3 Setting

#### 3.1 Notation

Let us first introduce the notation used in this work. We denote  $P$  as the base point on an Elliptic Curve. As will be discussed later, a reader has multiple key pairs. We denote these reader’s private and public-key pairs as  $y_i$  and  $Y_i(= y_iP)$ , where  $y_iP$  denotes the point derived by the point multiplication operation on the Elliptic Curve group. Also an RFID tag will have multiple key pairs, corresponding to its identity and the (sub)groups where it belongs to. These private and public-key pairs are respectively denoted by  $x_i$  and  $X_i$ .

#### 3.2 Group structure

In our setting there are two types of entities involved: tags and readers. Each tag has a unique identity and communicates to a (potentially untrusted) reader during the execution of the hierarchical authentication protocol. A reader that is part of the RFID system is denoted as an authorized reader, all other readers are unauthorized readers. Only authorized readers are allowed to learn (some) information from a tag.

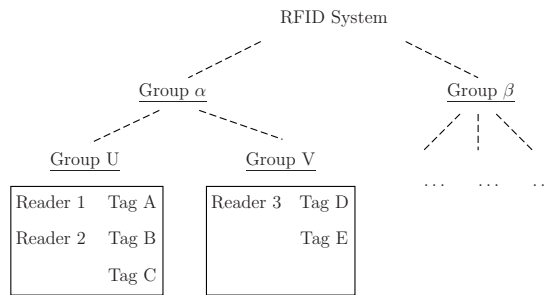


Fig. 1: Example RFID system divided into groups  $\alpha$  and  $\beta$ , in which group  $\alpha$  is subdivided in subgroups  $U$  and  $V$ .

The complete set of readers and tags within the system is divided into a hierarchical group structure consisting of groups and subgroups. The top level in the tree is the RFID system itself. The leaves of the tree are individual tags and readers. Fig. 1 shows an example with two main groups  $\alpha$  and  $\beta$ , in which group  $\alpha$  is further subdivided in subgroups  $U$  and  $V$ .<sup>1</sup> Both readers and tags are assigned to one subgroup at the

<sup>1</sup> For clarity we have limited this example to 2 layers of groups, but our scheme allows an arbitrary level of subgroups.

lowest level (group  $U$  or  $V$ ) in our example. Entities which are part of a subgroup, automatically obtain membership of the parent group. This inheritance of group membership continues until the root of the group tree has been reached. For example, Reader 3 has been assigned to the lowest level subgroup  $V$  and therefore automatically obtained membership of the parent group  $\alpha$ . Because he is now part of group  $\alpha$  he also becomes part of the top level group, *i.e.*, the RFID system itself.

Once these groups have been set up, tags and reader can start using the protocol described in Sect. 4 to allow readers to verify group membership of tags. The level of detail of group membership a reader can verify depends on the group membership of both the tag and the reader:

1. The reader and the tag belong to the same lowest level group in the tree. In this scenario, the reader will be able to verify all group memberships of the tag, including the identity of the tag itself. For example, Reader 3 in Fig. 1 can verify that Tag D is part of the RFID system, part of group  $\alpha$ , part of subgroup  $V$  and has identity  $D$ .
2. The reader and the tag do not belong to the same lowest level group, but do share a higher level group. In this scenario, the reader will be able to check group membership of the tag *up to the level of group they share – plus one, starting from the top*. For example, Reader 3 in Fig. 1 can verify that Tag A is part of the RFID system, part of group  $\alpha$ , and part of subgroup  $U$ . But because they are not part of the same subgroup at the lowest level, Reader 3 is not able to obtain/verify the identity of Tag A.
3. The reader and the tag are not part of the same RFID system (*i.e.*, reader and tag do not share any group). In this scenario, the reader is not authorized and is not able to obtain any information on the tag.

**Key setup.** We will now introduce the key setup that is used in our protocol (described in Sect. 4). To illustrate the notation discussed above, let us revisit the example. Figure 2 shows the group structure and the private keys associated to the groups and subgroups. First, consider the readers in the system. Every reader obtains the private key  $y_{i,G}$  of the group  $G$  of which it is a member at level  $i$ . This key is required to check the group membership of tags of subgroups at level  $i$ . For example, Reader 3 obtains key  $y_{1,V}$  and can use this to obtain the identities of tags  $D$  and  $E$ . He also obtains key  $y_{2,\alpha}$  that can be used to verify membership of either group  $U$  or  $V$ . Finally, he obtains private key  $y_3$  that can be used to verify membership of either group  $\alpha$  or  $\beta$ .

In order to prove membership, tags require a set of private keys. Again, a tag obtains a single private key  $x_{j,G}$  at each level  $j$  for the group  $G$  of which this tag is a member. For example, tag  $A$  has knowledge of the private keys  $x_{1,A}$  (to prove its identity),  $x_{2,U}$  (to prove membership of group  $U$ ), and  $x_{3,\alpha}$  (to prove membership of group  $\alpha$ ). Table 1 give a complete overview of the private keys stored by the different entities in the example RFID system.

**Protocol use.** Before explaining the details of the protocol, we will demonstrate how the protocol is used to obtain the group membership of a particular tag. Take for example the case in which Reader 3 interrogates tag  $E$  in Fig. 2. First, the tag will generate a proof that it is part of group  $\alpha$  using the private key  $x_{3,\alpha}$ . Because the reader has key  $y_3$ , it is able to verify this claim. Next, the tag constructs a proof of membership of group  $V$  using the private key  $x_{2,V}$ . The reader can verify this using the private key  $y_{2,\alpha}$ . Finally, the tag will prove its identity using the private key  $x_{1,E}$ . The reader opens this proof using the private key  $y_{1,V}$ . This tree traversal is indicated with the dotted arrow in Fig. 2.

