A Gen2v2 Compliant RFID Authentication and Ownership Management Protocol

Haifeng Niu and S. Jagannathan
Department of Electrical and Computer Engineering
Missouri University of Science and Technology
Rolla, USA

Eyad S. Taqieddin
Network Engineering and Security Department
Jordan University of Science & Technology
Irbid, Jordan

***Abstract*—Passive RFID tags and readers are initialized with secret keys which are updated after a successful cycle of authenticatio. Ownership transfer builds upon mutual authentication where a tagged item is shifted from one owner to another. Since the available protocols provide limited security for passive RFID systems and are vulnerable to attackers, we propose a novel ultra-lightweight authentication and ownership management protocol which conforms to the EPC Class-1 Generation-2 Version 2 standard while taking into account the storage and computational resources of the tags. The protocol is successfully implemented on hardware to overcome the weaknesses of the available protocols. The experimental results show that the use of such protocol ensures security with little added communication and computation overhead.**

***Index Terms*—authentication, Gen2v2, ownership transfer, RFID, security**

I. INTRODUCTION

In a passive RFID system, the communicating parties (tag and readers) authenticate each other before information exchange in order to prevent an attack on the wireless communication. Moreover, the information should be concealed from unauthorized parties by encryption. As such, it is necessary for the reader and the tag to share certain secrets which are used to authenticate each other.

Besides authentication, it is also important to implement ownership management protocols as many tagged items change owners more than once during their lifetime. Also, special attention in terms of security must be paid since this process is more vulnerable to attackers because of the exchange of secret keys or/and passwords. In addition, it is necessary for the ownership management protocol to protect the privacy of the old owner from being tracked by the new owner and vice versa.

In order to add security functionality to the tags, recently, the EPC Class-1 Generation-2 standards version 2, referred as "EPC Gen2v2" [1], has been ratified. The new standard, which is backward-compatible with the previous version "EPC Gen2v1" [2], provides some new features with intention to improve security of the RFID systems by allowing the manufactures of RFID tags to customize and implement the cryptographic authentication protocols to avoid unauthorized access and verify identity and provenance.

Similar to EPC Gen2v1, only the cyclic redundancy check function (CRC), pseudo random number generator (PRNG), and the EXOR operation are permitted to use. However, other measures should be adopted to provide an acceptable level of security considering their limited computational capabilities, because the functions mentioned above are not cryptographic functions. Therefore, it is a major challenge to secure information among RFID devices because of the limited computational capabilities and storage space on the passive tags.

Though an initial work of [3] introduces the ownership transfer protocol, the privacy of the old owner cannot be guaranteed because that the way the secret keys are updated will lead to a de-synchronization attack. In [4], a protocol based on the XOR operation, symmetric cryptography, and hash functions is proposed. However, it is vulnerable to DoS and by manipulating the binary data of the random number (nonce) sent to the tag, the attacker is able to track the location of the tag. Another protocol called product-flow ownership transfer protocol (POP) appeared in [5]. The protocol supports querying, disabling, and updating the secrets on the tag. However, the protocol does not protect the new owner's privacy as the previous owner can still access the tag by exploiting his knowledge of the shared secret keys. Besides, it is prone to desynchronization attacks similar to the protocol in [6].

The ownership management protocols, mentioned above [3-6], are not EPC compliant because of the cryptographic functions used for computing the messages. The authors in [9] propose an EPC compliant lightweight ownership transfer protocol, where they use PRNG and XOR functions on the tag side. But the protocol is sensitive to man-in-the-middle (MitM) and replay attacks. Another EPC compliant protocol is proposed in [10] where a modular division operation is added to the functions on the tag considering that it does not require too many gate elements. However, an attacker can destroy the security by disguising as an owner and updating the secret keys by the same fashion as the new owner does.

