Improved Flyweight RFID Authentication Protocol

Thokozani Felix Vallent[1], Eun-Jun Yoon[2], and Hyunsung Kim[2]

[1] Department IT Convergence, Kyungil University / Gyeongsan, South Korea tfvallent@gmail.com
[2] Department of Cyber Security, Hyungil University / Gyeongsan, South Korea {ejyoon, kim}@kiu.ac.kr

* Corresponding Author: Eun-Jun Yoon

Received July 31, 2012; Revised August 6, 2012; Accepted August 19, 2012; Published October 31, 2012

Abstract: The widespread implementation of RFID in ubiquitous computing is constrained considerably by privacy and security unreliability of the wireless communication channel. This failure to satisfy the basic, security needs of the technology has a direct impact of the limited computational capability of the tags, which are essential for the implementation of RFID. Because the universal application of RFID means the use of low cost tags, their security is limited to lightweight cryptographic primitives. Therefore, EPCGen2, which is a class of low cost tags, has the enabling properties to support their communication protocols. This means that satisfying the security needs of EPCGen2 could ensure low cost security because EPCGen2 is a class of low cost, passive tags. In that way, a solution to the hindrance of low cost tags lies in the security of EPCGen2. To this effect, many lightweight authentication protocols have been proposed to improve the privacy and security of communication protocols suitable for low cost tags. Although many EPCgen2 compliant protocols have been proposed to ensure the security of low cost tags, the optimum security has not been guaranteed because many protocols are prone to well-known attacks or fall short of acceptable computational load. This paper proposes a remedy protocol to the flyweight RFID authentication protocol proposed by Burmester and Munilla against a desynchronization attack. Based on shared pseudorandom number generator, this protocol provides mutual authentication, anonymity, session unlinkability and forward security in addition to security against a desynchronization attack. The desirable features of this protocol are efficiency and security.

Keywords: RFID authentication, Pseudo-random number generator, Lightweight authentication protocol, Desynchronization, Forward secrecy, EPCGen2

1. Introduction

The current concern of radio frequency identification (RFID) technology is to diversify its applications to all fields while retaining the users' privacy and security. Typical entities interacting in RFID are backend servers, readers and tags. This paper considers the backend server and reader as a single entity for simplicity. RFID technology is versatile and has more advantages than barcodes. Therefore, it is applied widely in many areas, such as access control, e-passports, library inventory systems, logistics, animal identification, electronic payment systems and many more wireless-identification applications. Hence, RFID surpasses the bar coding method of identification over a number of features, such as no requirement for direct line of sight, high communication speed, large operating and communication range, ability to perform multiple identification with anti-collision in the range of 50-100 tags, hands free operation, requires low power, high data capacity and ability to be integrated to an internet system and reusability of the transponder among others [1-5].

The ubiquitous implementation of RFID suggests the deployment of affordable tags ($ 0.10) to be attached to the item of interest. On the other hand, the technology has challenges in preserving data privacy and data security due to tag's automatic stimulation causing the tag to respond once within the radio frequency (RF) range of a compatible reader except for car transponders, which have enhanced security measures [6, 7]. This need for privacy becomes more imperative, particularly in sensitive applications within the consumers' mainstream, ensuring that the information stored in the is only revealed to authorized parties. Two types of tags are used in RFID

technology. Tags without an internal power source (battery) that rely only on the power supply of the RF field generated by the reader called are *passive tags*. On the other hand, *active tags* have an internal energy power source to supply the circuitry and generate response data. Although their functionality is similar, their main difference is that active tags have higher computational performance. Nevertheless, a passive tag has gained high market share due to the number of preferable properties over active tags, such as low cost, indefinite operational life time, small size fitting for a range of applications and environmental friendly because they do not require the disposal of batteries as active tags do. Despite the diversification of RFID technology, low cost tags have an inherent inability to handle enhanced security and privacy preserving cryptosystems, such as asymmetric and symmetric cryptosystems and some cannot use hash functions. This inadequacy in computational and processing capability is a function of the cost reduction to match with their make-up [1, 2, 8-11]. Therefore, the focus of research is to design compatible lightweight cryptographic measures that are suitable for low cost tags for security purposes. One typical use of RFID is the electronic product code (EPC) system of identifying products, where each product is assigned a unique code that is stored in a tag attached to the product for easy remote identification and tracking. Depending on the system's operational properties of the tags, the EPC system categorizes six indexes of the tag called classes ranging from *class*-0 to *class*-5. *Class*-0 is for read only tags, *class*-1 is for write and read many tags, *class*-2 is for read/write tags and can store dynamic data, *class*-3 is for read/write with on-board sensors tags, *class*-4 is for read/write tags with integrated transmitters and can communicate independent of the readers and *class*-5 is for read/write tags with integrated transmitters that can communicate with passive devices.

