Received: 22 April 2017 | Revised: 30 May 2017 | Accepted: 10 Jun 2017

DOI: 10.37917/ijeee.13.1.11 | Vol. 13| Issue 1 | June 2017

Open Access

Iraqi Journal for Electrical and Electronic Engineering

Original Article

Design and Implementation of RFID Active Tags and Mutual Authentication Protocol with OwnershipTransfer Stage

Issam A. Hussein
Department of Electrical Engineering
College of Engineering
University of Basrah
Issam_Abdulkareeem@yahoo.com

Ramzy S. Ali
Department of Electrical Engineering
College of Engineering
University of Basrah
rsalwaily@ieee.org

Basil H. Jasim
Department of Electrical Engineering
College of Engineering
University of Basrah
Basil.jasim@uobasrah.edu.iq

Abstract *Radio frequency identification (RFID) technology is being used widely in the last few years. Its applications classifies into auto identification and data capturing issues. The purpose of this paper is to design and implement RFID active tags and reader using microcontroller ATmega328 and 433 MHz RF links. The paper also includes a proposed mutual authentication protocol between RFID reader and active tags with ownership transfer stage. Our protocol is a mutual authentication protocol with tag's identifier updating mechanism. The updating mechanism has the purpose of providing forward security which is important in any authentication protocol to prevent the attackers from tracking the past transactions of the compromised tags. The proposed protocol gives the privacy and security against all famous attacks that RFID system subjected for due to the transfer of data through unsecure wireless channel, such as replay, denial of service, tracking and cloning attacks. It also ensures ownership privacy when the ownership of the tag moves to a new owner.*

***Index Terms*— Active tags, microcontroller, RFID, RFID privacy and security, RF 433 MHz links, Tag ownership transfer.**

1. INTRODUCTION

Radio Frequency Identification (RFID) is a wireless technology that uses radio waves to collect and transfer data from RFID tag or (transponder) attached with an object to the RFID reader or (interrogator) for the purpose of identification and data capturing. RFID technology is expected to be the next generation mutation for tracking, auto identification and data capturing. The first use of this technology was in military field during the World War II to identify friend aircrafts where the system was called Identify Friend or Foe (IFF) [1]. Later, it adopted in many different fields such as security, healthcare, supply chain management, parking, payment systems, inventory, asset tracking, etc. [2, 3]. The reason of using RFID instead of the other auto identification techniques such as barcode and smart cards is, no line of sight is required, works for long range between tag and reader may reach up to more than 100 meters, capability of writing on tags, multiple tags could be read simultaneously, does not affected under harsh weather conditions such as rain and fog and the ability to be integrated with sensors [2]. But in spite of these advantages, RFID technology has the limitation of high cost compared to the barcode technique and it also suffers from privacy and security problems since its data is transferred through unsecure wireless channel, so many protocols have been proposed by researchers to achieve privacy protection and integrity assurance [4].

In this paper, section II shows the basic principle of RFID technology. Section III discusses the security and privacy of RFID system. Section IV shows the advantages and the limitations of the widely known symmetrical authentication protocols used in RFID. Section V introduces the design and implementation of the RFID active tags and reader

using microcontroller ATmega328 and RF 433 MHz modules. Section VI presents our suggested authentication protocol with ownership transfer stage. Section VII includes test and results on the built RFID tags and reader. In the end, section VIII concludes this paper.

2. THE BASIC PRINCPLE OF RFID TECHNOLOGY

RFID technology is used to identify the objects (attached with tags) placed within a reader range. It is one of many technologies gathered under the term of auto identification, such as barcode, smart cards and biometrics. Its main advantage on barcode is, the reader does not need to contact neither see the tag object.

RFID technology basically consists of two parts, reader (transceiver) and tags (transponders) and classified according to the manner in which its tags are powered into three types: active, passive and semi passive or (semi active).

Active tags are trusty, accurate and used for a long range of reader reading. They use internal power source (battery) to transmit their data. But they have the limitations of size, cost and shorter lifetime compared to passive tags because they have power sources. Passive tags are smaller and have no power sources but obtain their power from RF waves sent out by RFID reader when they in reader range. Semi passive tags are similar to the active tags (contain a power source) but they do not transmit any signal unless they are within the reader's range, unlike active tags, which transmit signals in the presence and absence of readers and eventually shortens the battery's life time. The three types of RFID tags are essentially consisting of an antenna to transmit and receive data and a microchip, integrated circuit embedded in silicon chip, to store the data and the unique identifier (ID) of the tag. These tags can work in different frequencies depending on the target application [5].

3. PRIVACY AND SECURITY OF RFID SYSTEM

Recently, RFID technology has been extensively used as a part of daily life. But, most security and privacy issues are yet to be completely resolved. Unsecure RFID tags are subject to revealing their private data, such as tag ID, when they are queried by illegitimate readers. Such tags are vulnerable to spoofing and cloning attacks.

3.1 Privacy

Privacy is one of the main concerns in RFID systems. The transferring of data through unsecure wireless channel could reveal tag information including tag ID.

Tag privacy of RFID tags is suffering from two main threats:

. Information leakage: RFID tags are responding to the legal readers' query and provide their IDs and information. If the tags are unprotected (i.e. do not ensure tag anonymity), that would result in leaking information (such as passport ID, medical record and personal information) to illegitimate readers which could lead to spoofing and cloning attacks.

. Tracking: RFID tags are subjected for tracking attack if the tag responses are related or differ from other tags responses (distinguishable). For example, if the tag is responding with static ID, its location could be tracked by unauthorized entities.

There are several physical approaches used in RFID tags to protect the privacy.

- Blocker tag: a blocker tag is a tag which responds with a fake signal when it gets queried by an RFID reader, so the reader could not use the tag's information [6].

- Faraday cage: it is a method of shielding RFID tag with a cage made of metal mesh or foil to prevent the penetration of RF waves (of certain frequencies) to the tag [6].

- Kill command: this mode of operation has been proposed by Auto-ID center in MIT. The customer can deactivate RFID tag by using "kill command" which is transmitted by RFID reader. The killed tag cannot be reactivated any more. To temporarily

deactivated and reactivate the tag, a "sleep command" is proposed [7].

- Active jamming: the customer can use an active device to broadcast RF signals which block or disturb any nearby RFID reader. But broadcasting with too high power could be illegal by blocking a legitimate RFID system [6].

3.2 *Security Peculiarities*

Many security peculiarities such as confidentiality, integrity, availability (CIA triangle), nonrepudiation, mutual authentication, anonymity, access control and forward security are related to RFID field and usually not achieved unless a specific security mechanism is performed.

- Confidentiality

Confidentiality means allowing only authorized individuals or entities to access or use information assets. There are many approaches to achieving confidentiality, such as physical protection and mathematical algorithms.

- Integrity

Integrity is the feature of assuring the reliability and completeness of information assets. An example of integrity failure is, replacing the price information of a tag attached expensive object with the price information of a tag attached with a cheaper one in an electrical equipment store.

- Availability

Availability means allowing authorized individuals or entities to access or use information assets and prevent the unauthorized from denial the request of information prepared by authorized individuals or entities.

- Non repudiation

Non repudiation is the feature of preventing the sender or receiver from denying a transmitted message.

- Mutual authentication

Mutual authentication is the feature of identifying the tag and the reader for each other.

- Tag anonymity

Tag anonymity means to ensure not to reveal tag sensitive information such as current location and tag identifier.

- Access control

Access control is the feature of providing protection against the unauthorized users.

- Forward security

The meaning of forward security is that, if the adversary has the secrets shared between the legal tag and reader or server, it would not be able to trace the past transactions of the compromised tag. The authentication protocols must ensure forward security.

3.3 *RFID Attacks*

The communication between tags and readers through unsecure wireless channel is vulnerable to eavesdropping by adversaries which leads to passive and active attacks. Preventing the action of the three attacks described below in addition to, achieving privacy requirements and forward security would guarantee the protection of the system and it will be considered as a secure system.

1) Denial of service (DoS) attack

This attack blocks the transmission of information between RFID tags and readers by desynchronizing them. An example of desynchronization is, updating the secrets of RFID server while the tag does not update its secrets during an interrupted or failed authentication session. This will prevent the future authentication of the tag [8].

2) Replay/ Spoofing attack

An adversary can replay the messages of a legitimate tag or reader eavesdropped from the previous authentication sessions between RFID reader and tag. It can fool the reader or the tag by impersonating a legitimate tag or reader. The authentication protocol must ensure that the adversary cannot impersonate a legitimate tag or reader by just replaying the messages eavesdropped from the previous authentication sessions [9].

3) Cloning attack

An adversary can clone the legitimate tag if it has the secrets shared between that tag and an authentic RFID reader or server. The authentication protocol must not reveal the secret information [9].

4. RELATED WORKS ON SYMETRIC KEY RFID AUTHENTICATION PROTOCOLS

Many authentication protocols have been proposed by researchers to solve the problems of security and privacy in RFID system.

Authentication protocols are divided into four classifications according to the primitives they use:

1) Full-fledged protocols which support the traditional cryptography algorithms such as one-way hash function, symmetric key (AES, DES) and public key (RSA, elliptic curve) algorithms, and so on.
2) Simple protocols which comprise from intelligible operations such as using of, one-way hash function, bit wise logical XOR and PRNGs (pseudorandom number generators).
3) Lightweight protocols which consist only a PRNGs and simple function such as CRC (cyclic redundancy check).
4) Ultra-lightweight protocols which include only a bit wise logical operations such as AND, OR, XOR, etc. and permutation or rotation operations. These protocols do not support neither one-way hash function nor PRNGs.

