Group Authentication in Heterogeneous RFID Networks

Bo Sheng
Department of Computer Science
University of Massachusetts Boston
shengbo@cs.umb.edu

Chiu C. Tan
Department of Computer and Information Sciences
Temple University
cctan@temple.edu

***Abstract*—The proliferation of RFID tags means that an RFID reader will often interact with large groups of RFID tags. However, the majority of RFID security research is still focused on securing single reader-single tag interaction, which is unlikely to be scalable to larger groups of tags. In this paper, we propose a protocol that allows a large group of tags to authenticate a reader. Our technique relies on using a more powerful RFID tag, known as a computational RFID tag (CRFID), which is included into the large group of tags. Theoretical analysis and simulation experiments show that our new protocol is secure and efficient.**

I. Introduction

RFID technology is increasingly being used in applications ranging from warehouse inventory control to supply chain product tracking. RFID security is an important component for these applications, especially when the tags are used for sensitive applications such as counterfeit detection [1]. There have been significant research efforts on RFID security [2], [3] developing protocols that allow an RFID reader to authenticate an RFID tag, and vice versa.

As the number of RFID tags increases, however, applying a single reader-single tag authentication protocol multiple times for every tag in the group is unlikely to work well. The reason is that when an RFID reader encounters a set of tags, an underlying network operation, known as *singulation* or *anti-collision*, needs to take place before any authentication protocol can be executed. Singulation helps the tag identify a specific period of time in which other tags will not interfere with its communication with the reader. This is a necessary procedure in RFID communication because of the fact that RFID tags have limited hardware and they cannot perform carrier sensing. Absent some coordination step by the reader, all tags will compete for the same wireless channel and interfere with each other. Consequently, the communications between the reader and tags can hardly be successful. While there is ongoing research on algorithms to improve this process [4], [5], the time needed to complete the singulation will undoubtedly increase as the group size increases. Applying authentication on top of singulation will lead to scalability issues.

In this paper, we propose an efficient authentication protocol between a single reader and a group of RFID tags. An practical example of a group of tags will be a crate containing multiple items (each item with its own RFID tag) which make up a group of tags. To overcome the limitations of RFID tags, we consider a group of heterogeneous RFID tags consisting of regular RFID tags and some more advanced computational RFID tags (CRFID). A CRFID tag is capable of performing more complex operations while still retaining a sufficiently low energy footprint. The CRFID tags will likely only constitute a small portion of the group because they are more expensive. Our basic idea is to let CRFID tags authenticate the reader first, and then assist the reader to prove to other passive tags that the reader has been authenticated. There are two major challenges in the protocol design. First, the CRFID tags can not communicate with regular passive tags directly. Thus all communication has to go through the reader via insecure wireless channels which gives an adversary the opportunity to manipulate, replay or crack the messages. Second, it is difficult to efficiently enable multiple tags to authenticate the reader when they do not share a common secret. We proposed a novel scheme that uses a bloom-filter like structure as a certificate for authentication (as shown in Fig. 1). Our solution is scalable as the authentication can be completed in a constant rounds of message exchanges regardless of the group size.

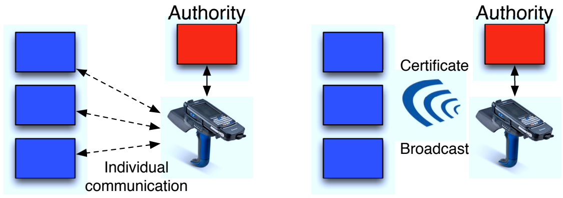

Fig. 1. Individual Authentication vs. Group Authentication

The rest of the paper is as follows. In Section II, we review the related work for research into group RFID algorithms. We formulate our problem and introduce the system model in Section III. Section IV contains our solution and theoretical analysis, Section V contains the simulation results, and Section VI concludes.

