Security Analysis of a PUF based RFID Authentication Protocol

Masoumeh Safkhani[1], Nasour Bagheri[2] and Majid Naderi[1]

[1] Electrical Engineering Department, Iran University of Science and Technology, Tehran, Iran. {M_Safkhani,M_Naderi}@iust.ac.ir
[2] Electrical Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran. Nbagheri@srttu.ac.ir

Abstract. In this paper we consider the security of a PUF based RFID Authentication protocol which has been recently proposed by Bassil *et al.* [2]. The designers have claimed that their protocol offers immunity against a broad range of attacks while it provides excellent performance. However, we prove in contrary to its designers claim, this protocol does not provide any security. We present an efficient secret disclosure attack which retrieves all secret parameters of the protocol. Given those secret parameters, it would be trivial to apply any other attack in the context on the protocol. However, to highlight other weaknesses of the protocol we present extra reader traceability, impersonation and desynchronization attacks that do not require disclosing the secret parameters necessarily. Success probability of all mentioned attacks is almost "1" while the complexity is at most two runs of protocol.

Keywords: RFID, Authentication, PUF, Traceability Attack, Reader Impersonation Attack, Tag impersonation Attack, Desynchronization Attack.

1 Introduction

Radio Frequency Identification systems is a wireless system which uses radio frequency to identify objects, animals, human and so on. Crescive spread of RFID leads security and privacy problems to become propounded. To address the mentioned issues and to help the admission of this technology, a lot of RFID authentication protocols have been proposed already in the literatures [3, 5–8, 12, 13, 16, 17, 19–21, 23–25, 29–36]. To address the resources constraint restrictions of RFID systems several researchers have tried to provide ultralightweight protocol to be employed in the RFID applications. This class of RFID protocols only use limited number of ultralightweight operations in their construction, e.g. bit wise AND, OR, XOR and Rotate. Examples of these protocols are

[12,16,18,20] and [24]. However, the later analysis demonstrated that it is not any easy task to design a secure protocol in this way [1,4,11,22,26–28]. On the other hand, some recent works attempted to employee Physically Unclonable Functions (PUF) to design ultralightweight authentication protocols [2, 14, 15].

A Physically Unclonable Function (PUF), is a piece of hardware that produces a signature, either based on the unique characteristics of a particular instance alone, or in concert with a user defined input. Several different types of PUFs exist [10]. Common to all solutions is that they rely on the variation of delays in wires and gates that exist in all electronic devices. Furthermore, despite efforts to reduce this normally unwelcome feature, delays seem to increase with newer technology as IC designs are becoming smaller [9]. The reason why PUFs are so attractive in the security field is not only that they are cheap to implement, both monetarily and in hardware, they are also hard for an attacker to tamper with. If the attacker tries to evaluate the PUF or IC, e.g. using probes to measure wire delays, the characteristics of that particular PUF will be changed (perhaps forever), therefore it will not give the information that the attacker expected.

1.1 Overview of the Current Work

In this paper we consider the security of a PUF based RFID Authentication protocol which has been recently proposed by Bassil, El-Beaino, Kayssi and Chehab [2] which we denote it in short by BEKC protocol henceforth. They have claimed that the designed protocol resists the known attacks despite of its excellent performance. However, in this paper we demonstrate that BEKC protocol does not provide resistance against secret disclosure attack, traceability attack, tag and reader impersonation attack and desynchronization attack.

Paper Organization : The notations used in the paper are presented in Section 2. BEKC protocol is described in Section 3. In Section 4, Section 5, Section 6 and Section 7 we present our secret disclosure attack, traceability attack, reader impersonation attack and desynchronization attack respectively. Finally, we conclude the paper in Section 8.

2 Preliminaries

Throughout the paper, we use the following notations:

- R_i: RFID reader i.
- T_i: RFID tag i.
- PUF: Physically Uncloneable Function.
- SVT_i: The 96 bits secret value of T_i which is generated by a PUF embedded in T_i.
- SVR_i: The 96 bits secret value of R_i which is generated by a PUF embedded in T_i.
- $Rotl(x, y)$: is a circular shift on the value of x by (y mod 96) to the left.
- $Rotr(x, y)$: is a circular shift on the value of x by (y mod 96) to the right.
- n_1 and n_2 : Two 96 bits random numbers generated by the reader.
- $\oplus$: The XOR operation.
- $+$: The bitwise OR operation.
- $A \leftarrow B$: Refers to assigning B to A.

