UKACC International Conference on Control 2012
Cardiff, UK, 3-5 September 2012

A RFID mutual authentication protocol based on AES algorithm

Tuan Anh Pham, Mohammad S. Hasan and Hongnian Yu
Faculty of Computing, Engineering and Technology, Staffordshire University, Stafford, UK
tapham@live.com, {m.s.hasan, h.yu}@staffs.ac.uk

Abstract— **The emergence of RFID applications has huge influence to become pervasive in modern life. However the vulnerability of the transmission through the air and the unique identification number of RFID tag are the drawbacks that impact the popularity of RFID technology. In this paper, a mutual authentication protocol is proposed based on the challenge – response model. The Advanced Encryption Standard (AES) is used as a cryptographic primitive to secure the data. The experimental works are carried out to validate the protocol in term of security and privacy. The timing analysis is also presented and applied to a case study of conveyor belt system.**

Keyword: RFID, AES algorithm, mutual authentication

I. Introduction

Recently Radio Frequency Identification (RFID) systems have been becoming popular and aiming to be ubiquitously applied in many areas including library, banking, logistics, transportation, manufacturing, supply chain system, military etc. Some large corporations who have deployed this technology are Wal-Mart, Procter and Gamble, and the United State Department of Defense [1] etc.

A RFID tag can be read as long as the item is within the range of the reader without requiring the line-of-sight operation as bar code technology. However, one of the biggest difficulties to the adoption of RFID technology is the lack of security and privacy. There is little security on the RFID tags or during the communication with reader which causes the RFID system vulnerable to many types of attacks e.g. information leakage, replay, denial of service [2], [3].

Many authentication protocols have been proposed to enhance the robustness of a RFID system. Some of them were developed based on cipher algorithms such as Advanced Encryption Standard (AES) or Elliptic Curve Cryptography (ECC) [4], [5], [6], [7]. Some utilised the hash-based algorithm, pseudo random number generator, Cyclic Redundancy Check (CRC) function and/or some ExOR and rotation operations [8], [9], [10], [11], [12]. Normally, cipher-based approaches are not preferred for passive tag because of its high computational cost and the large hardware area. However, with the advances in technology, it is feasible to embed the cipher engine on the passive tags but still guarantee the low tag's cost [5].

The aim of this paper is to develop a protocol to provide a strong, high security and trustful authentication scheme which can protect against most of well-known RFID system attacks. Thus, a novel mutual authentication protocol based on AES primitives and challenge – response method is proposed. The symmetric block cipher AES-128 is utilised in this proposal because it has been standardised and proved to be secure [13].

The rest of the paper is organised in six parts from section II to section VII. Section II briefly introduces the recent works. Section III proposes the novel protocol. In section IV, the analyses of the protocol in term of security and privacy are presented. Section V provides some experimental results. And section VI introduces a case study on conveyor belt system. The last section, section VII, is the conclusion of the paper.

II. Related works

A. RFID security proposals

There are the number of researches to address RFID security and privacy issues. This paper roughly categorises them into two categories: **cipher-based protocols** which are developed on AES or ECC algorithms and **hash-based protocols and others** which are based on hash functions and/or some simple operators such as ExOR, rotation etc.

1) Cipher-based proposals

A strong authentication for RFID systems is introduced in [5] by implementing one-way encryption AES algorithm on RFID passive tag. A modified communication method between the reader and tag is also proposed in order to satisfy the strict timing requirement. However, this research reveals a possibility for the adversary to achieve the shared key of the AES encryption block [7]. In addition, the identical challenge always results in the identical cipher response which causes it to be susceptible to replay attack and tag clone. For example, assuming the challenge is the 16-bit output of Pseudo Random Number Generator [14], it is feasible for attackers to collect a database of 65536 entries to impersonate the legitimate devices to obtain the authentication.

Extending from the research of [5], [7] offers an advanced mutual authentication using the AES algorithm as a cryptographic primitive. The main issue of this protocol is easy to lose the synchronisation between the reader and the tag if the response from tag is blocked. Another security issue of this research is the man in the middle attack [6].

