Yet Another Strong Privacy-Preserving RFID Mutual Authentication Protocol

Raghuvir Songhela and Manik Lal Das

DA-IICT, Gandhinagar, India
{songhela_raghuvir, maniklal_das}@daiict.ac.in

Abstract. Radio Frequency IDentification (RFID) systems are gaining enormous interests in industry due to their vast applications such as supply chain, access control, inventory, transport, health care and home appliances. Although tag identification is the primary security goal of an RFID system, privacy issue is equally, even more important concern in the RFID system because of pervasiveness of RFID tags. Over the years, many protocols have been proposed for RFID tags' identification using symmetric key cryptography and other primitives. Many of them have failed to preserve tags' privacy. In order to achieve privacy and to provide scalability and anti-cloning features of RFID system, public-key primitives should be used in an RFID authentication protocol [1]. In this paper, we present a mutual authentication protocol for RFID systems using elliptic curves arithmetic. The proposed protocol provides *narrow-strong* and *wide-weak* privacy under standard complexity assumption.

Keywords: RFID System, Mutual Authentication, Tracking Attack, Elliptic Curve Cryptography, Privacy, Un-traceability.

1 Introduction

Radio Frequency IDentification (RFID) systems have found enormous applications in industry such as supply chain management, access control system, inventory control, transport system, health care, home appliances, object tracking, and so on. An RFID system consists of a set of tags, one or more readers and a back-end server. Typically, all the readers are connected with the back-end server. The communication channel between the readers and the back-end server is assumed to be secure. For simplicity, the reader and the back-end server can be considered as a single entity, we consider it "a reader". A tag is basically a microchip with limited memory along with a transponder. Based on RFID chip capacity, RFID tags can be divided into three types - Active, Passive and Battery-Assisted Passive (Semi-Passive). Passive tags are less expensive and they can be made small enough to fit on almost any product. A passive tag does not have a power source. It only transmits a signal upon receiving RF energy

emitted from a reader in its proximity. Active and semi-passive tags have internal batteries to power their circuits. An active tag uses its battery to broadcast radio waves to a reader, whereas a semi-passive tag gets activated in the presence of an RFID reader and relies on the reader to supply the power for broadcasting the message. A reader is a device used to interrogate RFID tags. The reader consists of one or more transceivers which emit radio waves.

Although tags' authentication is the main goal of RFID system, the system should guarantee that tags are not being tracked by attackers with a motive of compromising privacy of tag-enabled objects. Furthermore, RFID authentication protocols should preserve operational and cryptographic properties like system scalability and security against cloning and tracking attacks. Recent works in RFID authentication protocols suggest that public-key cryptography (PKC) primitives are necessary to address these requirements [2], [1]. In particular, ECC (Elliptic Curve Cryptography) arithmetic is preferred over other PKC algorithms because of its smaller key size and existence of efficient algorithms for elliptic curve arithmetic.

Privacy of tags has become an important issue in the RFID system. Privacy can be termed in two concepts: anonymity and un-traceability [3]. The real ID of a tag must not be known by others to achieve anonymity. To achieve un-traceability, the equality or inequality of two tags must be impossible to ascertain. Therefore, un-traceability is a stronger privacy requirement than anonymity. Several theoretical models have been proposed so far which address the privacy of RFID systems [4], [5], [6], [2]. The privacy model of Vaudenay [2] was one of the first and most complete privacy models that featured the notion of strong privacy. According to [2], if an attacker has access to the result of the tag's authentication (accept or reject) in a reader, he is defined as a wide attacker. Otherwise, he is a narrow attacker. If an attacker is able to extract the tag's secret and still that tag remains active in the set of tags, then he is a strong attacker. If the tag is inactive after the corruption by the attacker then he is a weak attacker. Therefore, a wide-strong attacker is defined as the most powerful.

In this paper, we present a mutual authentication protocol for RFID system using ECC arithmetic, which provides *narrow-strong* and *wide-weak* privacy under standard complexity assumption. We compare the proposed mutual authentication protocol with similar works and show that the protocol is efficient and provides strong privacy in comparison to other protocols.

The remainder of this paper is organized as follows. In section 2, we discuss preliminaries and security and privacy properties of RFID system. In section 3, we review some ECC-based RFID security protocols. In section 4, we present our protocol. We analyze the proposed protocol in section 5. We conclude the paper with section 6.

