A Novel Mutual Authentication Scheme Based on Quadratic Residues for RFID Systems

*Jue-Sam Chou [1], Guey-Chuen Lee [2], Chung-Ju Chan [3]

[1] Department of Information Management, Nanhua University Chiayi 622 Taiwan, R.O.C

*: corresponding author

jschou@mail.nhu.edu.tw

Tel: 886+ (0)5+272-1001 ext.56226

[2] Department of Information Management,

Central Taiwan University of Science and Technology, Taichung 406, Taiwan, R.O.C

gclee@ctust.edu.tw

[3] Department of Information Management, Nanhua University Chiayi 622 Taiwan, R.O.C

chanchungju@gmail.com

Tel: 886+ (0)5-2721001 ext.2017

Abstract

In 2004, Ari Juels [1] proposed a Yoking-Proofs protocol for RFID systems. The aim is to permit tags to generate a proof which is verifiable off-line by a trusted entity even when the readers are potentially untrusted. However, we find that their protocol not only doesn't possess the anonymity property but also suffers from both of the off-line and replay attacks. In 2006, Kirk H.M. Wong et al. [3] proposed an authentication scheme on RFID passive tags, attempting to as a standard for apparel products. Yet, to our view, their protocol suffers from the known-plaintext attack. In this paper, we first point out the weaknesses in the two above mentioned protocols. Then, we propose a novel efficient scheme which not only can achieve the mutual authentication between the server and tag but also possess the anonymity property needed in a RFID system.

***Keywords*:** mutual authentication; RFID system; binary search; quadratic residues

1. Introduction

Automatic Identification (Auto-ID) systems have become commonplace in the area of access control and security applications such as, in the industry requiring the product tracking or industry requiring the identification of products. The most widely recognized Auto-ID system is the bar code system developed during the early 1970's. Recently, Radio-Frequency Identification (RFID) systems have got intensive attention and been used in this field of automatic identification applications. It mainly consists of Radio Frequency (RF) tags (transponders) and RF tag readers (transceivers). Basically, the tag reader broadcasts a radio frequency signal to access information stored on the nearby tags. After receiving the signal, tags respond by transmitting back the resident data which typically is a unique serial number or an electronic product code (EPC) to the reader. However, as in many unsecure application systems, a RFID system may suffer from security threat as well. To get rid of those possible security problems, many cryptography scientists have proposed methods to solve the problems.

In 2003, Weis et al. [5] proposed several security schemes for a RFID system. Their schemes use of Pseudo Random Number Generator (PRNG) functions and hash functions. But all of their schemes expose the tag ID. In other words, in their system, the requirement of the tag anonymity property is violated. In the same year Ohkubo et al. [14], based on hashing chain, proposed a mutual authentication scheme for RFID systems. Their scheme aims to provide the forward secrecy; that means if an attacker can compromise a tag, he can not trace the past communications of the tag. Unfortunately, in 2004, Henrici-Mauller [11] found [14] can not resist on the reply attack. They proposed a scheme which can update the tag's ID after each successful authentication, hopping to use this varying identification to protect the location privacy and assure the anonymity of the tag. However, in 2005, Yang et al. [16, 17]

found that a tag always responds with the same hash value for the same ID in each round of the authentication phase in [11]. This property allows an attacker can trace the tag. Therefore, they improved Henrici-Mauller scheme [11] to achieve the anonymity and privacy properties.

In 2004, Molnar-Wagner's [13] proposed a mutual authentication scheme for RFID systems based on PRNG function and hash function. In 2005, Rhee et al. [15] found that their scheme [13] can not provide forward secrecy. Since once a tag is compromised, the past communications from this tag can be traced. Therefore, based on PRNG function and hash function as well, they proposed a mutual authentication scheme for RFID systems to improve Molnar-Wagner scheme [13]. Later, in the same year of 2005, Karikeyan-Nesterenko [12] found [15] can not provide forward secrecy too. Hence, they proposed a new method based on XOR and matrix operations, intending to get rid of the weakness found in [15]. However, in 2006, Duc et al.s' [10] found [12] can not resist the DOS attack, reply attack and individual tracing. Hence, they proposed a new protocol using PRNG and CRC operations to fix the problems. Yet, also in 2006, Chien et al. [2] found that [10] still can not resist on DOS attack.