The recent ownership transfer protocols [11-13] conforming to EPC Gen2v2 standards use CRC operation as the encryption function and cannot guarantee security due to the linearity property of the CRC function. In fact, as analyzed in [14], the attacker manages to recover all the secrets stored in tags with only a few interactions. Therefore, an EPC compliant authentication and owner management protocol with a satisfactory level of security is to be developed for passive RFID tags.

In this paper, we propose a lightweight mutual authentication

978-1-4799-3780-6/14/$31.00 ©2014 IEEE

and ownership management protocols where PRNG and permutation are used as basic functions to provide the cryptographic functionality. The main contributions of this paper are: 1) the development of a novel authentication and ownership management protocols for passive RFID tags 2) the demonstration of how to implement the proposed protocol within the EPC Gen2v2 standard framework, and 3) hardware implementation and evaluation.

The rest of this paper is organized as follows. The detailed description of proposed ownership management protocol is given in Section II, followed by the security analysis and a comparison with existing protocols in Section III. The hardware implementation and evaluation are given in Section IV. Finally, this paper is concluded in Section V.

II. Proposed Protocol

The following assumptions are made in this paper: 1) links between the readers are secure. Similarly, links between the trusted third party (TTP) and any reader are assumed to be secure. This assumption is reasonable and commonly used because in most cases, the readers are implemented with powerful processors which can incorporate very complex encryption methods to secure the data transmission. 2) Links between the tags and any other entity are considered insecure. 3) The tag and its current owner share some certain secret keys that are only known to them.

A. Initialization

The tag is initialized with the following values:

1) K_M: master key only shared with its owner. A reader with K_M is able to modify K but a reader with key K does not have access to K_M.
2) K: secret key shared with both owners
3) K_{TTP}: key shared between the tag and the TTP.
4) R_{IDi}: The ID of the reader *i* currently owning the tag
5) EPC: short for "electronic product code", the changeless identifying ID of a tag
6) IDS: In the proposed protocol, index pseudonym (IDS) is exchanged instead of using the tag identifier (ID). IDS is a pointer to a database entry in which the information of the tag can be found. The IDS is used in the proposed protocol for the following two reasons: 1) EPC is a constant value and its use in multiple runs of the protocol may reveal information about the tag as well as its location. 2) It is possible for the old owner to track the EPC.

All data units in the proposed protocol are 96 bits long in order to conform to the EPC Gen2v2 standard. However, these 96-bit data units are broken into six 16-bit subunits, for the convenience of implementation. For example, a 96-bit key *A* is broken into six 16-bit subunits, denoted as $A(1), A(2), \ldots A(i), \ldots A(6)$, where $A(i)$ is the *ith* 16-bit subunit. Consequently, all the computations have to be executed six times to obtain the complete 96-bit data.

Before the ownership transfer phase (Section B), a mutual authentication (Section A) needs to be performed in order to verify the authority of the involved parties.

B. Phase I: Mutual authentication

A general scenario of an authentication session starts with the reader querying a tag and the tag sends an index pseudonym (IDS) in response. A sequence of message exchanges follows for the purpose that the reader securely sends the random numbers to the tag by making use of the shared key. Then the reader authenticates the tag and vice versa, and both the IDS and the secret keys are updated.

The transactions that take place during the authentication phase are shown in Fig. 1-a. In Fig. 1-b, we show the details of how the authentication is performed using the command and response set defined by EPC Gen2v2 standards. In the rest of this section, we will look into these two figures and explain the authentication phase.

