Open access • Proceedings Article • DOI:10.1109/RFID.2008.4519370

EC-RAC (ECDLP Based Randomized Access Control): Provably Secure RFID authentication protocol — Source link

Yong Ki Lee, Lejla Batina, Ingrid Verbauwhede

Institutions: University of California, Los Angeles, Katholieke Universiteit Leuven

Published on: 16 Apr 2008 - International Conference on RFID

Topics: Authentication protocol, Cryptographic protocol, Message authentication code, Public-key cryptography and Cryptography

2008 IEEE International Conference on RFID
The Venetian, Las Vegas, Nevada, USA
April 16-17, 2008

EC-RAC (ECDLP Based Randomized Access Control): Provably Secure RFID authentication protocol

Yong Ki Lee[(1),(2)]
jfirst@ee.ucla.edu

Lejla Batina[(2)]
lejla.batina@esat.kuleuven.be

Ingrid Verbauwhede[(1),(2)]
ingrid@ee.ucla.edu

[(1)] Department of Electrical Engineering,
University of California, Los Angeles,
CA 90024, USA

[(2)] Department of Electrical Engineering,
ESAT-COSIC,
Katholieke Universiteit Leuven,
Belgium

Abstract— **Operational and security requirements for RFID systems such as system scalability, anonymity and anti-cloning are difficult to obtain due to constraints in area, memory, etc. Due to scarceness of resources most of the proposed protocols were designed using symmetric key cryptographic algorithms. However, it has been shown that it is inevitable to use public-key cryptographic algorithms to satisfy these requirements [1]. Moreover, general public-key cryptography based authentication protocols are vulnerable in terms of anonymity, which is shown in this paper. Accordingly, we design a new authentication protocol named EC-RAC using EC (Elliptic Curve) cryptography. EC-RAC can be proved for its security in the generic group model and is carefully designed to minimize its computational workload. Moreover, we present the implementation results of EC-RAC to show its feasibility for RFID systems.**

I. INTRODUCTION

RFID (Radio Frequency Identification) systems are one of the most challenging devices recent years in many fields such as wireless communication, circuit and electromagnetic areas. The reason is that there are so many potential or ongoing applications of RFID systems such as supply chains, livestock/inventory tracking, toll management, airline baggage management, access control and so on. It can also be used to discriminate between counterfeits and authentic products. Especially since the adoption of EPCglobal Gen2 [2], the RFID is expected to completely replace the bar code systems in near future.

For commercial markets, RFID systems should overcome not only the restriction of cheap RFID tags but also operational and security problems such as scalability, the tracking problem and the cloning problem. In many cases, the security part is simplified in order to minimize a tag's price. For example, Class-1 EPCglobal Gen2 [2] has a very simple authentication scheme where a password is transmitted in a plain text, which can cause many security problems. Fortunately, the CMOS technologies steadily advance and the fabrication costs decrease, which allows stronger security solutions on tags. Moreover, some applications such as expensive goods and access control systems that should be highly secured can afford more expensive tags which may include more resources such as an extra power source, gate area and memory.

First, we summarize some essential operational and cryptographic properties for general RFID systems in order to clarify the issues of the paper.

- **Scalability:**
  If the computational workload of an authentication protocol increases linearly as the number of the tags, the system is not scalable. Noting that most RFID applications should accommodate a large number of tags, e.g. a large library may have millions of books and each book should have a tag, the scalability is a critical property in RFID systems.
- **Anti-cloning:**
  Since a large number of tags will be spread out in the RFID applications, an attacker may be able to capture a tag, investigate it by microscope probing [4], learn all the information in the tag, and make a counterfeit. However, an attacker should not be able to forge other tags except the cracked one. If a group of tags share secret information and a reader authenticates tags by the shared secret, it will be possible to clone some other tags with the learned secret. This will also cause the tracking problem since an attacker can decrypt the exchanged messages. Therefore, the secret information on a tag should be pertinent to the tag so that the other tags except the cracked one are still secure.
  One possible way to protect the secret stored in a tag is to use a secure memory [5]. However, it is not practical to store a long-term secret (a group key, shared secret among a group of tags and readers) in tags and to use it for authentication since only single cracked tag may endanger all the tags and readers having the shared secret.
  In this paper, assuming that an attacker is able to crack and reveal the secret in a tag, we define an RFID system secured against the cloning attack as long as the secret of a tag is pertinent to the tag and secured from passive or active skimming attacks.
- **Anonymity:**
  RFID tags are supposed to respond with some message

978-1-4244-1712-4/08/$25.00 ©2008 IEEE

whenever they receive a query message from a reader. If the responses are fixed or predictable by an attacker, it results in a privacy problem. An attacker is possibly able to track a tag, and hence its owner too, and collect data for malicious purpose. Therefore, the responses of tags should be randomized so that it is infeasible to extract any information in communications between a tag and a reader.

Some of the proposed authentication protocols use hash algorithms and/or symmetric key algorithms due to their simplicity compared to public-key algorithms. However, they fail to satisfy the mentioned basic requirements of RFID systems. This is consequential noting the proof in [1], where it is shown that a public-key cryptographic algorithm is necessary to satisfy the required properties. Some other propose to adopt well-known public-key based authentication protocols such as the Schnorr protocol [19] and the Okamoto protocol [20], which are suitable for general authentication systems that do not concern anonymity but not for RFID systems.

The contributions of this paper can be summarized as follows:

1) The security against the tracking attack is formalized. The definition is general and covers not only passive attacks but also active attacks.
2) Based on the security definition, we analyze the security of some well-known ECDLP based authentication protocols and show that they are vulnerable to the tracking attack.
3) We propose a new authentication protocol named EC-RAC and formally prove its security in the generic group model.
4) We present the implementation results of EC-RAC to show that it is also feasible for high-end tags.

