Volume 2, Issue 11, November 2012 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

RFID Authentication Protocol: A Review

Ritu Dahiya *
CSE, Kurukshetra University
India

Ravinder Dahiya
CSE, Thaper University
India

Abstract—***RFID (Radio Frequency Identification) is recently becoming popular due to its convenience and economical efficiency. Furthermore, RFID nowadays comes into the spotlight as a technology to substitute the barcode. On the other hand, RFID is jeopardized from various attacks and problems as an obstacle of widespread RFID deployment: replay, spoofing, traceability, de-synchronization, un-scalability, and tag cloning. We focus ourselves on the un-traceability and scalability in this paper. In this paper we will propose two authentication protocols for RFID systems. First, we will present an authentication protocol without proxy and its working. After that we will focus on another authentication protocol with proxy, including the properties of the proxy, and how it helps in authentication.***

Keywords— ***Untraceability, Scalability, synchronization, forward Secrecy, Ownership Transfer***

I. INTRODUCTION

A Radio Frequency Identification (RFID) tag is a microchip that is capable of transmitting a unique serial number and other additional data through RF (radio frequency) signals. The goal of a RFID system is to identify objects remotely by embedding tags into the objects. For example, goods in shops can be tagged in order to provide automatic theft-detection, or to manage the goods inventory by using wireless scanning without any handwork[4]. RFID tags are useful tools in manufacturing, supply chain management, inventory control, etc. A RFID system is composed of three components; tag, reader and Back-end database. The characteristics of each component are as follows.

RFID tag carries an object identifying data. When a tag receives a query from a reader, the tag transmits information to the reader using RF signals.

RFID reader reads and re-writes the stored data in a tag. After a reader queries to a tag and receives information from the tag, the reader forwards the Information to a Back-end database.

Back-end database is powerful in computational capacity and manages lots of information related to each tag. Generally we assume that an adversary can monitor all messages transmitted in wireless communication between a reader and a tag. However in wired communication between a reader and a Back-end database, we assume that the reader can establish secure connection with the Back-end database.

II. AUTHENTICATION PROTOCOL WITHOUT PROXY

In this section, an emphasis is given on a scalable and untraceable RFID authentication protocol based on hash function.

A. Main idea

Our main idea is to use a shared secret key k when R sends a query to T; k is written as a new value when enrolling tags in the system or doing ownership transfer while ID_i is updated as ID_{i+1} when successful mutual authentication happens with only authorized readers.

B. Proposed Protocol

TD [11] does not guarantee UNT-DVS; and so, we suggest a protocol to make up for the weakness of TD. In addition, we propose how R communicates with T using timestamp to prevent replay attack without implementing time clock in T.

C. Initialization and Assumption

Any T has four non-volatile memories ID_0, k, access PIN and TS_{last} which are initialized into the memory of T during manufacturing process; ID_0, pseudo-EPC, which is produced by hash function or the other encoding schemes, written into the memory of T; access PIN is written into the reserved memory of T; k is written into the memory of T; TS_{last} is set to 0 during the initialization process. TS_{last} is updated with TS sent by an authorized R to prevent replay attack after successful mutual

authentication. R only has k which is stored during manufacturing process or ownership transfer. S keeps four fields: EPC, $h(ID_i)$, ID_i, and access PIN; ID_i and access PIN are shared between T and S.

D. PROTOCOL DESCRIPTION

Step 1: R gets TS from its timestamp information. R computes h(k, TS), and then transmits h(k, TS), TS to T. T compares TS and TS_{last}. If TS is greater than TS_{last}, then T generates h(k, TS) using TS and k. Otherwise, T considers it as an unauthorized request. If the value received is the same as the value computed, they authenticate R as an authorized one. The step 1 is quite different from the other protocols: the other protocols authenticate R at the last steps(4 - 5) while our protocol authenticates R at the step 1. In other words, T_k responds to R while T_k' does not respond.