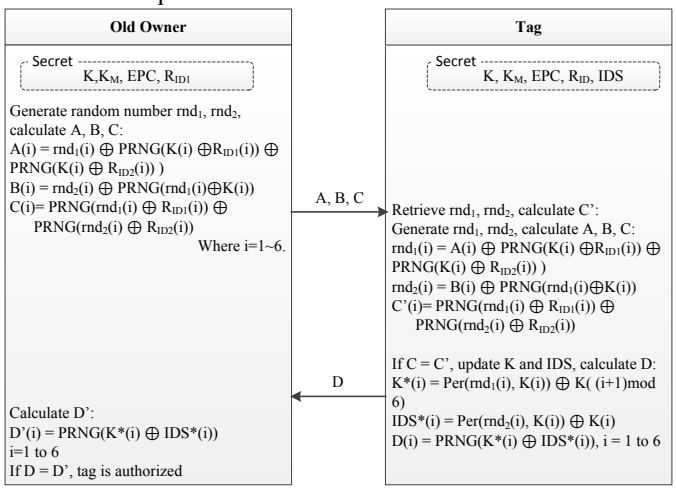

Fig 1-a. Mutual authentication and keys update.

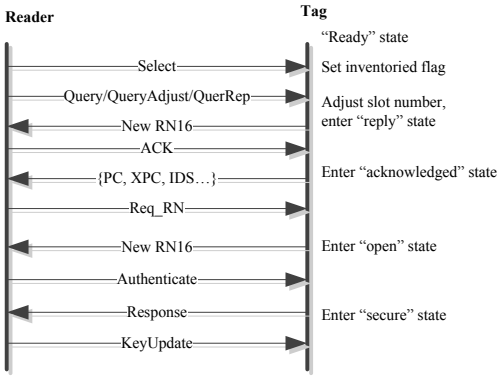

Fig 1-b. Mutual authentication under EPC Gen2v2 standards.

First of all, as shown in Fig. 1-b, the current (old) owner sends "select" and "query" command (and "QueryAdjust", "QueryRep" commands, if necessary) for the purpose of identifying the target tag from a large population of tags. Eventually, the target tag replies with a new 16-bit random number RN16 and shifts its state from "ready" to "reply". After that, the reader issues an ACK command containing the same RN16 (so that the other tags will not reply in the following session) and the tag replies with IDS, XPC, PC, and other information described in the EPC Gen2v2 standards specifications. Next, upon receiving the "Req_RN" command

with the correct RN16 and access key, the tag responds with the new RN16 and enters the "open" state.

Next, we make use of the "Authenticate" and "KeyUpdate" commands, which are newly introduced in EPC Gen2v2, to complete the mutual authentication phase. As specified in [1], the "Authenticate" command contains fields listed in Table I. In particular, we define the contents of "message" field in the "Authenticate" command as described in Table II. The "command ID" is used to indicate that this command will send the necessary security parameters (R_{ID}, A, B…) and start the authentication phase.

TABLE I AUTHENTICATE COMMAND [1]

	Command	RFU, SenRep…	Length	Message	RN	CRC
# of bits	8	…	12	variable	16	16
Note	11010101	Details in EPC Gen2v2 standards[1]				

TABLE II "MESSAGE" FIELD IN "AUTHENTICATE" COMMAND

	Command ID	R_{ID1}	R_{ID2}	A	B	C
# of bits	8	96	96	96	96	96
Note	00000001	Details in Fig. 1-a				

The main purpose of the mutual authentication phase is to: 1) prove the possession of shared secret keys to each other without disclosing it; 2) pass the nonces that are used to update secret keys to the tag. To achieve this, the reader first generates two 96-bit random values (rnd_1, rnd_2) as the nonces, then computes messages A, B, and C in a way described in Fig. 1-a. Particularly, in the computation of message A and B, the secret keys are parts of the input of PRNG function so that the keys are protected meanwhile the tag is still able to verify the reader's possession of the keys by doing the same computation. Furthermore, message C is used to check if the tag has retrieved the correct nonces (rnd_1, rnd_2) from messages A and B.

If C equals to C', then it is believed that the reader does have the secret key and the tag has retrieved rnd_1 and rnd_2 successfully. Then the new key and IDS are computed in a way specified in Fig. 1-a. Similarly, we use message D in order to: 1) prove the tag's possession of the secret keys; 2) inform the reader that the tag has computed the new keys and IDS. This value D is contained in the response message of the "Authenticate" command, as described in Table III. A non-zero value in the "status" field indicates that the tag has retrieved the nonces and computed the new keys and IDS.