2 Preliminaries

2.1 Elliptic Curves and Computational Assumptions

An elliptic curve E over a field F is a cubic curve with no repeated roots [7]. The general form of an elliptic curve is $Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_5$, where $a_i \in F$, $i = 1, 2, \cdots, 5$. The set $E(F)$ contains all points $P(x, y)$ on the curve, such that x, y are elements of F along with an additional point called the point at infinity ($\mathcal{O}$). The set $E(F)$ forms an Abelian group under elliptic curve point addition operation with ($\mathcal{O}$) as the additive identity. For all $P, Q \in E(F)$, let F_q be a finite field with order of a prime number q. The number of points in the elliptic curve group $E(F_q)$, represented by $\#E(F_q)$, is called the order of the curve E over F_q. The order of a point P is the smallest positive integer r, such that $rP = \mathcal{O}$. Without loss of generality, the elliptic curve equation can be simplified as $y^2 = x^3 + ax + b \pmod q$, where $a, b \in F_q$ satisfy $4a^3 + 27b^2 \neq 0$, if the characteristic of F_q is neither 2 nor 3. There are mainly three operations on ECC, namely point addition, scalar multiplication of a point and map-to-point operation, which are commonly used in security protocols.

Elliptic Curve Discrete Logarithm Problem: Elliptic Curve Discrete Logarithm Problem (ECDLP) is a standard assumption upon which ECC-based cryptographic algorithm can rely. The ECDLP is stated as: Given two elliptic curve points P and Q $(= xP)$, where x is sufficiently large, finding scalar x is an intractable problem with best known algorithms and available computational resources. x is called the discrete logarithm of Q to the base P.

Decisional Diffie-Hellman (DDH) assumption: Let P be a generator of $E(F_q)$. Let $x, y, z \in_R Z_q$ and $A = xP$, $B = yP$. The DDH assumption states that: The distribution $\langle A, B, C(= xyP)\rangle$ and $\langle A, B, C(= zP)\rangle$ is computationally indistinguishable.

2.2 Security and Privacy properties of RFID System

An RFID system must meet following security and operational properties [2], [8].

Security: Ensuring that fake tags are rejected.
Authentication: Authentication of tag ensures its legitimacy to reader. Depending on application requirement, tags' authentication or tag-reader mutual authentication is achieved in RFID system.
Integrity: Integrity allows a reader to detect data tampering/alteration upon receiving data from a tag. As tag-reader communication takes place over radio waves, RFID security protocol must ensure data integrity property.

Privacy: Ensuring that privacy of legitimate tags is not compromised.
RFID tags are small and thus, can be attached to consumer goods, library books, home appliances for identification and tracking purposes. In case of any misuse (e.g., stolen RFID-enabled items), the reader can trigger an appropriate message to seller/vendor/owner of the item. The use of radio waves makes adversary's task easy for eavesdropping tag-reader communication and thereby, the information relating to the tag is an easy target of the adversary. Furthermore, the tag of an object can be tracked or monitored wherever the object is lying.

Resistance: Ensuring that the protocol is secure against cloning.
If a group of tags share the same secret key and use it for the authentication, then it will be possible for an attacker to clone all tags in the group once any single tag of the group is cracked by him. It can also cause the tracking problem, as the attacker can decrypt the exchanged messages. Therefore, secret information should be pertinent only to a single tag so that an attacker cannot use revealed secret information to clone other tags but the cracked one.

Forward/Backward Un-traceability: Ensuring that the cracked tag cannot be tracked from its past or future sessions.
Suppose, a tag is cracked and the private key of that tag is stolen by an attacker. A protocol satisfies the feature of the forward/backward un-traceability if the attacker is unable to decode the messages of the previous/future protocol runs initiated by the same tag.

