An Advanced Mutual-Authentication Algorithm Using AES for RFID Systems

Batbold Toiruul,[†] and KyungOh Lee[††],

Computer Science Department, Sunmoon University, #100 Tangjeong-myun ChungNam, Asansi 336-708, Korea

Abstract

The biggest challenge for current RFID technology is to provide the necessary benefits while avoiding any threats to the privacy of its users. Although many solutions to this problem have been proposed, almost as soon as they have been introduced, methods have been found to circumvent system security and make the user vulnerable. We are proposing an advanced mutual-authentication protocol between a tag and the back-end database server for a RFID system to ensure system security integrity. The three main areas of security violations in RFID systems are forgery of the tags, unwanted tracking of the tags, and unauthorized access to a tag's memory. Our proposed system protects against these three areas of security violations. Our protocol provides reader authentication to a tag, exhibits forgery resistance against a simple copy, and prevents the counterfeiting of RFID tags. Our advanced mutual-authentication protocol uses an AES algorithm as its cryptograph primitive. Since our AES algorithm has a relatively low cost, is fast, and only requires simple hardware, our proposed approach is feasible for use in RFID systems. In addition, the relatively low computational cost of our proposed algorithm compared to those currently used to implement similar levels of system security makes our proposed system especially suitable for RFID systems that have a large number of tags.

Key Words

RFID, AES, cryptograph.

Introduction

Radio-frequency identification (RFID) is an emerging technology. It is the next generation of an optical barcode with several major advantages over an optical barcode since a line-of-sight between the reader and the barcode is not needed, and several tags can be read simultaneously. RFID technology is rapidly finding more diversified applications in today's marketplace. For example, RFID technology is now being used for automatic tariff payment in public transport, animal identification and tracking, automated manufacturing, and logistical control for automatic object identification since every object can be identified by a unique identification tag number. A RFID system consists of three parts: the radio-frequency (RF) tags, the RF readers, and the back-end database server. The back-end server associates records with the tag data collected by the readers. Tags are typically composed of a microchip for storage and performing logical operations and a coupling element such as an antenna coil for wireless communications. Memory chips on the tags can be read-only, write-once/read-many, or fully writable. Each memory chip holds a unique ID and other pertinent information transmitted to the tag reader using a RF. The tag readers interrogate the tags using a RF antenna and interact with the back-end database for more functionality.

However, RFID tags may pose a considerable security and privacy risk to organizations and individuals using them. Since a typical tag answers its ID to any reader and the replied ID is always the same, an attacker can easily copy the system by reading out the data of a tag and duplicating it to bogus tags. Unprotected tags may have vulnerabilities to eavesdropping, location privacy, spoofing, or denial of service (DoS). Unauthorized readers may compromise privacy by accessing tags without adequate access control. Even when the content of the tags is protected, individuals may be tracked through predictable tag responses. Even though many cryptographic primitives can be used to remove these vulnerabilities, they cannot be applied to a RFID system due to the prohibitive cost of including protection for each and every RFID tag. The RFID tag is the most costly item in a RFID system as such systems inherently use at least a minimum of several tags. Economic constraints usually dictate that the tags cost as little as possible and that as few as possible are used. Power consumption, processing time, storage, and gate count are all severely limited. For example, a practical tag costing in the order of $0.05 US may be limited to having only hundreds of bits of storage and roughly 500–5,000 gates in order to meet cost restraints.

In this paper, we propose a new mutual authentication protocol that uses AES (Advanced Encryption Standard) for the security of a RFID system. In November 2001, NIST announced that the AES algorithm, based on the Rijndael algorithm, was the new encryption standard [11], [12]. We chose the AES as our cryptographic primitive because it is standardized and considered to be secure. The AES algorithm consists of one s-box, two other kinds of transformations, and a key schedule. It supports key lengths of 128, 192, and 256 bits, and many hardware implementations of the AES algorithm exist. However, several papers have presented a low-power implementation of the AES suitable for use in RFID tags in terms of power

consumption and die size [10], [13], [14].

Our proposed mutual-authentication protocol can be used to solve the inherent security problems of RFID systems. Our protocol allows high-value goods to be protected against adversarial attackers. Also, our protocols can easily meet current data rate restrictions and are compliant with existing standards as well as requirements concerning chip area and power consumption. With mutual authentication we can provide a proof for each entity of a RFID system based on an AES encryption. Therefore, our proposed protocol is sufficiently robust to withstand active attacks such as the man-in-the-middle attack, the replay attack, the eavesdropping attack, and the unwanted tracking of customers.