RFID air interfaces are governed by two international organizations, EPCGlobal and ISO, which are international standard organizations. Therefore, EPCGen2 stands for EPC class-1 generation-2 ultra-high frequency (UHF), which is an example of EPC systems that focuses specifically on passive tags operating in the frequency range 860-960 MHz, as defined by EPC Global [12]. RFID interoperability Standards of EPCGen2 in an EPCGlobal system are equivalent to the standards of ISO 180006 in the ISO14443 system. Because of its characteristics conducive to the adoption of low cost tags, EPCGen2 is viewed as a standard communication platform for low cost oriented cryptographic protocols [11]. Therefore, many protocols designed to provide the security needs of low cost tags are EPCGen2. The EPCGen2 platform is designed to strike a balance between the cost of tags and functionality with less attention on security in promoting the deployment of low cost tags. Therefore, the security of low cost tag communication protocols are constrained to the lightweight cryptographic primitives of a 16-bit pseudorandom number generator (PRNG), 16-bit cyclic redundancy code (CRC16) and exclusive or (XOR) [8, 9, 13]. The platform has two layers of operation, the physical layer and tag identification layer. The basic operations for the tag identification layer are as follows: *select*-an operation to specify tag population, *inventory*- a singulation operation used to disambiguate a tag out of many and *access*- an operation to read/write on tag. The normal procedure of an inventory operation has the following message flows:

- The server initiates the process by sending a query tag to determine which tag participates in the session.
- Upon receipt of a query, the tag draws a 16-bit random number (RN16) from its PRNG and sets the counter on. Subsequently, the tag backscatters the number to the server when the counter is zeroed because tags use far field coupling to respond to the reader.
- The server sends an acknowledgement message, ACK, back to the tag that requests the product code (PC) or EPC data of the tag.
- The tag then compares the random number from the server and then sends the EPC data if the server's response is found to be valid.

Therefore, the design of reliable EPCGen2 RFID compliant protocols for mutual authentication, which are lightweight for low cost tags that satisfy the communication requirements of data integrity, privacy and untraceability, is the current concern of researchers [14-25]. Accordingly, many lightweight RFID authentication protocols have been proposed but their security was found to be lacking in different respects, as illustrated below [26-29]

- **YA-TRAP protocol [26]:** Prone to replay attacks due to an attacker putting clock forward and re-sending the recorded message.
- **YA-TRAP* protocol [27]:** Prone to replay attacks due a lack of proving the reader's authenticity fully.
- **Chatmon protocol [28]:** Prone to replay attacks and its search will become exhaustive if the key is not stored with the back-end server due to a lack of protocol design.
- **Burmester-Munilla protocol [29]:** Subject to a desynchronization attack due to the dual usage of a random number [3].

This paper presents an improved flyweight RFID authentication protocol to resolve the problems between the tag and server desynchronization discovered in Burmester and Munilla's protocol as a result of a reflection attack. The protocol utilizes the synchronized feature of PRNG that is pre-shared between the tag and server. Therefore, the tag and server have the same algorithm for producing random numbers. Once they share the same seed, they will generate the same pseudorandom numbers. With this architecture, the tag and server authenticates each other by exchanging consistent random numbers drawn from their synchronized state. If the exchanged random numbers are same with each other's state, they respond throughout the challenge phase. Hence, the tag cannot be cloned because synchronization is based on a variable shared secret determined by the PRNG only. To

thwart attacks, the internal state of the PRNG is also refreshed systematically in a synchronized manner between the parties. This eventually solves the problem of backward and forward secrecy because it frustrates statistical analysis. The protocol presented in this study retains all the desirable security features of Burmester and Munilla's protocol while remaining secure from desynchronization attacks.

The remaining sections of this paper are organized as follows: Section 2 gives an awareness of the communication model, security threats and security features. Section 3 reviews Burmester and Munilla's protocol and highlights its weakness, while section 4 presents the improved flyweight RFID authentication protocol. Section 5 provides security analysis and performance analysis. Finally, section 6 reports the conclusions.

2. Preliminaries

This section first presents a system communication model for how entities interact in a wireless network environment. The tag is envisioned as a mobile communicating entity that automatically relays messages with a reader using RF. The reader is an entity connected physically to a back-end server that contains all records of the registered tags. Fig. 1 illustrates the interaction of the entities. The section also mentions the critical threats and attacks associated with RFID technology. The later part of the protocol considers security objectives.

