SURV: Shelled Ultralightweight Randomized Value Authentication Protocol for Low-Cost RFID Tags

Muaway Naser[1]*, Yazn Alshamaila[2], Rahmat Budiarto[3], Pedro Peris-Lopez[4]

[1] Computer Science & IT Department, Higher College of Technology, UAE.
[2] Business Information Systems Department, King Abdullah II School for Information Technology, The University of Jordan, Jordan.
[3] Smart Networked Computing Research Group, College of Computer Science & Information Technology, Albaha University, KSA.
[4] Computer Security Lab (COSEC), Computer Science Department, Carlos III University of Madrid, Spain.

* Corresponding author. Tel.: 00971569727302; email: muawya.aldalaien@hct.ac.ae
Manuscript submitted January 5, 2015; accepted April 28, 2015.
doi: 10.17706/ijcee.2015.7.3.206-214

Abstract: Current RFID ultralightweight authentication protocols have security pitfalls; thus, SURV, a potentially high-security protocol, based on the use of on-tag lightweight cryptography and on-reader standard cryptography, is proposed. Furthermore, the updating of internal values is only executed on tags and conditioned to a successful mutual authentication, which prunes desynchronization attacks. Results of performance and security analysis of SURV, and comparison with existing protocols, are presented.

Key words: Authentication, lightweight cryptography, privacy, RFID, security.

1. Introduction

Radio frequency identification (RFID) systems produce methodologies that offer data portability by embedding data in small devices (tags) that can be mobile and held within other objects. This data may include identification data to grant access authentication for several transactions or entry passes through secured checkpoints in different locations. The security and privacy issues for this data are vulnerable to various threats [1]-[4]. Recent studies in RFID have attempted to raise solutions for several applications within the field. The solutions offered are often hindered by the fact that, in most cases, different RFID deployments do not meet required security specifications and needs [5]-[7], [2]. These solutions could also be hindered by other implementation obstacles such as cost and compliance with existing RFID standards.

RFID tags can be classified into several categories, each with a distinct processing capability. Each category can be implemented in more than one application domain, where the tag-price is the main factor in deciding the functionality and security features of the tag. Several studies [8]-[14] have agreed with Chien [15] in his classification of RFID tags into four different classes based on their security-related computational power: full-fledged, simple, lightweight, and ultralightweight.

Ultralightweight mutual authentication protocols are some of the most challenging areas in terms of security-related computational power. In these protocols, the tag can only implement simple bitwise operations such as XOR, AND, and OR, among others. In addition, the tags have very limited storage memory. The threat of attacks could hamper communication with the readers. Researchers have proposed protocols to bridge the gap between the limited capability of ultralightweight tags and the assumed security level

against passive and active attacks. Considering passive attacks is mandatory for these protocols, while active attacks are also considered for further security improvement. Moreover, the vast majority of studies in this field have not focused on the tag identification time from the database. Reduced identification time enhances the efficiency of the overall system. Therefore, a few studies [16]-[18] have proposed scalable mutual authentication protocols focused on identification time. However, none of these studies has implemented their research in the area of ultralightweight mutual authentication protocols. In this paper, we propose a new ultralightweight RFID mutual authentication protocol for low-cost RFID tags. Our protocol has been analyzed to confirm its immunity against the most common attacks, and its ability to offer data integrity, availability, confidentiality, and mutual authentication, in addition to reducing computational complexity for identifying individual tags. This paper is organized as follows. In the next section, we explore previous work in this field. Section introduces the SURV protocol and its procedures. Section 4 discusses the security analysis. The performance evaluation is presented in Section 5. Finally, Section 6 presents our conclusions.

2. Related Work

A set of ultra lightweight mutual authentication protocols known as the UMAP family, were proposed by Peris-Lopez *et al.* in 2006 i.e. M2AP [19], EMAP [20], and LMAP [21] respectively. These protocols were designed to create mutual authentication between tags and readers without overloading the tag side with strong computational power. Therefore, these protocols used only simple bitwise operations (XOR, AND, OR, and additional mod 2^m). These protocols go through three stages, in the first stage, the tag identifies itself to the reader using the IDS pseudonym. In the second stage, the reader exchanges two messages with the tag based on common secret keys by sending a message with three sub-messages and receiving one response message. Finally, if the authentication succeeds, the tag and the reader will update their common secret keys, as well as the IDS.

