STUDY OF DIFFERENT RFID PROTOCOL BASED SOLUTIONS TO IMPROVING BACKEND PERFORMANCE USING LIGHTWEIGHT PRIMITIVES

[1]ANKESH KUMAR, [2]MANISH KUMAR SRIVASTAVA, [3]DR. PANKAJ KUMAR
[12]Research Scholar, [3]Professor
[123]Department of Physics
[123]B. R. A. Bihar University, Muzaffarpur
[3]kbsaku@gmail.com

***Abstract*—In this research paper, we focus our attention on study of different RFID protocol-based solutions to improving backend performance using lightweight primitives.**

***Index Terms*—RFID, Barcode, Frequency.**

I. Introduction

RFID tags can be identified automatically and simultaneously despite light. Whereas barcode requires an attendant (person) to scan codes through line-of-sight one by one to identify the codes. These features improve efficiency of an identification system significantly. Moreover, RFID tags can identify every single object with a unique ID, while barcode possesses limited capability in this aspect. Barcode is usually used to identify one type of item. For example, the same type of Cadbury Chocolates has the same barcode in a mall nowadays. In the future, it is possible that every chocolate bar will be attached with an RFID tag with a unique ID. There are a number of attacks on RFID that means that even assuming secure, private, and scalable RFID identification/authentication protocols at the application layer, there exists the need for designing protocols that thwart the challenging attacks. It is estimated that in the near future, millions of RFID tags will transmit information to thousands of RFID readers so as to enrich our interaction with the environment and make our processes more efficient and resilient. Therefore, gathering huge databases of trajectories by using the potential of the RFID technology to track moving objects will be a reality. Analyzing this kind of databases can lead to useful and previously unknown knowledge [1-5].

RFID has demonstrated its importance in a wide range of markets, including livestock identification and automated vehicle identification systems. This broad range of applications arises from its capability for tracking moving objects. Consequently, over the years, the number of solutions based on RFID has considerably grown. In fact, RFID systems are nowadays more related with business than with the military industry. As we know that in the 1980s and 1990s, RFID applications emerged in transport, access control, animal identification, tracking nuclear material and trucks and electronic toll collection [6-8]. This trend is increasing exponentially in the 21st century due to tag's price reduction [9] and RFID standardization [10]. In the future, appliances with RFID systems can be intelligent and provide a big convenience in people's everyday life.

II. BASIC PROTOCOL BASED SOLUTIONS

Protocol based RFID solutions rely on the RFID tags performing certain operations to provide the key security requirements to prevent unauthorized access, tracking, and skimming.

2.1 Different RFID Protocols

There are too many RFID protocols in the literature to be included in this chapter. Instead, we attempt to categorize them based on the focuses of these protocols, and highlight just a few works in each category. We have elected to avoid discussing RFID protocols designed from specialized applications such as banknotes [11] or supply chains. A good resource for the latest updates can be found in [11]

Improving backend performance: One approach lies in improving the performance of the backend server. From Figure 1, we see that the backend server needs to try all *(s:id)* pairs to determine the correct secret *s* to use in order to obtain the tag *id*. The reason that the RFID tag does not inform the backend server which secret *s* to use is to defend against illicit tracking. As a result, the RFID tag has to output a different random value each time it is queried. A more detailed analysis of protocols designed to alleviate the bottleneck can be found in [12].

One example of such protocols is a time-based solution proposed by [13]. The intuition is to let the backend server maintain a lookup table associated with the tag secret that is hashed with a timestamp, *(h(s,t):id)*. The backend server can pre-compute this table each time *t*. Each time the reader queries the tag, the reader will send the timestamp *t*, and the tag will respond with *h(s,t)*. This way, the backend server can obtain the corresponding *id* immediately using the lookup table. Later work by [14] and [15] improves on this approach.

