Applied Mathematics & Information Sciences
An International Journal

Secure and Lightweight Authentication Protocol for Mobile RFID Privacy

Hyeong-Chan Lee, TaeYang Eom, and Jeong Hyun Yi

Department of Computer Science and Engineering, Soongsil University, Seoul, Korea

Received: 28 Jan. 2012; Revised 2 Jun. 2012; Accepted 11 Jul. 2012
Published online: 1 Jan. 2013

Abstract: A mobile RFID system is a radio frequency identification technology that allows users to read the information on its tags. When this free reading function is combined with a mobile RFID system featuring radio frequency identification, it may violate an individuals privacy. This is because others may obtain personal information by reading the tags. In addition, user tracking can be a problem, because the fixed ID values of tags can be traced in network segments. Although various solutions have previously been proposed to resolve this RFID privacy problem, most of these solutions involve numerous calculations within the tags. Therefore, these techniques can only be applied to expensive active tags with high-capacity embedded processors. In addition, it is not practical to apply these techniques to a mobile RFID system based on passive tags attached to devices because of the increase in the price and volume of the tags themselves. In this paper, we propose an efficient protocol that can be used to implement authentication functions in order to transfer the high-performance calculation functions to mobile devices, thus storing only the resulting values on the tags. This study mainly focuses on improving the limitations of the existing RFID authentication functions, which usually assume active tags. It shows that the same security level and performance can be obtained through passive tags. The proposed protocol meets various security requirements such as tag protection, location- and traffic-tracking prevention; the proposed protocol also meets other requirements such as lightweightness and the desired level of performance.

Keywords: Passive tag, mobile RFID, authentication protocol, privacy

1. Introduction

A mobile RFID system allows a user to directly obtain information from RFID tags using mobile devices, as opposed to a fixed RFID system. A mobile RFID system functions to provide information in a convenient manner. However, if a mobile device can function as an RFID reader and thus enable anyone to read an RFID tag, the information contained on the tagging product can be used to identify the owner and his/her characteristics. Thus, this may cause an invasion of privacy [1]. Therefore, when a customer purchases a tagged product, it is necessary to provide privacy protection, and thus, security functions to protect their information. Although various solutions [2–6] have previously been proposed to resolve the RFID privacy problem, most of these techniques require that numerous calculations be performed within the tags. Therefore, these techniques can only be used with expensive tags, which have embedded special-purpose processors. In addition, it is impractical to apply these techniques to a mobile RFID system based on passive tags attached to devices because of the increase in the price and volume of the tags themselves. Therefore, we propose an efficient authentication protocol that can be used to implement authentication functions using only passive tags by transferring computation-intensive functions such as hash functions and random number generation to mobile devices, storing only the results of these functions on tags. This paper is organized as follows. Section 2 explains the problem statement. Section 3 contains a description of the proposed authentication protocol. Section 4 presents the analyses of the proposed protocol in terms of security and performance. Section 5 describes the implementation work. Finally, section 6 concludes the paper.

* Corresponding author: e-mail: jhyi@ssu.ac.kr

2. Problem Statement

A mobile RFID system consists of one or more tags, an RFID reader embedded in a mobile device such as a smartphone, and a database, as shown in Figure 1. Each tag contains an antenna and a chip, which is used to store an ID, or unique identification code. The mobile reader identifies tag data by querying the tag and reading the ID obtained from it, or by retransmitting the ID to the database in the server and reading the corresponding tag information. This mobile reader plays the role of an intermediary. In a mobile RFID system, the reader is mobile, and thus, communicates over wireless networks. Lastly, the database plays the role of storing and providing tag information from and to the RFID reader.

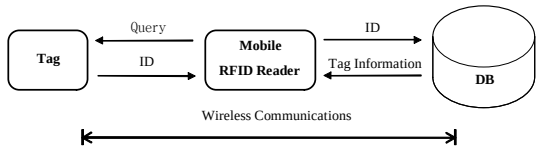

Figure 1 : Mobile RFID system consisting of tags, mobile devices equipped with RFID readers, and database. When the reader sends a query to a tag, it gets the tags ID back from the tag. Then, the actual tag information is retrieved from the database using the tags ID.

The requirements for a mobile RFID system are largely divided into security and performance requirements.