2.1 Communication model

As shown in Fig. 1, a RFID system can be partitioned into three basic functional blocks, which are the tag, reader and back-end server. Although the reader is connected to a server that contains a database of registered tags, the linkage is considered as a single block and is secure in many instances. (1) The reader communicates remotely with the tag once it interfaces with the reader's radio frequency field. In turn (2), the tag responds with its identifying information to the reader that (3) transfers the tag's response to the server in what is known as "interrogator talks first" communication concept. (4) The servers' message passes through the reader to the tag in that manner until the transaction is complete. The reader-tag communication takes two forms, either full duplex or half duplex communication, depending on the way the tag uses the RF energy to respond. In a full duplex transmission, the tag uses the RF energy to respond to the reader's message at the same time as it is supplied simultaneously. On the other hand, the response of the half duplex communication system tag and RF energy supply occur sequentially and intermittently. Half duplex transmission involves radio frequency modulation of the tag with the assistance of a storage capacitor before responding to the reader.

Therefore, the half duplex mode is an ideal candidate for EPCGen2 applications because it is the cheapest two way enabling communication system and the transmission has a lower collision incidence [7]. Furthermore, the two way property in the half duplex communication model enables the implementation of mutual authentication protocols between the tag and reader. In terms of information security, the remote access feature of the reader-tag transmission and the automatic activation of the tag whenever in the reader's RF field are the major concerns for people's identity safety and personal information privacy in the implementation of RFID technology. Therefore, finding measures that are privacy preserving and security enhancing in the interplay between the reader and tag wireless communication is the goal of authentication protocols.

2.2 Threats and attacks

RFID threats and attacks occur in different manners and different application areas [6]. For example, a typical threat to privacy pops up in the implementation of e-passports if personal information is broadcasted involuntarily to any transceiver without range and tags consent. This would lead to other malicious instances because some readers might be counterfeit ones, resulting in potential identity theft and privacy violations, such as tracing. An adversary has unlimited capability to carrying out attacks due to eavesdropping, intercepting and modifying messages between the tag and reader if not secured. Some of the well-known attacks in RFID, which are a concern towards the widespread of the technology in the roaming ubiquitous computing, include

- **Tag cloning:** An attacker obtains the tag's identifying information and uses it instead of a legal tag
- **Tag tracing:** An attacker tracks the tag's protocol flows using its identity obtained from a previous transaction.
- **Replay attack:** An attacker uses the tag's responses to the reader to represent itself as the legal tag in a later interrogation.
- **Impersonation attack:** An attacker tries to present

Fig. 1. RFID communication system architecture.

its information as if it is a trustworthy tag to obtain access or authentication illegally from the reader.
- **Online man in the middle attack:** An attacker intrudes and intercepts the messages between the server and tag, and injects false messages or tampers with the normal way of correspondence.
- **Desynchronization:** An attacker causes the server and tag to have inconsistent values that leaves the legal entities unable to authenticate each other.

To this effect, the main concern of this protocol is to prevent attacks that exploit the relationship between successive interrogations to carry out an attack or tamper with the security mechanism for authentication between the server and tag and safeguards against desynchronization.

2.3 Security objectives

Security measures of the wireless channel should consider the EPCGen2 compliant cryptographic tools in achieving the basic security requirements of privacy, confidentiality, integrity, tracking and tag anti-cloning measures. The wireless channel can be protected in two ways, securing tag-to-reader communication or making the tags responses indistinguishable from random data that would provide no useful information to an eavesdropper. The protocol utilizes the latter because it is based on the PRNG, which is good for the protocol's scalability. By using this aspect of pseudo random numbers, the protocol shows good efficiency. Consistent with the EPCGen2 security measures, this protocol was designed to attain security and privacy preservation while achieving the following security features.

- **[FE1] Mutual authentication**: This is a security feature by which each party between the server and tag proves the identity of the other before they exchange valuable information.
- **[FE2] Session unlinkability:** This is a situation where an attacker cannot determine any relationship between any two successfully completed consecutive interrogations.
- **[FE3] Forward secrecy:** Future outputs, after refreshment, look randomly to an attacker even if the state of the pseudorandom number generator of the tag is accessed by the attacker before refreshment.
- **[FE4] Desynchronization property:** This is a security feature where a synchronization from a prior shared algorithm with certain secret values between parties is disturbed so they can no longer be synchronized by using it.

