Low-Cost Untraceable Authentication Protocols for RFID

[Extended and corrected version] *

Yong Ki Lee
EE – EmSec
University of California
Los Angeles, CA, USA
yklee93@kg21.net

Lejla Batina
Digital Security group
Radboud University Nijmegen
Nijmegen, The Netherlands
lejla@cs.ru.nl

Dave Singelée
IBBT – COSIC
Katholieke Universiteit Leuven
Heverlee, Belgium
dsingele@esat.kuleuven.be

Ingrid Verbauwhede
IBBT – COSIC / EE – EmSec
K.U.Leuven / UCLA
Heverlee, Belgium
iverbauw@esat.kuleuven.be

ABSTRACT

The emergence of pervasive computing devices has raised several privacy issues. In this paper, we address the risk of tracking attacks in RFID networks. Our contribution is threefold: (1) We repair the revised EC-RAC protocols of Lee, Batina and Verbauwhede and show that our improved authentication protocols are both narrow-strong and wide-weak privacy-preserving; (2) We present the search protocol, a novel scheme which allows for privately querying a particular tag, and proof its security properties; and (3) We design a hardware architecture to demonstrate the implementation feasibility of our proposed solutions for a passive RFID tag. Due to the specific design of our authentication protocols, they can be realized with an area significantly smaller than other RFID schemes proposed in the literature, while still achieving the required security and privacy properties.

Categories and Subject Descriptors

B.7.1 [**Integrated Circuits**]: Types and Design Styles—*Algorithms Implemented in Hardware*
; E.3 [**Data Encryption**]: [Public Key Cryptosystems]

General Terms

Algorithms, Security

Keywords

Authentication Protocol, Search Protocol, Privacy, Tracking Attack, Elliptic Curve Cryptography, RFID

*This version contains extended and corrected privacy proofs and repairs the privacy issues identified in [11].

WiSec'10, March 22–24, 2010, Hoboken, New Jersey, USA.

1. INTRODUCTION

RFID (Radio Frequency Identification) systems are rapidly expanding their applications to many areas: supply chains, access control, health care, road pricing, etc. However, due to the wide spread of tags and its cheap implementations, these applications have the potential to cause security and privacy risks at a tag carrier. Especially, the privacy of RFID tags has been a hot issue recently [25, 26, 28]. Unfortunately, privacy features have initially been close to non-existent in the design of conventional authentication systems and therefore there is a lot of work to be done in the context of RFID applications.

Privacy should be clearly distinguished from security: while *security* addresses the soundness of a protocol, *privacy* addresses the resistance against unauthorized identification, tracking or linking tags. Privacy can be considered in two concepts: *anonymity* in which the real ID of a tag must be unknown, and *untraceability* in which the (in)equality of two tags must be impossible to determine. Therefore, untraceability is a stronger privacy requirement than anonymity. In the rest of this article, we will denote a scheme as privacy-preserving when it achieves untraceability. Let us illustrate the difference between security and privacy with a practical example. The Schnorr protocol [27] is a well-known authentication protocol whose security properties can be formally proven [3]. However, it is not privacy-preserving because a tag (prover) can be traced by an eavesdropper as shown in [20].

Several theoretical models to address the privacy of RFID systems have been proposed in the literature [2, 18, 24, 30]. To define privacy in this paper, we import 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 (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.

In this paper, we present two RFID authentication protocols, which are a revision of the protocols of Lee, Batina

and Verbauwhede [21]. Each protocol has different security characteristics and computational workload. These improved authentication protocols are both narrow-strong and wide-weak privacy-preserving. We also present the search protocol, a novel scheme in which a server (or a reader) can efficiently query for a specific tag, without compromising privacy. Moreover, we present a hardware architecture that can realize the proposed protocols. Its performance results show that such schemes are feasible, even for a passive tag, and outperforms other secure and privacy-preserving protocols proposed in the literature.

The remainder of the paper is organized as follows. In Section 2, related work is reviewed. We propose our authentication protocols and a search protocol in Section 3 and 4, respectively. We show the implementation results of the proposed protocols for a particularly designed hardware architecture in Section 5. We conclude our paper in Section 6.

2. STATE OF THE ART

Recently, several solutions using public-key algorithms have been proposed in order to protect RFID tags from tracking attacks. In [20], it is shown that some conventional public-key based authentication protocols, such as the Schnorr protocol and the Okamoto protocol, do not resist tracking attacks. Accordingly, EC-RAC (Elliptic Curve Based Randomized Access Control) protocols have been proposed in the same paper to address the established vulnerability. However, in [6, 8], it is shown that EC-RAC is also vulnerable to tracking attacks and replay attacks, and in addition, the randomized Schnorr protocol has been proposed as an alternative of EC-RAC in [6]. Furthermore, Lee, Batina and Verbauwhede presented the revised EC-RAC protocols in [21].

However, as also discussed in [9], one can demonstrate that both the randomized Schnorr protocol and the revised EC-RAC are narrow-strong privacy preserving, but not wide-weak privacy-preserving.

