Wide–Weak Privacy–Preserving RFID Authentication Protocols

Yong Ki Lee[1], Lejla Batina[2,3], Dave Singelée[2], and Ingrid Verbauwhede[2]

[1] Samsung Electronics Research and Development, South-Korea,
yklee93@kg21.net
[2] IBBT – COSIC, Katholieke Universiteit Leuven, Heverlee, Belgium,
firstname.lastname@esat.kuleuven.be
[3] Digital Security group, Radboud University Nijmegen, Nijmegen, The Netherlands,
lejla@cs.ru.nl

Abstract. The emergence of pervasive computing devices such as RFID tags raises numerous privacy issues. Cryptographic techniques are commonly used to enable tag-to-server authentication while protecting privacy. Unfortunately, these algorithms and their corresponding implementations are difficult to adapt to the extreme conditions implied by the use of RFID. The extremely limited budget for energy and area do not allow the use of traditional cryptography.
In this paper, we address the risk of tracking attacks in RFID networks. Many lightweight protocols have been proposed so far that are founded on both, private- and public-key cryptosystems. We give an overview of existing solutions and discuss the latter ones in more detail. The solutions we advocate in this paper rely exclusively on Elliptic Curve Cryptography (ECC). We describe several authentication protocols that have different computational demands and accordingly different security features. To the best of our knowledge, these protocols are the first ECC-based authentication protocols which offer privacy protection against a wide-weak attacker. Compared to other RFID schemes proposed in the literature, our protocols remain light-weight in terms of area and computation time, while still achieving the required security and privacy properties.

Key words: Authentication Protocol, Privacy, Tracking Attack, Elliptic Curve Cryptography, RFID

1 Introduction

RFID tags, smart labels, sensor nodes are involved in the distributed, wireless, mobile computing revolution, which moves information gathering and processing into the human environment. This evolution has a profound impact on security. Traditional security applications, such as secure gateways, virtual private networks (VPNs), *etc.* focus on protecting the communication channels between computers against attacks. This protection is based on security protocols and cryptographic algorithms running on the powerful processors of physicallyprotected servers. In an environment of small embedded, distributed, wireless connected devices, this assumption is not valid anymore. The embedded device itself is vulnerable to attacks, and a hacker will select the method of attack that breaks the weakest link in an entire system, including the embedded device as well as its communication channel. On top, the embedded device has limited computing and energy resources, and security is expensive (in terms of extra processing, memory, energy and development cost).

Due to the wide-spread of RFID tags, several security and privacy issues arise. Privacy addresses the resistance against unauthorized identification, tracking or linking tags. More in detail, one typically wants to achieve *untraceability*, in which the (in)equality of two tags must be impossible to determine. Several theoretical models to address the privacy of RFID systems have been proposed in the literature [1, 15, 21, 27]. To define privacy in this paper, we import 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 verification (accept or reject) in a server, 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, he is a *strong* attacker. Otherwise he is a *weak* attacker. A *wide-strong* attacker is hence the most powerful. If a protocol is untraceable against a *wide-strong* attacker, we call the protocol *wide-strong* privacy-preserving.

Operational and security requirements for RFID systems include system scalability, anonymity and anti-cloning. Obtaining all these properties presents a substantial research challenge due to rigid constraints in area, memory, power, *etc.* A common work-around is to use protocols using symmetric key cryptographic algorithms. However, the symmetric key based solutions cannot meet all the requirements and it was shown in several publications that public-key cryptography (PKC) is a must in order to have strong security for embedded applications.

In this paper, we present two authentication protocols that use public-key cryptography to achieve the required security and privacy goals. The protocols rely exclusively on the use of Elliptic Curve Cryptography (ECC) and are, to the best of our knowledge, the first ECC-based RFID authentication protocols that are both narrow-strong and wide-weak privacy preserving.

The remainder of the paper is organized as follows. In Section 2, related work is reviewed. We discuss our authentication protocols in detail in Sect.3. We conclude our paper in Section 4.

2 State of the Art

Various RFID authentication protocols have been proposed in the literature. In the beginning the main efforts were on designing solutions that rely exclusively on private-key (also called symmetric-key) cryptography. One of the first was the work of Feldhofer [11] that proposed a challenge-response protocol based on the AES block-cipher. Toiruul and Lee presented an mutual authentication algorithm based on AES [25]. Of other notable solutions for authentication protocols we mention here the HB^+ protocol [16] that was presented as an extremely cheap solution but still secure against active adversaries. It meets even the cost requirements for the tags of 5-10 cents range. Other variants of HB followed, as a result of attacks that appeared, such as the work of Gilbert *et al.* [13], and the most recent one is of Frumkin and Shamir [12]. 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 (as in [13]) but it requires additional secret key material and universal hash functions to detect the attacks. In the follow-up work Bringer and Chabanne [4] proposed a new HB^+ variant (so-called Trusted-HB) that builds upon Krawczyk's hash-based authentication schemes using special LFSR constructions (via Toplitz matrix).

