Anonymous RFID Authentication Protocol Without a Trusted Party

Mehmet Sabır KİRAZ, Muhammed Ali BİNGÖL,
Süleyman KARDAŞ, Fatih BİRİNCİ

Abstract—**Anonymous authentication is a technique enabling users to prove that they have privilege without disclosing real identities. This type of authentication can be useful especially in scenarios where it is sufficient to ensure the server that the claiming parties are indeed registered. Many existing anonymous authentication protocols assume absolute trust to the back-end server in which all private keys are stored. This trust may result in serious security and privacy issues in case of private key leakage from the server.**

In this paper, we propose forward secure anonymous and mutual RFID authentication protocols where trust to the server-side is not needed. That is to say, even the private keys are obtained from the corrupted tags or the server owners of these tags cannot be traced from the past authentication actions. In fact, anonymity of the users will continue even the private keys of their tags are compromised. Our schemes are based on the threshold cryptosystems which provide security and simplicity. We assume all computations can be carried out with limited resources such as RFID tags.

Index Terms—**RFID, Authentication, Anonymity, Threshold cryptosystem**

I. Introduction

RADIO Frequency IDentification (RFID) is a means for identifying objects via a radio signal, and enables automated data gathering in a variety of applications. A typical RFID system is setup by a set of readers, a number of RFID tags and a back-end server. In general sense, an RFID tag is known as a small integrated circuit with a unique identifier which transmits data over the air in response to interrogation by an RFID reader [1]. A smart tag, on the other hand, has on-board processors that are typically capable of performing cryptographic operations. These smart RFID cards are being deployed in a range of applications, including electronic tickets, access control, public transportation payment cards, and novel forms of credit cards, and they are likely to be carried by most of the people as a means of identification, e.g. as a national ID card or electronic passport. Some recent works show that public-key cryptosystems can be deployed in RFID systems [2]–[4]. Our interest here is in smart RFID tags with public-key cryptosystem, and in particular we focus on the means of authentication used to access tag-specific information stored in a back-end server.

This work has been partially funded by COGSA (Multimedia Security Systems) project.

M.S. Kiraz, M. A. Bingöl, S. Kardaş, and F. Birinci are with TUBİTAK BİLGEM UEKAE (National Research Institute of Electronics and Cryptology), Gebze 41470 Kocaeli/Turkey e-mail: {m.kiraz, muhammedalib, skardas, fatih}@uekae.tubitak.gov.tr

There exists some applications (e.g., electronic tickets, public transportation, e-voting, reusable payment and cash system etc.) where it is only necessary and sufficient to verify whether a tag is a legitimate one without determining its identity. In these examples, people may prefer to register only once (e.g., after some payment or being a member) and would like to keep their anonymity and privacy when they use these applications. Nonetheless, most of existing approaches consider a back-end server as trusted entity and assumed to be physically secure and not attackable. However, the security of server-side cannot be guaranteed for some real-life applications. A variety of security and privacy threats to RFID authentication protocols have been widely studied, including, cloning tag, eavesdropping, replay attacks, denial of service (DoS) attacks, tracking, and traceability. Also, it is usually assumed that back-end server maintains all private information about tags, assigned keys, etc. In addition to these security and privacy threats, an adversary that has compromised a server could impersonate a valid tag using knowledge of the tags internal state. In this paper, we introduce this practical threat, namely Untrusted Server Side Attack (Big brother attack). In fact, a person, who is responsible for operating the server, may be interested in detailed user profiles (e.g., for later misuse). Hence, this malicious officer can link the users with their authentication information like authentication time, location. Therefore, if the contents of a server are revealed or controlled by an attacker, then such an attack could be used to cause cloning the tag, impersonating or tracing the users which is the case for most of the conventional RFID protocols.

Motivated by this need, we consider anonymous RFID based authentication protocol. In these protocols, even if an adversary corrupts the reader as well as the back-end server, she is not be able to trace any tags in the system but can authenticate them.