[4] develops a mutual authentication protocol based on Elliptic Curve Cryptography (ECC) which is an asymmetric

cryptographic algorithm. However, it is susceptible to DOS attack. The attackers can modify the counter value in the message sent from the server to the tag by a much larger value in order to deceive the tag into updating it. In next authentication, because the counter value in the tag is greater than the one the server sends, the tag terminates the authentication process right away.

2) Hash-based proposals and others

[11] proposed a mutual authentication protocol for passive tag which is based on cryptographic hash functions. A method to prevent the desynchronisation between a reader and a tag was proposed as well. This proposal however might disclose the issues of tracking, tag impersonation and the DOS attack [15].

Another hash-based two-way authentication scheme which employs Cyclic Redundancy Check (CRC) and Pseudo Random Number Generator (PRNG) function was presented in [9]. Although it was claimed to be robust, the attackers are still able to track the tag due to its identical response. Furthermore, the replay attack and tag impersonation can be carried out to compromise the valid reader. For example, the attackers possibly play on the response from the tag embedded in the cheap product and then replay that to the standard item.

[12] proposed a scheme for RFID system conforming to EPC Class 1 Generation 2 specifications. However, [16] has pointed out many security failures in this protocol. Moreover, the DOS attack can be employed to desynchronise the database server and the tag [10]. [16] has also indicated the possibility of auto-desynchronisation at backend database up to 0.93 if the population of tags is greater than 2^{18} tags.

To overcome the security issues of [12], [10] enhanced the scheme to defend against DOS attack, replay attack, forward secrecy and privacy concern. However, [17] and [8] claimed that this proposed protocol is not invulnerable due to the insufficient size of the secure keys. Consequently, it is susceptible to Tag/Reader impersonation attack and desynchronisation as [17] also claimed.

III. THE PROPOSED PROTOCOL

A. Assumptions

- A tag has the AES-128 encrypting block on-board proposed in [26].
- A reader and database server can perform AES-128 encryption and decryption.
- The channel between the reader and tag, the reader and backend database are vulnerable.

B. Initialisation phase

At the beginning, the backend database, the reader and the tag have the same shared key *K*. The seed *s* of each tag is stored on the server and on the rewriteable memory of tag. The *ID* is placed in tag's read-only memory and server as well. In addition, the server and reader keep the reader identification number ID_R for reader authentication purpose.

C. Authentication phase

The authentication phase is illustrated in Figure 1.

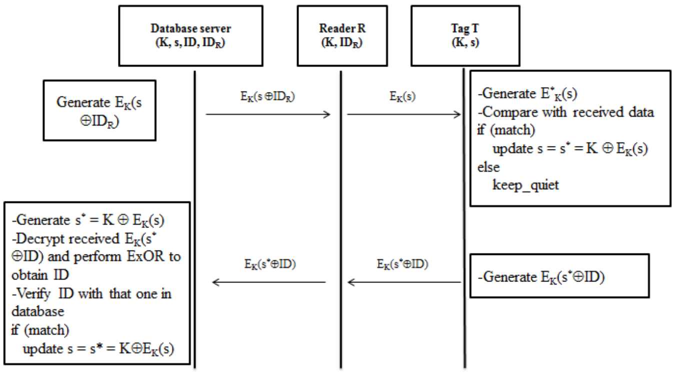

Figure 1. The proposed protocol

- Step 1: Query

Initially, the server generates $E_K(s\oplus ID_R)$ and sends to the reader. The reader will decrypt this value to obtain the seed *s* by performing $s=ID_R\oplus(ID_R\oplus s)$. Afterward, reader encrypts the seed *s* and forward to the queried tag.

- Step2: Reader-to-Tag authentication

On receiving $E_K(s)$, the tag employs the on-board AES encrypting block to produce its own $E^*_K(s)$ and then check whether $E^*_K(s)=E_K(s)$ holds. If not, the tag keeps quiet. On the contrary, tag updates the seed *s* by $s^*=K\oplus E_K(s)$ and again performs AES encrypting block to compute $E_K(s^*\oplus ID)$ in order to convey to the reader.