A novel authentication and forward private RFID protocol is proposed by Berbain *et al.* [3]. The protocol is using pseudo-random number generators and universal hash functions as basic building blocks, which makes it suitable for low-footprint solutions. The security of their scheme is proven in the standard model but it remains unclear whether it can withstand physical attacks (*i.e.* tampering with the tag, such that the tag can be cloned).

The main reason why most work focussed on symmetric-key solutions 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 [20, 14] consume less area than any known secure cryptographic hash function (*e.g.*, the candidate algorithms proposed in the SHA-3 competition [22]). One alternative is therefore, to pursue protocols that use only public-key cryptography. In [18], it is shown that some conventional public-key based authentication protocols, such as the Schnorr protocol [24] and the Okamoto protocol [23], do not resist tracking attacks. Accordingly, the EC-RAC (Elliptic Curve Based Randomized Access Control) protocol has been proposed to address the established privacy threat. However, in [6, 8], it is shown that EC-RAC is also vulnerable to tracking attacks and replay attacks, and in addition [6], the randomized Schnorr protocol has been proposed as an alternative for EC-RAC. This protocol is narrow-strong privacy preserving, but does not offer privacy protection against a wide-weak attacker. EC-RAC has been gradually revised in [19, 17]. However, Fan *et al.* [10] have shown that the most recent version of EC-RAC [17] is not wide-weak privacy preserving.

In addition, we also mention RFID authentication protocols that are based on Physical Unclonable Functions (PUFs) [26]. It was shown that those solutions can also prevent counterfeiting in on-line and off-line scenarios and are feasible for active tags. However, they require both private-key and public-key algorithms.

Note that in this paper, we only consider RFID authentication protocols on the logical level. Danev *et al.* [7] have shown that one can also identify RFID tags with a high accuracy from a small distance (*e.g.*, less than 1 meter), based on their physical-layer fingerprints. This technique automatically enables tag-to-server authentication. However the downside of this solution is the requirement that the distance between RFID tag and reader should be small, in order to have a high accuracy. On the other hand, allowing a large distance between reader and tag, as is the case for RFID authentication protocols on the logical level, gives more freedom to the attacker and hence makes him more powerful (*e.g.*, it becomes easier to carry out man-in-the-middle attacks).

In the next Section of this paper, we focus more in detail on authentication protocols based on public-key cryptography, more specifically on ECC.

3 ECC-based Untraceable RFID Authentication Protocols

3.1 System parameters

Table 1 shows the notation that is used in the rest of this paper. We denote P as the base point, and y and $Y(= yP)$ are the server's private-key and public-key pair, where yP denotes the point derived by the point multiplication operation on the Elliptic Curve group. x_1 and $X_1(= x_1P)$ are a tag's private-key and public-key pair. We will denote these values as the (secret) *tag's ID* and the *tag's ID-verifier* respectively. One should note, although the name suggests that it can be publicly known, that the public-key of the tag (i.e. the ID-verifier) should be kept secret in the server. Revealing this key causes tracking attacks.

Table 1. System Parameters

System Parameters	y : Server's private-key $Y(= yP)$: Server's public-key x_1 : Tag's ID x_2 : Tag's password (Pwd) $X_1(= x_1P)$: Tag's ID-verifier $X_2(= x_2P)$: Tag's Pwd-verifier P : Base point in the EC group n : Prime order of P
ID-transfer	y, X_1, P, n (Server) x_1, Y, P, n (Tag)
ID&Pwd-Transfer,	y, X_1, x_1, X_2, P, n (Server) x_1, x_2, Y, P, n (Tag)

3.2 Narrow vs Wide Privacy

Several solutions using public-key algorithms have been proposed in order to protect RFID tags from tracking attacks. Since they are only narrow-strong privacy-preserving, they are all vulnerable to man-in-the-middle attacks carried out by a wide attacker. Let us illustrate this with the ID-transfer scheme of the revised EC-RAC protocol [19], which is shown in Fig. 1.

Deursen and Radomirović [9] demonstrated a man-in-the-middle attack on this scheme in [9], as shown in Fig. 2. The attack is performed by manipulating

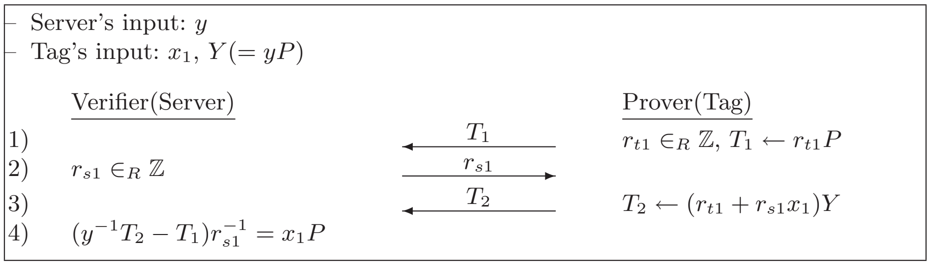
– Server's input: y
– Tag's input: x_1, $Y (= yP)$
Verifier(Server)
Prover(Tag)
1)
T_1
$r_{t1} \in_R \mathbb{Z}$, $T_1 \leftarrow r_{t1}P$
2)
$r_{s1} \in_R \mathbb{Z}$
r_{s1}
3)
T_2
$T_2 \leftarrow (r_{t1} + r_{s1}x_1)Y$
4)
$(y^{-1}T_2 - T_1)r_{s1}^{-1} = x_1P$

Fig. 1. ID-Transfer Scheme [19].

messages exchanged in previous protocol instances. A similar problem arises in the randomized Schnorr protocol [6] and the password-transfer scheme of the most recent version of EC-RAC [17]. No solution founded on public-key cryptography had yet been proposed that is both narrow-strong and wide-weak privacy preserving.