2. Related Works

RFID security and privacy issues have been an active and continuing area of research. We describe some of the related studies below.

Hash lock scheme, developed by MIT [6]. In this scheme, each tag verifies the reader as follows. The reader has a key (k) for each tag, and each tag holds the result metaID, where metaID = hash (k) of a hash function. A tag receives a request for ID access and sends a metaID in response. The reader sends a key that is related to the metaID received from the tag. The tag then calculates the hash function from the received key and checks whether the result of the hash function corresponds to the metaID held in the tag. Only if both data sets agree does the tag send its own ID to the reader. However, in this scheme, the adversary can track the tag via the metaID. Furthermore, both the random key and the tag ID is subject to eavesdropping by an attacker.

Randomized hash lock scheme, developed by MIT [6]. This is an extension of the hash lock scheme, and requires the tag to have a hash function and a pseudorandom generator. Each tag calculates the hash function based on the input from a pseudorandom generated r and id, i.e., c = hash (id, r). The tag then sends c and r to the reader. The reader sends the data to the back-end database. The back-end database calculates the hash function using the input as the received r and id for each ID stored in the back-end database. The back-end database then identifies the id that is related to the received c and sends the id to the reader. The tag output changes with each access, so this scheme deters tracking. However, the attacker can impersonate the tag to a legitimate reader. Also the attacker can know the r and ID_k because eavesdropping is possible.

A RFID security approach for a supply chain system, developed by the IBM China Research Lab [2]. This approach requires read-access control. When a tag receives an inquiry from a reader, the tag will first create a random number k, which it then transmits. After the random number k is received by the reader, the reader sends k back to the backend database. The backend database hashes (ReaderID || k) and sends out the hash value to the reader. The reader then sends it to the tag. In the meantime, the tag also hashes (ReaderID || k). Then the tag compares the hash value calculated by the tag to that by the reader. If they are equal, the reader passes the authentication and the tag can then provide tag ID-related information. However, in this approach, an attacker can eavesdrop on the Reader ID as no security is required for the tag to get the reader ID. Therefore, an attacker can impersonate a reader to a tag.

Cryptographic approach to "privacy-friendly" tags, developed by the NTT lab [8]. The basic idea of Ohkubo et al. is to modify the identifier of the tag each time it is queried by a reader so that the identifiers can be recognized only by authorized parties. The tag refreshes its identifier autonomously using two hash functions, G and H, as described below. Readers are (untrusted) devices that do not have cryptographic functionalities but a hash function can be embedded into the tags. Soon, this may well be a realistic assumption. Ohkubo et al.'s scheme has a complexity of mn hash computations in a closed environment (2 hash operations are carried out mn/2 times), and of 2 mn in an open environment since the database computes all of the hash chains when trying to identify a foreign tag. Thus, when the number of tags, n, or the number of read operations, m, is large, the complexity becomes unmanageable so this scheme is not scalable.

Strong authentication for RFID systems using the AES algorithm, developed by project ART [1]. The main theme of this paper is the assumption that an AES is feasible for current RFID technology without major additional costs. The ART project team selected AES as a cryptographic primitive for symmetric authentication. They analyzed several architectural possibilities for implementing AES-128 encryption functionality. The implementation of the data path of an AES-128 encryption design has a current consumption of 8.15 μA on a 0.35-μm CMOS process. It operates at a frequency of 100 kHz and needs 1,016 clock cycles for encrypting a 128-bit data block. The required hardware complexity is estimated to be 3,595 gate equivalents (GEs).

This report uses unilateral authentication, which works as follows. There are two partners, *A* and *B*. Both possess the same private key, *K*. *B* sends a random number, ***r,*** to *A*. *A* encrypts the random number $E_k(r)$ with the shared key *K* and sends it back to *B*. *B* proofs the result and can verify the identity (in other words, the possession of *K*) of *A*.

In this case, the man-in-the-middle-attack is possible. The attacker sends a random number to a tag. Then the tag replies with the encrypted value of *r* to the attacker. Therefore, it is possible for the attacker to obtain the shared *k* value from many combinations of *r* and $E_k(r)$. Then the attacker can impersonate a legitimate reader to the tag or a legitimate tag to the reader. Therefore, we need mutual

authentication.

