Secure Mutual Authentication Protocol for Low-Cost RFID

O. Omolola
Department of Computer Science
University of Ibadan
Ibadan, Nigeria
Email: omololaolamidex [AT] gmail.com

O. Osunade
Department of Computer Science
University of Ibadan
Ibadan, Nigeria

Abstract— **Radio Frequency Identification (RFID) is becoming an integral part of the society. It is capable of contactless communication and it is replacing barcodes. However with the introduction of RFID, new security threats are also introduced. In this paper, we propose a new protocol that builds on an existing protocol that was proposed in [4]. The protocol proposed in in an earlier work was chosen because it is lightweight and it can work in low cost RFID tags unlike most of the others. The new protocol named SMAP – Secure Mutual Authentication Protocol makes use of Psuedo-random number generator instead of cyclic redundancy check (CRC) because of bad linear properties CRC has. SMAP is needed at this time because most of the other protocols were discovered to be flawed and insecure. A lot of the existing protocols were also not truly applicable to low cost RFID tags. The new protocol is simulated and formally verified with CASPERFDR, a formal protocol verifier.**

Keywords- ***RFID, Security, low-cost, protocol design***

I. INTRODUCTION

Radio Frequency Identification (RFID) technology uses radio waves to identify people or objects. The RFID has been commercially available since the 1970 [11]. Information can be read from the RFID device at a distance. The device does not require contact with reader and it does not necessarily have to be in the line of sight of the reader. Data is exchanged automatically and does not usually require operator intervention. Intermec Technologies Corporation gives detailed advantages of RFID over other forms of data collection (Intermec, 2007). Some of the advantages are as follows:

- The data on the tags can be changed repeatedly
- More than a thousand reads can be performed per second on RFID tags
- Organizations have harnessed the power of RFID to monitor their processes, provide real time data accuracy, and track their assets and inventories.

RFID offers great value for many industries and applications and can be used for access control, manufacturing automation, maintenance, supply chain management, parking garage management, automatic payment, tracking, and inventory control.

The RFID system is made up of three components: RFID tag or transponder, Reader or encoder or interrogator, and backend server.

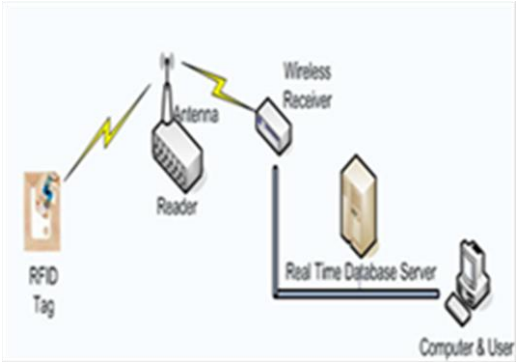

Figure 1: RFID System (www.ukrfid.co.uk)

RFID tag is a microchip that has a certain amount of computational and storage capabilities attached to a small flat aerial and mounted on a substance. The whole setup can then be enclosed in materials (such as plastic, rings) dependent upon its intended usage. The tag can be attached to an object, typically an item, box, or pallet, and read remotely to ascertain its identity, position, or state. There are two types of RFID tags: Passive and Active. The active RFID tag has a power source for example a battery. The passive RFID tag relies on energy transferred from a reader to power up and transfer its information.

A reader queries tags to obtain tag contents through wireless communications and sends this information to the backend server through a secure channel. The backend server is composed of a database and some processors.

RFID systems are susceptible to different kinds of malicious attacks such as passive eavesdropping and active interference. This usually poses a huge risk to the individuals or the organizations that use them. A typical tag replies with its ID to any reader that queries it. This makes it easy for an attacker to hack the system easily or create bogus tags. A RFID tag needs to be protected so as to ensure privacy. Even though many cryptographic primitives can be used to remove these

vulnerabilities, they cannot be applied to a RFID system due to the prohibitive cost of including protection for each and every RFID tag [5].