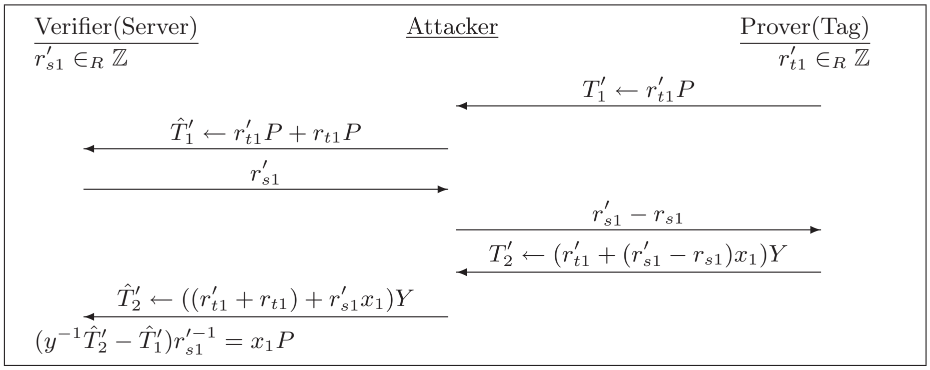
Verifier(Server)
Attacker
Prover(Tag)
$r'_{s1} \in_R \mathbb{Z}$
$r'_{t1} \in_R \mathbb{Z}$
$T'_1 \leftarrow r'_{t1}P$
$\hat{T}'_1 \leftarrow r'_{t1}P + r_{t1}P$
r'_{s1}
$r'_{s1} - r_{s1}$
$T'_2 \leftarrow (r'_{t1} + (r'_{s1} - r_{s1})x_1)Y$
$\hat{T}'_2 \leftarrow ((r'_{t1} + r_{t1}) + r'_{s1}x_1)Y$
$(y^{-1}\hat{T}'_2 - \hat{T}'_1)r'^{-1}_{s1} = x_1P$

Fig. 2. Illustration of a Man-in-the-Middle Attack on the Revised EC-RAC [9].

3.3 New ID-Transfer Scheme

In this paper, we present two RFID authentication protocols which are both narrow-strong and wide-weak privacy preserving. The first authentication protocol is an ID-transfer scheme, which allows the tag to demonstrate its knowledge of its secret ID. The second authentication protocol combines two sub-modules: the ID-transfer scheme and a password-transfer (shortly, Pwd-transfer) scheme. Both RFID authentication protocols fulfill a specific set of privacy and security requirements.

Protocol Description To prevent man-in-the-middle attacks carried out by a wide attacker, one can use a cryptographic hash function to introduce nonlinearity, as noted in [9]. However, this requires additional hardware to implement the cryptographic hash function, which is undesirable due to the limited hardware resources of a tag. To avoid this, we suggest to introduce the required nonlinearity by reusing EC-operations. Our proposed ID-transfer scheme is shown in Fig. 3, where $\dot{r}_{s1} = x(r_{s1}P)$ denotes the x-coordinate of $r_{s1}P$. To ensure valid authentication claims, the value r_{s1} should be different from zero and the order of P on the elliptic curve. The computation of the x-coordinate of $r_{s1}P$ only introduces a slight increase in the cost: the server and the tag need to perform one extra EC point multiplication.

- Server's input: y
- Tag's input: x_1, $Y(= yP)$

	Verifier(Server)		Prover(Tag)
1)		$\xleftarrow{\quad T_1 \quad}$	$r_{t1} \in_R \mathbb{Z}$, $T_1 \leftarrow r_{t1}P$
2)	$r_{s1} \in_R \mathbb{Z}$	$\xrightarrow{\quad r_{s1} \quad}$	
3)	$\dot{r}_{s1} \leftarrow x(r_{s1}P)$		$\dot{r}_{s1} \leftarrow x(r_{s1}P)$
4)		$\xleftarrow{\quad T_2 \quad}$	$T_2 \leftarrow (r_{t1} + \dot{r}_{s1}x_1)Y$
5)	$(y^{-1}T_2 - T_1)\dot{r}_{s1}^{-1} = x_1P$		

Fig. 3. ID-Transfer Scheme Resistant to Man-in-the-Middle Attacks (Protocol 1).