Using lightweight primitives: In Figure 1, the RFID tag uses a hash function *h()* to protect its response. Given the hardware limitations of the RFID tag, an area of RFID research attempts to design solutions that do not use hash functions or symmetric keys to provide security. One popular approach is generally known as HB family of protocols which is after the authors [16]. The HB family of protocols uses scalar products and exclusive-ors to design their protocols. Work by [16] first proposed HB protocols that defend against different RFID attacks. A general survey of the HB family can be found in [16].

Generating random numbers: RFID protocols make extensive use of random numbers. A weak source of random numbers will allow the adversary to launch tracking. The use of random numbers to defend against tracking depends on the quality of the random number generated by the weak RFID tag. Work by Holcom *et al.*, J.Melia *et al.*, and Peris *et al.*explores this problem in further detail.

2.2 Detailed Look at a Simple RFID Protocol

Here we introduce a protocol modified from Tan et al. to illustrate how a protocol-based solution provides key RFID security requirements. Table 10.2 lists the notation used in this chapter. Figure 1: illustrates the protocol.

Table 1: Notations used

nt	Random number generated by RFID tag
nr	Random number generated by RFID reader
s	RFID tag secret
id	RFID tag id
h()	Cryptographic hash function
t	Timestamp

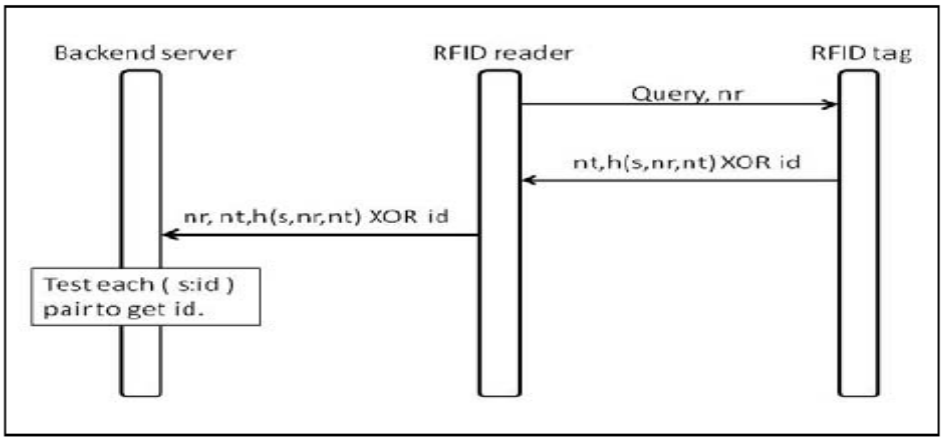

Figure 1: Basic protocol that defends against key RFID security requirements. Modified from Tan et al.

Random numbers from the reader and the tag are denoted as nr and nt respectively. The variables s and id denote the RFID tag's secret and id. Each tag has a unique secret and id that is assigned by the backend server. A conventional hash function is denoted as h().

From the protocol shown in Figure 1, we see that when the reader queries the tag, the reader will first transmit a random number, nr, to the tag. The RFID tag will respond by first generating its own random number, nt, and then compute a response to protect its tag id using h(s,nr,nt) XOR id. The reader then re-directs the tag's response, together with the random number it chose, to the backend server. The role of the backend server is to determine the id of the RFID tag. Since the backend server is responsible for all the RFID tags, it maintains a list of all tag secret to tag id pairs (s:id). Upon receive the message for the RFID reader, the backend server will know the two random numbers nt and nr chosen by the tag and reader respectively. From the list of (s:id) pairs, the backend server will hash the secret s to generate h(s,nr,nt) and XOR that against the response by the RFID reader, i.e. h(s,nr,nt) XOR h(s,nr,nt) XOR id. If the result matches the id in the list, the backend server will have determined the tag id. Otherwise, the backend server will continue to the next pair.