UMAP protocol security resistance has been thoroughly investigated. [22] and [23] presented de-synchronization attacks, as well as a full disclosure attacks. Based on those attacks, [24] presented another full disclosure attack that was much more comprehensive. Subsequently, passive attacks were proposed by [25] and [26]. The main reason UMAPs are not secure and suffered from several disclosure attacks is that they only use bitwise logical or arithmetic operations such as bitwise XOR, OR, AND and addition, which are triangular functions [27], [28]. Reader should note that the composition of a triangular functions results in another triangular function. A triangular function has the property that outputs bits only depend on the leftmost input bits, instead of on all input bits. This undesirable characteristic (lack of diffusion) greatly facilitated its analysis. Furthermore, the composition of messages using bitwise XOR or AND operation is faulty from a security point of view. More precisely, if we compute the XOR or AND between two variables (i.e., $C=A\&B$ or $C=A|B$) the resulting value is highly biased. In other words, the probability of zeros (ones for the OR operation) is $^3/_4$.

The Strong Authentication and Strong Integrity (SASI) protocol was proposed by H. Chien in 2007 [15]. SASI protocol came as an improvement to UMAP, featuring a Rot() function that was not a triangular function, which is considered an interesting addition to arrive at the assumed secure protocol. Unfortunately SASI protocol failed due to the same mistakes that broke UMAP, and full disclosure attacks had been shown in [29]. T. Cao *et al.* [30] illustrated two desynchronization attacks against SASI, a DoS and a tractability attack. Another desynchronization attack was proposed in [31]. The ultralightweight mutual authentication protocol proposed by Lee *et al.* [11] in 2008 reused the rotation function proposed in SASI; however, Lee *et al.* used one random number and one key. This protocol was analyzed by Peris-Lopez *et al.* in [32] with regards to cloning, tractability, full disclosure, and desynchronization attacks.

Gossamer was proposed recently by Peris-Lopez *et al.* [33]. Gossamer is based on SASI, with a significant

variation of using modulus rotation function instead of hamming weight rotation. Additionally Peris-Lopez came out with MixBits function, which has a very tiny footprint and was obtained by using Genetic Programming (GP). In the MixBits function, the relation between inputs and outputs is highly non-linear (see the original paper for details [33]). Gossamer was designed to cover the gaps in the UMAP family by releasing the bit dependency of messages and secret values. Nevertheless, desynchronization attacks for Gossamer have been proposed in [34]. Rama and Suganya [35] proposed an enhancement for Gossamer using a sign/logarithm number system. Recently, two passive attacks have been proposed in [36]. In conclusion, all ultralightweight mutual authentication protocols created so far are insufficiently robust to prevent security attacks.

3. SURV Protocol

A new protocol was developed utilizing the strengths and addressing the weaknesses of existing protocols, particularly the vulnerabilities in Gossamer. The aspiration in creating the new protocol was to produce an ultralightweight protocol with a higher security level and lower computational power requirements for low-cost RFID tags. The protocol presents transactions of privacy data combined with random values, encapsulated in shell values capable of transporting hidden data between the tag and the reader without compromising or revealing the aforementioned data. From its characteristics, the proposed protocol has been termed the Shelled Ultralightweight Randomized Value (SURV) protocol.

The SURV protocol focuses on securing the channel between the tag and the reader by presuming that the channel between the reader and the third entity, the backend database (DB), is already secured. From here, the reader, and database are treated as one entity. In the initial stage prior to starting any session, each tag is initialized with two keys (k_1, k_2) with fixed values and a tag ID (TID). We recommend that these two secret keys be changed from time to time to maintain a higher security level. These three values are shared with the database where (k_1, k_2) are indexed by TID. Each reader has two distinct fixed common secret keys for all tags: (K_e, K_h) encryption and hash keys. The reader on each transaction creates new four distinct values: i) a random number (R) hidden in a shell value (A), ii) a random number (N), iii) an encryption value (Ψ) holding the TID, and iv) a hash value (φ) preserving the messages' integrity. These four values overwrite their respective previous values in the tag.