There were many attempts to obtain authentication protocols for RFID tags by means of symmetric-key primitives [4, 12, 13, 29]. Engberg et al. [10] proposed a zero-knowledge authentication protocol for RFID tags which employs symmetric operations such as an XOR and cryptographic hash functions. Of other notable solutions for authentication protocols, we mention here the HB^+ protocol [19] that was presented as an extremely cheap solution (especially in hardware) but still secure against active adversaries. The HB^+ protocol is based on the work of Hopper and Blum (HB) [17]. However, many attacks followed, such as [15], and the most recent one is of Frumkin and Shamir [14]. Several fixes have been proposed, such as the HB^{++} protocol from Bringer et al. [5]. HB^{++} is claimed to be secure against man-in-the-middle attacks (as in [15]) but it requires additional secret key material and universal hash functions to detect the attacks.

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).

The solutions we will propose in this paper rely exclusively on public-key cryptography. In particular we use ECC (Elliptic Curve Cryptography) [23], since it reduces the length of certificates (compared to conventional public-key cryptographic techniques) and it has been shown that it can be realized on extremely low-cost platforms such as RFID tags and sensor nodes [22, 16].

Let us introduce some notation. 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 should be kept secret in the server. Revealing this key causes tracking attacks.

2.1 Revised EC-RAC Protocol

In the revised EC-RAC protocols [21], which solve the weaknesses of the original EC-RAC protocol proposed in [20], the tag-to-server authentication (a tag proves its authenticity to a server) is composed of the ID-transfer scheme and the password-transfer scheme. The two schemes are based on the same design concept, and therefore, we introduce the ID-transfer scheme only.

2.1.1 ID-Transfer Scheme

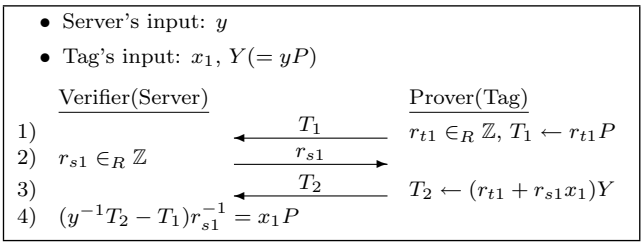

Figure 1: ID-Transfer Scheme [21].

The ID-transfer scheme of EC-RAC is shown in Fig. 1. In this scheme, a tag generates a random number r_{t1} and a point T_1, and transfers T_1 to the server. Then, the server responds with a random challenge r_{s1}, and a tag produces and transfers another point T_2 to the server. After receiving T_2, the server calculates a tag's ID-verifier $x_1P(= X_1)$, which is used to check whether the corresponding tag is registered in the server.

2.1.2 Security Analysis

In [21], the security proof of the ID-transfer scheme is done by reducing it to well-known hard cryptographic problems. In order to show the security against replay attacks, the ID-transfer scheme is reduced to the Schnorr protocol, and in order to show the resistance against tracking attacks, it is reduced to the Decisional Diffie-Hellman (DDH) problem. However, in the attacker's model, an attacker's ability

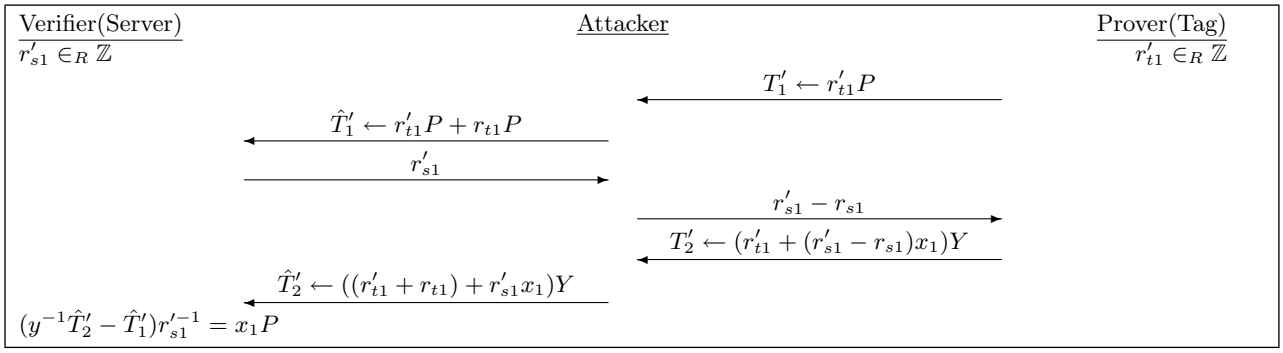

Figure 2: Man-in-the-Middle Attack on the ID-Transfer Scheme [9].

is limited to observing the exchanged messages and forging messages to impersonate a server or a tag. In other words, an attacker does not know when the server accepts or rejects (forced) messages (i.e. the attacker is assumed to be narrow).

In [9], Deursen and Radomirović demonstrate that man-in-the-middle attacks can be carried out on the EC-RAC protocols when an attacker is wide, as shown in Fig. 2. An attacker can utilize a set of messages $\{T_1(= r_{t1}P),\ r_{s1},\ T_2(= (r_{t1} + r_{s1}x_1)Y))\}$ exchanged in a previous protocol instance to generate the new set of messages $\hat{T}'_1(= T_1 + T'_1)$, $r'_{s1} - r_{s1}$ and $\hat{T}'_2(= T_2 + T'_2)$. By checking whether the server accepts the forged messages, an attacker can find out whether the currently communicating tag is equal to the tag that generated the messages that were eavesdropped during the previous protocol instance. As a result, a tag can be traced by a wide attacker.

