ISSN No. 0976-5697

Volume 3, No. 3, May-June 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A New Methodology for Preventing Vulnerabilities and Attacks on RFID Based Ultra-Lightweight SASI Protocols

Mahmoud Oussama*
Electrical and Computer Engineering Department
Benha University
Shoubra, Cairo Egypt
mahmoud.osama.ouf@gmail.com

Alaa Eldeen Sayed Ahmed
Electrical and Computer Engineering Department
Benha University
Shoubra, Cairo Egypt
alaaeldeen.sayed@feng.bu.edu.eg

Raafat Elkammar
Electrical and Computer Engineering Department
Benha University
Shoubra, Cairo Egypt
rafat.alkmaar@feng.bu.edu.eg

Abstract:In this work we present a new methodology for preventing de-synchronization attacks over the SASI ultra-lightweight authentication protocol. The methodology is based on securing the communication channel between the tag and the reader. We modify in the original protocol authentication phase so that minimizing the probability of fooling out the reader with invalid tags. One of the benefits of this methodology is keeping the communication cost for the original SASI protocol as is while achieving better secured solution. The proposed methodology doesn't affect the cost of messaging exchange between the tag and the reader. However, there is slight increasing in storage overhead. We also present results that show the effectiveness of the modified protocol on the overall all RFID system performance

Keywords:RFID, synchronization, authentication, passive tags, ultra lightweight

I. INTRODAUCTION

Radio frequency identification (RFID) is a technology used for remotely store and retrieve data using devices called RFID tags. An RFID tag is a small microchip designed for wireless data transmission. It is generally attached to an antenna in a package that resembles an ordinary adhesive sticker. The microchip itself can be as small as a grain of sand, some 0.4 mm2 [1]. An RFID tag transmits data over the air in response to interrogation by an RFID reader [2].

In general, RFID tags is classified into passive, semi passive and active tags. The first one is characterized by having no on-board power source; they drive their transmission power from the signal of an interrogating reader. Passive tags can operate in low-frequency, high-frequency bands or ultra-high frequency bands. When RFID tags include batteries in their design it can be classified either semi-passive or active tags. The main function of the batteries is to power their transmission. Active tags can initiate communication and have read ranges of 100m or more.

As an application, RFID systems can gather any amount of data on the tagged objects [3]. The collected data could be stored on a back end server to be searched or accessed by either a private or a public group of people. Stock management, traffic management, ID cards and tracking are possible application for RFID systems. One of the challenges in RFID systems is securing the data. Securing data may occur using some kind of authentication protocols. The main goal of RFID authentication protocols is to enable the tag to communicate with the reader in such as way the reader can be convinced that the tag represent a valid reading and not a kind of hacking or attacking process by an intruder or adversary in the middle of the Tag-Reader communication process. As RFID applications are expected to grow widely, researchers have proposed a various protocols that deal with securing the communication channels between the reader and the tags. A robust authentication protocol plays a very important role in the RFID system [4]. Unfortunately Most of the existing protocols are vulnerable to different attacks.

Generally speaking, attacks can be classified to the following types: Disclosure attack, Replay attack, Man in the middle attack and De-synchronization attack. In Disclosure attack, the attacker can slightly modify the challenge from the reader and then infer partial information from the response from the tag. In Replay attacks, the attacker pretend as a valid tag and replay the message to the reader Or the attacker pretend as a valid reader and intercept the data in one session and then replay an old message from the reader to the tag. In Man in the middle, the attacker makes independent connections with the tag and reader and replays messages between them. Finally, in the De-synchronization attack the attacker modifies the shared data between the reader and the tag to make them out of synchronization without being noticed. In this paper we modify in the Strong Authentication and Strong Integrity (SASI) protocol so that overcome its vulnerabilities to the de-synchronization attacks.Also, we implement the new protocol to show that our proposed solution provide a secure RFID environment

without affecting the original SASI protocol communication cost and with a slight increase in the storage cost.

