Novelty Journals

ISSN 2394-7349

International Journal of Novel Research in Engineering and Science
Vol. 2, Issue 1, pp: (7-13), Month: March 2015 - August 2015, Available at: www.noveltyjournals.com

Analysis of a Recent Quadratic Residue Based Authentication Protocol for Low-Cost RFID Tags

Mehmet Hilal ÖZCANHAN

Computer Engineering, Dokuz Eylul University, Izmir, Turkey

Abstract: **Radio frequency identification is the hot topic of wireless security research, because RFID message exchanges through open air are attracting the attention of malicious users. The Quadratic Residue assumption supported by the Chinese Remainder Theorem (QR-CR duo) is one of the many diverse functions used for encrypting the exchanged messages. However, authentication protocols using the QR-CR duo have been facing severe analysis. The present work analyzes one of the very latest works using the QR-CR duo in a scheme, similar to protocols previously shown to have vulnerabilities. The analysis demonstrates the presence of serious vulnerabilities in the design. The consequences of the deficiencies in the scheme are important in reaching a final decision whether to recommend the proposed scheme; because the protection of the users of an authentication protocol is of great concern.**

Keywords: **Full-disclosure attack, mutual authentication, Quadratic Residue, RFID, security, tag, traceability.**

I. INTRODUCTION

RADIO Frequency Identification (RFID) is now mature and yet a growing technology [1]. The second version of Electronic Product Code (EPC) Global Class-1 Generation-2 Standard (Gen-2) for Ultra High Frequency (UHF) tags has been released, in 2013 [2]. On the other front, the High Frequency (HF) version of RFID which is named as Near Field Communication (NFC); now has readers fitted in all high-end mobile phones. The popularity of RFID is increasing due to its growing integration in ubiquitous systems. RFID is used in object identification in diverse areas such as tracking commercial goods in supply chains, managing patients and assets in hospitals, tracking inmates in prisons and transportation payment systems. Any object worth identifying or tracking is a potential RFID sticker holder. RFID identification stickers are called tags, which can be in the form of wristbands, paper stickers or plastic cards. The tags contain the vital, unique identification (ID) information; i.e. the EPC; uniquely identifying the tagged object [3]. Invariably, an RFID set up is made of a server, a reader and a tag, as shown in Fig. 1. Basically, the reader requests the ID of the tag and passes it to the server, after receiving it. The tag ID is linked to the information of the tagged object, in the server's database. Among other purposes, the ID is used with some pre-shared secrets, during the mutual authentication of the tag and the server.

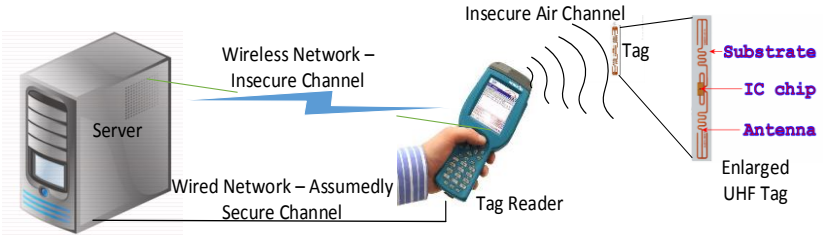

Fig. 1: A typical RFID server-reader-tag communication set-up.

The operating distance is the most important feature which differentiates RFID tags. Battery-less UHF (passive) tags are electromagnetically energized, via their antennae and can be read from a few meters. Meanwhile, the NFC tags or cards have a range of only a few centimeters. Whatever the distance, the message exchange between a tag and its reader is through air and can be eavesdropped. Therefore, the air channel is always accepted as being insecure, in all theoric work. The presence of malicious eavesdroppers, rogue (dishonest, counterfeit, unauthorized) readers and tags raise serious security issues [4,5]. The attackers (intruder, adversary) intervene, capture or block the messages to gain advantages, by exploiting the RFID technology. The detected attacks are called known attacks on RFID protocols. The need to secure the exchanged messages is obvious. Otherwise, the users of RFID face certain losses. For example, the privacy of a purchaser is violated, if the ID of the purchased item is tracked. The RFID tags are the target of attackers because they have limited computing power and resources. The dilemma of providing security over an insecure channel with limited tag resources, forces the researchers to use non-standard functions that are usually unavailable in low-cost tags. The readers however have abundant resources and the communication channel between the reader and the server is assumed to be secure. The abundant resources allow readers to contain strong cryptographic tools, which can be used for ill intentions such as launching attacks, in the hands of adversaries. In addition, the reader-server channel cannot be assumed secure everywhere, especially if it is wireless [6].

