INTERNATIONAL JOURNAL FOR INNOVATIVE RESEARCH IN MULTIDISCIPLINARY FIELD ISSN: 2455-0620 Volume - 6, Issue - 4, Apr – 2020
Monthly, Peer-Reviewed, Refereed, Indexed Journal with IC Value: 86.87 Impact Factor: 6.497
Received Date: 11/04/2020 Acceptance Date: 28/04/2020 Publication Date: 30/04/2020

A Secure and Efficient System for Authentication in RFID

[1] Arwinder Kaur, [2] Gurpreet Singh, [3] Jagmohan Singh
[1] Asst. Prof (CSE) - CGC, Landran, [2]Asst. Prof(CSE)- SSSIET, Derabassi , [3] RS- MM University, Sadopur-Ambala Computer Science &Engineering, CGC-Landran, Mohali, India
Email – [1] arwinder.cse@cgc.edu.in, [2] gurpreet.dullat@gmail.com, [3] saini.jagmohansingh@gmail.com

Abstract: *RFID (Radio Frequency Identification) is newly becoming popular and plays definitely an imperative role in moving to omnipresent society due to deploying its convenience and economic efficiency. Furthermore, RFID nowadays comes into the attention as a technology to alternate the barcode system. RFID is expected to achieve unlimited economic gain. But RFID technology, on the other hand, is jeopardized from various attacks and problems preventing widespread RFID deployment. Previously designed Hash Lock Schemes (HLS) are scalable, but simply noticeable from unauthorized devices and Randomized Hash Lock Schemes (RHLS) are untraceable, but performance degrades when we try to scale it. Therefore, it is necessary to design a scalable and untraceable protocol. Proposed protocol without proxy supports ownership transfer, considers multi-tag-reader environment and reader can send queries after getting replies from the tags to authenticate them. Protocol with proxy for the individual and the universal re-encryption has several rewards like ownership transfer, untraceability against the compromised tags and data access authorization level-based service by the back-end server.*

Key Words: *RFID, smart tag, reader, authentication protocol, RF System.*

1. INTRODUCTION:

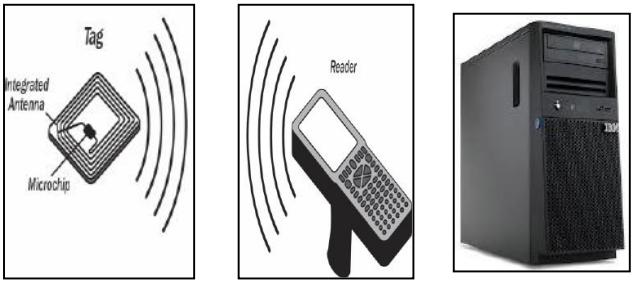
A wireless AIDC technology that uses radio signals to recognize a product, animal or person. Radio-frequency identification (RFID) is the wireless non-contact exercise of radio-frequency electromagnetic fields to relocate data, for the purposes of robotically identifying and tracking tags attached to objects[1-3]. The tags contain electronically stored information.

1.1 AIDC (Automatic Identification and Data Capture) Technology:

Identification processes that rely on AIDC technologies are significantly more dependable and less expensive than those that are not computerized. The most common AIDC technology is bar code skill, which uses optical scanners to read labels [12]. Most people have direct experience with bar codes because they have seen cashiers scan items at supermarkets and retail stores. Bar codes are an vast enhancement over regular content labels because personnel are no longer required to interpret numbers or letters on each label or manually enter data into an IT system; they just have to scan the label. The modernization of bar codes greatly improved the speed and accuracy of the identification process and facilitated better management of inventory and pricing when coupled with information systems.

1.2 RFID System Component:

The RF subsystem consists of three components:



Tag **Reader** **Server**

- Tag: small electronic procedure that are affixed to objects or entrenched in them? Each tag has a sole identifier and may also have other features such as memory to store additional data, environmental sensors, and security mechanisms.
- Reader: devices that wirelessly converse with tags to identify the item connected to each tag and possibly associate the tagged item with related data.
- Server: secure database containing information for a tag that it manages appreciably superior computational capability than a tag.

1.3 Threats Applied to RFID System:

The first step in building a protected system is to realize the intimidation. *Threats* are likely events that cause a system to respond in an unpredicted or detrimental way. It is useful to categorize threats to determine strategies for justifying them. In this section, threats to RFID are categorized using the well-known STRIDE model used in the design of secure software systems. STRIDE is an acronym for six threat categories that are listed below [4-7].