The protocol consists of three phases: tag identification, mutual authentication, and tag updating. In the first phase, the reader sends "hello" message to the tag. The tag responds by obtaining the value R_i from the shell value A and using R_i to produce a new shell value B. Afterwards, the values Ψ_i, φ_i, B_i, and N_i are sent to the reader for tag identification purposes. The reader uses this data to extract the TID by decrypting Ψ_i using K_e, and locking-up the tag and its secret keys in the database. Then it starts the mutual authentication phase only if the TID was verified, where it validates the integrity of the data sent from the tag. The reader authenticates the tag by checking the hash value φ_i. Then, it generates a new shell value C_i to enable the tag to authenticate the reader. A new tuple of values A, Ψ, φ and N is also generated and is used to update the tag parameters. Finally, the reader sends these five values C_i, A_{i+1}, Ψ_{i+1}, φ_{i+1}, and N_{i+1} to the tag. When the tag receives this message, it generates R_t' from its initial data, extracts R_t from the shell value C_i, and verifies if $R_t' = R_t$ to complete the mutual authentication and ensure message integrity. If this phase is finished successfully, the third phase, tag updating, begins. This phase uses the rest of the data sent in the last message to update the tag and overwrite the tag parameters A_i, Ψ_i, φ_i, and N_i with the new data A_{i+1}, Ψ_{i+1}, φ_{i+1}, and N_{i+1}, respectively. The protocol procedures are described as follows.

Tag identification. In this phase, the reader begins the session by sending a "hello" message to the tag. The tag first resolves the A_i shell by inversing the function $\{A_i=\text{Rot}(\text{Rot}(R_i+k_1+\pi), k_2), k_1)\}$ and extracting the value of the random number R_i. Next, it re-hides this value in shell B_i by $\{B_i=\text{Rot}(\text{Rot}((R_i+k_2+\pi) \oplus k_1, k_1),$

k_2)+k_2}, and sends a message to the reader using four tag parameters: Ψ_i, φ_i, B_i, and N_i. The reader uses K_e and N_i to obtain the TID using {TID=Dec(Ψ_i) $k_e \oplus N_i$}.

Mutual authentication. The reader locks-up the TID and the responding (k_1, k_2) from the database, resolves the B_i shell by inversing the function {B_i= Rot(Rot((TID +R_i +k_2+ π)$\oplus$ k_1, $k_1 \oplus k_2$), k_2)+k_2} and extracts the value of the random number R_i. Next, the reader refigures the hash value using {φ_i' = h(Ψ_i || R_i || TID)$_{Kh}$} and compares this value with the responding value sent by the tag and verifies if {φ_i'=φ_i}. If true, the reader authenticates the tag and uses k_1, k_2 and R_i to compute Y via {Y= Rot($R_i \oplus k_2$, k_1)}. It then uses Y in creating R_{Temp} via the function {R_{Temp}=MixBits (R_i, Y)}. R_{Temp} will thereafter be used in creating S via {S=Rot(A_{i+1} +Ψ_{i+1}+ φ_{i+1}+ N_{i+1}) $\oplus k_1$, R_{Temp})}, and both S and R_{Temp} will be used to calculate R_t using two-chain MixBits in {R_t=MixBits (MixBits (S ,Y), RTemp)}. After this step, the reader hides R_t in shell C_i using the function {C_i= Rot(Rot((R_t + k_2+ π) $\oplus$ k_1, R_i), k_2+ R_i)+k_1}, and creates new tag parameters as follows: A_{i+1} = Rot (Rot(R_{i+1}+ k_1+ π), k_2), k_1), Ψ_{i+1} = Enc(TID$\oplus N_{i+1}$)$_{Ke}$, φ_{i+1} = h(Ψ_{i+1}|| R_{i+1}|| TID)$_{Kh}$, C_i = Rot(Rot((R_t+k_2+ π)$\oplus k_1$, R_i), k_2+ R_i)+k_1. These parameters (C_i, A_{i+1}, Ψ_{i+1}, φ_{i+1} and N_{i+1}) will be sent in one message to the tag, where the tag calculates R_t', extracts R_t from C_i and verifies if R_t'=R_t to conclude the mutual authentication phase.