3 Related Works

In recent times, many RFID protocols have been devised using public key cryptographic primitives in order to prevent tracking attacks [9], [10], [11], [12]. In particular, elliptic curve cryptography (ECC) [7] has been realized in RFID authentication protocols [13], [12], [8], [14], [15], [16], [17], [18],

[19], [3]. Many RFID protocols use the concept of the Schnorr [20] identification protocol, where, the prover acts as the tag and the verifier acts as the reader. The RFID protocol which is based on the Schnorr protocol might not preserve the privacy of tag, as the goal of the Schnorr protocol is to identify the communicating principal. Lee *et al* [12] proposed an RFID authentication protocol, known as EC-RAC (Elliptic Curve based Randomized Access Control), claiming that it is secure against tracking attack. However, the claim is not correct as shown in [16] and [17]. Subsequently, randomized Schnorr protocol [16], revised EC-RAC [8] (we refer here EC-RAC mutual authentication version only, termed it as EC-RAC-4) have been proposed to eliminate tracking attacks. Later, attacks on revised EC-RAC have been found [21]. Both randomized Schnorr and EC-RAC-4 protocols are *narrow-strong* privacy-preserving, but not *wide-weak* privacy-preserving. Lee *et al* then proposed low-cost untraceable authentication protocols [3] claiming narrow-strong and wide-weak privacy. However, it is found that the protocol in [3] suffers from man-in-the-middle attack [19].

4 The Proposed Protocol

The protocol has two phases – Setup and Authentication. The Setup phase is a one-time computation, configured with tags and reader before they are deployed into the field. The Authentication phase is invoked when tag and reader start communication.

Protocol's Goal and Assumptions: The protocol aims to provide mutual authentication along with *narrow-strong* and *wide-weak* privacy. If there are more than one readers in the RFID system then all the readers share the same private key. If we keep the private keys different then all the tags need to store the public keys of all the readers, which is not preferred. Moreover, privacy is preserved even if the private key is kept same across all the readers. In the protocol, it is assumed that, before mutual authentication, the tag should have the public key of the reader. The reader should also have access to the public key of all tags. In our protocol, we consider active tags who can initiate communication with a reader. We further assume that communicating tags have similar computing resource that we have in contactless smart cards [22].

4.1 Setup phase

Setup phase is implemented only once, before the deployment of the tags and the reader. Let P be the base point of an admissible elliptic curve.

The reader shares its public key Y $(=yP)$ with all the tags and stores its private key y securely with it. Each tag shares its public key X $(=xP)$ with the reader (which gets stored in back-end server) and stores its private key x securely with it.

4.2 Authentication phase

The Authentication phase works as follows.

Tag $\rightarrow$ Reader : r_{t_1}, K, T_1

The tag chooses random numbers k and r_{t_1}. Then it computes

1. $r_s \leftarrow f(r_{t_1}, [kY])$
2. $K \leftarrow kP$
3. $T_1 \leftarrow r_s xY$

Here, $[P]$ indicates the x-coordinate of the Elliptic Curve point P. To avoid the man-in-the-middle attack as shown in [19], the value of k should be different from the multiples of order of Y on the elliptic curve and zero. $f()$ is a cryptographic pseudo-random function. Tag sends r_{t_1}, K, T_1 to the reader.

Reader $\rightarrow$ Tag : T_2

Upon receiving tag's message $< r_{t_1}, K, T_1 >$, the reader first computes $f(r_{t_1}, [yK])$ (say r'_s). It checks whether $T_1 y^{-1} r'^{-1}_s = X$. If it holds, then tag's authentication is confirmed. Reader now computes $T_2 \leftarrow y r'_s K$ and sends it to the tag.

After receiving reader's response, the tag checks whether $T_2 k^{-1} r_s^{-1} = Y$. If it holds, then the reader authentication is confirmed.

In order to get the value of X, the reader requires the access of the list of public keys of all the tags. If the reader finds the derived value matching with any entry in the list, the communicating tag is considered as authentic one. The protocol is depicted in Figure 1.

5 Analysis of the Protocol

5.1 Narrow-Strong Privacy

A narrow attacker does not have access to the **result** of authentication of the tag. It is noted that the outcome of the **result** query is a bit indicating successful/unsuccessful authentication of the tag at the reader side. A strong attacker can corrupt a tag and still that tag remains in the

