Secure and Efficient Offline RFID Authentication for Anti-counterfeiting

Hong Jun Choi, Sung Je Hong, Jong Kim
Department of Computer Science and Enginerring,
Pohang University of Science and Technology (POSTECH)
{formetel, sjhong, jkim}@postech.ac.kr

Abstract

The RFID system is a promising technology for anti-counterfeiting, but RFID tags can suffer from the security problems such as cloning or impersonation. A reliable authentication for tags is critical to deploy RFID system especially for anti-counterfeiting. Most of conventional RFID authentication protocols are based on the pre-shared secret between the tags and the back-end server. However, the back-end server demands a high cost to deploy and operate in practice and the reader has to always communicate with the back-end server in a real-time manner. Recently, a few offline protocols have been proposed for anti-counterfeiting. However, these protocols are quite expensive in respect to tag costs and they require an additional trusted third party to manage the pre-defined access lists. In this paper, we propose an offline RFID authentication protocol that is secure and efficient. The proposed protocol does not need a back-end server or trusted third party. With the security and efficiency analysis, we show that our protocol is secure against common attacks and efficient in terms of tag costs.

1. Introduction

Counterfeiting and piracy are a global phenomenon affecting a wide range of the world economy. Since 1982, the global trade in counterfeiting goods has increased from $5.5 billion to approximately $600 billion annually, which amounts to 5%~7% of the world trade [4][22]. The economic loss due to counterfeiting and piracy will be more than $1,500 billion in 2025 [1]. The threat of counterfeiting is not limited to the products. It harms society in many ways, for example, loss of employment opportunities, consumer health and safety, deterioration of tax base, etc. [10][11]. Companies are becoming aware of counterfeiting activities and the associated social harm. They have used several anti-counterfeiting measures such as holograms and special packaging designs but these techniques are easily cloneable.

Radio Frequency Identification (RFID) is a promising measure to fight against counterfeiting and it is receiving growing acceptance as an anti-counterfeiting measure in many industries [15]. RFID system [3][18] has a number of appealing features for anti-counterfeiting compared to other identification technologies such as the barcode system [6]. First, the RFID system can identify a unique product by the attached RFID tag while the barcode system only checks the type of product from the printed barcode. Barcodes, being optically scanned, require line-of-sight contact with readers. In contrast, RFID tags are readable without line-of-sight contact. This helps to automate and check large-scales for the item-level identification. Second, the tags have a unique factory programmed chip serial number (TID), which is similar to the unique MAC address of a network interface card. To clone a tag's TID would therefore require designing the chip. Third, tags can perform some logical operations. The logical operations make use of cryptographic for authentication while keeping the critical information secret to increase resistance against cloning.

However, the RFID tags may pose some security issues. Basically, the communication between a reader and a tag is performed via radio frequency. This implies that anyone within the signal range can obtain all transmitted data by eavesdropping. The adversary can make a cloned tag by simply copying this information [16][19]. The traditional wireless systems use the strong cryptography to solve these problems. However, it is difficult for the RFID system to adopt these strong functions since the tags have limited resources and the additional security functions are directly related to the tag cost. In order to design an efficient RFID authentication protocol, the tag requirements, for example, the ability to perform

logical operations, pseudo-random number generator and cryptographic encryption, should be reduced [8].

The RFID system usually needs a back-end server. The single server manages all of the tag's information that is used for tag authentication. If the RFID system operates in the multiparty supply chain, then the complicated technologies are needed for the server and database management. Furthermore, once the back-end server is compromised by an attacker, the entire system may fail, or all tag's information may leak, which cause serious privacy problems [26].

1.1. Related Work

There have been a number of RFID authentication protocols since RFID security issues had been introduced in [19]. RFID authentication protocol can be categorized into online and offline. Most of the existing RFID authentication protocols are online authentication protocols. The tags have the pre-shared secret with a back-end server. The reader authenticates tags with the pre-shared secret (e.g., [26, 5, 17, 12, 23, 25, 7, 2, 27]). Online protocols have inherent problems since they require a back-end server. First, a reader has to connect a back-end server because the reader transmits the tag's information to the back-end server for authentication. In other words, authentication cannot be performed without a connection. If the server is down, the entire system would break down. Second, these approaches demands a high cost to deploy in multi-party supply chain because they require great efforts for the server management. Third, all tag information would be stored in a back-end database. These may cause a "big brother" [13] problem, which is described as an overly-controlling authority in terms of privacy. Moreover, once a back-end server is compromised, all of the tag's information falls to an attacker. The threat of privacy grows when a tag serial number is combined with the personal information.