TABLE III. RESPONSE MESSAGE OF THE "AUTHENTICATE" COMMAND

	Command ID	Status	Length	Message	RN16	CRC
# of bits	8	2	10	96	16	16
Comments	00000010			D		

Upon receiving a response with the "status" of success, the reader computes D' in the same way of computing D. Only if D equals to D', the tag is considered as authorized. As a result, the reader issues a "KeyUpdate" command to the tag for conformation. Consequently, the tag updates to the new computed IDS and keys for future uses. Note that both the tag and the reader should maintain a copy of the old IDS and secret keys to avoid desynchronization problems (this will be explained in details in Section C).

Note that the permutation (Per) function [15] is an ultra-lightweight operation which offers diffusion of the bits and helps overcome any problem occurring because of the nature of bitwise operations.

C. Phase II: Complete ownership transfer

In this phase, we propose to use TTP to guarantee the correctness of the protocol. The need for the TTP arises from the fact that the old owner holds the same secret keys shared between the new owner and the tag. As a result, any update taking place by R_{ID2} may be mirrored by R_{ID1}. This violates an important security requirement of ownership transfer which is backward privacy.

Therefore, the goal here becomes to how to change the value of K_M stored on the tag such that it matches that stored on R_{ID2}. After that, R_{ID1} will no longer have access to the tag. In fact, this proposed approach adds an extra functionality that we may use the reverse process in case we wish to satisfy the ownership repossession property.

As described in Fig. 2, the ownership transfer phase of the proposed protocol consists of those steps:

1) TTP generates a random number $rnd1$ and uses it to update K_M to K_M^*. This will become the new master key shared between the tag and the new owner, R_{ID2}.

2) TTP sends K_M^* to RID2 using the secure channel.

3) The challenging part for the TTP becomes to send K_M^* to the tag. For that, we propose using the messages A and B shown in Fig. 2. Similar to what we have done in authentication phase, the secret key is set as the input of the nonlinear PRNG function while the nonce is EXORed with the PRNG output so that the key will not be disclosed and the nonce can be passed to the tag safely. Message B is used for the tag to verify TTP's possession of the secret key and to check the correctness of the nonce.

4) The tag retrieves $rnd1$ from A and verifies that $B? = B'$.

5) The value of $rnd1$ is used by the tag to update K_M^* in a manner same to that used by the TTP.

6) The new owner and tag need to challenge each other to verify that both have the same value of K_M^*.

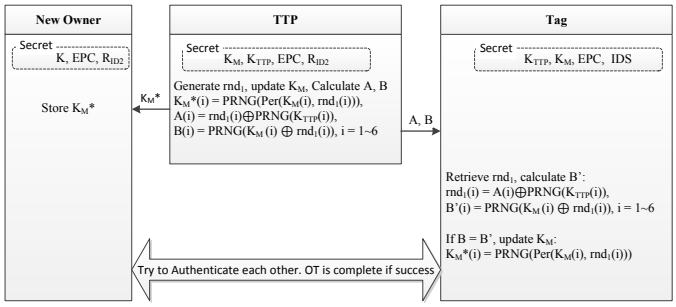

Fig. 2 Complete ownership transfer.

III. Comparison with Related Protocols on Security

In order to perform a comparison between the proposed ownership management protocol and the previous related work, we need to analyze the proposed protocol in terms of the security requirements mentioned earlier.