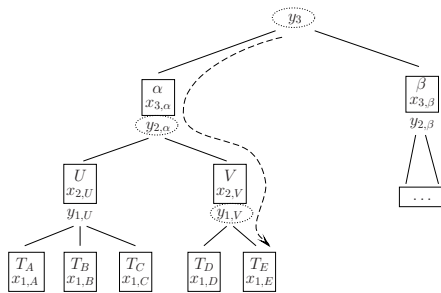


Fig. 2: Example RFID group structure showing groups ( $\alpha, \beta, U, V$ ), private keys required to check group membership ( $y_{i,G}$ ) and private keys to prove group membership ( $x_{j,G}$ ).

Entity	Group	Private keys
Tag A	$U$	$x_{1,A} x_{2,U} x_{3,\alpha}$
Tag B	$U$	$x_{1,B} x_{2,U} x_{3,\alpha}$
Tag C	$U$	$x_{1,C} x_{2,U} x_{3,\alpha}$
Tag D	$V$	$x_{1,D} x_{2,V} x_{3,\alpha}$
Tag E	$V$	$x_{1,E} x_{2,V} x_{3,\alpha}$
Reader 1	$U$	$y_{1,U} y_{2,\alpha} y_3$
Reader 2	$U$	$y_{1,U} y_{2,\alpha} y_3$
Reader 3	$V$	$y_{1,V} y_{2,\alpha} y_3$

Table 1: Private keys stored in tags and readers

To simplify the notation, we will denote the identity of the tag, or the group where it belongs to, by its private key. In the example above, the identity of tag  $A$  will be denoted by  $x_{1,A}$ , and the identity of group  $U$  by  $x_{2,U}$ . Note that, although the name suggests that it can be publicly known, the identity of a group or a tag should be kept secret (as these are equal to the corresponding private keys). To check the identity of a group or a tag, the corresponding public key is computed by the reader. For further simplification of the notation, we will assume that both the reader and the tag are part of the same lowest level subgroup and thus that the tree traversal will go from the top until the bottom of the tree. This means that for every private key  $x_{j,G}$  of the tag, the reader will have the corresponding verification key  $y_{i,G}$ . This means that we can omit the second subscript  $G$  in the notation of these keys.

## 4 Hierarchical authentication protocol

### 4.1 Security and privacy requirements

The goal of this paper is to propose a hierarchical RFID authentication protocol, in which a tag can prove to a reader to which group it belongs and/or its identity. The protocol should offer *impersonation resistance*. It

should be impossible for a tag to spoof the identity of another tag, or spoof the membership to another group than the one to which it belongs (i.e. membership to a group for which it does not possess the correct private keys). Note that it is impossible to prevent an attacker from (falsely) proving membership to a particular group of which he has obtained the corresponding private key (e.g., stolen from a tag that belongs to that group).

Besides impersonation resistance, our protocol should also offer *untraceability*, in which the (in)equality of two tags must be impossible to determine. Only a trusted reader should be able to check the identity and groups of the tags. To evaluate the privacy of RFID systems, several theoretical models have been proposed in the literature [1, 17, 24, 27]. We particularly focus on two characteristics of attackers from the theoretical framework of Vaudenay [27]: *wide* (or *narrow*) attackers and *strong* (or *weak*) attackers. If an attacker has access to the result of the authentication protocol (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 in an authentication protocol instance, he is a *strong* attacker. Otherwise he is a *weak* attacker. Vaudenay demonstrated that one needs to employ public-key cryptography to achieve strong privacy requirements [27]. Because of this observation, our narrow-strong privacy-preserving hierarchical RFID authentication protocol relies on public-key cryptography. For efficiency reasons, we will particularly use ECC.

It is important to stress that the notion of narrow-strong privacy only refers to the identity of a tag. Untraceability regarding the membership of a group can only be achieved partially. Readers can always check the membership of a tag to (sub)groups to which they also belong. If the reader does not belong to a particular (sub)group, then that reader should not be able to check that a tag belongs to this (sub)group. For example, in the scenario depicted in Fig. 1, a reader of group  $\beta$  should not be able to verify that tag A belongs to group U.

## 4.2 Protocol description

We describe here our basic privacy-preserving hierarchical authentication protocol, where each tag belongs to one group. There are no subgroups defined, so each tag has an identity  $x_1$  and belongs to a group  $x_2$ . Such a hierarchical scheme can be trivially designed as follows:

- In Sect. 2, we discussed several RFID authentication protocol. Out of this list, choose the appropriate protocol, according to the required privacy and security requirements.
- Carry out this protocol twice. The first protocol run uses the group's private key  $x_2$  and the public key  $Y_2$  of the reader, and is used to prove the group where the tag belongs to. The second protocol run uses the tag's private key  $x_1$  and the public key  $Y_1$  of the reader, and is used to prove the tag's identity.



Although the approach discussed above works, it is not efficient. Therefore, we propose a hierarchical authentication protocol in which only one protocol run will be carried out. After receiving a challenge from the reader, the tag will reply with a single response. Depending on the key used to check the correctness of the response, the reader will be able to verify the group where the tag belongs to, the identity of the tag, or even nothing at all. Figure 3 shows the basic protocol.