These four categories of RFID authentication protocols can be applied according to the computational power and memory storage of the used RFID tags. Below, some of the famous symmetric-key authentication and ownership transfer protocols.

In 2003, Weis et al., [8] proposed a deterministic hash locks scheme which uses a one-way hash function to lock or unlock RFID tags. Each tag has an identifier and a hash of a secret key. When the tag is queried by RFID reader, it replies with the hash value to the reader. Then the reader searches for the secret key in the database composed of (hash (key), key) pairs. If the key is available, it will be sent to the tag. Next, the tag applies the hash function on that key to verify the legitimacy of the reader. If the result is the same as the stored hash value in the tag, then the reader is legal and the tag sends its data or identifier to that reader. This protocol suffers from tracking and replay attacks due to the static responses of the tag.

To eliminate tracking attack, Weis et al., proposed a randomized hash locks protocol. This scheme includes a nonce r using in the tag side to achieve the randomness of tag responses. The tag instead of sending h(ID), it sends (r, h (ID || r)). In spite of eliminating tracking attack, this protocol suffers from poor scalability by prompts RFID reader to brute-force its database for any ID that matches the hash value h (ID || r) if concatenated with r. It also suffers from the replay attack, if an adversary responds to the reader with the same tag's response she learned from an earlier authentication session. This response will be considered legitimate.

In the same year, Ohkubo et al, [9] proposed a protocol scheme (OSK protocol) which prevent tracking attack and provide forward security. This protocol depends on hash-chains on its design. The database contains tags IDs (ID_i) and the secret key si^0 (as the initial value of tag secret key) of each tag T_i. The tag sends $h_1(si^0)$ when it gets a request by RFID reader and then updating its secret key as $s_i=h_2(s_i)$, where h_2 is another hash function. Next, the server identifies the tag by computing $h_1(h_2^j(si^0))$ on each tag in database until it matches $h_1(s_i)$, h_2^j refers to the

j iterations of the function h_2. This protocol suffers from replay attack when replaying with the last legal tag's response, and it also complains from poor scalability because of the exhaustive search on database. In order to enhance the scalability of the system, Molnar and Wagner, [10] proposed a tree based approach in which each tag has a series of secret keys instead of a single key. Molnar and Wagner approach reduce the complexity of the system (database loading) from linear O(n) search to logarithmic O(log(n)), where n is the number of tags in database.

In 2005, Dimitriou, [11] proposed a mutual authentication protocol between RFID readers and tags. The idea of this protocol is to update tag ID in each of the tag and server after a successful authentication session so the reader and tag in perfect synchronization. Dimitriou protocol provides a protection against cloning attack, but it is susceptible for tracking attack due to the static response of the tag after unsuccessful authentication process. It also suffers from denial of service attack if the tag could not receive reader message to update its ID after a successful authentication session. In the same year, Juels and Weis, [12] proposed a lightweight authentication protocol (HB+) to reduce the number of required cryptography functions used in RFID tags. But their protocol is susceptible for active attacks such as message manipulation.

In 2007, Juels and Weis, [13] proposed an improved randomized hash locks protocol to eliminate replay attack. RFID reader produces a nonce r_1 then query the tag with that nonce. Later, the tag generates another nonce r_2 and sends r_2, h (ID || r_1 || r_2) back to the reader. The reader would perform a linear search on database for any ID that matches the hash value h (ID || r_1 || r_2) if concatenated with r1 and r2. This protocol suffers from poor scalability.

In 2010, Kaleb L, [14] proposed a two-step mutual authentication protocol for small RFID networks based on randomized hash lock scheme proposed by S. Weis. Its protocol has been proved to enhance the security of RFID system dramatically, and it protects passive tags from almost all major attacks that threat RFID systems including cloning, reply, full disclosure, tracking and eavesdropping attacks. To prevent tag ID from being compromised in the randomized hash luck mutual authentication protocol, this proposed protocol responds with h (R || ID) instead of ID in the reader to tag round. Where R is a random number generated by tag side.

In 2013, Yi-Qi Gui and Jie Zhang, [15] presented a new lightweight high efficiency authentication with ownership transfer RFID protocol which achieves forward/backward security in addition to the security requirements. The proposed protocol treats the vulnerabilities that has been found in the other ownership transfer schemes such as Osaka protocol.

Finally, in 2015, Sviatoslav Edelev, Somayeh Taheri and Dieter Hogrefe [16] proposed a minimalist RFID Authentication protocol with ownership transfer stage compliant to the EPC C1G2 based on Quadratic residues. Through the detailed security and privacy analyses, they proved the possibility of their protocol to solve the vulnerabilities in privacy and security in which other protocols can fail to deal with. They use lightweight functions to make their scheme efficient, scalable and practicable for implementation using simple low cost RFID tags.

Since our work based on symmetric key technique for RFID security and privacy, the authentication protocols discussed above are all of symmetric key class (except for the last one). In the other side many works have been done to adapt the using of public key cryptography in RFID system [17, 18, 19, and 20].

5. DESIGN AND IMPLEMENTATION OF RFID ACTIVE TAGS AND READER

Each RFID tag stores a unique identification (ID) that consists of 64 or 96 bits for the purpose of

identification. The design of our RFID active tags and reader including:

A. ATmega328 Microcontroller

The ATmega328 microcontroller displayed in Fig. 1 is used in the proposed active RFID tags and reader to store reader and tag secrets and to perform the cryptography operations needed for the privacy and security of RFID system such as the md5 hash function, bit wise logical XOR and PRNGs. It is high performance Pico power 8-bit AVR microcontroller operates between 1.8 - 5.5 volt and consists of, 32KB ISP flash memory with read-while-write capabilities, 1024B EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, programmable serial USART, a byte-oriented 2-wire serial interface (Philips I2C compatible), master/slave SPI serial interface, a 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), six PWM channels, programmable watchdog timer with internal oscillator, on-chip analog comparator and five software selectable power saving modes. By executing powerful instructions in a single clock cycle, the ATmega328 microcontroller achieves throughputs approaching 1 MIPS per Mhz.

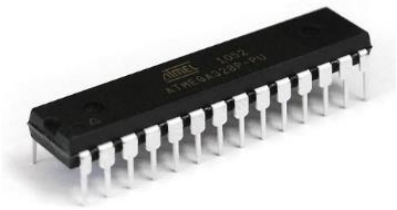


Fig. 1 ATmega328 microcontroller

B. RF 433 MHz Transmitter and Receiver Modules

RF transmitter and receiver modules shown in Fig. 2 are used in tag and reader destinations to transmit and receive data information. They are small and low cost chips. They use ASK/OOK modulation mode. The transmitter works in a range of voltages from 3 (9 mA) to 12 (40 mA) volts and can be used to transmit data up to 100 meters depending on antenna design, working environment and supply voltage. While the RF receiver work on 5 (5.5 mA) volts.

Fig. 2 RF 433 MHz transmitter and receiver modules

C. reader and tag Antenna

An antenna is the device which either converts current into radio wave or conversely capture radio waves from air (at specific frequency band) and convert them into current. It has a number of characteristics such as gain, bandwidth, directivity, impedance and other. RFID antennas have different shapes and configuration depending on the particular application they applied in. Their structures could be as simple as two conducting wires or it might be a single or array of elements amassed in a given situation similar to horn and dish antenna. The most suitable design of antenna will improve the overall system performance. The types of antenna differ in shapes, dimension, and method of feeding, gain, conductivity, and more. In our implemented system we use helix antenna which shown in Fig. 3.

Fig. 3 Reader and tag antenna

Anyway, antenna design is out of our study in this paper.

6. OUR PROPOSED PROTOCOL

In this work, we propose a fully scalable privacy-preserving hash-based protocol which allows tag ID update mechanism and provides mutual authentication to meet most of the security and privacy requirements in RFID systems. Additionally, our protocol allows ownership transfer, which ensures complete privacy and security of the information within the tag once tag's ownership is transferred to another person or entity.

6.1 *Symbols*

The following table gives the declaration for the symbols used in the proposed protocol.

Table. 1 Symbols and their declaration

Symbol	description
ID_0	Initial tag identifier uses as record index in the database (DB)
ID	Tag identifier, stored in tag memory
IDnew	Tag ID of the new authentication session, stored in the database
IDold	Tag ID of the old authentication session, stored in the database
IDc	Tag ID of the current authentication session
h	MD5 hash function
h_{32}	The first 32 bits (4 bytes) of the MD5 hash string, represent in the tag by eight characters
h_{64}	The first 64 bits (8 bytes) of the MD5 hash string, represented in the tag by sixteen characters
hIDnew	The first 32 bits (4 bytes) output of the MD5 hash string to the IDnew input, stored in the database as eight characters
hIDold	The first 32 bits (4 bytes) output of the MD5 hash string to the IDold input, stored in the database as eight characters
r	Random number generated by RFID reader
t	Random number generated by RFID tag
sqn	Sequential numbers string shared secretly between legal tags and server (tag's owner)-system key
q	Hidden numbers string shared secretly between legal tags and readers
||	Concatenation operator

6.2 *Configuration and Authentication Stage*

Each of the tag, reader and server has initial setup (configuration) as shown in Fig. 4. The tag initially stores ID = ID_0, system key sqn, which is used in the proposed protocol to defeat most of RFID attacks such as reply attack, cloning and other since it is shared secretly on all the tags and server. That prevents the attacker from producing a legitimate tag's message. sqn is also used to ensure location privacy when tag's ID is not updated after successful or unsuccessful authentication session by making the term h (sqn || t) unpredictable to the attacker. The tag also includes a secret key q. Moreover, it is programmed with hash function h (using the MD5 algorithm), bit wise logical XOR and pseudo random number generator (PRNG). The reader also

programed with the same hidden key q, hash, PRNG generator and bit wise logical XOR functions. Furthermore, the back-end database (the server) is designed with the same MD5 and XOR functions. The design of the database initially includes system key sqn, the unique index of each RFID tag ID_0, the IDs of the old and new authentication sessions, i.e. IDold = ID_0, IDnew = ID_0, and the MD5 hash of these values, i.e. h_{32} (IDold = ID_0) and h_{32} (IDnew = ID_0) respectively.