The remainder of this paper is organized as follows. In Sec. 2, some related work is introduced, and in Sec. 3, the security of ECDLP based authentication protocols of Schnorr and Okamoto is analyzed. EC-RAC is proposed and its security is analyzed in Sec. 4 and Sec. 5 respectively. We present the implementation results of EC-RAC in Sec. 6 followed by the conclusion in Sec. 7.

II. Related Work on RFID Authentication Protocols

Many protocols have been proposed for RFID systems using a hash algorithm due to their cheap implementations [6], [7], [8], [9], [10], [13], [14], [15]. Some other protocols using secret key cryptographic algorithms are also proposed in [11], [12]. These protocols are divided into fixed access control and randomized access control. Randomized access control again can be divided based on whether a system-wide common secret key (a group key) is used or not. However, they could not satisfy some of the basic operational and/or security requirements of RFID systems.

In the fixed access control, e.g. [6], a tag replies to a reader with a fixed message so that the protocol can be designed with simple cryptographic primitives, which allows a cheap price of tags. However, this kind of protocols is vulnerable to the tracking attack due to the constant responses of tags.

A solution to prevent the tracking problem is the randomized access control. In order to randomize messages, a reader and a tag need to share some secret information which is unknown to attackers so that only the entities which have the secret information can interpret the randomized messages. Without using a group key, randomized access controls are not scalable since the workload of the reader increases linearly as the number of the tags. Some protocols of this type are presented in [6], [9]. Protocols proposed in [7], [10] resolve the tracking problem and the scalability by sharing a group key among all the readers and the tags. However, they neglect the possibility of the compromised group key. Once the group key is revealed by cracking a tag, all the tags of the system will be vulnerable to not only the cloning attack but also the tracking attack.

Some protocols have been proposed to solve the tracking problem and scalability with hash algorithms [13], [14], [15]. This is done by updating the stored information in tags regularly. However, they still have some drawbacks. In [13], the keys of tags are updated only when the authentication protocols are successful, and hence all the response from malicious queries, which lead unsuccessful authentications, will be fixed until the next successful authentication. Therefore, between two consecutive successful authentications, tags are vulnerable to the tracking attack. In [14], [15], tags are vulnerable to the denial-of-service attack since tags updates their key or local information regardless of the success of the protocols.

There are some proposals to use asymmetric cryptographic algorithms for RFID systems. In [1], an IFP (Integer Factorization Problem) based protocol is proposed. In [16], [17], [18] they proposed to use ECDLP (Elliptic Curve Discrete Logarithm Problem) based authentication protocols for RFID systems, which will be analyzed in the following section.

III. Security Analysis of ECDLP Based Authentication Protocols

Some attempts to apply elliptic curve cryptography to RFID systems are done in [16], [17], [18]. In [16] no specific authentication protocol is mentioned, and the Schnorr protocol [19] and the Okamoto protocol [20] are adopted in [17] and [18] respectively. These two protocols are two of the most popular authentication protocols based on ECDLP (Elliptic Curve discrete Logarithm Problem). However, they are not proper for RFID systems since they are designed without considering anonymity. In these protocols, it is conventionally assumed that the ID (or public key) of a prover (a tag) is already known to a verifier (a reader or a server). However, transmitting ID's in secret is a main goal in the RFID authentication protocol. Even if we assume a tag's ID is conveyed to a reader securely, they still have the tracking problem.

In order to discuss the issue of the tracking attack, we put forward a formal definition for the security against the tracking attack which is very strong since it can be applied to not only passive attacks but also active attacks.

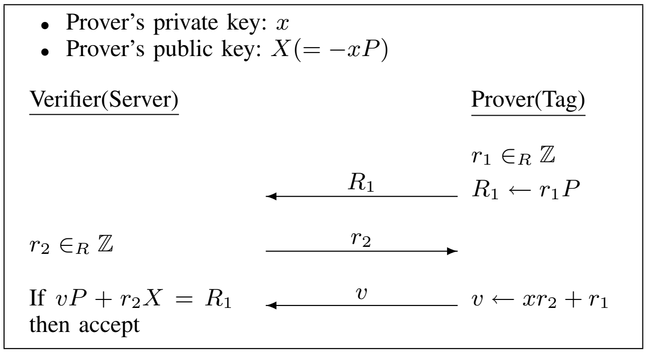
Definition 1: An authentication protocol is secure against the tracking attack if the following polynomial time oracle does not exist.

$$\begin{aligned} &Q\left(param_1, param_2, ..., param_m\right) \\ &= \widetilde{f}(var_1^T, var_2^T, ..., var_n^T) \end{aligned} \qquad (1)$$

where $\{param_1,\ param_2,\ ...,\ param_m\}$ is the set of the known values such as exchanged messages and possibly revealed values to an attacker, and $\{var_1^T,\ var_2^T,\ ...,\ var_n^T\}$ is the set of variables which can indicate a specific tag such as a tag's secret and public keys. The function $\widetilde{f}(\bullet)$ can be any polynomial time function whose output includes at least one variable indicating a specific tag and does not include any random variable.

Conceptually, the definition states that deriving any fixed value indicating a specific tag must be infeasible. In the remainder of this section, some ECDLP-based authentication protocols are introduced and the definition is applied to these protocols to show their vulnerability against the tracking attack.