In 2004, Ari Juels [1] suggested a new scheme to prevent an illegal tag from impersonating a legitimate one. However, after our analysis, we find that their protocol not only can not achieve the anonymity property but also suffers from both of the off-line attack and replay attack. In 2005, Kirk H.M. Wong et al.s, based on hash function, proposed a new simple authentication scheme for RFID systems. They claim that thir scheme is effective and secure, and can be applied to the existing apparel retail applications, especially in the point-of-sale applications (POS). However, we find that their method suffers from the off-line attack. In this paper, we will point out the weaknesses found in [1] and [3]. Then, we propose a much more secure and efficient scheme for the unsolved security problem of RFID systems

nowadays.

The organization of this article is as follows: in Section 2, we review both of Ari Juels's scheme and Kirk H.W. Wong et al.s' scheme. Also, we describe the weakness found in the two schemes. In Section 3, we present our protocol. Then, we analyze the security of our scheme in Section 4. Finally, a conclusion is given in Section 5.

2. Review of the two schemes [1, 3]

This section briefly reviews the two schemes proposed by Ari Juels [1] and Kirk H.M Wong et al. [3], respectively.

2.1 Review of Ari Juels's scheme [1]

The yoking-proof protocol assumes that the tags can perform some basic cryptographic operations. The protocol mainly consists of two methods: (1) Yoking-proof protocol using standard cryptographic primitives and (2) One-time yoking-proof using minimalist MACs. As will be described in Section 2.1.1 and Section 2.1.2, respectively.

2.1.1 Yoking-proof protocol using standard cryptographic primitives

We briefly introduce the protocol using the following steps. The processes are also delineated in figure 1.

Step1. Reader sends the left proof to tag *A*, T_A.

Step2. After receiving the left proof message, T_A uses a secure one-way hash function f to compute $r_A = f_{x_A}(c_A)$, where c_A is T_A's counter value and x_A is his secret key. Then, T_A sends the value $a = (A, c_A, r_A)$ to the reader.

Step3. After receiving value *a*, the reader sends the "*Right proof*" message and *a* to tag *B* (T_B).

Step4. After receiving *a* and "*Right proof*", T_B uses his secret key x_B to compute the HMAC of *a* and c_B, obtaining m_B, where c_B is T_B's counter value. Then, T_B sends his ID (*B*), c_B and m_B to T_A through the reader. He also adds one to the counter value c_B.

Step5. After receiving the value b= (B, c_B, m_B), T_A uses his secret key x_A to compute the HMAC of *a* and *b*, obtaining m_{AB}. He also adds one to the counter value c_A. Then, T_A sends m_{AB} to the reader. Finally, reader store the above values P_{AB}= (A, B, c_A, c_B, m_{AB}) to the database.