II. Related Work

The low-cost nature of RFID tags makes them suitable to be used in large numbers to track items. For applications which require more security protections, a natural extension to RFID reader-tag security protocols is to design protocols for a single reader and a group of RFID tags. One such technique is the “yorking proof” protocol [6]–[8]. The general idea is for the reader to interact with each RFID tag in a group individually one at a time. Each time, the reader collects the response from one tag to serve as the input to the other tags in the group. Eventually, the reader will obtain a "proof" that all tags are present. In this paper, our protocol does not require the reader to contact all tags one by one, thus improving performance, and also allows the tags to authenticate the reader, improving the security of the process.

Another related area of research for large group of RFID tags are protocols that take advantage of the MAC layer behavior. Work by [9] uses the response frame of the RFID tags (the MAC layer behavior) to quickly estimate the cardinality of a set of RFID tags. Later extensions that use the same MAC layer behavior for other purposes include energy-efficient querying [10], [11], privacy-preserving estimation [12], [13], data mining [14], and missing tag detection [15], [16]. Most of these prior work do not consider security and cannot enable the tags to authenticate the reader. More related to our work is [17] which uses the MAC layer to allow the reader to authenticate the group of tags. This is complementary to our work which is to let the tags authenticate the reader.

Our protocol uses a powerful tag as part of the authentication protocol. The use of more powerful hardware to address the limited computational capabilities of RFID tags is an active area of research [18]–[20]. The role of the powerful tag is to authenticate the reader, and to prevent unauthorized readers from querying the regular tags. Since regular RFID tags cannot be turned-off, this is usually done by disrupting the wireless communications between the reader and tag so that the unauthorized reader cannot get any meaningful data from the tags. The powerful tag can be a modified RFID tag [21], or a specially designed embedded device [22], [23]. The main difference is that in previous solutions, the powerful tag was designed to prevent an unauthorized reader from accessing the rest of the tags. The problem of ensuring that the group of tags are all legitimate and present was not considered. Our approach considers the authentication of the reader, powerful tag, and normal tags.

III. System Model and Problem Formulation

We consider a group of tags G consisting of one computational RFID tag (CRFID), indicated as CT, and a set of n regular passive RFID tags, $\{t_1, t_2, \ldots, t_n\}$. Passive tags have limited abilities. Besides backscattering data to RFID readers, they are able to generate (pseudo) random numbers and apply hash functions. Tag CT is more powerful and capable of encrypting, conducting loops, and etc. But CT can not directly communicate with passive tags.

We assume there is an adversary who is interested in obtaining *useful information* from the group of tags. Useful information is defined as any secret keys (from reader, CT or passive tags), what the tag IDs are, or the data stored within the tags. The adversary simply learning whether there are any tags present is *not* considered useful information so long as he is unable to infer *what* those tags are. The purpose of allowing the group of tags to each authenticate the reader is to prevent an unauthorized reader from obtaining useful information. We assume the adversary can launch the following attacks.

- Eavesdrop: In this attack, the adversary can observe all the interactions between a legitimate reader and a group of legitimate tags. The adversary is able to distinguish between the communication of a CT and a regular passive tag. The adversary succeeds if he is able to learn any useful information about the group of tags.
- Replay: In this attack, the adversary uses the eavesdropped information and replays them back to either the CT or the passive tags. The adversary succeeds if his reader can use the replayed information to convince a groups of tags (that is the CT, passive tags, or both) that it is a legitimate reader and obtain useful information.
- Tracking: In a tracking attack, the adversary tries to identify a specific group of tags over time. The adversary launches the attack by first repeatedly querying a particular group of tag to record its responses. Some time later, the adversary will query different groups of tags. The adversary succeeds if he is able to identify the original group with a certain probability.
- Physical access: In this attack, the adversary is able to physically remove the CT, and then attempts to read the rest of the passive RFID tags. The adversary succeeds if he is able to obtain any useful information. Defending against this type of attack will be useful to ensure security even in practical situations where the CT is damaged.

Considering the above threats, our objective is to let the group of tags authenticate the reader before transmitting data to it.

IV. Solution

In this section, we present our novel protocol for group authentication. First, we define three states for the CRFID tag and passive RFID tags which can be easily incorporated into the current RFID hardware.