A. The Schnorr Protocol



- Prover's private key: x
- Prover's public key: $X(= -xP)$
Verifier(Server)
Prover(Tag)
$r_1 \in_R \mathbb{Z}$
R_1
$R_1 \leftarrow r_1 P$
$r_2 \in_R \mathbb{Z}$
r_2
If $vP + r_2 X = R_1$ then accept
v
$v \leftarrow xr_2 + r_1$

Fig. 1. The Schnorr Protocol

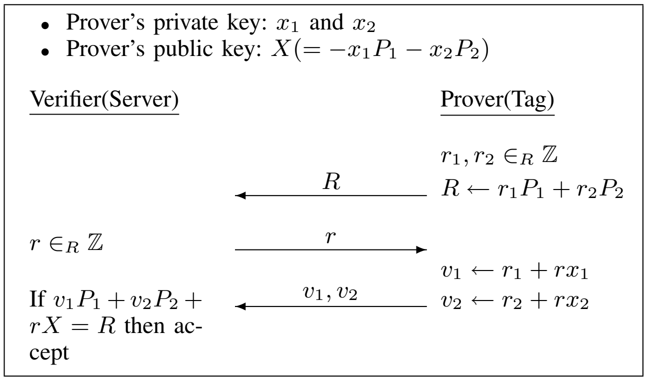
The message flow of the Schnorr protocol is shown in Fig. 1 where r_1 and r_2 are random numbers generated by a prover (tag) and a verifier (reader/server) respectively. The prover's secret key is x and its public key is $X = -xP$. If $vP + r_2X = R_1$ at the end of the protocol flow, then the verifier accepts the prover, else rejects.

If we apply Definition 1 to the Schnorr protocol, the parameters of Q are the exchanged messages, i.e. r_1P, r_2 and xr_2+r_1, and the system parameter, i.e. P. A polynomial time oracle can be defined as follows.

$$\begin{aligned} &Q\left(r_1P, r_2, xr_2 + r_1, P\right) \\ &= \{r_1P - (xr_2 + r_1) \cdot P\} \cdot r_2^{-1} = -xP \end{aligned} \qquad (2)$$

$-xP$ satisfies the requirements of $\widetilde{f}(\bullet)$ since there is a variable x which can be an indication of a tag and there is no random variable such as r_1 and r_2. Therefore, the Schnorr protocol is not secure against the tracking attack since a polynomial time oracle defined in Definition 1 exists. In other words, an attacker can track a tag by deriving $-xP$.

B. The Okamoto Protocol



- Prover's private key: x_1 and x_2
- Prover's public key: $X(= -x_1P_1 - x_2P_2)$
Verifier(Server)
Prover(Tag)
$r_1, r_2 \in_R \mathbb{Z}$
R
$R \leftarrow r_1P_1 + r_2P_2$
$r \in_R \mathbb{Z}$
r
$v_1 \leftarrow r_1 + rx_1$
If $v_1P_1 + v_2P_2 + rX = R$ then accept
v_1, v_2
$v_2 \leftarrow r_2 + rx_2$

Fig. 2. The Okamoto Protocol

The Okamoto protocol is described in Fig. 2 where r_1 and r_2 are random numbers generated in a prover and r is in a verifier. The secret key of the prover is a pair of x_1 and x_2 and its public key is $X = -x_1P_1 - x_2P_2$. After finishing the message exchanges, the verifier accepts the prover if $v_1P_1 + v_2P_2 + rX = R$, otherwise reject.

The Okamoto protocol also has the tracking problem since a polynomial oracle of Definition 1 can be described with the following equation.

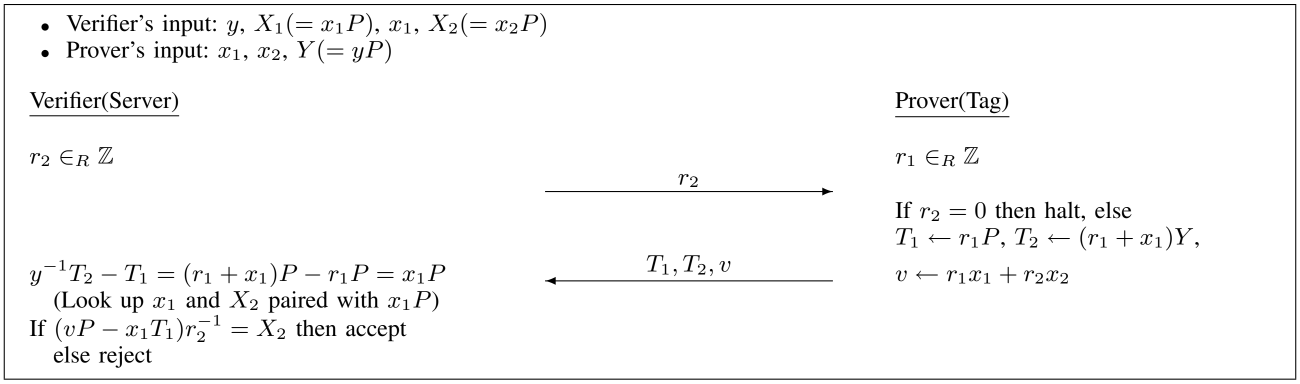
$$\begin{aligned} & Q\left(r_1P_1 + r_2P_2, r, r_1 + rx_1, r_2 + rx_2, P_1, P_2\right) \\ = \ & \{R - v_1P_1 - v_2P_2\}\, r^{-1} \qquad (3) \\ = \ & \{(r_1P_1 + r_2P_2) - (r_1 + rx_1)P_1 - (r_2 + rx_2)P_2\}\, r^{-1} \\ = \ & \{-rx_1P_1 - rx_2P_2\}\, r^{-1} = -x_1P_1 - x_2P_2. \end{aligned}$$

Note that the parameters of Q are the exchanged messages and the system parameters. The output of the oracle $(-x_1P_1 - x_2P_2)$ is the public key of a tag and it satisfies the conditions of Definition 1 since it has some variables indicating a specific tag, i.e. x_1 and x_2, and does not hvae any random variable.