Our contributions. In this paper, we propose an anonymous RFID authentication protocol without using a trusted third party. To the best of our knowledge, this is the first paper that addresses and tries to provide a solution to the problem of server side corruption. We show the security of our protocol against what we called 'Big-brother attack' in which the server-side is corrupted or controlled by a malicious administrator. In this way, no malicious party from the server side has sufficient information for computing the private content of the tags and tracing them. We prevent those attacks by using threshold cryptosystems between the server and the tags. Our protocol also achieves forward and backward secrecies. Namely, although the private keys of tags are obtained this

will give no advantage to trace neither in the past nor in the future.

Organization of the paper. After giving brief preliminaries about threshold homomorphic cryptosystems, we present our first unilateral anonymous authentication scheme for RFID in Section III. In Section IV, we propose our mutual anonymous authentication scheme. We prove its security in Section IV-A and finally we conclude our paper in Section V.

II. Related Work

There are many RFID authentication protocols available. For more information, we refer interested readers to the excellent resource maintained by Avoine [5] and survey papers [6], [7]. Instead, we only focus on anonymous RFID authentication protocols.

In [8], the authors focus on anonymous credential systems where they propose a generic anonymous payment system including anonymous authentication for RFID-powered public transport tickets. An alternative approach to anonymous RFID-based payment has been proposed in [9]. In [10], the authors present an implementation of an anonymous credential system on Java Cards.

In [11], the authors propose three anonymous RFID authentication protocols (a 2-pass authentication protocol and two 1-pass authentication protocols). The authors claim that the last protocol is "optimistic" the cost is minimal when the adversary is passive.

In [7], the authors present an authentication protocol and a search protocol for RFID tags. Their authentication protocol provides security and privacy requirements without the need of a persistent central database. they also address the disadvantages having a secure central database and suggested solutions for overcoming them. Finally, they introduce a new problem of performing secure search for RFID tags.

In [12], the authors extend the universally composable (UC) framework for RFID authentication protocols. Informally speaking, the security of UC protocols is maintained under general composition with arbitrary other protocols running concurrently, and therefore such protocols are easily plugged into more complex protocols in a modular fashion without requiring a new security analysis. In addition to the availability, anonymity, and authenticity, in this paper, the authors address the forward-security issues for this UC framework in the presence of key compromise. They propose new protocols which satisfy forward-secure anonymity, authenticity, and availability requirements in the UC model. The article [13] recently proposed a universally composable security framework especially for RFID applications. They adopt RFID setup, communication, and concurrency assumptions in a model that guarantees strong security, privacy, and availability properties. Unlike [12], they do not consider security issues in the presence of key-compromise and tag corruption.

[14] proposes mutual authentication protocols for RFID systems. Some significant characteristics of the protocols are forward security, tag anonymity, location privacy, low complexity on the back-end server, and scalability. The authors claim that their protocols offer the most enhanced security features in RFID mutual authentication protocols with respect to user privacy. They also show that forward security and tag anonymity are guaranteed.

The authors in [15] proposed an anonymous RFID authentication protocol that preserves the security and privacy properties, and achieves better scalability compared with other contemporary approaches. However, [16] shows that this protocol some of the claimed security properties (especial untraceability) are not fulfilled. In this attack, an adversary can manipulate the messages between a tag and a reader during the protocol runs and can successfully trace the tag. The authors in [16] also improves the flawed protocol to satisfy all the claimed security and efficiency properties.

In [17], the authors propose an anonymous authentication scheme that allows RFID tags to authenticate to readers without disclosing any other information that allows tags to be traced. Their protocol provides anonymity and untraceability of tags against readers, tag authentication and availability.

We note that none of the above work considers the case of a malicious back-end server.

III. Warm up: Our First Protocol for Anonymous and Unilateral Authentication

A. Threshold cryptosystems

Our protocols for RFID authentication are based on threshold cryptosystems [18]. For the completeness of the paper, we give a brief description of threshold cryptsystems below.