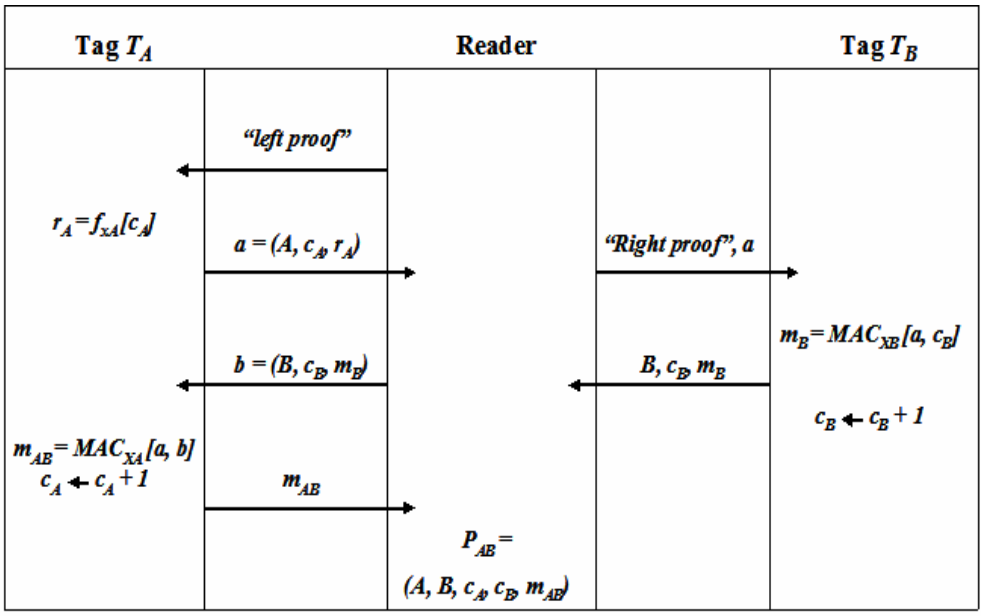

Figure 1.Yoking-proof protocol using standard cryptographic primitives

2.1.2 One-time yoking-proof using minimalist MACs

Step1. Reader sends the left proof to T_A.

Step2. After receiving the left proof message, T_A uses a secure one-way hash function *f* to compute $r_A = f_{x_A}(c_A)$, which is the HMAC of c_A, where c_A is T_A's counter value and x_A is his secret key. Then, T_A sends the value a= (A, c_A, r_A) to the reader.

Step3. After receiving value *a*, the reader sends the "right proof" message and r_a

to T_B.

Step4. After receiving the above value, T_B uses his secret key x_B to compute the HMAC of r_A, obtaining m_B. He also adds one to the counter value c_B. Then, T_B sends his ID(B), m_B and r_B to T_A through the reader.

Step5. After receiving the value r_B, T_A uses his secret key x_A to compute the HMAC, of r_B, obtaining m_A. He also adds one to the counter value c_A. Then, T_A sends m_A to the reader. Finally, reader store the above values P_{AB}= (A, B, m_A, m_B) to the database.

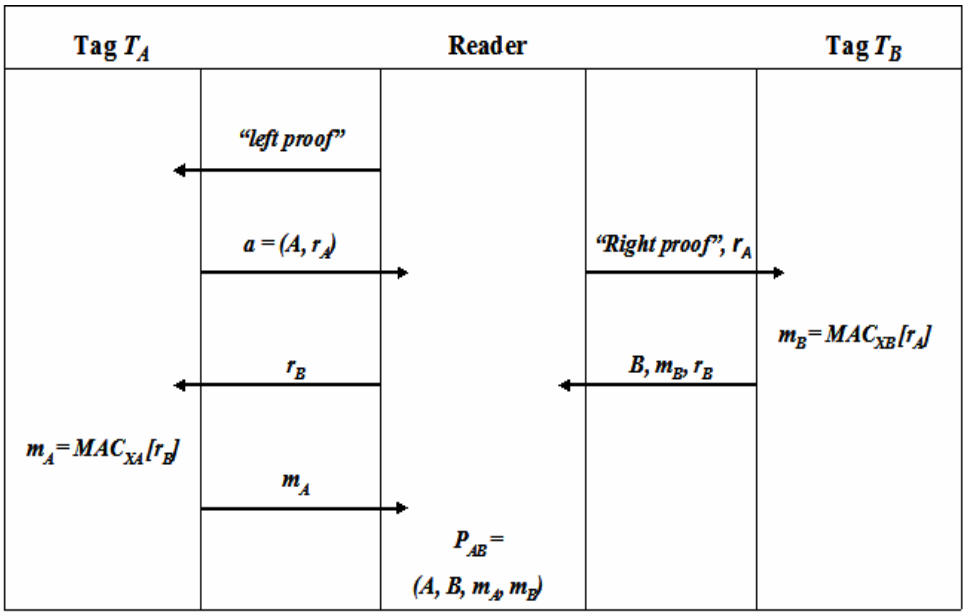

Figure 2.One-time yoking-proof using minimalist MACs

2.1.3 Cryptanalysis of Ari Juels's both methods

In protocol 1, the aim is to permit tags to generate a proof that is verifiable off-line by a trusted entity, even when the readers are potentially untrusted. However, we find that their protocol did not take anonymity into consideration. Moreover, it suffers from the off-line attack. For an adversary X can record the r_A, m_B, and r_B, m_A in the transmitted message. Whenever he has collected enough pairs of these values, that is, he has enough plaintext and ciphertext pairs, he can launch an off-line attack to

find x_A and x_B. It can succeed with a high probability by way of cooperation of computer computing through network, i.e., the collision finding of hash function MD5 is under such a computation cooperation [21].