- Step 3: Tag-to-Reader authentication

Upon obtaining $E_K(s^*\oplus ID)$, the reader will pass it to the backend database to decrypt it. The decrypted value $s^*\oplus ID$ is used to extract the tag's *ID* by carrying out the simple operation $ID = K\oplus E_K(s\oplus(s^* \oplus ID)$. Then this *ID* is verified by comparing with the *ID* existing in database to check whether this tag is legally acceptable or not. If the mismatch occurs, the server discards any data it has received and declines the authentication of the tag. Oppositely, the server updates the seed value of reader by $s^*=K\oplus E_K(s)$ to guarantee the synchronisation of the system.

IV. SECURITY AND PRIVACY ANALYSES

A. Information leakage

Due to the insufficient protections and unreliable security levels, the data transmitted through the air are easily compromised. In this protocol, there are two messages interchanged between the reader and the tag. However, these sensitive data are encrypted by AES-128 cryptographic block. The attackers cannot get the plaintext or the raw data. So the information leakage is evitable.

B. Tag tracking and tracing

For even EPC or IEEE standard, RFID tag is designed to have a unique number which can be tracked in range of any readers. However, this protocol provides the mechanism that whenever the counterfeit reader queries the tag, the tag does not send any response back. Consequently, the tracking and tracing privacy is secure in this protocol.

C. Tag clone

In theory, there is no tag which has similar unique identifier number to another one. However, adversaries can replicate the forged tag without much effort and expense [18]. To conduct the tag clone attack, the attackers have to obtain the key *K*, seed *s* and the *ID* of the tag. However, these shared data are kept safely and privately in the backend database and inside the tag; the attackers have no information to perform the same encryption block to acquire the authentication.

D. Man-in-the-middleattack

The adversative readers can impersonate the valid one in order to intercept, change and obtain the messages going between the parties which they need. In the proposed protocol, before querying the tag, the server sends the message $E_K(s\oplus ID_R)$ to the reader. Because only valid reader has the key *K* to decrypt this cipher text, there is no possibility for attackers' readers to achieve the seed *s* in order to communicate with the tag. Thus the man-in-the-middle attack can be avoided.

E. DOS(Denial of Service) attack

It affects any wireless communication e.g. WiFi, RFID etc. by obstructing or intercepting the wireless signal that causes loss of synchronisation among the devices. Let us assume the scenario in which the response $E_K(s\oplus ID)$ is intercepted by the adversary. Because the reader has not got the response from the tag, it does not update its seed. In the next authentication, reader sends $E_K(s)$ to challenge the tag. But the seed in tag now is $s^*=K\oplus E_K(s)$, therefore, $E_K(s)=E_K(s^*)$ is not satisfied and the tag keeps quiet. So the solution is that reader will attempt with $E_K(K\oplus E_K(s))$. At this attempt, the synchronisation is re-established. Hence, the proposed protocol is insusceptible to DOS attack.

F. Replay attack

In replay attack, the adversaries stand in the middle of communication channel to duplicate the valid transmission which will be fraudulently repeated later. However, the seed *s* is automatically updated after each successful authentication session. This leads to the fact that the cipher text $E_K(s)$ changes in every authentication cycle. Thus, the attackers cannot utilise the former data in order to deceive the authorised reader or tag to overcome the authentication process.

Overall, the proposed protocol is able to protect against many types of the known attacks. The comparison with other researches is shown in Table I to have a general view of the robustness of this protocol.

V. Simulation and Validation

The simulation is executed on a computer with Intel T3200 2GHz processor, 2GB of Ram memory on Windows 7 Ultimate. The C programming language has been used to develop the simulations. The AES core is the open source program designed by Niyaz PK [19].