We denote $E(m,r)$ the encryption of message m using randomness r with a semantically secure public key encryption scheme. In a (t,n)-threshold cryptosystem there are n parties, each of them holds a share of the overall secret key. In the set up phase a public key is generated which is available for all n parties to encrypt messages. If at least t parties cooperate, any encryption can be successfully decrypted, whereas any collusion of less than t parties cannot get any information about the plaintext.

The most widely used threshold cryptosystems are (based on) ElGamal or Paillier [19], [20]. Threshold ElGamal has the drawback of only allowing decryption of values belonging to a relatively small set, for which it is feasible to compute discrete logs. On the other hand, Paillier does not have this problem and allows decryption of encrypted values in an arbitrarily large set (e.g., 1024-bit integers). However, the distributed key generation protocol for threshold Paillier is very expensive compared to that for threshold ElGamal. Our both protocols are applicable to any threshold cryptosystem. Without loss of generality, in our protocols, we use the popular threshold ElGamal homomorphic encryption as an instance.

Denote x_i^P for a value x which belongs to a party P with an index i. Let q be a prime number of binary size of n where n is a security parameter. G_q forms a group of order q generated by $g \in G_q$. Let $f(x) = a_{t-1}x^{t-1} + \ldots + a_0$ be a polynomial of degree t where the coefficients are $a_i \in G_q$. The value $a_0 = f(0)$ is the overall secret key which is computed during the setup phase and unknown to everybody. We denote (x^P, y^P) as public and secret key pair of the party P respectively where y^P is computed as $y^P = f(x^P)$.

a) Threshold Decryption.: Given a ciphertext in the (t,n)-threshold cryptosystem and t decryption shares of t parties based on their respective shares of the secret key everyone can simply recover the plaintext by using a reconstruction algorithm. More formally, on ciphertext c, at least t parties broadcast $c_i = D_{sk_i}(c)$, where sk_i denotes the secret key share for the i-th party at this stage. Later, everyone can perform $m = R(c_1,\ldots,c_t)$ where $c = E(m)$, where R denotes the public reconstruction algorithm. Let us illustrate for the decryption and reconstruction algorithm using $(2,n)$-threshold ElGamal as an instance. The domain parameter of the threshold scheme is (G_q, g). There are two parties in our protocol, the server S and the i-th tag T_i. Each party has a unique secret key shares (x,y) which is computed from a secret curve $y = f(x) = a_1x + a_0$ where a_0 is the private key and $h = g^{a_0}$ is the corresponding public key. Given a message m, the encrypted message pair is computed as $(C = (g^r, h^r m))$ where $r \in_R \{0,1\}^\ell$. The server computes its decryption share as $\sigma_S = g^{ry^i \frac{x^T}{x^T - x^S}}$ and the tag computes its decryption share as $\sigma_T = g^{ry^T \frac{x^S}{x^S - x^T}}$. Finally, the original message could be recovered by computing $m' = \frac{h^r m}{\sigma_S \sigma_T}$.

B. Our protocol

We are now going to present our first protocol which satisfies anonymous and mutual authentication. This protocol is interesting since the overhead for the server can be significantly decreased. Namely, the server can pre-compute a large set of encryptions before running the protocol, in this way the protocol can be more efficient.

Note that the system uses $(2,n)$-threshold cryptosystems therefore two parties the server and a tag can decrypt encryptions in order to pass the authentication. Our protocol involves the following entities.

C. Entities

The protocol can be implemented using any threshold cryptosystem. For ease understanding, we are going to illustrate our protocol with threshold ElGamal encryption. In this system, each party stores only one unique decryption shares $(x, y = f(x))$, which is computed and distributed by an issuer.