- Idle state: This is the default state when all tags are powered up and waiting for queries from the reader.
- Authenticated state: This is the state after the tag has authenticated the reader. A passive tag in this state is ready to transfer data. The CRFID tag in this state is willing to assist the reader in the authentication process between passive tags and the reader.
- Jamming state: For both passive tags and CRFID tag, this state indicates that they have observed suspicious information transferred by the reader. Therefore, in the following data collecting phase, the tag in this state will jam the channel by responding random bits at every slot to protect data from other tags.

Table I lists some notations we will use in this section.

A. Sketch of Protocol

1) Authentication between CT and R: This part of authentication is relatively straightforward because both the reader R and the CRFID tag CT are capable of typical cryptographic operations. The existing authentication schemes can be directly applied here. Assume the reader has obtained the group key

R	the RFID reader
n	the number of passive tags
T / t_i	the set of passive tags / the i-th passive tag, $T = \{t_1, \ldots, t_n\}$
CT	the powerful CRFID tag in the group
K_G	the group key shared between server and CT
s_i	the shared secret between each tag t_i and CT
C / m	the certificate / the bit length of C
ft	the initial frame size
p	the probability a passive tag will respond in the initial frame

TABLE I
NOTATIONS

K_G from the server, the following protocol enables R and CT to authenticate each other.

1. $R \Rightarrow CT$: Request for authentication
2. $R \Leftarrow CT$: $\{r_1\}_{K_G}$ 3. $R \Rightarrow CT$: $\{r_1, r_2\}_{K_G}$

Here, r_1 and r_2 are two random numbers used as a challenge. After the third message, if CT can decrypt these two random numbers, the authentication is accomplished and the reader R and CT can agree on a session key (e.g., $h(r1||r2)$) for the subsequent communication.

2) Authentication between passive tag set T and R: In the second step, the reader R needs to prove to the set of passive tags T that it has been authenticated by CT. In fact, it is easy for each individual tag t_i to authenticate R by contacting CT through R. The following protocol is an example:

1. $R \Rightarrow t_i$: Request for authentication
2. $R \Leftarrow t_i$: r_1 3. $R \Rightarrow CT$: r_1
4. $R \Leftarrow CT$: $h(r_1||s_i)$ 5. $R \Rightarrow t_i$: $h(r_1||s_i)$

The above type of authentication, however, incurs a large overhead for each tag. In the given example, five messages have to be transmitted for each passive tag to authenticate the reader. For a large group with tens or hundreds of passive tags, the accumulated delay is unacceptable in practice.

In this paper, we propose a novel solution that allows the whole group of passive tags to authenticate the reader in one round of communication which is significantly more efficient for a large scale group. Our solution includes the following three steps.

Step 1: First, we want the group of passive tags T to raise a random nonce to challenge R. In our solution, we use the tags' response in a frame to form a bitstring. Initially, CT sends R a random number r and frame size ft. The reader R broadcasts r to the entire group and will use a frame of ft slots to scan the passive tags. Note that ft is preloaded to each passive tag, thus is not transmitted by the reader R. Upon receiving this message, each tag t_i will respond in the following frame with a probability p and the responding slot's index is calculated as $h(r||s_i) \bmod ft$. The parameter p is required to be resilient to replay attack. Even if R broadcasts the same random number r for multiple times, the tags' responses will be different. Meanwhile, tag CT also listens to the channel and if the broadcast random number is different from r, CT will turn to 'jamming' state.

Step 2: Next, the reader R uses a frame of ft slots to scan passive tags. Only partial tags will participate in this round and each of their responses is short random bitstring. The reader is able to determine each slot as an empty slot, single-reply slot, or collision slot. After scanning the frame, R will report the result r_T as a bitstring to CT in the following format: each slot is represented by 2 bits, '00', '01', and '11' indicate an empty slot, single-reply slot and collision slot respectively. The following figure illustrates an example:

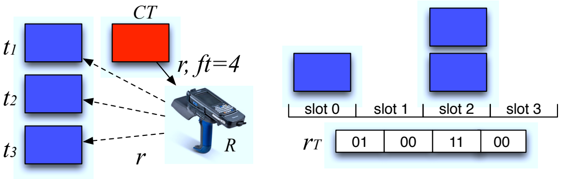

Fig. 2. An example with $n = 3$ and $ft = 4$.