The authentication process is started when RFID reader requests RFID tag by random number r. The following steps explain the authentication stage:

Step 1. The reader sends a query with a random number r to the tag.

Step 2. Once receiving r, the tag generates another random number t and computes both of, the hash value of the concatenated string (ID, t, r), i.e. $a_1 = h_{32}$ (ID || t || r) and hID = h_{32} (ID) ⊕ h_{32} (sqn || t). Next, the tag replies with a_1, hID and t back to the reader.

Step 3. When receiving tag messages, the reader sends hID and t to the server in order to perform a quick search on database to get ID_0 from the database record where hIDnew or old = hID ⊕ h_{32} (sqn || t). Next, the server forwards IDc (IDc = IDnew if hIDnew = hID ⊕ h_{32} (sqn || t) or IDc = IDold if hIDold = hID ⊕ h_{32} (sqn || t)) from the database to the reader by using of the selected index ID_0. Otherwise, the session is terminated

Step 4. To verify tag legitimacy, the reader computes the MD5 hash value of the concatenated string (IDc, t, r), i.e. $b_1 = h_{32}$ (IDc || t || r). If a_1 equals b_1, the tag is authorized and the reader forwards IDc.000 to the server in order to update its states so that IDold = IDnew, hIDold = hIDnew, IDnew = h_{32} (IDnew || sqn) and hIDnew = h_{32} (IDnew) for the selected record with IDnew = IDc if IDc = IDnew. Else if IDc = IDold, the database remains in the same state (without updating). Next, the reader replays with the MD5 hash value of the concatenated string (IDc, t), i.e. $a_2 = h_{64}$ (IDc || t) to the tag for the purpose of updating tag identifier ID. Else if a_1 doesn't equal b_1, the session is terminated.

Step 5. When the involved tag receiving a_2, it computes the MD5 hash value of the concentrated string (ID, t), i.e. $b_2 = h_{64}$ (ID || t) in order to verify reader legitimacy, then update its ID so that IDnew = h_{32} (ID ||sqn) If a_2 is equal to b_2. Else, the session is ended without updating. Fig. 5 illustrate the processing of the authentication stage.

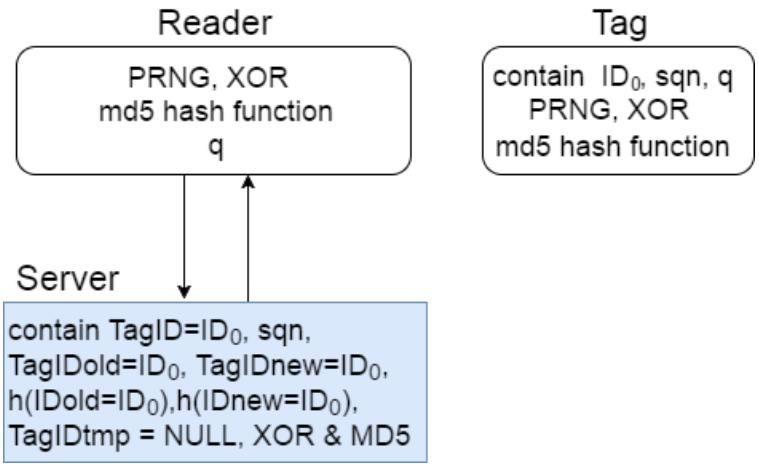


Fig. 4 The configuration of the tag, reader and server

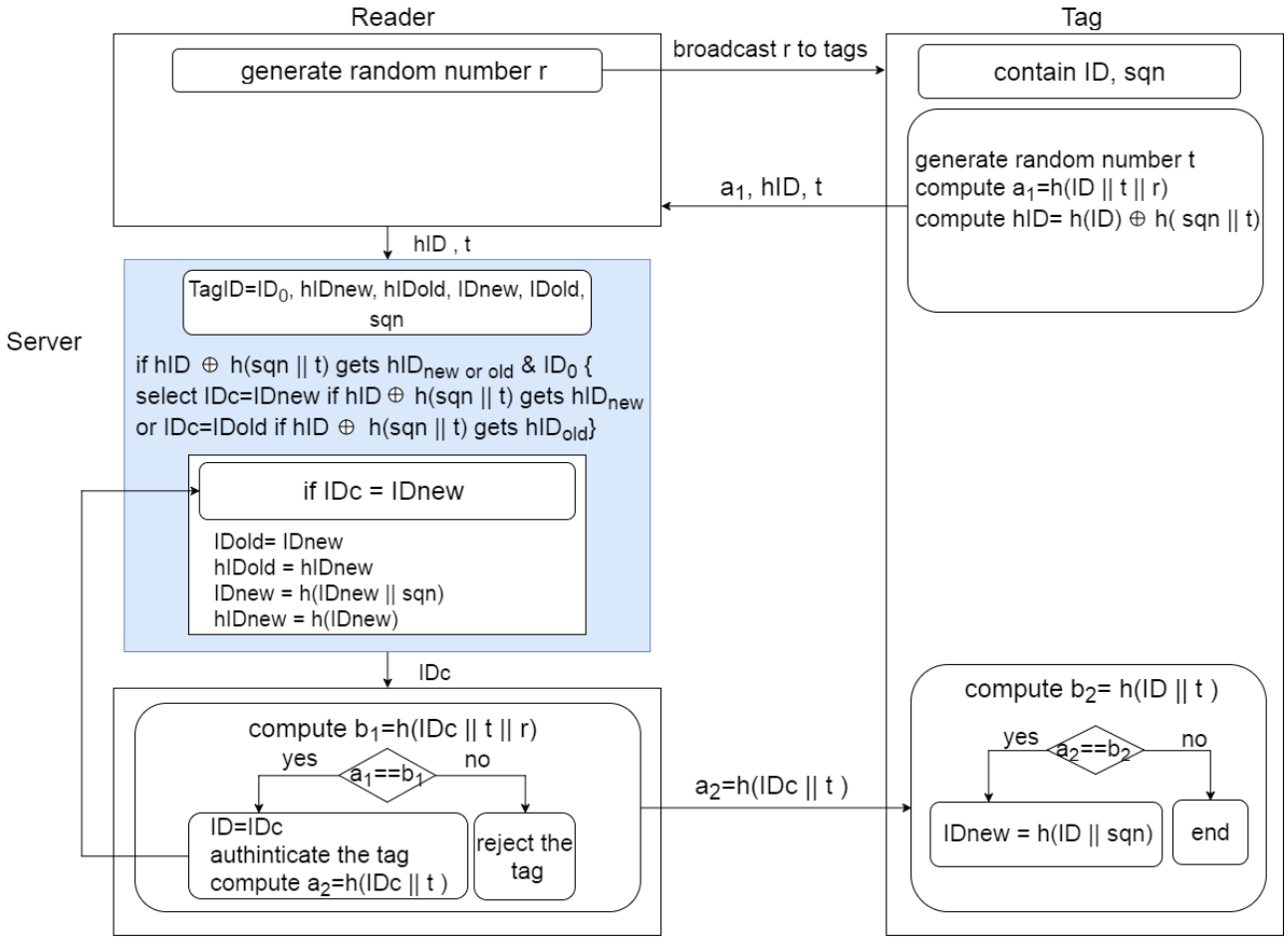


Fig. 5 The authentication stage

6.3 Ownership Transfer Stage

The transfer of ownership of RFID tag to the new owner involves two phases. In phase one the old owner update tag secrets ID and sqn to temporary values IDtmp and sqntmp respectively to protect the privacy of the old owner. While in phase two the new owner receives tag secrets IDtmp and sqntmp through secure channel then update these two secrets to IDnew and sqnnew respectively in order to prevent the accessing, controlling and tracking of the tag by the old owner. Note that ID_0 is not used in the tag side, so the new owner can choose any value for the unique index ID_0.

- Ownership transfer phase I (executed by the old owner)

This phase is running by the old owner and conducts the following steps:

Step 1. Request the concerned tag in transferring ownership by r.

Step 2. Once receiving r, the tag replies with $a_3=h_{32}$ (ID || t || r), hID = h_{32} (ID) ⊕ h_{32} (sqn || t) and t to the reader.