In protocol 2, using minimalist MACs, the basic protocol architecture is unchanged. Hence, the problem as mentioned in protocol 1 remains.

2.2 Review of Kirk H.M. Wong et al.s' scheme

In this section, we first briefly review Kirk H.M. Wong et al.s' scheme. After that, we will point out the weakness in their protocol.

2.2.1 The protocol

Their scheme mainly consists of three phases: (1) the preparing phase (2) the read phase, and (3) the authentication phase. We show it as follows. The processes are also delineated in figure 3 and figure 4.

(1) Preparing phase: Initially, the server randomly chooses a value $K_{pr(i)}$ (64-bit long) as the tag's private key and computes both $key_{(i)} = EPC \oplus K_{pr(i)}$, $\text{lock}_{(i)} = hash(key_{(i)})$ and $key_{(i)}^{*} = ShiftLeft(key_{(i)}, n)$. Then, he stores $EPC_{(i)}$, $Kpr_{(i)}$ and n in the database.

(2) Read phase:

Step1. Reader sends the "Request" to $\text{Tag}_{(i)}$.

Step2. $\text{Tag}_{(i)}$ sends $key_{(i)}^{*}$ to Back end through the reader. After receiving $key_{(i)}^{*}$, Back end shifts right n (which they negotiate previously in the preparing phase) bits of $key_{(i)}^{*}$ to get the $key_{(i)}$. Then, Back end XOR his private key with $key_{(i)}$, obtaining $EPC_{(i)}$.

Step3.Back end sends "EPC Read" message to the reader.

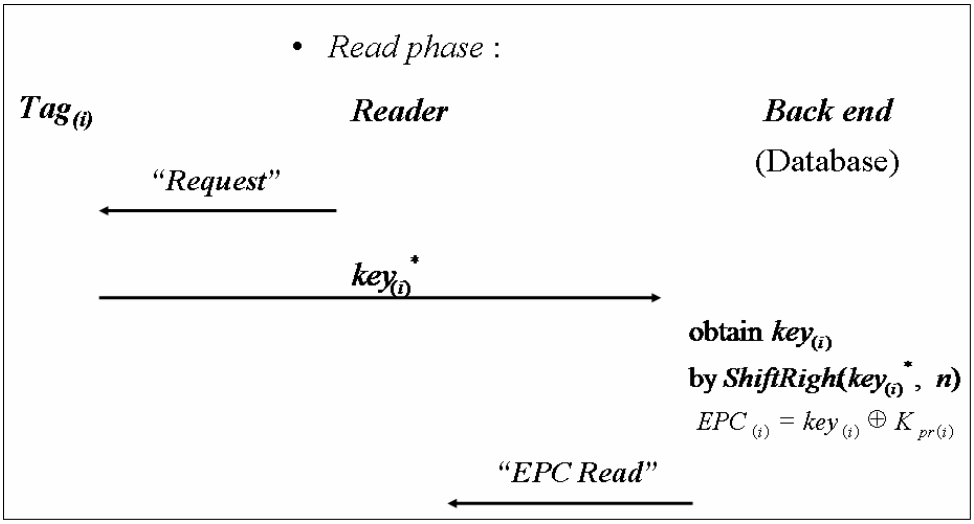

Figure 3.Read phase of Kirk H.M. Wong et al.s' scheme

(3) Authentication phase:

Step1. Back end uses a secure one-way hash function to compute $lock_{(i)}^{*} = h(key_{(i)})$. Then, he sends $lock_{(i)}^{*}$ to the reader. Reader then relays this value to $tag_{(i)}$.

Step2. After receiving $lock_{(i)}^{*}$, $tag_{(i)}$ computes hash ($key_{(i)}$) and checks to see whether the result is equal to $lock_{(i)}$. If so, he unlocks his tag memory and sends "ACK" to the reader.

