Cumhuriyet Üniversitesi Fen Fakültesi
Fen Bilimleri Dergisi (CFD), Cilt:36, No: 3 Özel Sayı (2015)
ISSN: 1300-1949

Cumhuriyet University Faculty of Science
Science Journal (CSJ), Vol. 36, No: 3 Special Issue (2015)
ISSN: 1300-1949

A multi authentication scheme for RFID systems

Fereshteh GHASEMİ[1], Reza.PARSAMEHR[2,*]

[1]*Institute for Advanced Studies in Basic Sciences, Zanjan, Iran; Email: ghasemi.fereshteh.2014@gmail.com*

[2]*Institute for Advanced Studies in Basic Sciences, Zanjan, Iran*

Received: 01.02.2015; Accepted: 05.05.2015

Abstract. In this paper, a new design and robust security in RFID system is proposed which is based on hidden data synchronization. Scheme prepare multi authentication that authenticate tag and reader to server and server for tag and reader, unlike the study protocol was doing. It means that the relationship between Tag- reader and reader-server are both unsafe assumptions. Hash functions used in the project are limited and some Hash values already stored by the server on tag, until the tag will have to be calculated every time the query is computed only once we hash in tag. TAG ID in any way change the query that is used to securely synchronize done and makes the protocol attacks such as eavesdropping, relay, tracking and DOS is resistant. We have used PRNG function to generate pseudo-random numbers. Security analysis relevant to this project shows that the proposed scheme is robust against attacks.

Keywords: Security, RFID, authentication and privacy

1. INTODUCTION

Radio Frequency Identification, abbreviated ‘‘RFID’’, basically provides a means to identify objects having RFID tags attached. Fundamentally, RFID tags provide the same functionality as barcodes but usually have a globally unique identifier. Using RFID, the identification is performed electromagnetically. Thus, there is, in contrast to barcodes, no line-of-sight necessary, and the identification can also be performed in contactless way. RFID also has the advantage that bulk reading is possible and that it is not susceptible to dust, dirt, or vibration like barcodes. Because of these characteristics, RFID is envisioned to be a convenient replacement for optical barcodes in the future. Unfortunately, RFID also introduces problems respecting data security and privacy arises. RFID systems have three main components:

RFID tags: RFID tags are used as a label for item identification and communicate with a reader. The reader passes tag data to the backend server for further processing, including tag identification and information retrieval.

RFID readers: RFID readers send and receive data to and from tags. RFID readers are the connecting element between the RFID tags and the backend systems.

Backend server: Readers are used for querying tags and reading and writing tag data. All the read data need to be processed, and the data to be written need to be available, so that an additional system component is required to form a complete RFID system which is the backend server.

Due to the advantages of RFID tags that are relative to the barcodes, RFID and more popularity and are expected to be in the near future to replace barcodes. There is an urgent need to provide more security to the technology viable for the work of the public. Therefore, privacy and authentication are two main security issues that must be considered for RFID.

*Corresponding author. *Email address: parsamehr@iasbs.ac.ir*

Special Issue: The Second National Conference on Applied Research in Science and Technology

2. RFID SECURITY ISSUES

Characteristics of RFID systems can cause various attacks and thus lead to the disclosure of sensitive information. In most of the studies we reviewed, papers have been under attacks.

DOS (Denial of Service) attack: Adversaries can disable the system by sending excessive data or simply shielding the RFID device to keep it from operating. Moreover, they can intercept the transmitted information so that the tag and the server are unable to update their shared secret data synchronously, in which case the following authentications and accesses fail.

Impersonation attack: Attackers can masquerade as the reader or the tag to pass the authentication by falsifying data and thereby gain illegal advantages.

Replay attack: Adversaries can intercept transmitted information and resend it illegally in an attempt to deceive a legal device and pass the authentication.

Eavesdropping attacks: the purpose of eavesdropping attacks, unauthorized surveillance information in a system.

Location tracking: By using a malicious reader, the adversary can acquire the tag information and find which tag it belongs to. To perform an attack, the adversary transmits the query continuously to the tag being traced. By this means, the location of the tag owner is exposed and privacy can be violated.

Asynchronous: Many authentication protocols use an update process for tracking robustness against attacks. Update process causes that after a successful meeting between the tag and reader, the parties set values including ID, keys and other secret values corresponding to the update and at the next meeting they will use of new values. Updates the values causes the secret information even if an attacker to gain access to a label at time t, others unable to tracking transactions between the tag and reader in time t' > t, Because the values used for authentication tag at times t and t' are different. However, this feature causes the other vulnerability for other protocols will be discussed. In a non-synchronization attack, the attacker tries different methods that server and tag updates different values, This leads to the secret values stored in the server not the same tag And therefore the next meeting, when the tag is intended to authenticate itself to the server, Due to the mismatch of the values it holds with values that are available in the server, the server refuses authentication tag and tag will delete of the system.