1) **Server** S: The server stores its own key share (x^S, y^S).
2) **Tag** i: Each tag is attached to a single object. A tag i has enough volatile for computation and non-volatile memory for storing its own share (x^i, y^i) and public share of server (x^S). Tags can compute modular exponentiation and inversion and can generate random nonces.
3) **Issuer** $\mathcal{I}$: Issuer generates a prime number q of binary size of n (n is a security parameter and should be large enough) that uniquely specifies a group G_q of order q. It also generates a generator g which is element of G_q. Moreover, $\mathcal{I}$ sets up a secret polynomial degree of 1 $(f(x) := a_1x + a_0)$ where the coefficients are elements of G_q and $a_0 = f(0)$ is the actual secret key. Lastly, $\mathcal{I}$ generates unique secret shares and sets up each entity with unique share in a secret channel.

The protocol steps are described as follows. The protocol is also sketched in Figure 1.

Step 1.The i-th tag picks a random value $r^i \in_R \{0,1\}^\ell$ and sends it to the server where ℓ is a security parameter.

Step 2.The server also picks a random value $r_S \in_R \{0,1\}^\ell$ and computes an ElGamal encryption of $m = r_S||r^i$. Note that because of using semantically secure randomized encryption scheme any two ciphertexts of a message m are completely different. To encrypt the message m, she first picks a random nonce $r \in \mathbb{Z}_q$. Then the encryption pair is computed $(C = (h^r m, g^r))$. She also computes the decryption share $\sigma = g^{ry^S}$. The server sends C and σ to the i-th tag.

Step 3.Upon receiving the message σ, C, the tag first completes the decryption share of the server as $\sigma_s = \sigma^{\frac{x^i}{x^i - x^S}}$. Then, it computes its decryption share as $\sigma_i = g^{ry^i \frac{x^S}{x^S - x^i}}$. Finally, it recovers the original message as $\widehat{r}^S||r'^i = \frac{h^r m}{\sigma_S \sigma_i}$. If the $\widehat{r}^i$ is equal to the original random value r^i, it sends $\widehat{r}^S$. Otherwise, it sends a random value $\widehat{r} \in_R \{0,1\}^\ell$.

Step 4.The server verifies whether the received value $\widehat{r}$ is equal to the value r_S she generated at Step 1.

In the next section, we will slightly adapt this protocol to be able to satisfy anonymous and secure mutual authentication. This protocol will be interesting since the overhead for the server can be significantly decreased. Namely, the server can pre-compute a large set of encryptions before running the protocol, in this way the protocol can be more efficient.

D. Adding/Removing a Tag

Whenever $\mathcal{I}$ wants to add a newly generated tag i to the system, he first picks a random x^i and computes y^i value from the curve $y = a_1x + a_0$. Then, (x^i, y^i) and public key share of the server x^S are attached to tag T_i. In order to revoke a tag from the system, the issuer simply generates another secret curve and re-compute the share of the each tag and the reader.

IV. Our Second Protocol for Anonymous and Mutual Authentication

We are now ready to present our second anonymous protocol in which the server can revoke at most $t-1$ tags. In this system, the server has $t-1$ different secret shares $((x_1^S, y_1^S), \ldots, (x_{t-1}^S, y_{t-1}^S))$ whereas each tag i has only one unique secret share (x^i, y^i). This protocol is based on (t,n) threshold cryptosystem. The protocol steps, which are also sketched in Figure 2, are described as follows.

Step 1.The i-th tag picks a random value $r^i \in R\{0,1\}^\ell$ and sends it to the server.