II. RELATED WORKS

Research on RFID security can be classified broadly into two categories. Protocol based is the first category. The emphasis here is on creating better protocols using mostly lightweight cryptographic primitives that can be implemented in RFID tags. The second is the hardware based research category, which focuses on enhancing the RFID tag and making it able to handle additional security primitives. Interested readers can refer to [21] for more details since this paper is focusing on the first category. Many protocols have been proposed under the first category but most of them were proven not to be secure and others were not suitable for low-cost RFID tags.

Weis proposed in [24] a protocol using a back-end database for performing RFID authentication. A reader queries the RFID tag and receives a metaID which it forwards to the database. The database then retrieves the real tag ID for the reader. Every time a tag is queried, it replies with the same MetaID. This causes a privacy problem because the tag can be tracked as a result. The authors then introduced a hashing locking scheme to solve the problem.

Molnar and Wagner however showed in [16] that the hash lock scheme does not protect the system against an eavesdropper. The authors in [16] then suggested that both the reader and the tag should contribute a random number r_1 and r_2 respectively. The tag secret is stored in the reader instead of the database. The issue with this protocol is the fact that once a reader is compromised, all tags that the reader has access to can easily be duplicated to fool other readers.

It was argued in [10] that the communications in RFID system could be made secure with a 3-round protocol based on a one-way hash function and a random number generator. The protocol begins as the tag is queried by the reader and the tag replies with its own hashed identification and the current transaction number. The reply is then sent to the server by the reader to check the validity of the identifying information of the tag. The server then concludes by sending a random number to the tag so the tag's identification is refreshed i.e. changed and synchronized. This scheme was discovered to be insecure in [6]. The scheme was found to be susceptible to man-in-the-middle attack while also falling prey to an attack based on de-synchronization of the counters shared by the tag and back-end server. Furthermore, the scheme does not guarantee untraceability.

A challenge – response mechanism for RFID authentication was introduced in [6]. However, the tag identifier remains constant between two successful sessions. As a result, the protocol is vulnerable to tracking attacks and tag impersonation attacks.

Tsudik proposed in [22] a tag authentication system called Yet Another Trivial Authentication Protocol (YA-TRAP) that does not allow tracking by using a keyed hash and monotonically increasing timestamps. This approach does not require on-demand computation for the server because the hash-table has been pre-computed to verify tags at a later time. Under this protocol the reader sends a timestamp to the tag that is compared to the previous value of the timestamp and bounded by a maximum allowable value. Based on the result of the comparison, the tag either sends the hashed message authentication code (MAC) timestamp or a pseudo-random number to the reader thereby allowing responses to be indistinguishable. The reader then forwards the information to the server where the tag is identified in a hash look-up table. [22] also points out the fact this protocol is subject to a DoS (Denial of Service) attack in which the locally stored values of the timestamps of the tag and server are desynchronized.

Having shown the Dos attack vulnerability of the protocol proposed by [22], [3] proposed an extension to YA-TRAP. This extension takes care of the Dos attack vulnerability of the original protocol. YA-TRAP is a 1-pass protocol because of the fact that the timestamp is broadcast (not unicast) by the reader to all tags in range. The extension proposed for YA-TRAP is also (essentially) a 1-pass authentication protocol with an optional pass. In the first step, the tag is authenticated and the server authenticates the timestamp in the second step. To solve the Dos attack vulnerability, the tag sends a keyed hash string instead of a pseudo-random string when the timestamp it gets is out of its bounds. This will normally save the tag from de-synchronization, but now the server must work harder because if the hash value is not in the look-up table, then the server must search exhaustively for the key. This optional second pass happens with tags the adversary has tried to kill.

[13] pointed out that cloning and counterfeiting attacks are applied on tags with Electronic product codes (EPC). He proposed a scheme the ensured the tags were unclonable. But, [7] presented some weaknesses related to privacy and information leakage in [13].

In [14], the authors suggested a security protocol using only exclusive OR (XOR) and matrix operations. This protocol was shown by [5] to be vulnerable to replay attacks and the tags can easily be tracked.

[4] then proposed a mutual authentication protocol under the EPC class 1 Generation 1 standard. They used XOR, CRC and Pseudo-Random Number Generator (PRNG) in the protocol. However, [18] showed that the scheme [4] proposed has some weaknesses. The protocol is vulnerable to tag and reader impersonation and de-synchronization attack. It was also shown that forward security was not guaranteed by the scheme. [9] also showed that a desynchronization attack was possible on the scheme [4] proposed. The attacks were mainly based on weak secure properties of CRC.