2.3 Security Analysis

Here we will analyze how the basic protocol in Figure 1 meets the key RFID security requirements. The first requirement is to prevent unauthorized access to the RFID tag information. We first consider an unauthorized reader querying the tag. The adversary will issue its own random number nr, and receive nt, h(s,nr,nt) XOR id from the tag. Since the backend server will not respond to the adversary, the adversary now has to determine id without any help from the backend server. The adversary succeeds if he is able to determine id from nt, h(s,nr,nt) XOR id. In order to get back id, we need to XOR with h(s,nr,nt) using the correct s value, but the adversary only knows nr and nt, and not s. Thus, the adversary is unable to obtain id. Since the protocol uses a conventional hash function such as SHA, the adversary cannot obtain s from h(s,nr,nt). The adversary can attempt to guess the value of s, but this can be defended against by using large enough values of s. Another method of unauthorized access is for the adversary to be eavesdropping when a legitimate reader is querying a tag. Since the wireless channel between the reader and tag is assumed to be insecure, the adversary is able to learn nr, nt, and h(s,nr,nt) XOR id. These pieces of information are similar to that obtained when the adversary queries the tag directly, which yields no useful information to the adversary [17].

The second requirement is to prevent illicit tracking. In this attack, the adversary needs to determine whether two tag responses belong to the same RFID tag. From Figure 1, we see that the RFID tag has two pieces of information that remains constant, the tag id and tag secret s. However, each time the tag replies to a query, the tag will select a different random number, nt, and thus, the resulting h(s,nr,nt) XOR id will

always be different for every response. This prevents any illicit tracking, since the adversary is unable to determine whether two responses are from the same RFID tag or not. This defense remains valid even if the adversary can select its own nr value.

The third key requirement is to prevent or detect skimming. The adversary launching a skimming attack will observing the responses of a real RFID tag in attempt to create a fake tag that can pass off as a real tag. In the basic protocol, the adversary is able to observe the return value of nt, h (s, nr, nt) XOR id, However, it is unable to learn s or id based on the response. The adversary thus can only store h(s,nr,nt) XOR id directly into a fake RFID tag. This skimming attack will be detected when a legitimate reader queries the RFID tag. The legitimate reader will issue its own random number, which we denote as nr' to distinguish from the earlier nr observed by the adversary. Since the fake tag does not know s or id, the fake tag can only return h (s, nr, nt) XOR id, and not the correct h(s, nr',nt) XOR id. Since the backend server will attempt to test using nr' and not nr, this leads the backend server unable to find a correct (s, id) pair. Thus, the skimming attack is detected [18].

III. ADVANCE PROTOCOL BASED SOLUTIONS

Beyond the key RFID security requirements, there are some other RFID security requirements. This section discusses some protocols that address these requirements.Note that the protocols presented here may not necessarily meet all the key security requirements because these advanced protocols are generally designed to address specific issues or applications.

3.1 Defending against Mafia fraud

The mafia fraud has emerged as a challenging problem for RFID applications. This type of attack cannot be defended by the protocols mentioned earlier because a legitimate RFID reader is accessing data from a legitimate RFID tag. In other words, this type of attack can still work even if both the reader and the tag authenticate each other. This is illustrated using the basic protocol shown in Figure 1.

Consider an application which uses RFID tag to open a door. The RFID reader will first read the tag and then transmit the information to the backend server. Once the backend server verifies the tag is legitimate, the door will open. To launch a mafia fraud attack, the adversary will first be in close proximity with a person holding a legitimate RFID tag. We refer to this person as the target. The adversary's accomplice will be standing near to the door. When the legitimate RFID reader issues a query, the adversary's accomplice will relay this message to the adversary, who will in turn issue it to the target's RFID tag. The target's RFID tag will respond to the adversary, who will then relay this back to his accomplice to transmit to the RFID reader. Since the RFID reader obtains the response from a legitimate RFID tag, the door will open and the adversary can gain access. Therefore, the choice of protocol does not defend against this type of attack. The intuition behind defending against a mafia fraud is to accept an RFID tag's response if it is both valid and timely. Since the wireless transmission speed, the RFID tag computational time, and distance between the reader and tag are known, the RFID reader can estimate the amount of time needed to receive a response. If the arrival of the RFID tag response is late, the reader can deduce the distance travelled is longer than what is allowed, and thus reject the tag answer [19].