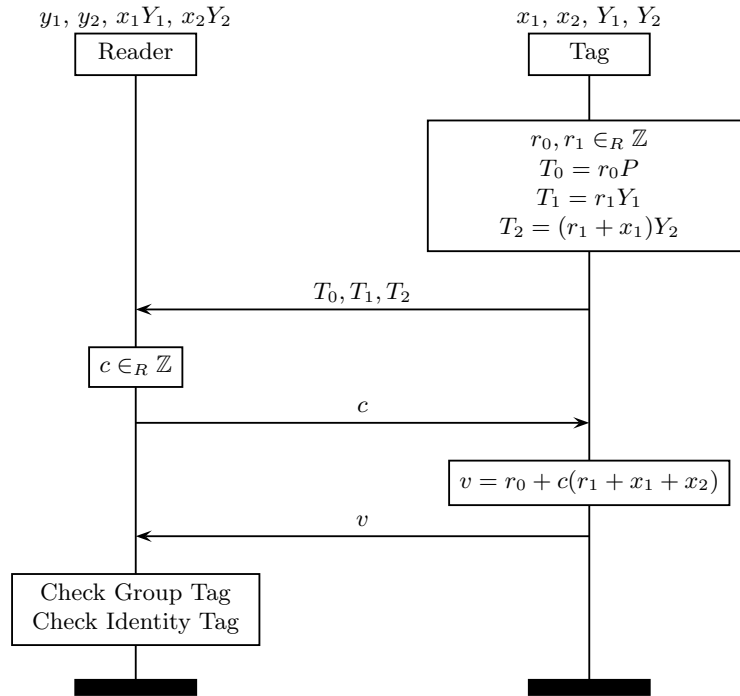


Fig. 3: Basic hierarchical RFID authentication protocol

The protocol starts by the tag generating two random numbers  $r_0$  and  $r_1$ . Next, it computes three points on an elliptic curve:  $T_0$ ,  $T_1$  and  $T_2$ , and sends them to the reader. Then, the reader responds with a random challenge  $c$ . After receiving this value, the tag computes the response  $v$  using the challenge  $c$  and the private keys  $x_1$  and  $x_2$ . The tag first checks that the challenge is not equal to zero in the group or the order of the point  $P$ . Next, the response is sent back to the verifier, to prove the tag's identity and/or being part of a group.

After having received the response  $v$ , the reader is going to perform several checks. First, it checks the group where the tag belongs to by

performing the following computation, using its private key  $y_2$ :

$$c^{-1}(vY_2 - y_2T_0 - cT_2) = x_2Y_2$$

If the correct private key  $y_2$  is used (*i.e.*, the reader belongs to the same RFID system as the tag), the result of the computation will be equal to  $x_2Y_2$ . This point on the curve is defined as the public key of group  $x_2$ . If the incorrect key is used, the output of the computation will be random (*i.e.*, output cannot be used to identify or track the tag or the group of tags).

Next, the reader checks the identity of the tag, using the private keys  $y_1$  and  $y_2$ . It performs the following computation:

$$(y_2^{-1}y_1)T_2 - T_1 = x_1Y_1$$

Since the reader already checked the group  $x_2$ , it knows that the key  $y_2$  was correct. If the correct private key  $y_1$  is used (*i.e.*, both the reader and the tag belong to the same group  $x_2$ ), the output of the computation will be equal to  $x_1Y_1$ . This point on the curve is defined as the public key of the tag. If the incorrect key is used, the output of the computation will be random (*i.e.*, the output cannot be used to identify or track the tag or the group of tags).

Note that it is very important that the reader first checks the group of the tag, and only then the identity. This order should not be altered. If the reader cannot compute the public key of the group, because the reader has the incorrect private key, it should immediately stop the verification procedure and not compute the identity of the tag. Otherwise, the protocol would become vulnerable to a replay attack.

To avoid timing attacks, the time needed by the reader to carry out the verification steps should be randomized. Otherwise, the outcome of the verification procedure and even the identity of the tag depends on this verification time, which would break the privacy properties of our scheme. For example, if the reader searches linearly in the database of the tags' public keys, then it takes less time to check the correctness of the public keys which are stored in the beginning of this database.

### 4.3 Extension to $n$ levels of subgroups

The basic protocol can be extended to the setting where there are  $n - 1$  levels of subgroups. As discussed in Sect. 3.2,  $x_1$  is the identity of the tag, the group  $x_2$  is the subgroup at the lowest level in the hierarchy, and  $x_n$  the group at the top level in the hierarchy. The protocol is shown in Fig. 4.

As in the basic protocol, the tag first generates two random numbers, and then computes the points  $T_0, T_1, \dots, T_n$ . The reader then generates a random challenge and sends it to the tag. After receiving this value, the tag computes the response  $v$  using the challenge  $c$  and the private keys