- An attacker modifies a tag.
- Spoofing Identity
- An attacker adds a tag to an object.
- Repudiation
- Information Disclosure
- Denial of Service

1.4 Security Objectives

The security objectives define the "information assurance requirements" of the RFID System. They draw upon industry best practices and adhere to the information security principles of confidentiality, integrity, availability, and non-repudiation. As such, they are desired goals for the deployed system [9]. Detailed explanation of the risks to these goals and proposed countermeasures to these risks are presented.

- Confidentiality
- Integrity
- Availability
- Non-repudiation

2. PREVIOUS WORK:

There have been many papers which are hash-based [14, 2, , 8,], pseudonym-based [12, 13], zero knowledge-based [16] using PUF (Physical Unclonable Function), and tree-based protocol [15] using pseudonym generator that attempt to address the security concerns raised by the use of RFID tag, but it is believed that there is no perfect protocol that avoids all of the threats with low cost as reasonable as applicable until now.

The earlier designed schemes are problems about their security and performance. Like Hash lock scheme are scalable, but traceable and randomized hash lock scheme are untraceable, but un-scalable.

There are no any protocols which have both scalable and untraceable properties.

Therefore, we designed scalable and untraceable protocols.

2.1 SECURITY REQUIREMENTS

When designing an RFID authentication protocol, the following properties should be considered to meet: forward secrecy, untraceability, scalability, synchronization, cloning, prevent spoofing and item privacy. Each clause describes how to devise a protocol which satisfies each property using the five well-known protocols: HLS, RHLS, OSK, TD, and LACP [12-13].

3.1 Forward Secrecy

OSK and TD are known to meet forward secrecy. OSK and TD develop a hash function to revise an identifier while HLS and RHLS did not revive an identifier; that is, upon compromising an identifier, the challenger learns all the prior transactions in HLS and RHLS. LACP uses XOR (exclusive OR) operation to update an identifier; consequently, LACP fails to guarantee forward secrecy. Hash function has a one-wayness property, while XOR operation does not. Pseudonym is used in [18,13], but the adversary can assemble all pseudonyms from the reaction of T in which case protocol based on pseudonym cannot reassurance forward secrecy; more seriously, it cannot guarantee untraceability. In order to design a protocol that guarantees forward secrecy, a protocol designer has to use a hash function when updating secret key as long as there is no alternative. When updating ID_i, finding a lightweight function or scheme that guarantees forward secrecy is also a big open research problem.

3.2 UNTRACEABILITY AND SCALABILITY

In Table 4.1, forward secrecy (FS), untraceability (UNT), and untraceability during a valid session (UNT-DVS) are classified as one category. FS and UNT-DVS are classified into UNT; Guaranteeing UNT means satisfying FS and UNT-DVS. OSK is successful in designing a UNT completely, but it causes the worst result in terms of scalability. The number of tags is going to increase sharply in the nearest future; furthermore, tag recognition rate is not perfect so far.

It increases read operation times; that is, the complexity of OSK $O(mn^2)$ definitely suffers from too much in multi-tag reader environments since all of the tags within the operating range of reader are supposed to respond a query. That's why scalability also cannot be overlooked. We introduce γ as the numbers of tags within a operating range since all tags, which are stored in back-end server, are not likely to be within a range of reader. After applying γ to complexity of OSK, it becomes $O(mn\gamma)$. In this thesis, we define scalability as that the computational complexity is quite suitable for multi-tag-reader environment in the back-end server [16-17].

3.3 Item Privacy

Item privacy can be stated verbally as: active A cannot find out the contents or price of a tagged item even though EPC is revealed.

Violation of item privacy gives A the seduction to steal tagged items after A eavesdrops EPC; in other words, item privacy should be guaranteed although A knows what kind of product after tampering T. For example, A tampers tiny jewelry such that the general public cannot tell genuine from imitation; in this case, A is difficult to decide to counterfeit or not if pseudo- EPC is used in RFID tag[14-16].

3.4 Spoofing and Cloning

HLS and RHLS send message in the clear; so, A can learn the shared secret keys by eavesdropping, and then can spoof R and T. In OSK, spoofing R is possible by replay attack. In LACP, A can spoof the T if R and T send message carelessly.

Cloning is divided into two groups: by eavesdropping or by tampering. Cloning by eavesdropping has the same significance with spoofing the R in terms of security; preventing A from cloning by tampering is hard to prevent since A learns all information of storage.

4. PROPOSED PROTOCOLS:

Two authentication protocols for RFID systems are proposed. First shows an authentication protocol with proxy including the properties of the proxy, and how it helps in authentication and second shows an authentication protocol without proxy.