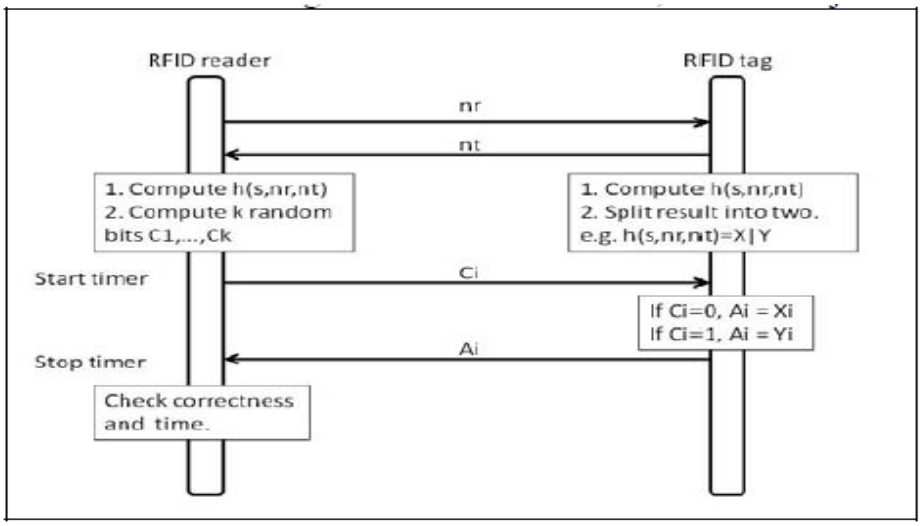

Figure 2: Distance bounding protocol from Hancke et al.

We assume that the reader already knows the tag secret s. We retain the notation from Table 1. X and Y are the bit string resulting from dividing h(s,nr,nt) into two. The protocol will repeat itself until Ck is transmitted from the reader to the tag. The variable k is a system defined parameter [20].

One of the main solutions against the mafia fraud is from Hancke et al. [21] and is shown in Figure 2. We assume the system will define a maximum distance d, over which the reader is not supposed to authenticate a tag. In the protocol, we see that both reader and tag exchange random numbers. Assuming that the reader knows the tag secret s, both entities can compute h(s,nr,nt). The tag will split this result into two queues, X and Y. At the same time, reader than generates a k bit challenge, C1,...,Ck, where k is a system defined parameter. The idea is that the reader will challenge the tag by sending over a bit Ci. If the Ci is 0, the tag will set Ai to the bit from queue X, and vice versa. The reader will keep the time it takes from sending Ci and receiving the Ai. A legitimate RFID tag that is within the approved distance will respond within the allocated time limit. A legitimate RFID tag that is further away will take a longer time to respond, due to the longer distance travelled, and thus is detected. More recent work on this topic can be found in [22] and [23]. An interesting idea of doing distance bounding for a group of RFID tags instead of just two tags has been proposed by Capkun et al.

3.2 Grouping Proofs

Grouping proofs are required for a reader to prove to the backend server that a set of RFID tags are physically close to each other. This type of proof typically requires a more advanced RFID tag that is able to maintain an atomic counter and a countdown timer. Each time an RFID tag is queried, the RFID tag will increment its counter after its timer expires. This is an atomic operation that cannot be disrupted. The intuition is for the reader to query each RFID tag one after the other to collect the responses to generate a proof before the timer expires [24].

