Wide-weak Privacy Preserving RFID Mutual Authentication Protocol

Raghuvir Songhela Manik Lal Das

DA-IICT, Gandhinagar, India.

{songhela_raghuvir, maniklal_das}@daiict.ac.in

Abstract

Radio Frequency IDentification (RFID) systems are gaining enormous interests at 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 RFID system because of pervasiveness of RFID tags. Over the years, many protocols have been proposed for RFID tags' identification using different cryptographic primitives. It has been observed that most of them provide tags' identification, but they fail to preserve tags' privacy. It has been also proven that public-key primitives are essential for strong privacy and security requirements in RFID systems. In this paper, we present a mutual authentication protocol for RFID systems using elliptic curves arithmetic. Precisely, the proposed protocol provides *narrow-strong* and *wide-weak* privacy and resists tracking attacks under standard complexity assumption. The protocol is compared with related works and found efficient in comparison to others.

Keywords. RFID system, RFID security, Privacy, Tracking attack, Elliptic curve cryptography.

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. It can also be used to discriminate between counterfeits and authentic products. In fact, RFID system is expected to replace the bar code system in near future. An RFID system consists of a set of tags, one or more readers and a back-end database. The communication channel between the back-end database and the reader is assumed to be secure. A tag is basically a microchip with limited memory along with a transponder. Every tag has a unique identity, which is used for its identification purpose. Several tags communicate with a single reader in RFID system. Based on chip capacity and cost factor, RFID tags can be of three types - active, semi-passive and passive. Active and semi-passive RFID 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 relies on a reader to supply its power for broadcasting. A passive tag does not have a power source and only transmits a signal upon receiving RF energy emitted from a reader in proximity of the tag. A reader is a device used to interrogate RFID tags. The reader also consists of one or more transceivers which emit radio waves by which passive tags respond back to the reader. The back-end server is assumed to be a trusted server that maintains tags' and readers' information in its database. Although tags' identification (or authentication) is the main goal of an RFID system, the system should also guarantee that tags are not being tracked by attackers with a motive of compromising privacy of tag-enabled objects. As a result, the precise goal of RFID system is identification/authentication of a tag to the reader without revealing its identity information in communicating message.

In conventional authentication protocols, communicating parties' identity information is transmitted to each other in plain text form. This can cause tracking attack if it is applied to the RFID authentication protocol. In RFID authentication protocol, tag's communication should be randomized in order to avoid tracking attacks. In addition, RFID authentication protocol should provide resistance against cloning attacks, replay attacks and impersonation attacks.

The privacy model of Vaudenay [1] was one of the first and most complete privacy models that featured the notion of strong privacy. According to [1], if an attacker has access to the result of the authentication (accept or reject) in a server, he is defined as 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. Hence, a wide-strong attacker is defined as the most powerful. The protocol is said to be wide-strong privacy-preserving, if it is untraceable against a wide-strong attacker.

Security addresses the correctness and soundness of a protocol, whereas privacy addresses the resistance against unauthorized identification, tracking or linking tags [19]. Privacy can be termed in two concepts: anonymity and untraceability. The real ID of a tag must be unknown to achieve anonymity. To achieve untraceability the equality or inequality of two tags must be impossible to ascertain. Therefore, untraceability is a stronger privacy requirement than anonymity.

In this paper, we present a mutual authentication protocol for RFID system using ECC (Elliptic Curve Cryptography). The proposed protocol provides *narrow-strong* and *wide-weak* privacy under standard complexity assumption. The protocol is compared with related works and is found efficient in comparison to others.

The remainder of this paper is organized as follows. In section 2, we give some preliminaries. In section 3, we discuss some ECC-based RFID security protocols and their merits and limitations. In section 4, we present our protocol. We analyze security of the proposed protocol followed by performance result in section 5 and 6, respectively. We conclude the paper with section 7.

2 Preliminaries

2.1 Elliptic Curve Cryptography

An elliptic curve E over a field F is a cubic curve with no repeated roots [2]. 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 protocol.

2.1.1 Operations on ECC

- **Point addition:** It is the addition of two points P, Q on an elliptic curve to obtain another point R on the same elliptic curve. The line joining of points P, Q intersects the curve at another point, call $-R$. The point $-R$ is reflected in the x-axis to the point R. i.e., $P + Q = R$. This is an interesting feature of elliptic curves and one has to choose a suitable elliptic curve to obtain an elliptic curve group of order sufficiently large to accommodate cryptographic keys.