2.2 Randomized Schnorr Protocol

2.2.1 Protocol Description

Another solution suggested to prevent tracking attacks is the randomized Schnorr protocol [6], which was proposed by Bringer et al. The protocol is illustrated in Fig. 3.

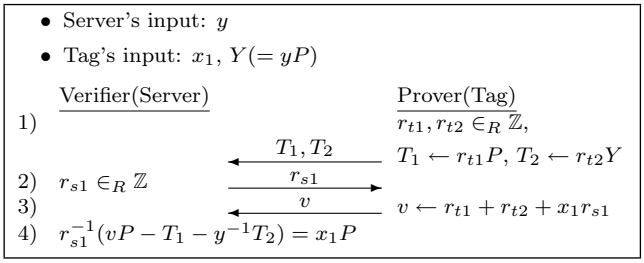

Figure 3: Randomized Schnorr Protocol [6].

In this protocol, a tag generates two random numbers r_{t1} and r_{t2}, and computes and transmits the two corresponding messages T_1 and T_2 to the server. After receiving a challenge r_{s1} from the server, a tag computes the authentication code v using its private-key x_1 and the random numbers r_{t1}, r_{t2} and r_{s1}. The value v is then sent to the server. Then, the server derives the tag's ID-verifier (x_1P) and checks if it is registered in the server.

2.2.2 Security Analysis

The randomized Schnorr protocol has a similar problem as the ID-transfer scheme. A man-in-the-middle attack, carried out by a wide attacker, that allows to track the tag is shown in Fig. 4.

3. WIDE-WEAK UNTRACEABLE AUTHENTICATION PROTOCOLS

In this section, we present two improved RFID authentication protocols, which are both narrow-strong and wide-weak privacy-preserving. To the best of our knowledge, these protocols are the first ECC-based authentication protocols which offer privacy protection against a wide-weak attacker. We use similar design concepts and system settings as the EC-RAC protocols (see [21]). There are two sub-modules: the ID-transfer scheme and the password-transfer (shortly, Pwd-transfer) scheme. The ID-transfer scheme itself can be used as an authentication protocol, or it can be combined with the Pwd-transfer scheme to achieve better security properties. As shown in Table 1, the server and the tag store more information when the two schemes are combined (ID&Pwd-Transfer).

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, Search Protocol	y, X_1, x_1, X_2, P, n (Server) x_1, x_2, Y, P, n (Tag)
Attacker's storage	Y, P, n

3.1 New ID-Transfer Scheme

3.1.1 Protocol Description

The man-in-the-middle attack shown in Fig. 2 is performed by manipulating messages exchanged in previous

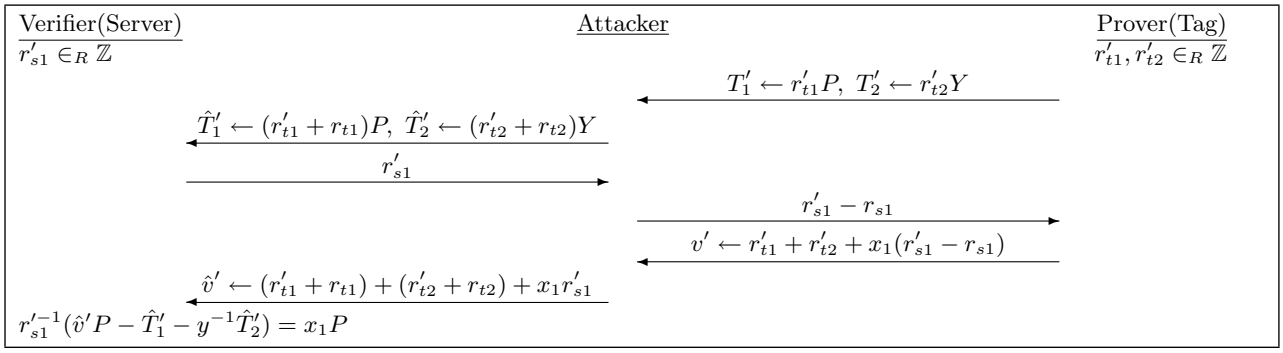

Figure 4: Man-in-the-Middle Attack on the Randomized Schnorr Protocol.

protocol instances. A possible solution to prevent this attack is to use a cryptographic hash function, 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 non-linearity by reusing EC-operations. Our solution is shown in Fig. 5, where $\dot{r}_{s1} = x(r_{s1}P)$ denotes the x-coordinate of $r_{s1}P$. This solution only introduces a slight increase in the cost: the server and the tag need to perform one extra EC point multiplication.