TABLE I. COMPARISONS IN TERM OF SECURITY AND PRIVACY

	[5]	*[7]*	*[11]*	*[9]*	*[12]*	*[10]*	**Proposed protocol**
Information leakage	Δ	✓	✓	✓	✓	✓	✓
Tag tracking and tracing	✓	✓	Δ	×	×	×	✓
Tag clone	×	✓	Δ	×	×	×	✓
Denial Of Service (DOS)/Desynchronisation	✓	×	Δ	✓	×	×	✓
Replay	×	×	✓	×	✓	✓	✓
Mutual authentication	×	✓	✓	✓	✓	✓	✓

✓: fully satisfied; ×: not satisfied; Δ: partially satisfied as assumption

A. Performance analysis

In this proposed protocol, the messages transmitted on the channel are always encrypted. However, the adversaries can play brute-force attack (BFA) to attempt to get the answer from the tag. But since the size of the message is 128 bits, it is not feasible to perform BFA entirely with all the possible cases (more than 3x10^{38}). Therefore, the simulation will randomly pick up an amount of random 128-bit input vectors (it is called sub-BFA) in order to verify the possibility of unintentional matches. The results are shown in Figure 2.

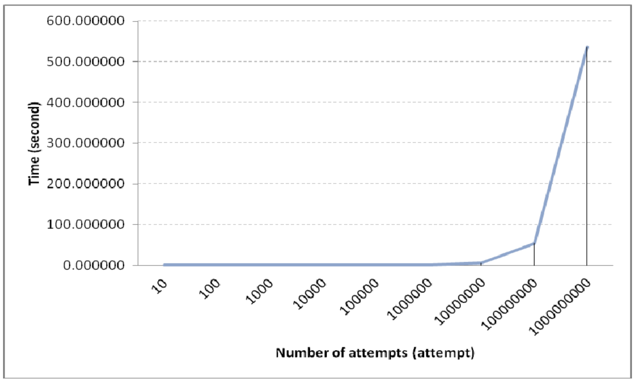

Figure 2 Time taken for sub brute force attack

Another simulation is conducted to measure the average time to perform one typical AES-128 encrypting operation. It is shown that it takes about **37µs** to complete.

From the simulation results, the protocol is robust to brute-force attack. And in case the server desynchronises, it takes just a few seconds to perform the AES encrypting computations to obtain the synchronisation.

B. Timing analysis

1) Timing components

In this section, the minimum and maximum time requirements of the communication among the server, the reader and the tag are analysed. The time complexity is calculated based on the parameters in [14]. The timing model is shown in Figure 3.

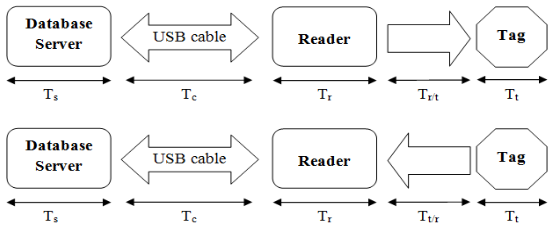

Figure 3. Timing model

In which:

T_s, T_r, T_t : time for the database server, reader and tag to perform the AES algorithm respectively.

T_c : time to transfer 128-bit message from the database to the reader through high-speed USB cable

$T_{r/t}$, $T_{t/r}$: time to transmit 128-bit message from the reader to the tag and from the reader to the tag respectively.

The values of T_s, T_r, T_t and T_c can vary depending on the configuration of the server, reader, tag and the length of the cable, respectively.

2) Assumptions

- T_s, T_r, T_t and T_c are assumed to be constants.
- The computational times for AES encryption and decryption are the same.

3) The minimum and maximum values

Figure 4 shows the timing details of communication between the reader and the tag.

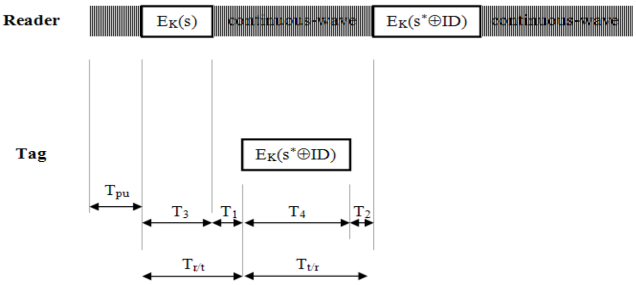

Figure 4. Time of reader & tag communication

In which:

T_{pu} : time for tag to power-up

T_1, T_2 : time from reader transmission to tag response and from tag response to reader transmission respectively

T_3, T_4 : time to transmit 128-bit message from the reader to tag and from the tag to reader

Let T be the total time of an authentication cycle.

$$T = T_{database\rightarrow tag} + T_{tag\rightarrow database}$$

$$= (T_s + T_c + T_r + T_{r/t} + T_t) +(T_t + T_{t/r} + T_r + T_c + T_s) \quad (1)$$

In tag-to-database direction, the value T_r is 0 because the reader just transfers the tag's response to the database without performing any AES operation.