The paper is organized as follows: in Section 2, the related research work is introduced, Review of SASI Protocol and its vulnerable attacksis described in section 3. Our proposed ultra-lightweight Anti-desynchronization RFID Identification methodology is proposed in Section 4. In Section 5, implementation and experiments results are introduced. Finally, section 5 introduces the conclusion.

II. RELATED WORK

The main goal of RFID authentication protocols is to enable the tag to communicate with the reader in such as way the reader can be convinced that the tag represent a valid reading and not a kind of hacking or attacking process by an intruder or adversary in the middle of the Tag-Reader communication process. Chein[5] classified protocols used in securing RFID systems into four classes. The classification is based on the way of handling and securing the transferred data. The first class is called "full-fledged class" that use cryptographic function or public key algorithms to provide mutual authentication between the reader and the tag. The second class is called "simple class" that use a random number generator and one way hashing function on each tag to secure the data. The third class is called "lightweight class " that use different securing technique based on using a cyclic redundancy code (CRC) check sum but not using a hash function.

The fourth class is called "Ultra-lightweight" that only use simple operations such as XOR, AND,OR to secure the transferred data. In [6] lopez et al. proposed a family of Ultra-lightweight Authentication protocols M2AP (minimalist mutual authentication protocols) which arte followed by EMAP (an efficient mutual authentication protocols) and LAMP (A Real Lightweight mutual authentication protocols). On all those protocols, tags involve only simple bit-wise operations like XOR, AND, OR, and addition mod 2m. Li and Wang [7] and Li and Deng [8][9][10], respectively, reported the de-synchronization attack, privacy & anonymity, mutual authentication and the full-disclosure attack on these protocols, and Chien and Hwang [11] further pointed out the weakness of Li-Wang's improved scheme. Hung-Yu Chien proposed a new Ultra-lightweight RFID authentication protocol (SASI) that provides strong authentication and strong integrity protection of its transmission and of updated data. The protocol requires only simple bit-wise operations on the tag and can resist all the possible attacks. It was designed to resist de-synchronization attack, replay attack, and man-in-the-middle attack. In [12] Hung-Min Sun, Wei- Chin Ting, and King-Hang Wang found two de-synchronization attacks to break the protocol

III. REVIEW OF SASI PROTOCOL AND ITS VULNERABLE ATTACKS

In this section we review SASI protocol. SASI [5][11] is an protocol that provides strong authentication and strong integrity protection of its transmission and of updated data.

The protocol requires only simple bit-wise operations on the tag and can resist most possible attacks. In the protocol, it is assumed that the reader and the database share a secure channel, but the channel between the reader and the tag is insecure. The tag is initialized with a static identification (ID), a pseudonym (IDS) which is used as the search index in the database, and two secret keys K1 and K2. The length of each variable is 96 bits. Each tag keeps two entries of the form (*IDS*, *K*1, *K*2), one for the old IDS_{old} value and the other for the potential or the next IDS_{next} value. The protocol includes three phases:

a. Tag identification phase
b. Mutual authentication phase
c. Pseudonym updating and key updating phase

In the first phase the reader try to identify the tag by sending hello message to the tag and the tag then responded with a potential IDS_{next} to the reader then the reader search for IDS_{next} in the backend database. If it is not found the reader request the IDS_{old}.

In the second phase the reader using the received IDS acquires the private information that is recorded in the database and access the corresponding secret information ID, k1, k2 for the tag. The reader generates two random numbers n1, n2. The reader use a combination of simple bit-wise operations such as XOR,AND, OR, etc to calculate the term A||B||C then send it to the tag. The tag extracts n1 and n2 from A and B and computes C which compared the one received from the reader.If the two values are matched, the reader is authenticated. Finally the tag use simple bit-wise operations to compute D which in turn sent to the reader.The reader matches the received D value with a computed local version. If it matched the tag is authenticated.