- **Scalar multiplication of a point:** For a scalar n, multiplication of a curve point P by n is defined as n-fold addition of P, i.e., $nP = P + P + \cdots + P$ (n-times).

- **Map-to-point:** Map-to-point is an algorithm for converting an arbitrary bit string into an elliptic curve point. Firstly, the string has to be converted into an integer and then a mapping is required from that integer onto an elliptic curve point. There are fast algorithms for computation of scalar multiplication of a point and map-to-point operation.

Typically, in ECC private key is a random number (i.e. k) and the corresponding public key is a point on the curve i.e. K=kP. The main advantage of ECC is its small key size. A 160-bit key in ECC is considered to be as secured as 1024-bit key in RSA.

2.1.2 Computational Assumptions

- **Elliptic Curve Discrete Logarithm Problem (ECDLP):** ECDLP is a standard assumption upon which ECC-based cryptographic algorithm can rely upon. 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 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 $< A, B, C(= xyP) >$ and $< A, B, C(= zP) >$ is computationally indistinguishable.

2.2 Security and Privacy properties of RFID System

An RFID system must meet following security and privacy goals [1], [3], [4], [5].

- **Security: Ensuring that fake tags are rejected.**

  - Identification: Identification of a tag ensures its legitimacy to a reader. Depending on application requirement, tags' identification or tag-reader mutual identification is achieved in RFID system.
  - Integrity: Integrity allows a reader to detect data tampering/alteration upon receiving data from sender. 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.** Privacy is related to anonymity and untraceability. 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. Privacy deals with the issues related to resistance against unauthorized access and tracking of tags, whereas security checks that how sound the protocol is. The privacy issue can be categorized into following:

  - Object Privacy: 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. The response of the tag to the query of the reader should not be fixed but it should be randomized so that the attacker can not extract any information from the exchanged messages of the protocol. It should be infeasible for the attacker to determine whether two tags are same or not.

  - Location Privacy: The tag of an object can be tracked or monitored wherever the object is lying, as the tag-embedded object carries information about the object, object owner, manufacturer, and so on.

- **Resistance to impersonation attacks and cloning.**
  - Impersonation attack: If the attacker does not know the secret information of a tag then he should not be able to generate a valid set of messages which can pass the authentication process.
  - Cloning: If the attacker cracks any tag of the system and retrieves all the information from that tag, then the attacker should not be able to forge other tags of the system except the cracked one. If a group of tags share the same secret key and use it for the authentication, then it will be possible to clone all tags in the group once any single tag of the group is cracked. It can also cause the tracking problem since the attacker can decrypt the exchanged messages. Therefore, secret information should be pertinent only to a single tag so that the attacker can't use revealed secret information to clone other tag.

3 ECC-based RFID Authentication Protocols

Initially, hash algorithms and secret key cryptographic algorithms were given more importance in RFID system as they have cheap implementation cost in comparison to public key based algorithms. Although hash-based protocols provide efficiency, but most of them lack scalability property, as the computational workload of the protocol increases linearly as the number of the tags added to the system. Furthermore, protocols for RFID system are divided into two categories - fixed access control and randomized access control. A tag replies to a reader with a fixed message in the fixed access control and hence is vulnerable to tracking attack.
ECC (Elliptic Curve Cryptography) based algorithm is preferred over other PKC (Public Key Cryptography) based algorithm due to its small key size and other interesting features. In RFID authentication protocols, the workload on the tag side should be as minimum as possible. It may result in an increase of workload on the reader side, but the reader has enough resources of power in comparison to the tag. In recent years, many RFID protocols have been proposed using PKC in order to prevent tracking attacks [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19]. Most of these protocols use the concept of the Schnorr [20] identification protocol, in particular, prover as tag and verifier as reader. Lee et al. [9] proposed an RFID authentication protocol, known as EC-RAC (Elliptic Curve based Randomized Access Control), using ECC to address the vulnerability present in the Schnorr protocol. This protocol uses two private keys x_1 and x_2 for a tag, which are linked to its identity and password. The corresponding public keys are $X_1 = x_1P$ and $X_2 = x_2P$. The reader stores y, x_1, X_1 and X_2, whereas the tag stores x_1, x_2 and Y. Although it is claimed [9] that EC-RAC is secure against tracking attack, the claim is

