Strong Security and Privacy of RFID System for *Internet of Things* Infrastructure

Manik Lal Das

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

Abstract. RFID (Radio Frequency IDentification) system has found enormous applications in retail, health care, transport, and home appliances. Over the years, many protocols have been proposed for RFID security using symmetric key and public key cryptography. Based on the nature of RFID tags' usage in various applications, existing RFID protocols primarily focus on tag identification or authentication property. *Internet of Things* (IoT) is emerging as a global network in which every object in physical world would be able to connect to web of world via Internet. As a result, IoT infrastructure requires integration of several complimentary technologies such as sensor networks, RFID system, embedded system, conventional desktop environment, mobile communications and so on. It is prudent that RFID system will play significant roles in IoT infrastructure. In that context, RFID system should support more security properties, such as mutual authentication, key establishment and data confidentiality for its wide-spread adoption in future Internet applications. In this paper, we present a strong security and privacy of RFID system for its suitability in IoT infrastructure. The proposed protocol provides following security goal:

- mutual authentication of tags and readers.
- authenticated key establishment between tag and reader.
- secure data exchange between tag-enabled object and reader-enabled things.

The protocol provides *narrow-strong* privacy and forward secrecy.

Keywords: Internet of Things, RFID security, identification, security, privacy, elliptic curves.

1 Introduction

Internet of Things (IoT) is envisioned as a general evolution of the Internet "from a network of interconnected entities to a network of interconnected objects" [1]. In IoT infrastructure, all physical objects (e.g. human, home appliances, vehicles, chemical reactors, consumer goods) would be able to interact to web of world with the help of software, hardware, and virtual entities through Internet, Bluetooth, and/or Satellite. A high-level view of IoT scenarios and applications is depicted in Figure 1. It is prudent that IoT infrastructure requires

B. Gierlichs, S. Guilley, and D. Mukhopadhyay (Eds.): SPACE 2013, LNCS 8204, pp. 56–69, 2013.

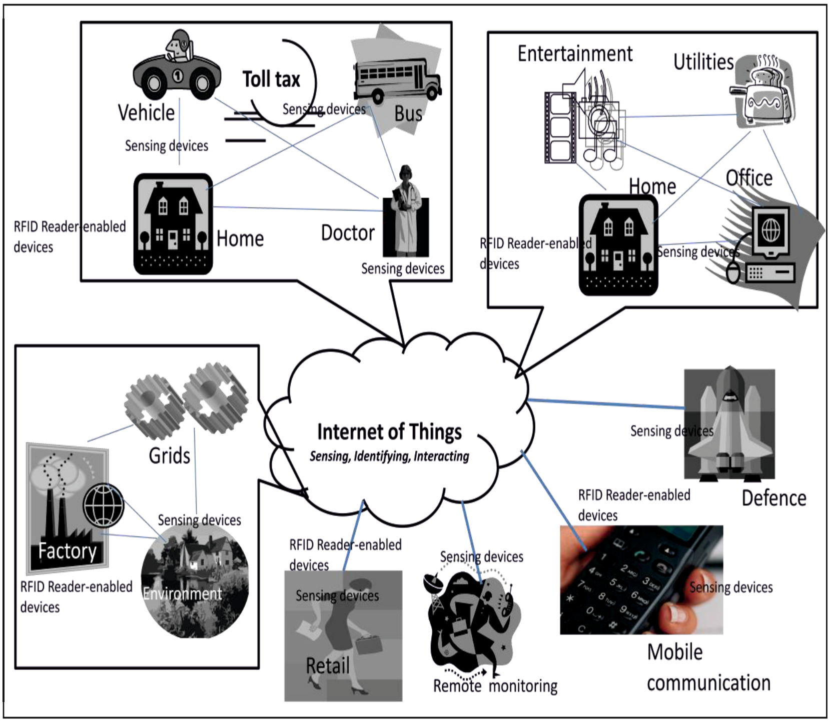

Fig. 1. *Internet of Things* scenarios and applications

integration several complimentary technologies such as sensor networks, RFID system, embedded system, conventional LAN setup with client-server environment, Grid system, cloud computing, mobile communications and so on. In order to address its core issues, IoT infrastructure should resolve some major challenges [2], as listed below.