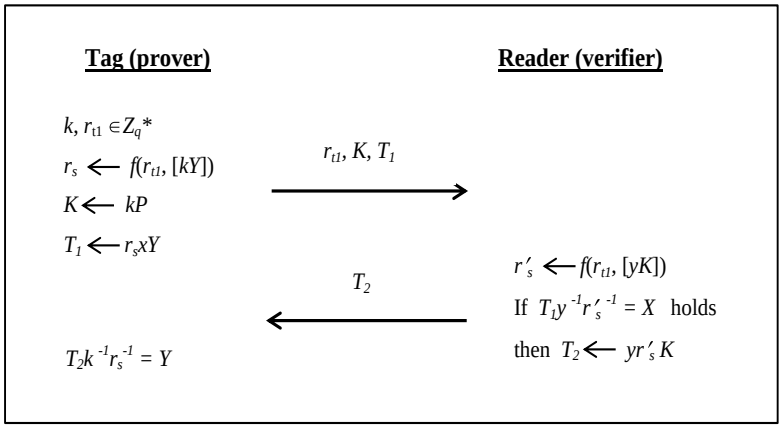

Fig. 1. The Proposed Protocol

set of the valid tags, that is, the tag can communicate with the reader even after it has been corrupted by the attacker. A narrow-strong attacker has properties of narrow attacker and strong attacker both. Suppose, the attacker has cracked tag and has retrieved the private key x of tag. Now, any of the tags starts a new protocol run with the reader. The attacker can manipulate messages sent by this tag. Given the messages sent by this tag, the narrow-strong attacker has to determine whether this tag is the same which is cracked by him or not with the probability significantly greater than $1/2$ to carry a successful attack.

The messages exchanged in our protocol are r_{t_1}, T_1, K and T_2, where K is a random ephemeral elliptic curve (EC) point, r_{t_1} is a random number generated by the tag, and T_2 is a EC point generated by the reader. It is easy to see that these three messages do not include any information about the tag. Message T_1 contains the private key of the tag (x), public key of the reader (Y) and the random number (r_s) which depends on r_{t_1} and k. It is computationally infeasible to link message T_1 with any particular tag, as r_s is a result of one-way pseudo-random function which takes two arguments. Out of these two arguments, r_{t_1} is communicated in plain text form to the reader. However, the attacker can not learn r_s without knowing k. Although $K = kP$, the attacker can not get any clue of k from K, as it is an ECDLP, an intractable problem. As a result, the attacker can not calculate the value of r_s, which is used to calculate T_1. Therefore, even if the attacker knows the private key of a tag, x, it does

not help him in decrypting T_1 as he does not have value of r_s. Therefore, given a private key of any tag and a message set sent by some other tag to the reader, the attacker can not determine if the protocol run was initiated by the corrupt tag or uncorrupt tag.

5.2 Wide-Weak Privacy

The attacks on protocols in [3] observed in [19] use the fact that the reader sends the random number in plain form to the tag, which can be modified by the attacker. In our protocol, we have taken care of this and the protocol provides wide-weak privacy as proved below.
A wide-weak attacker has properties of both, wide attacker and weak attacker. A weak attacker can not corrupt a tag. A wide attacker has one-bit extra information compared to a narrow attacker: the decision of the reader whether to accept a tag or not (result of the tag authentication). This extra bit of information can be used by a wide-weak attacker to perform a tracking attack. The goal of a wide-weak attacker is to determine if two sets of protocol instance originate from the same tag. One of these sets contains authentic messages from the past. We denote the source (i.e. the tag) of these messages by tag A. The other set contains the messages of tag B. The tracking attack is successful when the attacker can determine the (in)equality of these two tags with a probability significantly greater than 1/2.
The attacker has four messages from the protocol run initiated by tag A. We denote them by $r_{t_1}^A$, T_1^A, K^A and T_2^A. We also denote the messages sent by tag B to the reader by $r_{t_1}^B$, T_1^B and K^B. Before the messages from the protocol run of tag B reaches the reader, the attacker can manipulate them. Based on the result of the authentication of tag B, the attacker tries to guess whether both tags are same or not. Both the tags are same if x^A and x^B are same. Note that K^A and K^B are two random points on EC and contain no information about the tag. The same argument applies to $r_{t_1}^A$ and $r_{t_1}^B$ as both of them are random numbers. We now prove that this protocol is wide-weak privacy-preserving by the method of the contradiction. Suppose, the proposed protocol is not wide-weak privacy-preserving and the attacker manipulates messages sent by tag B to the reader and from the result of the tag authentication by the reader, it can determine if tag A and B are equal or not with probability greater than 1/2. Following three scenarios may arise.

Modification in $r_{t_1}^B$: The attacker changes the value of $r_{t_1}^B$ which is sent from the tag B to the reader.