Step 2.The server also picks a random nonce $r^S \in_R \{0,1\}^\ell$ and computes an ElGamal encryption of $m = r^S||r^i$. To encrypt the message m, she first picks a random nonce $r \in \mathbb{Z}_q$. Then the pair of encryptions are computed $(C = (h^r m, g^r))$. She also computes its decryption shares $\sigma_i = g^{ry_i^S} \forall i = 1,\ldots,t-1$. The server sends C and the decryption shares along with

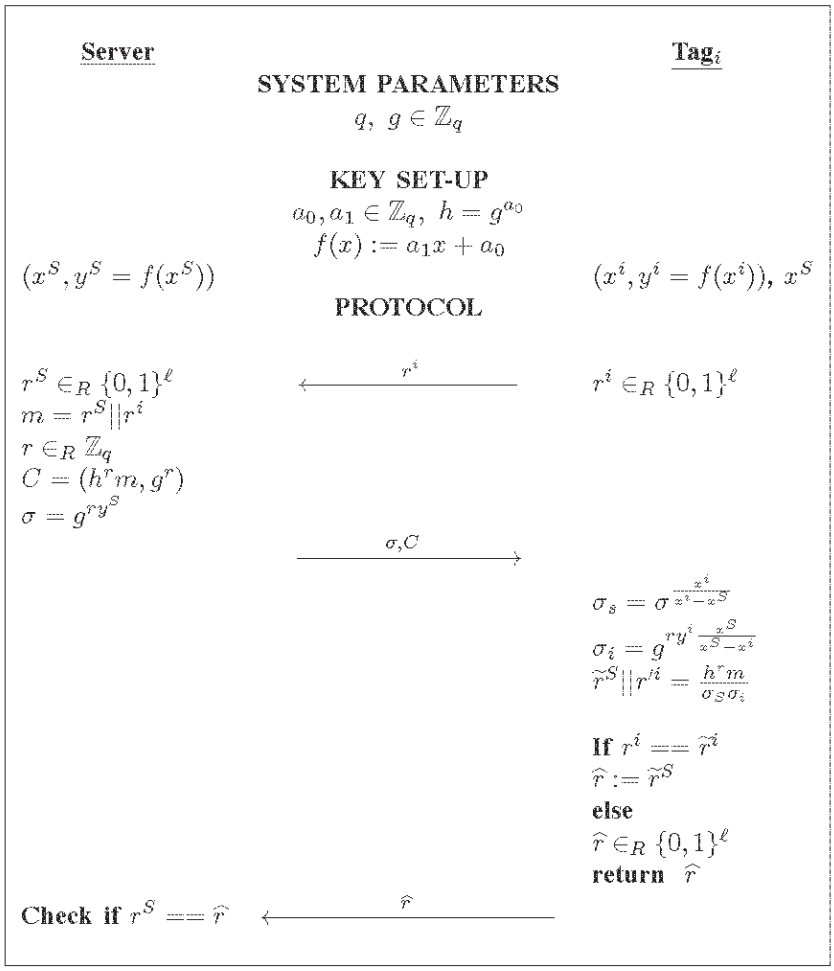

Fig. 1. Our anonymous mutual authentication protocol based on (2,n) threshold cryptosystem.

its public shares $((\sigma_1, x_1^S), \ldots, (\sigma_{t-1}, x_{t-1}^S))$ to the tag i.

Step 3. Upon receiving the message $((\sigma_1, x_1^S), \ldots, (\sigma_{t-1}, x_{t-1}^S), C)$, the tag first completes the decryption shares of the server as $\sigma_S = \prod_{j=0}^{t-1} \sigma_j^{\frac{x^i}{x^i - x_j^S}}$. Then, it computes its decryption share as $\sigma_i = g^{r y^i \prod_{j=1}^{t-1} \frac{x_j^S}{x_j^S - x^i}}$. Finally, it recovers the original message as $\widetilde{r}^S || \widetilde{r}^i = \frac{h^r m}{\sigma_S \sigma_i}$. If the $\widetilde{r}^i$ is equal to the original random value r^i, it sends $\widetilde{r}^S$. Otherwise, it sends a random value $\widehat{r} \in_R \{0,1\}^\ell$.

Step 4. The server verifies whether the received value $\widetilde{r}^S$ is equal to the value r_S she generated at Step 1.