In the third phase, after successfully completing the mutual authentication phase between reader and tag, the reader locally update IDS, K1 and K2.The protocol was designed to resist de-synchronizationattack, replay attack, and man-in-themiddleattack.

A. Attacks on SASI:

Hung-Min et al [12], introduced two de-synchronization attacks or security vulnerabilities that break down the SASI protocol.

In The first attack tag de-synchronization processes is performed in these steps:

a. The attacker object watch and record the messages exchange occurred between the tag and the reader. at the end of the protocol, the attacker interrupt the message D
b. Communication in such a ways so that final messages update is done for tags but not for the reader. This will cause unmatched data stored in the reader's database.
c. The attacker let the reader and tag to communicate again without any interruption.
d. After the reader leaves the reading range, the attacker initiates the connection with the tag and attacks it, and now the reader and the tag in out of synchronization.

In The second attack tag de-synchronization processes is performed in two steps:

e. Let the reader and the tag to communicate with each other without any interruption and the attacker will eavesdrop to the message exchange between them.
f. The attacker forge a tuple in the message that recorded and send it to the tag and try many time until coincidentally the tag will accept it and replay to the attacker and now the tag and the valid reader is out of synchronization.

IV. PROPOSED METHODOLOGY

In this section we present a methodology to overcome the security vulnerabilities or attacks in SASI [13], we modify in the authentication phase by changing in the communication scenario between the tag and the reader so that the attacker is fooled out. According to SASI, each tag has several parameters such as a static identification (ID), pre-shares a pseudonym (IDS), two keys (K1, K2) which are used on the authentication process. To overcome the two possible de-synchronization attacks we used two scenarios [12][14].

In the first scenario, both the tag and the reader keep and store two entries of (IDS, K1, and K2). The first entry is related to the potential value of IDS and the second value is randomly selected from a range of the old values of IDS's. This range is selected by the user. As far as the range is increasing the authentication becomes stronger but the required storage capability become higher. $IDS_{rnd(x)}$ represents a randomly picked IDS value that has been previously computed during the last x matches occurred during the communication between tag and the reader

IDS_{Next}	Next Potential IDS
$IDS_{rnd(x)}$	old IDS selected randomly from the last x values

In the second scenario, both the tag and the reader keep and store three entries of (IDS, K1, and K2). The first entry is related to the potential value of the Tag IDS and the second value is used in synchronization process in case the first IDS mismatched. The third entry is used to authenticate the tag to start the updating process by matching the $IDS_{rnd(x)}$ value sent by the reader with IDS values stored on the tag during the last x IDS updates.

IDS_{Next}	Next Potential IDS
IDS_{old}	Most recent Old IDS
$IDS_{rnd(x)}$	old IDS selected randomly from the last x values

Now, let us show how these two scenarios are employed to prevent the two previously mentioned synchronization attacks on SASI.

A. Preventing The First Attack at SASI:

Here the attacker permanently disables the authentication capability of a RFID tag by destroying synchronization between the tag and the RFID reader. First the attacker allows the reader and the tag to run the protocol without intervention to and record the Final computed IDS value. SASI protocol is based on using two value of IDS (next and old) to make the authentication where if IDS_{next} is not found IDS_{old} would be the alternative. The attacker would now have IDS_{old}, so it imitates as a valid reader and query the tag. The tag will reply IDS_{next}"Fig. 1".