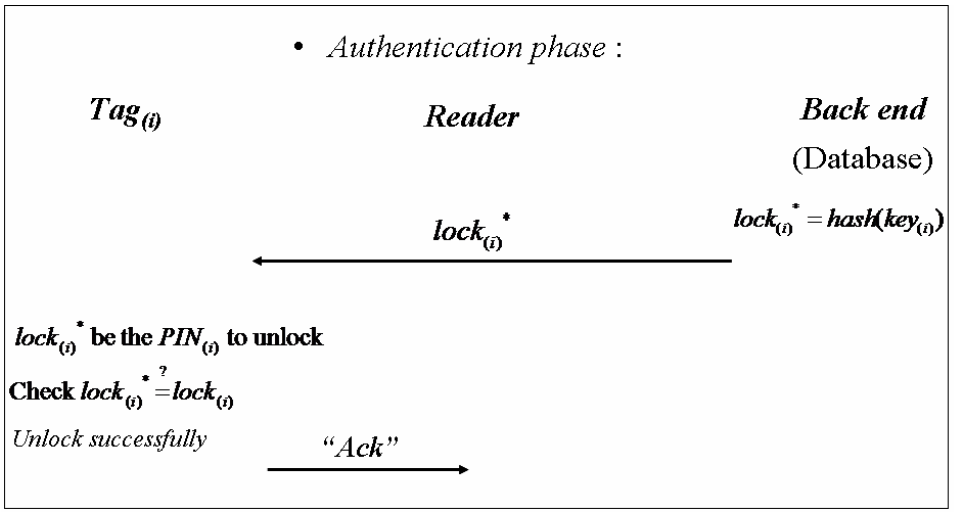

Figure 4.Authentication phase of Kirk H.M. Wong et al.s' scheme

2.2.2 Cryptanalysis of their scheme

In this section, we will show that their scheme is not secure enough. For in the read phase, the tag sends $key_{(i)}^{*}$ to the back end through the reader. We know that $key_{(i)}^{*}$ is the result of shifting left $key_{(i)}$ by n bits. Therefore, we can shift right $key_{(i)}^{*}$ by one bit a time and checks to see if the hash value of this result is equal to $lock_{(i)}^{*}$sent from back end to $tag_{(i)}$ through reader in the authentication phase. Hence, in the average case, we only need $| key_{(i)}^{*} | / 2$ times to find the right $key_{(i)}$, where $| key_{(i)}^{*} |$ denotes the length of $key_{(i)}^{*}$ in bit. Thereby, we break their scheme.

3. The proposed scheme

In this section, we present a simple protocol which not only can really achieve the security requirements of a RFID system but also can be implemented very efficiently. Moreover, our scheme has the anonymity property. We first describe the used quadratic residue theorem in Section 3.1. In Section 3.2, we define the notations in the protocol. Finally, in Section 3.3, we present our scheme.

3.1 Quadratic residue theorem

In this section, we briefly introduce the feature of quadratic residues theorem [19] first. We assume that there exists two large primes p and q such that $n = p * q$. The theorem says that if $x^2 = a \bmod n$ has a solution, then a is called a quadratic residue mod n. The symbol QR_n denotes the set of all quadratic residue numbers in [1, n-1]. It is computationally infeasible to solve x by just knowing a and n, because of the difficulty of factoring n [20].

3.2 Notations and definitions

The following notations are used throughout this paper.

- *Hello*: The message sent by the reader to query the tag.

- *a* and *b*: Denotes two quadratic residues under modulo n.
- *r*: A random number which is pre-computed in the tags.
- *n*: Denotes two large primes *p* and *q* such that $n = p * q$ which is pre-shared between the tags and server.
- *H(x)*: Denotes the hash value of x.
- *PRNG* (.): Denotes a pseudo random generator function used by the tags and server in the system.
- ID_T: The identifier of a tag.

3.3 Our scheme

Our scheme mainly consists of two phases: (1) the preparing phase (2) the read and authentication phase. We show it as follows. The processes are also delineated in figure 5.