II. RELATED WORK

The limited-capacity, low-cost UHF tags cannot afford to accommodate encryption or hashing functions [7]. The only available functions are 16 bit pseudo random number generator (PRNG), a cyclic redundancy check (CRC) and an XOR function. The XOR ($\oplus$) is a commutative and associative function, widely known as the addition without carry. Many authentication protocols using XOR as a cryptographic function have been fully analyzed. The CRC function is not an encryption algorithm either; and most protocols that used it for encryption have also been broken. PRNG is the last option for encryption, in a UHF tag. But, to the best of our knowledge there is no formal proof of using a PRNG as an encryption or hashing algorithm. Besides, the PRNG function of a Gen-2 tag is deterministic and public. The weaknesses of the protocols using the above three functions can be found in works enlisted at the regularly updated web site http://www.avoine.net/rfid/index.php and some well-known attacks are referenced in work [7].

It is difficult to obscure exchanged messages without strong encryption, or hashing functions. Therefore, numerous proposals have appeared in the literature to take the challenge. In the proposals, many different algorithms have been presented as secure and suitable for low-cost tags. However, the number of attacks launched on the proposed protocols is very high, because the cryptographic algorithms used in computers are simply unavailable in low-cost tags. The dilemma has forced researchers to try alternative functions and algorithms. One example is the group of proposals using the Quadratic Residue assumption supported by the Chinese Remainder Theorem (QR-CR duo). This particular group uses the property of 'finding the square root of a number, modulo a large composite n is hard', to obscure the sensitive secrets (EPC and shared keys), inside the exchanged messages [6]. According to Quadratic Residue assumption, supposing n is the product of two large primes p and q; it is computationally infeasible to find x satisfying $y = x^2 \bmod n$, without knowing p and q, due to the difficulty of factoring n [8]. Furthermore, there are exactly four solutions x for the equation $y = x^2 \bmod n$, according to the Chinese Remainder Theorem [9]. Hence, sending a secret inside x means; only the holder of p and q can solve the message and extract the secret. A very recent protocol using the QR-CR duo method is Zhou's work [10]. The protocol is recommended for low-cost tags, because calculating $x^2 \bmod n$ is considered lightweight. Lightweight is a classification for tags with low computing power and low memory capacity [11]. In his work, Zhou summarizes some attacks [9, 12] launched against previous QR-CR duo based schemes. But then, Zhou violates some critical recommendations given in security books, by removing certain security features and primitives [13]. Our criticism finds solid proof in the Analysis of Zhou's Protocol Section.

Zhou's Proposed Protocol

Zhou has presented the scheme in Figure 2, which we name as Zhou's Protocol (ZP) for reference. In ZP, the reader initiates the mutual authentication process, by challenging the tag using timestamp r_{time} (1). The tag verifies that r_{time} is newer than its old timestamp $Time_{old}$, left from the last session. The check eliminates the replay of old sessions or wrongly timed requests. If r_{time} is good, the tag generates its first random number r_1 and sends it to the reader (2). Without waiting for a reply, the tag continues to compute an interim value m′, by XORing r_1 and the received challenge r_{time}.