2.1. Security Requirements

Because a mobile RFID system may face the same security problems as a fixed RFID system, it must satisfy the following requirements:

Tag Protection: If the unique tag ID is delivered to any reader that queries a tag without checking whether it is trustworthy, the tagged item reveals information about what the item is and who owns it. Thus, when tags are queried by unauthenticated readers, their unique IDs must not be exposed before the reader is authenticated.

Location Tracking Prevention: The tag ID is given to the RFID reader and retransmitted to the database over wireless or wired networks. Then, the actual information for the item is provided through the database. Thus, although the data transmitted between the tag and reader does not provide meaningful information, if the tag ID is kept unchanged, an attacker over a network can identify the present location of the items owner. This is because the RFID reader is contained on a mobile device such as a smartphone in a mobile RFID system. To prevent this, the tag ID must be renewed with every query.

Traffic Analysis Prevention: When authenticating the RFID reader in the protocol, the authentication data are open between the tag and RFID reader and between the RFID reader and database. Even though the data are open to the public, it must be hard for an attacker to figure out the correlation between the tag and owner.

2.2. Performance Requirements

The easiest way to make an RFID system more secure is to implement numerous security functions on the tag. However, this raises the cost of the tag. To be practically useful for an RFID application, it is necessary to consider low-cost passive tags. Thus, the following performance requirement should be considered.

Lightweightness: A passive tag has limited functionality in terms of its computational capability and memory, i.e., it provides only reading, writing, and locking capabilities. However, in a mobile RFID system, it is possible to utilize the computational capability of mobile devices. In other words, it is possible to execute the computation-intensive functions on mobile devices, and thus, overcome the limitation of passive tags.

3. Proposed Authentication Protocol

In this section, we propose a lightweight authentication protocol using hash functions and XOR operations, and then analyze the security and efficiency of the proposed protocol. The notation used in the rest of this paper is shown in Table 1.

Table 1 : Notation used in rest of this paper.

Notation	Description
$\oplus$	bitwise XOR (exclusive-or)
$\|\|$	concatenation
PWD	password for tag status change (lock and unlock)
SK	secret key shared between DB and mobile reader
ID_{reader}	unchangeable identifier of mobile RFID reader
ID_{tag}	changeable identifier of the tag
${R_1}^i$	i-th random value that the mobile reader creates
${R_2}^i$	i-th random value that the database creates
$H_{SK}(x)$	hashed value of x using SK
α^i	i-th random challenge stored in tag and DB

3.1. Initialization Phase

To initialize the protocol, the tag and database store ID_{tag}, α^{i-1}, and PWD. The mobile reader stores its own ID_{reader} and secret key SK. The detailed process is as follows.

Database: It stores the ID_{reader} corresponding to the registered mobile reader and secret key SK. When the ID_{tag} is given from the reader, the database generates the PWD and calculates the value of α^{i-1}, where $\alpha^{i-1} = ID_{tag} \oplus H_{SK}({R_2}^{i-1})$. It then stores the ID_{tag}, PWD, and value of α^{i-1}.

Mobile RFID Reader: When the reader is registered in the database, it stores the SK to be used for computing the hash value of the random value generated by either the reader or database.

Tag: The original tag has only the pure ID_{tag}, but when registered in the database, it is replaced by the value of α^{i-1} that the database generates. This value, α^{i-1}, changes every tag authentication. The PWD is also kept secret for locking and unlocking the tag later.

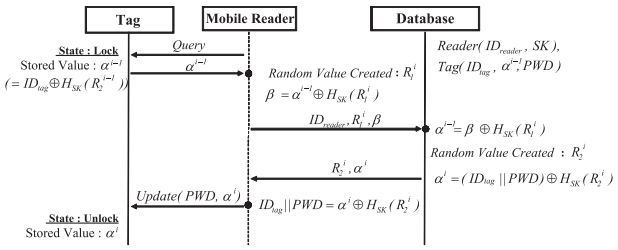

Figure 2 : Proposed authentication protocol can only function using passive tags by running highly-computational functions such as hash functions and random number generators on mobile devices and then storing resulting values on tags.

3.2. Authentication Phase

The process to authenticate the tag information is shown in Figure 2. The detailed authentication procedure is as follows.

Step 1: If the mobile reader sends queries to the tag, the tag transmits the stored value, α^{i-1}.