- *Tag/reader impersonation*: Any impersonated tag/reader with unmatched keys will results in totally different A, B, and C values thus leads the failure of authentication.
- *Replay attack*: If an attacker eavesdrops all the messages a tag/reader uses to prove authority, he can neither retrieve the secret keys nor use the messages for another round of authentication because the random numbers are newly generated every time.
- *MitM attack*: The PRNG function ensures that if the attacker flips even one bit of R_{ID2}, the tag will get a totally different (and incorrect) rnd1. Consequently, rnd_2 derived by the tag will be incorrect as we use rnd_1 to compute rnd_2. Therefore, even the attacker flips the same bit of message B, it will not pass the test that C' equals to C. In fact, authentication fails if any bit(s) of the transmission data are modified by the attacker.
- *Location privacy*: The proposed protocol uses IDS rather than the unique and life-long changeless identifier EPC to protect location privacy because IDS is updated after each successful authentication. Consequently, the attacker cannot determine the presence of the target tag during the ownership transfer process and thus the location privacy is protected.
- *Forward untraceability*: The forward untraceability is ensured by maintaining a copy of previous secret keys on the old owner's side which the new owner has no right to access after a successful ownership transfer.
- *Backward untraceability*: As mentioned earlier, backward without TTP cannot be satisfied as the old owner can perform the same action to update the secret keys. The use of TTP guarantees that only the new owner can update the keys and thus provide backward untraceability.
- *Desynchronization problem*: Desynchronization problem cannot be completely prevented because the adversary can always choose to block the last conformation message and consequently one party updates the keys while the other one does not. Our solution is that TTP should always keep a copy of the previous secret keys and the corresponding tag's ID in case of confronting desynchronization attacks. In that case, the new owner will not be able to authenticate the tag and then TTP should attempt to resend the key update message until the ownership transfer succeeds.
- *Windowing problem*: It is quite easy to verify that in the proposed protocol, the old owner and the new owner never possess the master key at the same time.

A comparison of some most important performance indices with previous related work is shown in Table. IV, where a "Y" means the scheme qualifies the requirement while an "N" the opposite. From the table it can be concluded that among the non-EPC-compliant protocols, Kapoor's [16] has the best performance but it still suffers from windowing problem and is not suitable for low-cost RFID tags due to the use of Hash functions. On the other hand, the existing EPC compliant protocols [9][11] either fail to provide backward untraceability or are vulnerable to replay attack because of using CRC as encryption method. In contrast, the proposed protocol not only conforms to the EPC standards, supports delegation, and also satisfies the security requirements.

Table IV. Comparison with Previous Related Work

Schemes	[4] Osaka	[16] Kapoor	[17] Song	[9] Seo	[11] Chen	Our scheme
EPC compliant	N	N	N	Y	Y	Y
Resist Replay attack	Y	Y	N	N	N	Y
Location Privacy	Y	Y	N	Y	Y	Y
Backward untraceability	N	Y	Y	N	Y	Y
Desynchronization	N	Y	N	Y	Y	Y
Windowing	N	N	N	Y	Y	Y

IV. Hardware Implementation and Evaluation

In this section, the proposed authentication and ownership transfer protocol is implemented and analyzed in hardware. Since the new EPC Gen2v2 protocol was ratified very recently, there is no reader available in the market that supports the new standards yet. Our solution is choosing a Gen1v1 RFID tag and simulating the Gen2v2-only commands ("Authenticate", "KeyUpdate") by using the "BlockWrite" and "Read" commands. As these commands are of similar amounts of bits, theoretically the differences in terms of processing time and energy consumption are negligible.

A. Implementation

The RFID platform presented in [18] is chosen to implement and evaluate the proposed protocol. Operating in UHF frequency range, this platform is modified based on the Wireless Identification and Sensing Platform (WISP), developed by Intel Research Seattle [19]. Similar to the WISP tags, the program running in the modified WISP tags also conforms to the EPC Gen2v1 standards. Therefore, the tag is able to communicate with most of the UHF RFID readers. On the modified WISP tag, a "bowl tie" antenna (Fig. 3) and a four-order Dickson charging pump are adopted to convert the RF signal to DC power to support the on-board circuitry. The 16-bit microprocessor, MSP430F2132 by Texas Instrument, has ultra-low power consumption (about 600μA at 1.8V & 4MHz). It can execute an instruction in as little as 0.25μs. Moreover, the 1Mbit EEPROM, 24AA1026, embedded only on the modified WISP tags ensures enough storage space for the data such as IDS or secret keys.