Tag updating. If R_t'=R_t is true, the third phase starts by overwriting the tag parameters A_i, Ψ_i, φ_i, and N_i with the new data A_{i+1}, Ψ_{i+1}, φ_{i+1}, and N_{i+1} respectively, and then ends the session.

It can be seen from Fig. 1 that, three shells, A, B, and C, are generated in every session. This makes it difficult to predict the embedded values, which would be useless if obtained after the session termination. There are also two verification challenges, φ and R_t, allowing the system to terminate the unsuccessful session in three positions, including the TID lock-up, and starting a new session in another timeframe.

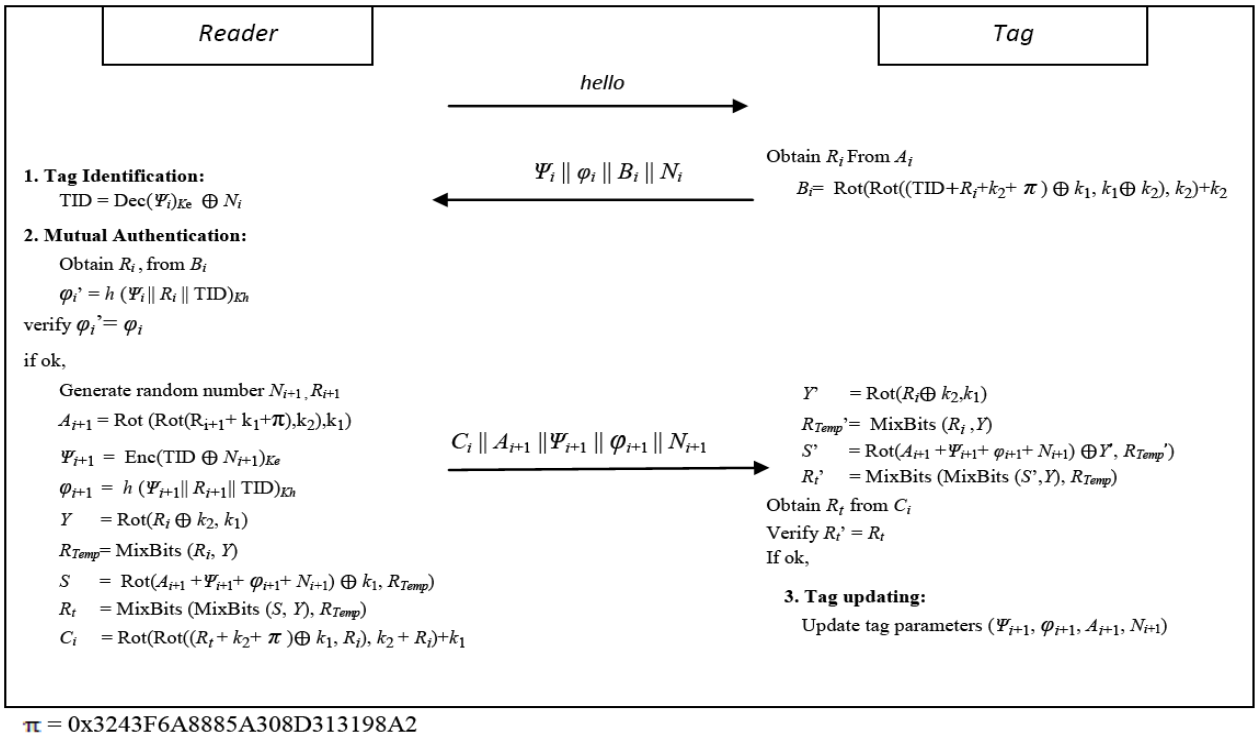

Fig. 1. SURV protocol.

4. Security Analysis

We conducted a security analysis against the most relevant threats to the security of our protocol. The analysis was conducted by investigating each attack and its requirement and properties.