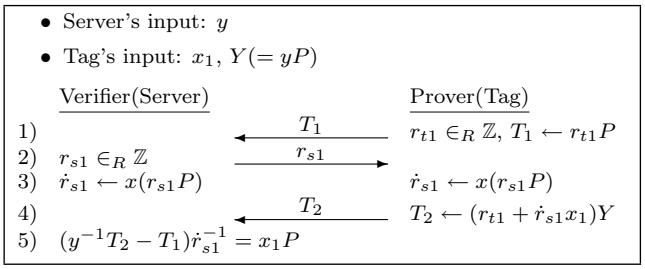

Figure 5: New ID-Transfer Scheme Resistant to Man-in-the-Middle Attacks (Protocol 1).

3.1.2 Protocol Analysis

The proposed ID-transfer scheme is analyzed 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. 6). Since the security of the Schnorr protocol is proven in [3], 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. 5.

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

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. 6). 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.

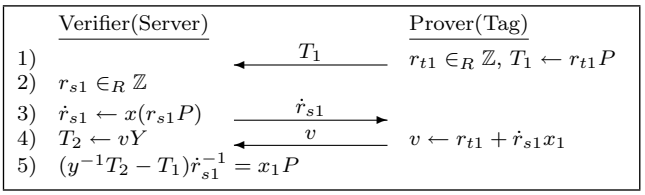

Figure 6: Reduced Scheme from Fig. 5.

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. 6) 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 [21]. 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:

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

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:

$$T_2' = yT_1' + \dot{r}'_{s1}x_1Y \tag{4}$$

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 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 equals yT_1', 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.

3.2 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.

3.2.1 Protocol Description

The design concept of the Pwd-transfer scheme is similar to the ID-transfer scheme as shown in Fig. 7. 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}, which is used to derive $\dot{r}_{s1}$. Finally, using the received 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.

3.2.2 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:

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

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

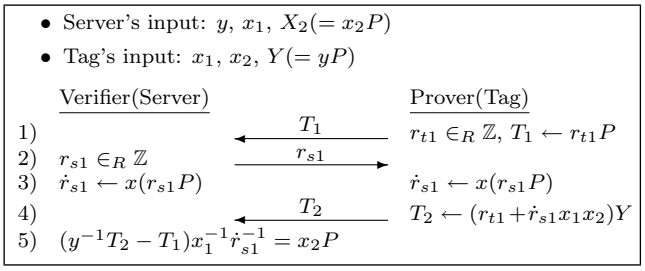

Figure 7: New Pwd-Transfer Scheme Resistant to Man-in-the-Middle Attacks.

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.

3.3 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.

3.3.1 Protocol Description

The proposed ID-transfer scheme (Fig. 5) and Pwd-transfer scheme (Fig. 7) can be combined in two different ways: Fig. 8 (vulnerable to tracking attacks) and Fig. 9 (Protocol 2).

3.3.2 Security Analysis

Let us now analyze both combinations. In the protocol shown in Fig. 8, 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:

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

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. 9. 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. 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.

4. SEARCH PROTOCOL DESIGN

The search protocol for an RFID system aims to find a specific tag in a pool of many tags. If one of the secure authentication protocols presented in this paper is used to search for a specific tag, the server must authenticate each tag one by one in a random order. In this case, the computation complexity will increase linearly with the number of the tags. Suppose we have a large library where each book is equipped with a tag. A book can be easily misplaced by any chance (e.g., because of a visitor's negligence or a librarian's mistake). If we just use a randomized authentication protocol to find a specific book, the server should authenticate half of the books in the library on average. Therefore, designing an efficient, secure search protocol is essential in an RFID system.

In an efficient search protocol, the server would expect to only receive a response from a designated tag. Otherwise, the server should handle responses from multiple tags. On the other hand, a tag should not respond before properly authenticating the server since a query may not be from an authentic server, but from an attacker who wants to track the tag. Therefore, the protocol should be a one-round protocol, and a tag should authenticate the server without giving any challenge. Note that we should also consider the possibility of replay attacks, since an attacker can reuse messages from the past to force a specific tag sending responses. Moreover to prevent tracking attacks, the messages from the server should be only understandable to the designated tag.