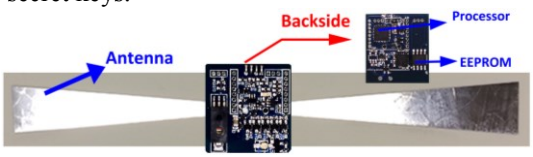

Fig. 3. Modified WISP: Class-1 Generation-2 UHF passive RFID tag platform.

On the reader side, the proposed protocol is implemented in Java in Eclipse, beyond the "Reader library" and the "LLRP [20]" layers. On the tag side, the proposed protocol is

implemented at a higher level to stay compliant with the EPC Gen2v1 standards. The IAR Workbench for MSP430 is used for debugging and downloading the program. The RFID reader used in this experiment is Impinj Speedway Revolution R220, with transmitting power set to 30dBm and receiving sensitivity -70dBm.

B. Energy/Time consumption

First, it is of interest to know the execution time for a complete ownership transfer process (including authentication phase) when sufficient energy is provided on the tag. To achieve this, the RFID tag is placed as close as 0.5m away from the reader antenna to ensure it can harvest enough power. Note that however complicated one protocol is, it can be broken into steps that belong to one of the four categories: a) computation on tags, b) computation on readers (here consider TTP as a reader), c) data exchange between readers and tags ($T \leftrightarrow R$), data exchange between two different readers ($R \leftrightarrow R$).

As mentioned earlier, both the computation on readers and data exchanges between two readers can be negligible due to the powerful processor and broad network bandwidth. As presented in Table V, it can be seen that the total time of the on-tag computation plus the data exchange between the tag and the reader is 331.02ms, which is very close to the actual measured total time (380.08ms). This result agrees with the previous analysis. In fact, the time consumed for the on-tag computation is only 15.02ms (45060 instruction cycles @ 3MHz) for the authentication and ownership transfer phase and 11.13ms for the delegation phase (33396 instruction cycles @ 3MHz), which confirms the ultralight weight property of the proposed protocol.

TABLE V. MEASURED TIME AND INSTRUCTION CYCLES

Notation	Definition	Value	Cycles
N_{TR}	Number of T↔R rounds	8	-
T_{TR}	Time for each T↔R round	39.50ms	-
T_{auth}	Time of computation on tag during authentication phase	11.12ms	33360
T_{tran}	Time of computation on tag during OT phase	3.90ms	11700
T_{tag}	$T_{tag} = N_{TR}* T_{TR}+ T_{auth}+ T_{tran}$	331.02ms	-
T_{total}	Actual measured total time	380.08ms	-

Since passive RFID tags are powered by the RF signal emitted from the reader antenna, the energy being harvested decreases when the tag is moved away from the reader antenna. Therefore it is also of interest to measure the execution time where there is insufficient energy on the tag. In this case, the tag is placed at different distances away from the reader antenna and the corresponding number of successful ownership transfer per minute is calculated. In contrast, the experiments are repeated using the same tag running the protocol, with all the computation eliminated. In other words, the control group only executes the program codes of running the same number of data exchanges.

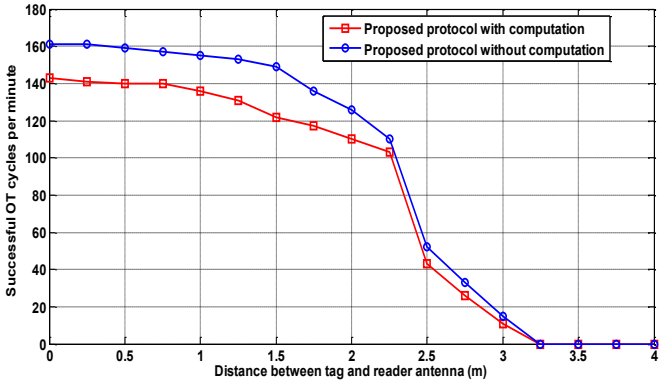

Fig. 4. Number of successful OT cycles per minute.