(1) Preparing phase: Initially, the server and all the tags share two large primes *p* and *q* which satisfies $n = p * q$. Besides, the server share a random number r_k with tag_k, for $k = 1$ to k_i (number of tags). And each r_k is different. Then, the server and all tags each stores a sorted table consisting of *h(ID)* and the corresponding ID of all the tags and server using *h(ID)* as the primary sorting key in each tag memory and the server database.

(2) Read and authentication phase:

Step 1: The reader queries the tag by sending a Hello message.

Step 2: The tag computes $h(ID_T) \oplus r$ to obtain the value x. He than computes $x^2 = a \bmod n$ and $r^2 = b \bmod n$ and sends the values *a*, *b*, *h(x)*, *h(r)* to the server (back-end database) through the reader.

Step 3: After receiving *a*, *b*, *h(x)* and *h(r)*, the server uses *a*, *b* and the secret *p*, *q* to get the four possible *x* values, (x_1, x_2, x_3, x_4), and the four possible *r* values, (r_1, r_2, r_3, r_4). Then, he compares $h(x_i)$ and $h(r_i)$ with *h(x)* and *h(r)* for i =1 to 4 respectively to get correct *x* and *r*. The server then compares the hash value of each x_i to *h(x)* and the hash value of each r_i to *h(r)*. He then can get the right *x* and *r*.

Step 4: After obtaining the values of *x* and *r*, the server can get $h(ID_T)$ by computing $x \oplus r$ and then using binary search to seek for the tag ID_T using $h(ID_T)$ as the searching key in the sorted table.

Step 5: Server computes the new value of r_{new} by using the PRNG function. He then computes $x = h(ID_T) \oplus r_{new}$ and $b = r_{new}^2 \bmod n$. Then, server sends the values (b, $h(x)$, $h(r_{new})$) to the tag.

Step 6: After receiving these values, tag solves the equation $b = r_{new} \bmod n$, obtaining r_i for i = 1 to 4. Then he compares each $h(r_i)$ with the transmitted $h(r_{new})$ to get the correct r_{new}. After that, tag XOR $h(ID_T)$ and r_{new} to get *x*, and checks to see if *h(x')* is equal to the transmitted *h(x)*. If so, he updates the r_{new} value in his tag memory.

4. Security analysis

In this section, we analyze our protocol and prove that it is anonymous and secure using the following lemmas.

analysis 1. *The proposed protocol is anonymous*

In our scheme, only the parameters (*a*, *b*, *h(x)*, *h(r)*) are transmitted between the reader and the tag. The identification information is wrapping into *x* and then *a* which is a quadratic residue modulo n and can not be solved in polynomial time under the

difficulty of factoring n. Therefore, the anonymity property is assured.

analysis 2. *The protocol can resist the replay attack*

For in our scheme, the tag and server pre-shared a nonce *r*. It is updated after each successful authentication. If an attacker uses an old *h(r)*, he will be found. Hence, our protocol can resist the replay attack.

analysis 3. *The proposed protocol can achieve mutual authentication between the tag and server*

Because only the real server knows the identification of the tag, in step 4, the server can get $h(ID_T)$ by computing $x \oplus r$. He can then find ID_T by searching the sorted table stored in his database. In step 6, the tag XOR $h(ID_T)$ with r_{new} to get x' and then checks the equality of $h(x')$ with the transmitted $h(x)$. If the equality holds, the mutual authentication is achieved.

Table 1.Comparisons among related security schemes for RFID

	Anonymity	Resist to replay attack	Resistance to DOS attack	Forward secrecy	Efficiency
Weis et al. [5]	×	×	o	×	×
Ohkubo et al. [14]	o	×	o	o	×
Henrici-Mauller [11]	×	×	o	×	×
Rhee et al. [15]	o	o	o	×	×
Molner-Wagner [13]	o	o	o	×	×
Yang et al. [16][17]	×	o	o	×	×
Karthikeyan-Nesterenko [12]	×	×	×	×	×
Duc et al. [10]	o	×	×	×	×
Chien [18]	o	o	o	o	×
Ari Juels [1]	×	×	×	o	×
Kirk H.M. Wong et al. [3]	×	o	o	o	o
Our scheme	o	o	o	o	o

o: represents the corresponding scheme possessing the relative property listed in the first row
×: represents the corresponding scheme which does not possesses the relative property listed in the first row