A lightweight RFID mutual authentication protocol known as Simplest-Lightweight Authentication Protocol (SLAP) was presented in [8]. The protocol is resistant to known attacks and does not demand complex computational processes. This protocol is considered as the most secure RFID authentication protocol for lightweight environment. However demonstrated in [1] that server impersonation attacks can be carried out on SLAP. [1] then proposed a revised SLAP which is almost the

same as the original work by simply reordering the content of the authentication message in the original SLAP.

Even with this improvement of SLAP, the protocol still shows partial of the secret to help the server know the full secret of the tag. This can pave way for an attacker to deceive the tag to show parts of its secret [8]. Even though this is hard to achieve, knowing part of the secret could pave way for other threats such as pattern analysis and brute-force to reveal the complete secret. As a result, a mutual authentication protocol called Provably Lightweight RFID Mutual Authentication Protocol (PLAP) was proposed in [2] to cater for the weakness of the SLAP and the revised SLAP protocol. PLAP is however more computationally expensive than the SLAP protocol.

[25] went on to show the weaknesses of CRC due to its linear properties. The authors then presented a mutual authentication protocol which made sole use of PRNG function instead of the CRC function.

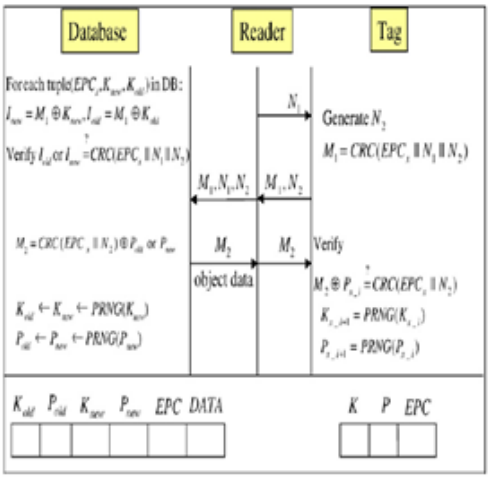

Figure 2: Proposed protocol (Chien & Chen, 2006)

Taking into consideration of the weaknesses of these protocols, a new protocol is proposed which improves on the protocol in [4] because this protocol is especially suitable for low-cost RFID. The new protocol uses PRNG instead of CRC in most of the computations and it ensures user privacy while also preventing Denial of service attacks.

III. PROPOSED PROTOCOL

After carefully studying Chien and Chen's protocol, several weaknesses were discovered in it. The discovered weaknesses are:

a) The heavy use of CRC affects the protocol adversely. As a result of its linearity, sensitive values can be gotten by performing an Exclusive OR operation on the two separate results that have a common variable [4] This therefore, does not guarantee the non – impersonation of the tags.

b) Even though [4] asserts that the proposed protocol can resist DoS attack, if the message M_2 in figure 2 is missing or tampered with up to two times, the database will not have any matching old authentication key and access key. Therefore the tag could be de-synchronized from the server.

c) [4] also does not guarantee location privacy of the tags. By XORing two messages (M_1 and M^i_1) that were not authenticated, an attacker is able to get a constant value which can be used for tracking. More details is show in [25]

These issues prompted the improvements proposed in this paper to the Chien and Chen's model.

Our proposed protocol has two phases. The initialization phase and the authentication phase. The notations used for the protocol descriptions are listed below:

Table 1: Notations and descriptions used in protocol

Notation	*Description*
C_{old}	The old database index kept in the database
C_{new}	The new database index kept in the database
K_{old}	The old authentication key kept in the database
K_{new}	The new authentication key kept in the database
P_{old}	The old access key stored in the database
P_{new}	The new access key stored in the database
$A \longrightarrow B$	A forward a message to B
$PRNG(A)$	To produce Pseudo Random Number based on the value A.
RID	Reader identification Number
$h_K(.)$	Keyed hash function
$A \oplus B$	Message A is XORed with message B
$H(.)$	Hash function.
N_A	The random number generated by device A
X	The value kept as either new or old to show which key in the record of the database is found matched with the one of the tag
C_i	Database index stored in the tag to find the corresponding of the tag in the database.
P_i	The access key stored in the tag for the tag to authenticate the database at the $(i + 1)th$ authentication phase