Novelty Journals

ISSN 2394-7349

International Journal of Novel Research in Engineering and Science
Vol. 2, Issue 1, pp: (7-13), Month: March 2015 - August 2015, Available at: www.noveltyjournals.com

Using m′, the tag computes M′ as a preparation for the readers reply, as the value of n is public. The reader uses r_1 with r_{time} to compute m. Using m, the reader computes M and sends it to the tag (3). The tag uses M and the secret K value to calculate M′′. Next, the tag checks if M′′ equals to the previously calculated M′. If the check is good, then the reader is authenticated and the tag continues to prepare its reply. At first, a second random number r_2 and a timestamp $Time_{new}$ are generated. The tag uses the generated parameters together with the reader's r_{time} to obscure its ID_t in an interim value u. Next, the tag computes two encrypted messages T and U, by using modular arithmetic of quadratic residue. Finally, the tag sends T, U, $Time_{new}$ to the reader in message (4) and replaces old timestamp $Time_{old}$ with the new $Time_{new}$. The reader solves T and U to get eight values { r_2^1, r_2^2, r_2^3, r_2^4}, { u^1, u^2, u^3, u^4}, since it knows the factors p and q of n. By permutation of each r_2^i value against each u^i value (i=1, 2, 3, 4), the reader computes a tag ID_t and checks if it is in the database. If it is, the reader stops the computation and identifies the tag. But, completion of the tag verification requires $Time_{new}$ to be larger than r_{time}, and smaller than the validity expiration time e_{time}. If the condition is met, then the tag is verified and the mutual authentication is completed, as successful and secure.

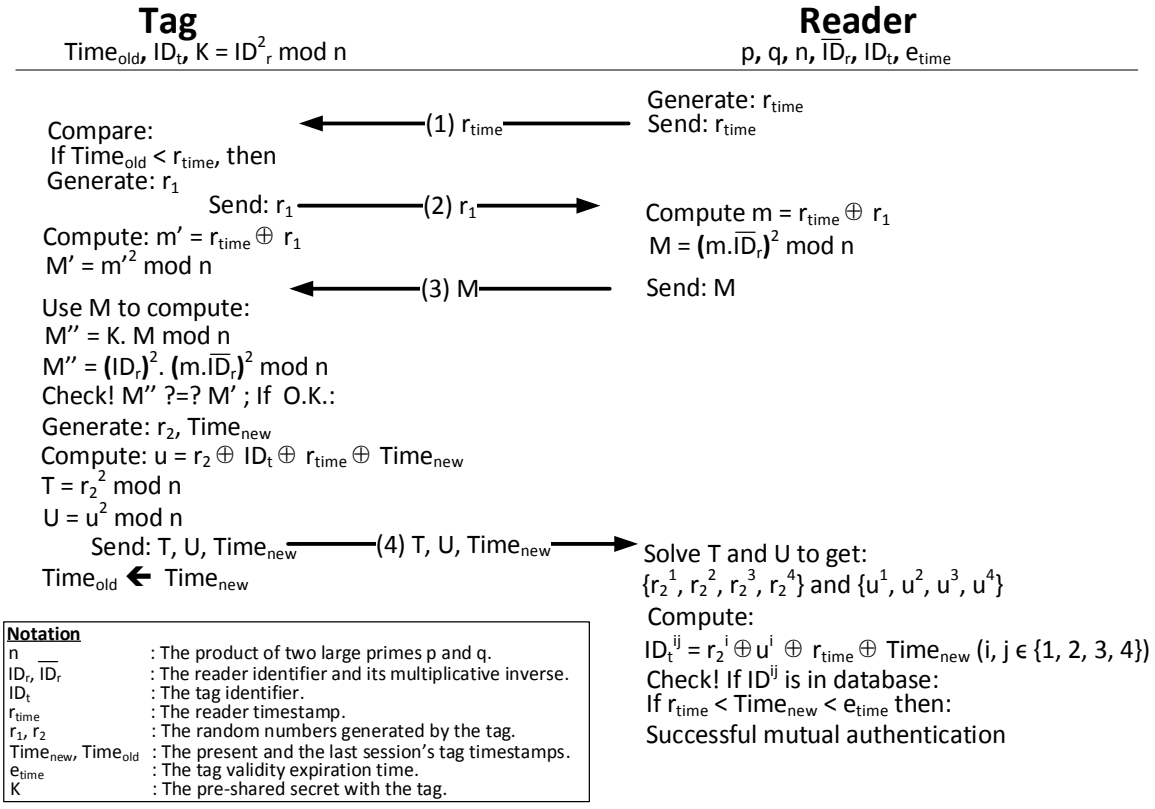

Fig.1 Zhou's proposed protocol (ZP) [10].