Therefore, (1) can be written as (2).

$$T = 2 \text{ x } (T_s + T_c + T_t) + T_r + (T_{pu} + T_1 + T_2 + T_3 + T_4) \quad (2)$$

4) T_s & T_r

The database requires one AES operation in either database-to-tag or tag-to-database authentication. According to section V.A, the time requirement to perform one AES encrypting function is about 37µs, therefore T_s = 37µs.

The reader requires two AES operations on the way from the database to tag. Based on the assumptions presented in section V.B.2), T_r can be presented by (3).

$$T_r = 2 \text{ x } 37 = 74 \ \mu s \quad (3)$$

5) T_t

The AES encrypting operation at the tag takes 356 clock cycles at the frequency of 100 KHz [20]. So T_t is calculated as below.

$$T_t = 356 \text{ x } \frac{1}{100\times10^3} = 3560\mu s \quad (4)$$

6) T_c

T_c is computed by adding the time of transferring 128 bits from the server to the reader and the delay of the cable. Because the USB 2.0 can transfer at 480 Mbps and the cable delay is 26ηs [21]. So the value of T_c is:

$$T_c = 128 \text{ x } \frac{1}{480\times10^6} + 26\eta s = 0.29267\mu s \quad (5)$$

7) T_{pu}

[14] does not define the minimum of T_{pu}, it only indicates the maximal value of 1500µs for powering up the tag. Therefore, let us assume the $\min(T_{pu}) = \max(T_{pu})$ = 1500 µs.

8) T_1

[14] has given the equation to compute T_1 as (6).

$$\max(RTcal, 10T_{pri})x(1 - FT)-2\mu s\leq T_1\leq\max(RTcal, 10T_{pri}) \quad (6)$$

To calculate RTcal and T_{pri}, some values are given from [14] such as $6.25\mu s\leq Tari\leq 25\mu s$, $2.5Tari\leq RTcal\leq 3.0Tari$, $1.1RTcal\leq TRcal\leq 3RTcal$, $BLF = \frac{DR}{TRcal}$, $T_{pri}=\frac{1}{BLF}$.

Therefore, the minimum and maximum values for RTcal and TRcal can be expressed in (7).

$$\min(RTcal) = 15.625\mu s; \max(RTcal) = 75\mu s$$
$$\min(TRcal) = 17.1875\mu s; \max(TRcal) = 225\mu s \quad (7)$$

Based on the minimum and maximum values of TRcal, the values of BLF and T_{pri} can be found according to [14].

$$\max(BLF) = 465KHz; \min(BLF) = 95KHz$$

$$\min(T_{pri}) = \frac{1}{465\times10^3} = 2.15\mu s; FT = 19\%$$

$$\max(T_{pri}) = \frac{1}{95\times10^3} = 10.53\mu s; FT = 5\% \quad (8)$$

Hence the minimum and maximum values of T_1 can be derived from (6) as (9) and (10), respectively.

$$\min(T_1) = \min(\max(RTcal,\ 10T_{pri})\ x\ (1 - FT) - 2\mu s) \quad \textbf{(9)}$$

$$= \max(\min(RTcal),\ 10\ x\ \min(T_{pri)})\ x\ (1 - FT) - 2\mu s$$
$$= 15.415\mu s$$

$$\max(T_1) = \max(\max(RTcal,\ 10T_{pri})) \quad \textbf{(10)}$$

$$= \max(\max(RTcal),\ 10\ x\ \max(T_{pri})) = 105.3\mu s$$

9) T_2

The equation used to calculate T_2 is referred from [14].