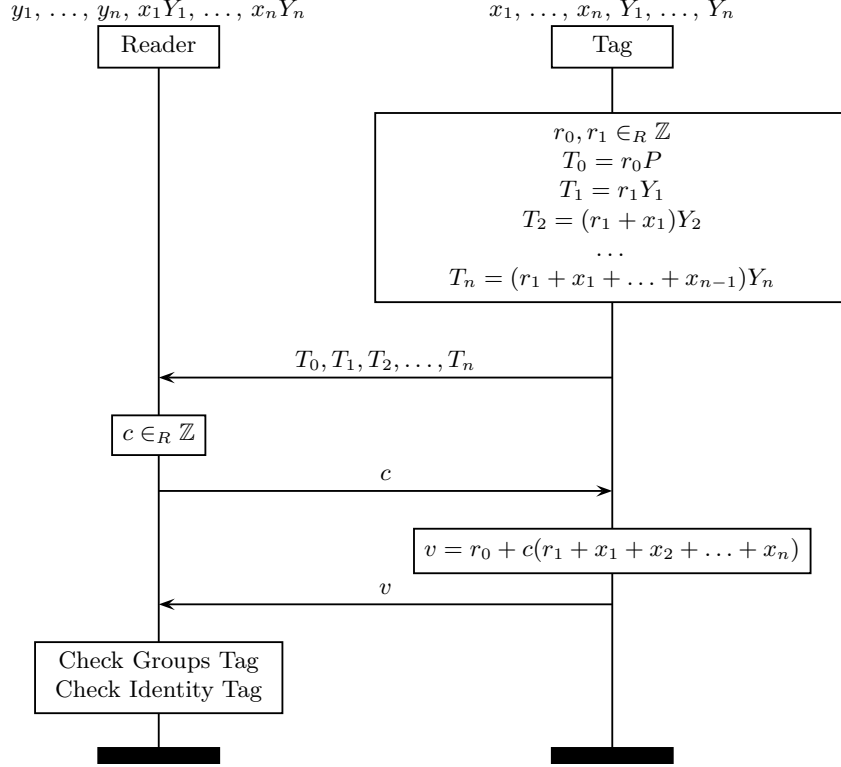


Fig. 4: Extended hierarchical RFID authentication protocol

$x_1, x_2, \dots, x_n$ . This response is sent back to the verifier, to prove the tag's identity and/or being part of a particular subgroup.

After having received the response  $v$ , the reader is going to perform several checks. First, it checks the top-level group where the tag belongs to ( $x_n$ ) by performing the following computation, using its private key  $y_n$ :

$$c^{-1}(vY_n - y_nT_0 - cT_n) = x_nY_n$$

If the correct private key  $y_n$  is used (*i.e.*, the reader belongs to the same RFID system as the tag), the result of the computation will be equal to  $x_nY_n$ , the public key of the group  $x_n$ . If the incorrect key is used, the output of the computation will be random (*i.e.*, the output cannot be used to identify or track the tag or the group of tags).

Next, the reader checks the tag's membership to the subgroup at the second highest layer in the hierarchy ( $x_{n-1}$ ), using the private keys  $y_{n-1}$  and  $y_n$ :

$$(y_n^{-1}y_{n-1})T_n - T_{n-1} = x_{n-1}Y_{n-1}$$

Since the reader already checked the group  $x_n$ , it knows that the private key  $y_n$  was correct. If the correct private key  $y_{n-1}$  is used (*i.e.*, both the reader and the tag belong to the same group  $x_n$ ), the output of the computation will be equal to  $x_{n-1}Y_{n-1}$ , the public key of the group  $x_{n-1}$ .

In the next stage, the reader checks the subgroups  $x_{n-2}, \dots, x_2$  until the verification is not successful. As an example, we show the equation needed to check the membership to subgroup  $x_{n-2}$ :

$$(y_{n-1}^{-1}y_{n-2})T_{n-1} - T_{n-2} = x_{n-2}Y_{n-2}$$

If the reader belongs to the subgroup  $x_3$ , all these checks will be correct. In that case, the reader can try to check the identity of the tag, using the private keys  $y_1$  and  $y_2$  as follows:

$$(y_2^{-1}y_1)T_2 - T_1 = x_1Y_1$$

Since the reader already checked the subgroup  $x_2$ , it knows that the key  $y_2$  was correct. If the correct private key  $y_1$  is used (*i.e.*, both the reader and the tag belong to the same subgroup  $x_2$ ), the output of the computation will be equal to  $x_1Y_1$ , the public key of the tag.

As in the basic protocol, it is of uttermost importance that the reader first checks the group  $x_n$ , then the subgroup  $x_{n-1}$ , *etc.* This order should not be altered and the protocol should stop when one of these checks fail (since subgroups at a lower level cannot be checked by that reader). Only when all the checks are correct (because the reader has all the correct private keys), it should check the identity of the tag. As before, the time needed by the reader to carry out the verification steps has to be randomized to avoid timing attacks.

## 5 Analysis

In this section we give the security and privacy analysis of our basic scheme ( $n = 2$ ). Both proofs are related to the basic protocol of Fig. 3, but can be easily extended to the more general case where the hierarchical group structure has depth  $n$  (shown in Fig. 4). First, we remind to some common computational assumptions.

### 5.1 Computational assumptions

The security of ECC protocols is founded on the **ECDL problem**. The ECDL problem is defined as follows:

Let  $E$  be an elliptic curve over  $\mathbb{F}_q$  and let  $P \in E$  be a point of order  $k$  *i.e.*  $ord(P) = k$ . Let  $Q \in \langle P \rangle$  and  $Q = \alpha P$  for  $\alpha \in [0, k)$ . The problem of finding the logarithm  $\alpha$  for given  $P$  and  $Q$  is called the elliptic curve discrete logarithm problem (ECDLP).

Here  $\mathbb{F}_q$  denotes the finite field containing  $q$  elements, where  $q$  is a prime power. In practice, commonly used finite fields are a prime field  $\mathbb{F}_p$

or a binary field  $\mathbb{F}_{2^n}$ . In addition,  $\langle P \rangle$  denotes a group of points on an elliptic curve generated by  $P$ .

The Decisional Diffie-Hellman (DDH) problem is:

Given  $P, \alpha P, \beta P$  where  $\alpha$  and  $\beta$  are randomly chosen in  $[0, k)$  and given  $\gamma P = \alpha\beta P$  with probability  $1/2$  and  $\gamma P = \delta P$  with probability  $1/2$  with  $\delta$  randomly chosen in  $[0, k)$ , decide whether  $\alpha\beta P = \gamma P$ .