To summarize, the conventional ECDLP based authentication protocols shown in this section are not suitable for RFID systems. Therefore, we need a new RFID protocol that considers not only secure transmissions of a tag's identity but also the tracking attack.

IV. EC-RAC Protocol

To solve all the requirements for RFID systems, we design a new RFID protocol based on the elliptic curve discrete logarithm problem. Among public-key cryptographic algorithms, a ECC based algorithm would be the best choice due to its small key size and computational efficiency. Moreover, when a protocol is designed, the computational workload on tags should be minimized. This may cause an increase of the workload of the server (or reader). Since the server is supposed to have sufficient resources such as power and



- Verifier's input: y, $X_1(= x_1P)$, x_1, $X_2(= x_2P)$
- Prover's input: x_1, x_2, $Y(= yP)$

Verifier(Server)
Prover(Tag)
$r_2 \in_R \mathbb{Z}$
$r_1 \in_R \mathbb{Z}$
r_2
If $r_2 = 0$ then halt, else
$T_1 \leftarrow r_1P$, $T_2 \leftarrow (r_1 + x_1)Y$,
$v \leftarrow r_1x_1 + r_2x_2$
T_1, T_2, v
$y^{-1}T_2 - T_1 = (r_1 + x_1)P - r_1P = x_1P$
(Look up x_1 and X_2 paired with x_1P)
If $(vP - x_1T_1)r_2^{-1} = X_2$ then accept
else reject

Fig. 3. EC-RAC Protocol Flow

1) Tag and Server generate random numbers r_1 and r_2 respectively.
2) Server sends r_2 to Tag.
3) If $r_2 = 0$ then stop the protocol. Otherwise Tag generates and sends three messages $T_1 = r_1P$, $T_2 = (r_1 + x_1)Y$ and $v = r_1x_1 + r_2x_2$ to Server.
4) Server calculates $y^{-1}T_2 - T_1 = y^{-1}(r_1 + x_1)yP - r_1P = x_1P = X_1$ and using the result (X_1) searches for x_1 and X_2. If there is a valid set for X_1, Server calculates $(vP - x_1T_1)r_2^{-1} = \{(r_1x_1 + r_2x_2)P - x_1r_1P\}\, r_2^{-1} = x_2P$ and check whether it is the same as the stored X_2. If it is, Server authenticates Tag as a valid one.

Fig. 4. EC-RAC Protocol Description

memory compared to tags, transferring the workload of tags to the server is desirable if it is possible.

Before designing the EC-RAC protocol, we should note that RFID systems have different situations from conventional password systems and public-key cryptography based authentication systems as the following:

1) Unlike conventional password protocols, RFID systems should not just transfer a tag's ID.
2) Unlike conventional public-key cryptography based authentication protocols, the protocols are many to one protocols, i.e. many RFID tags communicate with one reader/server. Due to this property, tags' public keys do not need to be publicly announced and hence, they can and should be securely stored and used for authentications in the server.

Similarly to conventional password protocols which require two values for each prover, i.e. ID and Password, our protocol starts with two secret keys, x_1 and x_2, which are compatible to ID and Password. The public keys, x_1P and x_2P, are used as ID-verifier and Password-verifier which are securely stored in the server unlike general public keys.

The protocol flow and the description are shown in Fig. 3 and Fig. 4 respectively. The EC (Elliptic Curve) point scalar multiplication is the critical operation in the protocol. While a server needs 3 scalar multiplications, a tag needs only 2 scalar multiplications. It is desirable to reduce the workload in a tag even if it increases the computational workload in a server. Another noticeable thing is that general EC point additions/subtractions and scalar inverse operations are avoided in a tag while they are not in a server. This results in a minimized control and gate area on a tag.

In this protocol, it is assumed that a server stores y, $X_1(= x_1P)$, x_1 and $X_2(= x_2P)$, and a tag stores x_1, x_2 and $Y(= yP)$. During the protocol flow the ID (x_1) and Password (x_2) of a tag are encrypted for the transmission to a server. After decrypting $X_1(= x_1P)$, the server searches for x_1 and $X_2(= x_2P)$ paired with X_1 and verify that X_2 is correct by checking whether $(vP - x_1T_1)r_2^{-1} = X_2$.

V. Security Analysis

In order to analyze the security of EC-RAC, we use the generic group model [21], [22], [23]. In this model, an attacker does not have access to group elements but to the images of the group elements, which are one-to-one mapped to random strings. For a given group G, the random mapping of the group elements to the images can be described as $\sigma : G \rightarrow \{0, 1\}^l$ where l is the length of the random strings. An attacker can perform an addition oracle Add, an inverse oracle Inv and a scalar multiplication oracle for Mul for the group operations as follows.

$$\begin{aligned} Add(\sigma(x), \sigma(y)) &= \sigma(x + y) \\ Inv(\sigma(x)) &= \sigma(-x) \qquad (4) \\ Mul(k, \sigma(y)) &= \sigma(k \cdot y) \end{aligned}$$

Mul is redundant since it can be easily implemented by a polynomial time algorithm using Add, e.g. a scalar multiplication can be implemented with the *double and add algorithm* where the doubling and the addition can be done by Add.

When the generic group model is instantiated on an EC group, group elements x and y can be considered as scalar values, and $\sigma(x)$ and $\sigma(y)$ as $x \cdot P$ and $y \cdot P$ where P is the base point. The generic group model ensures that an attacker has no gain at deriving the group element x from $\sigma(x)$, i.e. $x \cdot P$, which means that there is no efficient (or polynomial time) algorithm which derives scalar values from EC points. This fact will be used when we analyze the EC-RAC protocol. In this paper, we use naive forms of EC points such as $x \cdot P$ instead of using $\sigma(x)$ and hence assume that $x \cdot P$ is a randomly mapped string just like $\sigma(x)$.