Suppose, the attacker replaces $r_{t_1}^B$ with $r_{t_1}^{'}$. However, he can not pass T_1^B validation at the server end. The reason for the same is r_{t1}^B is used for calculating r_s^B, which in turn is used to calculate T_1^B. But, to calculate r_s^B by its own, the attacker has to retrieve the value of k^B from K^B, which he can not do because of the ECDLP hardness problem.
Now suppose, he selects his own ephemeral random number $k^{'}$, calculates $K^{'}$ and replaces K^B with K'. However, he can not calculate a valid $T_1^{'}$ to replace T_1^B, because T_1' should have involvement of the private key x^B of the tag B. But, the attacker does not have the information of the private key of the tag B. Therefore, the attacker can not generate the valid pair of messages in this case and hence attack is not feasible.

Modification in K^B: The attacker changes the value of K^B which is sent from the tag B to the reader.
Suppose, the attacker does not change the value of r_{t1}^B and keeps it as it was sent originally by the tag B. As mentioned in the previous point, if the attacker tries to replace K^B by selecting his own $K^{'}$, then he has to calculate a valid $T_1^{'}$. However, without knowing the private key of the tag B, he can not calculate a valid $T_1^{'}$, and the attack can not take place.

Modification in T_1^B: The attacker changes the value of T_1^B which is sent from the tag B to the reader.
Suppose, the attacker modifies T_1^B by adding T_1^A or any T_1 message intercepted from the previous run of the protocol. Suppose, the tag A and tag B are same. As tag A and tag B are same, $x^B = x^A$ and the following condition will hold.

$$r_s^B x^B\, Y(=\, T_1^B) + r_s^A x^A\, Y(=\, T_1^A) = (r_s^B + r_s^A) x^B\, Y$$

Now, for successful authentication at the reader end, the attacker has to replace r_{t1}^B by $r_{t1}^{'}$ and/or K^B by $K^{'}$ such that the reader gets the value of r_s as $(r_s^B + r_s^A)$. If the attacker successfully derives these values and if the reader authenticates the tag B then the attacker can conclude that tag A and tag B are same. If the reader does not authenticate the tag B then the attacker can conclude that both the tags are different.
In order to derive the values of the $r_{t1}^{'}$ and/or $K^{'}$, the attacker has to retrieve the value of $(r_s^B + r_s^A)$ from the message which was resulted after addition of two messages, that is, $(r_s^B + r_s^A)\, x^B\, Y$. However, this can not be done, as the attacker has to solve the ECDLP which he can not, with the best available algorithms and resources. Therefore, the attacker can not retrieve the value of $(r_s^B + r_s^A)$, and the attack is not possible. Similarly,

if both the tags are not same then the following condition will hold.

$$r_s^B x^B\, Y(=\, T_1^B) + r_s^A x^A\, Y(=\, T_1^A) = (r_s^B x^B + r_s^A x^A)\, Y$$

Here, the attacker has to replace the values of r_{t1}^B and/or K^B such that the reader gets the value of r_s as $(r_s^B x^B + r_s^A x^A)$. But the attacker can't retrieve the value of $(r_s^B x^B + r_s^A x^A)$ from the $(r_s^B x^B + r_s^A x^A)\ Y$ as it is an ECDLP, an intractable problem. Therefore, in both the cases modification in T_1^B does not help the attacker to carry a successful attack. Our initial assumption stated that the attacker can manipulate the messages sent by the tag B and can break wide-weak privacy. As we have shown above, the attacker is unable to carry out wide-weak attack by manipulating messages. These results show that the initial assumption was false and the proposed protocol provides the wide-weak privacy.

5.3 Forward/Backward Un-traceability

Suppose the attacker cracks the tag and reveals all the information pertinent to that tag. However, the attacker cannot track the tag in the past communications. The tag chooses two random numbers r_{t_1} and k. r_{t_1} is sent in plain text form by the tag to the server and hence accessible to the attacker. However, the attacker cannot decrypt the value of T_1 due to dependence of T_1 on r_s. The r_s is calculated by passing two parameter values to the pseudo-random function, out of which first one r_{t_1} is accessible to the attacker. But, to calculate the second parameter, the attacker has to calculate k from K $(=kP)$, which is ECDLP, an intractable problem. As attacker cannot calculate the value of r_s, he cannot operate inverse functions on T_1 and cannot get clue whether the communication has been originated from the same tag or not. In the similar way, backward un-traceability can be proved in which the attacker cannot track the tag in the future communications. Therefore, the proposed protocol provides both forward and backward un-traceability.