Zhou claims that if a tag fails authentication, the reader ignores the session. But, this is a wrong decision because, the objects with tags not passing the authentication have to be separated ("red marked") from the others that pass the authentication. Otherwise, there would be no meaning in tagging objects. Wrongly red marking an object is another problem, if the authentication failed because of poor communication due to environmental conditions.

III. ANALYSIS OF ZHOU'S PROTOCOL

Weaknesses of Zhou's Proposal:

ZP has two important weaknesses. The first and straight forward weakness is the idea of using a timestamp generated by the tag. The low computing power tags described in Zhou's work are defined as passive tags. Passive tags do not have a battery and get energized by the electromagnetic energy transferred from the reader's antenna to the tag's [7]. It is common knowledge that without a continuous power source, an electronic device cannot keep time or provide a timestamp. Therefore, relying on a low-capacity tag's timestamp is not founded. The second weakness is in the method of

tag identification by the reader. In Figure 2, the reader computes values for ID_t by at most 16 permutations among the $\{r_2^1, r_2^2, r_2^3, r_2^4\}$ and $\{u^1, u^2, u^3, u^4\}$ values. If a calculated ID_t happens to be in the database, the reader relinquishes further computation and identifies ID_t as the tag being searched for. If by sheer coincidence the result of the computation matches the ID_t of another tag, a wrong tag is identified because there is no further verification. Although the probability of a wrong identification can be small for one tag, it gets bigger as 16 possible ID_t candidates are tested each session. Assuming that a reader in a supply chain challenges 1000 tagged items 200 times/day, then approximately 16 × 200 × 1000 × 30 candidate ID_t values are computed per month. As observed, even the monthly probability of making a wrong identification is 96,000,000 ÷ number of tags in the database; a considerable probability that gets bigger by the day.

Vulnerabilities of Zhou's Proposal:

Although references to the weaknesses of previous works are made, Zhou falls into the trap of making over simplifications, in ZP. The most critical mistake is removing the encryption over some of the exchanged messages. For example, removing the hashing of the messages in works [8,9] and passing them in cleartext in exchanges (1), (2) and (4) opens a critical vulnerability in the scheme. Passing multiple messages in cleartext is not a preferred security practice. The second vulnerability is more technical. The system model of ZP is designed for low-cost tags which typically operate at 100 kHz and prepare a reply in 1800-2000 clock cycles [14]. One clock cycle's period is calculated by the formula period = 1 ÷ operating frequency. Hence, a tag has to finish its computations (2000 ÷ 100000) in less than 0,02 second. In other words, a ZP authentication finishes within a second, because there are 4 steps. Most generators increment their timestamp by one count, after every second. The claim can be easily verified by visiting an online timestamp generator. Therefore, the timestamp $Time_{new}$ is almost always predictable and equal to $r_{time} + 1$.

Reader Impersonation Attack:

Impersonation attack occurs when a malicious user poses like a legal party in an authentication protocol and manages to extract critical information about the communicating parties [12]. In ZP, the reader timestamp r_{time}, the tag random number r_1 and the tag timestamp $Time_{new}$ are sent in cleartext, in messages (1), (2) and (4) respectively. Using cleartext messages the malicious user discovers the identity of the legal reader, after eavesdropping just one session. The r_{time} in (1) is XORed with the r_1 of message (2) to obtain m. The adversary is now capable of calculating m^2 mod n, because n is public. Using message M of message (3), the adversary obtains the square of the multiplication inverse (¬) of the ID of the reader, from $(\neg ID_r)^2$ mod n = $(M \div m^2)$ mod n. From here, the adversary is able to take the multiplication inverse of the obtained value and capture the square of the ID of the reader $(IDr)^2$ mod n, which is the secret value K stored in the tag. The adversary also knows the value of $Time_{old}$ after recording it in message (4). In a reader impersonation session, the adversary fabricates r_{time} larger than $Time_{old}$, and programs a rogue reader. The presence of rogue readers is accepted as a reality in the RFID community [4,5]. When the tag replies with a new random number r_1, the rogue reader immediately calculates the new M using the obtained $(\neg ID_r)^2$ mod n and sends it to the tag. As a result, the rogue reader receives T, U and $Time_{new}$. The adversary can repeat the process and record unlimited T, U and $Time_{new}$ values. According to the tag, the authentication finishes after sending the messages in (4). Up to this point, Zhou's security analysis of the scheme is valid, because there is no information exposure apart from the reader's constant identification $(ID_r)^2$.