A. Security & Complexity Analysis

For the security analysis, we are going to show that this protocol fulfills the privacy and security requirements for RFID authentication.

Theorem IV.1. *Our first protocol depicted in Figure 1 achieves anonymous authentication even the server is fully corrupted.*

Proof: (Sketch) Assume that an adversary compromises the server and has access to all the private information. The adversary has two choices for an attack.

(i) In the first case, she behaves like a semi-honest party, i.e., follows the protocol properly but try to identify the tag. In that case, at Step 1 of the protocol, the adversary cannot obtain any information about the tag since r^i is completely random. At Step 2, the adversary computes the decryption shares properly and sends them to the tag. At Step 3, the adversary receives only $\widehat{r}$ which gives only the information that the tag is one of the member in the database.

(ii) Now, the adversary does not behaves like a malicious party. Assume that she access to all the private values of the tags from the server. Similar to the previous attacks, the adversary cannot obtain any information about the tag since r^i is completely random. However, at Step 2, the adversary computes the decryption shares with one of the secret shares of the tags in the system and sends them to the tag. At Step 3, the adversary receives only $\widehat{r}$ which is random because the tag could not extract his r^i, and hence the tag sends random bits. The adversary still could not distinguish this tag from others and therefore cannot identify it. ∎

Corollary IV.2. *Our protocol depicted in Figure 1 achieves both forward and backward secrecies without any assumption.*