The security proof is done by contradiction as the following procedure. We use the fact that the Diffie-Hellman scheme [24] is secure in the generic group model, which is already proven in [21].

1) We assume that the protocol is un-secure and then there exists a polynomial time oracle Q which calculates some secrete information in polynomial time with publicly known or possibly revealed values.
2) We show that the oracle Q defined in step 1 can be reduced to another oracle which is obviously impossible to solve or to the Diffie-Hellman problem. If the oracle Q is reduced to the Diffie-Hellman problem, the existence of such Q implies that Diffie-Hellman problem is solvable in polynomial time.
3) By contradiction, the proof of the security is done.

We analyze the security in three different settings: attacking as a third observer, attacking as a valid server and attacking as a valid tag. Moreover, we analyze the security against the tracking attack. In the analysis, we assume that x_1, x_2 and y are randomly chosen.

A. Security Analysis Against an Attacker as a Third Observer

In this sub-section, we prove that a third observer cannot extract any secret information, i.e. x_1, x_1P, x_2, x_2P, y, and yP. As a start of the security proof, we assume the worst case: all the exchanged messages between tags and the server are revealed and collected for an attacking purpose; all the system parameters including P, and yP are also publicly known by cracking a tag. Note that even if we assume that yP is known, checking whether the system is actually using the same yP or a different one must be infeasible. Leaking yP may not be a problem in general public-key cryptographic systems since it is a public key. However, in some RFID applications such as supply chains, the public key of the server can be an indication of a product's brand name which is also private information.

- Security for x_1P (and hence for x_1) :

Note that the security of x_1P is a sufficient condition of the security of x_1. This is because if x_1 is compromised, x_1P can also be calculated. We assume there is a polynomial time oracle Q which calculates x_1P.

$$Q\left(r_2, r_1P, (r_1+x_1)yP, r_1x_1+r_2x_2, yP, P\right) = x_1P$$

In order to utilize $r_1x_1+r_2x_2$, we need to convert this parameter to an EC point by multiplying by an EC point (Though we can do some scalar operations before converting to an EC point, there is no meaningful operations considering that there is only one more scalar parameter, r_2). Note that in the generic group model, the allowed group operations for an attacker are the point addition and the point inversion. Therefore, each term of EC points must be considered to be independent, e.g. r_1P and r_1x_1P are independent terms. If $r_1x_1+r_2x_2$ is multiplied by any EC point among the given parameters, it generates one new parameter and two new terms. For example, if $r_1x_1+r_2x_2$ is multiplied by r_1P, the newly generated parameter is $r_1^2x_1P + r_1r_2x_2P$, and the newly generated terms are $r_1^2x_1P$ and $r_1r_2x_2P$. Therefore, it generates more terms than parameters, which means converting $r_1x_1+r_2x_2$ to an EC point does not help for solving x_1P. Therefore, we can eliminate $r_1x_1+r_2x_2$ without losing generality. r_2 also can be eliminated since $r_1x_1+r_2x_2$ is the only parameter having r_2. Actually, it does not help for any term of EC points, and hence, we exclude $r_1x_1+r_2x_2$ and r_2 when we need to derive an EC point throughout this paper.

Therefore, Q is simplified as follows.

$$\Rightarrow Q\left(r_1P, (r_1+x_1)yP, yP, P\right) = x_1P \qquad (5)$$

We reduce the oracle to Q' by assuming that r_1 is known.

$$\Rightarrow Q'\left(r_1P, (r_1+x_1)yP, yP, P, r_1\right) = x_1P$$

Q' is simplified noting that $r_1P \cdot r_1^{-1} = P$ and $(r_1 + x_1)yP - r_1 \cdot yP = x_1yP$.

$$\Rightarrow Q'\left(x_1yP, yP, P, r_1\right) = x_1P$$

Since r_1 is no more relevant to this problem, we eliminate it.

$$\Rightarrow Q'\left(x_1yP, yP, P\right) = x_1P$$

This can be reduced to the Diffie-Hellman scheme as follows, which is shown in Theorem 1.

$$Q''\left(x_1P, yP, P\right) = x_1yP$$

The existence of Q'' conflicts with the fact that the Diffie-Hellman scheme is secure in the generic group model. Therefore, security for x_1P is proven by contradiction.

Theorem 1: If a polynomial time oracle $Q(xyP, yP, P) = xP$ exists, then a polynomial time oracle $\hat{Q}(xP, yP, P) = xyP$ exists. Equivalently, if there is no polynomial time oracle $\hat{Q}(xP, yP, P) = xyP$ (i.e. the Diffie-Hellman scheme is secure), then there is no a polynomial time oracle of $Q(xyP, yP, P) = xP$.

Proof: We assume that a polynomial time oracle $Q(xyP, yP, P) = xP$ exists. Then, since $Q(xP, yP, P) = Q(xy^{-1} \cdot yP, yP, P) = xy^{-1}P$, the following oracle $\hat{Q}$ can be equivalently derived as follows.

$$\hat{Q}(xP, yP, P)$$

$$\Rightarrow \hat{Q}(xP, yP, xy^{-1}P, P)$$

Again, since $Q(xP, xy^{-1}P, P) = Q(y^{-1} \cdot xP, xP, P) = y^{-1}P$,

$$\Rightarrow \hat{Q}(xP, yP, xy^{-1}P, y^{-1}P, P)$$

Since $Q(xP, y^{-1}P, P) = Q(xy \cdot y^{-1}P, y^{-1}P, P) = xyP$, the following oracle exists.

$$\Rightarrow \hat{Q}(xP, yP, P) = xyP$$

Therefore, the theorem is proven. ■

- Security for x_2P (and hence for x_2) :

We reuse Eq. (5) where the oracle is simplified. In this time we need to derive x_2P.