- Standards: Standards and specifications by international forums are the foremost requirement in order to see IoT in its desired shape. Although European communities have been investing significant efforts towards IoT mission, a collective effort by IEEE, NIST, ITU, ISO/IEC, IETF could probably make this mission effective, implementable and deliverables.
- Identity management: While integrating trillions of objects in IoT infrastructure, managing identities would become a major task in IoT. Both addressing and uniqueness issues have to be resolved suitably. Some existing technologies such as smart cards, RFID tags [3], IPv6, are going to play important roles for identifying objects in IoT infrastructure.
- Privacy: One of the major challenges in global acceptance of IoT is the privacy of objects. The privacy issue involves object privacy, location privacy,

and human privacy. Indeed, object privacy and location privacy can link to human privacy.

- Security: In IoT, the primary means of communication is based on public channel like Internet. Therefore, IoT applications must be safeguarded from both passive and active attackers. In addition, IoT infrastructure needs substantial security measures for Intranet security, data security, system security, hardware security, and physical security.
- Trust and Ownership: IoT infrastructure should support interaction among hosts, intermediate systems and end-entity devices. As a result, trust relationship among entities is a key factor that should exist implicitly or explicitly. At the same time, data ownership is an important concern when one system relies on other system in order to serve some tasks.
- Integration: The main hurdle of IoT infrastructure is the integration of heterogeneous technologies and devices ranging from physical world to web of world. The factors that link to integration issue are computation, bandwidth, storage, interoperability and security.
- Scalability: IoT has a wider spectrum than the conventional Internet-based computing system. Therefore, basic functionalities like communication and service discovery need to function efficiently in both small scale and large-scale environments.
- Regulation: In order see IoT in its desired shape, regulatory issues are the key implementation issues for application and software that use public and/or proprietary technology. Every country has its own Information Technology Act and one could enforce certain regulatory norms before allowing a party to implement some application that has larger interest to its citizens. Roughly speaking, this is perhaps the most crucial and complex agenda in many countries in order to agree or disagree on IoT's adoption for future Internet applications.

RFID system will contribute significantly in IoT infrastructure. An RFID system consists of a set of tags, readers and a back-end server. 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. 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. In IoT, RFID-enabled things require to talk to other things such as sensors, mobile devices and embedded systems through RFID reader-enabled capability (assume that other devices are RFID reader-enabled). As a result, security is an important concern when RFID-enabled things interact with other system (and vice-versa). In that context, in addition to identification of RFID tags, reader authentication, key establishment and data confidentiality are to be addressed suitably in RFID system for secure integration of them into IoT scenarios and applications.

In recent times, many security protocols have been proposed for RFID system; however, most of them discuss about tags identification and tracking issues based on the nature of RFID applications. Feldhofer *et al* [4] proposed a protocol based on challenge-response mechanism using block cipher. A family of HB protocols and improvements have been appeared in literature [5], [6], [7], [8]. Subsequently, many protocols using hash function or symmetric-key algorithm have been proposed for RFID security [9], [10], [11], [12], each having specific security and privacy properties. After Vaudenay's [13] remark on the privacy notion of RFID system, public key cryptographic primitive, specifically elliptic curve cryptography (ECC) [14], is being realized for RFID security [15], [16],[17], [18], [19], [20], [21], [22], [23]. Recent progresses on RFID security are fast and we refer interested readers to [24] for a comprehensive list of recent developments in RFID security.

Our Contributions. RFID security in the context of IoT needs to evolve further to support additional security properties such as mutual authentication, key establishment and data confidentiality. However, the requirement of security service(s) depends on applications where RFID system is going to act. In this paper we present a protocol for RFID security using ECC in the context of IoT scenarios and applications. The protocol aims to provide following security goals.

- mutual authentication of tags and readers.
- key establishment between tag and reader.
- secure data exchange between tag-enabled object and reader-enabled things.

The protocol provides *narrow-strong* privacy and forward secrecy.

Organization of the Paper. The remainder of the paper is organized as follows. In Section 2 we give some perspectives of IoT, RFID security and elliptic curves arithmetic. In Section 3 we present our protocol. In Section 4 we analyze the protocol. We conclude the paper in Section 5.

2 Preliminaries

In this section we briefly discuss some preliminaries of RFID security and privacy issues with a brief review of elliptic curves arithmetic and some standard computational assumptions.

2.1 Security and Privacy Challenges in RFID System

An RFID system aims to achieve following attributes.
Security: Ensuring that fake tags are rejected.

- Identification: Identification of tags ensures its legitimacy to reader. Depending on application requirement, tags' identification or tag-reader mutual identification is achieved in RFID system.
- Integrity: Integrity allows receiver 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.

Data confidentiality and Key establishment: Data confidentiality aims to prevent unauthorized data access. Data confidentiality under dynamic session key is a classic approach for its additional security measures like forward secrecy, data unlinkability. Even though these properties are not required in the present scenarios of RFID system, when we integrate RFID system in IoT then RFID system may demand session key establishment and data confidentiality for future Internet applications.