Tag Impersonation attack on Zhou's scheme:

Objects with tags failing the authentication have to be separated from the rest, e.g. red marked. Before red marking an object due to a mechanical or miscommunication error with its tag, re-challenging repeatedly is common. The repeated re-challenging can be exploited to expose ID_t values; since fabricating T, U and $Time_{new}$ has become possible due to previously exposed $Time_{old}$ and secret K. The first three messaging steps are easily completed by a rogue tag. When the least significant bit (LSB) of r_{time} is 0, tag impersonation takes place by choosing r_2 as 1 and $Time_{new}$, as $r_{time} + 1$. Thus in Figure 2, $u = 1 \oplus ID_t \oplus 1$; because $r_{time} \oplus Time_{new}$ is uequal to 1. In other words u is forced to become $u = ID_t$. Hence, T = 1 mod n and $U = (ID_t)^2$ mod n. The adversary uses a fabricated ID_t value from a "to be investigated list" to provide T, U and waits for the reader's attitude. An accepted ID_t is a valid tag in the database of the server and is a breakthrough. In the next tag impersonation, the number with LSB values closest to the accepted ID_t can be used. If the authentication fails, the adversary removes the ID_t from the "to be investigated list" and uses the next fabricated value to counter the re-challenge. The adversary has unlimited number of tries for exposing more ID_t values in the server database. Every tag's ID_t in the database cannot be captured in a short time, because there are many ID_t values to be investigated. But capturing many

identities is guaranteed, because the attack can be launched unnoticed over a long period and at many reader locations. Furthermore, if the captured ID_t are shared on the Internet among adversaries as in pay-tv hacking; the analysis becomes faster because brute force search list gets fewer by the day, while the exposed valid ID_t list grows.

Traceability attack:

After obtaining a substantial amount of valid ID_t values in the impersonation attacks, the previously recorded exchanges can be analyzed to escalate the attack on ZP. The first phase of tracing tags is the analysis of recorded exchanges to identify the ID_t used in the exchange. The exposed valid ID_t list is used with the cleartext r_{time} and $Time_{new}$ values of the session to obtain the u value. The analysis of the recorded sessions proceed as follows:

$$u = r_2 \oplus \Theta \quad (1)$$

$\Theta = ID_t \oplus r_{time} \oplus Time_{new}$. A value for Θ is easily calculated by using the first ID_t value from the adversary's exposed valid list. T and U becomes:

$$T = (r_2)^2 \bmod n \quad (2)$$

$$U = (r_2 \oplus \Theta)^2 \bmod n \quad (3)$$

The XOR operator is known as the addition without carry and can be approximated to addition [7]. Hence, approximating $(r_2 \oplus \Theta)$ with $(r_2 + \Theta)$ and expanding equation (3):

$$U = [(r_2)^2 + 2 \times r_2 \times \Theta + \Theta^2] \bmod n \quad (4)$$

Using equation 2 in equation 4:

$$U - T = [(2 \times r_2 \times \Theta) + \Theta^2] \bmod n \quad (5)$$

But the value of Θ is known, thus from (5):

$$U - T - (\Theta)^2 = (2 \times r_2 \times \Theta) \bmod n \quad (6)$$

Rewriting equation (6) using the strategy in [12], the value of r_2 is obtained:

$$r_2 \bmod n = [(U - T - \Theta^2) \times (n + 1)\ ] \div (2 \times \Theta) \quad (7)$$