Step 2: When this value, α^{i-1}, is received, the mobile reader creates random value ${R_1}^i$, takes the hash function on ${R_1}^i$ along with SK, and computes a new value, β, where $\beta = \alpha^{i-1} \oplus H_{SK}({R_1}^i)$. It sends value β and ${R_1}^i$, along with the ID_{reader}, to the database. Then, the database checks whether or not ID_{reader} is registered. Once ID_{reader} is identified as registered, the corresponding SK is retrieved from the database. Next, the database calculates $H_{SK}({R_1}^i)$ using the received ${R_1}^i$ and SK, and then extracts the original α by computing $\alpha^{i-1} = \beta \oplus H_{SK}({R_1}^i)$. Once the value of α^{i-1} is obtained, the original ID_{tag} and PWD, matched with α^{i-1} from the database retrieval, are picked up. After this, the database creates another random value, ${R_2}^i$, calculates the hash value, $H_{SK}({R_2}^i)$, using secret key SK, and then computes value α^i by $\alpha^i = (ID_{tag}||PWD) \oplus H_{SK}({R_2}^i)$ using the retrieved ID_{tag} and PWD. Finally, it replaces α^{i-1} with α^i, and sends back the values of ${R_2}^i$ and α^i to the mobile reader.

Step 3: The mobile reader calculates $H_{SK}({R_2}^i)$ using ${R_2}^i$ and SK and extracts ID_{tag} and PWD by calculating $(ID_{tag}||PWD) = \alpha^i \oplus H_{SK}({R_2}^i)$. Once the reader obtains PWD, it unlocks the tag using PWD, updates the value of α^{i-1} with the new value, α^i, and locks the tag again if necessary. Now, the mobile reader is free to get the actual tag information using ID_{tag}.

4. Security and Performance Analysis

This section deals with the security and performance of the proposed protocol.

4.1. Security Analysis

The proposed protocol meets the security requirements described in Section 2.1, as follows:

Tag Protection: An attacker is not able to modify the information on the passive tag because they do not have PWD. In addition, the only value they can read is α, not the actual ID_{tag}. Thus, even if the attacker reads the value of α, the tag information is protected against such readings because the attacker does not have the authenticated ID_{reader}. In other words, because only authenticated readers know the secret key, SK, only they can derive the actual ID_{tag} at the end of the protocol steps. For more detail, consider the following equation.

$$\beta = \alpha^{i-1} H_{SK}({R_1}^i) \tag{1}$$

The value of α^{i-1} stored in the tag is the result of the previous authentication process. This value is updated at every authentication process. Even if an attacker comes to know β and ${R_1}^i$ from listening to wireless communications, he cannot obtain the actual ID_{tag}. In other words, to get the ID_{tag}, he would need to get α^i and then derive ID_{tag} from the equation $(ID_{tag}||PWD) = \alpha^i \oplus H_{SK}({R_2}^i)$. Thus, it is impossible for an attacker to compute both $H_{SK}({R_1}^i)$ and $H_{SK}({R_2}^i)$ without knowing SK, which is shared only between the authenticated reader and database.

Location Tracking Prevention: Because the data stored in the tag (i.e.,α) changes at every authentication process, no useful information is leaked to an attacker in the segment between the tag and reader. Moreover, when a mobile reader sends β to the database or the database sends α to the reader, both parameters are re-randomized using secret key SK every time. Thus, an unauthenticated reader is not able to trace who has what items without knowing the secret key, even though he can easily obtain ${R_1}^i$ and ${R_2}^i$.

$$\alpha^i = (ID_{tag}||PWD) \oplus H_{SK}({R_2}^i) \tag{2}$$

Traffic Analysis Prevention: Even if an attacker is able to collect all of the data from the communications, he fails to analyze the entire authentication protocol and thus cannot force the unauthenticated reader to be authorized unless he can obtain the secret key shared between the authenticated reader and database. For more details, suppose that the attacker obtains all of the following parameters:

$$ID_{reader}, {R_1}^i, {R_2}^i, \beta, \alpha^i \tag{3}$$

It is impossible to obtain the ID_{tag} and PWD, because the attacker cannot correctly compute $H_{SK}({R_1}^i)$ and $H_{SK}({R_2}^i)$ without having SK. Thus, there is no probability that false

authentication succeeds. In addition, let us assume that the attacker stores $(ID_{reader}, {R_1}^i, \beta)$ that an authenticated reader has already sent and resends them to the database. Then, the database replies to the attacker with α^i and ${R_2}^i$. Even in this case, because the attacker does not know SK, he is not able to compute $H_{SK}({R_2}^i)$ and thus find ID_{tag} and PWD.