Privacy: Ensuring that legitimate tags are protected from compromising privacy.

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 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 to the adversary.
- 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.

In addition to above two privacy concerns, human privacy comes into picture in RFID system. On one hand, person who carries tag-enabled item will be tracked by the reader, which could compromise person's privacy. On the other hand, RFID tags' can trace tag-enabled criminals or suspicious objects in a controlled way, which could save money, national assets and human lives.

2.2 Elliptic Curves Arithmetic

An elliptic curve E over a field F is a cubic curve [14] with no repeated roots. 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 prime 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 curve is determined by calculating $q + 1 - t$, where $|t| \leq \sqrt{q}$ is the trace of the Frobenius [14]. The order of a point $P \in E(F_q)$ 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 two operations on ECC, point addition and scalar multiplication of point, which are commonly used ECC-based security protocol.

Point Addition. The line joining of points P, Q intersects the curve at another point R. This is an interesting feature of ECC 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 [14] for computation of scalar multiplication of point and map-to-point operation.

Computational Assumptions
Elliptic Curve Discrete Logarithm Problem (ECDLP). Elliptic Curve Discrete Logarithm Problem (ECDLP) is a standard assumption in which ECC-based cryptographic algorithm can rely upon. The ECDLP is stated as: given two elliptic curves points P and $Q(= xP)$, finding scalar x is an intractable problem with best known algorithms and available computational resources.

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.

3 The Proposed Protocol

The protocol consists of two phases - (i) Setup phase and (ii) Authentication, Key establishment and Data confidentiality phase.

3.1 Setup Phase

System Setup: The system has three types of entities, namely, a back-end server, readers and tags. We assume that the back-end server is a trusted entity and connected to reader(s) securely. The back-end server chooses a suitable elliptic curve $E(F_q)$ over a finite field F_q where q is a prime number sufficiently large enough to accommodate cryptographic keys. Let $P \in E(F_q)$ be the generator of $E(F_q)$. We assume that the back-end server is configured with m number of tags and n number of readers. The parameters $E(F_q)$, q and P are made public.

SetupTag: For $i = 1, 2, \cdots, m$, the back-end server personalizes i^{th} tag with a private key x_i. The corresponding public key X_i $(=x_iP)$ is stored in tag and readers memory. We note that X_i provides identity information of the i^{th} tag during tag-reader communication.

SetupReader: For $j = 1, 2, \cdots, n$, the back-end server personalizes j^{th} reader with a private key y_j. The corresponding public key Y_j $(=y_jP)$ is stored in reader and tags memory.

3.2 Authentication, Key Establishment and Data Confidentiality Phase

This phase works as follows.

1. Tag chooses a random number $n_t \in_R Z_q$ and computes $N_t = n_tP$. Tag sends N_t to the reader.
2. Reader chooses a random $n_r \in_R Z_q$ and computes
   $N_r = n_rP$
   $d = \texttt{xcord}[yN_t + n_rX]$
   $k = \texttt{xcord}[n_rN_t]$
   $C_r = PRF(d\|X\|Y\|N_t\|N_r\|k)$
   Reader sends N_r, C_r to the tag. Here, $PRF()$ is a pseudo-random function that provides similar properties of a cryptographic hash function.
3. Upon receiving $< N_r, C_r >$, tag computes
   $d = \texttt{xcord}[xN_r + n_tY]$
   $k = \texttt{xcord}[n_tN_r]$
   $C'_r = PRF(d\|X\|Y\|N_t\|N_r\|k)$
   then checks whether $C'_r = C_r$. If it holds, then reader's authentication is confirmed. Now, tag computes $C_t = PRF(k\|Y\|X\|N_r\|N_t\|d)$ and sends C_t to the reader.
4. Reader computes $C'_t = PRF(k\|Y\|X\|N_r\|N_t\|d)$ and checks whether $C'_t = C_t$. If it holds, then tag's authentication is confirmed.

The shared key $SK = PRF(X, k, Y)$ between tag and reader is established after successful authentication of each other. This shared key SK can be used for data confidentiality for that session. Once the session is expired, the ephemeral secrets n_r and n_t will be erased from the respective local system.