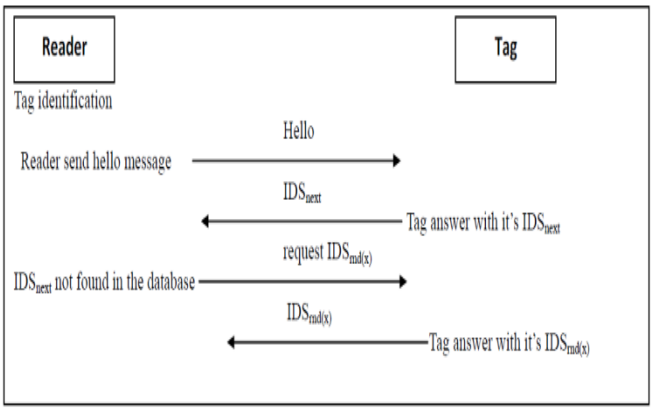

Figure: 1Scenario 1 description

The attacker pretends that he cannot find IDS_{next} and requests the old IDS. The tag response with the IDS_{old} value which the attacker knows. The tag will treat the attacker as a valid reader and the attack is done. To prevent this attack, we assume that there are two entries of (IDS, K1, K2)as follows:

a. $(IDS_{next},K1_{next},K2_{next})$
b. $(IDS_{rnd(x)},K1_{rnd(x)},K2_{rnd(x))}$

Where, the first one represents the next potential value which is used in normal operation. The second one replaces the most recent old value of IDS with a randomly selected from the last x computed values of the IDS. Now we are mislead the attacker where he doesn't know which old value of IDS could be used to communicate with the tag. Now, the attackers (the invalid reader) and the tag are desynchronized since the values stored in both entities don't match.

B. Preventing the second attack at SASI:

Here the attacker eavesdrops on successful session between the tag and the reader and recorded the components that leads to compute IDS's. When valid readers leave the reading range of the tag the attacker initiates the protocoland the tag response with $IDS_{next.}$ This value is not known to the attacker, so the attacker will ask for IDS_{old}. The tag responds with IDS_{old}. Based on the value of the IDS_{old}, the attacker will extract the IDS forming components and change them by flipping the most significant bit of that component then re-compute IDS and send to the tag. The tag will consider this value as IDS_{next} and execute the update phase.In the next time, when the reader tries to read the tag, it will be rejected by the tag, since the IDS components stored in the tag is no longer synchronized with the database. This makes them de-synchronized.

To prevent this attack, we assume that there are three entries of (IDS, K1, K2) are as follows:

a. $(IDS_{next},K1_{next},K2_{next})$
b. $(IDS_{old},K1_{old},K2_{old})$
c. $(IDS_{rnd(x)},K1_{rnd(x)},K2_{rnd(x)})$

The basic idea that the attacker depends on is to mislead the tag by making him update the IDS with values that doesn't match with the values stored in reader. By applying

the second scenario, the tag won't update the value sent by the reader unless the reader sends him the third random entries $IDS_{rnd(x)}$. The tag will validate the sent entry with the value already stored, if it does match it proceeds to the update phase else it consider the reading process as invalid. Now, the attacker will not be able to break the communication between the reader and the tags due to its ignorance about the random value of IDS used in the update authentication phase "Fig. 2".

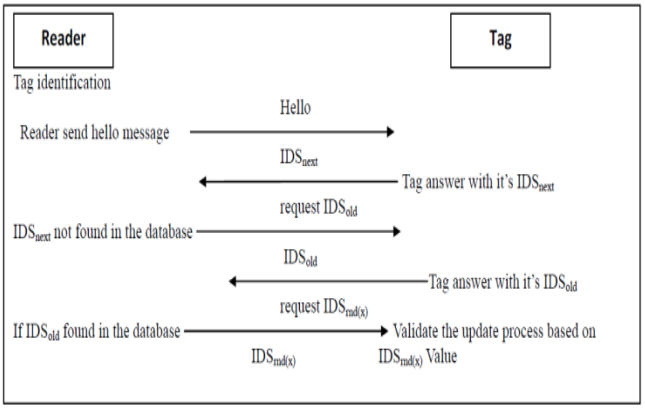

Figure: 2 Scenario 2 Description

V. IIMPLEMENTAION AND RESULTS

In this section we present the results achieved from applying our modification to SASI protocols. In the implementation we measure the rejection ratio value which reflects the ability of preventing attacks on SASI protocol. Rejection ratio indicates the percentage of the number of rejected tags after applying the attack. We followed the following steps in our implementation:

a. Implementing the original SASI protocol
b. Apply different attacks and measure its ability in preventing their effect
c. Apply our modification to SASI protocol.
d. Apply different attacks on the modified protocol to check out the enhancement occurred as result of this modification.