Step 2: T sends $h(ID_i)$ to R, which reduces time complexity to $O(\beta)$ in multitag-reader environment because all of the tags respond to a query of R in the previous protocols at all times while only T_k responds in our protocol.

*Step 3:*R forwards $h(ID_i)$ and TS to S. S finds ID_i ; S computes $h(ID_i, PIN)$ using ID_i and PIN; S updates ID_i to ID_{i+1} where $ID_{i+1} = h(ID_i, PIN, TS)$. Otherwise, S stops the procedure.

Step 4: S sends $h(ID_i, PIN)$ to R.

Step5: R forwards $h(ID_i, PIN)$ and TS to T. T compares received and sent TS. If two values equal, T also computes $h(ID_i, PIN)$ and compare the received and value with the computed one. If all comparisons are successful, T updates ID_i to ID_{i+1} like S does; T also updates TS_{last}. Otherwise, T stops the procedure.

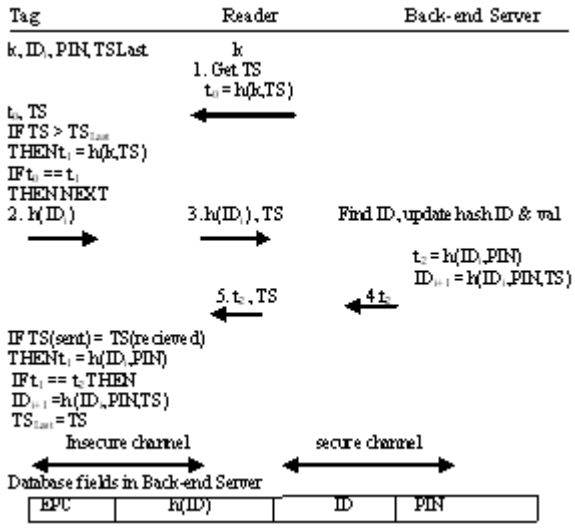

Figure 1: On tag access control authentication protocol without proxy

III. AUTHENTICATION PROTOCOL WITH PROXY

In this section, an emphasis is given on an off-tag access control mechanism using an external device (proxy) which has scalability and untraceability.

A. Overview and Main idea

Off-tag access control provides a chance to be widespread with low-cost tags since the external device takes care of almost high-cost computations instead of T. Low-cost is the most important factor to proliferate RFID technology into the billions of items.The proposed protocol has three properties (RFID tags with a pseudorandom number generator, exploiting universal re-encryption and a proxy) with main research goals (lightweight RFID tags and ownership transfer).

B. Initialization and Assumption

We assume the followings.

1. PKI (Public Key Infrastructure) is established.
2. One proxy manages only one tag.
3. Proxy is within backward channel which is the operating range of T.
4. All channels are insecure.

P has four database fields: Private key, Tag identifier, PIN, Server Location for each tag; Server Location field for each tag can contribute to reducing the work of S. In this protocol, the back-end server has to find a server location if SL is a NULL value where SL denotes a server location for T. P also has an access control list.

An example of an access control list is described in Table1., where, A and B are used to represent the data access authorization level of R; the degree of level depends on the system designer. T has a pseudorandom number generator and memory storages to store PIN and C; C is based on ElGamal [6] encryption algorithm.

C. How the Proxy works

A proxy, P is used for personal usage. P is a reader which can be integrated into cellular phones, PDAs (per-sonal Digital Assistants) or tiny portable device which manages owner's tags; P also enforces privacy policy desired by its owner using a access control list.

TABLE 1
ACCESS CONTROL LIST

Action	**Source**
Pass level A	List of readers which have authorization level A for some tag.
Pass level B	List of readers which have authorization level B for some tag.
. . .	. . .
No answer	The others

In the proposed protocol, P should exist around his own tags; so, the operating range of P works around 1 or 2 meters which is approximately from head to toe of an individual. Juels (REP) [2]'s proxy four different security properties; REP has tag acquisition, tag re-labelling, tag simulation and tag release. P has six functional properties which are depicted in Fig.2 in which an arrow represents a state transition; these properties in this protocol are a little different with REP [2].