$$3T_{pri} \le T_2 \le 20T_{pri}$$

Therefore, the minimum and maximum values of T_2 can be computed as (11).

$$\min(T_2) = 3\ x\ \min(T_{pri}) = 6.45\mu s$$
$$\max(T_2) = 20\ x\ \max(T_{pri}) = 210.06\mu s \quad \textbf{(11)}$$

10) T_3 & T_4

In the reader-to-tag transmission, the reader starts signalling with either a preamble or a frame-sync as specified in [14]. This investigation assumes that the reader uses the frame sync which includes 3 components: delimiter=12.5µs; data-0=1Tari; RTcal [14]. The 128-bit data are sent afterward. At the end of the signalling either 2-bit 00 or 11 indicates the end of a communication. On the other hand, the tag starts the tag-to-reader signalling with a 6-bit preamble, then 128-bit data and 2-bit end of signalling.

The minimum and maximum values of T_3 are shown in (12) and (13) at the data rate of 465 KHz and 95 KHz respectively.

$$\min(T_3) = t_{delimiter} + \min(1Tari) + \min(RTcal) + \min t_{130\ bits}$$

$$= 12.5 + 6.25 + 15.625 + \frac{130}{465 \times 10^3} \times 10^6 = 313.94\mu s \quad \textbf{(12)}$$

$$\max(T_3) = t_{delimiter} + \max(1Tari) + \max(RTcal) + \max t_{130\ bits}$$

$$= 12.5 + 25 + 75 + \frac{130}{95 \times 10^3} \times 10^6 = 1480.92\mu s \quad \textbf{(13)}$$

Likewise, the minimum and maximum values of T_4 are computed as (14) and (15).

$$\min(T_4) = \min t_{136\ bits} = \frac{6+128+2}{465\times10^3}\times10^6 = 292.47\mu s \quad \textbf{(14)}$$

$$\max(T_4) = \max t_{136\ bits} = \frac{6+128+2}{95\times10^3}\times10^6 = 1431.58\mu s \quad \textbf{(15)}$$

Finally, deriving from (2) and the calculations above, the minimum and maximum values of T now can be resulted as (16) and (17).

$$\min(T) = 2\ x\ (T_s + T_c + T_t) + T_r + \min(T_{pu}) + \min(T_1) + \min(T_2) + \min(T_3) + \min(T_4) \approx 0.0094s \quad \textbf{(16)}$$

$$\max(T) = 2\ x\ (T_s + T_c + T_t) + T_r + \max(T_{pu}) + \max(T_1) + \max(T_2) + \max(T_3) + \max(T_4) \approx 0.012s \quad \textbf{(17)}$$

VI. CASE STUDY: CONVEYOR BELT SYSTEM

The analysis and performance explained in section V.B are applied to estimate the maximum number of attempts for resynchronisation to a typical belt system as in Figure 5. The items with RFID tags are moving evenly from position A to position B on the conveyor belt. The range of the reader is L and θ is the angle covered by the antenna of the reader.

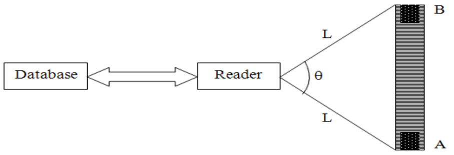

Figure 5. Model of RFID-based conveyor belt

The speed of the conveyor belt, V_b, is 2.5m/s which is the maximum speed in [22]. Hence, the time an item needs to move from A to B, denoted by t, is calculated as (18).

$$t = \frac{AB}{V_b} = \frac{2 \times L \times \sin\left(\frac{\theta}{2}\right)}{V_b} \quad \textbf{(18)}$$

[1] claimed that the read range varies from nominal range of 10cm to extended range of 50cm, so min(L) = 10cm and max(L) = 50cm. In addition, theoretically $0 \le \theta < 180$, it is inferred that $0 \le \sin\left(\frac{\theta}{2}\right) < 1$.