The first step is done by implementing the protocol, then we test the effect of attacks on SASI protocol according to the following:

a. We assumed that the group of tags is ranging from 1 to 100 tags with step equal 10.
b. At each of this group of tags we run SASI protocol,
c. Then apply the attack
d. Finally, calculate the rejection ratio.

Now we look at the results achieved to prevent different attacks using different proposed scenarios.

A. Preventing First attacks using scenario 1:

As shown in "Fig. 3", regardless the number of tags used, the rejection ratio is 100% which means that the attack is fully succeeded to deny the access to any of the valid tags

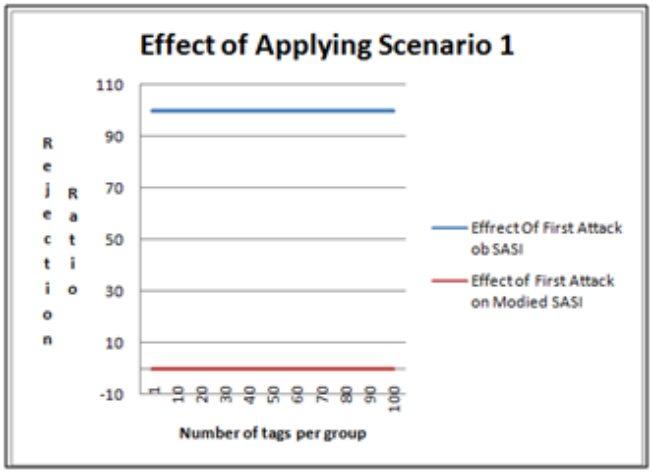

Figure: 3 Effect of Applying Scenarion 1 to prevent the first attack

Now and after applying the proposed scenarios 1 on SASI protocol the rejection ration becomes zero% for all number of groups. This indicates the success of the new methodology to prevent different attacks from breaking down through SASI protocol.

B. Preventing the first attack using scenario 2:

To test the effect of this scenario when applying the first attack we did the following method:

a. We start to apply the attack on one tag for ten times and then calculate the average rejection ratio. The result was 0%
b. Then we increase the number of tags gradually until reaching 10 tags in each trial. The result remains the same 0%.
c. We continued in increasing the number of tags per trial. We found out that the rejection ration changed to 4% when the tags group becomes 50 tags per trial.
d. We continued in increasing the number of tags until reaching 100 tags per trial. We found out that the rejection ration changed slightly and become 3%.

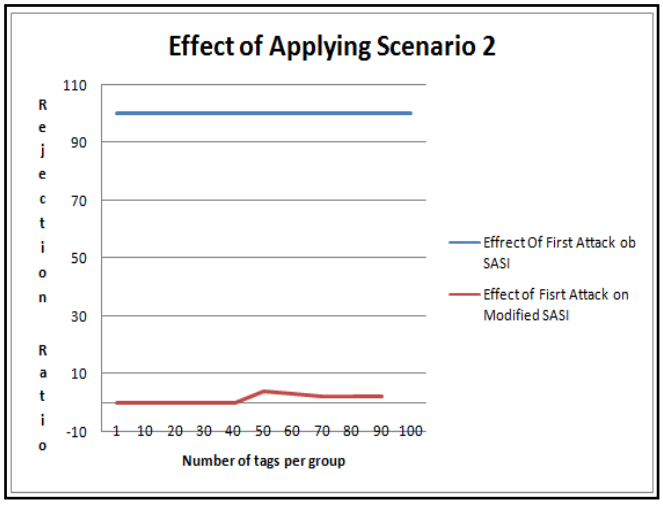

Figure: 4 Effect of Applying Scenario2 to prevent the first attack

From "Fig. 4", we conclude that the de-synchronization occurs when the number of tags per trial is greater than 50