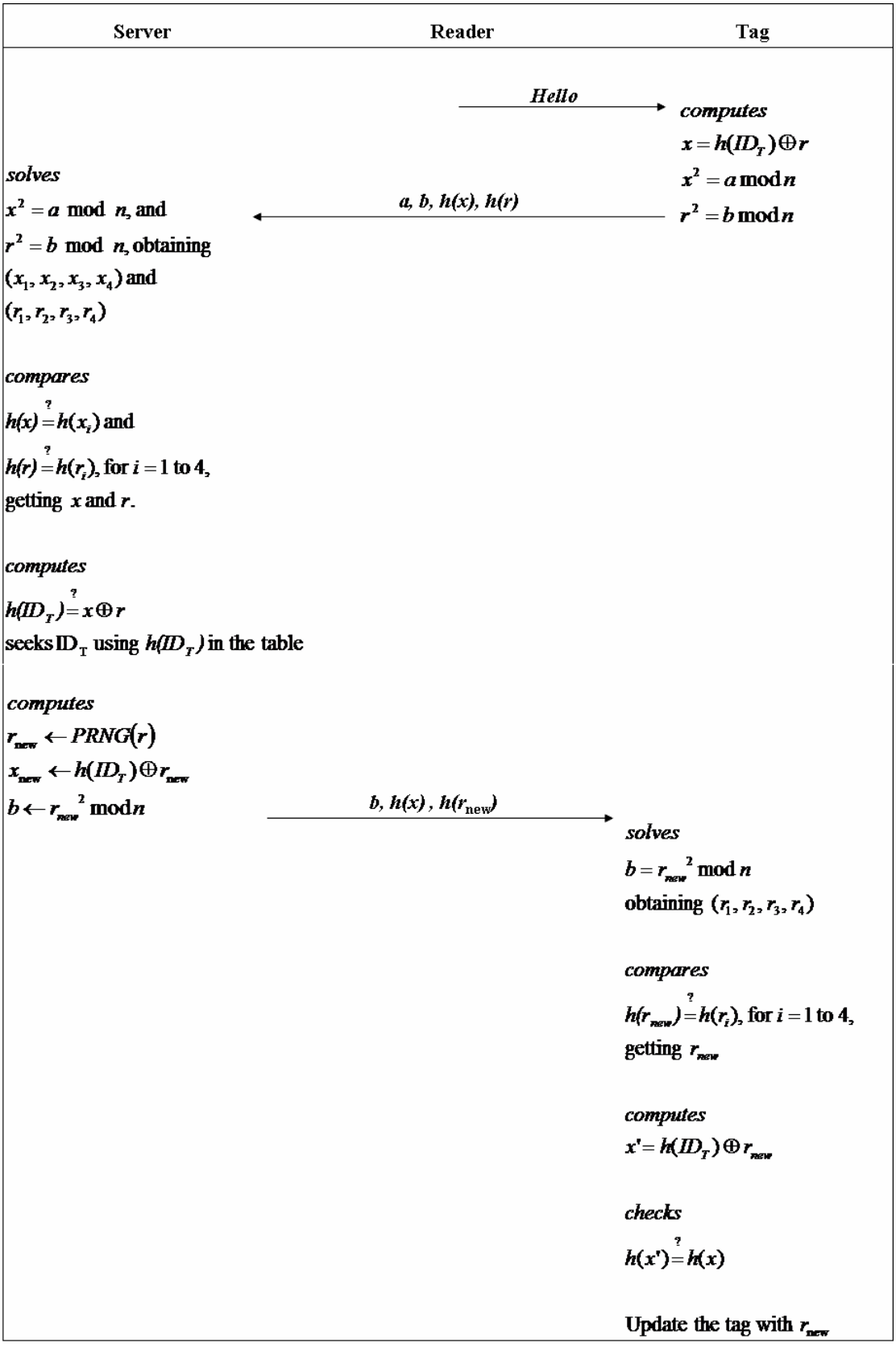

Figure 5.Our proposed scheme using quadratic residues technology

5. Conclusion

There has been several security schemes proposed for RFID systems but only few of them can authenticate each other between the server and the corresponding tag. In this paper, we demonstrated that Ari Juels's scheme [1] is vulnerable to off-line attack. We also found Kirk H.M. Wong et als' scheme [3] is quite easy to be broken. Finally, we present a new mutual authentication RFID scheme using quadratic residues. After our analysis, we can conclude that our scheme is not only very efficient but also much more secure than all of the already proposed schemes.