Table 2 : Proposed protocol satisfies all security requirements as well as performance requirement for lightweightness.

	Hash Lock	Randomized Hash Lock	MW	Proposed
Tag Protection	O	O	O	O
Location Tracking Prevention	X	O	O	O
Traffic Analysis Prevention	X	X	O	O
Lightweightness	X	X	X	O

Table 2 shows a comparison of the features provided by the existing techniques and the proposed protocol in terms of the security requirements described in Section 2.1. The hash lock protocol [7] satisfies only tag protection, while the randomized hash lock protocol [7] only satisfies tag protection and location tracking prevention. In the MW protocol [8], all of the security requirements are satisfied, except the lightweightness requirement. In contrast, the proposed protocol meets all of the security and performance requirements. An analysis of the lightweightness performance requirement will be separately discussed in the next section. In addition to an analysis of how the proposed scheme meets the security requirements, it is necessary to prove the security of the scheme in terms of playback and collision attacks. First, the proposed scheme employs the random number generator, secret key SK , and random values ${R_1}^i$ and ${R_2}^i$ when generating a hash value. However, even though ${R_1}^i$ and ${R_2}^i$ are known, it is impossible to determine the identical value in a hash function without knowing SK. Thus, the proposed scheme is secure against the playback attack. Second, we need to determine whether a collision attack is possible in the proposed protocol, which uses the hash function. In principle, the security of the hash function relies on collision resistance against a preimage attack [9]. In such an attack, if the attacker attempts to break the hash function, $2^{n/2}$ trials are needed, where n is the bit length of the hash output. For example, when we use the well-known SHA-1 as a hash function, $2^{160}/2 = 2^{80}$ trials are required to find the collision of the preimages because of the birthday paradox [9]. In other words, if we assume that the attacker has sufficient computation power to compute 2^{30} trials per second, the total amount of time needed to succeed in a collision attack is about 2^{50} seconds (i.e., approximately 10,000 years). Thus, the proposed scheme is computationally secure against a collision attack.

4.2. Performance Analysis

This section deals with the efficiency of the proposed protocol in comparison with the performance requirements defined in Section 2.2.

Lightweightness: Because the existing protocols are assumed to use active tags, not passive tags, they are almost not applicable to resource-limited passive tags. In the randomized hash lock protocol [7], it is necessary to implement a hash function and random number generator. In the MW protocol [8], a random number generator and a tree algorithm to search for the key need to be implemented in the tag and reader. None of the previous protocols meet the requirement of lightweightness. As can be seen in Table 3, the proposed protocol requires a large amount of memory space to store ID_{reader}, SK, ID_{tag}, and α^i on the database side, but it can implement the authentication process using only passive tags. As mentioned earlier, the main idea of lightweightness is to perform most of the necessary computations for the tag on the mobile devices.

Table 3 : Performance is compared with previous protocols in terms of memory usage, computational complexity, and communication overhead. Symbols and acronyms are defined as follows: $h-$hash value, $k-$password, $m-$number of mobile readers, $n-$number of tags, $r-$random number, $s-$secret key, $t-ID_{tag}$, and uID_{reader}.

		Randomized Hash Lock	MW	Proposed
Memory Usage (bit)	Tag	$log(kh)$	$log(kh)$	$log(kh)$
	Reader	$log(tkh^2r)$	$log(tkh^2r^2s)$	$log(tkh^2rs)$
	DB	$nlog(tk)$	$nlog(tkhs)$	$mlog(us)+nlog(tkh)$
Computation Complexity	Reader	$(n+1)h+r$	$3h+r$	$2h+r$
	DB	-	$(n+2)h+r$	$2h+r$
Communication Overhead (bit)	Tag-Reader	$log(kh^2)$	$log(kh^2)$	$log(kh^2)$
	Reader-DB	$(n+1)log(t)+log(rk)$	$log(h^3r^2)$	$log(uh^2r^2)$

In addition to lightweightness, the proposed protocol has many advantages compared to the previous protocols that are applicable for mobile RFID applications (i.e., randomized hash lock and MW), in terms of computational complexity, memory usage, and communication overhead.