4.1 Protocol with proxy

In this part, propose an off-tag access control mechanism using an external device which has scalability and untraceability. 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.

4.1.1 Initialization

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.

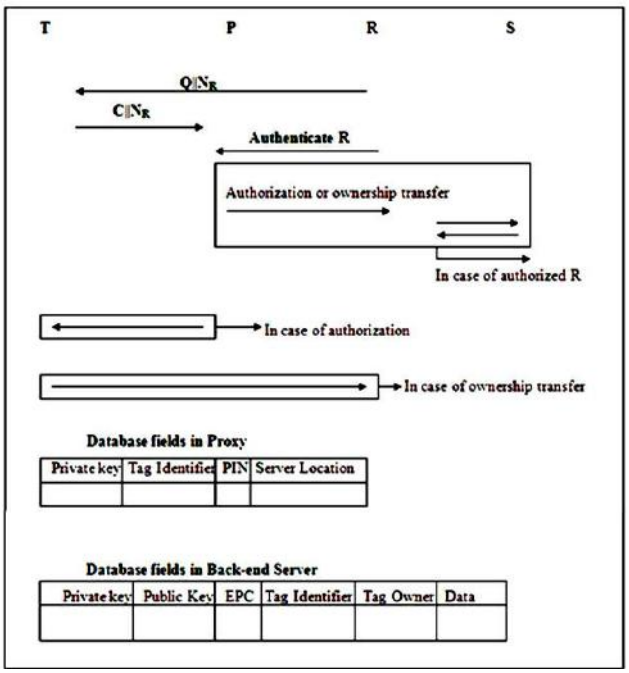


Figure 4.1.1: Authentication protocol with a proxy

4.1.2 Proposed 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 out-of-band or in-band means preferably over the secure channel (See more details in [21]. In this 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; ownership transfer is unusual case, so it requires human interaction to do ownership transfer.

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; the detail description is described in each protocol.
Nonce (N_R and N_P) in 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.

4.1.3 Proposed 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 proposed 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.

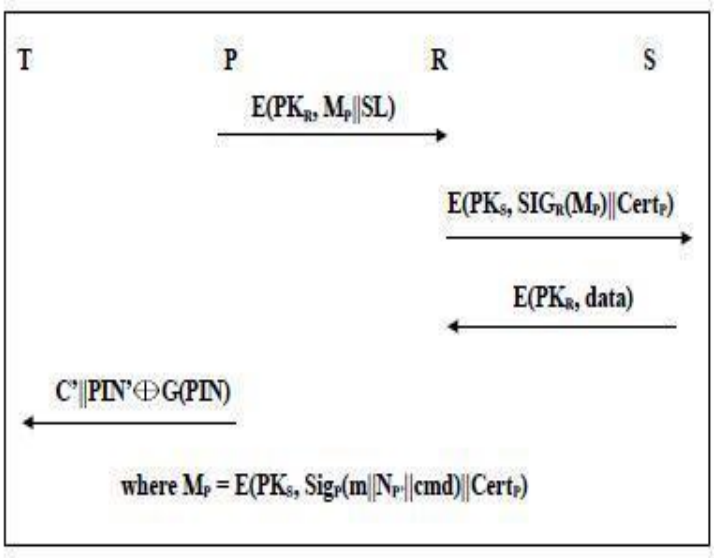


Figure 4.1.2: Authentication protocol with a proxy for authorization

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 indistinguish ability.

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'.

4.1.4 Our 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 P is the owner of T, then S generates SK and PK corresponding to SK. 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 ciphertext.