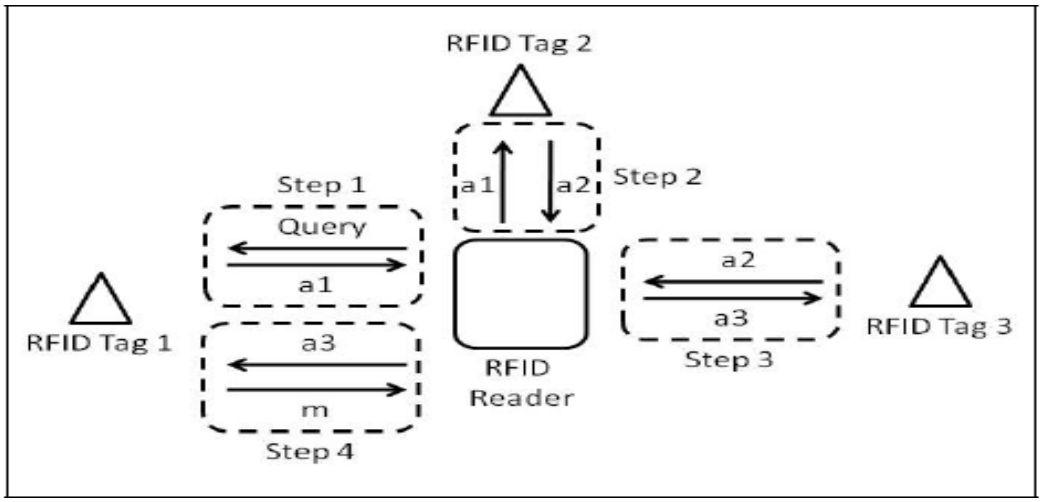

Figure 3: Grouping proof for 3 RFID tags.

After Step 4, the reader generates the roof, which is then transmitted to the backend server for verification. Steps 1 to 4 have to be completed before the RFID tag timer expires.

Figure 3 illustrates a grouping proof from Bolotnyy et al. [25]. The proof is to demonstrate that RFID Tag 1, Tag 2, and Tag 3 are present at the same time. To generate the proof, the reader will first query the first tag, Tag 1, and receive a1 where a1={id1, c1, h(s1,c1)}. At this point, the timer for Tag 1 has started. The reader will continue to send a1 to Tag 2 and receive a2 back (Step 2). The value of a2 is {id2,c2,h(s2,c2,a1)}. The reader will send a2 to Tag 3 and get back a3 (Step 3), which is {id3,c3,h(s3,c3,a2)}. At this time, the reader has collected responses from all the tags, and will send a3 back to the first tag, Tag 1. This has to be done before Tag 1's timer expires. If the reader is successful, the reader will obtain the message m, where m=h(a1,a3,s1). RFID Tag 1 will not respond if the timer expires. The reader will then submit the proof p to the backend server for verification, where p = {id1,id2,id3,c1,c2,c3,m}. Since the backend server knows the secrets for each of the ids, the backend server can determine whether c1, c2, c3, and m are valid.

IV. CONCLUSIONS

We can see that if the RFID reader does not complete the proof in time, the reader will be unable to return the correct m value to the backend server because computing *m* requires *s1*, which is only known to the Tag 1 and the backend server. RFID systems are subject to plenty of attacks, from attacks operating on the physical layer to attacks exploiting weaknesses on those protocols executed at the application layer [20, 21, 22]. Physical attacks may be as simple as wrapping an RFID tag in aluminium foil, which potentially causes denial of services (DoS) because readers will be not able to communicate with such tag. Other physical attacks are more sophisticated (e.g. jamming attacks that permanently damage radio devices or side-channel attacks that obtain information from the physical implementation of cryptosystems). The most relevant attack to RFID systems is the so-called spoofing or impersonation attack. In this attack, an adversary is able to clone a tag without physically replicating it. By doing so, the adversary gains the privileges of such tag, which is considered an important security threat for almost every RFID application. The worst situation occurs when the adversary is able to break the cryptosystem used during the authentication process (total break), i.e. the adversary gains knowledge of the authentication protocols and the secrets. After then we turned our attention to security protocols that provide more advance security requirements of preventing attack and providing grouping proofs. Finally, we concluded by studying a commercial RFID security protocol, the basic access control standard, used in passports.

References

[1] RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication, Third Edition. Klaus Finkenzeller, © 2010 John Wiley & Sons, Ltd

[2] Ari Juels. RFID Security and Privacy: A Research Survey. IEEE Journal on Selected Areas in Communications, 24(2):381–394,February 2006.