Step 3. When receiving tag messages, the reader sends hID with t to the server in order to perform a quick search on the database to get ID_0 from the database record where hIDnew or old = hID ⊕ h_{32} (sqn || t). Next, the server forwards IDc with the chosen sqntmp to the reader by using of the selected record index ID_0. Otherwise, the session is terminated

Step 4. The reader verifies tag legitimacy by comparing the received tag message a_3 with $b_3 = h_{32}$ (IDc|| t || r). If a_3 equals b_3, the reader forwards IDc.sqntmp to the server in order to update IDtmp state so that IDtmp = h_{32} (IDc || sqntmp). Next, it

computes m= sqntmp ⊕ h_{32} (IDc || r || q) and a_4 = h_{32} (sqntmp || t) then send both of m and a_4 to the tag. Else the session is terminated.

Step 5. When receiving reader messages m and a_4, the involved tag computes sqntmp = m ⊕ h_{32} (ID || r || q) and b_4 =h_{32} (sqntmp || t), then update sqn to sqntmp and ID to IDtmp=h_{32} (ID || sqntmp) if a_4 is equal to b_4. Else the session is ended without updating.

At this point in which tag secrets sqn=sqntmp and IDtmp = h_{32} (ID || sqntmp), phase one is ended and the new owner gets these secrets through secure channel to conduct phase two. Fig. 6 clarifies the processing of the ownership transfer phase one.

- Ownership transfer phase II (executed by the new owner)

This phase is running by the new owner and conducts the following steps:

Step 1. Request the concerned tag by r.

Step 2. Once receiving r, the tag replies with a_3 = h_{32} (IDtmp || t || r), hID = h_{32} (IDtmp) ⊕ h_{32} (sqntmp || t) and t to the reader.

Step 3. When receiving tag messages, the reader sends hID with t to the server. Next, the server computes hIDtmp = h (IDtmp). If the server state hIDtmp = hID ⊕ h_{32} (sqntmp || t), it forwards IDtmp with the chosen sqnnew to the reader.

Otherwise, the session is terminated

Step 4. The reader verifies tag legitimacy by comparing the received tag message a_3 with b_3 = h_{32} (IDtmp || t || r). If a_3 equals b_3, the reader forwards IDtmp.sqnnew to the server in order to update its states so that IDnew = h_{32} (IDtmp || sqnnew) and hIDnew = h_{32} (IDnew). Next, it computes m = sqnnew ⊕ h_{32} (IDtmp || r || q) and a_4 = h_{32} (sqnnew || t) then send both of m and a_4 to the tag. Else the session is terminated

Step 5. When receiving reader messages m and a_4, the involved tag computes sqnnew = m ⊕ h_{32} (IDtmp || r || q) and b_4 = h_{32} (sqnnew || t), then update sqntmp to sqnnew and IDtmp to IDnew = h_{32} (IDtmp || sqnnew) if a_4 is equal to b_4. Else the session is ended without updating.

At this point in which tag secrets sqn = sqnnew and IDnew = h_{32} (IDtmp || sqnnew) phase two is ended and the ownership of the tag has been moved to the new owner. Fig. 7 represents the processing of the ownership transfer phase two

.

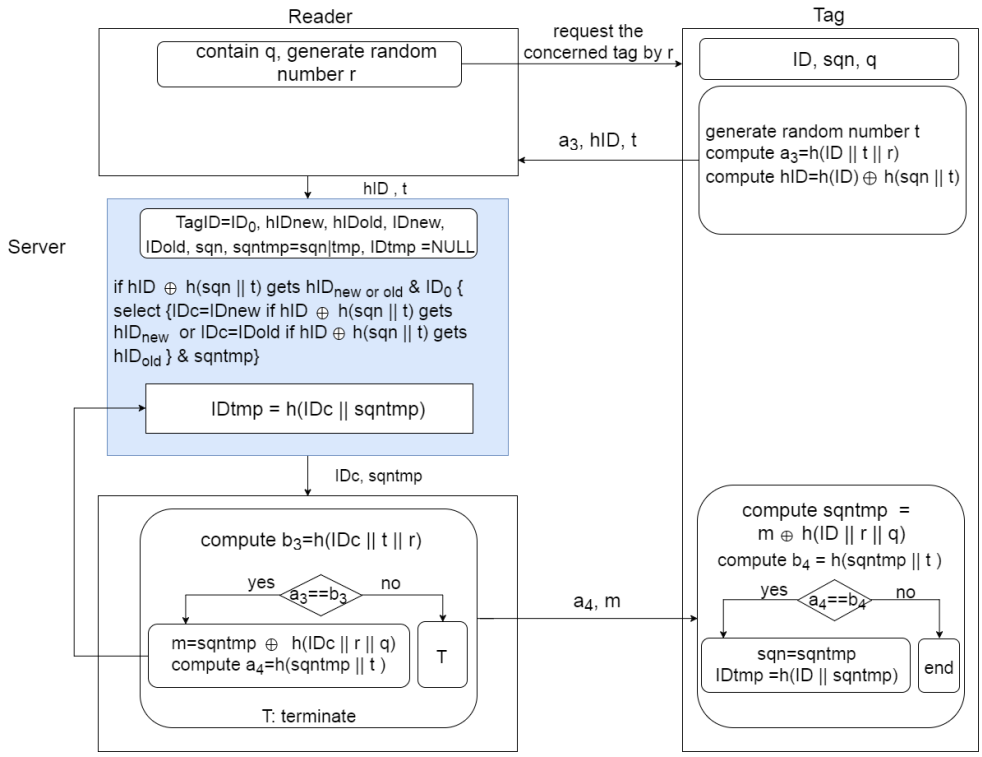


Fig. 6 Ownership transfer stage phase I

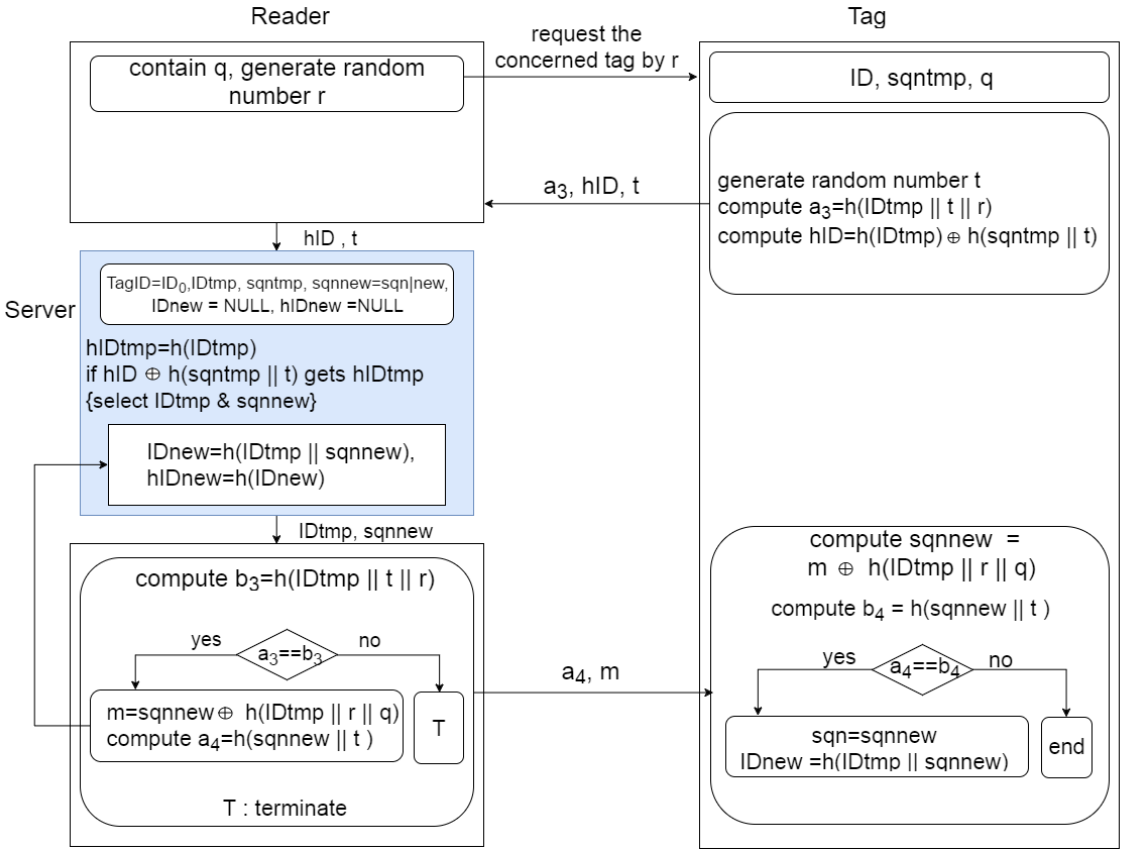


Fig. 7 Ownership transfer stage phase II

6.4 Security Analysis

In this section we discuss the security of our protocol for the points listed below.

- Tag anonymity

To achieve tag anonymity, the proposed protocol must ensure not to reveal tag secrets ID, sqn and q. Since our protocol uses a secure hash function MD5, which has the property of preimage resistance, and the bitwise logical XOR, it is impossible for the attacker to get tag secrets.

- Tracking

The proposed protocol accomplishes the privacy of tag location in a full way due to the using of tag random number t in each of tag messages a1 and hID, and reader message a_2. If a tag has conducted a completed successful authentication session so its ID will be updated for the next session, its hID message as well as a1 will be different from the previous ones even the tag receives the same reader request r of the previous session by an attacker.