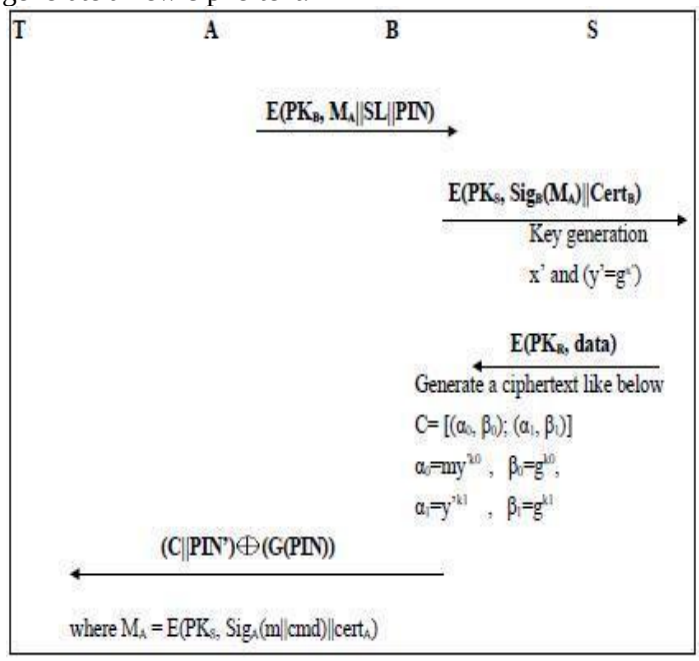


Figure 4.1.3: Authentication protocol with a proxy for ownership transfer

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'.

4.2 Protocol without proxy

Proposed Protocol is going to use a shared secret key k which is assumed to be writable and non-readable 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.

4.2.1 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, pseudoEPC, 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 six fields: EPC, $h(ID_i)$, ID_i, $h(ID_{last})$, ID_{last} and access PIN; ID_i and access PIN are shared between T and S, while rest are not.

S keeps two pair of ID value and its hash (ID, h(ID) and ID_{last}, $h(ID_{last})$). This is to provide synchronization as we will discuss later.

In proposed protocol assume that S can tell an authorized R from an unauthorized one; the clock which is built in R is tightly synchronized like the mobile phone in multi-tag-reader environment.

Proposed protocol works as follows:

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 proposed 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(β) 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 proposed protocol.

Step 3: R forwards h(ID_i) and TS to S. S finds ID_i;it first looks in the the ID list for an ID and if not find there ,then it will check its ID_{last} list.Let find ID is ID_i S computes h(ID_i, PIN) using ID_i and PIN; S updates ID_{last} to ID_i and 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.

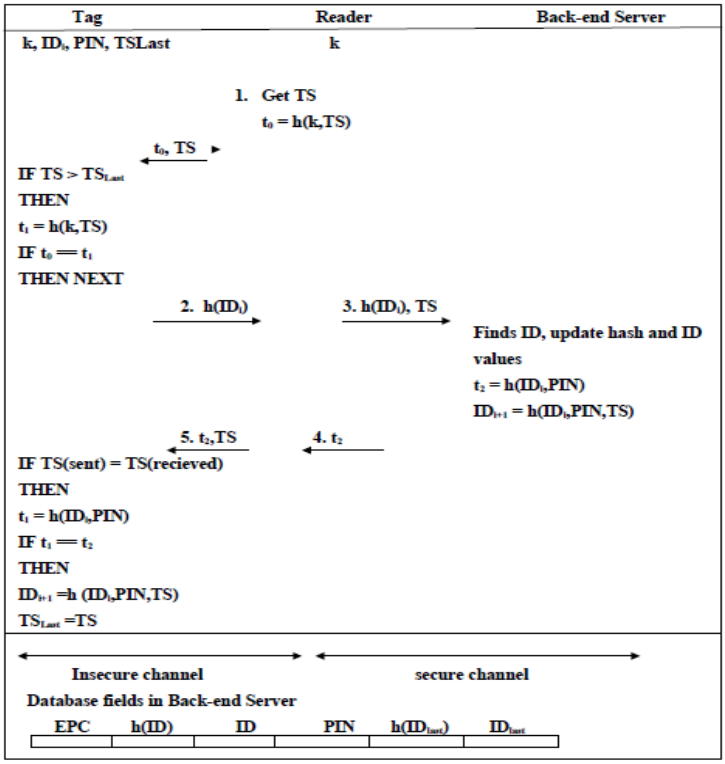


Figure 4.2.1: Authentication protocol without proxy

.

Step 5: 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.

5. SECURITY AND PERFORMANCE ANALYSIS:

Now, it is analyzed that security and performance characteristics of the proposed protocols. First, we analysis the protocol with proxy and then the protocol without proxy.

5.1 Protocol with proxy

5.1.1 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.

5.1.2 Ownership transfer. We described the protocol for ownership transfer. Ownership transfer is one of the advanced security requirements; but, Molnar et al. [27] supports sophisticated ownership transfer to the best of our knowledge.

5.1.3 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 our protocol is completely scalable

Table 5.1: Comparison to the previous schemes

Scheme	**Golle's[13]**	**SAITO[34]**	**Selective RFID Jamming[31]**	**Our Protocol**
External device	No	No	Yes	Yes
Jamming signal	No	No	Yes	No
Prevent two attacks	×	O	NA	O
Prevent swapping attack and garbage value	×	×	NA	O
R authentication mechanism	×	×	O	O
Secure over forward channel	×	×	×	O
Untraceability	×	×	×	O

5.2 Protocol without proxy

5.2.1Forward Secrecy. Our 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, we have to use one-way function to guarantee forward secrecy.