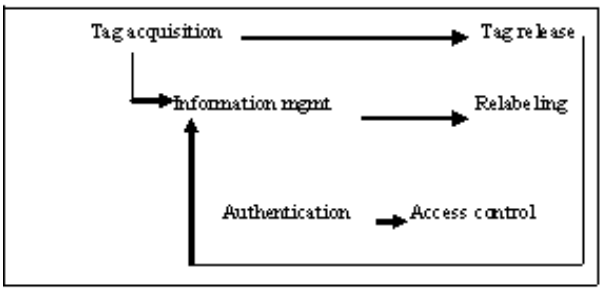

Fig .2: Six functional security properties of P

The description of each component is as follows:

1) *Tag acquisition:* P gets a new SK corresponding to PK and ID of T from S; P also gets PIN from the previous tag owner's P. P generates C, and then writes C and PIN into the acquired memory of T when P acquires T.
2) *Information management:* P manages ID of T, SK, PIN and a server location for each T. P inserts the record in a database when it acquires T; P deletes the record about T when it releases T.
3) *Relabeling:* P relabels T contents whenever the other devices try to write data into T managed by P, which means that P writes C' into T.
4) *Authentication:* P checks whether the queried R is an authorized party or an unauthorized one.
5) *Access control:* If an authorized party has sent a query, then P checks a data access authorization level and passes the proper message for level. Access control can consider three cases with P: which R, which T, which circumstances.
6) *Tag release:* An owner of T releases T when the owner of T does not want to keep his T anymore; that is, ownership transfer happens.

D. Proposed Protocol

In SAITO [6] which is one of on-tag access control scheme, T checks the first and second attacks by itself. however, it is big overhead on T. SAITO checks only the contents written in T not to authenticate R; that is, anybody can get information of T upon receiving C from T while we authenticate R exploiting the external device on behalf of T.

The working of the protocol is shown in Fig. 3, 4 and 5; Fig.3 shows protocol, Fig.4 shows protocol for authorization, Fig. 5 shows protocol for ownership transfer.

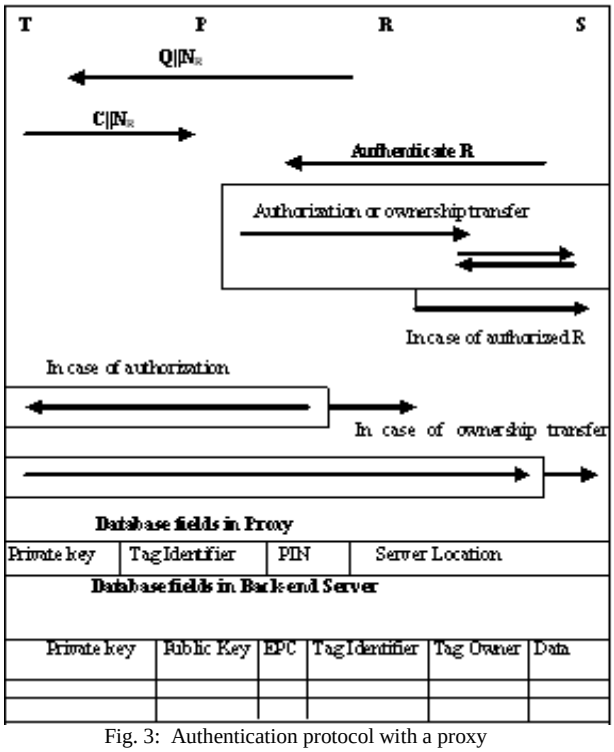

Fig. 3: Authentication protocol with a proxy

Overall protocol works as follows:

Step 1: R sends Q query and random nonce N_R generated by R to T.