$Q\left(r_1P,(r_1+x_1)yP,yP,P\right)=x_2P$

Since there is no parameter having x_2, it is impossible to derive x_2P with the given parameters.

- Security for yP :

Even if we assume that yP is already revealed, the information indicating whether a tag is actually using the same yP or not should be secured. Among the exchanged messages, only the message including y is $(r_1+x_1)yP$. This security can be proved by showing that deriving $(r_1+x_1)yP$ with other known values is infeasible.

First, we assume there is a polynomial time oracle which generates $(r_1+x_1)yP$ as follows.

$Q\left(r_2,r_1P,r_1x_1+r_2x_2,yP,P\right)=(r_1+x_1)yP$

Since r_2 and $r_1x_1+r_2x_2$ are useless for deriving $(r_1+x_1)yP$, we eliminate them.

$\Rightarrow Q\left(r_1P,yP,P\right)=(r_1+x_1)yP$

Now there is no parameter left which includes x_1 which should be used for deriving the output $(r_1+x_1)yP$. Therefore, it is impossible to derive $(r_1+x_1)yP$, and hence the security proof of this part is done.

- Security for y :

We assume there is a polynomial time oracle which calculates y.

$Q\left(r_2,r_1P,(r_1+x_1)yP,r_1x_1+r_2x_2,yP,P\right)=y$

Since the EC points are no use to derive a scalar value in the generic group model, we eliminate all EC points.

$\Rightarrow Q\left(r_2,r_1x_1+r_2x_2\right)=y$

Since there is no parameter left having y, there is no way to derive y.

B. Security Analysis Against an Attacker as a server

We assume that a server is cracked and all the information in the server is known to an attacker. Therefore, x_1 (and hence x_1P), x_2P and y (and hence yP) are revealed. Even if a server is cracked, the attacker should not be able to get the secret information of x_2. This will prevent an attacker from duplicating some tags after hacking a server. Now the secret information known for an attacker is x_1, x_2P and y and we need to show the security of x_2.

- Security for x_2 :

We assume there is a polynomial time oracle which calculates x_2.

$Q\left(r_2,r_1P,(r_1+x_1)yP,r_1x_1+r_2x_2,y,x_1,x_2P,P\right)=x_2$

Since the output is a scalar, we eliminate all EC points.

$\Rightarrow Q\left(r_2,r_1x_1+r_2x_2,y,x_1\right)=x_2$

Q has no more use of y, so we eliminate it.

$\Rightarrow Q\left(r_2,r_1x_1+r_2x_2,x_1\right)=x_2$

In this oracle, x_2 cannot derived since r_1 cannot be removed from $r_1x_1+r_2x_2$.

C. Security Analysis Against an Attacker as a tag

In this case, we assume that a tag is cracked and all the information stored in the tag is revealed by an attacker. In this situation, we prove that the other secret information is secure. Therefore, r_1, x_1 (and x_1P), x_2 (and x_2P) and yP are known. Now the secret information we need to protect is y.

- Security for y :

$Q\left(r_2,r_1P,(r_1+x_1)yP,r_1x_1+r_2x_2,yP,P,r_1,x_1,x_2\right)$ $=y$

We eliminate all EC points.

$\Rightarrow Q\left(r_2,r_1x_1+r_2x_2,r_1,x_1,x_2\right)=y$

There is no more parameter having y left. Therefore, deriving y is impossible and the security for y is proven.

D. Security Against the tracking attack

Against the tracking attack, securing the secret information is not sufficient. In Sec. 3 we analyzed the Schnorr protocol and the Okamoto protocol. Those protocols are secure in terms of not leaking secret keys. However, an attacker can derive their public keys by manipulating the exchanged messages. Using the derived public key of a tag, an attacker can track the tag.

In the case of the proposed EC-RAC, let us suppose a polynomial time oracle Q exists as follows. The parameters of Q not only include the exchanged messages, i.e. r_2, r_1P, $(r_1+x_1)yP$ and $r_1x_1+r_2x_2$, but also include some information which can be extracted by cracking a tag, i.e. yP and P. Note that it is impossible to prevent the tracking of already cracked tags. In this paper, we show that even if an attacker is able to extract the secret information inside of tags by cracking, the other un-cracked tags are still secure.

Moreover, we assume r_2 can be controlled by an attacker. It is possible that an attacker disguise as a reader so that r_2 may not be a random number but a certain constant. (At least, we suppose $r_2\neq 0$. Therefore, a tag should proceed the protocol only when $r_2\neq 0$.) Considering this worst case, $\widetilde{f}(\bullet)$ is allowed to include r_2. Therefore, the output of $\widetilde{f}(\bullet)$ must include x_1 or x_2 and must not include only r_1. In order to make it clear, we substitute r_2 with k.

$Q\left(k,r_1P,(r_1+x_1)yP,r_1x_1+kx_2,yP,P\right)=\widetilde{f}(\bullet)$

There are two possible forms of the output of $\widetilde{f}(\bullet)$, which are a scalar and an EC point.

- Suppose the output of $\widetilde{f}(\bullet)$ is a scalar.

Then, we can eliminate all the parameters of an EC point.

$\Rightarrow Q\left(k,r_1x_1+kx_2\right)=\widetilde{f}(\bullet)$

Now $r_1x_1+kx_2$ is the only parameter which includes some information of a tag. However, it is impossible to eliminate r_1 from $r_1x_1+kx_2$ with the given parameters.