Step 3: CT will calculate another random number r' and certificate bitstring $C = c_1 c_2 \ldots c_m$ based on r_T, and then send both of them to the reader. Intuitively, C is a bloom filter that allows each tag to verify its secret. We will present the algorithm that derives C in the next subsection. The bistring C also holds a property that half bits are '0's and the other half are '1's. R will further broadcast $\{r_T, r', C\}$ to each passive tag t_i. Upon receiving the message, each tag, if has responded in Step 2, will verify if its reply has been included in r_T, i.e., if its responding slot is represented by '01' or '11'. If the verification fails, the passive tag will turn to 'jamming' state. In addition, all the passive tags will verify if C holds the special property that half bits are '0's. Then, each tag will apply k hash functions to calculate k indexes

$$idx_j = h_j(r_T||r'||s_i) \bmod m, j \in [1, k].$$

If $\forall j, c_{idx_j} = 1$, tag t_i will turn to 'authenticated' mode and be ready to transfer data. For example, assume there are 3 tags and each uses 2 hash functions h_1 and h_2. And we use 8 bits to represent the certificate. If the calculated index values are as shown in the following table, the certificate will be 11010100.

	t_1	t_2	t_3
h_1	0	0	5
h_2	1	3	1

B. Parameter Setting and Security Analysis

In this subsection, we discuss how to set the parameters mentioned in the above protocol.

1) Setting p and ft: These two parameters are needed in Step 1 and Step 2. Basically, they help the group of passive tags generate a random number. For each one of ft slots in the responding frame, we hope it has as equal probability as possible to be an empty slot, single-reply slot, or collision slot. Let $P0$, $P1$ and PC be the probability for each slot to be an empty slot, single-reply slot, and collision slot respectively. Our objective is to find the best parameters that yield the minimal standard deviation among $P0$, $P1$, and PC.

In our protocol, expectedly, $n \cdot p$ passive tags participate in

Step 2. Therefore, we have

$$\begin{aligned} P0 &= (1-\frac{1}{ft})^{n\cdot p} = e^{-\frac{n\cdot p}{ft}} \\ P1 &= \frac{n\cdot p}{ft}\cdot(1-\frac{1}{ft})^{n\cdot p-1} = \frac{n\cdot p}{ft}\cdot e^{-\frac{n\cdot p}{ft}} \\ PC &= 1-P0-P1. \end{aligned}$$

Let $x = \frac{n\cdot p}{ft}$, $P0$, $P1$, and PC are all functions on x. By minimizing the standard deviation $std(P0, P1, PC)$, we derive the best value of x is 1.156, i.e., $\frac{n\cdot p}{ft} = 1.156$. Heuristically, we set $p = 0.5$, and correspondingly, the optimal frame size is $ft = 0.433n$.

2) Setting m and k: These two parameters are critical for the performance of a bloom filter. Traditionally, false positive is the major metric and m and k are set to yield the minimum false positive. According to the literature, the false positive FP can be expressed as $FP = (1-e^{-\frac{k\cdot n}{m}})^k$. Given n and m, the optimal m that gives the minimum value of FP is

$$m = -\frac{n\cdot \ln FP}{(\ln 2)^2}.$$

The optimal value of k is $k = \frac{m}{n}\cdot \ln 2$.

In our solution, however, the bloom filter is used for authentication and false positive is no longer an issue because only group members participate the authentication. Knowing each tag's secret s_i, the computational RFID tag CT can calculate a bloom filter as the certificate and give it to an authenticated reader to convince passive tags. Our major concern is that the adversary may use a fake certificate to pass the verification. For example, if all bits of the certificate are '1's, this special bloom filter will definitely convince all the passive tags. If CT is present and overhears the fake certificate, it can turn to 'jamming' state and jam the rest of the communication. But it is possible that the adversary may remove CT and conduct this attack. In our solution, we specify a requirement that half bits in the certificate have to be '0's ('1's) to prevent the adversary from using invalid certificate. When setting m and k, we want them to be feasible for CT to generate half bits of '0's ('1's).