3. Flyweight RFID authentication protocol

In light of necessitating the ubiquitous use of low-cost tags, many light authentication protocol have been proposed, even though many have failed to achieve their security claims. Therefore, in a quest to design a robust EPCGen2 protocol, Burmester and Munilla proposed a flyweight RFID authentication protocol (FRAP) [29]. FRAP aims to provide a solution to the security threats occurring in insecure communication between the tag and reader by using lightweight primitives satisfying the EPCGen2 standards. The cryptographic tool used to achieve the security of the communication protocol utilizes the PRNG shared between the tag and server. Basically the tag and server share the same PRNG that produces random numbers and the parties determine each other's authenticity by just sending pseudo random numbers sequentially to each other as determined by the PRNG output. Therefore security is also retained by updating their state at the same instances in a similar manner to thwart all attempts to analyze the transmitted messages. Although FRAP achieves mutual authentication by the exchange of pseudo-random numbers, which is a natural way of coping with synchronization, FRAP's design provides room for a desynchronization attack. This flaw is initiated by a reflection attack that ends up causing the tag and server to bear different pseudo random numbers. This means that all subsequent pseudo random numbers at any instance in time will not be tallied between them. Desynchronization eventually leads to the denial of service (DoS) of a legitimate tag [3].

3.1 Notations

From now on the following notations will be used to present the protocol flows.

S	: Server
T	: Tag
ID_{tag}	: Tags identity
g_{tag}	: State generated by $g(.)$, which is a hash function
$\partial(.)$	: Transition function/sate freshening function, which is the same as $f(.)$
$h(.)$	: Pseudo random number generator
RN_i	: pseudo random number, for $1 \leq i \leq n$ for number of rounds n
DB	: Server's data base
K^r	: Refreshing key
S_i	: State
cnt	: Counter
$alarm$	: Session status check

3.2 Flyweight RFID authentication protocol procedure

FRAP's message flow is devised as a function, $g_{tag}=g_{tag}(\text{state})$, which is used to randomize the responses that lead to authentication of the tag and server. The relationship can be transformed to envision the computational operations involved in refreshing and generation of the transmitted random numbers. For simplicity, $h(x)=g(f(x))$ is considered an analogous transformation of the function representing the PRNG. Therefore, the PRNG has two components, the state $f(x)$ and the generate $g(f(x))$. To refresh the PRNG, $h(x)$, means the updating state $f(x)$ with fresh randomness (new input) because it is the variable of the entire composition of the function. This means that if a server and tag share the same function, e.g. $g(f(x))$, they will produce the same output

provided the input (the seed) is the same. To determine the security of the random numbers produced, it is essential that the inputs be of high entropy to overcome an exhaustive search attack. FRAP works in the same general concept outlined thus far.

As simplified already, the tag and server share the PRNG $g(s_i)$, state s_i and the same preloaded secret key s_0, which is known as a seed. Basically, s_0 can be composed of a refreshing key, K^r and other identifying parameters pre-shared. Therefore, in a similar manner to the analogy given above, the procedure for authentication follows the same order of drawing pseudorandom numbers from the same state to output other pseudo random numbers. Obviously no other party could predict the exchanged pseudo random numbers precisely without knowledge of the interplay functions governing the generation of pseudo random numbers. Therefore, the security of FRAP hinges on the secrecy of the K^r and other pre-shared pseudo random numbers, either $RN_1 = RN_1^{cur}$ or $RN_1 = RN_1^{next}$. Following the recursive definition of the function PRNG with continuous work, such as $g(s_i)=s_{i+1}$, in producing pseudo random numbers, whenever the tag is queried by a reader it reports the value $g(s_i)$ for the tag's secret s_i during the ith message flow. This is possible because the tag is already initially preloaded with s_0. Using the randomizing algorithm, which is a refreshing (transition) function of the state, i.e. $\partial(RN_i, K^r) = RN_{i+1}$, FRAP generates new set of pseudo random numbers and guarantees the security of previous transactions because all stored data then is erased.

Systematic exchange of the pseudo random numbers at particular instances of the PRNG leads to mutual authentication of the tag and server. Therefore, the rule is to utilize two different functions to produce pseudo random numbers that use a similar algorithm, which is to generate pseudo random numbers. For example, $\partial(RN_i, K^r)$ generates pseudo random numbers as does $g(s_i)$, and $g(s_i)$ inputs the outputs of $\partial(RN_i, K^r)$, so we have the function, $g(\partial(RN_i, K^r))$, as required. S initiates the communication procedures. Once established, both parties exchange and update the proper indexed pseudo random number or their states. On the onset of a communication procedure, the server stores in its database { RN_1^{cur}, RN_2, RN_3, RN_4, RN_5, RN_1^{next}, ID_{tag}, g_{tag}, K^r, cnt } in that order whilst the tag stores { RN_1, RN_2, ID_{tag}, g_{tag}, K^r, cnt }. By exchanging these random numbers drawn from the generator, authentication can be achieved if the exchanged pseudo random numbers tally between the parties. Fig. 2 presents the flow of FRAP.

Step 1. (S→*T*) *query*
The communication procedure is initiated by *S* by sending a query to *T* upon interfering with the RF. which is the undisputable mode of the RFID communication system whenever a tag is stimulated. On the other hand, it is perceived that future tags would also have the ability to initiate an interrogation.