The maximum number of resynchronisation attempts, N is computed as (19) and (20) which are the best case scenario and worst case scenario respectively. These computations are based on (16), (17) and (18).

$$N = \text{floor}\left(\frac{t}{\min(T)}\right) = \text{floor}\left(\frac{L \times \sin\left(\frac{\theta}{2}\right)}{0.0094}\right) \quad \textbf{(19)}$$

$$N = \text{floor}\left(\frac{t}{\max(T)}\right) = \text{floor}\left(\frac{L \times \sin\left(\frac{\theta}{2}\right)}{0.012}\right) \quad \textbf{(20)}$$

Figure 6 and Figure 7 shows the value of N in (19) and (20) respectively with different values of L and θ.

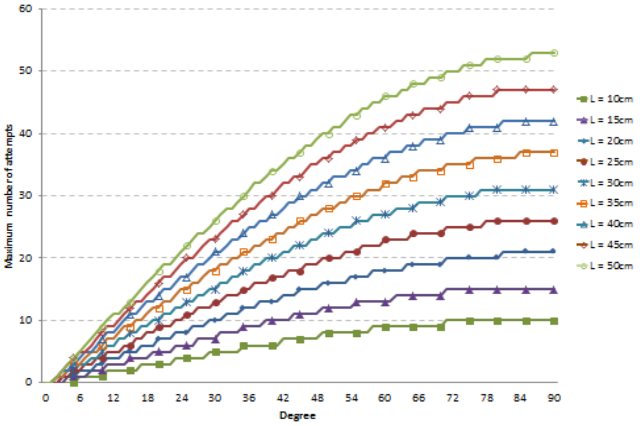

Figure 6. Maximum number of attempts to resynchronise a tag for various L and θ in case of min(T)

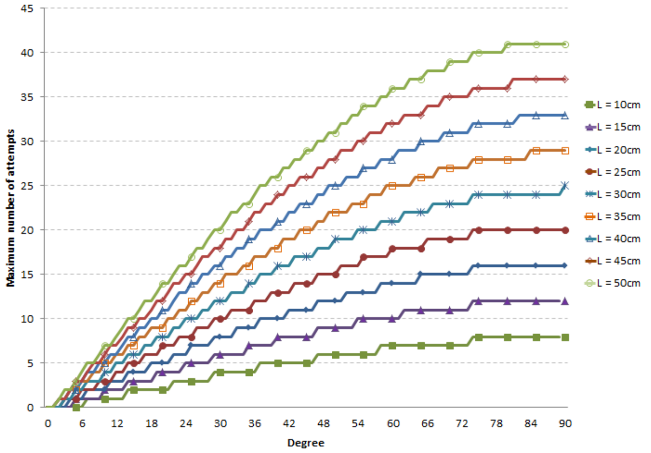

Figure 7. Maximum number of attempts to resynchronise a tag for various L and θ in case of max(T)

If θ = 68° is chosen as a reliable read angle as mentioned in [23], the values of N for min(L) = 10cm and max(L) = 50cm, are 6 and 29, respectively for the best case scenario or min(T). On the other hand, for the worst case scenario or max(T), the values are 4 and 23, respectively.

VII. CONCLUSION

The proposed protocol in this paper is a mutual authentication protocol which utilises AES-128 as a primitive to encrypt the messages transmitted on the channel. With that cipher block, the protocol can protect against many types of attacks such as information leakage, tag tracking etc. In addition, the secure keys stored in tag and sever are always updated in each authentication session, it is impossible for attackers to play the replay attack or trace back the previous data.

REFERENCES

[1] A. Juels, "RFID security and privacy: a research survey," Selected Areas in Communications, IEEE Journal on, vol. 24, pp. 381-394, 2006.

[2] D. Dang Nguyen, L. Hyunrok, D. M. Konidala, and K. Kwangjo, "Open issues in RFID security," in Internet Technology and Secured Transactions, 2009. ICITST 2009. International Conference for, 2009, pp. 1-5.

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

[4] J.-S. Chou, Y. Chen, C.-L. Wu, and C.-F. Lin, "An efficient RFID mutual authentication scheme based on ECC," IACR Cryptology ePrint Archive, vol. 2011, p. 418, 2011.