If an integer value for r_2 cannot be obtained, the predicted ID_t from the exposed tags list is incorrect and the analysis is repeated with a new ID_t. Observe that the analysis is offline and takes only 9 computations to calculate Θ and r_2. With present day's 64 bit computers, the analysis takes little time to test each captured ID_t. Once an integer r_2 is obtained the corresponding ID_t is related to a tag and stored in a new list named "captured ID_t list". The analysis is repeated for all recorded exchanges, until the entire exposed valid ID_t list is exhausted. The analysis can continue in the future exchanges as long as ZP is used. The second phase is attempting to identify tags in the captured ID_t list at a later time; simply by going around and broadcasting r_{time} with a zero LSB, using a rogue reader. Since the tags are going to reply and finish the authentication within one second the tag's timestamp $Time_{new}$ can only be $r_{time} + 1$. Hence, the value of Θ in equation (7) becomes $\Theta = ID_t \oplus 1$. If the LSB of ID_t is 0; $ID_t \oplus 1 = ID_t + 1$. If the LSB of ID_t is 1; then $IDt \oplus 1 = IDt - 1$. Letting $ID_T = IDt \pm 1$, Θ reduces to $\Theta = ID_T$. Using the ID_t values in the adversary's list, the two values of Θ are calculated. Next, the rogue reader uses equation (7) for the tracing attack. Since existing tag ID_t values are tested, a value for r_2 is bound to be found. The location of the identified tag is noted in the adversary's database. If at another location, the same ID_t is found using the same strategy, the adversary is now capable of identifying the tag and tracing its location changes.

Full-disclosure attack:

Full-disclosure attack is exposing all of the secrets of a tag. It is especially devastating to the security of a scheme, if a dishonest tag loaded with the captured secrets can successfully authenticate with the server. In the previous three subsections, it has been demonstrated that the secrets $Time_{old}$, ID_t, K of certain ZP tags can be exposed. Programming a blank tag with the captured values allows the copied tag to mutually authenticate with the ZP server. The dishonest tag is unnoticed by the server, because only the constant ID_t is checked. As if this is not destructive enough, ZP allows duplicate tags to co-exist in the system; because the last action of ZP is tag's replacement of $Time_{old}$ with $Time_{new}$. However, the server does not synchronize with $Time_{new}$, but simply finishes with a validity date check. Thus, a dishonest or a legal tag can be authenticated in any order and there is no precaution in the scheme to notice the dual existence. However, if

$Time_{new}$ had been saved and checked against the location of the authentication, the duality could have been noticed; since the same object cannot authenticate at two different locations, within a very close time.

IV. SECURITY ANALYSIS

The Adversarial and Security Models in Zhou's work are founded on widely accepted references. However, the security analysis used to prove that ZP is secure is not adequately rigorous. According to the adversarial model an adversary can:

- •Query: interrogate tags in the system.
- •Send: Act as a tag in the system.
- •Execute: Actively monitor the channel between the tag and the reader.
- •Block: Prevent a message reaching the intended receiver.
- •Reveal: Physically tamper with the hardware of a tag and extract its secret.

It can be observed that the analysis in present work does not use the invasive reveal attack. The query, send and execute oracles suffice to dismantle the scheme. The definitions of untraceability and secure mutual authentication are also provided in the security model of ZP. In summary, untraceability of a tag is related to the probability of an adversary's capability to correctly identify a tag from another. Keeping in mind that the value of a bit can be either 0 or 1, given the probability of predicting each bit b of a tag's ID_t is Pr[b´ ?= b], (b´: adversary's guess of bit b's value, b: real b bit value); then a ZP tag cannot be traced, if the adversary has no advantage of getting the correct value of ID_t. The advantage of an adversary is given as:

$$Adv_A = 2 \times (Pr[b' ?= b] - ½) \quad (8)$$

Observe that even prediction of the bit values means Pr[b´ ?= b] = ½ which causes equation 8 to be 0; thus meaning adversary has advantage close to zero. To prove that ZP protocol is indeed traceable, the results of our traceability attack are used. Once the attacker has the ID_t, location and last authentication time of an existing tag, it can search the captured ID_t list for the entry that predicts every bit of the tag, correctly. Hence, Pr[b´ ?= b] = 1; and $Adv_A = 1$ in equation (8). Thus, the adversary has full advantage. The computation needed to test each of the bits of each captured ID_t value can be long but it is taken care by software loaded on the reader. Without going into the details, according to Zhou a proposed protocol is honest and the authentication is a Secure Mutual Authentication; if a fake tag does not succeed to authenticate as a valid tag. Unfortunately, in our full-disclosure attack it has been demonstrated that both a valid and a fake tag can co-exist on the system. Such a consequence puts ZP in the insecure category.