3. Our Proposed Approach to RFID security

3.1 Notations

We use the notations summarized in Table 1 to describe our protocol throughout the remainder of this paper.

Table 1: Notations

T	RF tag, or transponder
R	RF tag reader, or transceiver
B	Back-end server, which has a database
k_1 , k_2	Random secret keys, shared between T and B
K	Cryptographic key, shared between T and B
ID_k	Unique identification number of T, shared between T and B
$E_k(k_1 \oplus k_2)$	AES cipher text, using k_1 , k_2 , and k
$E_k(k_1 \oplus k_2 \oplus ID_k)$	AES cipher text, using k_1 , k_2 , k, and ID_k
$E_k(k_1, k_2)$	The notated $E_k(k_1 \oplus k_2)$
$E_k(ID)$	The notated $E_k(k_1 \oplus k_2 \oplus ID_k)$

3.2 Assumptions and attacking model

In our protocol, we assume that *T* has AES encryption cryptographic hardware. In [16], since an AES encryption and decryption unit with a block size of 128 bits can be implemented with only about 3.4 K-gates, our protocol only requires a small gate size. Also, we assume that *T* only has its authentication-related information, ID_k, Also, *T* has a memory for keeping values of ID_k, k_1, and k_2 to process mutual authentication. We assumed that the communication channel between *R* and *B* was secure.

To solve the security risks and privacy issues, the following attacking model must be prevented [3]–[6]. However, in our protocol, we have not considered a physical attack such as removing a RFID tag physically from a product because it is hard to carry out in public view or on a wide scale without detection. We consider the following attacks.

Man-in-the-middle attack: The attacker can impersonate a legitimate reader and get the information from *T,* so he/she can then impersonate a legitimate *T* responding to *R*. Thus, a legitimate *R* can easily be fooled into authenticating an attacker before the next session.

Replay attack: The attacker can eavesdrop on the response message from *T,* and retransmit the message to the legitimate *R*.

Forgery of tags*:* A simple copy of *T's* information can be obtained through eavesdropping by an attacker.

Unwanted tracking of customers: It is possible to track people's movements, social interactions, and financial transactions by correlating data from multiple tag reader locations.

3.3 Security requirement

To protect user privacy, we consider the following requirement from a cryptographic point of view [7], [8].

Data confidentiality: T's private information must be kept secure to guarantee user privacy, and T's information must be meaningless to any unauthorized users even though it can be easily obtained through eavesdropping by an attacker.

Tag anonymity: Although T's data are encrypted, T's unique identification information can be exposed since the encrypted data are constant. An attacker can identify each T by using its permanent encrypted data. Therefore, it is important to make the information on T anonymous.

Data Integrity: If the memory of T is rewritable, forgery and data modification will occur. Thus, the linkage between the authentication information and T itself must be given in order to prevent a simple copy of T. However, data loss will result from a DoS attack, power interruption, message hijacking, etc. Thus, authentication information between T and B must be delivered without any failure, and data recovery must be provided.

In addition, we had to consider and evaluate the following security feature in the design of our RFID authentication protocol.

Mutual authentication and reader authentication: In addition to access control, the mutual authentication between T and B must be provided as a measure of trust. By authenticating mutually, the replay attack and the man-in-the-middle attack to both T and B is prevented.

3.4 Protocol design

Our overall protocol is shown in Fig. 1. The detailed procedures for each step are described in the following:

3.4.1 Initial setup

Each T is given two fresh random secrets, k1 and k2, and a unique identification, IDk. The database (D) of B also stores them as the shared secret. In addition, D manages a record pair for each tag consisting of (IDk, TagID). T has an AES-128 encryption circuit. If a reader requires a tag's ID, the tag must first authenticate the reader. After authentication, the reader can obtain the tag ID by the tag's response and reference to the database. In addition, both T and B have a cipher key, k, that is a 128-bit key.

3.4.2 Detailed description

In the following, we describe our proposed protocol

according to the sequence of message exchange. Also, we discuss the security goals that are achieved during the execution of each protocol message.