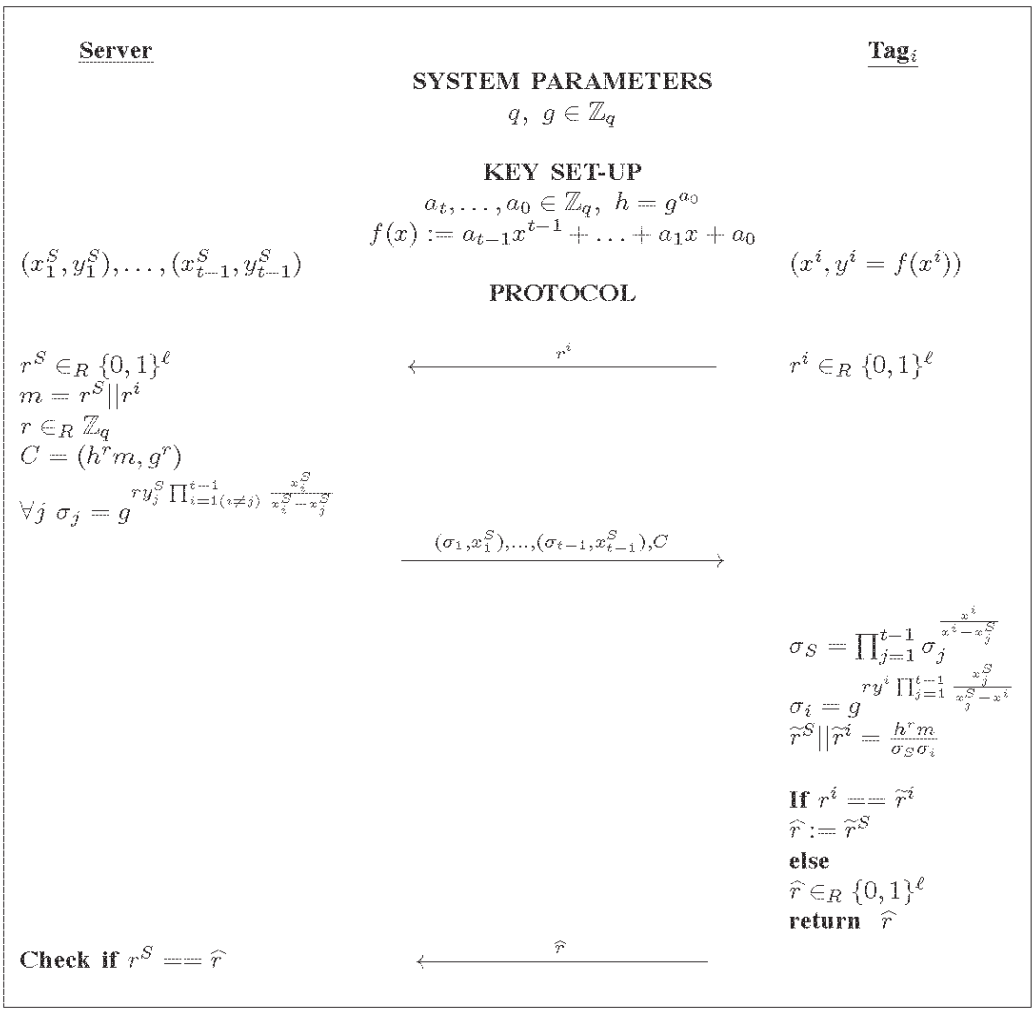

Fig. 2. Our anonymous mutual authentication protocol based on (t,n) threshold cryptosystem.

Proof: (Sketch) Since the protocol achieves secure and anonymous authentication against even Big Brother attack (Theorem IV.1), the adversary is not able to distinguish a tag from others at any time line. Therefore, our protocol achieves both forward and backward secrecies. ■

Unlinkability of tags. The term 'unlinkability' means that an attacker cannot distinguish tags based on their communications. In other words, the protocol messages generated by the tags should not leak any information to an adversary for identification or traceability. Since only the random numbers (which is independent of the secret keys) are sent at the first and the third steps of our protocol, no information about the tag identity is revealed. At the second step, the encryptions $C, (\sigma_i, x_i^S)$ for $i = 1, \ldots, t-1$ are sent by the server which is common for all the tags. Therefore, no attacker can be able to link this message to a tag.

Complexity analysis. Unlike previous protocols, although our protocols use public key operations it is completely secure against server side attacks described above. Still, in total there are only three exponentiations, three inversion and only one multiplication for a tag. At the tag side, two inversions ($\frac{x^i}{x^i - x^S}$, $\frac{x^S}{x^S - x^i}$) can be done off-line and can be stored on its memory beforehand. At he server side, only three exponentiations and one multiplication are performed. We highlight that all the encryptions and partial decryption done by the server can be pre-computed off-line. Note that adding a new tag does not change the complexity of the overall system. Namely, a new user will only get a new share of the secret key, and will do the same computations as other tags. This does not incur any additional complexity to the server.

V. Conclusion

In this paper, we have discussed and proposed a new anonymous and mutual RFID authentication protocol. Our protocol enables RFID tags to authenticate to readers without disclosing any information that allows the identification or tracking of tags even to malicious readers. First, we introduced our first anonymous RFID protocol to give a warm-up which is based $(2, n)$-threshold homomorphic encryption. This protocol does not provide tag revocation. We then proposed our second protocol based on (t, n)-threshold homomorphic encryption which allows tag revocations up to t tags. We highlight that the protocol is still secure even if the server side is

corrupted. However, our current solution does not capture removing an tag which is an interesting open problem for future research. Finally, we show that our protocol satisfies the security requirements like anonymity, privacy, authentication and unlinkability.

REFERENCES

[1] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, 2nd ed. New York, NY, USA: John Wiley & Sons, Inc., 2003.

[2] L. Batina, J. Guajardo, T. Kerins, N. Mentens, P. Tuyls, and I. Verbauwhede, "Public-Key Cryptography for RFID-Tags," in *International Workshop on Pervasive Computing and Communication Security – PerSec 2007*, IEEE. New York City, New York, USA: IEEE Computer Society, March 2007, pp. 217–222.

[3] M. McLoone and M. J. B. Robshaw, "Public key cryptography and rfid tags," in *Topics in Cryptology - CT-RSA 2007, The Cryptographers' Track at the RSA Conference 2007*, San Francisco, CA, USA, 2007, pp. 372–384.