3 Description of BEKC Protocol

In BEKC protocol, a PUF is embedded inside each tag to produce the secret value of the tag, SVT_i, which is computed as $SVT_i = PUF(challenge)$, where *challenge* is provided from an external source during an initialization phase before deploying the tags. Hence, due to the nature of the PUF function, each tag T_i will have a different secret value. Moreover, another secret value related to the reader, SVR_i, is stored in the tag which is computed as $SVR_i = PUF(SVT_i)$. Therefore, each tag has a unique pair of SVT_i and SVR_i stored in it initially. In addition, this pair is also stored in the back-end database. In BEKC protocol, which is depicted in Fig. 1, a reader R_i and a tag T_i authenticates each other as follows:

1. R_i sends the "Hello" message to T_i.
2. On receiving the message, T_i responds with its SVT_i.
3. Once R_i receipt the message, it will search for the entry corresponding to SVT_i in the back-end database. If there is no record for the SVT_i in the back-end database, a new request is sent by R_i to T_i, however, this time T_i replies with the old un-updated SVT_i to consider possible desynchronization between the reader and the tag. But if R_i finds a record for SVT_i in the back-end database, it generates two 96-bit random numbers n_1 and n_2, computes $A = SVT_i \oplus SVR_i \oplus n_1$, $B = Rotl(SVR_i + n_2, SVT_i)$ and $C = Rotl(SVT_i \oplus SVR_i \oplus n_1, n_2)$ and sends $A\|B\|C$ to T_i.
4. Once T_i receipt the message, it employees A and B to extract the random numbers n_1 and n_2. Then, it computes $C' = Rotl(SVT_i \oplus$

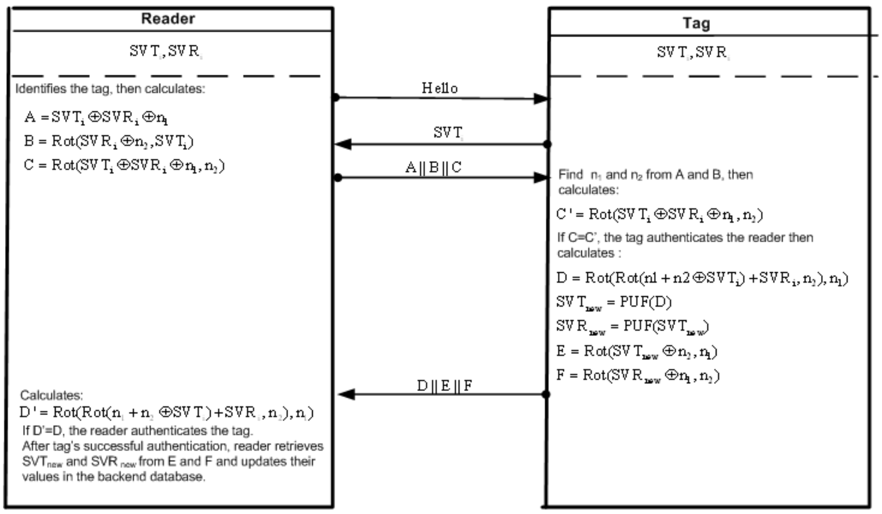

Fig. 1. BEKC Protocol.

$SVR_i \oplus n_1, n_2)$ and compares it with the received C. If $C \neq C'$ the tag will stop the authentication procedure, otherwise the reader is authenticated. Next, the tag computes D, E and F and updates SVT_i and SVR_i and sends $D\|E\|F$ to R_i. To calculate D, E and F and update SVT_i and SVR_i, the tag does as follows:

$$D = Rotl(Rotl(n_1 + n_2 \oplus SVT_i) + SVR_i, n_2), n_1)$$

$$SVT_{new} = PUF(D)$$

$$SVR_{new} = PUF(SVT_{new})$$

$$E = Rotl(SVT_{new} \oplus n_2, n_1)$$

$$F = Rotl(SVR_{new} \oplus n_1, n_2)$$

5. On receiving the message, R_i computes $D' = Rotl(Rotl(n_1 + n_2 \oplus SVT_i) + SVR_i, n_2), n_1)$ and compares it with the received D to make sure that the tag was able to retrieve the correct random numbers n_1 and n_2. If $D' = D$, R_i authenticates T_i as a legitimate tag, retrieves SVT_{new} and SVR_{new} from E and F and updates their related records in the back-end database.