Protocol Analysis We analyze our ID-transfer scheme in two phases: first the security analysis and then the privacy analysis. The security analysis is performed by reducing the proposed protocol to the Schnorr protocol. Reducing a protocol means that we modify a protocol to give an attacker more adversarial power (or more information). Therefore, the original protocol will be at least as secure as the reduced protocol (shown in Fig. 4). Since the security of the Schnorr protocol is proven in [2], the reduction concludes the proof. For the privacy analysis, we first show its narrow-strong privacy and then demonstrate that the protocol also offers privacy protection against a wide-weak attacker.

• **Security Analysis:** We modify the proposed protocol such that the server transmits the following values in Steps 2) and 3) in Fig. 3.

$$r_{s1},\ \dot{r}_{s1} \tag{1}$$

Since the mapping from r_{s1} to $\dot{r}_{s1}$ (the x-coordinate of $r_{s1}P$) is deterministic, even if the server transmits both the values r_{s1} and $\dot{r}_{s1}$ to a tag, the protocol derived is equivalent to the former one.

Now we reduce the protocol by dropping r_{s1}, so the server only transmits $\dot{r}_{s1}$ (as is shown in Step 3 of Fig. 4). Since r_{s1} is only used to derive $\dot{r}_{s1}$, $\dot{r}_{s1}$ is sufficient information for the tag to produce a response. However, by dropping r_{s1}, an attacker gets more freedom to manipulate $\dot{r}_{s1}$, since he does not need to derive it from r_{s1}. In other words, in this case a tag does no longer know if the received challenge is an actual output of the one-way function of the EC point multiplication.

	Verifier(Server)		Prover(Tag)
1)		$\xleftarrow{\quad T_1 \quad}$	$r_{t1} \in_R \mathbb{Z}$, $T_1 \leftarrow r_{t1}P$
2)	$r_{s1} \in_R \mathbb{Z}$		
3)	$\dot{r}_{s1} \leftarrow x(r_{s1}P)$	$\xrightarrow{\quad \dot{r}_{s1} \quad}$	
4)	$T_2 \leftarrow vY$	$\xleftarrow{\quad v \quad}$	$v \leftarrow r_{t1} + \dot{r}_{s1}x_1$
5)	$(y^{-1}T_2 - T_1)\dot{r}_{s1}^{-1} = x_1P$		

Fig. 4. Reduced Scheme from Fig. 3.

Another reduction is performed in Step 4. A tag transmits $v(= r_{t1} + \dot{r}_{s1}x_1)$ instead of $T_2(= (r_{t1} + \dot{r}_{s1}x_1)Y)$. Since given v and Y, T_2 can be easily computed, an attacker gets extra information by eavesdropping v (instead of T_2) in this reduced protocol.

The reductions described above result in a reduced protocol (Fig. 4) where the exchanged messages are equivalent to the Schnorr protocol. Hence, one can conclude that our proposed protocol can be reduced to the Schnorr Protocol.

• **Narrow-Strong Privacy:** This proof can be done similarly to the proof in [19]. The three messages exchanged in the protocol are:

$$r_{t1}P,\ r_{s1},\ (r_{t1} + \dot{r}_{s1}x_1)Y \tag{2}$$

$r_{t1}P$ is a random point generated by a tag, and r_{s1} a random value that is possibly controlled by an attacker. These two messages themselves include no information about a tag. The last message can be considered as an addition of two EC points as follows:

$$(r_{t1} + \dot{r}_{s1}x_1)Y = r_{t1}yP + \dot{r}_{s1}x_1yP \tag{3}$$

Assuming that the Decisional Diffie-Hellman problem is hard, the first point $r_{t1}yP$ is a random secret shared between the server and a tag upon the transmission of $r_{t1}P$. Therefore, the EC point addition can be considered as a one-time pad with a one-time secret key $r_{t1}yP$, which means that $(r_{t1}+\dot{r}_{s1}x_1)Y$ is nothing more than a random point for an attacker. Note that there is no effect from r_{s1} on the one-time pad, which is the only message that could possibly be controlled by an attacker. Therefore, the proposed protocol is narrow privacy-preserving.

Another thing we can note is that the secret of the one-time pad, $r_{t1}yP$, does not include any information about a tag. It only contains the public key of the server and random data which is unknown to the attacker. It does not depend on the identity of the tag. Therefore, even if an attacker knows the secret key of a tag, x_1, it doesn't help for interpreting the encrypted message. So, the protocol is narrow-strong privacy-preserving.

• **Wide-Weak Privacy:** For a wide attacker, there is one-bit extra information compared to a narrow attacker: the decision of the server whether to accept a tag or not. This extra bit of information can be used by a wide-weak attacker to perform a tracking attack. The goal of this attacker is to determine if two sets of protocol instances originate from the same tag. One of these sets contains authentic messages from the past. Let us denote the source (*i.e.* the tag) of these messages by A. The other set contains the responses of a tag B. The tracking attack is successful when the attacker can determine the (in)equality of the two tags A and B with a probability significantly larger than $\frac{1}{2}$.