3. RELATED WORK

PAP is an authentication and privacy protocol for passive RFID tags. In PAP, each tag has a secret numeric value, for which a reader and a tag establish authentication. Upon verification of the reader by the tag, the tag sets itself to a state that upon query, only gives an authenticated reader enough information to change the tag to a prior state and release its EPC information. However, the information given in this state is also general enough to not allow an unauthenticated reader to gain access to the EPC code or know what the product is, thereby establishing privacy. This protocol is practical and useful for two reasons. First, it requires only an extremely small amount of computation; therefore, it has the capacity to be implemented within passive RFID tags. Second, this protocol deals with both privacy and authentication. [1] But since tag always sends constant values corresponding to the ID, it is not resistant against tracking attacks and an adversary can impersonate a tag using the answers provided by a second legitimate reader and exploiting the symmetry of the messages computed by the reader and the tag in the PAP–see principle four for designing cryptographic protocols. So, an impersonation attack can be conducted between an adversary and two legitimate readers. [2]

SRFID is a new simple, low cost, and scalable security scheme relying on one-way hash functions and synchronized secret information. The proposed scheme provides a two steps mutual authentication between the backend server and the tag. The proposed scheme meets the requirements for tag delegation and secure tag ownership transfer. The general idea is to change the ID of a tag on every read attempt in a secure and synchronized manner. The proposed scheme requires a little calculation amount, because it needs a one-way hash function and it is suitable for a variety of tags. All relevant tag data store server. Tag content is indexed by a unique ID. The system is scalable and efficient search. Tag sends its current ID to reader that it is transmitted to the server as an index in the database. To combat the Impersonation attack, plan based on mutual authentication between the tag and the server. Mutual authentication based on shared secrets between the tag and the server. After successful mutual authentication, tag ID which is updated by both tag and server security system provides forward secrecy. This hidden updates mechanisms may lead to unsynchronized attacks. To prevent this attack, the server holds the current and previous records of update process. When the server does not authenticate a tag due to the unsynchronized attack, Previous ID of update previous secret record returns to complete the authentication. [3]

In 2008, Chen et al. proposed an RFID authentication scheme which can enhance security and privacy by using hash functions and quadratic residues. However, their scheme was found to be vulnerable to impersonation and repeated attacks and does not protect the privacy of the location. In 2011, an improved design of Chen scheme proposed which can prevent possible attacks, and in environments that require high levels of security are applied. The plan to prevent security threats mentioned, tag is allowed to generate a random number for each tag reading, to protect the data. [4]

Jung-sik Cho proposed a RFID mutual authentication protocol based on hash that uses random numbers generated by the tag and the reader to make the message random and secret information is secure against eavesdropping, because of hash functions. [5]

Hyunsung Kim protocol shows that the Jung-sik Cho protocol is weak against synchronization attacks and Attacker could perform DE synchronization attack between two parties by simply modifying the final message from reader to tag. Tag cannot be used by the server that is a type of attack focuses on accessibility and proposed a modified protocol that provides high security with key-based synchronized hash function. [6]

Another scheme present a novel approach to authentication and privacy in RFID systems based on the minimum disclosure property and in conformance to EPC Class-1 Gen-2 specifications. In this scheme, Cyclic Redundancy Check (CRC) and the pseudo random number generator (PRNG) used to be a passive RFID tag is able to do it. Scheme provides mutual authentication for RFID applications that is fit for constant reader that the channel between the reader and the server is secure. The cheapest hash function will cost approximately $1{:}7k$ gates to implement. In this scheme, the operators of tag is limited to module square computation, bit operators (XOR, multiplication), CRC calculation and random number generator, which only requires a few hundred gates, which is much cheaper than most simple hash function. Improved QR in comparison with the scheme, the scheme is consistent with the characteristics of EPC While improved QR is not. As the number of tag in the system increases, the authentication delay further. However, both schemes have good performance. [7]

4. OUR PROPOSED SCHEME

Our scheme is for solve the problems in the studied protocols. In the studied schemes only the tag and the server are authenticate for each other, and the connection between the server and the reader is assumed to be safe. Here, in addition to the authentication server and the tag to each other, reader and server are authenticating for each other. The relationship between tag - reader and reader-server both are assumed unsafe.