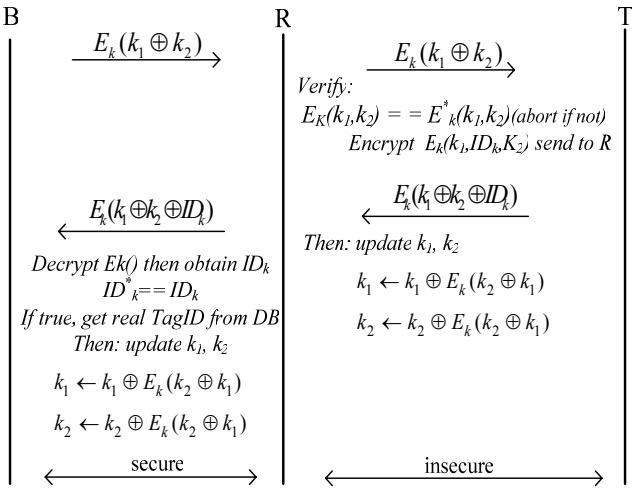

Fig. 1. The Proposed Mutual-Authentication Protocol

Step 1 (Challenging): In this step, reader *R* usually applies a collision protocol such as secure binary tree walking [4], an interleaved protocol [3], or the standard protocol of ISO 18000-3 MODE [7] to singularize *T* out of many. The Reader, *R*, receives $E_k(k_1,k_2)$ from the back-end server, *B*. Then *R* sends $E_k(k_1,k_2)$ to the queried *T*. The cipher key *k* and random numbers k_1 and k_2 are shared by *B* and *T*. Therefore, $E_k(k_1,k_2)$ is used to authenticate the validity of *R*.

Step 2 (Authentication of R): When queried, *T* generates $E^*_k(k_1,k_2)$ and verifies whether the received $E_k(k_1,k_2)$ is valid by comparing $E_k(k_1,k_2)$ with $E^*_k(k_1,k_2)$. If $E_k(k_1,k_2)==E^*_k(k_1,k_2)$, *T* authenticates *R*. Then *T* generates $E_k(k_1 \oplus k_2 \oplus ID_k)$, designated as $E_k(ID)$, which is the encryption of the AES-128 cryptographic algorithm. *T* uses this as the identification information and sends it to *R*.

Otherwise, *R* is not authenticated and *T* will keep silent. Therefore, being tracked by an attacker is not possible when no authorized readers are nearby. Cipher key *k* and random numbers k_1 and k_2 are shared only between *T* and *R*. Therefore, *T* can detect an illegal *R* and discard the message. Consequently, the man-in-the-middle attack by an illegitimate *R* and a passive eavesdropper can be prevented.

If *T* has successfully authenticated *R*, *T* updates the shared secrets keys, k_2 and k_1 by exclusive-ors with $E_k(k_1 \oplus k_2)$.

Step 3 (Authentication of T): *R* simply forwards $E_k(ID)$ to *B*. Within this step, *B* authenticates *T* with $E_k(ID)$. At first, *B* decrypts $E_k(ID)$ using cipher key *k* and random numbers k_1 and k_2 and obtains ID_k. Then *B* verifies whether ID_k is valid by comparing the obtained ID_k with ID^*_k. Random secrets, k_1 and k_2, and the cipher key, *k*, are shared only between *B* and *T*. Therefore, *B* can detect an illegal *T* and discards the message. Therefore, the man-in-the-middle attack by an illegitimate *T* and a passive eavesdropper can be prevented. If *T* is authenticated, *B* retrieves the records corresponding to ID_k and gets the real *TagID*.

Even if $E_K(ID)$ is discovered through eavesdropping, the eavesdropper cannot know the ID_k value, since he/she does not know k_1, k_2, and the cipher key *k*. Since *B* initially stores the unique identification, ID_k, *B* can evaluate the linkage between $E_k(ID)$ and *T* itself in order to prevent forgery. Forgery can be detected and prevented by *B* at this time.

At the same time, *B* can detect and prevent the man-in-the-middle attack since ID_k is used as the factor of the man-in-the-middle attack detection. Similarly, the replay attack can also be detected and prevented simultaneously.

If *B* successfully finishes the authentication process, *B* generates $E_k(k_1 \oplus k_2)$ with its shared random secrets, k_1 and k_2. The database of *B* updates the shared secrets keys, k_1 and k_2, by exclusive-ors with $E_k(k_1 \oplus k_2)$. Then, mutual authentication has finally succeeded.