We note that in Step 2, reader uses tag's public key X to compute d. The information of tag's X needs to be provided along with the message in Step 1. This can be done in multiple ways. For example, a separate parameter $\alpha = n_tY \oplus X \oplus Y$ can be communicated in the Step 1. The reader retrieves X by computing $\alpha \oplus Y \oplus yN_t$. Now, the reader checks its database for any entry of X. If so, the reader picks that X and proceeds further, else, rejects tag's request. Other possible solution could be of using a $PRF()$, secrets and nonce for generating a checksum for X in Step 1 that should convey the message to the reader for choosing the correct X for the communicating tag.

4 Security Analysis

Assumptions. Although, RFID system can have multiple readers, while analysing the protocol we simplify this by considering that the protocol has one reader and a set of tags $\mathcal{T} = \{T_1, T_2, \cdots, T_m\}$. Initially, $\mathcal{T}$ is empty, but tags are added as and when needed. The reader maintains a database of tuples $< ID_i, x_i, X_i >$, one for every tag $T_i \in \mathcal{T}$ with identity ID_i for $i = 1, 2, \cdots, m$. Every tag T_i maintains an internal state S_i.

Adversarial Model. We consider a *narrow-strong* adversary [13] for analyzing the privacy of our protocol. The adversary is capable of intercepting communication between tag and reader, and can inject data, alter content and delete data. The adversary has ability to use a virtual tag *vtag* in order to refer to a genuine tag that is in readers' vicinity. The adversary can invoke following oracles (the definition of the oracles is taken from [13], [25]).

$T_i \leftarrow$ `CreateTag(ID)`: on input a tag ID, this oracle creates a tag with the given ID and corresponding secrets, and registers the new tag with the reader. A reference T_i to the new tag is returned.

vtag $\leftarrow$ `DrawTag`(T_i, T_j): on input a pair of tag references, this oracle generates a *vtag* and stores the tuple (*vtag*, T_i, T_j) in a table $\mathcal{L}$ in reader database. If the drawing tag reference is already in the table $\mathcal{L}$ then the oracle returns $\perp$; otherwise, it returns *vtag*.

`Free(`$vtag_b$`)`: on input *vtag*, this oracle retrieves the tuple (*vtag*, T_i, T_j) from the table $\mathcal{L}$. If $b = 0$, it resets the tag T_i. Otherwise, it resets the tag T_j. Then, it removes the entry (*vtag*, T_i, T_j) from $\mathcal{L}$. When a tag is reset, its volatile memory is erased. The non-volatile memory that contains the state **S** is preserved.

$C_t \leftarrow$ `Launch`: this oracle launches a new protocol run with the reader. It returns a session identifier *sid*, generated by the reader.

$\lambda' \leftarrow$ `SendTag`$(\lambda, vtag_b)$: on input *vtag*, this oracle retrieves the tuple (*vtag*, T_i, T_j) from the table $\mathcal{L}$ and sends the message λ to either T_i (if $b = 0$) or T_j (if $b = 1$). It returns λ'. If the above tuple is not found in $\mathcal{L}$, it returns $\perp$.

$\lambda' \leftarrow$ `SendReader`(λ, sid): on input λ and *sid*, this oracle sends the message λ to the reader in session *sid* and returns λ' from the reader.

`Result`(sid): on input *sid*, this oracle returns a bit indicating whether or not the reader accepted *sid* as a protocol run that resulted in successful authentication of the tag. If there exists no session with identifier *sid*, then $\perp$ is returned.

`Corrupt`(T_i): on input a tag reference T_i, this oracle returns the complete internal state S_i of T_i.

In our protocol, the adversary can execute all oracles except the `Result` oracle.

4.1 Privacy Experiment

The goal of the adversary in this experiment is to distinguish between two different tags. The experiment consists of a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. The adversary $\mathcal{A}$ controls the communication channel between the reader and every tag. The experiment is defined as follows.

```
$\mathtt{Exp}^{b}_{\mathcal{S},\mathcal{A}}(k')$:
    1. $b \in_R \{0,1\}$
    2. $\mathtt{SetupReader}(k')$
```

3. $g \leftarrow \mathcal{A}^{Queries}(narrow\text{-}strong\ capability)$
4. Check whether $g = b$

The challenger $\mathcal{C}$ presents to $\mathcal{A}$ the system where either T_i (if $b = 0$) or T_j (if $b = 1$) are selected when returning a virtual tag reference through the `DrawTag` oracle. Here, k' is the security parameter.