To eliminate these inherent problems in the online protocol, the authentication protocols need to be offline. In other words, the authentication is carried only between a reader and a tag. In an offline system, there is no complicated server and database management, or a single point of failure and "big brother" problem.

Recently, Tan et al. [21] proposed a simple alternative of an online protocol, which downloads the tag's information (access list) from an additional trusted-third-party (TTP) before the authentication process. The access list contains information on the RFID tags, to which a particular reader can access. This protocol still needs central back-end database.

The protocol in [24], believed to be the first offline authentication protocol, is based on PUF (Physically Unclonable Function). PUF provides unique responses to various challenges. While this offline protocol does not need a back-end server, the tags are prohibitively expensive. Each tag has a public-private key pair and the reader executes zero-knowledge authentication protocol. This protocol requires expensive public-key operations.

1.2. Our Contribution

This paper proposes an efficient RFID authentication protocol, which does not require a connection to back-end servers for tag authentication. In the protocol, the reader can authenticate the tag that stores data related to an issuer and itself securely in its memory. In other words, the protocol has no database and server management for authentication. In addition, each tag has a single secret key instead of a public-private key pair. Also, the standard identification scheme, that is quite expensive, is not used. The tag performs a simple cryptographic function that is a special hash function. As a result, our system can be deployed easily even in the multi-party supply chain because whoever that has an offline reader can authenticate a genuine tag without a back-end server.

1.3. Organization

In Section 2, we discuss the security requirements of RFID system for anti-counterfeiting. We then present our secure and efficient offline RFID authentication protocol in Section 3. In Section 4, we present the security and efficiency analysis of our protocol. Finally, we summarize the paper in Section 5.

2. Security Requirements

In this section, we describe properties that should be guaranteed for secure offline RFID authentication. The RFID tag has various vulnerabilities since it communicates with the reader via insecure channels. An attacker can overhear all or part of the data transmitted between the tag and reader, and then he/she may use the data for cloning or impersonating the valid tag. Therefore, we consider the following requirements from the cryptographic point of view.

Secret Data (key) Confidentiality: The secret information of the tag must be kept secure to prevent cloning and impersonation. Moreover, all transmitted data between a tag and a reader should not reveal any information about the secret. In other words, an

attacker must not be able to recover secret information of the tag from the captured data.

Data Integrity and Authentication: The original issuer stores the data related to the genuine tag into the memory of the tag. The data should be protected from modification and even so a reader should be able to check it. An attacker also may produce the data related to the tag by himself and store it into the fake tag. To protect this attack, origin authentication of the stored data should be guaranteed.

Prevention of Impersonation: An attacker can perform a simple replay attack using data from a previous protocol execution, or he/she can produce a legitimate response in some ways (spoofing attack). Therefore, an attacker must not be able to impersonate the genuine tag without knowing the secret key.

3. The proposed Protocol

In this section, we describe our RFID authentication protocol. We first present the system model and introduce the commutative hash function used in our protocol. We then describe our secure and efficient offline RFID authentication protocol.

3.1. System Model

Our RFID system includes three entities, a reader, a tag and an issuer. The reader performs authentication of the tag. The tag is attached to the products inseparably. The issuer, who is a manufacturer of the product, stores the data related to authentication into the tag. We assume that the reader has more powerful resources compared to the tag. Therefore, in our protocol, the reader can perform more expensive cryptographic operations than the RFID tag. The reader can perform the following logical operations; 1) a public key decryption (e.g. RSA) to verify and recover data from the issuer's signature stored in the tag, 2) a pseudo-random number generation to make challenges and, 3) a commutative hash function to check the responses.