References

[1] A. Juels, "Yoking-Proofs for RFID Tags, "Proc. IEEE Int. Conf. Digital object identifier, 2004, pp. 138-143.

[2] H.Y Chien, C.H. Chen, "Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards, "Computer standards & Interfaces, 2006.

[3] Kirk H.M. Wong, Patrick C.L. Hui, Allan C.K. Chan, "Cryptography and authentication on RFID passive tags for apparel products, "Computer in Industry 57, 2005, pp. 342-349.

[4] S. Sarma, S. Weis, D. Engels, "RFID System, Security & Privacy Implications, "White paper, MIT Auto-ID Center, November 2002.

[5] S.A. Weis, S.E. Sarma, R.L. Rivest, D.W. Engels, "Security & Privacy Aspects of Low-Cost Radio Frequency Identification Systems, "Security in Pervasive Computing 2003, LNCS no. 2802, 2004, pp. 201-212.

[6] EPCglobal web site, http://www.epcglobalinc.org/

[7] Y.H. Ham, N.S. Kim, C.S. Pyo, J.W. Chung, "A Study on Establishment of Secure RFID Network Using DNS Security Extension, "Asia-Pacific IEEE Int. Conf. Communications, 2005, pp. 525-529.

[8] G. Tsudik, “YA-TRAP: Yet Another Trivial RFID Authentication Protocol, ”Fourth Annual IEEE Int. Conf. Digital object identifier, 2006, pp. 4.

[9] J. Ayoade, “Security implications in RFID and authentication processing framework, “Computers & Security 25, 2006, pp. 207-212.

[10] D.N. Duc, J. Park, H. Lee, K. Kim, “Enhancing security of EPCglobal Gen-2 RFID tag against traceability and cloning, ”The 2006 Symposium on Cryptography and Information Security, 2006.

[11] A.D. Henrici, P. Mauller, “Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers, ”PerSec’04 at IEEE PerCom, 2004, pp. 149-153.

[12] S. Karthikeyan, M. Nesterenko, “RFID security without extensive cryptography, ”Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks, 2005, pp. 63-67.

[13] D. Molnar, D. Wagner, “Privacy and security in library RFID: issues, practices, and architectures, ”Conference on Computer and Communications Security CCS’04, 2004, pp. 210-219.

[14] M. Ohkubo, K. Suzki, S. Kinoshita, “Cryptographic approach to privacy-friendly tags, ”RFID Privacy Workshop, 2003.

[15] K. Rhee, J. Kwak, S. Kim, D. Won, “Challenge-response based RFID authentication protocol for distributed database environment, ”International Conference on Security in Pervasive Computing SPC 2005, 2005, pp. 70-84.

[16] J. Yang, J. Park, H. Lee, K. Ren, K. Kim, “Mutual authentication protocol for low-cost RFID, ”Handout of the Encrypt Workshop on RFID and Lightweight Crypto, 2005.

[17] J. Yang, K. Ren, K. Kim, “Security and privacy on authentication protocol for low-cost radio, ”The 2005 Symposium on Cryptography and Information

Security.

[18] H.Y. Chien, “Secure access control schemes for RFID systems with anonymity, ”Proc. Int. Workshop on Future Mobile and Ubiquitous Information Technologies, 2006.

[19] K.H. Rosen, *Elementary Number Theory and Its Applications*, Addison-Wesley, Reading, MA (1988).

[20] W. Patterson, *Mathematical Cryptology for Computer Scientists and Mathematicians*, Rowman, 1987.

[21] Eric Thompson, “MD5 collisions and the impact on computer forensics, “Digital investigation 2005, pp. 36-40.

[22] J. Saito, K. Sakurai, “Grouping Proof for RFID Tags, ”Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA’05), 2005, pp. 621-624.