First, the memory usage is measured in terms of the tags, mobile readers, and database. On the tag side, the existing protocols and the proposed one all require the same amount of memory space, $log(kh)$, where k denotes the password and h is the hash value. On the mobile reader side, the proposed protocol requires $log(tkh^2rs)$ bits, where t indicates the number of tags, r is a random number, and s is the secret key. This amount is more than the randomized hash lock but less than the MW protocol. On the database side, the randomized hash lock protocol requires $nlog(t)+nlog(k)=nlog(tk)$ bits and the MW protocol requires $nlog(t)+nlog(k)+nlog(h)+nlog(s)=nlog(tkhs)$ bits. The proposed protocol needs to store n instances of ID_{tag}, k in-

stances of the password for the state change of the tag, the hash value h, and m mobile reader identifiers. Thus, it requires $nlog(t)+nlog(k)+nlog(h)=nlog(tkh)$ bits and $mlog(u)+mlog(s)=mlog(us)$ bits, where u denotes ID_{reader}. Compared to the previous protocols, the proposed one requires more memory space, but the memory requirement of the database is not a major concern in an RFID system.

The computational complexity can be measured at the reader and database. In the case of the randomized hash lock protocol, no computation is required in the database, but the computation in the mobile reader reaches $(n+1)h+r$, which means that $(n+1)$ hash computations and one random number generation are necessary. The MW protocol requires $3h+r$ computations in the mobile reader and $(n+1)h+r$ in the database. The proposed protocol constantly requires $2h+r$ computations in both the mobile reader and database. This is a significant performance gain because the proposed scheme is independent of the number of tags. It is noticed that the overall processing time for authentication depends on the computation cost and the number of tags.

The communication overhead cost can be analyzed for two communication segments: the tag-reader and reader-database sections. The communication overhead requirement between the tag and reader is the same in all three protocols. In the case of the reader-database section, the randomized hash lock protocol requires $(n+1)log(t)+log(rk)$ bits, which represent much more communication overhead than the other protocols. The MW protocol indicates that $log(h^3r^2)$ bits of communication overhead are used between the reader and database. The proposed protocol shows $log(uh^2r^2)$ bits, where u denotes the identifier of the mobile reader. Thus, the proposed protocol generates lower data traffic than the others.

5. Implementation

We performed a feasibility test of the proposed protocol using real world testing resources. The test was carried out in the following environment (see Figure 3).

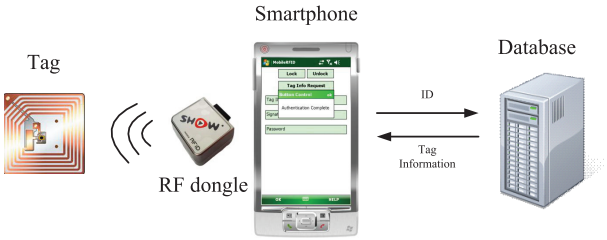

Figure 3 : Experimental testbed is set up as follows: one desktop computer is configured as a database server. Nesslab's 900 MHz -UHF RF reader dongle as a RFID reader and Power IDs EPC Gen-2 tags are used.

One desktop computer was used as a database server. As a RFID reader, we used Nesslab's 900 MHz UHF RF reader dongle [10]. This dongle was attached to a Samsung SCH-490 smartphone, which has a Marvall Monahans PXA 312 806 MHz processor and 160 MB of memory. For the passive tag, we used Power G2, which is compatible with the EPC Gen-2 standard [11]. This passive tag has 96 bytes of EPC space and 720 bytes of user memory space. We also used the OpenSSL crypto library [12] to implement the proposed protocol, and all of the testing code was written in C for Windows Mobile 6.1.4.

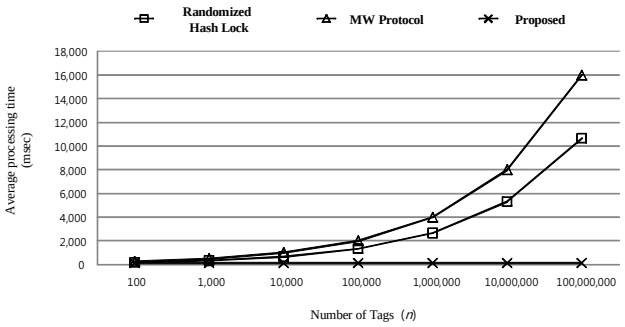

Figure 4 : Average processing time for authentication is shown. The performance of the randomized hash lock protocol and MW protocol are proportional to the number of tags, whereas the proposed protocol shows a constant processing time.