$h(ID) \oplus h(sqn \| t) \neq h(ID') \oplus h(sqn \| t')$ & $h(ID \| t \| r) \neq h(ID' \| t' \| r)$

where ID' is the updated state of tag ID and t' is a different random number of t.

While, if a tag has conducted an unsuccessful session or successful session without updating tag ID in the tag side (uncompleted successful authentication session), its ID will be static, its hID message as well as a1 will also be different from the previous ones even the tag receives the same reader request of the previous session r by an attacker.

$h(ID) \oplus h(sqn \| t) \neq h(ID) \oplus h(sqn \| t')$ & $h(ID \| t \| r) \neq h(ID \| t' \| r)$

Also reader updating message a2 differs between one session and another regardless if the tag ID is updated or not since its contains tag random number t, i.e. $h(ID \| t) \neq h(ID' \| t') \neq h(ID \| t')$.

As a result, our protocol achieves location privacy in a full way.

- Forward /Backward security

If the tag ID has been leaked out, the past and future transactions of the compromised tag cannot be tracked since they use different IDs due to the processing of tag ID updating mechanism. So our protocol achieves both of forward and backward security. To improve backward security, we can use the random number t which exists in both of the tag and server to update the ID instead of using sqn. So if the ID of the tag and even sqn are leaked out, the future transactions cannot be traced once the attacker loses the eavesdropping on the transmitted messages between the compromised tag and the reader for just one session (loses eavesdropping on random number t).

$h(ID^* \| t \| r) \neq h(ID' \| t \| r)$ &

$h(ID^* \| t) \neq h(ID' \| t)$

where ID' is the past or future state of tag ID and ID^* is the leaked out ID.

- DoS attack

An active attacker can prevent tag ID from being updated by intercept the last message prepared by the reader a_2 in the authentication stage while reader messages a_4 and m in the ownership transfer stage. This leads to desynchronization between the reader and the tag due to tag ID updating in the server side while the tag remains without updating. But because of the server stores IDold and its hash value h_{32} (IDold) for the mutual authentication session and IDtmp and h_{32} (IDtmp) for the ownership transfer session, Daniel of service attack cannot take place in our protocol.

- Replay attack

Since a_1 and a_3 depend in their hash value on a fresh random number generated by RFID reader r while a_2 and a_4 depend on a fresh random number t generated by tag side and m built upon tag ID that updated in the tag side after each successful ownership transfer session phase I or phase II, the replay of these messages by adversaries is useless.

$h(ID \| t \| r) \neq h(ID \| t \| r')$ & $h(ID \| t) \neq h(ID' \| t')$ & $sqn_{tmp\ or\ new} \oplus h(ID_{c\ or\ tmp} \| r \| q) \neq sqn_{tmp\ or\ new} \oplus h(ID_{tmp\ or\ new} \| r \| q)$ & $h(sqn_{tmp\ or\ new} \| t) \neq h(sqn_{tmp\ or\ new} \| t')$

where ID' is the updated state of tag ID while t' and r' are different random numbers of t and r respectively.

- Cloning/Impersonation attack

While our protocol ensures not to disclose tag secrets, the attacker tries to clone the tag by eavesdropping the replay a1, hID and t from a legal tag then try to fool the reader by putting them in a counterfeit tag. Since tag ID is updated after each completed successful authentication session and a_1 is depending on a random number r generated by the reader, the cloning attack completely fails without knowing tag secrets ID and sqn.

- Ownership privacy

Our proposed protocol achieves the two requirements of the ownership privacy:

1) Forward Untraceability

When the changing of tag's ownership to a new owner is occurred, the old owner must have no ability to access or trace that tag any more. Since the

proposed protocol changes the sequential numbers string of the tag "sqn" to a new sequential numbers "sqnnew" by using of q which is hidden from the old owner, the old owner would not be able to get sqnnew which makes the old owner have no chance to trace or access the tag any more.

In order to obtain sqnnew, the old owner must use m which got it from eavesdropping on phase II of the ownership transfer stage as follow:

sqnnew = m ⊕ h (IDnew || r || q) = sqnnew ⊕ h (IDnew|| r || q) ⊕ h (IDnew || r || q)

The term h (IDnew || r || q) cannot be computed by the old owner since q is unknown.

2) Backward Untraceability

To protect the previous transactions of the tag that happened before its ownership move to another person or entity of being tracked by the new owner, the new owner gets IDtmp and sqntmp which have not been used in any past transactions of that tag. So our protocol achieves forward and backward untraceability.

A brief comparison between our proposed protocol and a famous authentication protocols with ownership transfer feature is shown in Table. 2. From the comparison, our proposed protocol achieves the privacy and security requirements in addition to the providing strong security against the several of RFID attacks.

6.5 Performance Analysis

Now we give a brief performance analysis of our proposed protocol in term of computation cost, communication overhead and storage. In the implemented tags and reader, we use the length of 64 bits (8 characters) for each of tag ID, t, r, a_1, a_3, m, a_4 and hID. While 128 bits (16 characters) are given for a_2 and 24 bits (3 characters) are used for each of sqn and q. In the implemented RFID system (tags & reader), we use microcontroller Atmega328 which gives us a reasonable memory and the ability to perform complex and sophisticated computations.

- Computation cost

The computational cost in RFID tags is a real barrier for RFID systems. Since we use microcontroller in the implemented RFID tags and reader, we override this limitation. Our protocol uses hash function MD5, XOR and PRNG as the main operations. The proposed protocol suitable for implementation in the low cost RFID tags since it uses lightweight operations such as XOR and PRNG but the MD5 hash function must replace with the lightweight cryptographic hash functions proposed recently such as Keccak and Quark [21,22] since the MD5 function required about 8000-10000 gates for implementation which is not suitable for the low cost RFID tags.

- Communication overhead

Our proposed protocol is relatively light in term of communication overhead. It uses a total of 192 bits for tag messages a1, hID and t (64 bits for each) and also 192 bits for reader messages r and a_2 (64 bits for r and 128 bits for a2) in the authentication stage. In the ownership transfer stage phase one and phase two, the number of bits is also 192 for both of tag and reader messages. So the total number of messages bits that transfer in authentication stage is 384 and 768 bits for ownership transfer stage.

- Storage

According to the section V Point A, the ATmega328 Microcontroller has three memory types, 32 KB flash memory, 1 KB EEPROM memory and 2 KB SRAM memory. Flash memory is a nonvolatile memory type and used to save the program (the sketch). SRAM is volatile and used to store the temporary state of the variables. EEPROM is nonvolatile and used to recover the state of the variables after the microcontroller is powered off. In our proposed protocol, we just use 88 bits from 1 KB rewritable EEPROM memory to store the last updated ID of the tag and system key sqn in tag side. While the reader does not require for any type of rewritable memory. Table. 3 illustrates the performance analysis of the proposed protocol.

Table. 2 Comparison of protocols

	Osaka et al. [23]	Lei and Cao [24]	Kulseng et *al.* [25]	Song and Mitchell [26]	Miyaji et al. [27]	Chen et al. [28]	Proposed protocol
tags anonymity	√	√	√	√	√	√	√
Forward security	×	√	√	√	√	√	√
Replay attack	√	√	√	√	√	√	√
DoS attack	×	×	×	×	×	×	√
Tracking	×	√	√	√	√	√	√
Ownership privacy	×	×	√	×	×	×	√

√ : secure against the attack or achieving the property, × : unsecure against the attack or doesn't achieve the property.

Table. 3 performance analysis

computation cost			communication		storage	
Authentication stage						
T	R	S	R to T	T to R	T EEPROM	S
5H 1RN 1XOR 1CMP	2H 1RN 1CMP	3H+O (1) 1XOR, 3CMP if IDc = IDnew or 1H+O (1) 1XOR, 3CMP if IDc = IDold	$l_r+l_{a_2}$ (64+128 = 192 bits)	l_{a_1}+lhID+lt (64+64+64 =192 bits)	ID+sqn (64+24 =88 bits)	n (lID_0+lIDold+ lIDnew+lh(IDold)+ lh(IDnew)) +sqn n (64+64+64+64+64 =320 bits) +24 bits
Ownership transfer stage phase one						
6H 1RN 2XOR 1CMP	3H 1RN 1CMP 1XOR	2H+O (1) 1XOR, 2CMP	$l_r+l_{a_4}$+lm (64+64+64 =192bits)	l_{a_3}+lhID+lt (64+64+64 =192 bits)	ID+sqn (64+24 =88 bits)	lID_0+lIDold+lIDnew+l h(IDold)+lh(IDnew)+l IDtmp+sqn+sqntmp (64+64+64+64+64+64 +24+24) =432 bits
Ownership transfer stage phase two						
6H 1RN 2XOR 1CMP	3H 1RN 1CMP 1XOR	4H+O (1) 1XOR, 1CMP	$l_r+l_{a_4}$+lm (64+64+64 =192bits)	l_{a_3}+lhID+lt (64+64+64 =192 bits)	ID+sqntmp (64+24 =88 bits)	lID_0+lIDnew+lh(IDne w)+lIDtmp+lh(IDtmp) +sqntmp+sqnnew (64+64+64+64+64+24 +24) =368 bits

n: number of the registered tags in the database, l: the length of the message, T: tag side, R: reader side and S: server side

O (1): one operation required to get tag ID from the database

7. IMPLEMENTATION OF RFID TAGS AND READER TEST AND RESULTS

Our proposed protocol has been implemented and tested on a system using RF 433 MHz transmitter and receiver modules and ATmega328 microcontroller. The channel between the reader and server is safe (secure) since the reader is connected to the sever using USB cable (wire channel), while the reader to tag channel is considered as unsecure and vulnerable for Eavesdropping by attackers. The system setup is illustrated in Fig. 8 and the implemented system which consists of RFID server, reader and tag is shown in Fig. 9. The authentication and ownership transfer processes is controlled by visual studio GUI application and sql server shown in Fig. 10.