Step 2. (*T*→*S*) RN_1
Upon receipt of a query, *T* responds with a challenge RN_1 drawn from its state g_{tag} and switches on *alarm*←*cnt*←1 to check the transmission delay and transmission errors.

Step 3. (S→*T*) RN_2
When *S* receives RN_1, it checks the *DB* to determine if it has RN_1 appearing and in what format it is appearing. Therefore, the *S* checks whether the format is RN_1^{cur} or RN_1^{next} according to the way. If *S* finds that $RN_1 = RN_1^{cur}$ it simply sets *alarm*←*cnt*←1 and sends RN_2 to the *T*. On the other hand, if *S* receives $RN_1^{nex\ t}$ in the first place, it updates its state before broadcasting RN_2. Otherwise, *S* does not recognize *T* if RN_1 is non-existent in *DB*. In a similar manner, as RN_1 authenticates the legitimacy of *T* to *S,* RN_2 acts as a message authentication code *S* to *T*. This is perceived easily from the fact that no one else could predict RN_1 if not *T* due to it being a pseudo random number. In the same vein to *T* conception, no one else could respond with a correct RN_2 without knowledge of the pre-shared PRNG and seed.

Step 4. (*T*→*S*) RN_3 or RN_4
After authenticating RN_2, *T* proceeds by drawing five successive random numbers, RN_3, RN_4, RN_5, RN_1 and RN_2 in that order from g_{tag} because it does not have any more pseudo random numbers to carry on with the authentication process. *T* then sends RN_3 to *S* as its confirmation message in the case of clock setting, *alarm*←0 so it sets *cnt*←0 to indicate no transmission error. Therefore, the synchronizing state should remain in the current state. On the other hand, there are sometimes transmission errors and a need for cooperatively re-authenticating each member. The procedure then takes two additional message flows. In the case of transmission error, the clock indicates *alarm*=1. Therefore, tag instead sends RN_4 to *S*. Otherwise *T* quits. When *S* receives RN_3, it checks its validity and cross checks the clock. If the clock indicates *alarm*=0 then *S* authenticates *T* as valid. Otherwise if *S* receives RN_4, it sets $RN_5 = RN_5^{cur}$ and sends RN_3 to determine if the error is genuine or if it is a result of a malicious act. This assumes that if *T* is legal he/she will be able to send a valid re-authentication message. Subsequently, *S* updates its state otherwise it quits if RN_4 is invalid in the first place.

Step 5. (S→*T*) RN_3
T checks if RN_3 is valid, i.e. if *alarm*=1. If so, it broadcasts RN_5. Otherwise, *T* quits the operation.

Step 6. (*T*→*S*)
S checks the validity of RN_5, i.e. if $RN_5 = RN_5^{cur}$. When the checking holds, *S* authenticates *T* otherwise it quits if RN_5 is invalid.

Because the basis of FRAP is the exchange of pseudo random numbers from a synchronized generator for authentication, it means any measure that can break the bond would jeopardize the secure communication. This protocol, FRAP, has a weakness of desynchronization attack emanating from the dual use of RN_3 in steps 4 and 5 in two incidences of a passive attacker and active attacker, respectively.