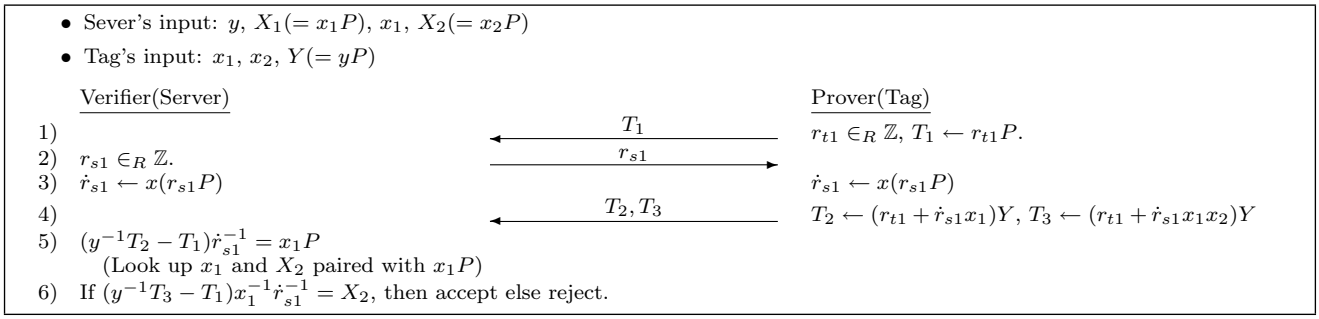

Figure 8: Authentication protocol vulnerable to tracking attacks

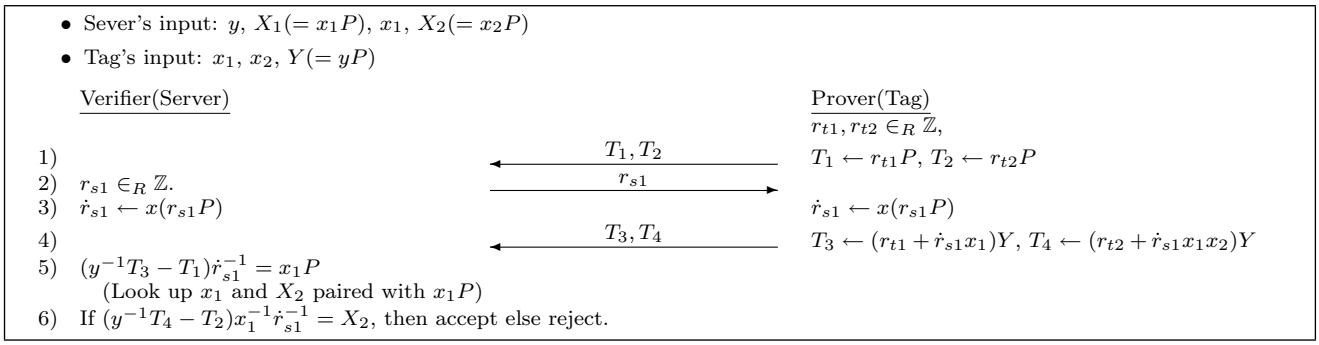

Figure 9: ID&Pwd-Transfer Scheme combined (Protocol 2)

There is a requirement and a limitation on the privacy in the search protocol. Since only the designated tag responds to the server, an attacker can know if the messages from the server is accepted by a tag (because in that case, the tag will send a response). Therefore, the protocol must be wide privacy-preserving. In addition, if an attacker knows the secret keys of a tag (i.e. the attacker is strong), he can interpret the server's messages as much as the tag itself, since the tag does not share any one-time session key (or secret) with the server. The attacker hence knows exactly which tag the server is looking for. From the moment he observes that a tag sends a response, he knows the requested tag is present. As a result, we will propose a one-round search protocol that is wide-weak privacy-preserving, but not narrow-strong. It remains an open problem if one can design a search protocol that achieves narrow-strong privacy without relaxing the efficiency requirements (such as having a one-round protocol).

Before discussing the design, let us first summarize the search protocol's requirements:

1. One-Round Authentication. The protocol should be completed in one round. Therefore, the server should generate messages without receiving a challenge from a tag.

2. Dedicated Authentication. Only the designated tag should be able to verify that the messages are generated by the server.

3. Security Against Replay Attacks.

4. Wide-Weak Privacy.

4.1 Protocol Description

In order to prevent replay attacks, the server should somehow utilize a challenge from a tag, which requires at least two rounds. So, we first design a two-round protocol and reduce it to a one-round protocol. A two-round protocol can be considered as a function $f(c)$ in the server, which outputs authentication messages using as input the challenge c (sent by the tag) as follows:

$$f(c) = \{rP, r(x_1 + c)x_2P\} \tag{11}$$

where r is a random number.

In order to reduce it to a one-round protocol, we change the protocol such that the server generates a challenge instead of receiving it from a tag. Therefore, the server will generate the following three messages and transmit.

$$\{c, rP, r(x_1 + c)x_2P\} \tag{12}$$

To prevent replay attacks, we need to make sure that c cannot be used twice. A tag can keep a counter and update it each time a valid message is received. This way, the received counter is always bigger than the stored one. The final search protocol is shown in Fig. 10.

After verifying the message from the server, a tag can respond to the server. Note that only the server can generate valid messages. In order to make sure that the proper tag is responding to the server, a tag-to-server authentication protocol should follow the search protocol. This will be further discussed in Sect. 4.3. The search protocol itself (without combining it with an authentication protocol) requires the server and a tag to perform two EC point multiplications each.

- **Server Input:** x_1, $X_2(= x_2P)$, c_s (server counter).
- **Tag Input:** x_1, x_2, c_t (tag counter).

A. Server → Tag (Message Generation): c_s, rP, $r(x_1 + c_s)X_2$

1. Increase the counter c_s.
2. Generate a random number r, $1 \leq r \leq n-1$.
3. Calculate $S_1 \leftarrow rP$ and $S_2 \leftarrow r \cdot (x_1 + c_s) \cdot X_2$.
4. Broadcast three messages: c_s, S_1, S_2.

B. Tag (Message Verification)

1. If $c_t \geq c_s$, then a tag ignores the messages and halts.
2. Verify whether S_1 has the desired prime order n. If $S_1 = O$ or $n \cdot S_1 \neq O$, then halts.
3. Otherwise, verify whether $S_2 = (x_1 + c_s) \cdot x_2 \cdot S_1$.
4. If S_2 is valid, then updates the counter as $c_t \leftarrow c_s$, and responds to the server.