K_i	The authentication key stored in the tag for the database to authenticate the tag at the $(i + 1)th$ authentication phase
DATA	The corresponding record for the tag kept in the database
EPC	Electronic product code, each Tag has the unique EPC number

A. Initialization Phase

During the initialization phase, the server selects an initial authentication key K_0, an initial access key P_0, and initial database index C_0 randomly for each tag denoted as Tag_x. These values along with the Electronic Product Code of the tag, EPC_x are stored in the tag.

The server also maintains a record of the following values in its database. The values are:

1. EPC_x: Electronic Product Code
2. K_{old}: This symbolizes the old authentication key for the tag and it is initially set to K_0
3. K_{new}: This symbolizes the new authentication key for the tag and it is initially set to K_0
4. P_{old}: This symbolizes the old access key for the tag and it is initially set to P_0
5. P_{new}: This symbolizes the new access key for the tag and it is initially set to P_0
6. C_{old}: This symbolizes the old database index for the tag and it is initially set to C_0
7. C_{new}: This symbolizes the new database index for the tag and it is initially set to C_0
8. RID: Reader Identification code
9. DATA: This symbolizes all other information about the tag.

B. Authentication Phase

The authentication phase is an adaptation and an improvement over the [4] protocol. The steps of the authentication phase are explained below:

Step 1: The reader generates a random nonce N_R and forwards it to the tag.

Step 2: Once the tag receives the random number N_R, it also generates random number N_T, calculates the response values:

$$M_1 = PRNG (K_1 \oplus N_R) \oplus PRNG (N_T) \quad (1)$$

$$D = N_T \oplus K_1 \quad (2)$$

If the flag = 0, the tag uses the stored value of C_i else, it generates another random number N_{TS} and computes:

$$C_i = C_0 \oplus N_{TS}. \quad (3)$$

The tag then forwards M_1, D, and C_i back to the reader.

Step 3: The reader then receives M_1, D, C_i. It also computes

$$v = H (N_R \oplus RID_0 \oplus M_1) \quad (4)$$

and sends it alongside the values it received from the tag to the backend server together with the NR initially generated in Step 1.

Step 4: Upon getting the incoming authentication request, the backend server performs the following actions

It searches its table of readers for one whose RID_i value satisfies v == H ($N_R \oplus RID_i \oplus M_1$). It checks for both values in RID_{new} and RID_{old}. Once such value is found, the reader is authenticated by the backend server.

The back-end server checks its lookup table for the value of C_i. As soon as it finds C_i, from the tuple where the value of C_i is found, the backend server computes:

$$I_{new} = D \oplus K_{new} \quad (5)$$

$$I_{old} = D \oplus K_{old} \quad (6)$$

The backend server then determines whether M_1 matches PRNG ($K_{new} \oplus N_R$) $\oplus$ PRNG (I_{new}). If it does, x is set as 'new'. If it doesn't, the backend server tries to match M_1 to PRNG ($K_{old} \oplus N_R$) $\oplus$ PRNG (I_{old}). If it does, x is set as 'old'.

If after searching using C_i and no match is found, the backend server computes Eq. (5) and (6) iteratively for each tuple:

It then tries to match M_1 to PRNG ($K_{new} \oplus N_R$) $\oplus$ PRNG (I_{new}) or PRNG ($K_{old} \oplus N_R$) $\oplus$ PRNG (I_{old}). X is set to 'new' or 'old' depending on the match.

However if at this point no match is found, the communication is terminated.