Therefore, the output of $\widetilde{f}(\bullet)$ is not a scalar.

- Suppose the output of $\widetilde{f}(\bullet)$ is an EC point.

Since the parameters k and $r_1x_1+kx_2$ are useless deriving an EC point, we eliminate them.

$Q'\left(r_1P,(r_1+x_1)yP,yP,P\right)=\widetilde{f}(\bullet)$

Since $(r_1+x_1)yP$ is the only parameter having x_1 or x_2, we must derive $\widetilde{f}(\bullet)$ by manipulating $(r_1+x_1)yP$. The only possible way to get rid of r_1 is to derive r_1yP and do the calculation of $(r_1+x_1)yP - r_1yP = x_1yP$. Therefore, the problem is reduced to the following oracle Q'.

$Q'\left(r_1P, (r_1+x_1)yP, yP, P\right) = r_1yP$

Since the number of the parameters are 4 and there are 5 terms (i.e. r_1P, r_1yP, x_1yP, yP and P), it is infeasible to derive r_1yP. Therefore, $\widetilde{f}(\bullet)$ cannot be an EC point either.

As a result, there is no polynomial time oracle Q which produces $\widetilde{f}(\bullet)$, which means that the proposed EC-RAC is secure against the tracking attack in the generic group model.

VI. Implementation Feasibility

We show the performance results using the security processor presented in [25] which is one of the most compact architectures in the literature. The architecture is described in Fig. 5, which is composed of the micro controller, the bus manager and the ECP (Elliptic Curve Processor). ECP, which computes EC point scalar multiplications, is composed of a controller, MALU (Modular Arithmetic Logic Unit) and a register file (whose size is 6×163 bits). In Fig. 5, the solid lines are for data exchange, the dash lines with numbers are for addressing, and the dash lines without numbers are control signals. The processor can perform different authentication protocols according to the programs stored in its ROM.

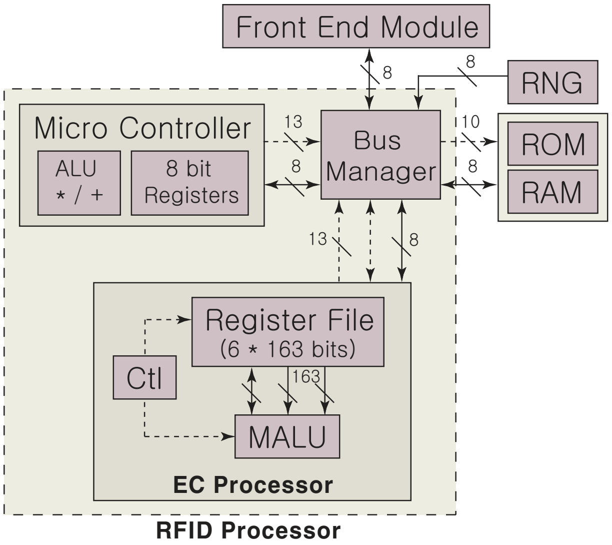


Fig. 5. RFID Processor Architecture [25]

The performance results of EC-RAC using this security processor are summarized in Tab. I. The digit size specifies the digit size of the digit serial MALU. By increasing the digit size, a higher performance can be achieved in the cost of the gate area. The gate area includes everything inside of the dash square in Fig. 5. The number of cycles includes all the computations of EC-RAC, which includes two EC scalar multiplications, two general modular multiplications, two general modular additions, random number generation and the data transmission/reception. The general modular operations, i.e. $r_1 + x_1$ and $r_1x_1 + r_2x_2$, are performed in the micro controller, and these computations can be done in parallel with EC scalar multiplications if there is no data dependency. In EC-RAC, all the general modular operations can be performed during the first EC scalar multiplication, i.e. r_1P, and hence the general modular operations does not contribute to the total number of cycles. At the frequency of 500 KHz, if the digit size is increased to 3 or more, EC-RAC can be finished within 500 ms. 500 ms is a very reasonable response time though it is too much delay for sequential access of multiple tags. However, it is possible to solve the throughput problem by applying a multiple access protocol that can handel multiple tags in parallel. This is possible because the most of the time taken in EC-RAC is caused by the calculation inside of tags and therefore, if we can make multiple tags start the authentication in parallel and the radio communication of each tag exclusive, the overall throughput can be effectively increased.

TABLE I
Performance results of EC-RAC

CMOS	Digit Size	Frequency	Gates	Cycles	Time $[ms]$
$0.13\mu m$	1	500KHz	15,619	554,343	1,109
	2		17,145	291,579	584
	3		17,703	202,753	406
	4		18,262	157,617	316
	5		18,820	131,825	264

The security processor of Fig. 5 needs extra ROM and RAM. RAM stores data and a program. The data stores system parameters, the public key of the server, the private key of a tag and etc, and the program is for the flow of the EC-RAC protocol. RAM is used to store temporary and final results of the computations. The required ROM and RAM sizes are summarized in Tab. II. The small sizes of required memories are due to the specialized processor architecture of [25].

TABLE II
Required memories for EC-RAC

Memory	ROM for program	ROM for data	RAM
Size	58 bytes	126 bytes	128 bytes

VII. Conclusion

We proposed the EC-RAC (ECDLP based Randomized Access Control) protocol. Previously proposed protocols using hash algorithms or symmetric key cryptographic algorithms cannot satisfy the requirements of RFID systems for scalability, tracking and cloning. In addition, well-known ECDLP based authentication protocols are not suitable for RFID systems not only because of the un-solved problem about the secure transmission of a tag's ID but also because of vulnerability against the tracking attack.