4 Secret Disclosure Attack

In this section, we present an efficient secret disclosure attack which leads to disclose all secret values of the reader and the tag that participate in BEKC protocol. To disclose the secret values, the adversary $\mathcal{A}$ can do as follows:

1. $\mathcal{A}$ eavesdrops one successful run of protocol between tag T_i and legitimate reader R_i and stores the transferred values include SVT_i, A, B,C,D,E and F where :

$$A = SVT_i \oplus SVR_i \oplus n_1 \tag{1}$$
$$B = Rotl(SVR_i + n_2, SVT_i) \tag{2}$$
$$C = Rotl(SVT_i \oplus SVR_i \oplus n_1, n_2) \tag{3}$$
$$D = Rotl((Rotl(n_1 + n_2 \oplus SVT_i) + SVR_i, n_2), n_1) \tag{4}$$
$$SVT_{new} = PUF(D) \tag{5}$$
$$SVR_{new} = PUF(SVT_{new}) \tag{6}$$
$$E = Rotl(SVT_{new} \oplus n_2, n_1) \tag{7}$$
$$F = Rotl(SVR_{new} \oplus n_1, n_2) \tag{8}$$

2. $\mathcal{A}$ sends a "Hello" message to T_i and T_i responds with its current SVT which is SVT_{new} in the above equations.
3. $\mathcal{A}$ does the following computations:
   (a) $\forall\ i = 0\ \ldots\ 95$:
      i. If $Rotr(C, i) = A$ then returns i as $n_2\ mode\ 96$.
   (b) $\forall\ i = 0\ \ldots\ 95$:
      i. $n_1\ mode\ 96 \leftarrow i$
      ii. $n_2' \leftarrow (Rotr(E, n_1)) \oplus SVT_{new}$
      iii. If $n_2'\ mode\ 96 = n_2\ mode\ 96$ then returns n_2' as n_2.
   (c) Given B, C and SVT_i from the eavesdropping phase of attack (Step 1) and n_2 from step 3(b)iii, to eavesdrop SVR_i and n_1, $\mathcal{A}$ does as follows:
      i. $SVR_i \leftarrow (Rotr(B, SVT_i)) - n_2$
      ii. $n_1 \leftarrow (Rotr(C, n_2)) \oplus SVR_i \oplus SVT_i$
4. To confirm the correctness of the retrieved parameters, the returned n_2 from Step 3(b)iii and the returned n_1 and SVR_i from Step 3c, $\mathcal{A}$ verifies whether $D \stackrel{?}{=} Rotl((Rotl(n_1 + n_2 \oplus SVT_i) + SVR_i, n_2), n_1)$. If the verification is passed $\mathcal{A}$ can retries SVR_{new} as $Rotr(F, n_2) \oplus n_1$; otherwise it returnees to Step 3a and continue with the remaining values of i.

An attacker which follows the above attack would be able to disclose all secret values involved in the protocol, i.e., n_1, n_2, $(SVR_i)_{old}$, $(SVR_i)_{new}$, $(SVR_i)_{old}$ and $(SVR_i)_{new}$. The success probability of our secret disclosure attack is "1" and the complexity is only two runs of protocol. It must be noted that at the end of the attack the current record of the back-end database for T_i is $(SVR_i)_{new}$ and $(SVT_i)_{new}$ and T_i holds $(SVR_i)_{old}$, $(SVT_i)_{old}$, $(SVR_i)_{new}$, and $(SVT_i)_{new}$.

It must be noted that since the adversary knows all secret parameters then it can easily do what attack it wants. For example it can impersonate the tag, impersonate the reader, desynchronize the tag and the reader, trace the tag and etc. However, to show other weaknesses of the protocol we present other attacks against the protocol in the rest of the paper.