From Fig. 4, it can be concluded that when the distance between the tag and the reader antenna is less than 1m, the number of successful OT cycles per minute is almost constant because sufficient power has been harvested at such short distance. As the distance increases, however, the number of successful OT cycles goes down because of the failure of data exchange caused by the insufficient power. As a consequence, the reader will either start over a new OT cycle or request for a retransmission, which both take longer time. When the distance is longer than 3.0m, the proposed protocol with or without computation can only be executed for a very limited number of cycles due to the lack of energy. However, the most important conclusion is that, if one compares the two curves with each other, the number of cycles executed per minute for the proposed protocol with computation is only slightly less than that the one without computation. As a result, this conforms the computation overhead brought by this protocol is insignificant. Next the location privacy is discussed.

C. Location Privacy

As mentioned earlier, we proposed to use IDS to protect the location privacy by updating IDS after each authentication round. Therefore, it is of interest to examine the degree of differences between the old and updated IDS. To do this, we let the reader run the authentication N times consecutively, and record each IDS as IDS_i, where i = 1, 2, …, N. Then the following two metrics are considered:

1) $HD_{avg,all}$: The average hamming distance of all pairs of IDS, as computed in (5), where H(x, y) is the hamming distance of two 96-bit binary number x and y.

$$HD_{avg,all} = \left(\frac{N(N-1)}{2}\right)^{-1} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} H(IDS_i, IDS_j) \qquad (5)$$

2) $HD_{avg,cons}$: The average hamming distance of two consecutive IDS, as computed in (6).

$$HD_{avg,cons} = (N-1)^{-1} \sum_{i=1}^{N-1} H(IDS_i, IDS_{i+1}) \qquad (6)$$

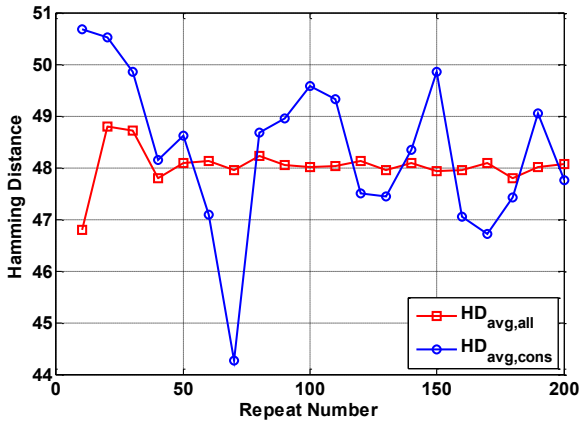

Fig. 5 Measured Average hamming distance

The results are shown in Fig. 5, from which we can see that the average hamming distance of all pairs of IDS stays stable at around 48, showing a good degree of randomness. The average hamming distance of consecutive IDS, on the other hand, varies from 44 to 51, also indicating that it is very difficult for the attacker to predict the next IDS from the previous one as no pattern can be easily found**.** As a result, the attacker cannot determine the presence of the tag by analyzing the values of IDS. Next the reader impersonation aspect is considered.

V. CONCLUSIONS

In this paper, a new EPC Gen2v2 compatible protocol by employing a lightweight permutation and PRNG function is introduced. Such use of a simple operation adds a minimal level of computation or energy consumption while, at the same time, supports the cryptographic goals of the protocol. The comparison with previous work shows the proposed protocol not only conforms to the EPC standards, but also satisfies the security requirements. The hardware implementation supports our initial goal of adding security to the existing EPC Gen2v2 based tags such that the system would be secure both in the case of being used by a single owner or in the more practical cases of having multiple owners during the lifetime of a tagged item.

ACKNOWLEDGEMENTS

This research is supported in part by Intelligent Systems Center.

REFERENCES

[1] EPC Radio-Frequency Identity Protocols, Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz–960 MHz, Version 2.0.0, EPC Global, Nov., 2013.