4. Analysis

4.1 Security analysis

We have evaluated our protocol from a security requirement standpoint. Our protocol guarantees a secure mutual authentication only with AES-128 encryption messages, $E_k(k_1 \oplus k_2)$, $E_k(k_1 \oplus k_2 \oplus ID_k)$, and ID_k, *T* does not store user privacy information. Thus, data confidentiality of tag owners is guaranteed and the user privacy on data is strongly protected. In every session, we use a fresh random nonce as the keys between entities. These keys are randomized and anonymous since they are updated for every read attempt. Thus, tag anonymity is guaranteed and the location privacy of a tag owner is also not compromised. Based on mutual authentication, our protocol guarantees the data integrity between *T* and *B*. The forgery-resistance feature was realized by exclusive-oring the unique authentication number, ID_k, of *T* with the authentication information. ID_k is originally stored during the initial step. Whenever *T* generates $E_k(ID)$, it refers to ID_k, so the linkage between ID_k and *T* itself can be determined. *B* keeps each tag's ID_k initially and authenticates the ownership of the authentication information for *T*. Table 2 shows the comparison of the security requirements and the possible attacks.

The man-in-the-middle attack. Through the

authentication steps 1 and 2, R sends $E_k(k_1 \oplus k_2)$ to T and T sends $E_k(k_1 \oplus k_2 \oplus ID_k)$ to B for preventing the man-in-the-middle attack. B can verify ID_k with the decryption of the AES-128 cryptographic value of $E_k(ID)$ transmitted from T. The key freshness is also guaranteed for each session. The replay attack for T and B is detected and prohibited in step 3 for B and in step 2 for T.

Table 2: Comparison of the secure requirements

Protocol	HLS [6]	RHLS [6]	Ref. [2]	Ref. [8]	Ref. [1]	Our scheme
User data confidentially	x	Δ	Δ	Δ	Δ	O
Tag anonymity	x	Δ	Δ	Δ	Δ	O
Mutual authentication	Δ	Δ	Δ	Δ	Δ	O
Reader authentication	x	x	x	x	x	O
Man-in-the-middle attack prevention	Δ	Δ	x	Δ	x	O
Replay attack prevention	Δ	Δ	x	Δ	Δ	O
Forgery Resistance	x	x	Δ	Δ	Δ	O
Tracking	x	x	Δ	x	x	O

Notation: x – not satisfied; O – satisfied; Δ - partially satisfied

Invulnerable to eavesdropping. In the process of authentication, even when an attacker eavesdrops on the output of tag, $E_k(k_1,k_2)$, it can not pretend to be an authorized reader in the next authentication session since the random secrets, k_1 and k_2, are changed in every session. Also, the required $E_k(k_1,k_2)$ value is an AES algorithm cipher, and random secrets, k_1 and k_2, and cipher key k are shared only between T and B. Since an AES-128 encryption is extremely difficult to inverse, the tag ID_k and random secrets, k_1 and k_2, are protected even if the output is captured by an attacker. Therefore, it is invulnerable to eavesdropping. In one word, our proposed approach is secure when any communication between readers and tags are subjected to eavesdropping.

Prevent being tracked by adversary. Tags keep silent to attackers. They only respond to authenticated readers. Furthermore, as explained above, it is impossible for attackers to pretend to be an authenticated reader. Since no tag output occurs, attackers are unable to track customers by the tag value that existed as they checked out. The privacy of location and the secrecy of what objects that the customers are carrying is protected.

4.2 Performance analysis

We analyzed the performance of our proposed scheme with respect to computation and its anticollision mechanism.

Low computation load. When identifying a tag from N known tags, a reader performs only two AES operations, while for other approaches of randomized access control, at least N hash operations and N searches [2] are required. In addition, the AES tag's hardware has a relatively low cost and fast computation time [10].

Since the computation load remains low even with an increasing number of tags, our proposed approach is suitable for protecting RFID systems with a large number of tags. This feature is very important for a supply chain. Each part along a supply chain deploys numerous tags. In warehouses or retail stores, thousands of products need to be tagged to accelerate the supply chain process. Therefore, a secure RFID scheme that is suitable for a large number of tags is a definite prerequisite for the implementation of a RFID supply chain system.