Studied protocols, except Cho did not speak of the reader and the tag information and only mutual authentication process is presented. In our scheme, after the tag and reader are both authenticated to server, server do XOR between tag stored data in the server with a random number generated for that meeting and then send the encrypted data and identity token to the reader.

According to the conclusions drawn from the studied protocols, restricted use hash functions and values already stored by the server on tag that tag will not have to calculate every time in every query.

4.1 Description of our scheme

The scheme includes two stages: 1) initialization 2) authentication and data delivery. The following steps of authentication are shown in figure 1. Symbols used in this study include:

ID: Tag identifier
h(ID): tag ID hash value is used as a database index
ID_r: reader ID
$h(ID_r)$: The hash value of the reader ID that can be used as a database index
r_r , s: the random numbers generated by the reader
r_t: random number generated by the tag
h(): one-way hash function
PRNG: pseudo-random number generator
K: the current authentication key that is shared between the tag and the server
K_{old}: Previous authentication key stored in the server

1) Initialization: the server generates random number k and (ID, h (ID), k, β) is stored in the tag memory where β is a shared secret value and (ID_r, $h(ID_r)$, β) is stored in the authenticated reader and (ID, k_{old}, h(ID), k, β, DATA) and (ID_r, $h(ID_r)$) is stored in the database that at the first k= k_{old}.

2) Authentication:

Step 1: reader generates random number s and with a query sends to tag.

Step 2: tag produce $\alpha = h(ID) \oplus s \oplus k \oplus r_t$ and $v = \beta \oplus r_t$ where r_t is a random number generated by the tag. Then the tag sends v and α to the reader.

Step 3: reader calculates $L = h(ID_r) \oplus \beta \oplus r_r$ and $w = \beta \oplus r_r$, and then sends L and w with α and v received from tag to the server.

Step 4: The server by using the received v and w and the amount of hidden β achieves r_r and r_t values. $r_t = \beta \oplus v$ and $r_r = \beta \oplus w \oplus r_t$.

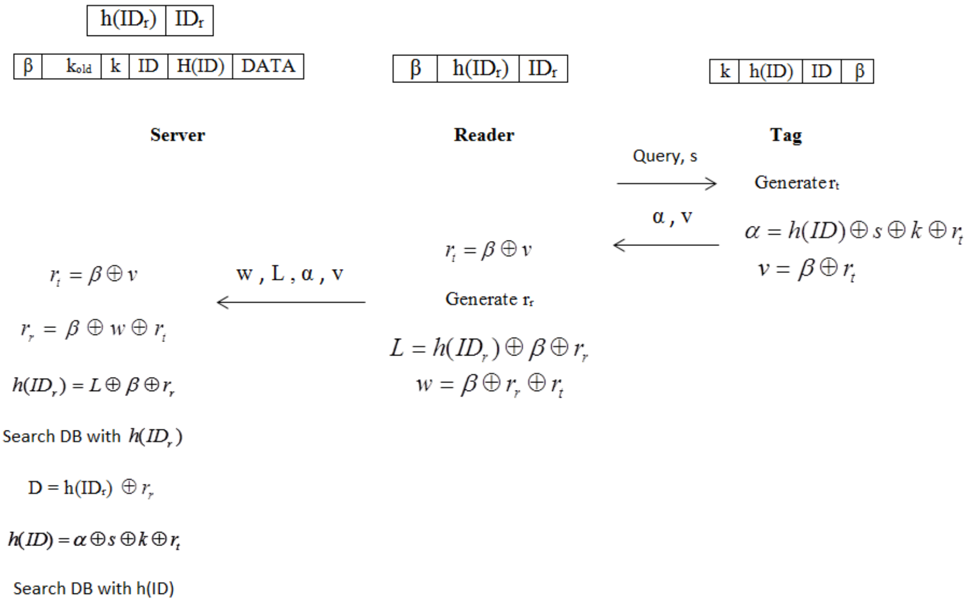
Then calculates $h(ID_r) = l \oplus \beta \oplus r_r$ and searches the database with h($_{\mathrm{IDr}}$). If a matching record is not found, the session remains inconclusive and no reader authentication. If the reader authenticated, calculates $D=h(ID_r) \oplus r_r$ and then calculates $h(ID) = \alpha \oplus s \oplus k \oplus s \oplus r_t$ and uses from h (ID) as an index to find relevant tag record .If a matching record is not found, the session remains inconclusive and no authentication tag. If the tag authenticated, thus k is investigated. If is not equal k or stored k_{old}, the session is canceled.