5 Traceability Attack

BECK protocol's designers have claimed that their protocol provides tag's location privacy. They have stated since each tag have a unique PUF which is used to update SVT_i and SVR_i and the reader uses two random numbers n_1 and n_2 in its responses, the tag responses in different run of protocol can not be linked and it is not possible to trace the tag. However, it is easy that as far as the tag has not updated its SVT_i and SVR_i it will return a same SVT_i as response to the "Hello" command sent by the legitimate reader or the adversary. This property is enough to trace the tag between two successful runs of protocol. In addition, following the protocol decryption, tag keeps a record of $(SVR_i)_{old}$ and $(SVT_i)_{old}$ and if the reader repeats the "Hello" command, tag will reply with $(SVT_i)_{old}$ as a response to the second "Hello" sent by the reader. Hence, even after one successful run of the protocol the adversary would be able to trace the tag.

The adversary will fail in its attack if two tags comes up with the same SVT_i. Since SVT_i is assumed to be random, the success probability of the given traceability attacks are not less than $(1-2^{-96})^2$ while the complexities are at most two runs of protocol.

6 Reader Impersonation Attack

Bassil *et al.* claimed that their protocol resists against reader impersonation attack. They have stated since only the tag and the legitimate reader have the correct mappings between SVT_i and SVR_i thus an impersonating reader will not be authenticated by the legitimate tag. However, in this section, we prove that this claim unfortunately does not hold. To impersonate the reader, an adversary $\mathcal{A}$ can do as follows:

1. **(Phase 1: Learning)** $\mathcal{A}$ eavesdrops one successful run of protocol and stores SVT_i, A, B,C,D, E and F.
2. **(Phase 2: Impersonation)** To impersonate R_i, $\mathcal{A}$ does as follows:
   (a) $\mathcal{A}$ supplants R_i and starts another session of protocol and sends a "Hello" command to the tag and the tag responds with its updated value of SVT_i, $(SVT_i)_{new}$.
   (b) $\mathcal{A}$ sends again "Hello" command to the tag and the tag responds with its old value of SVT_i, SVT_i.
   (c) $\mathcal{A}$ sends the eavesdropped $A\|B\|C$ to T_i.
   (d) Once T_i receipt the message, it employees A and B to extract the random numbers n_1 and n_2. Then, it computes $C' = Rotl(SVT_i \oplus SVR_i \oplus n_1, n_2))$ and compares it with the received C. Since $C = C'$ the tag authenticates $\mathcal{A}$ as a legitimate reader.

The success probability of our reader impersonation attack is "1" and the complexity is only two runs of BECK protocol.

7 Desynchronization Attack

BECK protocol's designers have claimed that in their protocol desynchronization problem can be overcame by storing two sequential versions of SVT_i and SVR_i at the tag, one pair before the updating and the other after the updating. In addition, they have mentioned that an explicit ACK may be sent by the reader to confirm the updating stage. However we show that their protocol suffers from explicit desynchronization attack. The given desynchronization attack is based on this fact that when in the last step of the protocol T_i sends $D\|E\|F$ to R_i, R_i verifies the correctness of D to authenticate the tag and if D passes the verification successfully it also accept the received E and F to extract the new recodes of SVT_i and SVR_i to be stored in the database. However, an active adversary can manipulate the transferred $D\|E\|F$ and replace it by $D\|E'\|F'$ for some $E' \neq E$ and $F' \neq F$. Therefore, the reader retrieves different values for SVT_i and SVR_i compared to the records of SVT_i and SVR_i in the tag. Hence, the tag and the reader would be desynchronized at the end of the attack with the probability of almost "1". The complexity of the given attack is only one run of the protocol.

8 Conclusion

In this paper we investigated the security of a PUF based RFID authentication protocol which recently has been proposed by Bassil *et al.* [2]. We have shown that, in contrary to its designers' claims, this protocol does not provide resistance against secret disclosure attack, traceability attack, reader impersonation attack and desynchronization attacks. Success probability of all mentioned attacks is almost "1" while the complexity is at most two runs of protocol. This result shows that designing a secure lightweight protocol is not an easy task.