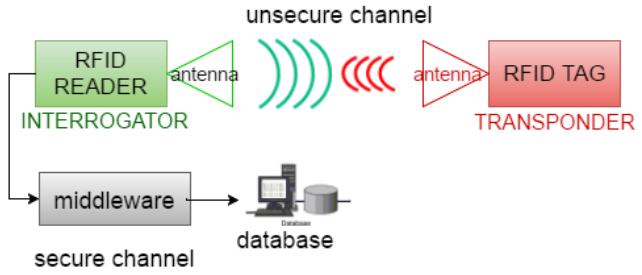


Fig. 8 System setup

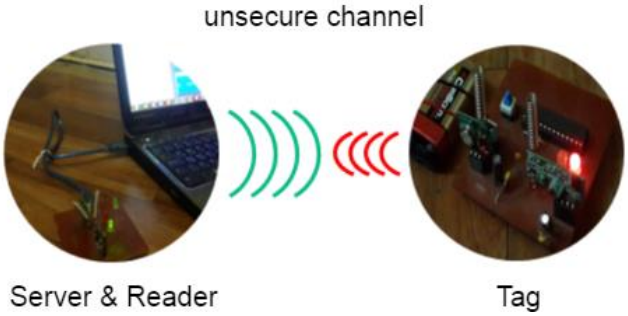


Fig. 9 implemented RFID system

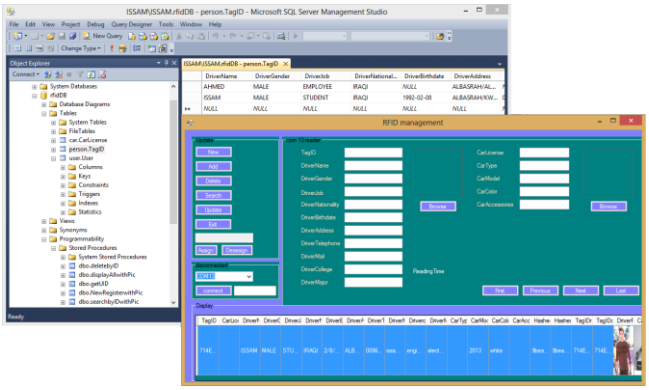


Fig. 10 System graphical user interface GUI and SQL server

7.1 Two Successful Authentication Sessions with & without ID Updating

As explained in suction VI, the authentication session is started when RFID reader request a tag by sending random number r**.** An example of two successful authentication sessions is presented; the first session begins when a specific tag with initial identifier ID_0 of 714E3D5F and system key sqn of 123 receive reader request with a random number r of 53543659. Then, the tag generates another random number t of 72854783 and computes both of a1 and hID, a1 = h_{32} (ID || t || r) = h_{32} (714E3D5F7285478353543659) = 086ae98d and hID = h_{32} (ID) ⊕ h_{32} (sqn || t) = h_{32} (714E3D5F) ⊕ h_{32} (12372854783) = 7bf3cabd ⊕ 1ae69fa5 = 61155518. Next, the tag replies with a1, hID and t back to the reader. Once receiving tag messages, the reader sends hID and t to the server (system database) in order to get $hID_{new\ or\ old}$ and the record index ID_0. Since this session is the first session after tag configuration, the database stores 714E3D5F in each of record index, IDold, IDnew and the MD5 hash value of 714E3D5F, i.e. h_{32} (714E3D5F) = 7bf3cabd in both of IDnew and IDold as illustrated in Table. IIII. When receiving hID and t, the server search for any $hID_{new\ or\ old}$ that equals to hID ⊕ h_{32} (sqn || t) = 61155518 ⊕ h_{32} (12372854783) = 61155518 ⊕ 1ae69fa5 = 7bf3cabd. Since the

database includes a record with hIDnew of 7bf3cabd, the server will forward IDc = IDnew = 714E3D5F (since hIDnew = hID ⊕ h_{32} (sqn || t)) from the database record with ID_0 = 714E3D5F to the reader by using of the same record index in order to verify the legitimacy of the involved tag. After receiving IDnew of 714E3D5F, the reader computes b1 = h_{32} (IDc || t || r) = h_{32} (714E3D5F7285478353543659) = 086ae98d. Since a1 equals b1, the tag is legitimate (here the reader takes an action of accessing the tag) and the reader forwards IDc.000 to the server. When the server receives IDc, it compares it with IDnew and IDold columns of the database records. Since IDc = TagIDnew = 714E3D5F for the record with index of 714E3D5F, the database updates its states so that IDold = IDnew = 714E3D5F, hIDold = hIDnew = 7bf3cabd, IDnew= h_{32} (IDnew || sqn) = h_{32} (714E3D5F123) = bfacbfe9 and hIDnew = h_{32} (IDnew) = h_{32} (bfacbfe9) = ce14ae6b then display the selected record. After database updating, a message of a2 = h_{64} (IDc || t) = h (714E3D5F72854783) = d7dc5e1ae6d32650 is sent by the reader to the tag for the purpose of updating its ID. When the involved tag receives reader message a2 of d7dc5e1ae6d32650, it computes b2 = h_{64} (ID || t) = h (714E3D5F72854783) = d7dc5e1ae6d32650. Since a2 equals b2, the reader message is legal and the tag updates its ID to IDnew = h_{32} (ID || sqn) = h_{32} (714E3D5F123) = bfacbfe9. At this moment, the mutual authentication session process is ended. The same procedure is performed in the server, reader and tag in the second session except for the tag ID updating in the tag side since a2 has been blocked from the tag. The first session called a completed successful authentication session since the tag updated its ID, while the second session called a successful authentication session since it verified a legitimate tag without updating its ID. In the end of these two sessions, the database server stores IDold = bfacbfe9, hIDold = ce14ae6b, IDnew = f9324ba7and hIDnew = c36b3131for the database record with index ID_0 of 714E3D5F as shown in Table. IIII. Fig. 11 shows the results of the two successful sessions in the reader and tag sides.

7.2 Replay and DoS Attacks

In order to prove the efficiency of our design and implemented protocol we have tested the system under replay and DoS attacks described previously.

In replay attack, the attacker can replay with the messages sent out by the reader through the legal previous session (reader to tag attack) or the messages sent by the tag (tag to reader attack). In reader to tag attack, the attacker tries to update tag ID in order to bring the tag and server out of the service (desynchronizing them) by replaying with the previous authentication reader messages r and a_2 or ownership transfer phase I or phase II reader messages r, m and a4. Since a2 depends on a fresh random number t generated by tag side which differs between one session and another and the last updated tag ID, a4 also depends on a fresh random number t generated by tag side and m is built upon tag ID which updated in the tag side after each completed successful session, the reader to tag attack fails. Tag to reader attack is more popular, it attempts to authenticate fake tags by only replay with the last previous authentication legal tag messages a_1, hID and t. By the reason of b_1 is using reader random number r which is not the same between two different authentication sessions, tag to reader attack is also fails. Fig. 12 shows that whatever times the attacker replies with the same a1, hID ant t of the previous legal authentication session which are b0491e13, af61156c and 84744174 respectively as shown in Fig. 11 (a), the tag will not authenticate since b_1 depends on fresh new generated random number r while a1 sent by the attacker contains r of the previous session which makes a1 not equal to b1.

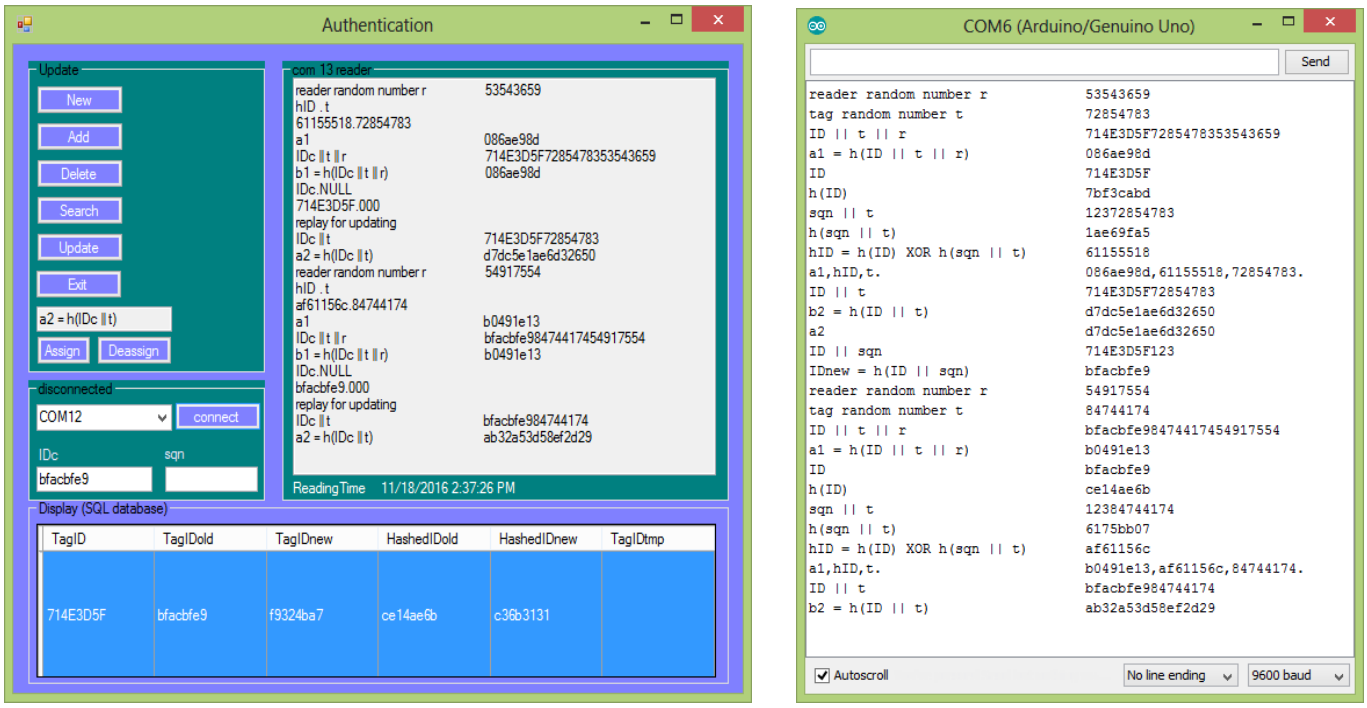
In DoS attack, the attacker target is to denial the service between RFID tags and reader by desynchronize them. In order to accomplish this goal, the server should have no ability to remember the old tag ID, IDold and its hash hIDold. Since the proposed protocol ensures to store IDold and hIDold in the database before updating to new ID and hID i.e. IDnew and hIDnew respectively in the mutual