not correct as shown in [14] and [15]. If an attacker impersonates a reader and sends the same random number twice to a tag then the attacker will get two set of messages. As a result, the attacker can retrieve a value which is pertinent to the tag and hence tracking attack is successful. The randomized Schnorr protocol was proposed as a replacement of EC-RAC in [14]. Subsequently, revised EC-RAC [11] has been suggested on EC-RAC to eliminate tracking attacks. Revised EC-RAC also supports reader authentication. The security proofs of revised EC-RAC [11] are implied by means of cryptographic reductions, that is, they are based on the security of the Schnorr protocol and the hardness of the Decisional Diffie-Hellman problem. However, attack on revised EC-RAC was found [16]. The attacker uses previous legitimate execution of the protocol and modifies the messages of the current run of the protocol. If the reader authenticates a tag then the attacker can identify the tag. Hence, revised EC-RAC and randomized Schnorr protocol both are narrow-strong privacy preserving, but not wide-weak privacy-preserving.

Lee *et al* then proposed low-cost untraceable authentication protocols [19] claiming both narrow-strong and wide-weak privacy. As in revised EC-RAC, protocols in [19] are reduced to the security strength of the Schnorr protocol. However, it is found that the protocol in [19] suffers from man-in-the-middle attack [18]. It is shown that the protocol is not even *wide-weak* privacy preserving. The highest possible privacy level that is achieved by [19] scheme is *narrow-strong* privacy [18].

The main drawback of EC-RAC versions [9], [11], [19] was adopting the notion of the Schnorr's protocol for privacy preserving. This can not be achieved because the Schnorr's protocol is not designed for privacy of the prover, rather it is mainly for identification of the prover.

4 Proposed Protocol

The protocol aims to provide mutual authentication along with preserving wide-weak privacy. The protocol has two phases - Setup and Identification. The Setup phase is a one-time computation, configured with tags and reader before they are deployed into the field. The Identification phase is invoked as and when tag and reader start communication. In our protocol, we consider active tags who can initiate communication with a reader. We assume that a tag can have similar computing resource that we have in contactless smart cards. Indeed, this type of tags are costly in comparison to passive tags. However, in near future we can see applications (e.g., health care, home appliances) which require resourceful tags which can compute and communicate to reader with their own energy.

Before we explain the phases in our protocol, we introduce some notation which we use throughout this paper. We denote P as the base point of the Elliptic Curve Group. Server's private-key and public-key pair is represented by y and Y $(= yP)$. Here, y is a scalar value and yP denotes the point derived by the scalar multiplication operation on the Elliptic Curve Group. Tag's private-key and public-key pair is represented by x and X $(= xP)$. Here, x and X $(= xP)$

are denoted as tag's *ID* and tag's *ID-Verifier*. Therefore, tag's public key is not publicly available, it is stored in reader's database securely instead.

4.1 Setup phase

Setup phase is implemented only once, before the deployment of the tags and the reader. The reader shares its public key (Y) with all the tags and stores its private key (y) securely with it. Each tag shares its public key (X) only with the reader and stores its private key (x) securely with it. Each tag also shares its first value of random variable (r_{t_1}) with the reader. The reader stores the first value of random variable of all the tags securely in its memory. Also, all the tags and the reader agree on a base point (P).

4.2 Identification phase

Identification phase works as follows.

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

The tag chooses two random numbers k, r_{t1} and computes
$K = kP$
$r_s = f(r_{t_1}, [kY])$
$T_1 = r_s xY$
Here, [P] indicates the x-coordinate of the EC point P, and $f()$ is a cryptographic pseudo-random function.
Tag sends r_{t_1}, K, T_1 to the reader.

Reader $\longrightarrow$ Tag : T_2

Upon receiving tag's message, the reader first confirms the freshness of received message by validating r_{t_1} with its stored value of r_{t_1} from the previous run with that tag. Initially, the value of r_{t_1} in the reader is set to the number which was shared by a tag to the reader in the setup phase. To pass the check at the server's end, the value of r_{t_1} received in the current protocol run should be greater than the value used in the previous run. i.e., the stored value of r_{t_1} in the reader for that particular tag. If the check is passed then the reader computes $r'_s = f(r_{t_1}, [yK])$, then checks whether $T_1 y^{-1} r'^{-1}_s == X$. If it holds, then tag's authentication is confirmed. Reader now computes $T_2 = yr'_s K$ 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.
The protocol is depicted in Figure 1.