Figure 10: The Search Protocol

Note that the search protocol has the same parameter settings as the ID&Pwd-transfer scheme (see Table 1).

4.2 Search Protocol Analysis

We prove that the proposed protocol satisfies all the four conditions for the search protocol.

4.2.1 One-Round Authentication

The search protocol is definitely a one-round authentication protocol. The server only sends a single (query) message to the tag. The authentication itself takes place during the tag-to-server authentication protocol.

4.2.2 Dedicated Authentication

Only the valid server can generate the messages since it requires x_1 and X_2, and only a specific tag can verify them since it requires x_1 and x_2.

4.2.3 Security against Replay Attacks

For this attack, an attacker should be able to generate $r(x_1 + c)x_2P = rx_1x_2P + rcx_2P$ for a new value of c using some of the previously exchanged protocol messages. Since x_1 and x_2 are fixed, independent random values, x_2P and x_1x_2P can be considered as two independent public-keys of a tag. Therefore, by the transmission of rP, the server and a tag can share two independent shared-secrets of rx_2P and rx_1x_2P, which are indistinguishable from a random point, assuming the hardness of the Decisional Diffie-Hellman problem. Therefore, $r(x_1+c)x_2P$ can be considered as follows for an attacker.

$$rx_1x_2P + rcx_2P = R_1 + cR_2 \tag{13}$$

where R_1 and R_2 are random points.

Note that R_1 and R_2 are unknown to an attacker and independently generated each time of the protocol. This can be reduced to ECDSA [1] (see Fig. 11) where a signature on c is computed, as shown in Theorem 1. Therefore, as long as ECDSA is a secure signature algorithm, an attacker should not be able to generate another valid message for a different value of c.

- **Generator's Input:** private-key d.
- **Verifier's Input:** Generator's public-key $Q(= dP)$.

Signature Generation

1. Calculate $e = \text{SHA1}(m)$.
2. Select a random integer k from $1 \leq k \leq n-1$.
3. Calculate $r = x_1 (\text{mod } n)$, where $(x_1, y_1) = kP$. If $r = 0$, go back to step 2.
4. Calculate $s = k^{-1}(e + rd) (\text{mod } n)$. If $s = 0$, go back to step 2.
5. The signature for the message m is (r, s).

Signature Verification

1. Verify that $1 \leq r, s \leq n-1$.
2. Calculate $e = \text{SHA1}(m)$.
3. Calculate $u_1 = es^{-1} (\text{mod } n)$ and $u_2 = rs^{-1} (\text{mod } n)$.
4. Calculate $(x_1, y_1) = u_1P + u_2Q$.
5. If $r = x_1 (\text{mod } n)$, then the signature is valid.

Figure 11: EC Digital Signature Algorithm

THEOREM 1. *The search protocol can be cryptographically reduced to the ECDSA assuming the hardness of the Decisional Diffie-Hellman problem.*

PROOF. In ECDSA, a signature on m is computed as follows.

$$\{r, s\} \quad = \quad \left\{x_1 \ (\text{mod } n),\ k^{-1}e + k^{-1}rd \ (\text{mod } n)\right\} \tag{14}$$

where $(x_1, y_1) = kP$, k is a random number, $e = SHA1(m)$ and d is the private-key of the signature generator.

We can consider a new signature algorithm which is stronger than ECDSA as follows.

$$\{r, s\} = \{r_1,\ r_2e + r_3\} \tag{15}$$

where r_1, r_2 and r_3 are independent random numbers.

In order to verify the signature of Eq. (15), a verifier should receive r_1, r_2 and r_3 securely. A way to transfer these values securely is out of the scope of this paper. At this moment, we just care about whether an attacker can get any advantage by eavesdropping the exchanged message (i.e. the signature on m). Nevertheless, what we can be sure of is that Eq. (15) is at least as secure as Eq. (14).

Now let us have a closer look to our proposed search protocol.

$$\{c, rP, r(x_1 + c)x_2P\} = \{c, rP, rx_1x_2P + crx_2P\}$$

c is comparable with the message e being signed, and rP, rx_1x_2P and rx_2P are comparable with three random values r_1, r_2 and r_3. rP is an actual random point, and rx_1x_2P and rx_2P are undistinguishable from real random points for an attacker as long as the Decisional Diffie-Hellman problem is hard. Therefore, the search protocol can be reduced to ECDSA with the assumption of the hardness of the Decisional Diffie-Hellman problem. □

4.2.4 Wide-Weak Privacy

There are three exchanged messages in the protocol: c, rP, $r(x_1 + c)x_2P$. Among these messages, rP is a random point and $r(x_1 + c)x_2P$ is indistinguishable from a random point as long as the Decisional Diffie-Hellman problem is

hard. Therefore, an attacker has no benefit from these two messages to track a tag. Therefore, the protocol is at least narrow-weak private. An wide attacker knows whether a set of messages is accepted or not. In order to utilize this decision, an attacker should be able to forge a set of messages related with a valid message set and check if it is accepted by a tag, similar to the attacks shown in Fig. 2 and 4. However, a successful attack would mean that an attacker can generate a valid digital signature (taking into account that the search protocol can be reduced to ECDSA). Therefore, the proposed search protocol is wide-weak private.