Step 2: T sends C and N_R to P. P decrypts C with private key SK_x.

Step 3: The way to communicate between R and P is using a variety of means preferably over the secure channel (See more details in [2]. In purposed protocol, R sends its information like $Sig_R(N_R)||Cert_R$ to P.

Step 4: P checks whether R is authorized or not using an access control list and checks data access authorization level in case of authorized R. As another case, ownership transfer happens in Step 4.

Step 5: Protocol descriptions for authorization and ownership transfer is handled within each protocol. For an unauthorized R, P sends random value to R, which cannot give a chance for the adversary to distinguish the tag from the other tags. In case of authorization protocol, P relabels the contents of T while R relabels the contents of T in case of ownership transfer protocol.

Nonce (N_R and N_P) in our protocol is to ensure that old communications cannot be reused in replay attacks. Nonce can be time-variant or generated with enough random bits which ensure a probabilistically insignificant chance of repeating a previously generated value.

The protocol in case of authorization works as follows:

Step 1: P sends $E(PK_R, M_P||SL)$ to R where M_P denotes $E(PK_S, Sig_P(m||N_P||cmd)||Cert_P)$; SL denotes a server location for T, N_P denotes a random nonce generated by P, cmd represents an authorization level, and m denotes a pseudo-EPC (ID of T) in this protocol.

Step 2: R decrypts $E(PK_R, MP||SL)$ with the private key SK_R of R. R gets a server location, and sends $E(PK_S, Sig_R(M_P)||Cert_R)$ to S which is same with the server location.

Step 3: S decrypts $E(PK_S, Sig_R(M_P)||Cert_R)$, with the private key of S. S finds out the identities of P and R, ID of T, and an authorization level. S checks whether P is the owner of T or not. If P is the owner of T, then S checks the authorization level of R for T. For example, In case that an authorization level is A, S sends $E(PK_R, Data_A)$ to R; In case that an authorization level is B, S sends $E(PK_R, Data_A||Data_B)$. The degree of an authorization level is decided by the system designer. If P is not the owner of T, S sends a random value to R to provide indistinguishability.

Step 4: P computes G(PIN) and generates PIN' where G is a pseudorandom number generator and PIN is used for a seed; G is used for matching the bit size of G(PIN) and (C'||PIN'). P selects a random encryption factor $r' = (k_0', k_1') \varepsilon Z_q^2$, re-encrypts C to $C' = [(\alpha_0', \beta_0'); (\alpha_1', \beta_1')] = [(\alpha_0\alpha_1^{k0'}, \beta_0\beta_1^{k0'}); (\alpha_1^{k1'}, \beta_1^{k1'})]$, and sends $(C'||PIN') \oplus G(PIN)$ to T; lastly, updates PIN with PIN'.

Step 5: T computes G(PIN) with PIN which is in the memory of T, performs XOR operation (between G(PIN) generated by T and $(C'||PIN') \oplus G(PIN)$ received from P), and can get C' and PIN'; lastly, T updates PIN with PIN' and C with C'.The working of the protocol for authorization is shown in Fig 4.

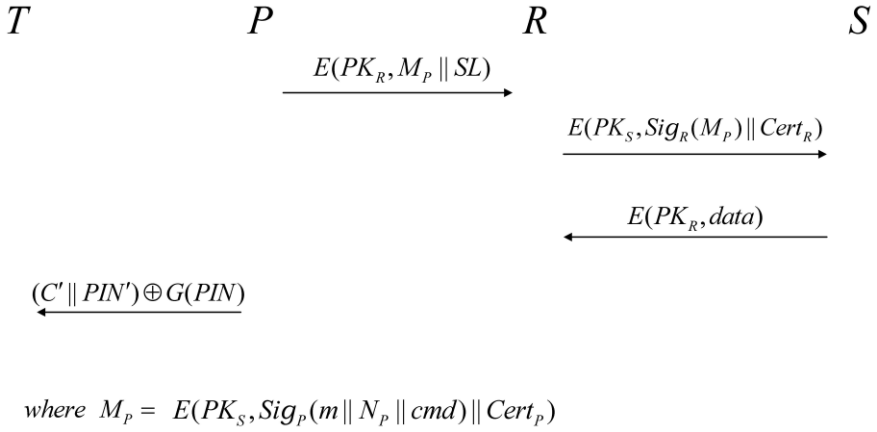

Fig. 4 Authentication protocol with a proxy for authorization

The protocol in case of ownership transfer works as follows:

Step 1: A sends $E(PK_B, M_A||SL||PIN)$ to B whereM_A denotes $E(PK_S, Sig_A(m||cmd) ||Cert_A)$, A denotes the current tag owner, B denotes the new tag owner, and cmd represents ownership transfer command.

Step 2: B decrypts $E(PK_B, M_A||SL||PIN)$ with the private key of B. B gets a server location and PIN, and sends $E(PK_S, Sig_B(M_A)||Cert_B)$ to S.

Step 3: S decrypts $E(PK_S, Sig_B(M_A)||Cert_B)$, with the private key of S. S finds out the identities of and A and B, ID of T, and ownership transfer command. S checks where A is the owner of T or not. If A is the owner of T, then S generates SK and PK corresponding to T. S updates previous key pairs with new key pairs for the tag and the previous tag owner with the new tag owner in the database. And then, S sends $E(PK_B, x||m)$ to B. If A is not the owner of T, S sends a random value to B. Lastly, B generate a new cipher text.

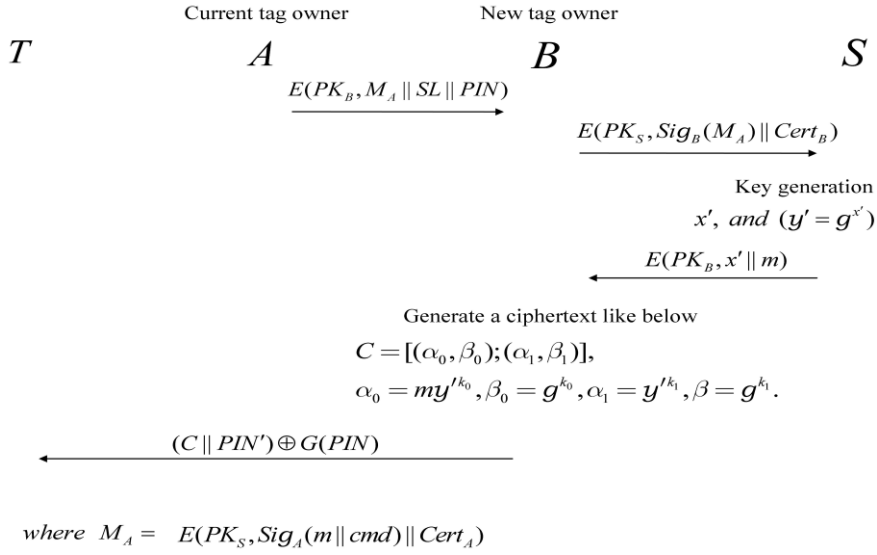

Fig. 5 Authentication protocol with a proxy for ownership transfer

Step 4: B computes G(PIN) and generates PIN', selects a random encryption factor $r = (k_0, k_1) \varepsilon Z^2_q$, generates $C' = (\alpha_0,\beta_0); (\alpha_1, \beta_1)] = [(my'^{k0}, g^{k0}); (y'^{k1}, g^{k1})]$, and sends $(C'||PIN') \oplus G(PIN)$ to T; lastly, B updates PIN with PIN'.

Step 5: T computes G(PIN) with PIN which is in the memory of T, performs an XOR operation (between G(PIN) generated by T and $C'||PIN') \oplus G(PIN)$ received from P), and can get C' and PIN'; lastly, T updates PIN with PIN' and C with C'.