For any bit in the certificate C, let the probability that it is '0' be δ, $\delta = (1-\frac{1}{m})^{n\cdot k}$. Then we have

$$m = -\frac{n\cdot k}{\ln \delta}.$$

In our setting, applying multiple hash function does not help in authentication. Thus we always set k to 1. We will use the above equation in the following subsection and eventually derive the value of m.

C. Calculating C and r'

In Step 3, CT will apply the following Algorithm 1 to calculate the certificate C and the corresponding random number r'. Basically, the algorithm is an infinite loop until a valid certificate is generated. Lines 3-8 are two embedded loops that calculate the index for every passive tags. All resulting bits in C are marked as '1's in Line 6, where $C(i)$ indicates the i-th bit in C. Line 9 is the termination condition. If the condition is not satisfied, CT will generate another random number and repeat the process in another round.

Algorithm 1 CT : Calculate C and r'

```
1:  while true do
2:    Generate a random number $r'$, and $C \leftarrow 0$
3:    for $i = 1$ to $n$ do
4:      for $j = 1$ to $k$ do
5:        $idx_j$=$h_j(r_T||r'||s_i)$ mod $m$
6:        $C(idx_j)$=1
7:      end for
8:    end for
9:    if $\sum_i C(i) \leq \frac{m}{2}$ then
10:     return $C$ and $r'$
11:   end if
12: end while
```

In this algorithm, m (the length of the certificate) is a critical parameter. For efficiency performance, we want it to be as small as possible. However, smaller C will make it more difficult for CT to generate a valid certificate. In the next, we will formulate the constraints for m and find the minimum value of m that satisfies the constraints.

For each round, let X be the random variable representing the number of 0 bits in C. It follows the Binomial distribution, $X \sim Bino(\delta, m)$. In our solution, we require X to be at least $\frac{m}{2}$. Note that if $X > \frac{m}{2}$, CT can randomly flip $(X - \frac{m}{2})$ '0' bits to generate a valid certificate. Let Pr the probability that C has at least $\frac{m}{2}$ bits of '0's

$$Pr = 1 - Binocdf(\frac{m}{2}, m, \delta).$$

With a set of inappropriate parameters, it may take a lot of trials for CT to generate a valid certificate, which further delays the whole authentication process. Therefore, we introduce a pair of user-specified parameter (W, ϵ) to confine the overhead on CT for this step. The requirements indicates that CT needs to generate a valid C within W rounds of trials with more than $1-\epsilon$ confidence. W can be determined based of the computation performance of CT and ϵ is a usually small constant (e.g., 0.01).

Therefore, our problem of setting m is formulated as

$$\begin{aligned} &\text{minimize } m \\ s.t. \quad &(1-Pr)^W < \epsilon, \end{aligned}$$

where $(1-Pr)^W$ is the probability that CT can not generate a valid certificate after W rounds.

D. Security Analysis

Next, we will analyze how our protocols defend against the various adversary attacks. For the eavesdropping attack, since we assume existing secure techniques can be used in the reader CT authentication, the adversary learns nothing from this process. For the remaining steps, the adversary only observes r, ft, r_T, r' and C. Of these, only ft will remain the same. However, since many different groups can have the same ft value, this is not useful information. The values of r and r' can selected by the CT each time, and do not leak any secret regarding the passive tags. The values of r_T and C are computed with the passive tag's secret s_i, but the s_i only serves as input into the hash function $h()$. Since $h()$ is a cryptographic secure hash function, the adversary cannot invert the output to derive s_i. Thus, the protocol defends against the eavesdropping attack.