(i) If the resulting k is identical to k in the corresponding tag record, the server calculates $C=h(ID) \oplus k \oplus r_t$ and updates tag record by replacing k_{old} with k with PRNG (k). Then C and D with DATA $\oplus$ r_r are sent to the reader.

(ii) If the resulting k is identical to k_{old} in the corresponding tag record, the server calculates $C=h(ID) \oplus k_{old} \oplus r_t$ and none of the k and k_{old} on servers are not updated. Then C and D with DATA $\oplus$ r_r are sent to the reader.

Step 5: reader receives DATA $\oplus$ r_r and C and D and finds the DATA and then calculates $D' = h(ID_r) \oplus r_r$, the reader then verifies whether the D is equal to D'. If it holds, server is authenticated and C sends to the tag.

Step 6: tag calculated $C'=h(ID) \oplus k \oplus t$ and compared with the received C. In case of equality, server is authenticated, and k updates with PRNG (k).


h(ID$_r$) | ID$_r$
β | k$_{old}$ | k | ID | H(ID) | DATA
β | h(ID$_r$) | ID$_r$
k | h(ID) | ID | β
Server
Reader
Tag
Query, s
Generate r$_t$
α , v
$\alpha = h(ID) \oplus s \oplus k \oplus r_t$
$v = \beta \oplus r_t$
$r_t = \beta \oplus v$
Generate r$_r$
$L = h(ID_r) \oplus \beta \oplus r_r$
$w = \beta \oplus r_r \oplus r_t$
w , L , α , v
$r_t = \beta \oplus v$
$r_r = \beta \oplus w \oplus r_t$
$h(ID_r) = L \oplus \beta \oplus r_r$
Search DB with $h(ID_r)$
D = h(ID$_r$) $\oplus r_r$
$h(ID) = \alpha \oplus s \oplus k \oplus r_t$
Search DB with h(ID)

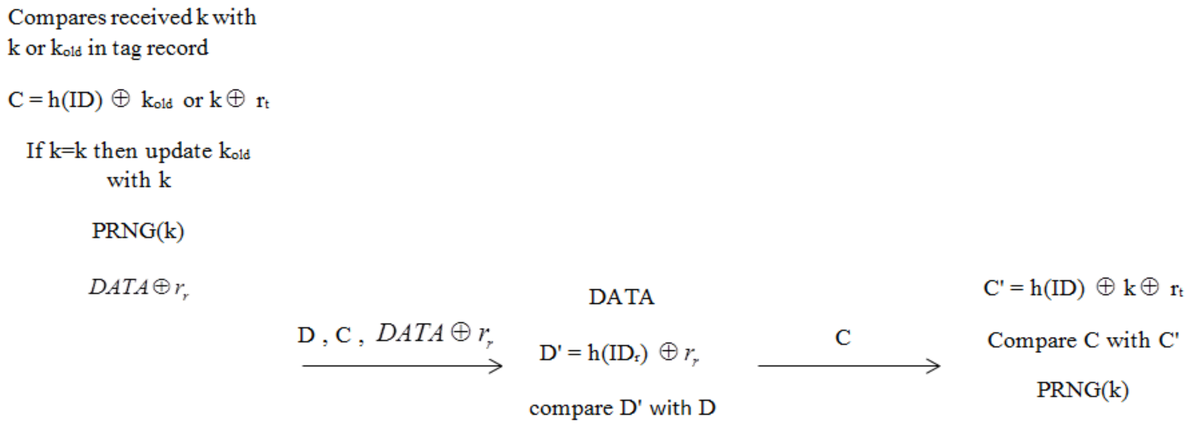

Compares received k with k or k_{old} in tag record
$C = h(ID) \oplus k_{old}$ or $k \oplus r_t$
If k=k then update k_{old} with k
PRNG(k)
$DATA \oplus r_r$
D , C , $DATA \oplus r_r$
DATA
$D' = h(ID_r) \oplus r_r$
compare D' with D
C
$C' = h(ID) \oplus k \oplus r_t$
Compare C with C'
PRNG(k)