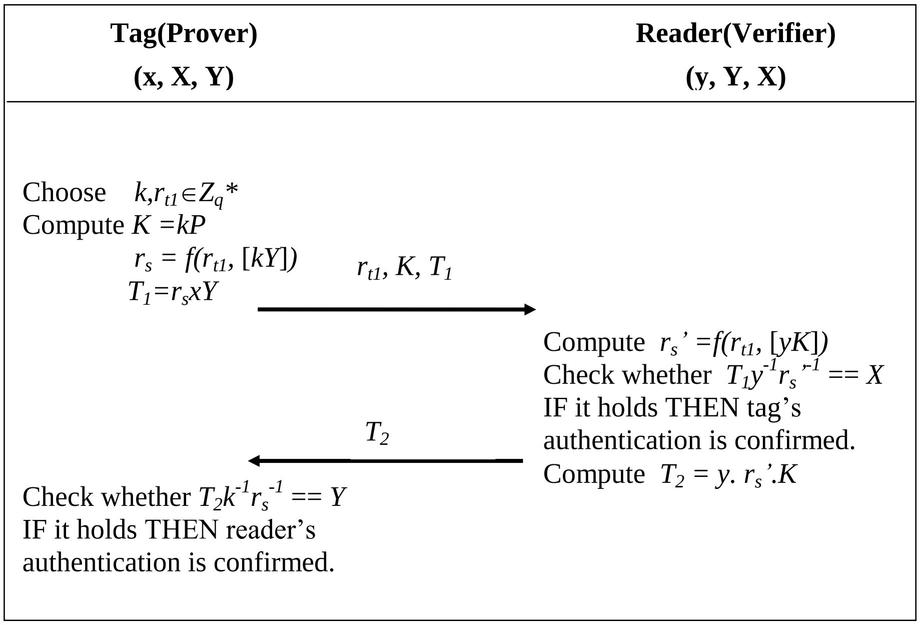
Tag(Prover)
(x, X, Y)
Reader(Verifier)
(y, Y, X)
Choose $k, r_{t1} \in Z_q^*$
Compute $K = kP$
$r_s = f(r_{t1}, [kY])$
$T_1 = r_s x Y$
r_{t1}, K, T_1
Compute $r_s' = f(r_{t1}, [yK])$
Check whether $T_1 y^{-1} r_s'^{-1} == X$
IF it holds THEN tag's authentication is confirmed.
T_2
Compute $T_2 = y. r_s'.K$
Check whether $T_2 k^{-1} r_s^{-1} == Y$
IF it holds THEN reader's authentication is confirmed.

Figure 1: The proposed protocol

5 Analysis

5.1 Privacy of the protocol

5.1.1 Narrow-Strong Privacy

A narrow attacker does not have access to the **result**[1] of authentication of the tag. A strong attacker can corrupt a tag and still that tag remains in the 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 A and has retrieved the private key x^A of tag A. Now, any of the tags starts a new protocol run with the reader and the attacker can manipulate messages sent by this tag. However, as the attacker is narrow, he does not have access to the **result** of the tag authentication. 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 EC point, r_{t_1} is a random number generated by the tag, and

[1] The outcome of the **result** query is a bit indicating successful/unsuccessful authentication of the tag at the reader side.

T_2 is a EC point generated by the reader. It is easy to see that these three messages do not include 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 an clue of k from K, as it is 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 set was initiated by the corrupt tag or the uncorrupt tag.

5.1.2 Wide-Weak Privacy

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. So, wide attacker has access to the **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 instances 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 a 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, 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 to 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 a 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 $r^B_{t_1}$ with $r^{'}_{t_1}$. In order to do this the attacker has to choose $r^{'}_{t1}$ in such a manner that $r^{'}_{t1} \geq r^B_{t1}$. If the attacker provides the smaller value then the tag authentication will get failed at the reader end. If the attacker selects a higher value of $r^{'}_{t1}$ then he can not pass the T^B_1 validation. The reason to that is r^B_{t1} is used for calculating the r^B_s, which in turn is used to calculate T^B_1. But, to calculate r^B_s 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^B_1, because T'_1 should have involvement of the private key x^B 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^B_{t1} and keep 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 $r^{'}_s$. However, without knowing the private key of the tag-B, he can not calculate a valid $r^{'}_s$, and the attack can not take place.