After the ownership transfer protocol, B should perform operation over the secure channel so that PIN' is not eavesdropped by A when writing a new cipher text.

IV. SECURITY AND PERFORMANCE ANALYSIS OF AUTHENTICATION PROTOCOL

In this section, we analyse the security of protocol (without proxy) and compare it with other protocols in Table 2.

1). Synchronization: The simplified TD protocol (without proxy) happens to de-synchronization problem. TD protects de-synchronization between S and tags at the last step in this enhanced TD protocol. In this protocol de-synchronization can occur only if T update its ID and S does not (this will happen if message 5 is lost or corrupted).

2). Forward Secrecy : The protocol updates ID_i to ID_{i+1} using a one-way hash function h() like OSK and TD. As long as there is no alternative, this protocol has to use one-way function to guarantee forward secrecy.

3). Untraceability during a valid session:. Tags authenticate R after receiving the first message, and then tags respond to only an authorized query of R. Therefore, tags do not respond to R with different k. As a result, tags are untraceable during a valid session since ***A*** doesn't impersonate even in the step 1.

TABLE 2

COMPARISON WITH OTHER PROTOCOLS

Protocol	HLS [4]	RHLS [4]	OSK [5]	TD [2]	LACP [3]	Our protocol
Forward secrecy	X	X	O	O	X	O
Untraceability (Valid session)	X	O	O	X	X	*
Untraceability	X	▲	O	▲	▲	Θ
Scalability	O(1)	O(m)	O(2mn)	O(1)	O(1)	O(1)
Scalability in multitag-reader environment	O(γ)	O(nγ)	O(2mnγ)	O(γ)	O(γ)	O(β)
Hash operation	0	1	2	5+α	2	4
Prevent spoofing R	X	X	O	O	O	O
Prevent spoofing T	X	X	X	O	X	O
Synchronization	NA	NA	▲	O	O	O

O Satisfied ▲ partially satisfied
X Not satisfied * if k is revealed, X, otherwise, O
Θ if k is revealed, ▲, otherwise, O.

4).Untraceability: Tags authenticate R after receiving the first message; R authenticates the tags after receiving the second message. In each step, tags and R authenticate counterpart to remove traceability. In addition, although ***A*** knows k, ***A*** cannot trace a particular T since Ts response to query is always different at the valid session.

5).Scalability : This is the biggest contribution of the work. S has time complexity O(β) to find a T in multi-tag-reader environment. This result in the best complexity in comparison with the previous protocols. Time complexity of each protocol changes in multi-tag-reader environment (See Table 2.1); from O(1) to O(γ) in most cases, from O(1) to O(β) in ours where $\beta < \gamma < n$.

6). Spoofing the tag: As long as A doesn't know the value of k, ***A*** cannot spoof T in this protocol. If ***A*** tampers with a T, then ***A*** can spoof the T at the step 1. However, S finds out that ***A*** is not an authorized R in the end. There is no way to spoof the tag unless ***A*** knows k and ID_i.

7).Spoofing the reader: As long as ***A*** doesn't know the ID_i, ***A*** cannot spoof R since tag response to a query by R will be different at all times.

8). Item Privacy : The party who has EPC is only S in this protocol; that is, guarantee item privacy is there as long as S is not compromised. The other previous protocols are also designed to meet item privacy if they hold following three conditions: only S has EPC, T doesn't have EPC, ID is not an EPC itself.

9).Performance Analysis: This protocol is more secure than TD in terms of traceability aspects even though it reduces hash operations from five and more to four. In this protocol, T needs four hash operations to communicate with R with quite good security performance. Under the assumption that tags cannot be tampered, there is no need to send even the last message.

10).Ownership Transfer : It supports ownership transfer using k. As far as we know, ownership transfer issue is dealt with rarely so far. For example, Alice has R that has k which is also stored in tagged items of Alice. When Alice gets some tagged items from Bob, Alice can write her own k.