Figure 4 shows the overall authentication processing time. As discussed in Section 4.2, the performances of the randomized hash lock protocol and MW protocol are proportional to the number of tags, whereas the proposed protocol shows a constant processing time. This large difference can be attributed to the fact that both the randomized hash protocol and MW protocol are required to search the entire list to find the correct in the database, whereas the proposed protocol does not require such a function.

6. Conclusion

A mobile RFID system with free reading functions may cause the problem of personal privacy violation. This is because others may read the information on the tags. Thus, we proposed a mobile RFID authentication technique that is applicable to low-cost passive tags by making the best use of the mobile devices computational power. The main idea was to execute highly computational functions on mobile devices and store only the processing results on the tags. We showed that the proposed protocol outperformed the previous protocols in terms of tag protection, location tracking prevention, traffic analysis prevention, and lightweightness. In addition, we implemented the proposed protocol using the cryptographic library, and tested its performance on real smartphones and RFID readers.

Acknowledgement

This work was supported by a grant from the KEIT funded by the Ministry of Knowledge Economy(10035219).

References

[1] S. Garfinkel and A. Juels and R. Pappu, RFID Privacy: An Overview of Problems and Proposed Solutions, IEEE Security and Privacy **3**, 1540–7993 (2005).
[2] B. Song and C. J Mitchell, RFID Authentication Protocol for Low-cost Tags, Proceedings of the first ACM conference on Wireless network security, 140-147 (2008).
[3] , D. Henrici and P. Muller, Providing Security and Privacy in RFID Systems Using Triggered Hash Chains, IEEE International Conference on Pervasive Computing and Communications, 50-59 (2008).
[4] , A. Juels, Minimalist Cryptography for Low-cost RFID Tags, International Conference on Security in Communication Networks, 149-164 (2004).
[5] , A. Juels and R. Pappu, Squealing Euros: Privacy Protection in RFID-Enabled Banknotes, Financial Cryptography, 103-121 (2003).
[6] , A. Juels and R. Rivest and M. Szydlo, The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy, ACM Conference on Computer and Communications Security, 103-111 (2003).
[7] , D. Molnar and D. Wagner, Privacy and Security in Library RFID: Issues, Practices, and Architectures, ACM Conference on Computer and Communications Security, 210-219 (2004).
[8] S. Weis and S. Sarma and R. Rivest and D. Engels, Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems, Security in Pervasive Computing **2802**, 201-212 (2004).
[9] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, (2001).
[10] UHF RFID Reader Dongle, http://www.nesslab.com/rfid_04_22.php.
[11] EPCglobal, Class 1 Generation 2 UHF Air Interface Protocol Standard Version 1.0.9: Gen-2, http://www.epcglobalus.org.
[12] OpenSSL Project, http://www.openssl.org.

Hyeong-Chan Lee received his B.S. and M.S. degrees in Computer Science from Soongsil University, Korea, in 2010 and 2012, respectively. His research interests include mobile application security and mobile platform security.

Tae Yang Eom received the M.S. degree in Computer Science from Soongsil University, Korea. He currently works for HandiSoft, Korea. His research interests are in the areas of mobile network and RFID security.

Jeong Hyun Yi is an Assistant Professor in the School of Computer Science and Engineering at Soongsil University, Seoul, Korea. He received the B.S. and M.S. degrees in computer science from Soongsil University, Seoul, Korea, in 1993 and 1995, respectively, and the Ph.D. degree in information and computer science from the University of California, Irvine, in 2005. He was a Principal Researcher at Samsung Advanced Institute of Technology, Korea, from 2005 to 2008, and a member of research staff at Electronics and Telecommunications Research Institute (ETRI), Korea, from 1995 to 2001. Between 2000 and 2001, he was a guest researcher at National Institute of Standards and Technology (NIST), Maryland, U.S. His research interests include mobile security and privacy, network security, cloud computing security, and applied cryptography. Some of his notable research contributions include Certificate Management Protocol (CMP) for Korean PKI Standards and integration of Korea PKI and U.S. Federal PKI.