Each tag has a unique identity and a secret key. The identity is the tag number, which is a factory programmed chip number (TID). The secret key assigned by the issuer should be protected from the malicious attacker and only be used inside the tag. The tag also has a read-only memory (ex, EEPROM) to save data related to its authentication. The logical operation performed by the tag is a single commutative hash function. We assume that the tag is bound to the products inseparably. In other words, the tag does not operate properly when it is detached from the product.

The issuer has a public-private key pair and digitally signs the *tag-related data* under a PKI system. He/She stores the data into the tag via secure channel in the enrollment phase.

3.2. Protocol Overview

The proposed protocol consists of two phases, the enrollment phase and the verification phase. The enrollment phase is performed once between the issuer and the tag initially. When the issuer manufactures a product and attaches the tag to the product, he/she stores the unique data, i.e., issuer's signature on *tag-related data*, via a secure channel into the tag. In the verification phase, the reader verifies a tag with the stored data based on a challenge-response when a customer wants to authenticate the product.

The *tag-related data* is the keyed hash of tag's id using tag's secret key, which is made by the legitimate issuer in the enrollment phase. There is a trust chain on that stored data since the *tag-related data* is generated from the tag's secret with a hash function and it is signed by issuer's private key. That is, the only legitimate tag can have its own secret and the only legitimate issuer can generate his/her signature on that secret.

The trust data in the tag is used for authentication later. The reader recovers the *tag-related data* from the issuer's signature with the public key and then generates a challenge with a partial of that data. The tag responds to the challenge using its own secret key and the reader compares the response with the *tag-related data* by using a hash function. If they are the same, then the reader assumes that the tag is a legitimate one.

3.3. Commutative Hash Function

Definition 1 : A commutative hash function is a one-way hash function H which has the following property for a given message M and any two keys K_1 and K_2,

$$H(K_1, H(K_2, M)) = H(K_2, H(K_1, M)) . \quad (1)$$

When commutative hash operations are applied on a message two times with two keys, the applying sequence does not change the result. This property of commutative hash function can be used to check if two values are equal without revealing them. Suppose that A has a secret key K_A and a message M_A and B has a secret key K_B and a message M_B. Now, A and B want to check if the messages are equal without revealing them to each other. A sends $H(K_A, M_A)$ to B, and B sends

$H(K_B, M_B)$ to *A*. *A* and *B* hash the received message with their own secret keys. At this point, if $H(K_B, H(K_A, M_A))$ is equal to $H(K_A, H(K_B, M_B))$, then *A* and *B* can know M_A is equal to M_B according to the property above. In this scheme, M_A and M_B are not revealed to either of them.

The Pohlig-Helman algorithm[14], which is one of the popular commutative ciphers, can be employed to construct the commutative hash function. A prime p is chosen, along with a secret key k, $1 \le k \le p-2$. A message *M* is hashed as $H(k, M)=M^k \bmod p$. Suppose $1 \le K_A, K_B \le p-2$, then we have

$H(K_B, H(K_A, M))=H(K_B, M^{K_A})=M^{K_A K_B} \bmod p$

$H(K_A, H(K_B, M))=H(K_A, M^{K_B})=M^{K_B K_A} \bmod p$

$\therefore H(K_B, H(K_A, M))= H(K_A, H(K_B, M))$

However, modular exponentiation is extremely expensive to compute. If point M_P is defined in an elliptic curve[9], then we can decrease the computational overhead with similar security.

$H(K_B, H(K_A, M_P))=H(K_B, K_A M_P)=K_B K_A M_P$

$H(K_A, H(K_B, M_P))=H(K_A, K_B M_P)=K_A K_B M_P$

$\therefore H(K_B, H(K_A, M_P))= H(K_A, H(K_B, M_P))$

3.4. Notations

Table 1 summarizes the notations used in our protocol.

Table 1: Notations

Notation	Interpretation
M	Issuer (Manufacturer)
R	RFID deader
T	RFID tag
H()	Commutative hash function
id	Unique tag's serial number
k	Tag's secret key
n_R	Nonce from the reader
sk, pk	Issuer's private key and public key
sig	Issuer's signature
$E_{sk}()$	Public key Encryption with sk
$D_{pk}()$	Pubilc key Decryption with pk
PRNG	Pseudo random number generator
‖	Concatenate function

3.5. The Proposed Protocol

We describe the process of the proposed protocol as shown in Fig. 1.