References

1. N. Bagheri, M. Safkhani, M. Naderi, and S. K. Sanadhya. Security Analysis of $LMAP^{++}$, an RFID Authentication Protocol. Cryptology ePrint Archive, Report 2011/193, 2011. `http://eprint.iacr.org/`.
2. R. Bassil, W. El-Beaino, A. Kayssi, and A. Chehab. A PUF-Based Ultra-Lightweight Mutual-Authentication RFID Protocol. In *6th International Conference on Internet Technology and Secured Transactions, 11–14 December 2011, Abu Dhabi, United Arab Emirates*, pages 495–499, 2011.
3. M. Burmester, B. de Medeiros, J. Munilla, and A. Peinado. Secure EPC gen2 compliant radio frequency identification. In P. M. Ruiz and J. J. Garcia-Luna-Aceves, editors, *ADHOC-NOW*, volume 5793 of *Lecture Notes in Computer Science*, pages 227–240. Springer, 2009.
4. T. Cao, E. Bertino, and H. Lei. Security Analysis of the SASI Protocol. *IEEE Transactions on Dependable and secure Computing*, 6(1):73–77, 2009.
5. Y.-Y. Chen, M.-L. Tsai, and J.-K. Jan. The design of RFID access control protocol using the strategy of indefinite-index and challenge -response. *Computer Communication*, 34(3):250–256, 2011.
6. H.-Y. Chien and C.-H. Chen. Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards. *Computer Standards & Interfaces*, 29(2):254–259, 2007.
7. J.-S. Cho, S.-S. Yeo, and S. K. Kim. Securing against brute-force attack: A hash-based RFID mutual authentication protocol using a secret value. volume 34, pages 391–397, 2011.
8. E. Y. Choi, D. H. Lee, and J. I. Lim. Anti-cloning protocol suitable to EPCglobal class-1 generation-2 RFID systems. *Computer Standards & Interfaces*, 31(6):1124–1130, 2009.
9. K. K. D. Chinnery. Closing the gap between asic and custom: An asic perspective. In *in Design Automation Conference*, pages 637–642, 2000.
10. S. D. G. E. Suh. Physical unclonable functions for device authentication and secret key generation. In *in Design Automation Conference - DAC 2007*, pages 495–499, 2007.
11. J. C. Hernandez-Castro, J. M. E. Tapiador, P. Peris-Lopez, and J.-J. Quisquater. Cryptanalysis of the SASI Ultralightweight RFID Authentication Protocol with Modular Rotations. Technical Report arXiv:0811.4257, Nov 2008.