In the proof of privacy of our scheme, we make use of the “**extended**” **DDH assumption**, which we can informally define as follows:

Given 5 random multiples of  $P$  on an elliptic curve  $y_1 P, y_2 P, r_1 P, \gamma_1 P, \gamma_2 P$ ; it is intractable to distinguish the case where  $\gamma_1 = y_1 r_1$  and  $\gamma_2 = y_2 r_1$  or where at least one  $\gamma$  has been selected at random.

Furthermore we assume the following theorem holds:

**Theorem 1.** *Assuming the hardness of the DDH problem, then the “extended” DDH problem is also hard.*

## 5.2 Security analysis

We observe that our scheme is clearly correct as a legitimate tag is accepted with probability 1. We can prove that it is also secure against an active adversary by using the fact that the Schnorr scheme, shown in Sect. A, is secure against active impersonation attacks under the OMDL assumption. This fact was proved in [2]. Our scheme is a modification of the Schnorr scheme. Therefore, a relevant adversary against our scheme can be transformed into a relevant adversary against the Schnorr scheme.

The proof is inspired on the *security game* defined in [2], and the security proof given in [6]:

**Security Game:** Assume an adversary is able to interrogate a system of tags via the protocol described in Fig. 3. In a first phase, the adversary pretends to be a verifier (reader) and is allowed to communicate with all tags. In a second phase, the adversary tries to impersonate a tag while communicating with a genuine verifier. The adversary wins if he is accepted as genuine by this verifier.

**Definition 1 (Security)** *A scheme is secure against active impersonation attacks if any adversary is not able to win the game, except with a negligible probability.*

In order to prove that our scheme is secure against active impersonation attacks, we define the reduced basic hierarchical RFID authentication protocol (denoted by RHP) as our basic scheme defined in Fig. 3 with  $r_1 = 0$ .

**Theorem 2.** *Assuming RHP is secure against active impersonation attacks, then our basic scheme (Fig. 3) is secure against active impersonation attacks.*

Note that setting  $r_1$  to 0 only affects privacy, and does not increase the impersonation resistance of RHP compared to our basic scheme.

**Theorem 3.** *Assuming the Schnorr scheme is secure against active impersonation attacks, then RHP is secure against active impersonation attacks.*

**Proof:** We will prove this last theorem by contradiction. Let us assume that there exists an active adversary  $\mathcal{A}$  relevant against RHP, while there exists no adversary  $\mathcal{A}^S$  relevant against the Schnorr scheme. In the following, we will show how to convert  $\mathcal{A}$  into  $\mathcal{A}^S$ .

During the first phase of the attack,  $\mathcal{A}$  interrogates a genuine tag that executes the Schnorr protocol. This protocol starts by the tag outputting  $T_0$  (denoted by  $T$  in Fig. A). We intercept  $T_0$ , randomly generate  $y_2$ , compute  $y_2P = Y_2$ , randomly choose  $x_1$ , and compute  $T_2 = x_1Y_2$ . Next, we send  $T_0$ , and  $T_2$  to  $\mathcal{A}$ . After receiving  $c$  sent by  $\mathcal{A}$ , the tag outputs  $v$ . We intercept this value and we send  $v + cx_1$  to  $\mathcal{A}$ . Because of our interceptions,  $\mathcal{A}$  is convinced that it executed RHP.

During the second phase of the protocol,  $\mathcal{A}$  tries to impersonate the tag it has interrogated in the first phase towards a genuine reader executing the Schnorr protocol. As before, we will intercept the communication.  $\mathcal{A}$  starts by outputting  $T_0$  and  $T_2$ . We intercept both values, and only forward  $T_0$  to the reader. The latter replies with a challenge  $c'$ , which we forward to  $\mathcal{A}$ . Next,  $\mathcal{A}$  responds with a value  $v'$  for which the following holds:  $x_2Y_2 = c'^{-1}[v'Y_2 - y_2(T_0 + y_2^{-1}c'T_2)]$  and  $T_2 = x_1Y_2$ . We intercept  $v'$ , compute  $v = v' - c'x_1$  and send  $v$  to the reader. One can verify that the reader will accept this value  $v$ . This means that we have successfully transformed  $\mathcal{A}$  into  $\mathcal{A}^S$ . This contradicts our assumption and the statement is proven.  $\square$

### 5.3 Privacy analysis

We now explain why our scheme is narrow-strong private. In our privacy analysis, which is inspired by [6], we will use the *privacy game* from [27]. In this game there are tags, an adversary, and a **blinder**. The blinder sits in between tags and the adversary, hiding the former from the latter. The model dictates that the blinder does not know which tag it is simulating and it cannot interact with genuine tags. For the details of the privacy game, we refer to [27]. Briefly, the game consists of two phases:

**Privacy game:** Assume every tag is known to the attacker by its pseudonym. First, the adversary is allowed to communicate with (or eavesdrop on, if he is passive) genuine tags. After this phase, the attacker receives the map  $\mathcal{T}$  of pseudonyms to real IDs of all the tags. After some analysis, the adversary is asked to output either true or false. The adversary wins the game if he outputs true<sup>2</sup>. In the second phase, the adversary is only allowed to communicate with the blinder, who is simulating the tags' outputs. Again, the adversary is given the map  $\mathcal{T}$  and asked to output either true or false.