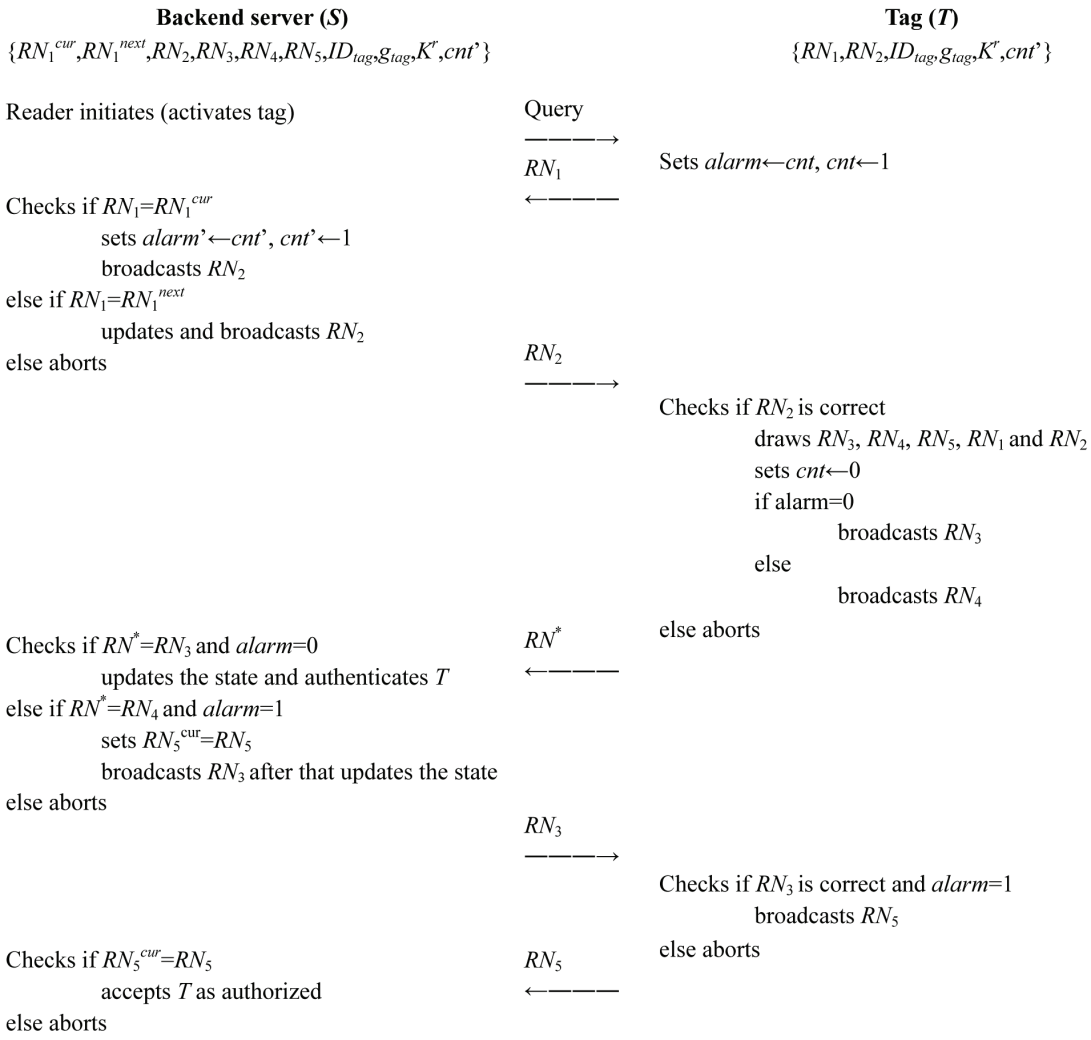

Fig. 2. Flyweight authentication RFID authentication protocol.

3.3 Desynchronization attack

Any possible measure to disturb the smooth flow of messages between T and S will eventually lead to desynchronization. This is because the security of the two parties can be defeated if a replay message is relayed at a faster speed than the set time interval of the tolerance mean of transmission. This is the scenario of an on-line-replay attack. The following gives an outline of how the synchronization bond is broken.

Step 4. ($T \rightarrow A$) RN_3
An attacker intercepts the authenticating message, RN_3 from T to S and then reflects it back to T before threshold time interval t_h has elapsed.

Step 5. ($A \rightarrow T$) RN_3
Upon the receipt of RN_3, T checks its validity and the clock reading. Obviously being a reflection, RN_3 will be valid. Because of the transmission delay experienced from steps 3 to 5, the clock will read $alarm \leftarrow 1$ thereby validating the received message. T will then send RN_5 trusting that it is communicating with the legal server. Even if A intercepts RN_5, the state of S and T will not tally on two accounts. If RN_5 is intercepted again by A clearly, S will quit the session simply because of the time delay and differing steps. S would be expecting RN_3 or RN_4. Therefore, it will not accept RN_5 and will quit. Although A intercepts the message, RN_5, still more S will quit with the state of step 2 while T quits with state of step 3. Therefore, due to quitting, their states at different stages means they bear different re-seeding inputs for subsequent communication. Consequently, the S and T become desynchronization because the tag's state would be ahead of the server's. Hence, for future communication, S will not recognize T's state and will deny the tag service. This is an occurrence of a DoS. A reflection attack is possible because of the FRAP's design by nature to accept RN_3 in both directions at different stages.

4. Improved flyweight RFID authentication protocol

To overcome the desynchronization problem of FRAP, this paper presents an improved flyweight RFID authentication protocol. An essential measure to avoid the