Anticollision mechanism. The most important command is the anticollision sequence, which is a command every tag must implement. Therefore, a reader sends an initial inventory command. All tags in the environment make a response that is the tag's unique ID. If only one tag answers the request, the ID can be retrieved by the reader and all subsequent commands can be addressed using the ID that addresses one single tag. If two or more tags answer a request, a collision occurs. This can be detected at the reader. The reader then uses a modified inventory request in which it adds a part of the tag's ID to the request. Only tags that have this part of the ID are allowed to answer. Once the ID of one tag is identified, the reader sends a "stay quiet" command to the tag with the identified ID. This method is used as long as no more collisions occur and all tags within the environment are identified. In our proposed approach, we have suggested two anticollision mechanisms, namely the interleaved protocol [3] and the binary-three algorithm [4].

5. RFID Tag Architecture

The RFID tag consists of the analog front-end; the controller for implementing software requirements such as data coding, implementation of the protocol commands, anticollision mechanisms, and error detection; the EEPROM that stores k_1, k_2, ID_k, and k; the key for cryptographic algorithms; and the AES hardware module.

We selected AES as the cryptographic primitive for our proposed approach. One important criterion for selecting the AES algorithm was its structure allowing efficient implementation in hardware. In addition, several previous implementations of the AES have proven it to be low-cost and relatively fast. The tag cost can be around \$0.05 US and the die size is less than 0.25 mm^2. Power consumption is about 10 μA [3], [10], [13].

Most hardware implementations of the AES algorithm have focused on realizing a high data throughput. Recently,

however, some attention has been given to hardware implementations that were designed with hardware efficiency in mind. Hardware efficiency can be increased by lower die sizes and reduced power consumption. Some recent papers have been published that focus on this issue [10], [13]–[15].

Mangard et al. [14] presented a highly regular approach. It is comparable to RFID requirements but requires a chip area of 8,500 gate equivalents while having a higher data throughput of 70 Mbps. The AES hardware of Satoh et al. [13] is a 32-bit architecture and has a hardware complexity of 5,400 gates and reaches a throughout of 311 Mbps.

Feldhofer et al. [10] presented a silicon implementation of the AES optimized for low die size that offers excellent power consumption characteristics. The AES core of the manufactured chip has an area of 0.25 mm^2 on a 0.35 mm CMOS technology, which is comparable in size to a grain of sand. In terms of circuit complexity, the size equals 3,400 gate equivalents, and the average power consumption can be lowered to <5 mW when operated at 100 kHz and 1.5 V. Feldhofer et al. [10] implemented the AES algorithm as an 8-bit architecture.

Our protocol only uses the encryption circuit of AES. Therefore, our protocol hardware requires less chip area and power consumption than previous implementations. It also has several advantages as follows.

- It is an 8-bit implementation of the AES architecture [1], [10], [17].
- We need only the encryption circuit of the MixColumns [10], [13], [15].
- The Rcon function is a constant value. It is implemented as two different constant values in the encryption and the decryption processes. Only circuitry to implement a constant value is required for the encryption process [13].
- By using RAM as detailed in [10], we do not need a ShiftRows transformation. The ShiftRows transformation can be implemented by an appropriate addressing of the RAM or we can use an 8-bit register as the ShiftRows [13].

6. ConclusionS

This paper proposes an advanced mutual-authentication protocol for security and privacy protection in RFID systems using an AES algorithm as a cryptographic primitive. This protocol protects high-valued goods against attackers. With mutual authentication, we can provide a proof for each entity of a RFID system, and since this proof is based on an AES encryption, our proposed protocol is sufficiently robust to withstand active attacks such as the man-in-the-middle attack, the replay attack, the eavesdropping attack, and the unwanted tracking of customers. Also, our protocols can easily meet current data rate restrictions and are compliant with existing standards as well as requirements concerning chip area and power consumption. In addition to cipher k, our proposed protocol uses k_1 and k_2 for security. These secret random numbers, k_1 and k_2, are changed in every session, so the attacker can not obtain important data from a tag even if the tag's outputs have been eavesdropped.

All authentication messages are randomized. In addition, each tag has its own unique identification data, so user data privacy and location privacy are guaranteed.

AcknowledgmentS

This research was supported by the Ministry of Information and Communication (MIC), Korea, under the Information Technology Research Center (ITRC) support program supervised by the Institute of Information Technology Assessment (IITA-2005-C1090-0502-0031).

REFERENCE:

[1] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. "Strong authentication for RFID systems using the AES algorithm." In *Conference of Cryptographic Hardware and Embedded Systems*, 2004. Proceedings, pp. 357–370. Springer 2004.