The next attack is the replay attack. Since the adversary is not an authorized reader, CT will not issue r and ft to the reader. Through eavesdropping, the adversary learns previous valid parameters, which we denote as $\hat{r}$, $\hat{ft}$, $\hat{r_T}$, $\hat{r'}$, $\hat{C}$. The adversary can try to replay $\hat{r}$ and $\hat{ft}$ to the passive tags. Each passive tag has some probability p to reply, and the returned r_T value will not match $\hat{r_T}$. Then in Step 3, the adversary will reply the incorrect $\hat{C}$ value to the passive tag because CT will not help it compute the correct value in Step 2. Thus, the passive tags will fail to authenticate the adversary, and not return useful information back to the reader.

In the tracking attack, an unauthorized reader may try to identify the group by conducting Step 1. Assuming the reader has eavesdropped ft, tags will respond in the following frame with probability p to generate r_T. Let us consider the adversary has queried the same group of tags twice in frame f_1 and f_2. For any empty slot in f_1, the probability that it remains empty in f_2 is the same as $P0 = (1 - \frac{1}{ft})^{n \cdot p}$. With our default parameter setting ($n = 100, p = 0.5$), $P0 = 0.31$. Similarly, the probabilities that a single-reply slot and collision slot remain the same are $P1 = 0.36$ and $PC = 0.33$. Thus, the probability of generating the same r_T is at most $max\{P0, P1, PC\}^ft$ which is negligible. Also, compared to querying another group with identical size and p, the probability of generating the same r_T is the same. Thus, the adversary has no clue to track a particular group.

In the physical access attack, the CT is not present. The analysis of the defense against this attack is similar to the replay attack, and we omit it for brevity.

V. Evaluation

In the section, we evaluate our scheme based on simulation. The major metrics we consider are efficiency in terms of the number of messages and bits transmitted for the authentication, and the overhead on the computational RFID tag CT. In our default setting, we consider a group of 100 passive tags ($n = 100$). W and ϵ are user-specified parameters, and their default values in our simulation are $W = 10$ and $\epsilon = 0.01$.

In terms of the messages exchanged, our solution is much more efficient than the individual authentication shown in the beginning of Section IV-A2 which costs 5 messages for each passive tag. In that case, letting a group of n passive tags authenticate R requires $5n$ messages. Our solution is scalable and requires only 6 round of message exchanges regardless of the group size.

Next, we evaluate the amount of data transferred in our solution in term of total number of bits. In the Step 1, CT transfers r' and ft to R, i.e., $16 + len(ft)$ bits assuming we use 16 bit random number in all steps. In Step 2, R broadcast r (16 bits) and collect $2ft$ bits (2 bits per slot) from passive tags which are further transferred to CT. In Step 3, CT sends a 16 bit random number and m bits certificate to R, which are further sent to all tags. In total, the number of bits are

$$16 + len(ft) + 16 + 2ft + 2ft + 16 + m + m.$$

Assuming the group contains no more than 1024 passive tags and replacing ft by $0.433n$, the above formulae is roughly

$$58 + 1.73n + 2m.$$

Thus, m is only variable that will change the performance. In the next, we mainly focus on evaluating m with different parameter settings.

First, we change the number of passive tags from 50 to 140 with an interval of 10, and calculate the minimum m in our solution. The results are illustrated in the following Fig. 3. The value of m is linearly increased when n increases range from 69 bits to 195 bits. The increasing rate is almost 14 bits per 10 additional tags.

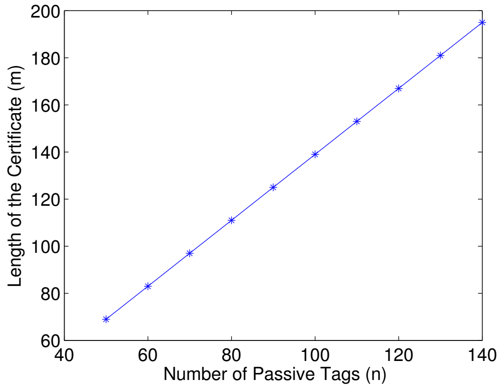

Fig. 3. Length of the certificate vs. number of passive tags

Second, we vary W, the maximum number of random numbers CT can pick for calculating m. Fig. 4 represents the resulting values of m. Intuitively, when W increases, CT is allowed to conduct more computation to find a good random number r' that can reduce the length of m while still satisfying the half '0's ('1's) property. The changing rate, however, is not as significant as in Fig 3. When W is doubled from 10 to 20, the length of m is reduced by 8 bits (from 139 to 131).