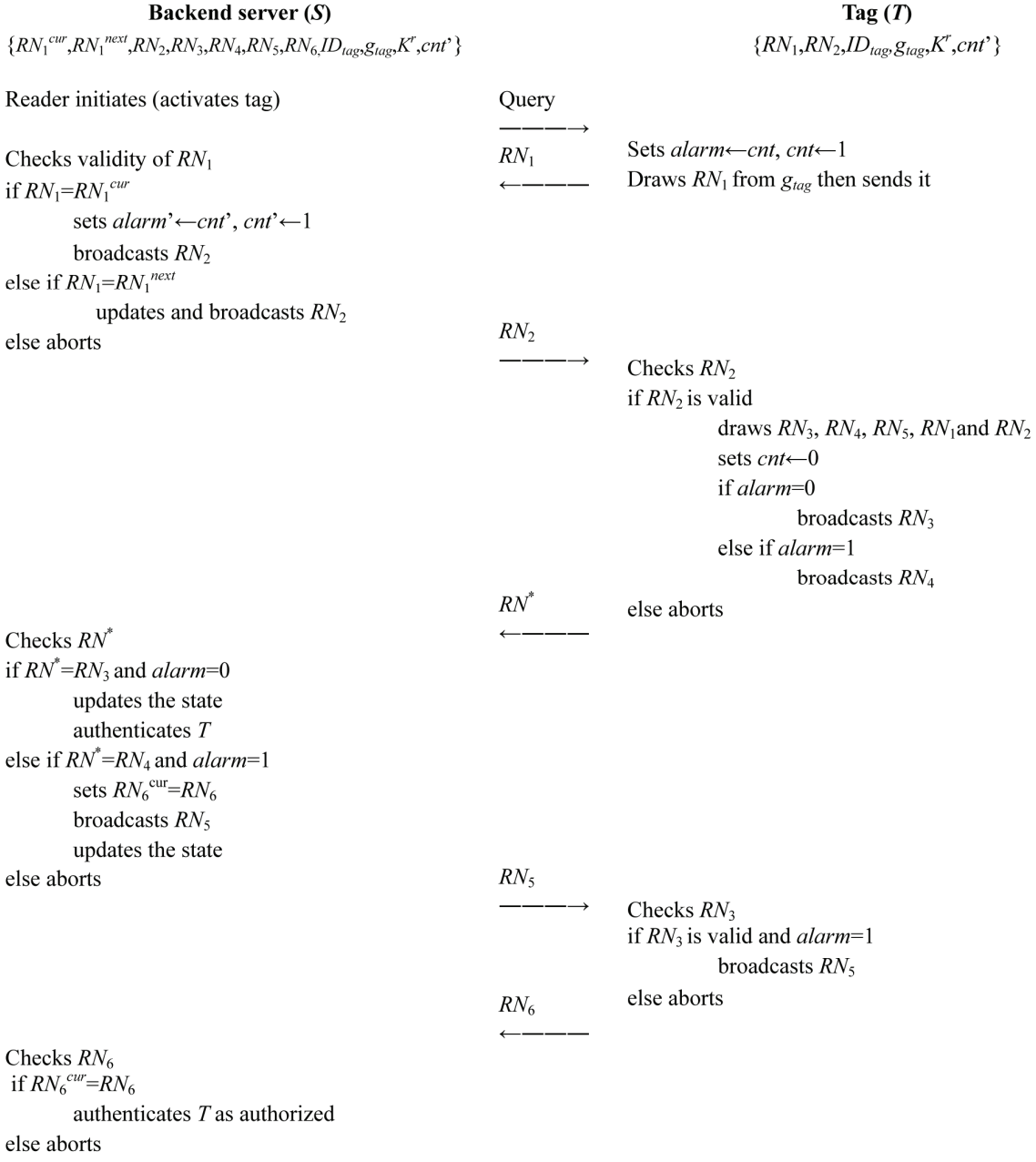

Fig. 3. Improved flyweight authentication RFID authentication protocol.

desynchronization problem is to design a protocol that disregards queries from unauthorized entities. The solution is just re-designing the protocol so that tag does not accept a reflection of RN_3. By eliminating the critical problem, the protocol accomplishes the desirable security features with just a minimal and bearable computational load. *S* and *T* will now communicate effectively and securely because the protocol accomplishes the desirable security features of mutual authentication, anonymity, session unlinkability, forward secrecy and backward secrecy. The novelty of our protocol to FRAP is that it uses 6 pseudo random numbers where individual pseudo random numbers are used only once. The reason for the unique use of pseudo random numbers is to avoid the dual usage of RN_3, which is the root cause of a desynchronization attack that leads to a DoS. Therefore, the server's *DB* will contain { RN_1^{cur}, RN_1^{next}, RN_2, RN_3, RN_4, RN_5, RN_6, ID_{tag}, g_{tag}, K^r, *cnt* } while the tag memory stores { RN_1, RN_2, ID_{tag}, g_{tag}, K^r, *cnt* }. The proposed protocol also uses a refreshment of state to prevent an attacker from exploiting the deterministic nature of the PRNG to simulate the tag's output. The proposed protocol retains all the merits of FRAP while strengthening security. The proposed protocol is discussed as follows:

Step 1. (S→*T*) *query*
S initiates by querying tag.