and with a low percentage value of 3 or 4%. This means that 4% of the 50 tags are able to do the de-synchronization effect successfully. According to that scnario2 is able to countering the first attack with a 4% exception. This small percent comes from the assumption that the most recent old IDS value is included in the range that the tag may pick one of them to send it back to the reader. This will lead to give the attacker the chance to execute its attack successfully and achieve the tag reader de-synchronization.

C. Preventing the second attack using scenario 1:

Here we show the result of testing the ability of the proposed scenario1 in counteracting the second attack we implemented the proposed scenario -which we call it Modified SASI- with the following parameters:

a. It is assumed that the number of tags is ranging from 1 to 100
b. Run the modified SASI protocol for the given each group of tags.
c. Apply the first attack then calculate the rejection ratio.

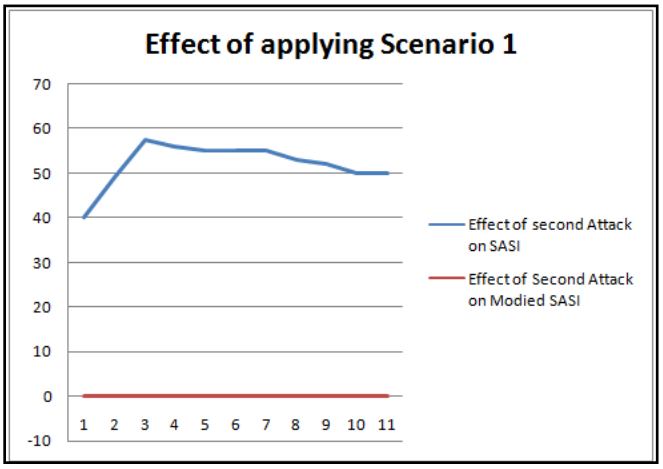

Figure: 5 Effect of Applying Scenario 1 to prevent the second attack

When applying the second attack on the original SASI protocol, the rejection ratio ranges from 49 to 57 % which also indicate that second attack is partially able to make tag reader de-synchronization. Now, when the modification is applied the rejection ration becomes 0%, which indicates that the proposed modification completely succeeded to overcome the effect of the second attack. Consequently, no de-synchronization will occur between the reader and the tag when communicating with each other "Fig. 5".

D. Preventing the second attack using scenario 2:

Here we show the result of testing the ability of the proposed scenario 2 in counteracting the second attack we repeated the same experiment but with the second attack. We found out that the rejection ratio is 0%, which indicate that scenario 2 is completely succeeded to overcome the effect of the second attack "Fig. 6".

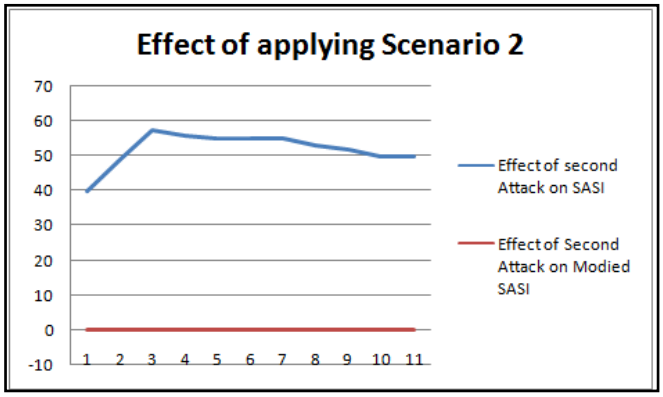

Figure: 6 Effect of Applying Scenario2 to prevent the second attack

E. Communication cost:

The communication cost is determined by the number of steps needed to exchange the authentication information. Since the communication cost of SASI is acceptable by most of the RFID environment, we tried in our modification to the keep the communication cost as the one used by SASI. However, even though the communication cost is the same, we were able to prevent the effect of different attacks that SASI was vulnerable to them. SASI protocol is proved to be more efficient in communication than other authentication protocols. Consequently, we can conclude that our modification is efficient in terms of the communication cost too.