Note that c does not involve any secret information of a tag. However, if c is a counter being queried for a specific tag, it could cause some leakage. This can be solved by increasing c in a different way. The server may keep only one counter and use it for all the tags. Since a tag will accept the counter as long as it is larger than the saved value, the protocol will work and the revealed counter will not indicate how many times a certain tag has been queried.

An alternative solution is the use of a time stamp. Since the time is incremental like a counter, it prevents reusing the value c. Note that the tag does not need to generate the value c (it only has to check that the value is larger than the saved value), and hence does not need to have a timer. By using a time stamp, c will no longer be a counted number of queries. Even if the time stamp covers up to 1000 years with a precision down to $nsec$, this resolution can be covered with 65 bits, which is much less than a full word size (e.g., 163 bits) for a reasonable security level.

4.3 Combining the Authentication Protocols

After a successful instantiation of the search protocol, an RFID tag can detect that the server is searching it. Next, the tag will authenticate itself to the server. However, one should take care that the response from the tag does not allow an attacker to track it. That is why the search protocol should be combined with the Pwd-transfer scheme. Since the server is only expecting a response from one particular tag, the ID-transfer scheme is not necessary. When combining the search protocol and the Pwd-transfer scheme, we need to check that the necessary privacy and security properties still hold. The exchanged messages in the Pwd-transfer scheme are as follows:

$$r_{t1}P,\ r_{s1},\ (r_{t1} + \dot{r}_{s1}x_1x_2)Y \tag{16}$$

The exchanged messages in the search protocol are as follows:

$$c,\ rP,\ (rx_1 + rc)x_2P \tag{17}$$

$(r_{t1} + \dot{r}_{s1}x_1x_2)Y$ is the only message using the base point Y, and therefore, it will be independent of the messages in the search protocol. Moreover, the other messages in the Pwd-transfer scheme, $r_{t1}P$ and r_{s1}, are random values that are not used in the search protocol. Therefore, they are independent. As a result, the combination of the search protocol and the Pwd-transfer scheme will inherit the weaker security and privacy properties of the two protocols. The combined protocol is hence wide-weak privacy-preserving (i.e. as the search protocol). It does not protect against a narrow-strong attacker, since such an attacker can interpret the message sent by the server in the search protocol (and hence knows which particular tag the server is looking for). If the attacker detects that a tag replies to the server's message (without the attacker being able to interpret the response), he knows that the requested tag is present, and can hence track it.

The privacy of the authentication protocols and the search protocol is summarized in Table 2. The table also shows the required number of EC point multiplications, which is the most exhaustive computation carried out in the protocols.

5. 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 that we designed has an architecture similar to the one of Lee et al. presented in [22]. However, further optimizations are performed in our work, and the overall architecture is shown in Fig. 12.

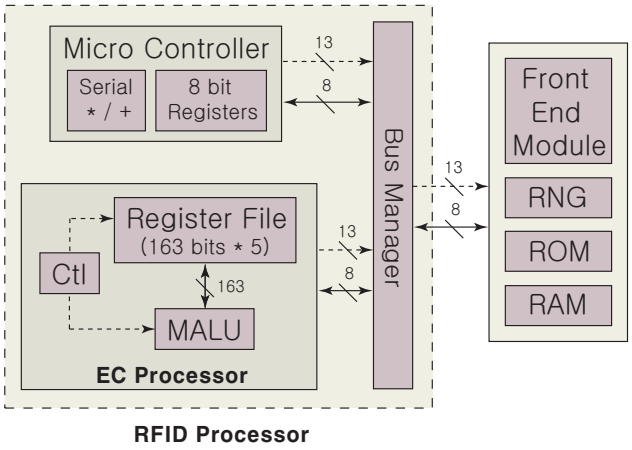

Figure 12: RFID Processor Architecture

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 of Fig. 12. The solid arrows are for data exchange and the dash arrows are for addressing. The control signals are omitted in the picture. The ROM stores program code 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 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 key (k) and an EC point (P) from ROM or RAM and executes the EC point multiplication (kP). After finishing the point multiplication, it stores the results in RAM.

The main differences when compared with [22] are in the register file and the MALU (Modular ALU). The original EC processor uses a MALU which performs modular addition and multiplications, and it reuses the logic of modular multiplications for modular square 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 while the modular multiplication is performed in a digit-serial fashion, the performance can be substantially increased with an overhead of the square logic. Moreover, the size of register file is reduced from 6×163 bits to 5×163 bits. This reduction is possible

Table 2: Privacy Summary

Protocols	Privacy	EC point mult.	
		Server	Tag
Authentication Protocol 1	Narrow-strong and wide-weak	3	3
Authentication Protocol 2	Narrow-strong and wide-weak	5	5
Search Protocol	Wide-weak	2	2
Search + Pwd-transfer	Wide-weak	5	5

since the specialized squarer requires only one operand while the reuse of a multiplier for squaring requires two operands of the same value. As a result, the overall circuit area can be reduced further even after including the squarer in the MALU while achieving a much higher performance.