The adversary $\mathcal{A}$ is allowed to query the oracles any number of times and then outputs a guess bit g. We say that $\mathcal{A}$ breaks the privacy of the protocol if and only if $g = b$, that is, if it correctly identifies which of the tags was active. The advantage of the adversary is defined as

$\mathtt{Adv}_{\mathcal{A}}(k') = \Pr\,[\mathtt{Exp}^{0}_{\mathcal{S},\mathcal{A}}(k') = 1\,] + \Pr\,[\,\mathtt{Exp}^{1}_{\mathcal{S},\mathcal{A}}(k') = 1] - 1$

Theorem 1*. The proposed protocol is narrow-strong private under the DDH assumption.*

Proof. Assume that an adversary $\mathcal{A}$ can break *narrow-strong* privacy of the protocol. That is, $\mathcal{A}$ is able to distinguish a tag from different instances of the protocol runs.

Setup phase:

- Personalize reader with its secret key y and public keys $(X_1, X_2, \cdots, X_m)$ of m tags.
- Personalize each tag with key x_i for $i = 1, 2, \cdots, m$ and with the public key Y of the reader.

Learning phase:
$\mathcal{A}$ calls following oracles in any order:

- `SendReader` oracle for q_r times (note that $\mathcal{A}$ can call this oracle any number of times, but for simplicity we assume that he calls it for q_r times). With these queries, $\mathcal{A}$ can gather (N_{r_i}, C_{r_i}) for $i = 1, 2, \cdots, q_r$.
- `SendTag` oracle, say for q_t times. With these queries, $\mathcal{A}$ can gather (N_{t_i}, C_{t_i}) for $i = 1, 2, \cdots, q_t$.
- `Corrupt` oracle for any $m - 2$ tags (except 2 tags). He then obtains the non-volatile memory state $\mathcal{S}$ of $m - 2$ tags. For simplicity, we assume that tags $T_3, T_4, \cdots, T_m$ are being compromised and tags T_1 and T_2 are not compromised. In other words, $\mathcal{A}$ has knowledge of $m - 2$ tags secrets $x_3, x_4, \cdots, x_m$, but no knowledge of x_1 and x_2.

Challenge phase:
- $\mathcal{A}$ selects two tags T_1 and T_2 on which he did not execute `Corrupt` oracles.
- $\mathcal{C}$ picks $b \in_R \{1, 2\}$ and submits following to the adversary $\mathcal{A}$:

```
Exp^b_{S,A}(k'):
    1. $n_b \in_R Z_q$
    2. $N_b, C_{b,real} \leftarrow \mathtt{SendTag}(\cdot, x_b)$
    3. $C_{b,real} \leftarrow \mathtt{SendTag}_{real}(\cdot, x_b)$
    4. Return $T_{b,real}$
```

```
Exp^b_{S,A}(k'):
    1. $n_b \in_R Z_q$
    2. $N_b, C_{b,random} \leftarrow \mathtt{SendTag}(\cdot, r)$
    3. $C_{b,random} \leftarrow \mathtt{SendTag}_{random}(\cdot, r)$
    4. Return $T_{b,random}$
```

Now, $\mathcal{A}$'s task is to guess whether $C_{b,r} \in \{T_1, T_2\}$, where $r \in \{real, random\}$.

$\mathcal{A}$ can use his database that he gathered in *Learning phase* to check whether $C_{b,real}$ and $C_{b,random}$ match to any records in his database. None of the record in $\mathcal{A}$'s database can find any mapping to $C_{b,real}$ or $T_{b,random}$, as $\mathcal{A}$'s database contains all T_i for tags $i = 3, 4, \cdots, m$ which are created by the corresponding tags' secret key x_i.

If $\mathcal{A}$ can distinguish $C_{b,real}$ and $C_{b,random}$ then he can break the privacy of the protocol. Firstly, it is computationally infeasible for $\mathcal{A}$ to find any clue of n_b from N_b (as $N_b = n_b P$). Secondly, $C_{b,real}$ is an authentication code, computed by a cryptographically secure pseudo-random function with secret parameters of the reader and a transient secret n_b. Without knowing x_b, it is computationally infeasible to guess (with probability not significantly $> \frac{1}{2}$) whether $C_{b,r}$ belongs to T_1 or T_2. As a result, if $\mathcal{A}$ aims to distinguish $C_{b,real}$ and $C_{b,random}$, then he has to first obtain the pre-image of $PRF()$ and then solve the ECDLP, which are infeasible problems. Therefore, $\mathcal{A}$ cannot break the privacy of the protocol. In other words, the advantage $\mathtt{Adv}_{\mathcal{A}}(k')$ of the privacy experiment that $\mathcal{A}$ can have is negligible in k'. □

4.2 Security Experiment

Assume that the adversary $\mathcal{A}$ can convince the reader to accept a fake tag. In order to convince the reader, $\mathcal{A}$ requires to compute a valid authentication code on a target tag (say, T_{target}). Note that T_{target} has not participated in `SendTag` and `Corrupt` oracles. Without these two queries, $\mathcal{A}$ cannot have x_{target}. As a result, he cannot compute d_{target} and k_{target}.