- **Modification in T^B_1: The attacker changes the value of T^B_1 which is sent from the tag-*B* to the reader.**
  Suppose, the attacker modifies T^B_1 by adding T^A_1 or any T_1 message intercepted from the previous run of the protocol. Suppose, the tag-A and tag-B are same so x^B and x^A will be same. As tag-A and tag-B are same, the following condition will hold.
  $r^B_s x^B Y \ (=T^B_1) + r^A_s x^B Y \ (=T^A_1) = (r^B_s + r^A_s) \ x^B Y$
  Now, for successful authentication at the reader end, the attacker has to replace r^B_{t1} by $r^{'}_{t1}$ and/or K^B by $K^{'}$ such that the reader gets the value of r_s as $(r^B_s + r^A_s)$. 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.
  In order to derive the values of the $r^{'}_{t1}$ and/or $K^{'}$, the attacker has to retrieve the value of $(r^B_s + r^A_s)$ from the message which was resulted after addition of two messages, that is, $(r^B_s + r^A_s) \ x^B Y$.
  However, this can not be done, as the attacker has to solve the ECDLP which he can not with best available algorithms and resources. Therefore, the attacker can not retrieve the value of $(r^B_s + r^A_s)$, and the attack is not possible in our protocol.

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 message. These results show that the initial assumption was false and the proposed protocol provides the wide-weak privacy.

5.2 Security of the protocol

5.2.1 Security Against Replay Attacks

In our protocol, the random number r_{t_1} is used to avoid replay attacks. The reader stores the number r_{t_1} for a tag after successful run of the protocol till the next session with the tag. Initially, this number is set to the number which was shared by a tag to the reader in the setup phase, and both tag and reader follow to a predefined rule of r_{t_1}'s acceptance. The tag communicates the value of r_{t_1} to the reader, who then verifies it by its previous value stored in some temporary memory. Once the current r_{t_1} is accepted at the reader, it stores this value for next run of the protocol from the same tag. If the value of the r_{t_1} is changed by an adversary then the tag authentication fails at the server end, because other parameters involve r_{t_1} along with tag's secret key. Suppose, the adversary picks his own random numbers r_{t_1} and k. In order to calculate r_s, he needs the value of private key of the tag. As the adversary does not know private key of the tag, he can not generate corresponding T_1. Therefore, the protocol successfully resists replay attacks.

5.2.2 Anti-cloning

ECC is public key cryptography which ensures anti-cloning. Here, each tag has its own private key. Cloning is an important issue when we use group key. In case of group key, if one tag is cracked then the attacker can forge other tags to the system. In the protocol which uses ECC, the attacker is unable to forge the other tags of the system but the cracked one.

6 Efficiency

The mutual authentication protocol of revised EC-RAC protocol [11] requires four point multiplications on each side (tag and server). It also requires one point addition on the server side. Also, the protocol is vulnerable to tracking attack and hence provides only narrow-strong privacy. Low-cost untraceable authentication protocol [19] 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. Also, the protocol is not wide-weak privacy preserving. The highest privacy preserved by it is narrow-strong. The proposed protocol requires four EC point multiplications on the tag side and three point multiplications on the server side. It requires no EC point addition on any side. Also, the proposed protocol uses only two messages for a complete run. It performs better than the revised EC-RAC mutual authentication protocol as it is wide-weak privacy preserving. It also performs better than the [19] tag authentication protocol as it provides mutual authentication with less communication cost and wide-weak privacy. Therefore, the proposed protocol is more efficient compared to others. Furthermore, the proposed protocol is scalable as the computation amount is fixed and independent of the number of tags. It also fulfils the property of anti-cloning as each tag has a secret key pertinent to it. The comparison is provided in Table 1.