Enrollment phase

Step 0. *M* generates *k* from *PRNG* and computes *H(k, id)* which is the *tag-related data*. *M* then generates *sig* on *id*‖*H(k, id)* with *sk*. The *sig* and *k* are stored into the *T*'s memory securely.

Verification phase

Step 1. *R* sends a request to *T* to start verification process.

Step 2. *T* sends back to R *sig* and *id* stored in its memory. *R* verifies whether *sig* is valid or not by using *pk* and *id*. If *sig* is invalid, then *R* declares that *T* is invalid. Otherwise, *R* recovers *H(k, id)* from *sig*.

Step 3. *R* generates n_R by using *PRNG*. Then, *R* generates a challenge $c=H(n_R, id)$, and sends *c* to *T*

Step 4. *T* computes the response $r=H(k, H(n_R, id))$ from the received challenge $H(n_R, id)$ with *k* and responses *r* to *R*.

Step 5. *R* computes $r'=H(n_R, H(k, id))$ and compares it with the received $r=H(k, H(n_R, id))$. If they are the same, then *R* declares that *T* is valid. Otherwise, *R* declares that *T* is invalid.

The main idea of our protocol is to use the issuer's signature on the *tag-related data* and the tag's response using its own secret key. The tag's secret key and the issuer's signature can form a trust chain. The only valid tag can have its secret and the only valid issuer can digitally sign on the tag's secret.

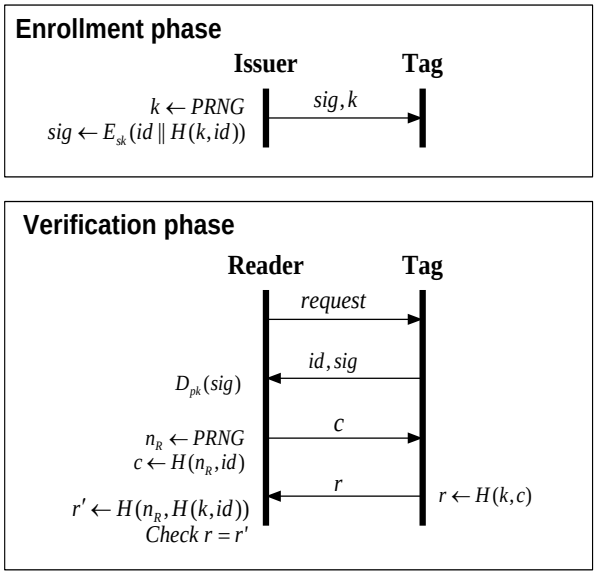

Fig. 1: The proposed protocol

4. Security and Efficiency Analysis

In this section, we analyze the security of the proposed protocol according to the security requirements introduced in Section 2: secret data (key) confidentiality, data integrity, authentication, and prevention of impersonation. We then analyze the

efficiency in terms of the need for a back-end server and the tag requirements.

4.1 Security Analysis

Key Confidentiality: The tag's secret key must be kept secure to protect tag cloning and impersonation. Although attacker can easily recover the tag-related data, *H(k, id)*, from the sig stored in the tag, *H(k, id)* does not reveal any information of the secret key because of the one-way property of hash function. Moreover, the tag's response, $H(k, H(n_R, id))$, is computed from the random nonce n_R, which can be changed every session and it is the hash of the challenge with the secret key. Although the hash can guarantee a strong security level, there can be some attacks to the hash function such as the chosen text attack with offline brute force search. To prevent this attack, we can improve our protocol in which the tag makes a response with a self-chosen random number. In Step 4, the tag generates the nonce n_T and hashes its response. $H(n_T, r)$ and *r* are transmitted to the reader. In Step 5, the reader computes $H(n_T, r')$ and checks it with the response.

Data Integrity and Authentication: The data stored in the tag may be modified by an attacker, or attacker may make the data strategically for cloning and impersonation. However, the stored data is the issuer's signature. An issuer signs this data with its own private key in the enrollment phase, which guarantees data integrity and origin authentication.