5.4 Anti-cloning and Replay prevention

Cloning is an important issue when an RFID system is relying on group key management. In case of group key, if one tag is cracked then the attacker can forge other tags of the group of the system as all tags within the group use the same key for communication. In our protocol, the attacker is unable to forge the other tags of the system. However, if the attacker crack a tag and retrieve its private key along with the other parameters

pertinent to that tag then he can clone that tag to the system. The protocol also prevents replay attempts, as a new session must be composed of a random number chosen by the tag, which has to be validated by the reader with tag's previous sessions' state stored in it.

5.5 Computational Cost

We provide the computational cost of the protocols in Table 1. The notations used in the Table 1 indicate as follows: PM - Point Multiplication; PA - Point Addition. The low-cost untraceable authentication protocol [3] provides only tag authentication and requires three point multiplications on each side. It requires one point addition on the server side. However, it does not provide mutual authentication. Moreover, the protocol is not wide-weak privacy-preserving [19]. The EC-RAC-4 [8] requires four point multiplication operations on each side. It also requires one point addition on the server side. EC-RAC-4 provides only narrow-strong privacy and not wide-weak privacy. Moreover, EC-RAC-4 is vulnerable to tracking attack [21].
The proposed protocol requires four and three point multiplication on the tag and the reader side, respectively. The protocol doesn't require any point addition operation on either side. However, the pseudo-random function is used on each side to generate a random number from two arguments. When compared to low-cost untraceable authentication protocol [3], the proposed protocol requires one more point multiplication on the tag side and requires pseudo-random function on each side. But, the proposed protocol provides the mutual authentication as well as wide-weak privacy whereas the former one does not. In comparison to EC-RAC-4 [8] (which supports mutual authentication), the proposed protocol takes one less point multiplication and one less point addition operation on the server side. In addition, the proposed protocol provides wide-weak privacy whereas the former does not. However, the proposed protocol requires pseudo-random function computation on each side. Therefore, the proposed protocol is computationally comparable with other protocols, and it also provides wide-weak privacy along with mutual authentication.

5.6 Communication Cost

Table 2 depicts the communication cost of the protocols in terms of the total number of parameters sent by the tag and the reader in one protocol run. The notations used in the Table 2 indicate as follows: m_r - scalar number; m_{ec} - EC point. Low cost untraceable protocol [3] and EC-RAC-4

Performance ⇒ *Protocol* ⇓	Tag side comp.	Reader side comp.
Low-cost untraceable Protocol [3]	3 PM	3 PM + 1 PA
EC-RAC-4 [8]	4 PM	4 PM + 1 PA
Proposed Protocol	4 PM	3 PM

Table 1. Comparison of computational cost

[8] - each consists of three messages in a protocol run. In these protocols, the tag sends two messages and the reader sends one message in a protocol run. Whereas, our protocol is a two-message protocol in which the tag and the reader sends one message each in entire protocol run. EC-RAC-4 [8], the tag sends three EC points; the reader sends one scalar number and one EC point. Whereas, in our protocol, the tag sends two EC points and one scalar number, and the reader sends one EC point only. As a result, our protocol takes less communication cost than EC-RAC-4 [8] (as it is reasonable to assume that the size of the random number is less than size of EC point). Furthermore, the proposed protocol is scalable as the computation amount is fixed and independent of the number of tags.

Comparison ⇒ *Protocol* ⇓	Tag side comm.	Reader side comm.
Low-cost untraceable Protocol [3]	2 m_{ec}	1 m_r
EC-RAC-4 [8]	3 m_{ec}	1 m_r + 1 m_{ec}
Proposed Protocol	1 m_r + 2 m_{ec}	1 m_{ec}

Table 2. Comparison of the communication cost

6 Conclusions

We have proposed a new RFID mutual authentication protocol. The proposed protocol provides wide-weak and narrow-strong privacy with less computational load compared to [8], [3]. The proposed protocol resists to all attacks that occur in EC-RAC variants and other related protocols. The performance analysis provided in tables 1 and 2 showed that the proposed protocol is comparable to related RFID authentication protocols, and it preserves privacy of the tags.