In SASI protocol tag identification phase and mutual authentication phase are performed in four steps using four messages. In Our proposed modification to SASI, there are a little change in the tag identification phase. The change is mainly obvious when dealing when requesting next IDS. In that the Next IDS is found in the backend database the behavior of the modified protocol doesn't change and the communication will be the same as the original protocol. In case, If the next IDS is not found in the backend database, a step is added in which old IDS is requested. In some cases other values of IDS are requested such as (old IDS, random IDS). The nature of the requested IDS depends on which one of the two scenarios is used. For example in scenario 2 there will be a double check on IDS value and consequently the communication message size will increase. In SASI, The total number of bit that is sent over the communication channel between the reader and the tag is 424 bit while in some of our proposed scenarios the total number of sent bits is 520 bit. This calculation is computed based on the assumption of having 5 bytes "hello" message.

F. Storage cost:

Generally speaking, the tag only stores the information related to authentication such as different IDS values. Other information is stored in the back-end server. Thus, our proposed protocol could meet the potential storage constraints in a low cost RFID environment.To run SASI protocol, it requires storage space in each tag to store two types of data, static data and variable data. Each tag store:

a. its static identifier (ID)

b. Two records of the tuple (IDS, K1, K2) which represents the old and potential new values. Each of these records requires storage space of 96 bit in length according to EPC Global.

The ID stored in Rom because it's a static value and the remaining values (IDS,K1,K2) are stored in rewritable memory because it is updated frequently. These dynamic values need a storage space (96 x 6= 576 bits).

To run our proposed scenarios, the only change will be in the number of tuples stored in the rewritable memory (IDS, K1, K2). In this case, the number of bits needed is calculated by (96 x (3 x n)) bit where n is the number of tuples needed. For example:

a. in Scenario 1 we store 6 tuples so we need to store (96*(3*6))= 1728 bits
b. In Scenario 2 we need the same number of bits because we store also 6 tuples of variables list.

The following table shows a comparison between Different existing authentication protocols (LMAP, M2AP, EMAP)[5][6][15] that named to U-map family, SASI protocol and the 2 proposed modifications. The table shows comparison on the:

	U-map family	SASI	Modification 1	Modification 2
Resistance to de-synchronization attacks	No	No	Yes	Yes
Total message for mutual authentication	4-5L	4L	4L	4L
Memory size on tag	6L	7L	(1+(3xn))L	(1+(3xn))L
Memory size for each tag on server backend database	6L	4L	4L	4L
Operation types on tag	⊕,∨,∧,+	⊕,∨,∧ ,+,rot	⊕,∨,∧ ,+,rot	⊕,∨,∧ ,+,rot

Note: **n** is the number of saved tuple and **L** denote to bit length of 96 bits

Figure: 7 Performance comparison of ultra-lightweight authentication protocols.

a. Effect of counter acting different attacks.
b. the total number of message need to make the authentication process between the reader and the tag
c. the total memory size needed in each tag it depend on the protocol and on the modification that used,
d. the memory size need on the backend server for each tag in the database
e. ultra-lightweight authentication protocols
f. The operation type for each protocol needed.

The comparison indicates that our modification is within the scope of other of ultra-lightweight authentication protocols beside it give more resistant to different attacks.