[5] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong Authentication for RFID Systems Using the AES Algorithm," presented at the Workshop on Cryptographic Hardware and Embedded Systems CHES 2004, Boston Marriott Cambridge, Cambridge (Boston), USA, 2004.

[6] M. F. Mubarak, J. l. A. Manan, and S. Yahya, "Mutual Attestation Using TPM for Trusted RFID Protocol," in Network Applications Protocols and Services (NETAPPS), 2010 Second International Conference on, 2010, pp. 153-158.

[7] B. Toiruul and K. Lee, "An Advanced Mutual-Authentication Algorithm Using AES for RFID Systems," IJCSNS International Journal of Computer Science and Network Security, vol. 6 No.9B, 2006.

[8] Y. Eun Jun, "Improvement of the securing RFID systems conforming to EPC Class 1 Generation 2 standard," Expert Systems with Applications, vol. 39, pp. 1589-1594, 2012.

[9] H. Li, P. Yin, X. Wang, and L. Pang, "A Novel Hash-based RFID Mutual Authentication Protocol," in Computational Intelligence and Security (CIS), 2011 Seventh International Conference on, 2011, pp. 774-778.

[10] T. C. Yeh, Y. J. Wang, T. C. Kuo, and S. S. Wang, "Securing RFID systems conforming to EPC Class 1 Generation 2 standard," Expert Systems with Applications, vol. 37, pp. 7678-7683, 2010.

[11] L. Yanfei, "An Efficient RFID Authentication Protocol for Low-Cost Tags," in Embedded and Ubiquitous Computing, 2008. EUC '08. IEEE/IFIP International Conference on, 2008, pp. 180-185.

[12] H. Y. Chien and C. H. Chen, "Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards," Computer Standards &amp; Interfaces, vol. 29, pp. 254-259, 2007.

[13] National Institute of Standards and Technology (NIST). (2001, 05 Dec 2011). Announcing the ADVANCED ENCRYPTION STANDARD (AES). FIPS 197. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[14] EPCglobal, 2008. EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communication at 860 MHz - 960 MHz Version 1.2.0. Available: http://www.epcglobalus.org

[15] I. Erguler and E. Anarim, "Attacks on an Efficient RFID Authentication Protocol," in Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on, 2010, pp. 1065-1069.

[16] P. P. Lopez, J. C. H. Castro, J. M. E. Tapiador, and A. Ribagorda, "Cryptanalysis of a novel authentication protocol conforming to EPC-C1G2 standard," Comput. Stand. Interfaces, vol. 31, pp. 372-380, 2009.

[17] M. Habibi, M. Gardeshi, and M. R. Alaghband, "Practical Attacks on a RFID Authentication Protocol Conforming to EPC C-1 G-2 Standard," International Journal of UbiComp (IJU), vol. Vol.2, No.1, 2011.

[18] [18] A. Mitrokotsa, M. Rieback, and A. Tanenbaum, "Classifying RFID attacks and defenses," Information Systems Frontiers, vol. 12, pp. 491-505, 2010.

[19] N. PK, P. Kumar, and A. A. Philip. AES Encrypt – Source code in C/C++ for AES Encryption. Available: http://www.hoozi.com/downloads/

[20] T. Good and M. Benaissa, "692-nW Advanced Encryption Standard (AES) on a 0.13-um CMOS," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, vol. 18, pp. 1753-1757, 2010.

[21] Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, and Philips. (2000, USB 2.0 Specification. Available: http://www.usb.org/developers/docs/usb_20_1011111.zip

[22] L. Simon, P. Saengudomlert, and U. Ketprom, "Speed Adjustment Algorithm for an RFID Reader and Conveyor Belt System Performing Dynamic Framed Slotted Aloha," in RFID, 2008 IEEE International Conference on, 2008, pp. 199-206.

[23] H. D. Chon, S. Jun, H. Jung, and S. W. An, "Using RFID for Accurate Positioning," Journal of Global Positioning Systems, vol. 3, pp. 32-39, 2004.