The performance comparison is summarized in Table 3 where both architectures have the digit size of 4 in the MALU. This work achieves about 24% better performance with a smaller circuit area, and the energy consumption is much smaller. Moreover, this work includes the coordinate conversion to affine-coordinates from Z-coordinates while the work of [22] gives outputs in Z-coordinates.

Table 3: Performance Comparison

Criteria	[22]	This Work
Circuit Area (Gage Equi.)	15,356	14,566
Cycles for EC point mult.	78,544	59,790
Frequency	323 KHz	700 KHz
Power	12.1 μW	13.8 μW
Energy for EC point mult.	2.94 μJ	1.18 μJ

The performance results of our proposed protocols are summarized in Table 4 where a $0.13\mu m$ CMOS technology is used, and the gate area does not include RNG, ROM and RAM which are required to store or run the programmed protocols. The area specifies a complete EC processor with the required registers. The protocols 1 and 2 are the two improved RFID tag-to-server authentication protocols (Fig. 5 and 9, respectively).

According to [13], the current consumption for all security services on RFID should not exceed $15\mu A$, which corresponds to $22.5\mu W$ for $1.5V$ in our CMOS library. Therefore, the power consumption of $13.8\mu W$ in our design will be low enough, even if we count the extra power consumption in the required memory.

6. CONCLUSIONS

In this paper two new RFID authentication protocols that are resistant to man-in-the-middle attacks are presented, as improvement of the work of Lee, Batina and Verbauwhede [21]. Each protocol has different security characteristics and computational workload. In order to analyze the privacy properties of our protocols, we used parts of the adversarial model defined in the theoretical framework of Vaudenay [30]. In this context, both of our proposed protocols are narrow-strong and wide-weak privacy-preserving. Further on, the search protocol is presented as a novel scheme where a server (or a reader) can efficiently query for a specific tag, without compromising the tag's privacy.

In addition, we presented a hardware architecture, which can be produced with less than 15 Kgates, that can realize the proposed randomized authentication and search protocols. The performance results show the feasibility of the proposed protocols, even for a passive tag, and outperform other secure and privacy-preserving protocols published in the literature.

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.

7. REFERENCES

[1] ANSI. X9.62 The Elliptic Curve Digital Signature Algorithm (ECDSA). `http://www.ansi.org`.

[2] G. Avoine. Adversarial Model for Radio Frequency Identification. Cryptology ePrint Archive, Report 2005/049, 2005. `http://eprint.iacr.org/`.

[3] 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.

[4] 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.

[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] S. Engberg, M. Harning, and C. Jensen. Zero-knowledge Device Authentication: Privacy &

Table 4: Performance Results of Our Proposed Protocols

Protocols	Cycles	Time (ms)	ROM for program	ROM for data	RAM	Nonvolatile RAM
Auth. Protocol 1 (Fig. 1)	206,202	295	36	105	107	–
Auth. Protocol 2 (Fig. 9)	328,074	469	80	126	107	–
Search Protocol (Fig. 10)	120,505	172	61	105	128	21

- Gate Equivalent Area: 14,566 GE, Frequency: 700 KHz, Power: 13.8 μW, Technology: 0.13 μm.
- All the required memories are in bytes, the search protocol needs an extra 21 bytes of nonvolatile RAM for a counter.
- Gate area and power do not include memory.
- Data transmission in protocols is counted as 1 cycle per byte.

Security Enhanced RFID preserving Business Value and Consumer Convenience. In *Proceedings of the Second Annual Conference on Privacy, Security and Trust (PST '04)*, pages 89–101, 2004.

[11] 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/`.

[12] M. Feldhofer. An Authentication Protocol in a Security Layer for RFID Smart Tags. In *IEEE Mediterranean Electrotechnical Conference - IEEE MELECON'04*, 2004.

[13] 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.

[14] D. Frumkin and A. Shamir. Un-Trusted-HB: Security Vulnerabilities of Trusted-HB. In *Proceedings of RFIDSec09*, Leuven, Belgium, 2009.

[15] 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.

[16] 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.

[17] N. Hopper and M. Blum. Secure human identification protocols. In *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT '01)*, pages 52–66. Springer-Verlag, 2001.

[18] A. Juels and S. Weis. Defining Strong Privacy for RFID. Cryptology ePrint Archive, Report 2006/137, 2006. `http://eprint.iacr.org/`.

[19] 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.

[20] 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.

[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*, 57(11):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. 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] T. Phillips, T. Karygiannis, and R. Kuhn. Security standards for the RFID market. *Security & Privacy*, 3(6):85–89, 2005.

[26] A. Razaq, W. Luk, K. Shum, L. Cheng, and K. Yung. Second-Generation RFID. *Security & Privacy*, 6(4):21–27, 2008.

[27] 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.

[28] S. Spiekermann, and S. Evdokimov. Critical RFID Privacy-Enhancing Technologies. *IEEE Security and Privacy*, 7(2):56–62, 2009.

[29] 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.

[30] 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.