<sup>2</sup> Note that there are a number of trivial adversaries, such as the one that always outputs true)

**Definition 2 (Privacy)** *A scheme is private if there exists a blinder such that no adversary has an advantage (except with negligible probability) between the two phases of the privacy game.<sup>3</sup>*

“Informally, an adversary is trivial if it makes no effective use of protocol messages. Namely, these messages can be simulated without significantly affecting the success probability of the adversary [27].” In other words, a scheme is private if it is possible to build a simulator that is indistinguishable from genuine tags. Although a narrow-strong adversary has knowledge of the secrets of all tags, we can show that he cannot distinguish a genuine protocol run from a simulated run in the privacy game. This shows that the adversary is not able to link the tags’ outputs and their secrets under the “extended” DDH assumption (defined above):

**Theorem 4.** *Assuming the hardness of the “extended” DDH problem, the scheme described in Fig. 3 is narrow-strong private.*

**Proof:** In order to prove the scheme, we have to show that we can build a simulator (blinder) that can simulate the tag’s outputs and that these simulated outputs cannot be distinguished from genuine outputs by a narrow-strong attacker.

A genuine protocol run between a tag and a reader are of the form  $T_0 = r_0P$ ,  $T_1 = r_1Y_1$ ,  $T_2 = (r_1 + x_1)Y_2$ ,  $c$ , and  $v = r_0 + c(r_1 + x_1 + x_2)$ . A simulator outputs random instances  $A_0, A_1, A_2, c, \alpha$ . In order to win the game, an adversary has to distinguish these random instances from genuine instances  $r_0P, r_1Y_1, (r_1 + x_1)Y_2, c, r_0 + c(r_1 + x_1 + x_2)$ . This is equivalent to distinguishing between  $A_0, A_1, A'_2 = A_2 - x_1Y_2, c, \beta = \alpha - c(x_1 + x_2)$  and  $r_0P, r_1Y_1, r_1Y_2, c, r_0 + cr_1$ . Note that both  $A'_2$  and  $\beta$  are as random as  $A_2$  and  $\alpha$  respectively. Note also that the adversary has knowledge of  $x_1, x_2$  and the public parameter  $Y_2$ . We will now show that distinguishing legitimate quintets from simulated quintets is harder than solving the “extended” DDH problem.

Given an instance  $y_1P, y_2P, r_1P, \gamma_1P, \gamma_2P$  of the “extended” DDH problem, we randomly choose the values  $\beta$  and  $c$ , and compute  $A_0 = \beta P - cr_1P$ . One can now see that the quintet  $A_0, \gamma_1P, \gamma_2P, c, \beta$  are equivalent to a simulation of a protocol transcript. If  $\gamma_1 = r_1y_1$  and  $\gamma_2 = r_1y_2$ , we have  $\beta\gamma_1P = y_1A_0 + c\gamma_1P$  and  $\beta\gamma_2P = y_2A_0 + c\gamma_2P$  and thus the quintet comes from a valid transcript. Otherwise it is a random quintet because either  $\gamma_1$  or  $\gamma_2$  is random. For this reason, if there exists an adversary able to distinguish between simulated protocol runs and genuine ones, he can solve the “extended” DDH problem.  $\square$

Similarly, one can prove that an adversary with knowledge of the private key  $y_2$  (the verifier’s private key of the group  $x_2$ ) is not able to learn anything about the personal private key  $x_1$  of any tag.

---

<sup>3</sup> This definition is sufficient to prove the privacy of a scheme in Vaudenay’s model.

## 6 Performance Results

Cost reduction is an important requirement when designing RFID authentication protocols. In the hierarchical authentication protocol proposed in this paper, a RFID tag has to carry out point multiplications, field multiplications and additions. Of these operations, the former is by far the most complex and energy consuming. Therefore, the number of EC multiplications has to be reduced to a minimum. The Randomized Schnorr protocol proposed by Bringer *et al.* [6], which does not include the concept of groups, requires two EC point multiplications on the tag side. Our basic hierarchical authentication protocol, described in Fig. 3, requires three EC point multiplications. For each extra level introduced in the group hierarchy, the number of EC point multiplications is increased by one. So in the extended protocol shown in Fig. 4, the tag has to compute  $(n + 1)$  EC point multiplications. This makes our solutions scalable for the setting where many hierarchical group levels need to be defined.

Although the reader has more computational resources than the RFID tag, its resources are not inexhaustible. In the Randomized Schnorr protocol proposed by Bringer *et al.*, the reader has to compute three EC point multiplications. In our basic hierarchical authentication protocol, shown in Fig. 3, the reader also has to perform three EC point multiplications. For each extra level introduced in the group hierarchy, the number of EC point multiplications is increased by one. So in the extended protocol shown in Fig. 4, the reader has to compute  $(n + 1)$  EC point multiplications.

This amount of computation is assumed feasible for RFID tags. The advantage of both approaches mentioned above is that the schemes require no additional primitives as they use ECC-only operations. Let us for example consider the ECC hardware processor of Lee *et al.* [22], since its architecture allows for the execution of our protocols. Assuming ECC over a binary field  $\mathbb{F}_{2^{163}}$ , the special curve as in [22], projective coordinates and the use of a Montgomery ladder for point multiplication, we get the following estimates for the protocol. We adjust the clock frequency in order to produce an acceptable performance, which we estimate to 200 *ms* for 1 point multiplication. To have this latency, a frequency of 293 *kHz* is required, as the arithmetic unit has a digit size of 4, resulting in a total number of 58,678 cycles for one point multiplication. In this way, the basic version of the protocol (with  $n = 2$ ) would require 400 *ms* for completion. These numbers show the feasibility of our protocols even for a passive tag and prove the suitability of ECC-based solutions for RFID applications. The implementation details are left out due to the space limitation, but we refer to [20] for a description of an ECC processor (described in detail in [22]) that can perform all operations required. Furthermore, the increase in storage is linear when the depth of the hierarchical group structure (*i.e.* the parameter  $n$ ) is increased. In particular, for each extra level in the group hierarchy, we have to add



additional  $163 \times 4$  bits for key storage (a scalar for a private key and a point  $P(X, Y, Z)$  for a public key).