V. Security and Performance Analysis of Protocol with Proxy

In this section, we check whether the protocol guarantees the following security requirements.

1).Protection against tracing : T sends different message at any time R sends a query. C and C' are indistinguishable, and write command of P is secure because the adversary doesn't have a way to know PIN. Even if the adversary gets PIN after tampering

T, the adversary have to be within 1-2m to trace T at all time while almost all the other previous protocols in the literature can be easily traced after tampering T. In addition, write command by physical contact guarantees updating PIN securely.

2).Protection against cloning and spoofing : Cloning T and spoofing R are meaningless since P maintains a private key and an access control list for each tag. Spoofing T is also meaningless because T doesn't have a way to check whether write command sent by some devices is authorized or not; since the adversary doesn't have any gains, the adversary does not try to spoof T. The adversary's write command make T replace PIN with $PIN_{A,}$ where PIN_A is the generated by the adversary; but, P also checks PIN_A and can writes re-encrypted cipher text generated by P with the PIN_A.

3).Privacy: Privacy is provided as C emitted is provably secure since it is based on UR. As another way to provide privacy, pseudo- EPC as ID of T should be used; S has EPC and Tag identifier field to use pseudo-EPC. It supports data access authorization level-based service, which enhances privacy for individual tag.

4).Protection against DoS: DoS attack can cause battery consumption of P, which is one big problem when using the battery-powered device to protect T.

5).Ownership transfer: The protocol for ownership transfer is Described . Ownership transfer is one of the advanced security requirements; but, Molnar et al. [3] supports sophisticated ownership transfer to the best of our knowledge.

6).Protection against swapping: Swapping attack is one of the vulnerabilities on UR. This protocol, prevents from swapping attack using PIN.

7).Scalability : Since P sends m in encrypted form to authorized R which forwards received message to S, the complexity of tag identification on S is O(1). In other words, S does not need computations related to non-relevant T, which means the protocol is completely scalable.

8).Cost: T requires only one lightweight cryptographic primitive, a Pseudorandom number generator, and re-writable memory to store C and PIN. Consequently, the protocol can be implemented with reasonable low-cost.

VI. Comparison of Protocol with Proxy Vs Related Work

Selective RFID jamming (see details in [7]) makes a signal jam up the airwaves under lots of queries from an unauthorized R while an external device just re-encrypts a new valid C in our protocol. In addition, the use of jamming signal is legally questionable. REP[2] send the secret value of T in unencrypted form, which is insecure since REP and give the adversary a chance to eavesdrop secret values while our P does not reveal the secret information of T.

SAITO [6] has several weaknesses:

1) Big overhead on T.
2) Tracking with only eavesdropping within forward channel,
3) No R authentication mechanism.
4) Allowing swapping attack which is the venerability of UR[9].

Unlike SAITO, we resolve all the problems of SAITO using P and PIN.

VII. Conclusion

There is a trade-off between scalability and untraceablility in RFID authentication protocol; therefore, many literatures did not suggest a protocol which guarantees scalability and untraceability together. However, in this paper, two scalable and untraceable protocols, enabling ownership transfer, have been proposed.

The final conclusion of our protocols is as given in following subsections.

Properties of authentication protocol without Proxy

The protocol without proxies has several security properties which are as follows:

- Supports ownership transfer
- Considers multi-tag-reader environment
- Receives messages from the tags what a reader wants.

Properties of authentication protocol with Proxy

The protocol with a proxy supports several security properties which are as follows:

- Ownership transfer
- Granular data access
- Scalability
- Untraceability
- Privacy
- Protection against several attacks which are spoofing, cloning, and swapping
- The untraceable way even under compromising a tag due to use of pseudo-EPC.
- The more fast way to find a server location.

As extra contributions, we deal with what item privacy is, why item privacy is important and how item privacy can be applied to our protocol. Consequently, we make sure that the proposed protocols can contribute to make RFID deployment widespread.