The proposed EC-RAC protocol resolves all the requirements mentioned in this paper and is proved for its security in the generic group model. Moreover, the proposed protocol is carefully designed to minimize the computational workload of a tag. We also expect that the results of this work is not limited to RFID systems but can be applied to other authentication applications which are counting the tracking problem.

VIII. ACKNOWLEDGMENTS

This work is supported by NSF CCF-0541472, SRC, FWO and funds from Katholieke Universiteit Leuven.

REFERENCES

[1] M. Burmester, B. Medeiros and R. Motta. Robust, anonymous RFID authentication with constant key-lookup. Cryptology ePrint Archive: listing for 2007 (2007/402), 2007.

[2] EPCglobal. Specification for RFID Air Interface. http://www.epcglobalinc.org.

[3] A. Juels and S. A.Weis. Authenticating pervasive devices with human protocols. In V. Shoup, editor, Advances in Cryptology, Proceedings of CRYPTO 2005, volume 3621 of LNCS, pages 293-308. Springer Verlag, 2005.

[4] R. Anderson and M. Kuhn. Low Cost Attacks on Tamper Resistant Devices. In B. Christianson, B. Crispo, T.M.A. Lomas and M. Roe, editors, Proceedings of the 5th International Workshop on Security Protocols, volume 1361 of LNCS , pages 125-136. Springer Verlag, 1997.

[5] M. Neve, E. Peeters, D. Samyde and J. Quisquater. Memories: a Survey of their Secure Uses in Smart Cards. The 2nd International IEEE Security In Storage Workshop (IEEE SISW'03), pages 62-72, Washington DC, USA, 2003.

[6] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels. Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. The First International Conference on Security in Pervasive Computing (SPC'03), March 2003.

[7] X. Gao, Z. Xiang, H. Wang, J. Shen, J. Huang and S. Song. An Approach to Security and Privacy of RFID System for Supply Chain. Proceedings of the IEEE International Conference on E-Commerce Technology for Dynamic E-Business (CEC-East'04), 2004.

[8] G. Avoine and P. Oechslin. A Scalable and Provably Secure Hash-Based RFID Protocol. The 2nd IEEE International Workshop on Pervasive Computing and Communication Security (Persec'05), March 2005.

[9] M. Ohkubo, K. Suzuki and S. Kinoshita. Cryptographic Approach to "Privacy-Friendly" Tags. RFID Privacy Workshop @ MIT, 2003.

[10] M. Feldhofer. An Authentication Protocol in a Security Layer for RFID Smart Tags. IEEE Mediterranean Electrotechnical Conference (IEEE MELECON'04), May 2004.

[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), volume 3156 of LNCS, pages 357-370. Springer Verlag, 2004.

[12] B. Toiruul and K. Lee. An Advanced Mutual-Authentication Algorithm Using AES for RFID Systems. International Journal of Computer Science and Network Security. VOL.6 No.9B, September 2006.

[13] Y. K. Lee and I. Verbauwhede. Secure and Low-cost RFID Authentication Protocols. In Proc. 2nd IEEE International Workshop on Adaptive Wireless Networks (AWiN), pp. 1-5, 2005.

[14] M. Burmester, T. van Le and B. de Medeiros. Provably secure ubiquitous systems: Universally composable RFID authentication protocols. In Proceedings of the 2nd IEEE/CreateNet International Conference on Security and Privacy in Communication Networks (SECURECOMM 2006). IEEE Press, 2006.

[15] G. Tsudik. YA-TRAP: Yet another trivial RFID authentication protocol. In Proc. IEEE Intern. Conf. on Pervasive Computing and Communications (PerCom 2006). IEEE Press, 2006.

[16] J. Wolkerstorfer. Is Elliptic-Curve Cryptography Suitable to Secure RFID Tags? Workshop on RFID and Light-weight Cryptography, Graz, Austria, August 2005.

[17] P. Tuyls and L. Batina. RFID-tags for Anti-Counterfeiting. In D. Pointcheval, editor, Topics in Cryptology - CT-RSA 2006, volume 3860 of LNCS, pages 115-131. Springer Verlag, February 13-17 2006.

[18] L. Batina, J. Guajardo, T. Kerins, N. Mentens, P. Tuyls, and I. Verbauwhede. Public-Key Cryp-tography for RFID-Tags. In Proceedings of IEEE International Workshop on Pervasive Com-puting and Communication Security , 6 pages, 2007.

[19] C.-P. Schnorr. Efficient Identification and Signatures for Smart Cards. In Gilles Brassard, editor, Advances in Cryptology - CRYPTO '89, volume 435 of LNCS, pages 239-252. Springer Verlag, 1989.

[20] T. Okamoto. Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. In E. F. Brickell, editor, Advances in Cryptology - CRYPTO'92, volume 740 of LNCS, pages 31-53. Springer Verlag, 1992.

[21] V. Shoup. Lower bounds for discrete logarithms and related problems. In W. Fumy, editor, Eurocrypt '97, volume 1233 of LNCS, pages 256-266. Springer, 1997.

[22] U. Maurer. Abstract models of computation in cryptography. In 10th IMA Conference On Cryptography and Coding, volume 2796of LNCS, pages 1-12. Springer-Verlag, 2005.

[23] A. W. Dent. The hardness of the DHK problem in the generic group model. Cryptology ePrint Archive, Report 2006/156.

[24] ANSI X9.42-2003 Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography, American National Standards Insti-tute, 2003.

[25] Y. K. Lee, K. Sakiyama, L. Batina and I. Verbauwhede. Elliptic Curve Based Security Processor for RFID. Submitted to IEEE Transactions on Computers. 2007.