This (in)equality can be checked by verifying if both protocol instances use the same secret value x_1 (this is the only value used in the protocol which is tag-dependent). This value is exclusively used to compute T_2. The message T_1 only depends on a random number r_{t1} generated by the tag. Note that a wide-weak attacker does not know the secret x_1 and the random values r_{t1}. Since the decisional Diffie-Hellman problem is assumed to be hard, the attacker cannot extract the value x_1 out of the protocol message T_2. The only strategy that an attacker can carry out, is construct a message pair (T_1', T_2'), using messages $(T_{1,i}, T_{2,i})$ [1], in such a way that T_2' will only be accepted by the server if tag A equals tag B (*i.e.* if the same secret value x_1 is used in both sets of protocol instances).

Without loss of generality, let us assume that tag A equals tag B. When carrying out the ID-transfer scheme, the server will send the challenge r'_{s1}, and receive the messages (T_1', T_2') from the attacker. It will accept these messages if the following equation hold:

$$T_2' = yT_1' + \dot{r}'{}_{s1} x_1 Y \tag{4}$$

Note that the attacker does not know the secret key y. However, the attacker can exploit the linear property of addition on an elliptic curve to construct a valid pair (T_1', T_2'). The attacker first chooses a linear function $f()$ and computes T_1' as follows:

$$T_1' = f(\bigcup_i (T_{1,i})) \tag{5}$$

In the equation above, $\bigcup_i (T_{1,i})$ denotes a cluster of messages $T_{1,i}$, selected by the attacker, from both sets of protocol instances. Next, the attacker can compute T_2' as follows:

$$T_2' = f(\bigcup_i (T_{2,i})) \tag{6}$$

In the equation above, $\bigcup_i (T_{2,i})$ denotes a cluster of messages $T_{2,i}$, selected by the attacker, from both sets of protocol instances. Note that $T_{1,i}$ and $T_{2,i}$ have

[1] The index i denotes that the cluster of messages can originate from both sets of protocol instances.

to originate from the same protocol instance. *I.e.*, the following relation holds:

$$T_{2,i} = yT_{1,i} + (\dot{r}_{s1,i}x_1)Y \tag{7}$$

When combining Eqs. (5), (6) and (7), one can notice that the first term of Eq. (4) will always be equal to yT'_1 due to the linear property of the function $f()$. The second term in the addition is also correct if the following equation holds:

$$\dot{r'}_{s1} = f(\bigcup_i(\dot{r}_{s1,i})) \tag{8}$$

Since the attacker has to send the message T'_1 to the server before it receives the challenge r'_{s1}, the attacker has to select the set of protocol instances of tag A and the function $f()$ in advance. After having received the challenge, the attacker can only control the challenge r_{s1} that it sends to tag B. The attacker hence has to select a challenge r_{s1} such that Eq. (8) holds. However, since point multiplication on an elliptic curve is assumed to be a one-way function, an arbitrary control of $x(r_{s1}P)$ is infeasible. As a result, an attacker cannot construct the message pair (T'_1, T'_2) using Eq. (5) and Eq. (6). Note that when a non-linear function $f()$ would be used, the first term of Eq. (4) never holds, and the attack will hence not work.

Since a wide-weak attacker cannot carry out the tracking attack described above, the ID-transfer scheme (Protocol 1) is wide-weak privacy-preserving.

3.4 New Pwd-Transfer Scheme

After the ID-transfer scheme, one can carry out a Pwd-transfer scheme. This offers increased security protection (we will come back to this issue later in the paper). By performing the ID-transfer scheme, the server will obtain the ID-verifier X_1. Using this verifier, the server can look up the tag's information (x_1 and X_2) in a local database. We hence assume that the server knows x_1 and X_2 during the execution of the Pwd-transfer scheme.

Protocol Description Let us first focus on the Pwd-transfer scheme itself. Its design concept is completely equivalent to the ID-transfer scheme, as is shown in Fig. 5. After generating r_{t1} and T_1, a tag transmits T_1 to the server. Then, the server responds with a random challenge r_{s1} (not equal to zero or the order of P on the elliptic curve), which is used to derive $\dot{r}_{s1}$. Finally, after having received the message T_2 from a tag, the server derives $X_2(= x_2P)$ and verifies it by comparing it with the stored Pwd-verifier in the database.

Protocol Analysis If one compares the Pwd-transfer scheme and the ID-transfer scheme, one can notice that the only difference is the message T_2, where $(r_{t1} + \dot{r}_{s1}x_1x_2)Y$ is used instead of $(r_{t1} + \dot{r}_{s1}x_1)Y$. In this message, the secret identity x_1 is used to mask the secret password x_2. One can represent T_2 as follows:

- Server's input: y, x_1, $X_2(= x_2P)$
- Tag's input: x_1, x_2, $Y(= yP)$

	Verifier(Server)		Prover(Tag)
1)		$\xleftarrow{\quad T_1 \quad}$	$r_{t1} \in_R \mathbb{Z}$, $T_1 \leftarrow r_{t1}P$
2)	$r_{s1} \in_R \mathbb{Z}$	$\xrightarrow{\quad r_{s1} \quad}$	
3)	$\dot{r}_{s1} \leftarrow x(r_{s1}P)$		$\dot{r}_{s1} \leftarrow x(r_{s1}P)$
4)		$\xleftarrow{\quad T_2 \quad}$	$T_2 \leftarrow (r_{t1} + \dot{r}_{s1}x_1x_2)Y$
5)	$(y^{-1}T_2 - T_1)x_1^{-1}\dot{r}_{s1}^{-1} = x_2P$		

Fig. 5. Pwd-Transfer Scheme Resistant to Man-in-the-Middle Attacks.

$$(r_{t1} + \dot{r}_{s1}x_1x_2)Y = (r_{t1} + \dot{r}_{s1}x_3)Y \tag{9}$$

Since the secret ID x_1 and the secret password x_2 are two independent numbers, their product can be substituted by the secret value x_3. The Pwd-transfer scheme can hence be considered as an ID-transfer scheme with secret identity x_3. As a result, the Pwd-transfer scheme is completely equivalent to the ID-transfer scheme. Therefore, the Pwd-transfer scheme has the same security and privacy properties as the ID-transfer scheme: it is as least as secure as the Schnorr protocol, and is both narrow-strong and wide-weak privacy-preserving.

3.5 ID&Pwd-Transfer Scheme

As described above, it is interesting to combine the ID-transfer scheme with the Pwd-transfer scheme. If only the ID-transfer scheme is used for authentication, the security level could be reduced if the number of tags is extremely large. Since the authentication is performed by checking the existence of a derived ID-verifier in the server's database, the probability that an attacker randomly generates an ID that also appears in the server's database (and hence will be accepted by the server during the protocol) increases when the number of tags grows. In applications where this would cause security problems, one can use an RFID authentication protocol that combines the ID-transfer scheme with the Pwd-transfer scheme. We will now discuss this more in detail.

Protocol Description The proposed ID-transfer scheme (Fig. 3) and Pwd-transfer scheme (Fig. 5) can be combined in two different ways: Fig. 6 (vulnerable to tracking attacks) and Fig. 7 (Protocol 2).

Security and Privacy Analysis Let us now analyze both combinations. In the protocol shown in Fig. 6, the same random number r_{t1} is used for both the ID-transfer scheme and the Pwd-transfer scheme. While this reduces the computation load in a tag, this also causes a vulnerability to tracking attacks. An eavesdropper can track the tag by observing the exchanged messages. This can be seen in the following computation:

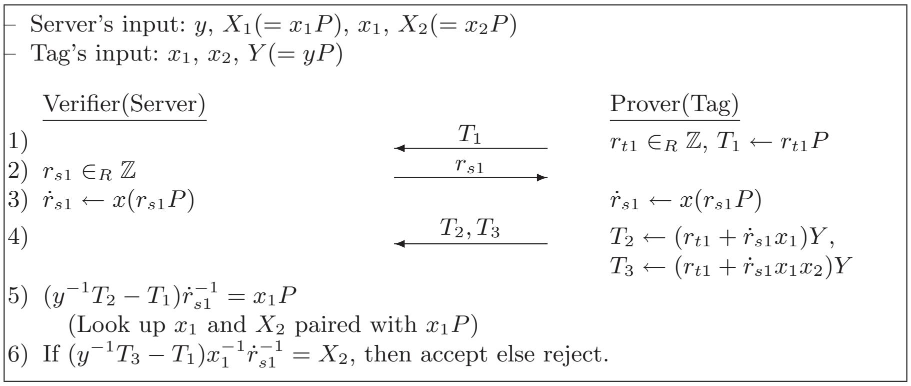
– Server's input: y, $X_1(=x_1P)$, x_1, $X_2(=x_2P)$
– Tag's input: x_1, x_2, $Y(=yP)$
Verifier(Server)
Prover(Tag)
1) T_1 $r_{t1} \in_R \mathbb{Z}$, $T_1 \leftarrow r_{t1}P$
2) $r_{s1} \in_R \mathbb{Z}$ r_{s1}
3) $\dot{r}_{s1} \leftarrow x(r_{s1}P)$ $\dot{r}_{s1} \leftarrow x(r_{s1}P)$
4) T_2, T_3 $T_2 \leftarrow (r_{t1} + \dot{r}_{s1}x_1)Y$, $T_3 \leftarrow (r_{t1} + \dot{r}_{s1}x_1x_2)Y$
5) $(y^{-1}T_2 - T_1)\dot{r}_{s1}^{-1} = x_1P$
(Look up x_1 and X_2 paired with x_1P)
6) If $(y^{-1}T_3 - T_1)x_1^{-1}\dot{r}_{s1}^{-1} = X_2$, then accept else reject.