Theorem 2: *The protocol is secure if for any adversary, the advantage that the adversary guesses a session key is negligible in security parameter k'.*

Proof: Suppose that the adversary $\mathcal{A}$ can successfully guess a session key with a non-negligible advantage ξ. Let $\mathcal{D}$ be a distinguisher who can distinguish the distribution spaces Q_R and Q_S, where Q_R is a set of random numbers and Q_S is a set of real session keys. Suppose that $\mathcal{D}$ acts as the adversary while making the target-session query[1] interacting with reader and tag. At the start of the experiment, $\mathcal{D}$ is given $SK \in \{\text{reader, tag, } r\}$. Here, r is a random number or a session key, each with probability $\frac{1}{2}$. $\mathcal{D}$'s goal is to predict if SK is the real session key between reader and tag with non-negligible advantage in security parameter k'. That is, $\mathcal{D}$ outputs 0 if it says $SK \in Q_R$ and 1 if $SK \in Q_S$. The experiment works as follows.

1) assume that $\mathcal{D}$ has gathered l numbers of the real session keys SK_i between reader and tag for $i = 1, 2, \cdots, l$.
2) $\mathcal{D}$ chooses a target session and invokes $\mathcal{A}$ (who can win the target-session query with non-negligible probability ξ) to interact with tag and reader during the target session.

[1] Target-session query: Target-session query is pertaining to any unexpired session and unexposed session. A session is termed exposed if either the local state of the tag or the session key of a session is known or the tag is corrupt.

3) $\mathcal{D}$ acts as a simulator for $\mathcal{A}$ and does following:

- $\mathcal{A}$ activates the session i. Then $\mathcal{D}$ sends the message (N_r, C_r) to the tag, acting as the reader. Upon receiving (N_r, C_r) tag responds with (N_t, C_t). Now $\mathcal{D}$ sends (N_t, C_t) to reader pretending as the tag.
- Whenever $\mathcal{A}$ issues `Corrupt` query on tags (except the tag which participates in the target-session query), $\mathcal{D}$ learns the corresponding private-public keys (x_i, X_i) along with the state S_i.
- If $\mathcal{A}$ issues a target-session query on T_{target} then $\mathcal{D}$ obtains the local secrets from the reader and computes $SK = n_r N_{target}$. Now, $\mathcal{D}$ gives SK to $\mathcal{A}$ as a challenge. Then, $\mathcal{D}$ outputs whatever $\mathcal{A}$ outputs and halts.
- If $\mathcal{A}$ chooses any other session as the target-session, $\mathcal{D}$ outputs random SK and halts.

Now $\mathcal{A}$ tries to guess SK whether $SK \in \{Q_S, Q_R\}$. If $SK \in Q_S$, then SK has the same property as that of the real session key. Otherwise, SK has the property of a random key. As per our assumption, $\mathcal{A}$ can output his guess with probability $\frac{1}{2} + \xi$, where ξ is non-negligible in k'.

When $\mathcal{A}$ picks i as its target-session, $\mathcal{D}$ outputs the same output as that of $\mathcal{A}$. Therefore, $\mathcal{D}$ wins with the probability $\frac{1}{2} + \xi$. Otherwise, $\mathcal{D}$ outputs a random guess with probability $\frac{1}{2}$. But, the case of $\mathcal{A}$ choosing i as its target-session occurs with a probability of $\frac{1}{l}$. Therefore, total probability of $\mathcal{D}$ wins the experiment is $(\frac{1}{l}(\frac{1}{2} + \xi) + (1 - \frac{1}{l})\frac{1}{2}) = (\frac{1}{2} + \frac{\xi}{l})$. As l is a large number, $\frac{\xi}{l}$ is negligible in k'. As a result, $\mathcal{D}$'s advantage to win the experiment is negligible in k'. Therefore, the adversary cannot guess a session key with non-negligible probability in k'. □

Theorem 3. *The proposed protocol provides strong security, as no polynomial-time adversary can distinguish corrupt and uncorrupt tags with non-negligible advantage in security parameter k'.*

Proof. Suppose that the adversary has invoked `Corrupt` oracle on the target tag, say T_c. He then knows x_c and Y. Assume that the adversary can track T_c whenever T_c communicates to reader further. This implies that the adversary can distinguish T_c and any uncorrupt tags.