VIII. FUTURE PROPOSAL

Both of the protocols are secure and computationally efficient. In protocol (*without proxy*), the reader has to know the key of the tag before it can communicate with a tag that is only authorized tags can communicate with the tag. But there are some situations where we should allow fresh readers to communicate with the tag. This is an interesting area to find out ways to allow legal readers to communicate with the tag even if they don't know the tag key. Universal Re-encryption is used in protocol (*with proxy*) as a method to update the tag parameters. However, finding another ways other than universal Re-encryption is another area of future work. Both the protocols have dealt with the ownership transfer but still there is a lot of future work to be done in this area.

REFERENCES

[1] Ari Juels, "Minimalist Cryptography for Low-cost RFID Tags", In C. Blundo and S.Cimato, editors, The Fourth International Conference on Security in Communication Networks – SCN 2004, LNCS 3352, pp.149-164, Sep. 2004, Springer-Verlag, Amalfi, Italia.

[2] Ari Juels, Paul Syverson and Dan Bailey, "*High-power Proxies for Enhancing RFID Privacy and Utility*", Workshop on Privacy Enhancing Technologies – PET 2005, May-Jun. 2005, Dubrovnik, Croatia.

[3] David Molnar, Andrea Soppera and DavidWagner, "*A Scalable, Delegatable* Pseudonym Protocol Enabling Ownership Transfer of RFID Tags", Selected Areas in Cryptography – SAC 2005, LNCS 3897, pp.276-290, Aug. 2005, Springer-Verlag, Kingston, Canada.

[4] Eun Young Choi, Su Mi Lee, and Dong Hoon Lee, "*Efficient RFID Authentication Protocol for Ubiquitous Computing Environment",* International Workshop on Security in Ubiquitous Computing Systems- SECUBIQ, 05, LNCS 3823, pp. 945-954, Dec. 2005, Springer-Verlag, Nagasaki-JAPAN.

[5] Istv´an Vajda and Levente Butty´an, "*lightweight Authentication Protocol for Low-Cost RFID Tags*", International Conference on Computational Science and its Applications - ICCSA 2005, LNCS 3480, pp.619-627, May 2005, Springer-Verlag, Singapore.

[6] Junichiro Saito, Jae-Cheol Ryou and Kouichi Sakurai, "*Enhancing Privacy of* Universal Re-encryption Scheme for RFID Tags", Embedded and Ubiquitous Computing – EUC 2004, LNCS 3207, Aug. 2004, AizuWakamatsu City, Japan.

[7] Melanie R. Rieback, Bruno Crispo and Andrew S. Tanenbaum, "Keep on Blocking' in the Free World: Personal Access Control for Low-Cost RFID Tags", International Workshop on Security Protocols – IWSP'05, Apr. 2006, Cambridge, England.

[8] Miyako Ohkubo, Koutarou Suzuki and Shingo Kinoshita, "*Cryptographic Approach to Privacy-friendly Tags*", In RFID Privacy Workshop, 2003, MIT, USA.

[9] Philippe Golle, Markus Jakobsson, Ari Juels and Paul Syverson."*Universal Re-encryption for Mixnets*", The Cryptographers' Track at the RSA Conference – CT-RSA, LNCS 2964, Feb. 2004, San Francisco, California, USA.

[10] Stephen Weis, Sanjay Sharma, Ronald Rivest and Daniel Engels, "*Security and* Privacy Aspects of Low-Cost Radio Frequency Identification Systems", Conference on Security in Pervasive Computing – SPC 2003, LNCS 2802, pp.454-469, Mar. 2003, Springer-Verlag, Boppard, Germany.

[11] Tassos Dimitriou, "A Secure and Efficient RFID Protocol that could make Big *Brother (partially) Obsolete*", International Conference on Pervasive Computing and Communications – PerCom 2006, Mar. 2006, Pisa, Italy.