Fig. 6. Authentication protocol vulnerable to tracking attacks

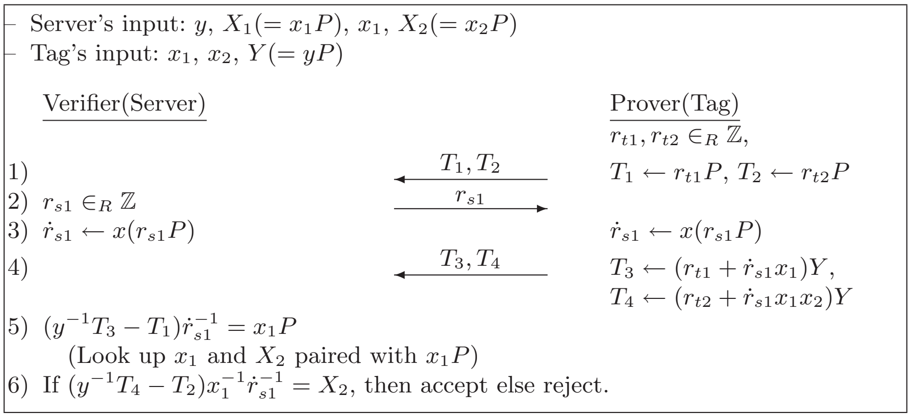
– Server's input: y, $X_1(=x_1P)$, x_1, $X_2(=x_2P)$
– Tag's input: x_1, x_2, $Y(=yP)$
Verifier(Server)
Prover(Tag)
$r_{t1}, r_{t2} \in_R \mathbb{Z}$,
1) T_1, T_2 $T_1 \leftarrow r_{t1}P$, $T_2 \leftarrow r_{t2}P$
2) $r_{s1} \in_R \mathbb{Z}$ r_{s1}
3) $\dot{r}_{s1} \leftarrow x(r_{s1}P)$ $\dot{r}_{s1} \leftarrow x(r_{s1}P)$
4) T_3, T_4 $T_3 \leftarrow (r_{t1} + \dot{r}_{s1}x_1)Y$, $T_4 \leftarrow (r_{t2} + \dot{r}_{s1}x_1x_2)Y$
5) $(y^{-1}T_3 - T_1)\dot{r}_{s1}^{-1} = x_1P$
(Look up x_1 and X_2 paired with x_1P)
6) If $(y^{-1}T_4 - T_2)x_1^{-1}\dot{r}_{s1}^{-1} = X_2$, then accept else reject.

Fig. 7. ID&Pwd-Transfer Scheme combined (Protocol 2)

$$
\begin{aligned}
&\dot{r}_{s1}^{-1}(T_2 - T_3) \\
&= \dot{r}_{s1}^{-1}((r_{t1} + \dot{r}_{s1}x_1)Y - (r_{t1} + \dot{r}_{s1}x_1x_2)Y) \\
&= \dot{r}_{s1}^{-1}(\dot{r}_{s1}x_1 - \dot{r}_{s1}x_1x_2)Y \\
&= (x_1 - x_1x_2)Y
\end{aligned} \tag{10}
$$

Since $(x_1 - x_1x_2)Y$ is a fixed value for a specific tag, it can be used to track a tag. This protocol does hence not provide any privacy protection.

To overcome this problem, one needs to use independent random numbers in the ID-transfer scheme and the Pwd-transfer scheme, as is shown in Fig. 7. Protocol 2 can be considered as two instances of the ID-transfer scheme which are executed in parallel. One protocol instance uses the secret ID x_1, the other one uses the secret ID $x_3 = (x_1x_2)$. Since x_2 is random and independent of the value x_1, and since r_{t1} and r_{t2} are two independent random values, both protocol instances are hence independent. They only use the same challenge r_{s1}. Note that the following two statements hold:

- The ID-transfer scheme can be reduced to the Schnorr protocol (as is demonstrated in Sect. 3.3). The former is hence at least as secure as the latter.
- The Schnorr protocol offers protection against active man-in-the-middle attacks, including the reuse of the same challenge in different protocol instances.

By combining these two findings, one can prove that protocol 2 inherits the security properties of the ID-transfer scheme (protocol 1).

The same argumentation can be used to prove the privacy properties of protocol 2. Both a narrow-strong and a wide-weak attacker can perform man-in-the-middle attacks, where the same challenge is sent to one particular tag in several different protocol instances. Since the ID-transfer scheme is narrow-strong and wide-weak privacy-preserving, the parallel execution of two protocol instances using the same challenge r_{s1} does not change its privacy properties. Protocol 2 is hence also narrow-strong and wide-weak privacy-preserving.

3.6 Hardware Realization

The two secure and privacy-preserving authentication protocols proposed in this paper rely exclusively on the use of Elliptic Curve Cryptography. They do not require other cryptographic building blocks. A hardware architecture that realizes the computation required in our RFID protocols is presented in [17]. The processor is composed of a micro controller, a bus manager and an EC processor (ECP). It has a power consumption of $13.8\mu W$ and it can complete any of the protocols in less than 500 ms. In addition, it can be produced with less than 15 Kgates. These performance results show the feasibility of the protocols proposed even for a passive tag and outperform other secure and private protocols proposed in the literature.