User data confidentiality. The secret keys k_1 and k_2 are carefully hidden inside *A*, *B*, and *C*. In every new session, the keys are mixed with two different random numbers using the MixBits function. In fact, in our protocol, even if any of the sub-messages in either *A*, *B*, or *C* was broken, the tag identity will still be anonymous to the adversary. This is because the tag ID was encrypted after it was XORed with a random number using a secret key that exists only in the reader database. Therefore, the tag identity can be recognized only by legitimate readers.

Tag anonymity. The sub-messages are updated in each session transaction, and the authentication messages between the tag and the reader are mixed with random numbers. This renders the adversary unable to recognize the tag's location or trace it unless the adversary keeps interrupting the communication between the same tag and any legitimate reader, which leads to the transmission of the same message (Ψ_i || φ_i || B_i || N_i) every time. This scenario was not considered of any considerable value and had been ignored in most previous studies in the domain since the tag was unable to randomize itself due to its limited recourses. An in-depth analysis for all these scenarios has been given in detail in [37].

Mutual authentication and data integrity. Our mutual authentication protocol can be performed only between legitimate readers and legitimate tags. This is because sub-messages *A*, *B*, and *C* can be generated using the common secret keys k_1 and k_2, which are only held in the tag and in the backend database. In addition, verifying the values of φ and R_t composed by the reader and the tag to the values φ' and R_t' respectively, provides strong data integrity validation.

Forward security. There are no possibilities for inferring any data from past communications between the tag and the reader because any previous data not overwritten by the end of the session will be dropped due to its uselessness. Keys k_1 and k_2 are not dropped; however, they are difficult to obtain and can be changed repeatedly, making this attack quite impossible. Assuming that tag is somehow compromised, there are still several unknown data variables in the server, such as K_e, K_h.

Resistance to replay attacks. An adversary might eavesdrop on any of the exchanged messages; however, it would not be useful to send it back to either the reader or the tag. This is because each message is based on random numbers that are changed in every successful authentication session. Accordingly, a replay attack can be detected immediately once the message is received by either the tag or the reader.

Data update confirmation and desynchronization. Most of the recent authentication protocols require updating the secret keys data between the tag and reader, which in cases where the transmitted data has been modified or even interrupted, leads to desynchronization. A de-synchronization attack is the first vulnerability that commonly appears in all the current protocols. In our protocol, the tag does not require the updating of its local data in other entities. Moreover, even if this message was modified or interrupted, any modification can be discovered easily when the values of φ_i and R_t are verified, and interruptions will not make any difference because the tag data will be updated only after receiving and verifying the last message. Thus, the reader is never affected by this and always obtains the original TID for every new session.

Resistance to man-in-the-middle attack and disclosure attacks. Man-in-the-middle attacks cannot affect SURV protocol since all exchanged messages are verified and all modifications can be simply detected. Similarly, in a disclosure attack where the attacker changes in any message sent from the reader to the tag (or vice-versa), SURV protocol will detect any change and ignore the message. Moreover, a meaningless message cannot affect the tag or the reader, but will only result in ending the current session unsuccessfully, enabling a new session to begin. Finally, to cover all possible threats to SURV, an adversary can perform DOS attacks against SURV by modifying any of the values in the message forwarded from the tag. This will either result in performing excessive TID lock-up processes in the database for invalid-TIDs when manipulating the Ψ_i or N_i values, or in the verification of a manipulated hash value(s). However, this attack will not have a significant effect because SURV uses a binary search algorithm for the TID lockup, which is fast where the lockup

complexity is $O(\log n)$. Furthermore, the database in SURV maintains the following assumptions: that all values are fixed once added, the TIDs are serialized, and the data is indexed. Therefore, the resulting complexity value of $O(1)$ for TID lockup minimizes the effect of DOS attacks. A comparison of Ultralightweight Authentication Protocol is shown in Table 1.