When the corrupt tag T_c communicates to the reader in a new session, the adversary can intercept parameters (N_t, N_r, C_r), where $N_t = n_t P$, $N_r = n_r P$ and $C_r = PRF(d \| X \| Y \| N_t \| N_r \| k)$. In order to check whether C_r belongs to X, the adversary has to calculate d and k from x_c, Y, and session specific parameters. However, d and k require the ephemeral secret n_t (*resp.* n_r) for the current session, which he cannot guess with non-negligible probability in security parameter k'. Furthermore, guessing n_t from N_t (*resp.* n_r from N_r) is basically solving the ECDLP, which he cannot solve. As a result, the adversary cannot compute d and k to link C_r to T_c that he has got from the reader. In other words, every C_r from reader looks a random string which the adversary cannot distinguish with the captured C_r from previous runs of the protocol using the same tag. □

Forward Secrecy. The proposed protocol provides forward secrecy. If the adversary gets hold of the private keys of tag and reader, he cannot learn any previous session keys, because the session key $SK = PRF(X\|k\|Y)$, where $k = \texttt{xcord}[n_t n_r P]$, involves the ephemeral secrets n_t and n_r chosen by tag and reader, respectively. The ephemeral secrets are erased from the local state of tag and reader once the session is expired. Furthermore, the adversary cannot guess n_t from n_T (*resp.* n_r from N_r), as it relies on ECDLP. As a result, even though the adversary knows the private key x_t of tag and/or x_r of reader, he cannot compute k of previous session without having the corresponding n_t or n_r, and thereby, he cannot compute any previous session keys. Therefore, the protocol provides *forward secrecy.*

4.3 Efficiency

There are many authentication protocols for RFID system based on elliptic curves arithmetic. Bringer *et al* [19] proposed a randomized RFID protocol, which is based on the Schnorr's identification protocol [26]. Subsequently, several protocols [17], [15], [23] have been proposed based on the Schnorr's identification protocol. However, most of them suffer from tracking attacks [20], [22]. The main reason behind the tracking attacks is that their protocol's security strength did reduce to Schnorr's identification protocol, but they failed to support claimed privacy preserving. An adversary can intercept some previous messages and then would be able to link to a tag by manipulating intercepted messages. As far as efficiency is concerned, communication and storage cost of our protocol is as efficient as [19], [17], [15], [23]. Our protocol takes little more computation cost than [19], [17], [15], [23]. However, the protocols [19], [17], [15], [23] failed to support strong security and privacy which are important concerns in RFID system. Whereas, our protocol supports strong privacy and security.

5 Conclusions

Internet of Things (IoT) is projecting as a global network that could connect every object around us. RFID system is one of the core components in IoT infrastructure. In order to make secure communication in several complementary technologies in IoT infrastructure, integration of RFID system in IoT infrastructure requires strong security and privacy notion. We proposed a protocol for RFID security and privacy in the context of IoT scenarios and applications. The proposed protocol provides mutual authentication, key establishment and data confidentiality. While writing this paper, RFID system needs only authentication property, but key establishment and data confidentiality are additional security properties provided by the protocol for their usage in future Internet applications. We have showed that the proposed protocol is secure and provides *narrow-strong* privacy.

Acknowledgement. This work was supported in part by Department of Science and Technology, Ministry of Science & Technology, Government of India through DST/INT/SPAIN/P-6/2009 Indo-Spanish Joint Programme of Cooperation in Science and Technology.