4 Conclusions

In this paper, we have addressed the risk of tracking attacks in RFID networks. We gave an overview of cryptographic authentication protocols which have been proposed so far, and discussed the public-key based techniques more in detail. We proposed two new authentication protocols that are exclusively based on the use of Elliptic Curve Cryptography. Both RFID authentication protocols are narrow-strong and wide-weak privacy preserving. To the best of our knowledge, our protocols are the first ECC-based authentication protocols which offer privacy protection against a wide-weak attacker. Each of the protocols has different computational demands and accordingly different security features. Compared to other RFID schemes proposed in the literature, our protocols remain lightweight in terms of area and computation time, while still achieving the required security and privacy properties.

Acknowledgments

This work was partially supported by the US National Science Foundation CCF-0541472, by the IAP Programme P6/26 BCRYPT of the Belgian State (Belgian Science Policy), by K.U. Leuven-BOF (OT/06/40), by the FWO project G.0300.07, and by the Flemish IBBT projects.

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 *Advances in Cryptology - CRYPTO'02, Lecture Notes in Computer Science*, volume 2442, pages 162–177. Springer-Verlag, 2002.
3. C. Berbain, O. Billet, J. Etrog, and H. Gilbert. An efficient forward private RFID protocol. In *Proceedings of the 16th ACM conference on Computer and communications security (CCS '09)*, pages 43–53. ACM, 2009.
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*, 54(9):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*, 2006.
6. J. Bringer, H. Chabannel, 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*. Springer-Verlag, 2008.
7. B. Danev, T. S. Heydt-Benjamin, and S. Čapkun. Physical-layer Identification of RFID Devices. In *Proceedings of the 18th USENIX Security Symposium (USENIX Security '09)*, pages 125–136. USENIX, 2009.
8. T. Deursen and S. Radomirović. Attacks on RFID Protocols. In *Cryptology ePrint Archive: listing for 2008 (2008/310)*, 2008.
9. T. Deursen and S. Radomirović. Untraceable RFID protocols are not trivially composable: Attacks on the revision of EC-RAC. In *Cryptology ePrint Archive: Report 2009/332*, 2009.
10. J. Fan, J. Hermans, and F. Vercauteren On the Claimed Privacy of EC-RAC III. Cryptology ePrint Archive, Report 2010/132, 2010. `http://eprint.iacr.org/`.
11. 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.
12. D. Frumkin and A. Shamir. Un-Trusted-HB: Security Vulnerabilities of Trusted-HB. In *Proceedings of RFIDSec09*, Leuven, Belgium, 2009.
13. H. Gilbert, M. Robshaw, and H. Sibert. An active attack against HB^+ - a provably secure lightweight authentication protocol. *IEE processing letters*, 41(21):1169–1170, 2005.
14. D. Hein, J. Wolkerstorfer, and N. Felber. ECC is Ready for RFID - A Proof in Silicon. In *Selected Areas in Cryptography, Lecture Notes in Computer Science*, volume 5381, pages 401–413, 2009.

15. A. Juels and S. Weis. Defining Strong Privacy for RFID. Cryptology ePrint Archive, Report 2006/137, 2006. `http://eprint.iacr.org/`.
16. A. Juels and S. Weis. Authenticating pervasive devices with human protocols. In *In Proc. of CRYPTO'05, volume 3126 of LNCS*, pages 293–308. IACR, Springer-Verlag, 2005.
17. Y. K. Lee, L. Batina, D. Singelée, and I. Verbauwhede. Low-Cost Untraceable Authentication Protocols for RFID. In *ACM Conference on Wireless Network Security - WiSec '10*. ACM, 2010.
18. Y. K. Lee, L. Batina, and I. Verbauwhede. EC-RAC (ECDLP Based Randomized Access Control): Provably Secure RFID authentication protocol. In *IEEE International Conference on RFID*, pages 97–104. IEEE, 2008.
19. 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.
20. Y. K. Lee, K. Sakiyama, L. Batina, and I. Verbauwhede. Elliptic Curve Based Security Processor for RFID. *IEEE Transactions on Computer*, 57(11):1514–1527, November 2008.
21. C. 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.
22. NIST National Institute of Standards and Technology. Cryptographic Hash Algorithm Competition. `http://csrc.nist.gov/groups/ST/hash/sha-3/index.html`.
23. T. Okamoto. Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. In E. F. Brickell, editor, *Advances in Cryptology - CRYPTO'92, Lecture Notes in Computer Science*, volume 740, pages 31–53. Springer-Verlag, 1992.
24. 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.
25. B. Toiruul and K. Lee. An Advanced Mutual-Authentication Algorithm Using AES for RFID Systems. *International Journal of Computer Science and Network Security*, 6(9B), September 2006.
26. P. Tuyls and L. Batina. RFID-tags for Anti-Counterfeiting. In D. Pointcheval, editor, *In Topics in Cryptology - CT-RSA - The Cryptographers' Track at the RSA Conference*, number 3860 in Lecture Notes in Computer Science, pages 115–131, San Jose, USA, February 13-17 2006. Springer Verlag.
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.