Table 1. Simple Comparison of Ultralightweight Authentication Protocol

	UMAP	SASI	Gossamer	SURV
Resistance to Desynchronization Attacks	No	No	No	Yes
Resistance to Disclosure Attacks	No	No	No	Yes
Privacy and Anonymity	No	No	Yes	Yes
Mutual Authentication and Forward Security	No	Yes	Yes	Yes
Total Messages for Mutual Authentication	4-5L	4L	4L	3L
Memory Size on Tag	6L	7L	7L	6L
Memory Size for each Tag on Database	6L	4L	4L	3L
Searching algorithm	Linear	Linear	Linear	Binary
Operation Types on Tag	$\oplus$, $\vee$, $\wedge$, +	$\oplus$, $\vee$, $\wedge$, +, Rot[2]	$\oplus$, +, Rot[3], MixBits	$\oplus$, +, Rot[2], MixBits

L designates the bit length of variables used.
[2] Rot(x, y) = x << wht(y), being wht(y) the Hamming weight of vector y.
[3] Rot(x, y) = x << (y mod L) for a given value of L -in our case L = 96.

5. Performance Evaluation

The performance of our protocol can be measured by determining the computational cost, storage requirements, communication cost, and security level as the main points. The SURV protocol uses simple bitwise XOR, addition mod 2^{96}, left hamming rotation, and MixBits function in the tag, and all of these operations can be simply implemented on-tag chip. In addition, we claim that SURV protocol implements strong security primitives (Ψ, φ), without adding any computational cost on the tag side. These primitives will not overload the tag with computational cost since all computational processes are performed on the backend server that has no severe limitations. Regarding storage requirements, tags that implementing SURV need to store values for Ψ, φ, A, N, k_1 and k_2, and each of these six values has a length of 96-bits, requiring a total of 576 bits of rewritable memory on the tag side. The TID and two common keys in the backend database have a total of 288 bits per tag. In terms of communication cost, the protocol in each session needs to exchange three messages, including the initial 5-byte "hello" message, four sub-messages in the second message with 384 bits, and five sub-messages in the third with 480 bits, creating a total of 904 bits from session-start until session-termination per session. The SURV protocol does not require any data updating in the database; thus, keeping the data sorted and leading to a higher search speed using binary search in retrieving the tag keys. Furthermore, SURV provides a better security level compared to UMAP family, SASI, and Gossamer, since one or more attacks have been proposed against each of these protocols, and because it also solves the desynchronization problem [34], [37] proposed against the most recent protocol, Gossamer.

6. Conclusion

We have proposed SURV as a new ultralightweight authentication protocol capable of providing transactions of shell-values that are able to transport encapsulated encrypted privacy, data between the tag and the reader without compromising the transmitted data. This protocol guarantees the privacy and anonymity of tags' holder. The main advantage of the SURV protocol is each session is considered as an atom

entity where no data from previous sessions are stored after session termination. Additionally, no data values can be updated on the tag side until all transactions have been executed and validated successfully, ensuring data integrity on the RFID tag, reader, and backend database entities at all times.

References

[1] Knospe, H., & Pohl, H. (2004). RFID security. *Information Security Technical Report*, *9(4)*, 30-41.

[2] Sarma, S., Weis, S. A., & Engels, D. W. (2003). RFID systems and security and privacy implications. *Proceedings of the 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES2002)* (pp. 454–469). Springer-Verlag: Berlin.

[3] Shih, D., Lin, C., & Lin, B. (2005). RFID tags: Privacy and security aspects. *Int'l Journal of Mobile Communications, 3,* 214-230.

[4] Weis, S., Sarma, S., Rivest, R., & Engels, D. (2003). Security and privacy aspects of low-cost radio frequency identification systems. In D. Hutter *et al.* (Eds.), *Security in Pervasive Computing* (pp. 50-59). Springer Verlag: Berlin.

[5] Kim, D., Shin, T., and Park, J. (2007). A security framework in RFID multi-domain system. *Proceedings of 2nd International Conf. on Availability, Reliability and Security* (pp. 1227-1234).

[6] Naser, M., Majali, M., Rafie, M., & Budiarto, R. (2008). A framework for RFID systems' security for human identification based on three-tier categorization model. *Proceedings of International Conference on Signal Acquisition and Processing* (pp. 103-107). Kuala Lumpur, Malaysia.

[7] Rotter, P. (2008). A framework for assessing RFID system security and privacy risks. *Pervasive Computing, 7(2)*, 70-77.