Step 5: After getting a match for the tag, the backend server computes and sends the following to the reader.

$$M_2 = PRNG (EPC \oplus N_T) \oplus P_x \quad (7)$$

$$Info = DATA \oplus RID_i \oplus hRID(N_R) \quad (8)$$

$$MAC = H (DATA \oplus N_R) \quad (9)$$

If during the initial authentication of the reader, $RID_i = RID_{new}$, the backend server uses RID_{new} to compute info and MAC or else, it uses RID_{old}. It then sets the following if x = 'new'

$$C_{old} \leftarrow C_{new} \leftarrow PRNG (N_T \oplus N_R) \quad (10)$$

$$K_{old} \leftarrow K_{new} \leftarrow PRNG (K_{new}) \quad (11)$$

$$P_{old} \leftarrow P_{new} \leftarrow PRNG (P_{new}) \quad (12)$$

However, if x = old it computes:

$$C_{new} \leftarrow PRNG (N_T \oplus N_R) \quad (13)$$

The same process applies to the reader secret.

If $RID_i = RID_{new}$ then: Otherwise, the secret value of the reader is not updated.

$$RID_{old} \leftarrow RID_{new} \leftarrow H(RID_{new}) \quad (14)$$

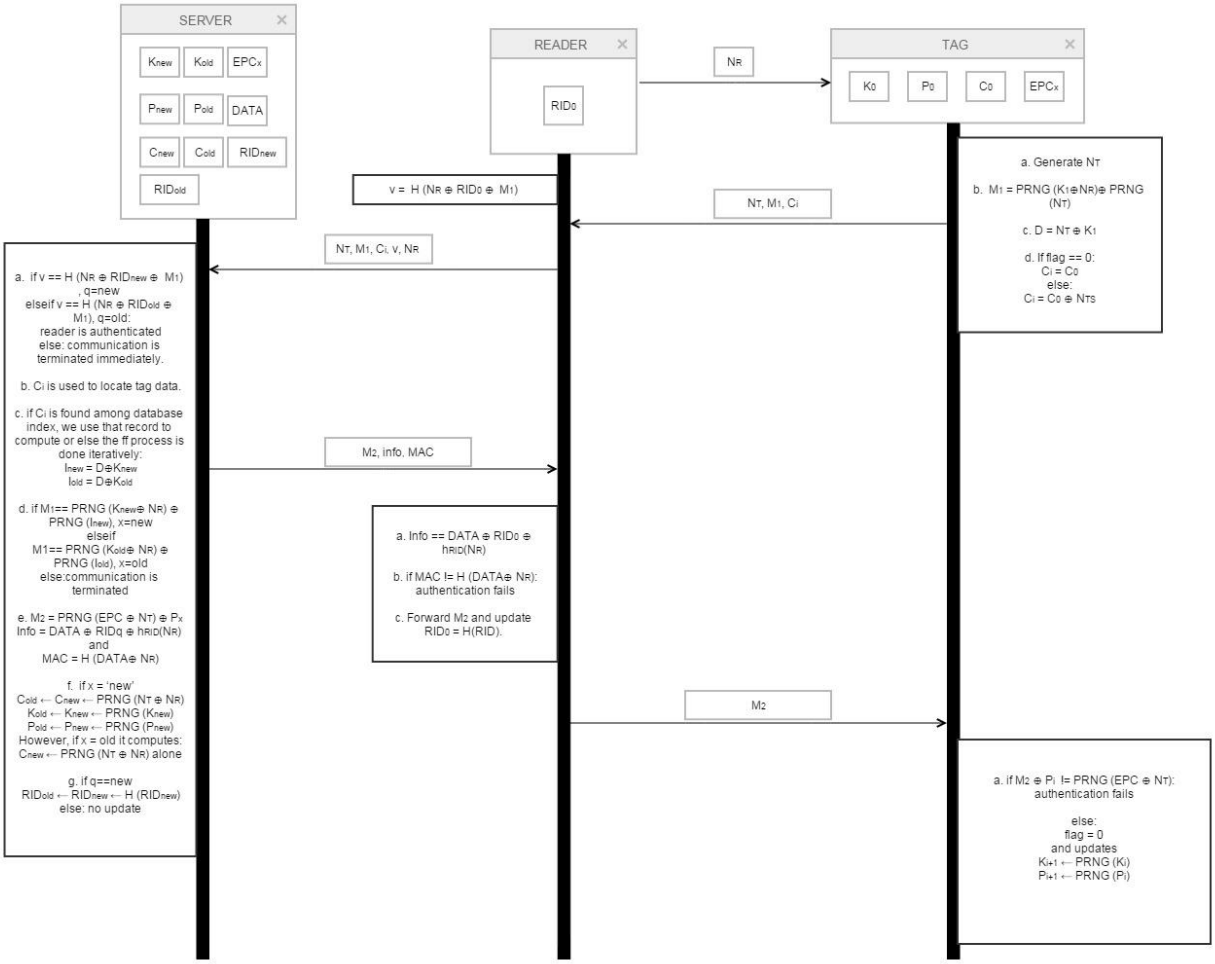

Figure 3: Proposed protocol Sequence diagram

Step 6: The reader retrieves the DATA from info using its RID and hRID(N_R). It checks whether the received MAC is equal to H (DATA $\oplus$ N_R). Once the MAC is verified, it forwards M_2 alone to the tag and updates the secret value RID = H(RID). The tag then does a XOR operation on M_2 and its own P_i and checks if the value $M_2 \oplus P_i$ and the computed value PRNG (EPC $\oplus N_T$) are same or not. If both values are different, authentication fails. Else, the tag updates the record kept in it by computing

$$K_{i+1} \leftarrow \text{PRNG}(K_i) \quad (15)$$

$$P_{i+1} \leftarrow \text{PRNG}(P_i) \quad (16)$$