12. C. Hung-Yu. SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity. *IEEE Transactions on Dependable and Secure Computing*, 4(4):337–340, December 2007.
13. G. Jin, E. Y. Jeong, H.-Y. Jung, and K. D. Lee. RFID authentication protocol conforming to EPC class-1 generation-2 standard. In H. R. Arabnia and K. Daimi, editors, *Security and Management*, pages 227–231. CSREA Press, 2009.
14. L. Kulseng, Z. Yu, Y. Wei, and Y. Guan. Lightweight Mutual Authentication and Ownership Transfer for RFID systems. In *The proceedings of IEEE INFOCOM 2010*, pages 1–5, March 2010.
15. Y. S. Lee, Y. Park, S. Lee, T. Kim, and H. J. Lee. Rfid mutual authentication protocol with unclonable rfid-tags. In *ICMIC*, pages 74–77, 2011.
16. T. Li. Employing Lightweight Primitives on Low-Cost RFID Tags for Authentication. In *VTC Fall*, pages 1–5. IEEE, 2008.
17. N.-W. Lo and K.-H. Yeh. An efficient mutual authentication scheme for EPCglobal class-1 generation-2 RFID system. In M. K. Denko, C.-S. Shih, K.-C. Li, S.-L. Tsao, Q.-A. Zeng, S.-H. Park, Y.-B. Ko, S.-H. Hung, and J. H. Park, editors, *EUC Workshops*, volume 4809 of *Lecture Notes in Computer Science*, pages 43–56. Springer, 2007.
18. P. Peris-Lopez, J. C. H. Castro, J. M. Estévez-Tapiador, and A. Ribagorda. LMAP:A Real Lightweight Mutual Authentication Protocol for Low cost RFID tags. In *RFIDSec*, 2006.
19. P. Peris-Lopez, J. C. H. Castro, J. M. Estévez-Tapiador, and A. Ribagorda. M^2ap: A minimalist mutual-authentication protocol for low-cost rfid tags. In J. Ma, H. Jin, L. T. Yang, and J. J. P. Tsai, editors, *Ubiquitous Intelligence and Computing, Third International Conference*, volume 4159 of *Lecture Notes in Computer Science*, pages 912–923. Springer, 2006.
20. P. Peris-Lopez, J. C. H. Castro, J. M. Estévez-Tapiador, and A. Ribagorda. Advances in Ultralightweight Cryptography for Low-Cost RFID Tags: Gossamer Protocol. In K.-I. Chung, K. Sohn, and M. Yung, editors, *WISA*, volume 5379 of *LNCS*, pages 56–68. Springer, 2008.
21. P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. EMAP: An Efficient Mutual Authentication Protocol for Low-Cost RFID Tags. In *OTM Federated Conferences and Workshop: IS Workshop – IS'06*, volume 4277 of *LNCS*, pages 352–361, Montpellier, France, November 2006. Springer-Verlag.
22. R. C.-W. Phan. Cryptanalysis of a New Ultralightweight RFID Authentication Protocol - SASI. *IEEE Transactions on Dependable and Secure Computing*, 6(4):316–320, 2009.
23. Y. W. Q Cai, Y Zhan. A minimalist mutual authentication protocol for rfid system and ban logic analysis[c]. In *In Proceedings of the 2008 ISECS International Colloquium on Computing, Communication, Control, and Management*, volume 2, page 449453, 2008.
24. A. Sadighian and R. Jalili. FLMAP: A Fast Lightweight Mutual Authentication Protocol for RFID Systems. In *The 16th IEEE International Conference On Networks (ICON 2008)*, pages 1–6, New Delhi, India, 2008.
25. A. Sadighian and R. Jalili. AFMAP: Anonymous Forward-Secure Mutual Authentication Protocols for RFID systems. In *The Third IEEE International Conference on Emerging Security Information, Systems and Technologies(SECURWARE 2009)*, pages 31–36, Athens, Greece, 2009.
26. M. Safkhani, N. Bagheri, M. Naderi, and S. Sandhya. Security analysis of LMAP++, an RFID authentication protocol. In *6th International Conference on*

Internet Technology and Secured Transactions (ICITST 2011), Abu Dhabi, UAE, Dec. 2011.
27. M. Safkhani and M. Naderi. Cryptanalysis and Improvement of a Lightweight Mutual Authentication Protocol for RFID system. In 7^{th} *International ISC Conference on Information Security and Cryptology 2010(ISCISC'10)*, pages 57–59, 2010.
28. M. Safkhani, M. Naderi, and H. F.Rashvand. Cryptanalysis of the Fast Lightweight Mutual Authentication Protocol (FLMAP). *International Journal of Computer & Communication Technology (IJCCT)*, 2(2,3,4):182–186, 2010.
29. J. Shen, D. Choi, S. Moh, and I. Chung. A Novel Anonymous RFID Authentication Protocol Providing Strong Privacy and Security. In *2010 International Conference on Multimedia Information Networking and Security*, 2010.
30. B. Song and C. J. Mitchell. RFID Authentication Protocol for Low-cost Tags. In *WiSec' 08*, pages 140–147, 2008.
31. B. Song and C. J. Mitchell. Scalable RFID security protocols supporting tag ownership transfer. *Computer Communications*, 34(4):556–566, 2011.
32. H.-M. Sun and W.-C. Ting. A Gen2-Based RFID Authentication Protocol for Security and Privacy. *IEEE Transactions On Mobile Computing*, 8(8):1052–1062, 2009.
33. C. C. Tan, B. Sheng, and Q. Li. Secure and Serverless RFID Authentication and Search Protocols. *IEEE Transactions on Wireless Communications*, 7(4):1400–1407, 2008.
34. R. Xueping and X. Xianghua. A Mutual Authentication Protocol For Low-cost RFID System. In *2010 IEEE Asia-Pacific Services Computing Conference*, pages 632–636, 2010.
35. W. W. Y. Gu. A light-weight mutual authentication protocol for ISO 18000-6B standard RFID system . In *Proceedings of ICCTA 2009*, pages 21–25, 2009.
36. T.-C. Yeh, C.-H. Wu, and Y.-M. Tseng. Improvement of the RFID Authentication Scheme based on Quadratic Residues. *Computer Communications*, 34(3):337–341, 2011.