[3] David Molnar and David Wagner. Privacy and Security in Library RFID: Issues, Practices, and Architectures. In Birgit Pfitzmann and Peng Liu, editors, Conference on Computer and Communications Security – ACM CCS, pages 210–219, Washington, DC, USA, October 2004. ACM, ACM Press.

[4] K. Takaragi, M. Usami, R. Imura, R. Itsuki, and T. Satoh. An ultra small individual recognition security chip. IEEE Micro, 21(6):43–49, 2001.

[5] RFID Security and Privacy:, A Research Survey, Ari Juels, RSA Laboratories, 28 September 2005

[6] RFID Journal Staff. Air Canada gets asset tracking. RFID Journal, arch 2003. http://www.rfidjournal.com/article/print/335.

[7] RFID Journal Staff. Boeing finds the right stuff. RFID Journal, September 2003. http://www.rfidjournal.com/article/print/715/.

[8] Bob Violino. Farm harvests RFID's benefits. RFID Journal, March 2004. http: //www.rfidjournal.com/article/print/810.

[9] Colin C Haley. Are you ready for rfid?. InternetNews.com, November 2003. http://www.internetnews.com/infra/print.php/3109501.

[10] Pedro Peris-Lopez, Julio César Hernández Castro, Juan M. Estévez-Tapiador, and Arturo Ribagorda. RFID systems: A survey on security threats and proposed solutions. In PWC, pages 159–170, 2006.

[11] Basel Alomair and Radha Poovendran. Privacy versus scalability in radio frequency identification systems. Computer Communications, 33(18):2155–2163, 2010.

[12] Privacy in RFID and Mobile Objects, Ronaldo Trujillo-Rasua, Dr. Agusti Solanas and Dr. Josep Domingo-Ferrer.

[13] Melanie Rieback, Bruno Crispo, and Andrew Tanenbaum. The Evolution of RFID Security. IEEE Pervasive Computing, 5(1):62–69, January–March 2006.

[14] Benny Bing. Broadband Wireless Access. Norwell, MA, USA, 2000. Kluwer Academic Publishers.

[15] Robert M. Metcalfe and David R. Boggs. Ethernet: distributed packet switching for local computer networks. Communication. ACM, 26(1):90–95,1983.

[16] Sanjay Sarma, Stephen Weis, and Daniel Engels. RFID Systems and Security and Privacy Implications. In Burton Kaliski, C¸ etin Kaya c¸o, and Christ f Paar, editors, Cryptographic Hardware and Embedded Systems -CHES 2002, volume 2523 of Lecture Notes in Computer Science, pages 454– 469, Redwood Shores, California, USA, August 2002. Springer-Verlag.

[17] H. Stockman. Communication by means of reected power. Proceedings of The Institute of Radio Engineers, 36(10):1196{1204, October 1948.

[18] J. Landt. Shrouds of time: The history of RFID. 1 October 2001. http://www.rfidconsultation.eu/docs/ficheiros/shrouds_of_time.pdf.

[19] M. Rieback, B. Crispo, and A. Tanenbaum. The Evolution of RFID Security. IEEE Pervasive Computing, 5(1):62{69, January {March 2006.

[20] EPC global. EPC Standard Specification, version 1.1 rev. 1.24, April 2004.

[21] J. Landt. The history of RFID. Potentials, IEEE, 24(4):8–11, 2005.

[22] RFID News: The Five-Cent Tag is Here, the Five-Cent Tag is Here! Well, almost. http://www.scdigest.com/assets/On_Target/09-01 72.php?cid=2201&ctype=content.

[23] EPCGlobal, 2004. http://www.gs1.org/epcglobal.

[24] Pawel Rotter. A Framework for Assessing RFID System Security and Privacy Risks. IEEE Pervasive Computing, 7(2):70–77, June 2008.

[25] B. Glover and H. Bhatt. RFID Essentials. O'Reilly, Gravenstein Highway North, Sebastopol, CA, USA.