And resets the flag to '0'

C. *Distinguishing features of the new protocol*

The distinguishing features of the new protocol are:

a. In contrast to [4] which updates the values on the server with every run, the new protocol does not allow the values on the server to be updated if the value of x = old which shows that the server and the tag are not in synchronization. The records are only updated when x = new.

b. The new protocol also authenticates the reader to ensure that the sensitive information – DATA is not accessible by malicious readers which is not obtainable [4] protocol.

c. The PRNG function is used to secure the sent values instead of CRC because of its bad properties

d. The random number generated by the tag is not sent in plain view as [4] sends it.

e. The new protocol is more efficient than [4] because a database reference is usually sent from the tag. This location is first checked before if searches through the other locations if the values are not in sync with the values found in that location.

IV. PROTOCOL IMPLEMENTATION

The new protocol was implemented in CASPER and FDR. These are tools used for the formal verification and simulation of the protocol to check for the reliability of the protocol and also to discover any likely attacks on the protocol. The protocol is compiled in CASPER and the result is shown below:

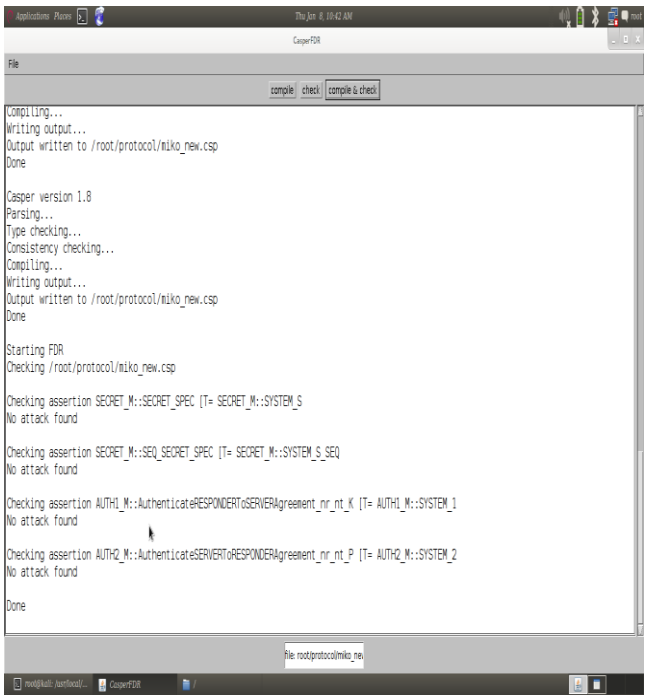

Figure 4: Result of the new protocol implemented in CASPER

A. *Performance evaluation and security analysis*

Upon verification of the protocol and its simulation, no attack was discovered. This section analyzes and explains the security features of the new protocol based on the following criterions.