Furthermore, we measure the values of m with different confidence parameter ϵ. It is apparent that lager value of ϵ tolerate more exceptions when calculating the certificate, and thus yield shorter m. Fig. 5 confirms this intuition. But the change on m is minor. When ϵ is increased from 0.01 to 0.1, m is shrunk by 8 bits.

Finally, we simulate Step 3 on CT and evaluate the overhead for calculating the certificate C. We consider the default setting with 100 tags and the corresponding m calculated with $W = 10$ and $\epsilon = 0.01$ is 139. The simulation mimics CT's behavior and measure the number of trials needed to find an appropriate r' for generating C. We repeat the test for 10,000 times with the same setting and the results are

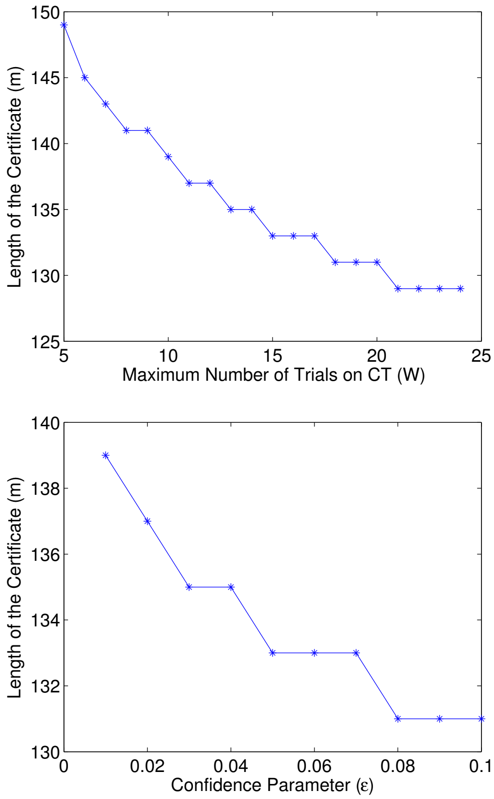

Fig. 5. Length of the certificate vs. confidence parameter (ϵ)

illustrated in Fig. 6. The average value is 3.68 and the number of violations, i.e., > 10 trials, is 76 (0.76%), which is lower than the confidence parameter ϵ.

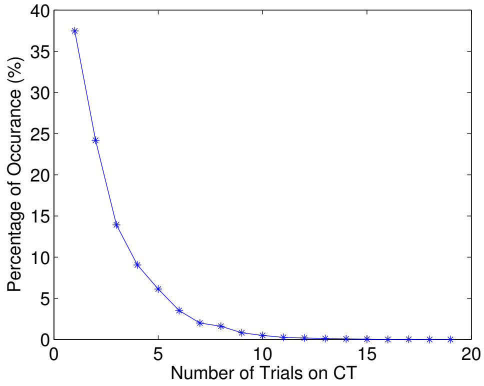

Fig. 6. Distribution of the number of trials.

VI. Conclusion

This paper proposes a novel solution to allow a group of RFID tags to authenticate an RFID reader. We consider a heterogeneous group consisting powerful CRFID tags and regular passive tags. Our new scheme is secure and extremely efficient for a large scale group of RFID tags based on our simulation results and security analysis.

Acknowledgements

This research was supported in part by NSF grants CNS 1156574 and CNS 1138963.

References

[1] T. Staake, F. Thiesse, and E. Fleisch, “Extending the epc network: the potential of rfid in anti-counterfeiting,” in *ACM symposium on Applied computing (SAC)*, 2005.

[2] A. Juels, “Rfid security and privacy: a research survey,” *IEEE Journal on Selected Areas in Communications (JSAC)*, 2006.