Performance ⇒ *Protocol* ⇓	Tag side computation	Server side computation
EC-RAC I [9]	2M	3M + 2A
Revised EC-RAC with mutual authentication [11]	4M	4M + 1A
Low-cost untraceable EC-RAC [19]	3M	3M + 1A
Proposed mutual authentication protocol	4M	3M

Table 1: Performance of the protocols

M: scalar multiplication to an elliptic curve point
A: point addition

7 Conclusion

We have proposed a new RFID mutual authentication protocol. The proposed protocol provides wide-weak and narrow-strong privacy under the standard complexity assumption. The protocol is efficient in comparison to other related protocols.

References

[1] S. Vaudenay. On privacy models for RFID. *In Advances in Cryptology (ASIACRYPT'07), LNCS*, volume 4833, pages 68–87. Springer-Verlag, 2007.

[2] D. Hankerson, A. Menezes, and S. Vanstone. Guide to Elliptic Curve Cryptography. Springer, 2004.

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

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

[5] C. Ng, W. Susilo, Y. Mu, and R. Safavi-Naini. RFID Privacy Models Revisited. In *European Symposium on Research in Computer Security, LNCS*, volume 5283, Springer-Verlag, pp. 251–266, 2008.

[6] J. Wolkerstorfer. Is Elliptic-Curve Cryptography Suitable to Secure RFID Tags? In *Workshop on RFID and Light-weight Cryptography*, 2005.

[7] P. Tuyls and L. Batina. RFID-tags for Anti-Counterfeiting. In *D. Pointcheval, editor, Topics in Cryptology, LNCS*, volume 3860, pp. 115–131. Springer-Verlag, 2006.

[8] L. Batina, J. Guajardo, T. Kerins, N. Mentens, P. Tuyls, and I. Verbauwhede. Public-Key Cryptography for RFID-Tags. In *IEEE International Workshop on Pervasive Computing and Communication Security*, 2007.

[9] Y. K. Lee, L. Batina, I. Verbauwhede. EC-RAC (ECDLP based randomized access control): Provably secure RFID authentication protocol. In *IEEE Conference of RFID*, pp. 97–104, 2008.

[10] 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, 2008.

[11] Y. K. Lee, L. Batina, and I. Verbauwhede. Untraceable RFID Authentication Protocols: Revision of EC-RAC. In proceedings of the *IEEE International Conference on RFID*, pp. 178–185. 2009.

[12] D. Hein, J. Wolkerstorfer, and N. Felber. ECC is Ready for RFID - A Proof in Silicon. In *Selected Areas in Cryptography, LNCS*, volume 5381, pages 401–413, 2009.

[13] Y. Oren and M. Feldhofer. A low-resource public-key identification scheme for RFID tags and sensor nodes. In proceedings of the second ACM conference on Wireless network security, pp. 59–68, 2009.

[14] J. Bringer, H. Chabanne, and T. Icart. Cryptanalysis of EC-RAC, a RFID Identification Protocol. In *proceedings of the International Conference on Cryptology and Network Security, LNCS*, volume 5339, pp. 149–161, 2008.

[15] T. Deursen and S. Radomirovic. Attacks on RFID Protocols. In Cryptology ePrint Archive: listing for 2008 (2008/310), 2008.

[16] T. Deursen and S. Radomirovic. Untraceable RFID protocols are not trivially composable: Attacks on the revision of EC-RAC. In Cryptology ePrint Archive: Report 2009/332, 2009.

[17] T. Deursen and S. Radomirovic. EC-RAC: Enriching a Capacious RFID Attack Collection. In *proceedings of RFIDSec 2010, LNCS*, volume 6370, pp. 75–90, 2010.

[18] J. Fan, J. Hermans, and F. Vercauteren. On the Claimed Privacy of EC-RAC III. In *proceedings of the RFIDSec 2010, LNCS*, volume 6370, pp. 66–74, 2010.

[19] Y. K. Lee, L. Batina, D. Singelee, and I. Verbauwhede. Low-Cost Untraceable Authentication Protocols for RFID (extended version). In *Proceedings of the 3rd ACM Conference on Wireless Network Security (WiSec)*, pp. 55-64, 2010.

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

[21] M. Burmester, B. Medeiros, and R. Motta. Robust Anonymous RFID Authentication with Constant Key Lookup. In *ACM Symposium on Information, Computer and Communications Security.* ACM, 2008.

[22] A. Juels and S. Weis. Authenticating pervasive devices with human protocols. In *Proceedings of CRYPTO'05, LNCS*, volume 3126, Springer-Verlag, pp. 293–308, 2005.