[8] Batina, L., Guajardo, J., Kerins, T., Mentens, N., Tuyls, P., & Verbauwhede, I. (2006). An elliptic curve processor suitable for RFID-tags. *Proceedings of 1st Benelux Workshop on Information and System Security* (pp. 1-14*)*. Antwerpen, Belgium.

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

[10] Feldhofer, M., *et al*. (2004). Strong authentication for RFID systems using the AES algorithm. In M. Joye & J.-J. Quisquater, (Eds.), *Cryptographic Hardware and Embedded Systems* (pp. 357-370). Berlin Heidelberg: Springer.

[11] Lee, Y. C., *et al*. (2009). A new ultralightweight RFID protocol with mutual authentication. *Proceedings of WASE International Conference on Information Engineering: Vol. 2* (pp. 58-61). Taiyuan, Shanxi.

[12] Henrici, D., & Müller, P. (2004). Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers. *Proceedings of 2nd IEEE Annual Conf. on Pervasive Computing and Communications Workshops*: *IEEE Computer Society* (pp. 149-153).

[13] Lee, S., Hwang, Y., Lee, D., & Lim, J. (2005). Efficient authentication for low-cost RFID systems. *Proceedings of Int'l Conf. of Computational Science and Its Applications: Vol.* 3480 (pp. 619-627).

[14] Tuyls, P., & Batina, L. (2006). RFID-tags for anti-counterfeiting. In D. Pointcheval, (Eds.), *Topics in Cryptology* (pp. 115-131). Berlin Heidelberg: Springer.

[15] Chien, H. (2007). SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. *IEEE Transactions on Dependable and Secure Computing*, *4*, 337-340.

[16] Burmester, M., *et al*. (2008). Robust, anonymous RFID authentication with constant key-lookup. *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security* (pp. 283-291). Tokyo, Japan, ACM.

[17] Molnar, D., *et al.* (2006). A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags. In B. Preneel & S. Tavares, (Eds.), *Selected Areas in Cryptography* (pp. 276-290). Berlin

Heidelberg: Springer.

[18] Yeo, S., & Kim, S. (2005). Scalable and flexible privacy protection scheme for RFID systems, *Security and Privacy in Ad-Hoc and Sensor Networks, 3813*, 153-163.

[19] Peris-Lopez, P., *et al*. (2006). M2AP: A minimalist mutual-authentication protocol for low-cost RFID tags. In J. Ma, H. Jin, L. Yang, & J. P. Tsai, (Eds.), *Ubiquitous Intelligence and Computing* (pp. 912-923). Berlin Heidelberg: Springer.

[20] Peris-Lopez, P., *et al.* (2006). EMAP: An efficient mutual-authentication protocol for low-cost RFID tags. In R. Meersman, Z. Tari, & P. Herrero, (Eds.), *Proceedings of OTM 2006 Workshops on the Move to Meaningful Internet Systems* (pp. 352-361). Berlin Heidelberg: Springer.

[21] Peris-Lopez, P., Hernandez-Castro, J. C., Tapiador, J. M. E., & Ribagorda, A. (2006). LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags. *Proceedings of the 2nd Workshop on RFID Security* (pp. 100–112). Graz, Austria.

[22] Li, T., & Deng, R. (2007). Vulnerability analysis of EMAP-an efficient RFID mutual authentication protocol. *Proceedings of 2nd Int'l Conf. on Availability, Reliability and Security* (pp. 238-245).

[23] Li, T., & Wang, G. (2007). Security analysis of two ultra-lightweight RFID authentication protocols. In H. Venter, M. Eloff, L. Labuschagne, J. Eloff, & R. von Solms, (Eds.), *New Approaches for Security, Privacy and Trust in Complex Environments* (pp. 109-120). Springer US.

[24] Chien, H., & Huang, C. (2007). Security of ultra-lightweight RFID authentication protocols and its improvements. *ACM SIGOPS Operating Systems Review, 41*, 83-86.

[25] Barasz, B., Ligeti, P., Loja, K., & Nagy, D. (2007). Passive attack against the M2AP mutual authentication protocol for RFID tags. *Proceedings of 1st International EURASIP Workshop on RFID Technology*.