References

1. European Commission: Internet of Things – An action plan for Europe, `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2009:0278:FIN:EN:PDF`
2. Roman, R., Najera, P., Lopez, J.: Securing the Internet of Things. IEEE Computer 44(9), 51–58 (2011)
3. ISO/IEC 14443-2:2001. Identification cards – Contactless integrated circuit(s) cards – Proximity cards – Part 2: Radio frequency power and signal interface
4. Feldhofer, M., Dominikus, S., Wolkerstorfer, J.: Strong Authentication for RFID Systems using the AES Algorithm. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 357–370. Springer, Heidelberg (2004)
5. Hopper, N., Blum, M.: Secure Human Identification Protocols. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 52–66. Springer, Heidelberg (2001)
6. Juels, A., Weis, S.A.: Authenticating Pervasive Devices with Human Protocols. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 293–308. Springer, Heidelberg (2005)
7. Gilbert, H., Robshaw, M., Sibert, H.: An Active Attack Against HB+ - a Provably Secure Lightweight Authentication Protocol. IET Electronic Letters 41(21), 1169–1170 (2005)
8. Bringer, J., Chabanne, H., Dottax, E.: HB++: a Lightweight Authentication Protocol Secure against Some Attacks. In: proceedings of Security, Privacy and Trust in Pervasive and Ubiquitous Computing, pp. 28–33 (2006)
9. Avoine, G., Oechslin, P.: RFID Traceability: A Multilayer Problem. In: S. Patrick, A., Yung, M. (eds.) FC 2005. LNCS, vol. 3570, pp. 125–140. Springer, Heidelberg (2005)
10. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A.: An Efficient Authentication Protocol for RFID Systems Resistant to Active Attacks. In: Denko, M.K., et al. (eds.) EUC-WS 2007. LNCS, vol. 4809, pp. 781–794. Springer, Heidelberg (2007)
11. Peris-Lopez, P., Hernandez-Castro, J.C., Tapiador, J.M.E., Li, T., van der Lubbe, J.C.A.: Weaknesses in Two Recent Lightweight RFID Authentication Protocols. In: Bao, F., Yung, M., Lin, D., Jing, J. (eds.) Inscrypt 2009. LNCS, vol. 6151, pp. 383–392. Springer, Heidelberg (2010)
12. Hernandez-Castro, J.C., Peris-Lopez, P., Phan, R.C.-W., Tapiador, J.M.E.: Cryptanalysis of the David-Prasad RFID Ultralightweight Authentication Protocol. In: Ors Yalcin, S.B. (ed.) RFIDSec 2010. LNCS, vol. 6370, pp. 22–34. Springer, Heidelberg (2010)
13. Vaudenay, S.: On privacy models for RFID. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 68–87. Springer, Heidelberg (2007)
14. Hankerson, D., Menezes, A., Vanstone, S.: Guide to Elliptic Curve Cryptography. Springer (2004)
15. Lee, Y.K., Batina, L., Verbauwhede, I.: Untraceable RFID Authentication Protocols: Revision of EC-RAC. In: Proceedings of the IEEE International Conference on RFID, pp. 178–185 (2009)
16. Hein, D., Wolkerstorfer, J., Felber, N.: ECC Is Ready for RFID – A Proof in Silicon. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC 2008. LNCS, vol. 5381, pp. 401–413. Springer, Heidelberg (2009)
17. Lee, Y.K., Sakiyama, K., Batina, L., Verbauwhede, I.: Elliptic Curve Based Security Processor for RFID. IEEE Transactions on Computer 57(11), 1514–1527 (2008)

18. Oren, Y., Feldhofer, M.: A Low-resource Public-key Identification Scheme for RFID Tags and Sensor Nodes. In: Proceedings of the ACM Conference on Wireless Network Security, pp. 59–68 (2009)
19. Bringer, J., Chabanne, H., Icart, T.: Cryptanalysis of EC-RAC, a RFID Identification Protocol. In: Franklin, M.K., Hui, L.C.K., Wong, D.S. (eds.) CANS 2008. LNCS, vol. 5339, pp. 149–161. Springer, Heidelberg (2008)
20. Deursen, T., Radomirovic, S.: Attacks on RFID Protocols. Cryptology ePrint Archive: listing for 2008(2008/310) (2008)
21. van Deursen, T., Radomirović, S.: EC-RAC: Enriching a Capacious RFID Attack Collection. In: Ors Yalcin, S.B. (ed.) RFIDSec 2010. LNCS, vol. 6370, pp. 75–90. Springer, Heidelberg (2010)
22. Fan, J., Hermans, J., Vercauteren, F.: On the Claimed Privacy of EC-RAC III. In: Ors Yalcin, S.B. (ed.) RFIDSec 2010. LNCS, vol. 6370, pp. 66–74. Springer, Heidelberg (2010)
23. Lee, Y.K., Batina, L., Singelee, D., Verbauwhede, I.: Low-Cost Untraceable Authentication Protocols for RFID (extended version). In: Proceedings of the ACM Conference on Wireless Network Security (WiSec), pp. 55–64 (2010)
24. RFID Security & Privacy Lounge, `http://www.avoine.net/rfid/`
25. Hermans, J., Pashalidis, A., Vercauteren, F., Preneel, B.: A new RFID privacy model. In: Atluri, V., Diaz, C. (eds.) ESORICS 2011. LNCS, vol. 6879, pp. 568–587. Springer, Heidelberg (2011)
26. Schnorr, C.P.: Efficient identification and signatures for smart cards. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 239–252. Springer, Heidelberg (1990)