authentication stage sessions while it stores IDtmp and hIDtmp in the ownership transfer stage sessions, DoS attack is completely fails. Fig. 11 (a) and (b) shows that, the tag did not receive a_2 = ab32a53d58ef2d29 due to blocking a_2 from the tag after a successful authentication session, so its ID will remain in its old state IDold = bfacbfe9. The next authentication session of this tag will be a1 = h_{32} (ID || t || r) = h (bfacbfe91074746218835226) = afd7c076, hID = h_{32} (IDold) ⊕ h_{32} (sqn || t) = h_{32} (bfacbfe9) ⊕ h_{32} (12310747462) = ce14ae6b ⊕ cfdeb078 = 01ca1e13 and t = 10747462. Once receiving hID and t, the server first computes h_{32} (sqn || t) = h_{32} (12310747462) = cfdeb078 then search for any

$hID_{new\ or\ old}$ that equals to hID ⊕ h_{32} (sqn || t) = 01ca1e13 ⊕ cfdeb078 = ce14ae6b which is existed in the database record with the record index ID_0 of 714E3D5F and hIDold of ce14ae6b. By using of the record index ID_0 of 714E3D5F the server forwards IDc = TagIDold = bfacbfe9 to the reader and the process of tag authentication will continue without server updating since IDc = IDold while the tag in the end of this session updates its ID to IDnew = h (ID || sqn) = h (bfacbfe9123) = f9324ba7 which is the same value stored in the IDnew column for the database record with ID_0 = 714E3D5F. as a result the tag and server in synchronizing again and DoS attack fails to denial the service between the tag and the server. Fig. 13 shows the results of the resynchronizing session in the reader and tag sides and Table. IIII shows the database states after one authentication session after the resynchronizing session process.

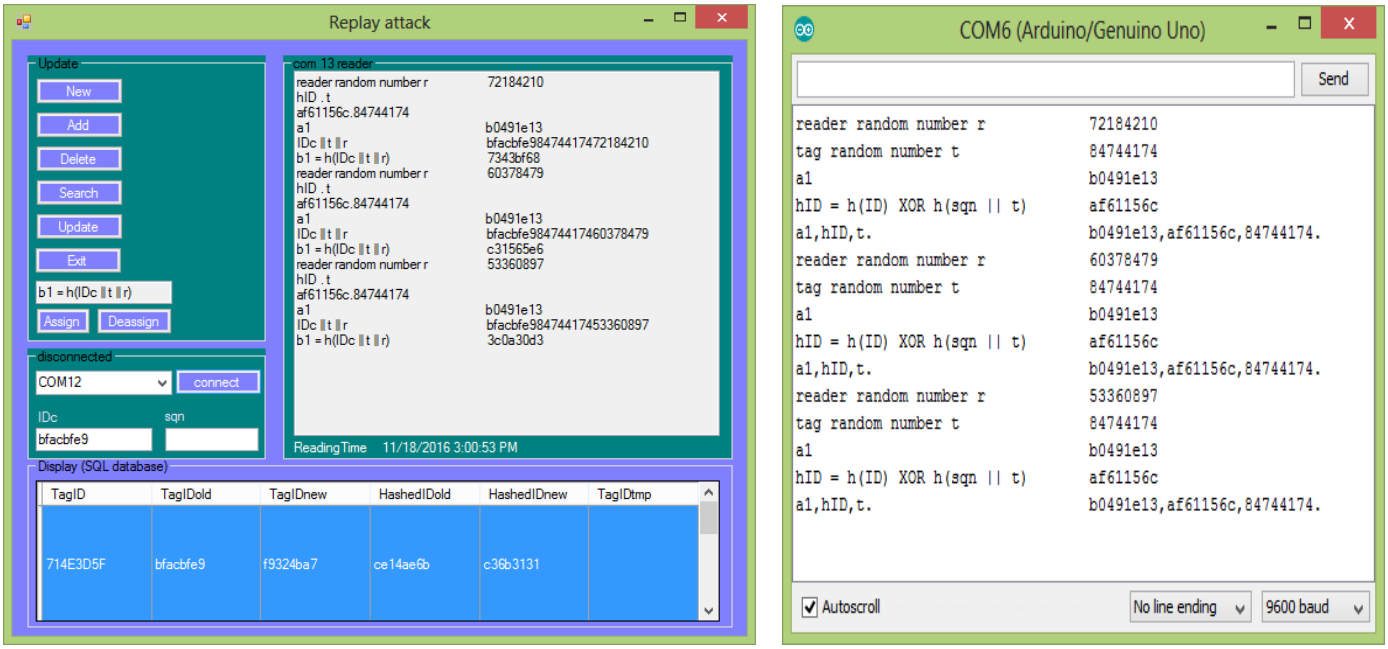
7.3 *Tag Ownership Transfer*

In order to transfer the ownership of a specific tag to new owner (database server), it conducts phase I and phase II illustrated in Fig. 6 and Fig. 7. If the concerned tag in transferring ownership has an ID of 4cf1e265 and a system key sqn of 123 and the old owner uses sqntmp of 456 to update tag secrets ID and sqn to IDtmp and sqntmp respectively while the new owner updates the temporary values of tag secrets IDtmp and sqntmp to IDnew and sqnnew respectively by using of a system key sqnnew of 789, the states of tag secrets will be ID = IDtmp = h_{32} (ID || sqntmp) = h_{32} (4cf1e265456) = bdfde48c and sqn = sqntmp =456 after the execution of phase I while ID = IDnew = h_{32} (IDtmp || sqnnew) = h_{32} (bdfde48c789) = 06838fde and sqnnew = 789 after the execution of phase II. Table. 4 shows server states after the execution of ownership transfer phase I and phase II.



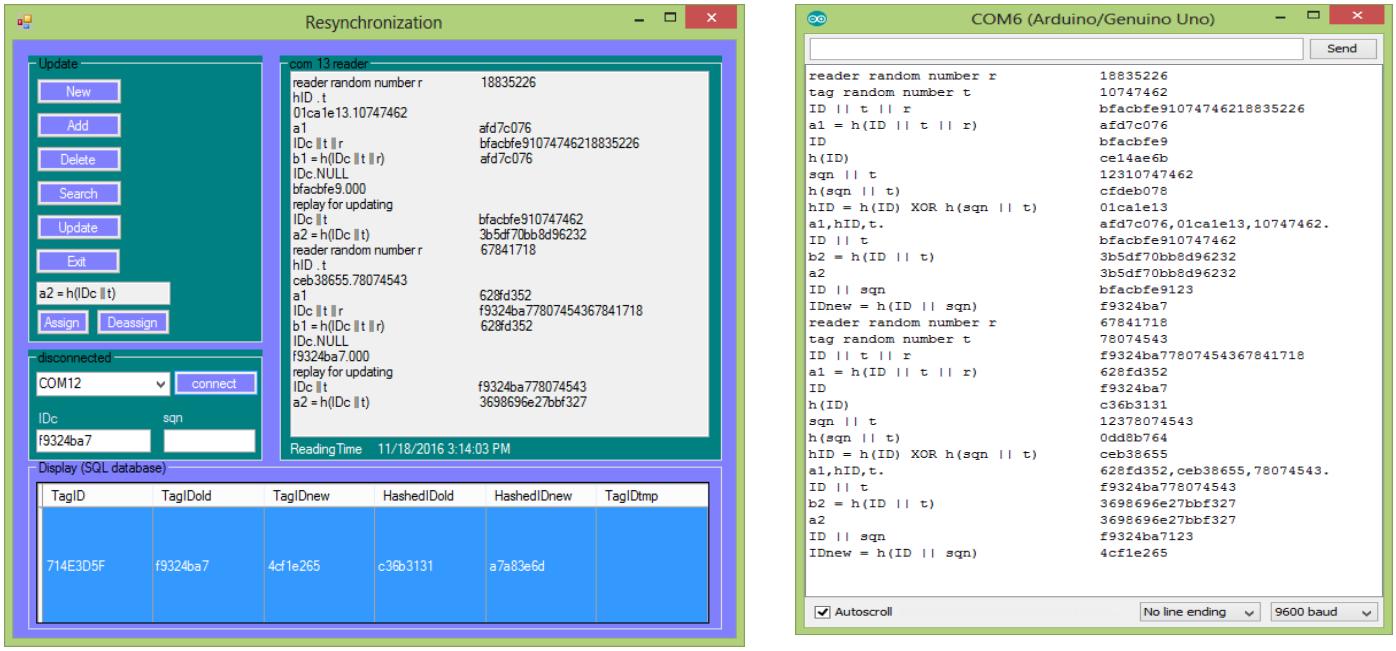
(a) (b)