VI. CONCLUSION

SASI protocol is an Ultra-lightweight authentication protocol that was designed to resist de-synchronization attack, replay attack, and man-in-the-middle attack. However, two de-synchronization attacks or security vulnerabilities was founds. In order to prevent the first attack, we modified the protocol and changed the criteria of selecting the old IDS values. The criteria are based on random selection. In the second attack, the attacker try to make the tag update invalid IDs value, we added a third step in the update phase so that tags perform the updates based on agreed random values previously stored on both reader and tags. We implemented and test the effect of the modification. The results indicate that our solution doesn't affect in the communication cost or the storage cost however introduce a more secure methodology that is able to resist different hacking attacks.

VII. REFERENCES

[1] K. Takaragi, M. Usami, R. Imura, R. Itsuki, and T. Satoh "An ultra small individual recognition security chip" IEEE Micro, vol 21(6), 2001, pp 43–49.

[2] Ari Juels , "RFID Security and Privacy: A Research Survey" J-SAC, vol. 24, 2006, pp. 381—395.

[3] Radomirovic, T. van Deursen " Attacks on RFID Protocols", \url{http://eprint.iacr.org/, Cryptology ePrint Archive: Report, Jul 2008.

[4] Yawer Yousuf, Vidyasagar Potdar "A Survey of RFID Authentication Protocols", 22nd International Conference on Advanced Information Networking and Applications - Workshops, IEEE Xplore ,Dec 2008, pp. 1346-1350.

[5] H.-Y.Chien,"SASI "A new ultralightweightrfid authentication protocol providing strong authentication and strong integrity" IEEE Transactions on Dependable and Secure Computing, vol. 4, no. 4, 2007, pp. 337-340.

[6] Pedro Peris-Lopez, Julio Cesar Hernandez-Castro, Juan M. E. Tapiador, and Arturo Ribagorda "Advances in Ultralightweight Cryptography for Low-cost RFID Tags: Gossamer" Protocol. In Workshop on Information Security Applications, vol 5379 of LNCS. Springer-Verlag, Jeju Island (Korea), Sept 2008, pp. 56-68.

[7] T. Li and G. Wang, "Security Analysis of Two Ultra-Lightweight RFID Authentication Protocols," Proc. 22nd IFIP TC-11 Int'l Information Security Conf., My 2007, pp. 14-16.

[8] T. Li and R.H. Deng, "Vulnerability Analysis of EMAP-An Efficient RFID Mutual Authentication Protocol," Proc. Second Int'l Conf. Availability,Reliability, and Security (AReS 07), 2007, pp. 10-13.

[9] "A SUMMARY OF RFID STANDARDS" RFID Online Journal, http://www.rfidjournal.com/article/view/1335

[10] syed ahson, Mohammad Ilyas "RFID HANDBOOK Application, Technology, Security, and privacy" CRC Press, 2008.

[11] Chien, Hung-Yu. "The Study of RFID Authentication Protocols", Development and Implementation of RFID Technology, Jan 2009, pp. 14-44, ISBN 978-3-902613-54-7

[12] Hung-Min Sun, Wei-Chih Ting, King-Hang Wang: On the Security of Chien's Ultralightweight RFID Authentication Protocol. IEEE Trans. Dependable Sec. Comput. vol 8(2), 2001, pp. 315-317.

[13] Mahmoud Oussama, Alaa Eldeen Sayed Ahmed and Raafat Elkammar "Counter Attack Methodology for Preventing Vulnerabilities or Attacks on SASI", 1st International Conference on Advanced Computing and Communications (ACC-2010), Sep 2010, pp. 22-26 .

[14] Pedro Peris-Lopez, Julio Cesar Hernandez-Castro, Juan M. Estevez-Tapiador," RFID Systems: A Survey on Security Threats.". s.l. : Lecture Notes in Computer Science, vol. 4217/2006, pp. 159-170.

[15] H.-Y. Chien and C.-W. Hung, "Security of Ultra-Lightweight RFID Authentication Protocols and Its Improvements," ACM Operating System Rev., vol. 41, no. 2, July 2007, pp. 83-86.