Protection of Impersonation: An attacker can scan all the messages transmitted between the reader and the tag. Then he/she can try to replay the response in order to impersonate the legitimate tag. The reader uses a randomly generated challenge that changes at every new session. Hence the response is also changed at every session. Therefore, an attacker cannot impersonate the valid tag by directly reusing the previous response. The response, $H(k, H(n_R, id))$, from random challenge $H(n_R, id)$ can be computed only by the tag with its secret key *k*. Without knowing the tag's secret key, an attacker cannot compute the valid response, which prevents the spoofing attack. An attacker can generate key *k* strategically by oneself and store the *tag-related data*. However, the attacker cannot make the valid issuer's signature.

4.2 Efficiency Analysis

We compare our protocol with other offline protocols in terms of the need for a back-end server and the tag cost. The protocol in [21] cannot fully obtain the benefits of the offline protocol. In [21], the reader does not need real-time connectivity. However, the reader has to download and maintain the access list from TTP before the verification phase. In other words, this protocol still needs a back-end database. In addition, the access list has the following problems inherently. First is a list update and revocation. The TTP has to update the access list whenever the association between tags and readers are changed. Then readers have to download the updated list. Second is the scalability. The size of access list is directly proportional to the number of tags. This is also related to the size of a database and reader's memory.

The offline protocol in [24] needs no back-end server, but it is expensive to make such a tag. First, considering the functions to be performed, , the tag needs a random value generation and a key generation by using PUF. To use PUF, the tag performs key extraction algorithm to extract a key from noisy data. In the verification process, this protocol performs a zero-knowledge identification algorithm such as the Okamoto's identification protocol which needs a public-private key pair per each tag. Considering the number of operations, the tag performs two random number generations, two public-key encryptions, and one modular multiplication.

On the contrary, our protocol does not need a back-end server. The data to authenticate the tag is stored in the tag's memory, thus the reader does not need to manage additional data before authentication. In terms of the tag costs, the tag performs only one commutative hash operation to make the response in the verification phase, which needs only one scalar multiplication if it is implemented on the elliptic curve cryptography. Elliptic curve cryptography has a significantly smaller key than in other competitive systems with similar levels of security. Smaller key sizes produce faster computation and reduction in processing power, storage space and bandwidth. Recently, Batina et al.[20] presented an elliptic curve processor implementation suitable for RFID tags requiring between 8500 and 14000 gates. This shows that the elliptic curve cryptosystems are feasible on RFID. However, the operation of the elliptic curve cryptography is still expensive relative to RFID tag. Therefore, finding an efficient commutative hash function still is remained as our future work.

5. Summary

In this paper, we proposed a secure and efficient offline authentication protocol for anti-counterfeiting. The protocol consists of two phases, the enrollment

phase and the verification phase. During the enrollment phase, the issuer stores his/her signature on *tag-related data* into the tag's memory. During the verification phase, the reader authenticates the validity of the tag, with the tag-related data stored and the random challenge using a commutative one-way hash function.

Our protocol can protect against basic attacks such as the replay attack or the spoofing attack for anti-counterfeiting. An attacker is not able to impersonate the legitimate tag without knowing the secret key. In contrast to most of the previous protocols, we need no back-end server. Therefore, our protocol eliminates the complicated server and database management, the single points of failure, and the big brother problem. Moreover, the tag costs are reduced compared to the previous offline protocols. The tag has a single secret key and performs only one commutative hash function. A future research direction is to enhance our scheme to solve the privacy problems mentioned in [26].