[2] EPC Radio-Frequency Identity Protocols, Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz–960 MHz, Version 1.2.0, EPC Global, 2008.

[3] J. Saito, K. Imamoto, and K. Sakurai, “Reassignment Scheme of an RFID Tag’s Key for Owner Transfer,” In T. Enokido, L. Yan, B. Xiao, D. Kim, Y. Dai, and L.T. Yang, editors, Embedded and Ubiquitous Computing – EUC 2005 Workshops, LNCS volume 3823, pp. 1303–1312, Springer, Berlin, November 2005.

[4] K. Osaka, T. Takagi, K. Yamazaki, O. Takahashi, “An efficient and secure RFID security method with ownership transfer," International Conference on Computational Intelligence and Security, vol. 2, pp.1090-1095, 2006.

[5] S. Koralalage, S. Mohammed Reza, J. Miura, Y. Goto, and J. Cheng, “POP method: an approach to enhance the security and privacy of RFID systems used in product lifecycle with an anonymous ownership transferring mechanism”, Proceedings of the 2007 ACM symposium on Applied computing, pp. 270-275, 2007.

[6] A. Fernandez-Mir, R. Rasua, J. Roca, and J. Ferrer, “A Scalable RFID Authentication Protocol Supporting Ownership Transfer and Controlled Delegation. RFID”, Security and Privacy, LNCS, Volume 7055, pp.147-162, 2012.

[7] Y. Seo, T. Asano, H. Lee, and K. Kim, “A lightweight protocol enabling ownership transfer and granular data access of RFID tags,” Proceedings of the Symposium on Cryptography and Information Security, pp. 1–7, 2007.

[8] R. Doss, Z. Wanlei, and Y. Shui, "Secure RFID tag ownership transfer based on quadratic residues," IEEE Transactions on Information Forensics and Security, vol. 8, no. 2, pp. 390-401, 2013.

[9] C.-L. Chen and C.-F. Chien, “An ownership transfer scheme using mobile RFIDs”. Wireless Personal Communications, 2012.

[10] X. Fu and Y. Guo, “A Lightweight RFID Mutual Authentication Protocol with Ownership Transfer”, Advances in Wireless Sensor Networks Communications in Computer and Information Science Volume 334, pp 68-74, 2013.

[11] C. Chen, Y. Huang, and J. Jiang, “A secure ownership transfer protocol using EPCglobal Gen-2 RFID”, Telecommunication System, Volume 53, Issue 4, pp 387-399, 2013.

[12] J. Munilla, G. Fuchun, and S. Willy, "Cryptanalysis of an EPCC1G2 standard compliant ownership transfer scheme". Wireless Personal Communications, pp. 1-14, 2013.

[13] Y. Tian, G. Chen, and J. Li. “A new ultralightweight RFID authentication protocol with permutation”. IEEE Communications Letters, Vol. 16, No. 5, pp. 702-705, May 2012.

[14] G. Kapoor and S. Piramuthu. "Single RFID tag ownership transfer protocols," IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol. 42, No. 2, pp. 164-173, March 2012.

[15] B. Song. RFID Tag Ownership Transfer. In Workshop on RFID Security RFIDSec’08, Budapest, Hungary, July 2008.

[16] H. Niu and S. Jagannathan, “High memory passive RFID tags with multimodal sensor design and application to asset monitoring in-transit”, Proc. of the IEEE Conference on Int. Instrumentation and Measurement, May 2013.

[17] A. P. Sample, D. J. Yeager, P. S. Powledge, A. V. Mamishev and J. R. Smith, “Design of an RFID-Based Battery-Free Programmable Sensing Platform,” IEEE Transactions on Instrumentation and Measurement, vol. 57, no. 11, pp. 2608-2615, Nov. 2008.

[18] EPCglobal, “Low Level Reader Protocol (LLRP) Version 1.1 Ratified Standard,” Oct, 2010.