Table 2 gives a comparison in the number of point multiplications. These results demonstrate that our scheme requires significantly less point multiplications (on both sides) than the trivial solution where the Randomized Schnorr protocol is executed  $n$  times.

Table 2: Feasibility and Privacy Summary

Protocols	Privacy	EC point mult.	
		Server	Tag
$n$ instances of Randomized Schnorr	Narrow-strong	$3n$	$2n$
Our hierarch. protocol (hierarchy level $n$ )	Narrow-strong	$(n + 1)$	$(n + 1)$

## 7 Conclusions and future work

In this work the concept of RFID groups and a hierarchical authentication protocol is introduced. During its lifetime a RFID tag encounters various readers, each of which is not necessarily supposed to learn all the details of the tag. As a solution to this problem we propose a hierarchical authentication protocol that allows a RFID tag to tune its identification process to the type of reader it is communicating with. Hence, only a (designated) subset of readers can learn the identity of a particular tag, while others can only acquire information on the group to which the tag belongs. We also demonstrate that the concept is extendable to multiple number of levels in the group hierarchy.

Furthermore, we prove the security against active adversaries and the privacy properties of our protocols. More precisely, our protocols offer impersonation resistance under the OMDL assumption and are narrow-strong privacy-preserving. Using the performance results for a suitable ECC-based hardware architecture we also demonstrate the feasibility of our proposed protocols for RFID tags.

For  $n$  levels in the group hierarchy, our protocol reduces the number of EC point multiplications at the tag and server by respectively a factor 2 and 3, compared to the trivial solution where  $n$  instances of an RFID authentication protocol are carried out. The tag has to store  $n$  private keys, corresponding to its unique identity and the subgroups where it belongs to. It remains an open problem how to construct a hierarchical RFID authentication protocol where each tag only has one private key. Tags belonging to the same subgroup could have keys which are mathematically related. This relation could then be used by an authorized reader to check the subgroup where the tag belongs to, while any other party should not be able to verify this mathematical property or compute the identity of the tag.

## Acknowledgments

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

## References

1. G. Avoine. Adversarial Model for Radio Frequency Identification. Cryptology ePrint Archive, Report 2005/049 (2005). <http://eprint.iacr.org/>
2. M. Bellare and A. Palacio. GQ and Schnorr Identification Schemes: Proofs of Security Against Impersonation under Active and Concurrent Attacks. In M. Yung, editor, *Advances in Cryptology - CRYPTO'02, Lecture Notes in Computer Science*, volume 2442, pages 162–177. Springer-Verlag, 2002.
3. E. Brickell, J. Camenisch, and L. Chen. Direct Anonymous Attestation. In *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS '04)*, pages 132–145. ACM, 2004.
4. J. Bringer and H. Chabanne. Trusted-HB: A Low-Cost Version of  $HB^+$  Secure Against Man-in-the-Middle Attacks. *IEEE Transactions on Information Theory*, volume 54(9), pages 4339–4342, 2008.
5. J. Bringer, H. Chabanne, and E. Dottax.  $HB^{++}$ : a Lightweight Authentication Protocol Secure against Some Attacks. In *Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU '06)*, pages 28–33. IEEE Computer Society, 2006.
6. J. Bringer, H. Chabanne, and T. Icart. Cryptanalysis of EC-RAC, a RFID Identification Protocol. In *International Conference on Cryptology and Network Security - CANS'08, Lecture Notes in Computer Science*, volume 5339, pages 149–161. Springer-Verlag, 2008.
7. J. Camenisch and A. Lysyanskaya. Signature Schemes and Anonymous Credentials from Bilinear Maps. In *Advances in Cryptology - CRYPTO'04, Lecture Notes in Computer Science*, volume 3152, pages 56–72. Springer-Verlag, 2004.
8. D. Chaum. Security Without Identification: Transaction Systems to Make Big Brother Obsolete. *Communications of the ACM*, volume 28(10), pages 1030–1044, 1985.
9. T. Deursen and S. Radomirović. Attacks on RFID Protocols. In *Cryptology ePrint Archive: listing for 2008 (2008/310)*, 2008.
10. T. Deursen and S. Radomirović. EC-RAC: Enriching a Capacious RFID Attack Collection. In *International Workshop on RFID Security (RFIDSEC '10), Lecture Notes in Computer Science*, volume 6370, pages 75–90. Springer-Verlag, 2010.
11. J. Fan, J. Hermans, and F. Vercauteren. On the Claimed Privacy of EC-RAC III. In *International Workshop on RFID Security (RFIDSEC '10), Lecture Notes in Computer Science*, volume 6370, pages 66–74. Springer-Verlag, 2010.
12. M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. Strong Authentication for RFID Systems using the AES Algorithm. In M. Joye and J. J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems (CHES'04), Lecture Notes in Computer Science*, volume 3156, pages 357–370. Springer-Verlag, 2004.
13. D. Frumkin and A. Shamir. Un-Trusted-HB: Security Vulnerabilities of Trusted-HB. In *International Workshop on RFID Security (RFIDSEC '09)*, pages 62–71, 2009.
14. S. L. Garfinkel, A. Juels, R. Pappu. RFID privacy: An overview of problems and proposed solutions. In *IEEE Security & Privacy*, volume 3(3), pages 34–43. IEEE, 2005.