Acknowledgments

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRCs(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement)" (IITA-2007-(C1090-0701-0035) and IITA-2007-(C1090-0701-0045))

References

[1] "Business Action to Stop Counterfeiting and Piracy", BASCAP Case Study Database, http://www.iccwbo.org/bascap/id6170/index.html

[2] Dimitriou, T., "A Secure and Efficient RFID Protocol that could make Big Brother (partially) Obsolete", International Conference on Pervasive Computing and Communications, PerCom 2006, IEEE Computer Society Press, 2006

[3] Klaus Finkenzeller, RFID Handbook, John Wiley and Sons, 2003

[4] International Chamber of Commerc, "The fight against piracy and counterfeiting of intellectual property", submitted to the 35th ICC World Congress, Marrakech, June 2004

[5] Juels, A., "Minimalist Cryptography for Low-Cost RFID Tags" International Conference on Security in Communication Networks, SCN 2004, Springer-Verlag, 2004, 3352, 149-164

[6] Juels, A., "RFID security and privacy: a research survey", Selected Areas in Communications, IEEE Journal on, 2006, 24, 381-394

[7] Juels, A., Weis, S., "Authenticating Pervasive Devices with Human Protocols", Advances in Cryptology. CRYPTO'05, Springer-Verlag, 2005, 3126, 293-308

[8] Lehtonen, M., Staake, T., Michahelles, F., Fleisch, E., "From Identification to Authentication - A Review of RFID Product Authentication Techniques", RFIDSec 06, July 2006.

[9] Lopez, J., Dahab, R., "An Overview of Elliptic Curve Cryptography", Institute of Computing, Sate University of Campinas, Sao Paulo, Brazil, Tech. Rep., May, 2000

[10] National Camber Foundation, "Countering Global Counterfeiting and Piracy", Summer, 2006.

[11] National Camber Foundation, "What are Counterfeiting and Piracy Costing the American Economy?", 2005

[12] Ohkubo, M., Suzuki, K., Kinoshita, S., "Cryptographic Approach to ``Privacy-Friendly'' Tags", RFID Privacy Workshop, 2003

[13] Orwell, George, "Nineteen Eighty-Four", 1949

[14] Pohlig, S.C., Helman, M.E., "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance", IEEE transactions on Information Theory, vol. IT-24, pp. 106-110, 1978

[15] RFID Journal, RFID news, http://www.rfidjournal.com

[16] Rieback, M.,Crispo, B., Tanenbaum, A., "The Evolution of RFID Security", IEEE Pervasive Computing, 2006, 5, 62-69

[17] Saito, J., Ryou, J., Sakurai., "Enhancing Privacy of Universal Re-encryption Scheme for RFID Tags", Embedded and Ubiquitous Computing, EUC 2004, Springer-Verlag, 2004, 3207, 879-890

[18] Sarma, S., Weis, S., Engels, D., "RFID systems and Security and Privacy implications", Cryptographic Hardware and Embedded Systems, CHES2002, Springer-Verlag, 2002, 2523, 454-469

[19] Sarma, S., Weis, S., Engels, D., "Radio-Frequency Identification: security Risks and Challenges", Cryptobytes, RSA Laboratories, 2003, 6, 2-9

[20] Batina, L., Guajardo, J., Kerins, T., Mentens, N., Tuyls, P., Verbauwhede, I., "An Elliptic Curve Processor Suitable For RFID-Tags", 2006

[21] Tan, C. C., Sheng, B., Li, Q., "Serverless Search and Authentication Protocols for RFID", International Conference on Pervasive Computing and Communications, PerCom 2007, IEEE Computer Society Press, 2007

[22] The International AntiCounterfeiting Coalition Inc.(IACC), "GET REAL – The Truth About Counterfeiting" http://www.iacc.org/counterfeiting/counterfeiting.php

[23] Tsudik, G., "YA-TRAP: Yet Another Trivial RFID Authentication Protocol", International Conference on Pervasive Computing and Communications, PerCom 2006, IEEE Computer Society Press, 2006

[24] Tuyls, P., Batina, L., "RFID-Tags for Anti-Counterfeiting", Topics in Cryptology. CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006, Springer-Verlag, 2006

[25] Vajda, I., Buttyán, L., "Lightweight Authentication Protocols for Low-Cost RFID Tags", Second Workshop on Security in Ubiquitous Computing -- Ubicomp 2003, 2003

[26] Weis, S., Sarma, S., Rivest, R., Engels, D., Hutter, D., "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems", International Conference on Security in Pervasive Computing, SPC 2003, Springer-Verlag, 2003, 2802, 454-469