Fig. 11 Two successful authentication sessions, (a) Reader side, (b) Tag side



(a) (b)

Fig. 12 Replay attack, (a) Reader side, (b) Tag Side



(a) **(b)**

Fig. 13 resynchronizing process after DoS attack, (a) Reader side, (b) Tag side

Table. 4 Server states after the execution of the proposed protocol

DB States / After sessions	ID_0	IDold	IDnew	hIDold	hIDnew	IDtmp
rfidDB (old owner) configuration	714E3D5F	714E3D5F	714E3D5F	7bf3cabd	7bf3cabd	NULL
Two successful authentication sessions	714E3D5F	bfacbfe9	f9324ba7	ce14ae6b	c36b3131	NULL
One session after resynchronizing	714E3D5F	f9324ba7	4cf1e265	c36b3131	a7a83e6d	NULL
Ownership transfer p1	714E3D5F	f9324ba7	4cf1e265	c36b3131	a7a83e6d	bdfde48c
RfidDB (new owner) configuration	714E3D5F	NULL	NULL	NULL	NULL	NULL
Ownership transfer p2	714E3D5F	NULL	06838fde	NULL	fcc82f4c	NULL

8. CONCLUSION

In this paper, we proposed a fully privacy-preserving hash-based protocol with ownership transfer stage. Our protocol can handle all common RFID attacks, such as eavesdropping, replay, Spoofing, man in middle, cloning and denial of service attacks. It also prevents the tag from being tracked and it achieves the privacy requirements such as anonymity, integrity, confidentiality, forward security etc. The ownership of the tag may change during its life cycle, so we proposed an ownership transfer stage to ensure the privacy of the new owner as well as the old owner. The purpose of the ownership transfer stage is to give the access and control of the tag to the new owner and prevent the old owner from accessing, controlling or tracking the tag. Time complexity is one of the main issues in designing RFID protocols. To design a scalable protocol, it must be not linear i.e. the required operations to find the tag identifier in the database must not equal to the number of tags in database, O(n). Our protocol is constant, i.e. it needs just one operation O (1) to get tag identifier from DB, so it is scalable and can be applied in a dense RFID networks. The proposed protocol has been implemented and tested against replay and DoS attacks in RFID tags and reader built from ATmega328 microcontroller and RF 433 MHz links. The ATmega328 microcontroller provides the ability to perform complex operations such as symmetrical encryption algorithms, unsymmetrical encryption algorithms, one-way hash functions, bitwise logical XOR and pseudo random number generators and gives enough storage to implement any RFID authentication protocol.

References

[1] Landt J.: The history of RFID. pp. 8-11, October/November (2005)

[2] Kaur, M., Sandhu M., Mohan N., S. Sandhu, P.: RFID Technology Principles, Advantages, Limitations & Its Applications. International Journal of Computer and Electrical Engineering. Vol.3, No.1, pp. 151-157, February (2011)

[3] N. Nambiar, A.: RFID Technology: A Review of its Applications. Proceedings of the World Congress on Engineering and Computer Science Vol II. San Francisco, USA, 20-22 October (2009)

[4] Pateriya, R.K., Sharma, S.: The Evolution of RFID Security and Privacy: A Research Survey. International Conference on Communication Systems and Network Technologies. 115-119 (2011)

[5] Ahsan, K., Shah, H, Kingston, P.: RFID Applications: An Introductory and Exploratory Study. IJCSI International Journal of Computer Science Issues. Vol. 7, Issue 1, No. 3, pp. 1-7, January (2010)

[6] Juels, A., L Rivest, R., Szydlo, M.: The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy. Proceedings of the 10th ACM conference on Computer and communications security. New York, USA, 103-111 (2003)

[7] Juels, A.: RFID Security and Privacy: A Research Survey. IEEE Journal on Selected Areas in Communications. VOL. 24, NO. 2, pp. 381-394, February (2006)

[8] A. Weis, S., E. Sarma, S., L. Rivest, R., W. Engels, D.: Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. First International Conference. Boppard, Germany, pp. 201-212, 12-14 March (2003)

[9] Ohkubo, M., Suzuki, K., Kinoshita, S.: Cryptographic Approach to "Privacy-Friendly" Tags. RFID Privacy Workshop. MIT. MA. USA (2003)

[10] Molnar, D., Wagner, D.: Privacy and Security in Library RFID Issues, Practices, and Architectures. Proceedings of the 11th ACM conference on Computer and Communications Security. Washington DC, USA, 210-219 (2004)

[11] Dimitriou, T.: A Lightweight RFID Protocol to protect against Traceability and Cloning attacks. Proceeding of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05). 59-66 (2005)

[12] Juels, A., A. Weis, S.: Authenticating Pervasive Devices with Human Protocols. Victor Shoup, editor, Advances in Cryptology – CRYPTO'05, volume 3126 of LNCS, California, USA, 293-308 (2005)

[13] Juels A., A. Weis, S.: Defining Strong Privacy for RFID. Proceedings of Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW '07). New York, USA, 342-347 (2007)

[14] Lee, K.: A Two-Step Mutual Authentication Protocol Based on Randomized Hash-Lock for Small RFID Networks. International Conference on Data and Knowledge Engineering (ICDKE). Melbourne, Australia,1-3 September (2010)

[15] Gui, YQ., Zhang, J.: A New Authentication RFID Protocol with Ownership Transfer. International Conference on ICT Convergence (ICTC). Jeju, South Korea, 359-364 (2013)

[16] Edelev, S., Taheri S., Hogrefe, D.: A Secure Minimalist RFID Authentication and an Ownership Transfer Protocol Compliant to EPC C1G2. IEEE International Conference on RFID Technology and Applications (RFID-TA). 126-133 (2015)

[17] Tuyls, P., Batina, L.: RFID-Tags for Anti-Counterfeiting. D. Pointcheval, editor, Topics in Cryptology-CT-RSA, vol. 3860, LNCS Springer, Verlag, 115-131 (2006)

[18] Bringer, J., Chabannel, H., Icart, T.: Cryptanalysis of EC-RAC, a RFID Identification Protocol. Proceedings of the International Conference on Cryptology and Network Security - CANS'08. Hong Kong, China, LNCS, Springer-Verlag. 149-161 (2008)

[19] Deursen, T., Radomirovi´c, S.: Untraceable RFID protocols are not trivially composable: Attacks on the revision of EC-RAC. Cryptology ePrint Archive: Report 2009/332 (2009)

[20] Lee, Y., Batina, L., Verbauwhede, I.: Untraceable RFID Authentication Protocols: Revision of EC-RAC. Proceedings of IEEE International Conference on RFID. Florida, USA, 178-185 (2009)

[21] Kavun, E., Yalcin, T.: A lightweight implementation of Keccak hash function for radio-frequency identification applications. Radio frequency identification: security and privacy issues. vol. 6370, Springer Berlin/Heidelberg, 258–269 (2010)

[22] Aumasson, J-P., Henzen, L., Meier, W., Naya-Plasencia, M.: Quark: a lightweight hash. Cryptographic hardware and embedded systems. CHES 2010, vol. 6225, Springer Berlin/Heidelberg, 1–15 (2010)

[23] Osaka K., Takagi, T., Yamazaki, K., Takahashi, O.: An Efficient and Secure RFID Security Method with Ownership Transfer. Computational Intelligence and Security, Vol. 2, 1090-1095 (2006)

[24] Lei, H., Cao, T.: RFID Protocol enabling Ownership Transfer to protect against Traceability and DoS attacks. First International Symposium on Data. Privacy and E-Commerce, pp. 508-510, Nov (2007)

[25] Kulseng, L., Yu, Z., Wei, Y., Guan, Y.: Lightweight Mutual Authentication and Ownership Transfer for RFID Systems. *IEEE INFOCOM*. pp. 1–5, Mar (2010)

[26] Song, B., J. Mitchell, Chris.: Scalable RFID security protocols supporting tag ownership transfer. Computer Communications 34. 556–566 (2011)

[27] Miyaji, A., S. Rahman, M., Soshi, M.: Efficient and Low-Cost RFID Authentication Schemes. Journal of Wireless Mobile Networks. Ubiquitous Computing, and Dependable Applications (JoWUA), 2(3), pp. 4-25, Sep (2011)

[28] Chen, CL., Huang, YC., R. Jiang, J.: A Secure Ownership Transfer Protocol Using EPCglobal Gen-2 RFID. Telecommunication Systems 53(4), 387-399 (2013)