[3] M. Langheinrich, “A survey of rfid privacy approaches,” *Personal and Ubiquitous Computing*, 2009.

[4] K. Nejad, X. Jiang, and M. Kameyama, “High performance tag singulation for memory-less rfid systems,” in *IEEE International Conference on Communications (ICC)*, 2011.

[5] L. Xie, B. Sheng, C. Tan, H. Han, Q. Li, and D. Chen, “Efficient tag identification in mobile rfid systems,” in *IEEE International Conference on Computer Communications (INFOCOM)*, 2010.

[6] A. Juels, “”yoking-proofs” for rfid tags,” in *IEEE Conference on Pervasive Computing and Communications Workshops*, march 2004.

[7] L. Bolotnyy and G. Robins, “Generalized ”yoking-proofs” for a group of rfid tags,” in *International Conference on Mobile and Ubiquitous Systems: Networking Services*, july 2006.

[8] J. Saito and K. Sakurai, “Grouping proof for rfid tags,” in *International Conference on Advanced Information Networking and Applications (AINA)*, march 2005.

[9] M. Kodialam and T. Nandagopal, “Fast and reliable estimation schemes in rfid systems,” in *International Conference on Mobile Computing and Networking (Mobicom)*, 2006.

[10] T. Li, S. Wu, S. Chen, and M. Yang, “Energy efficient algorithms for the rfid estimation problem,” in *IEEE International Conference on Computer Communications (INFOCOM)*, 2010.

[11] W. Luo, S. Chen, T. Li, and Y. Qiao, “Probabilistic missing-tag detection and energy-time tradeoff in large-scale rfid systems,” in *ACM international symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2012.

[12] M. Kodialam, T. Nandagopal, and W. C. Lau, “Anonymous tracking using rfid tags,” in *IEEE International Conference on Computer Communications (INFOCOM)*, 2007.

[13] H. Han, B. Sheng, C. Tan, Q. Li, W. Mao, and S. Lu, “Counting rfid tags efficiently and anonymously,” in *IEEE International Conference on Computer Communications (INFOCOM)*, 2010.

[14] B. Sheng, C. C. Tan, Q. Li, and W. Mao, “Finding popular categories for rfid tags,” in *ACM international symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2008.

[15] C. Tan, B. Sheng, and Q. Li, “How to monitor for missing rfid tags,” in *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2008.

[16] T. Li, S. Chen, and Y. Ling, “Identifying the missing tags in a large rfid system,” in *ACM international symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2010.

[17] L. Yang, J. Han, Y. Qi, and Y. Liu, “Identification-free batch authentication for rfid tags,” in *IEEE International Conference on Network Protocols (ICNP)*, 2010.

[18] J. Gummeson, S. S. Clark, K. Fu, and D. Ganesan, “On the limits of effective micro-energy harvesting on mobile CRFID sensors,” in *ACM/USENIX International Conference on Mobile Systems, Applications, and Services (MobiSys 2010)*, Jun. 2010.

[19] M. Buettner, B. Greenstein, and D. Wetherall, “Dewdrop: an energy-aware runtime for computational rfid,” in *USENIX conference on Networked systems design and implementation (NSDI)*, 2011.

[20] C. Pendl, M. Pelnar, and M. Hutter, “Elliptic curve cryptography on the wisp uhf rfid tag,” in *Workshop on RFID Security (RFIDsec)*, 2011.

[21] A. Juels, R. L. Rivest, and M. Szydlo, “The blocker tag: selective blocking of rfid tags for consumer privacy,” in *ACM conference on Computer and communications security (CCS)*, 2003.

[22] M. Rieback, B. Crispo, and A. Tanenbaum, “Rfid guardian: A battery-powered mobile device for rfid privacy management,” in *Information Security and Privacy*, 2005, vol. 3574.

[23] G. Narayanaswamy, S. Jagannatha, and D. Engels, “Blocking reader: Design and implementation of a low-cost passive uhf rfid blocking reader,” in *IEEE International Conference on RFID*, 2010.