5.2.2Untraceability. 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.

5.2.3 Scalability. this is the biggest contribution of our 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; from O(1) to O(γ) in most cases, from O(1) to O(β) in ours where $\beta < \gamma < n$.

Table 5.2: Comparison with other protocols

Protocol	HLS	RHLS	OSK	TD	LACP	Our protocol
Forward secrecy	X	X	O	O	X	O
Untraceabilityy during Valid session	X	O	O	X	X	*
Untraceability	X	▲	O	▲	▲	Θ
Scalability	O(1)	O(m)	O(mn)	O(1)	O(1)	O(1)
Scalability in multitag-reader environment	O(γ)	O(nγ)	O(mnγ)	O(γ)	O(γ)	O(β)
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.

6. 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, we propose a scalable and untraceable protocol. In addition, R gets response from tags what R wants. As future work, we will propose scalable and untraceable RFID authentication protocol with specific ownership transfer.

REFERENCES:

Journal Papers:

1. National Institute of Standards and Technology "Guidelines for securing Radio Frequency Identification (RFID) systems" NIST special publication 800-98.
2. Gildas Avoine and Philippe Oechslin, "A Scalable and Provably Secure Hash based RFID Protocol", In International Workshop on Pervasive Computing and Communication Security – PerSec 2005, pp.110-114, Mar. 2005, IEEE Computer Society Press, Kauai Island, Hawaii, USA.
3. Gildas Avoine and Philippe Oechslin, "RFID traceability: A multilayer problem" In Andrew Patrick and Moti Yung, editors, Financial Cryptography – FCï05, LNCS 3570, pp.125-140, Feb.-Mar. 2005, Springer-Verlag, Roseau, The Commonwealth Of Dominica.
4. Mihir Bellare, Ran Canetti and Hugo Krawczyk, "Keying hash functions for message authentication" Advances in Cryptology – Crypto 96, LNCS 1109, pp.1-15, Aug. 1996, Springer-Verlag, California, USA.
5. Steve Bono, Matthew Green, Adam Stubblefield, Ari Juels, Avi Rubin, and Michael Szydlo, "Security analysis of a cryptographically-enabled RFID device", 14th USENIX Security Symposium, Jul.-Aug., 2005, Baltimore, Maryland, USA.
6. Claude Castelluccia and Gildas Avoine, "Noisy Tags: A Pretty Good Key Exchange Protocol for RFID Tags", International Conference on Smart Card Research and Advanced Applications - Cardis, LNCS 3928, Apr. 2006, Tarragona, Spain.
7. 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 2005, LNCS 3823, pp.945-954, Dec. 2005, Springer-Verlag, Nagasaki, Japan.
8. Tassos Dimitriou, "A Lightweight RFID Protocol to protect against Traceability and Cloning attacks", Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm'05, pp.59-66, Sep. 2005, Athens, Greece.
9. 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.
10. Gildas Avoine. Security and privacy in rfid systems,
11. Xingxin (Grace) Gao, Zhe (Alex) Xiang, Hao Wang, Jun Shen, Jian Huang and Song Song, "An Approach to Security and Privacy of RFID System for Supply Chain", Conference on E-Commerce Technology for Dynamic E-Business, pp.164-168, Sep. 2005, IEEE Computer Society, Beijing, China.
12. Simson L. Garfinkel, Ari Juels and Ravi Pappu, "RFID Privacy: An Overview of Problems and Proposed Solutions", IEEE SECURITY and Privacy, pp.34-43, May-Jun. 2005.
13. 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.
14. Sindhu Karthikeyan and Mikhail Nesterenko "RFID Security without Extensive Cryptography**"** SASN'05, November 7, 2005, Alexandria, Virginia, USA.
15. Ari Juels, "RFID Security and Privacy: A Research Survey", IEEE Journal on Selected Areas in Communication, Vol. 24, No. 2, pp. 381-394, Feb. 2006.
16. Ari Juels and Stephen Weis, "Defining strong privacy for RFID", Cryptology ePrint Archive, Report 2006/137, 2006.
17. Ari Juels, Ronald Rivest and Michael Szydlo, "The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy", Conference on Computer and Communications Security - ACM CCS, Oct. 2003, Washington, DC, USA.
18. 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.

Web References:

- http://lasecwww.epfl.ch