Figure 1: The proposed scheme

4.2 Security analysis

- Location privacy tag: As the tag in every query generate random number r_t, so make different α and v and unexpected response of the tag. If the tag does not generate random number r_t, and only construct α, the attacker is able to prevent or modify C when sending the reader to the tag, prevents tag's k to update, So with querying tag again with equal s, the value obtained for α as before came and tracing attack was carried out successfully. But this scheme is to preserve the location privacy of the tag.
- Repeat attack: the attacker cannot be reused messages used in the previous session, if the attacker wants to prevent C in the sending of reader to tag to prevent of updating k in tag and it wants to impersonation in the next query, due to the use of different random number r_t and C in each time would not be possible at any time. And also because the reader uses a random number r_r in calculating L, an attacker cannot forge the reader with an L at the next meeting and D is the same. Therefore, our scheme is robust against this attack.
- Impersonation attack: random number r_t makes unexpected the response of tag on each step and the reader and server response are unexpected due to using r_t at each stage and attackers cannot obtain the secret values and forging their identity.
- DOS Attack: Server always holds k_{old}. If the value of C is sent from reader to tag prevented or modified, K and k_{old} stored on the server can be updated, but tag's k is not updated. Tag's k_{old} can still be adjusted by server's k_{old}. DOS attack is prevented by updating secret key in a same time.
- Eavesdropping attack: values that from the attacker eavesdropping can be obtained are: s, v, w, α, L, C, D and DATA$\oplus r_r$ respectively. DATA XOR with r_r, and protected. r_r XOR with β, and protected and β is a hidden value between legal tag and reader and server. ID_r and ID are hashed and protected by XOR.
- Confidentiality: data from the tag (ID, h (ID), k, β) cannot be retrieved of communications, and the confidentiality is established.
- Authentication: server receives α and computes h (ID) and finds h (ID) in its database to confirm the identity of the tag. Only a valid tag can build up properly α, since it has h (ID) and the common current key. Tag receives C and computes C', and compares them to authenticate the server. Only a legal server can build C properly. Server with receiving L and computing h (ID_r) and finding h (ID_r) in its database, authenticates the reader. Only a legal reader can make L. and reader with receiving D and computing D' and comparing them can confirm the server. Only a valid server can build up to D properly.

5. CONCLUSIONS

This article proposed a scheme that with consider all the points is tried to be robust against different attacks. In the proposed scheme, server and tag are authenticating to each other, reader and server are authenticating for each other too. It means the relationship between tag-reader and reader-server are both unsafe assumptions. In The reviewed protocols, server and tag authenticate for each other only and channel between the server and the reader is assumed to be safe. If the server tries to put data to the reader, reader should verify the identity.

In The reviewed protocols, there is not any present about the received information by the reader is. Secure protocols for sending data must to be used too. In The proposed scheme, after that tag and reader are both authenticate by the server, server do XOR tag's stored data with a random number is generated for the session and then sends hiding data and it's to the reader.

To prevent of DE synchronization attack between the tag and the server, the server also keeps a secret that is used for the previous meeting that if this attack runs and the hidden values of tag don't updating, the next query to tag, with comparing the session secret value, if it is not update, to be updated.

If the server's transferring data is prevent by an attacker to obtain confirmation of his identity and don't allow the tag hidden values to be updated, the attacker can impersonate the server with requiring tag and sending receiving data. Therefore, to prevent a repeat attack, server's response must be different. in this scheme to create this situation, the tag generates a random number and give it to server that it can make different responses with that random number too.

Due to the Chen improved protocol, to create tag location privacy, tag response must be different for each query and if the attacker prevent or modify transmission data in the process of authentication and don't allow to update shared secret value between the tag and the server, and thus cause to returns a response similar to the previous query with the tag in the next query, tag must generate a random number in each query to make different responses. In The proposed scheme to avoid tracing tag, the tag generate a random number that makes different responses.

REFERENCES

[1] A.X. Liu, L.A. Bailey, "PAP: a privacy and authentication protocol for passive RFID tags", Computer Communications 32, 1194-1199 , 2009.

[2] Mu'awyaNaser, Pedro Peris-Lopez, Rahmat Budiarto, Benjamin Ramos Alvarez, "A note on the security of PAP" , Computer Communications 34, 2248–2249, 2011.

[3] WalidI. Khedr, "SRFID: A hash-based security scheme for low cost RFID systems", Egyptian Informatics Journal 14, 89-98, 2013.

[4] Tzu-Chang Yeh, Chien-Hung Wua, Yuh-Min Tseng, "Improvement of the RFID authentication scheme based on quadratic residues", Computer Communications 34, 337-341, 2011.

[5] Jung-sikCho, Sang-Soo Yeo, Sung Kwon Kim, "Securing against brute-force attack: A hash-based RFID mutual authentication protocol using a secret value", Computer Communications 34, 391-397, 2011.

[6] HyunsungKim, "RFID Mutual Authentication Protocol based on Synchronized Secret", International Journal of Security and Its Applications, Vol. 7, No. 4, 2013.

[7] Robin Doss, Wanlei Zhou, Shui Yu and Longxiang Gao, "A Novel Mutual Authentication Scheme with Minimum Disclosure for RFID Systems", ISSNIP, 544-549, 2011.