[26] Barasz, B., Ligeti, P., Loja, K., & Nagy, D. (2007). Breaking LMAP. *Proceedings of RFIDSec* (pp. 11-16).

[27] Klimov, A., & Shamir, A. (2004). Cryptographic applications of T-functions. In M. Matsui, & R. Zuccherato, (Eds.), *Selected Areas in Cryptography* (pp. 248-261). Berlin Heidelberg: Springer.

[28] Klimov, A., & Shamir, A. (2005). New applications of T-functions in block ciphers and hash functions. In H. Gilbert, & H. Handschuh, (Eds.), *Fast Software Encryption* (pp. 18-31). Berlin Heidelberg: Springer.

[29] Hernandez-Castro, J., Tapiador, J., Peris-Lopez, P., & Quisquater, J. (2008). *Cryptanalysis of the SASI Ultralightweight RFID Authentication Protocol with Modular Rotations*.

[30] Cao, E. Bertino, T., & Lei, H. (2008). Security analysis of the SASI protocol. *IEEE Trans. on Dependable and Secure Computing, 6*, 73-77.

[31] D'Arco, P., & Santis, A. de (2008). Weaknesses in a recent ultra-lightweight RFID authentication protocol. In S. Vaudenay, (Eds.), *Progress in Cryptology — AFRICACRYPT 2008* (pp. 27-39). Berlin Heidelberg: Springer.

[32] Peris-Lopez, P., Hernandez-Castro, J., Tapiador, J., & Lubbe, J. van der (2009). *Security Flaws in a Recent Ultralightweight RFID Protocol.*

[33] Peris-Lopez, P., *et al.* (2009). Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol. In K.-I. Chung, K. Sohn, & M. Yung, (Eds.), *Information Security Applications* (pp. 56-68). Berlin Heidelberg: Springer.

[34] Yeh, K., & Lo, N. (2010). Improvement of two lightweight RFID authenti-cation protocols. *Information Assurance and Security Letters*, *1*, 6-11.

[35] Rama, N., & Suganya, R. (2010). SSL-MAP: A more secure gossamer-based mutual authentication protocol for passive RFID tags. *International Journal on Computer Science and Engineering*, *2*, 363-367.

[36] Ahmed, E., Shabaan, E., & Hashem, M. (2010). Lightweight mutual authentication protocol for low cost RFID tags. *International Journal of Network Security & Its Application, 2*, 27-38.

[37] Avoine, G. (2005). Cryptography in radio frequency identification and fair exchange protocols. PhD

thesis, Universit´e de Caen Basse-Normandie.

Muawya Naser is a faculty member in the Dept. Computer and Information Science, Khalifa City Women's Colleges, Higher Collges of Technology HCT ,UAE. He received a M.Sc. degree in computer science and a Ph.D. degree in network security. His research interests are in the field of protocols design, lightweight cryptography, cryptanalysis etc. Nowadays, his research is focused on radio frequency identification systems (RFID).

Yazan Alshamaieh is an assistant professor at the King Abdullah II School for Information Technology, University of Jordan. He received his B.Sc. degree in computer information systems from Mu'tah University, Jordan, followed by the M.Sc. degree in business information technology from Northumbria University, England in 2013. He has been awarded the Ph.D. degree in business information systems from University of Newcastle, UK.

Rahmat Budiarto received his B.Sc. degree from Bandung Institute of Technology in 1986, then received the M.Eng, and Dr.Eng degrees in computer science from Nagoya Institute of Technology in 1995 and 1998 respectively. Currently, he is a professor and the chairman of Smart Network Research Group at College of Computer Science and IT, Albaha University, Saudi Arabia. His research interests include IPv6, network security, wireless sensor networks and MANETs.

Pedro Peris-Lopez is a visiting lecturer at the Department of Computer Science, Universidad Carlos III de Madrid, Spain. He holds a M.Sc. degree in telecommunications engineering and a Ph.D. degree in computer science. His research interests are in the field of protocols design, primitives design, lightweight cryptography, cryptanalysis etc. Nowadays, his research is focused on radio frequency identification systems (RFID) and implantable medical devices (IMD). In these fields, he has published a great number of papers in specialized journals and conference proceedings.