1) Data Security: The new protocol provides protection for the data that is exchanged between the reader and the tag. This is achieved in our protocol by transmitting only bit-scrambled (XORed) or transformed (PRNG-function generated) data message such as M_1, D, M_2. The secret values kept in the tag cannot be revealed to the adversary. Even though N_R is sent in plain text, it is a random generated one-time-valid number and performs the computation to give M_1

2) Mutual Authentication: In the new protocol, the EPC_x is never sent in plain text. The attacker will not be able to derive its identity from the data messages transmitted. This means the parties that are communicating can be sure that the other party is genuine as an adversary cannot gain knowledge of the EPC_x during the transaction.

3) Tag Anonymity: The fact that we are using two random numbers (N_T, N_R) and the PRNG and XOR function to transform / scramble messages from the tag to the reader makes the messages indistinguishable from a spectator perspective. As there is no confidential data sent in plain text, the tag remains anonymous before a third-party.

4) Prevention of Replay Attack: Replay attack is an attack in which the adversary tries to insert an old message into a

new round of sessions. Since the adversary does not know the random numbers that will be generated ahead of time, the attack will be unsuccessful. The keys are also updated in the tag, reader and server after each successful authentication.

5) Prevention of DoS Attack: The database in [5] protocol keeps the old values of the tag's authentication key and access key (K_{old} and P_{old}) only once, then they are renewed. However, if M_2 is tampered with or missing up to twice, there will be no matching old authentication key and access key to complete the mutual authentication. So [5] protocol is not fully resistant to DoS attacks. In our scheme, we design a way of checking. If the matching record in the dataset is found by matching the old secret, it means an asynchronous update has occurred. Therefore the values of the keys (K_{new}, K_{old}, P_{new}, P_{old}) will be kept the same instead of being updated by new values.

6) Forward security: In our protocol, after every successful access, the keys stored in the tag are updated using the PRNG function. The adversary cannot use the key which is being used to recover the previous key. There is no way to trace the past communications between the Tag and the database even though the tag is compromised.

7) Prevention of tracking attack: Different random numbers N_T and N_R to compute the transmitted messages M_1 and D for every access. Therefore the attacker cannot easily trace a specific tag because there are no consistent clues revealed in each tag's response. Moreover we use PRNG function instead of the CRC function. PRNG has properties such that the adversary cannot use M_1 and M_2 to calculate a fixed number to trace the tag.

B. Comparison with previous protocol

The table below shows the result of the formal verification and simulation of our protocol and the [4] protocol

Table 2: Comparison with previous protocol

	New Protocol	(Chien, 2007)
Denial of Service	Y	N
Tracking	Y	N
Resistant to replay attacks	Y	N
Anonymity	Y	N

From the table above, we compare the new protocol proposed with the [4] protocol. It is discovered that while [4] does not protect against Denial of Service attacks (DoS), the new protocol does. In addition, the new protocol ensures privacy of the tags and it cannot be tracked. This is especially useful if the tags are used for authentication of personnel. The personnel details stored in the tag are safe and secure from all attacks except physical tampering. Finally, the new protocol is resistant to replay attack.

CONCLUSION

In conclusion, the new protocol presented by this research work is reliable and can be deployed widely for use with low cost RFID tags. The protocol has been well tested and it will aid the wide deployment of low cost RFID systems for use in day-to-day activities such as authentication of personnel, tracking of goods from warehouses to supermarkets and checking out of goods in the supermarkets. RFID system has a wide range of use and it can be deployed for even other purposes not mentioned above. The major fear about security (i.e. privacy) has been handled by the new protocol that we proposed. Further studies should be carried out on the communication between the reader and the server and how to make it secure. In addition, this new protocol does not ensure security against physical attacks. Once the tag is tampered with, there is no guarantee of security again. Therefore, more research has to be done on ensuring security even if the tag is tampered with physically.

REFERENCES

[1] Akgün M., Caglayan M. U., (2010). Server Impersonation Attacks and Revisions to SLAP, RFID Lightweight Mutual Authentication Protocol. *International Conference on Systems and Networks Communication*, pp.148-153.