References

1. M. Burmester, B. Medeiros, and R. Motta. Robust Anonymous RFID Authentication with Constant Key Lookup. In proc. of ACM Symposium on Information, Computer and Communications Security (ASIACCS'08), pp.283–291, 2008.
2. S. Vaudenay. On Privacy Models for RFID. In proc. of Advances in Cryptology (ASIACRYPT'07), LNCS 4833, Springer-Verlag, pp.68–87, 2007.
3. Y. K. Lee, L. Batina, D. Singelee, and I. Verbauwhede. Low-cost Untraceable Authentication Protocols for RFID (extended version). In proc. of the ACM Conference on Wireless Network Security (WiSec'10), p.55–64, 2010.
4. G. Avoine. Adversarial Model for Radio Frequency Identification. IACR Cryptology ePrint Archive, Report no.49, 2005.
5. A. Juels and S. Weis. Defining Strong Privacy for RFID. IACR Cryptology ePrint Archive, Report no.137, 2006.
6. C. Ng, W. Susilo, Y. Mu, and R. Safavi-Naini. RFID Privacy Models Revisited. In proc. of European Symposium on Research in Computer Security (ESORICS'08), LNCS 5283, Springer-Verlag, pp.251–266, 2008.
7. D. Hankerson, A. Menezes, and S. Vanstone. Guide to Elliptic Curve Cryptography. Springer, 2004.
8. Y. K. Lee, L. Batina, and I. Verbauwhede. Untraceable RFID Authentication Protocols: Revision of EC-RAC. In proc. of the IEEE International Conference on RFID, pp.178–185, 2009.
9. J. Wolkerstorfer. Is Elliptic-curve Cryptography Suitable to Secure RFID Tags? In proc. of the Workshop on RFID and Light-weight Cryptography, 2005.
10. P. Tuyls and L. Batina. RFID-tags for Anti-counterfeiting. In proc. of Topics in Cryptology (CT-RSA'06), LNCS 3860, Springer-Verlag, pp.115–131, 2006.
11. L. Batina, J. Guajardo, T. Kerins, N. Mentens, P. Tuyls, and I. Verbauwhede. Public-key Cryptography for RFID-tags. In proc. of the IEEE International Workshop on Pervasive Computing and Communication Security (Persec'07), 2007.
12. Y. K. Lee, K. Sakiyama, L. Batina, and I. Verbauwhede. Elliptic Curve based Security Processor for RFID. In: IEEE Transactions on Computer, 57(11):1514–1527, 2008.
13. T. Okamoto. Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. In proc. of Advances in Cryptology (CRYPTO'92), LNCS 740, Springer-Verlag, pp.31–53, 1992.
14. D. Hein, J. Wolkerstorfer, and N. Felber. ECC is Ready for RFID - A Proof in Silicon. In proc. of Selected Areas in Cryptography, LNCS 5381, Springer-Verlag, pp.401–413, 2009.
15. Y. Oren and M. Feldhofer. A Low-resource Public-key Identification Scheme for RFID Tags and Sensor Nodes. In proc. of the ACM conference on Wireless network security, pp.59–68, 2009.
16. J. Bringer, H. Chabanne, and T. Icart. Cryptanalysis of EC-RAC, a RFID Identification Protocol. In proc. of the International Conference on Cryptology and Network Security, LNCS 5339, Springer-Verlag, pp.149–161, 2008.
17. T. Deursen and S. Radomirovic. Attacks on RFID Protocols. IACR Cryptology ePrint Archive, Report no.310, 2008.
18. T. Deursen and S. Radomirovic. EC-RAC: Enriching a Capacious RFID Attack Collection. In proc. of RFIDSEC 2010, LNCS 6370, Springer-Verlag, pp.75–90, 2010.

19. J. Fan, J. Hermans, and F. Vercauteren. On the Claimed Privacy of EC-RAC III. In proc. of the RFIDSec 2010, LNCS 6370, Springer-Verlag, pp.66–74, 2010.
20. C. Schnorr. Efficient Identification and Signatures for Smart Cards. In proc. of Advances in Cryptology (CRYPTO'89), LNCS 435, Springer-Verlag, pp.239–252, 1989.
21. T. Deursen and S. Radomirovic. Untraceable RFID Protocols are not Trivially Composable: Attacks on the Revision of EC-RAC. IACR Cryptology ePrint Archive, Report no.332, 2009.
22. ISO/IEC 14443-4:2008(E), Identification cards – Contactless integrated circuit cards – Proximity cards – Part 4: Transmission protocol. Retrieved September 2013, https://www.iso.org/obp/ui/#iso:std:iso-iec:14443:-4:ed-2:v1:en