[4] Y. Yao, J. Huang, S. Khanna, A. Shelat, B. H. Calhoun, J. Lach, and D. Evans, "A Sub-0.5V Lattice-Based Public-Key Encryption Scheme for RFID Platforms in 130nm CMOS," in *Workshop on RFID Security – RFIDSec Asia'11*, ser. Cryptology and Information Security, vol. 6. Wuxi, China: IOS Press, April 2011, pp. 96–113.

[5] G. Avoine, "Rfid security & privacy lounge," http://www.avoine.net/rfid, 2012.

[6] A. Juels, "Rfid security and privacy: A research survey," *JOURNAL OF SELECTED AREAS IN COMMUNICATION (J-SAC)*, vol. 24, no. 2, pp. 381–395, 2006.

[7] A. S. T. Melanie R. Rieback and, Bruno Crispo and, "The evolution of rfid security," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 62–69, 2006.

[8] T. S. Heydt-benjamin, H. jin Chae, B. Defend, and K. Fu, "K.: Privacy for public transportation," in *In: Proceedings of Privacy Enhancing Technologies workshop (PET*, 2006.

[9] E.-O. Blass, A. Kurmus, R. Molva, and T. Strufe, "Psp: private and secure payment with rfid," in *Proceedings of the 8th ACM workshop on Privacy in the electronic society*. ACM, 2009, pp. 51–60.

[10] P. Bichsel, J. Camenisch, T. Groß, and V. Shoup, "Anonymous credentials on a standard java card," in *Proceedings of the 16th ACM conference on Computer and communications security*, ser. CCS '09. ACM, 2009, pp. 600–610.

[11] C. Chatmon, T. van Le, and M. Burmester, "Secure anonymous rfid authentication protocols," no. Technical Report TR-060112, pp. 1–10, 2006.

[12] T. V. Le, M. Burmester, and B. de Medeiros, "Universally composable and forward-secure rfid authentication and authenticated key exchange," in *Proceedings of the 2nd ACM symposium on Information, computer and communications security*, ser. ASIACCS '07. ACM, 2007, pp. 242–252.

[13] M. Burmester, T. V. Le, B. D. Medeiros, and G. Tsudik, "Universally composable rfid identification and authentication protocols," *ACM Trans. Inf. Syst. Secur.*, vol. 12, 2009.

[14] A. Sadighian and R. Jalili, "Afmap: Anonymous forward-secure mutual authentication protocols for rfid systems," *Emerging Security Information, Systems, and Technologies, The International Conference on, SECURWARE'09*, pp. 31–36, 2009.

[15] M. Burmester, B. de Medeiros, and R. Motta, "Robust, anonymous rfid authentication with constant key-lookup," in *Proceedings of the 2008 ACM symposium on Information, computer and communications security*, ser. ASIACCS '08. New York, NY, USA: ACM, 2008, pp. 283–291.

[16] B. Liang, Y. Li, C. Ma, T. Li, and R. Deng, "On the untraceability of anonymous rfid authentication protocol with constant key-lookup," in *Proceedings of the 5th International Conference on Information Systems Security*, ser. ICISS '09. Springer-Verlag, 2009, pp. 71–85.

[17] F. Armknecht, L. Chen, A.-R. Sadeghi, and C. Wachsmann, "Anonymous authentication for rfid systems," in *Workshop on RFID Security (RFIDSec)*, ser. LNCS, vol. 6370. Springer, 2010, pp. 158–175.

[18] *Threshold Cryptosystems*, ser. Lecture Notes in Computer Science, vol. 435. Springer, 1990.

[19] T. E. Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Proceedings of CRYPTO 84 on Advances in cryptology*. Springer-Verlag New York, Inc., 1985, pp. 10–18.

[20] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques*, ser. Lecture Notes in Computer Science. Springer, 1999, pp. 223–238.