[2] Alakrut R. H. E., Samsudin A. and Syafalni A. (2013). Provably Lightweight RFID Mutual Authentication Protocol. *International Journal of Security and Its Applications.* Vol. 7, No. 4.

[3] Chatmon C., van Le T., and Burmester M. (2006). Secure anonymous RFID authentication protocols. *Florida State University, Department of Computer Science, Tech. Rep.*

[4] Chien H.Y., Chen C.H. (2006). Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards. *Elsevier Computer Standards & Interfaces,29 (2007) 254–259*

[5] Chien H.Y. (2007). SASI: "A new ultralightweight rfid authentication protocol providing strong authentication and strong integrity". *IEEE Transactions on Dependable and Secure computing,* 4(4), 337–340.

[6] Dimitriou T. (2005). A lightweight RFID protocol to protect against traceability and cloning attacks, *SecureComm,* 59-66.

[7] Duc D.N., Park J., Lee H., and Kwangjo K. (2006). Enhancing security of epcglobal Gen-2 RFID tag against traceability and cloning, *In Proc. of Symposium on Cryptography and Information Security*.

[8] Godor G., Antal M., Imre S. (2008). Mutual Authentication Protocol for Low Computational Capacity RFID Systems. *In Proceedings of IEEE Global Telecommunications Conference*. pp. 1-5.

[9] Han D., Kwon D. (2009). Vulnerability of an RFID authentication protocol conforming to EPC Class-1Generation-2 Standards. *Computer Standards & Interfaces, Elsevier Science Publishers*, 31, 648–652

[10] Henrici D. and Muller P.(2004). Hashed-based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers. *PerSec '04*.

[11] Homeland Security. (2014, August 14). *Radio Frequency Identification (RFID): What is it?* Retrieved from Homeland Security website: http:// www.dhs.gov

[12] Intermec Technologies Corporation. (2007). *ABCs of RFID: Understanding and using radio frequency identification.* Washington, U.S.A: Author

[13] Juels A., Weis S.A. (2007). Defining strong privacy for RFID, *In Proceedings of PerCom'07*, 342–347.

[14] Karthikeyan S., Nesterenko M., (2005). RFID security without extensive cryptography. *In Proc. of SASN '05.* , ACM 63–67

[15] Lee Y. K. and Verbauwhede I. (2005). Secure and Low-cost RFID Authentication Protocols.

[16] Molnar D. and Wagner D. (2004). Privacy and security in library RFID: Issues, practices, and architectures, *CCS*

[17] Ohkubo M., Suzuki K., Kinoshita S. (2003). Cryptographic approach to "privacy-friendly tags", *MIT RFID Privacy Workshop.*

[18] Peris-Lopez P., Li T., Lim T.L., Hernandez-Castro J.C., Estevez-Tapiador J.M., and Ribagorda A., (2008). Cryptanalysis of a novel authentication protocol conforming to EPC-C-1 G-2 standard. *Computer Standards & Interfaces, Elsevier Science Publishers*. doi:10.1016/j.csi.2008.05.012.

[19] RFID Journal. (2014). RFID Journal. Retrieved from http://www.rfidjournal.com/

[20] Rieback M. , Crispo B., and Tanenbaum A.(2006). Is Your Cat Infected with a Computer Virus? *IEEE Pervasive Computing*

[21] Rieback M. , Crispo B., and Tanenbaum A. (2007). The evolution of RFID security, *IEEE Pervasive Computing*

[22] Tsudik G. (2006). YA-TRAP: Yet another trivial RFID authentication protocol. *International Conference on Pervasive Computing and Communications*.

[23] Tutorialspoint (2014). *Data Communication / Computer Networks Overview* retrieved from http://www.tutorialspoint.com/data_communication_computer_network

[24] Weis S., Sarma S., Rivest R., and Engels D. (2003). Security and privacy aspects of low cost radio frequency identification systems, *SPC*

[25] Zhang J., Wang W., Ma J., Li X. (2012). A Novel Authentication Protocol suitable to EPC Class 1 Generation 2 RFID system.