V. CONCLUSION

The present work is an effort to protect the users of RFID tagged objects, by pointing at the continued vulnerabilities of protocols recommended for tags. A very recent authentication protocol proposed for low computation power tags has been analyzed. The design not only uses an unfounded tag timestamp, but also can misidentify tags. Analyses presented shows that both tag and reader impersonation attacks can be launched against the protocol. Furthermore, the attacks can be escalated to the point where the protocol is fully analyzed. Although the protocol prevents the sale or use of objects with tags that have expired validity dates, the designed scheme allows both legal and dishonest tags to co-exist on the system, with the same identification number. In short, the protocol contains disadvantages and serious vulnerabilities. The detected disadvantages and vulnerabilities merge into critical security weaknesses. The security threats rise due to passing many messages in cleartext and disregarding the results of previously demonstrated security analyses. As a conclusion, the analyzed protocol is not secure to be used in low-cost tags.

REFERENCES

[1] R. Das, P. Havrop, "RFID forecasts, players and opportunities 2011–2021", IDTechEX, (2010).

[2] "EPC Radio-Frequency Identity Protocols Generation-2 UHF RFID Specification for RFID Air Interface Protocol for Comm. at 860 – 960 MHz Version 2.0.0", http://www.gs1.org/sites/default/files/docs/epc/uhfc1g2_2_0_0_standard-_20131101.pdf.

[3] M.H. Özcanhan, G. Dalkılıç and S. Utku, "Is NFC a better option instead of EPC gen-2 in safe medication of inpatients," Radio Frequency Identification, Springer Berlin Heidelberg, pp. 19-33, 2013.

[4] Y.C. Yen, N. W. Lo, T. C. Wu, "Two RFID-based solutions for secure inpatient medication administration", Journal of Medical Systems, Vol.36, No. 5, pp. 2769–2778, 2012.

[5] P.P. Lopez, A. Orfila, A. Mitrokotsa, A. and J.C.A van der Lubbe, "A comprehensive RFID solution to enhance inpatient medication safety", International Journal of Medical Information, Vol. 80, pp. 13–24, 2011.

[6] R. Doss, S. Saravanan and Wanlei Zhou, "A practical quadratic residues based scheme for authentication and privacy in mobile RFID systems", Ad Hoc Networks, Vol. 11, No. 1, pp. 383–396, 2013.

[7] M.H. Özcanhan, G. Dalkiliç and S. Utku, "Cryptographically Supported NFC Tags in Medication for Better Inpatient Safety", Journal of Medical Systems, Vol. 38, No. 8, pp. 1–15, 2014.

[8] C. Yalin, J.S. Chou and H.M Sun, "A novel mutual authentication scheme based on quadratic residues for RFID systems", Computer Networks, Vol. 52, No. 12, pp. 2373–2380, 2008.

[9] T.C. Yeh, C.H. Wu and Y.M. Tseng, "Improvement of the RFID authentication scheme based on quadratic residues", Computer Communications, Vol. 34, No. 3, pp. 337–341, 2011.

[10] J. Zhou, "A Quadratic Residue Based Lightweight RFID Mutual Authentication Protocol with Constant-Time Identification", Journal of Communications, Vol. 10, No. 2, pp. 117–123, 2015.

[11] H.Y. Chien, "SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity", Dependable and Secure Computing, IEEE Transactions, Vol. 4, No. 4, pp. 337–340, 2007.

[12] T. Cao, S. Peng and E. Bertino, "Cryptanalysis of some RFID authentication protocols", Journal of Communications, Vol. 3, No. 7, pp. 20–27, 2008.

[13] T. Stapko, the Practical Embedded Security Building Secure Resource-Constrained Systems, MA: Elsevier, ISBN: 978-0-7506-8215-2, 2008.

[14] M. Feldhofer and J. Wolkerstorfer, "Hardware implementation of symmetric algorithms for RFID security", RFID Security, Springer, NY, pp. 373–415, 2009.