15. H. Gilbert, M. Robshaw, and H. Sibert. An Active Attack Against  $HB^+$  - a Provably Secure Lightweight Authentication Protocol. *IET Electronic Letters*, volume 41(21), pages 1169–1170, 2005.
16. D. Hein, J. Wolkerstorfer, and N. Felber. ECC is Ready for RFID - A Proof in Silicon. In: R. Avanzi, L. Keliher, F. Sica (eds.) *Selected Areas in Cryptography, Lecture Notes in Computer Science*, volume 5381, pages 401–413. Springer-Verlag, 2009.
17. A. Juels and S. Weis. Defining Strong Privacy for RFID. Cryptology ePrint Archive, Report 2006/137 (2006). <http://eprint.iacr.org/>
18. A. Juels and S. Weis. Authenticating Pervasive Devices with Human Protocols. In *Advances in Cryptology - CRYPTO'05, Lecture Notes in Computer Science*, volume 3126, pages 293–308. Springer-Verlag, 2005.
19. N. Koblitz. Elliptic Curve Cryptosystem. *Math. Comp.*, volume 48, pages 203–209, 1987.
20. Y. K. Lee, L. Batina, D. Singelée, and I. Verbauwhede. Low-Cost Untraceable Authentication Protocols for RFID (extended version). In S. Wetzels, C. N. Rotaru, and F. Stajano, editors, *Proceedings of the 3rd ACM Conference on Wireless Network Security (WiSec '10)*, pages 55–64. ACM, 2010.
21. Y. K. Lee, L. Batina, and I. Verbauwhede. Untraceable RFID Authentication Protocols: Revision of EC-RAC. In *IEEE International Conference on RFID*, pages 178–185. IEEE, 2009.
22. Y. K. Lee, K. Sakiyama, L. Batina, and I. Verbauwhede. Elliptic Curve Based Security Processor for RFID. *IEEE Transactions on Computer*, volume 57(11), pages 1514–1527, November 2008.
23. V. Miller. Use of Elliptic Curves in Cryptography. In *Advances in Cryptology - CRYPTO'85, Lecture Notes in Computer Science*, volume 218, pages 417–426. Springer-Verlag, 1986.
24. C.Y. Ng, W. Susilo, Y. Mu, and R. Safavi-Naini. RFID Privacy Models Revisited. In *European Symposium on Research in Computer Security (ESORICS'08), Lecture Notes in Computer Science*, volume 5283, pages 251–266. Springer-Verlag, 2008.
25. Y. Oren and M. Feldhofer. A low-resource public-key identification scheme for RFID tags and sensor nodes. In *Proceedings of the second ACM conference on Wireless network security - WiSec '09*, pages 59–68, ACM, 2009.
26. C.-P. Schnorr. Efficient Identification and Signatures for Smart Cards. In G. Brassard, editor, *Advances in Cryptology - CRYPTO'89, Lecture Notes in Computer Science*, volume 435, pages 239–252. Springer-Verlag, 1989.
27. S. Vaudenay. On privacy models for RFID. In: *Advances in Cryptology (ASIACRYPT'07), Lecture Notes in Computer Science*, volume 4833, pages 68–87. Springer-Verlag, 2007.

## A ECC-based Schnorr authentication protocol

Many attempts to design an RFID authentication protocol which relies exclusively on the use of ECC, are based on the Schnorr protocol [26], a conventional identification scheme that offers resistance to impersonation attacks, as has been proven by Bellare and Palacio [2]. The protocol of Schnorr is shown in Fig. A.

Although the scheme offers interesting security properties and can be implemented quite efficiently on an RFID tag, it cannot be used directly in the context of RFID networks, as it does not resist tracking attacks. An eavesdropper can compute the value  $X' = c^{-1}(vP - T)$ , which is constant and unique for every tag. As a result, solutions such as the EC-RAC protocol [20] have been proposed to solve these privacy problems.

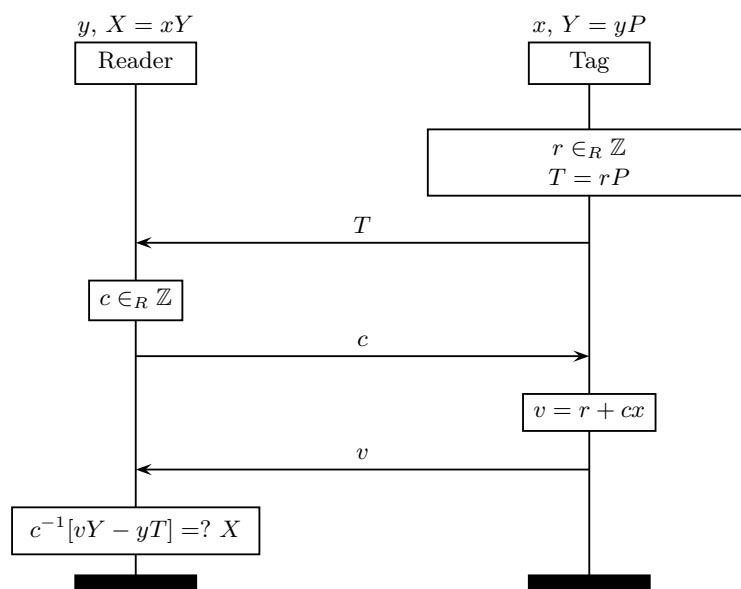


Fig. 5: ECC-based Schnorr identification scheme [26]