[2] X. Gao, Z. Xiang, H. Wang, J. Shen, J. Huang, and S. Song. "An approach to security and privacy of RFID system for supply chain," *CEC-East*, pp. 164–168, IEEE International Conference on E-Commerce Technology for Dynamic E-Business (CEC-East'04), 2004.

[3] M. Aigner and M. Feldhofer. "Secure symmetric authentication for RFID tags." *Telecommunication and Mobile Computing*, March 2005.

[4] R. Juels, L. Rivest, and M. Szydlo. "The blocker tag: selective blocking of RFID tags for consumer privacy." In V. Atluri, editor, *8th ACM Conference on Computer and Communications Security*, pp. 103–111. ACM Press, 2003.

[5] D. Henrici and P. Muller, "Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers." *PerSec'04 at IEEE PerCom*, pp. 149–153, March 2004.

[6] S.A. Weis, S.E. Sarma, R.L. Rivest, and D.W. Engels, "Security and privacy aspects of low-cost radio frequency identification systems." *First International Conference on Security in Pervasive Computing*, 2003.

[7] S. Weis, "Security and privacy in radio-frequency identification devices." *Master's thesis*, MIT, 2003.

[8] M. Ohkubo, K. Suzuki, and S. Kinoshita, "Cryptographic approach to "privacy-friendly" tags." In *RFID Privacy Workshop*, MIT, USA, 2003.

[9] ISO/IEC JTC 1/SC 31/WG 4, "Information technology AIDC techniques—RFID for item management air interface, part 3: parameters for air interface communications at 13.56 MHz." *Version N681R*, April 2004.

[10] M. Feldhofer, J. Wolkerstorfer, and V. Rijmen, "AES implementation on a grain of sand." *IEE Proceedings Information Security,* October 2005, Vol. 152, Issue 1, pp. 13–20.

[11] J. Daemen and V. Rijmen, "The design of Rijndael." *AES—The Advanced Encryption Standard* (Springer–Verlag, Berlin, Heidelberg, New York, 2002)
[12] National Institute of Standards and Technology (NIST). "FIPS-197: advanced encryption standard, November 2001." *http://www.itl.nist.gov/fipspubs/*, accessed 18 March, 2006.
[13] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact Rijndael hardware architecture with S-Box optimization." In C. Boyd, editor, *Proc. 7th Int. Conf. on the Theory and Application of Cryptology and Information Security, Advances in Cryptology*, ASIACRYPT 2001, Gold Coast Australia, December 2001, LNCS 2248, pp. 239–254, Springer, 2001.
[14] S. Mangard, M. Aigner, and S. Dominikus, "A highly regular and scalable AES hardware architecture." *IEEE Trans. Comput*., 2003, 52 (4), pp. 483–491.
[15] J. Wolkerstorfer, "An ASIC implementation of the AESMixColumn operation." *Proc. Austrochip 2001*, Vienna, October 2001, pp. 129–132.
[16] N. Pramstaller, S. Mangard, S. Dominikus, and J. Wolkerstorfer, "Efficient AES implementations on ASICs and FPGAs." In H. Dobbertin, V. Rijmen,, and A. Sowa, editors, *Proc. Fourth Workshop on the Advanced Encryption Standard* ''AES—state of the crypto analysis.'' AES 2004, LNCS 3373, pp. 98–112, Springer, 2004.
[17] J. Wolkerstorfer, E. Oswald, and M. Lamberger, "An ASIC implementation of the AES SBoxes Proc." *The Cryptographer's Track at the RSA Conf. Topics in Cryptology*, CT-RSA 2002 San Jose, CA, USA, February 2002, LNCS 2271, pp. 67–78, Springer, 2002.

Batbold Toiruul was born in Ulaangom, Mongolia, on August 9, 1979. He received his B.Sc. in Computer Science from Mongolian National University in 2002, and a M.Sc. in Computer Science from Sunmoon University in 2005. He is now studying Ph.D. courses at Sunmoon University. His areas of interest are RFID and network security.

KyungOh Lee was born in KyungBook, Korea, on July 3, 1965. He received his B.Sc., M.Sc., and Ph.D. in Computer Science from Seoul National University in 1989, 1994, 1999, respectively. In 1999, he joined Sunmoon University. His areas of interest are databases, real-time multimedia, and mobile computing. He is a cochair of the program committee of APIS2004 (Asian–Pacific International Symposium).