Step 2. (*T*→*S*) RN_1
T becomes powered and sets *alarm*←*cnt*←1, draws RN_1 from its memory and then sends to *S*. The sent random number is pre-shared with the server. On the tag's side, it serves as a seed to determine the random

numbers of the following flows.

Step 3. ($S \rightarrow T$) RN_2

S checks RN_1 in DB and in what format to determine how to proceed. If $RN_1 = RN_1^{cur}$, it means proceeding with readily available random numbers so S also sets $alarm \leftarrow cnt \leftarrow 1$ and sends RN_2. Otherwise, if S receives $RN_1 = RN_1^{next}$, it implies a renewal of the set of random numbers to be used. Therefore, S updates the state and sends the subsequent random number RN_2 as the response of RN_1. Otherwise, S aborts the session if RN_1 is invalid.

Step 4. ($T \rightarrow S$) RN_3 or RN_4

T checks validity of RN_2 to confirm the authenticity of S. If RN_2 is correct then T draws six random numbers from g_{tag} and assigns them to the variables, RN_3, RN_4, RN_5, RN_6, RN_1 and RN_2 in that order. Subsequently, T sends RN_3 and sets $alarm \leftarrow cnt \leftarrow 0$. Otherwise, T sends RN_4 or aborts. The basic assumption is that no other entity can send T correct synchronized subsequent random numbers apart from real S.

Step 5. ($S \rightarrow T$) RN_5

Similarly, if S receives RN_3, S checks if RN_3 is correct to confirm if T's message tallies with what it has in its DB and if the equation, $alarm$=0, holds. Only if the validations are satisfied, S authenticates T. On the other hand, if S receives RN_4 and if the equation $alarm$=1 holds, S sets RN_6=RN_6^{cur} and sends RN_5. The received message RN_4 implies the notification of a transmission error and the need for re-authentication. This is why S goes ahead with and additional challenge RN_5 to T. Otherwise, S aborts the session.

Step 6. ($T \rightarrow S$) RN_6

In turn, T checks if RN_5 is valid and $alarm$=1. Only if they are valid does T sends RN_6 back to S. After receiving the message, S checks if the equation RN_6=RN_6^{cur} holds. If so, S authenticates T.

If the rounds are completed successfully, the two parties achieve mutual authentication and can progress to communicate accordingly.

5. Security and performance analysis

This section reports how the protocol satisfies the security requirements that make it comparably better than FRAP. The protocol addresses the problem of desynchronization posed by an adversary intercepting the message flow from the tag to the server and then reflects which necessitates desynchronization that eventually leads to a DoS. This paper also shows how the protocol achieves mutual authentication, session unlinkability, forward secrecy, desynchronization attack resilience, replay attack and anonymity. These are well known desirable security features and a check point for many studies to determine if they satisfy the expected standards.

5.1 Security analysis

The proposed protocol retains all the desirable security features of Burmester and Munilla's FRAP protocol in [29] while standing secure from a desynchronization attack. A comparison of FRAP with other related protocols showed that the proposed protocol maintains all the security merits and solves the desynchronization problem in [29], as shown in Table 1.

Table 1. Security feature comparisons among the related RFID authentication protocols.

Features / Protocols	FE1	FE2	FE3	FE4	FE5
YA-TRAP in [26]	No	Strong	Strong	Yes	Weak
YA-TRAP* in [27]	No	Strong	Strong	No	Weak
Chatmon in [28]	No	Strong	Strong	No	Weak
FRAP in [29]	Yes	Strong	Strong	No	Weak
Our protocol	Yes	Strong	Strong	Yes	Strong

FE1: Mutual authentication, FE2: Session unlinkability, FE3: Backward secrecy, FE4: Desynchronization possibility FE5: Replay attack resistance

5.1.1 Mutual authentication

Each party cross checks the validity of the pseudo random numbers received against its own state. On the other hand, the counter is used at every step to check the freshness of the message and an alarm is used to measure the propagation time between sending and receiving of the message. For example, the server needs to cross check RN_1, RN_3, RN_4 and RN_6 from the tag and in the same way, the tag verifies the validity of RN_2 and RN_5 from the server. Because the state generated g_{tag} is pre-shared, the renew of the state $g(s_i)$=s_{i+1} between S and T should tally for the same input. Therefore, T and S authenticate each other by exchanging such numbers.

5.1.2 Session unlinkability

The protocol provides session unlinkability by virtue of refreshing the state for the server and tag use different temporary identification. This feature is provided by refreshing the secret information stored in the tag. In such instances, the old data is deleted giving no clue to an attacker wishing to extract the secret from analyzing the session outputs. Therefore, the protocol gives no opportunity for tracing by eavesdropping. This is a desirable feature for RFID applications regarding its radio frequency communication for generating pseudo random numbers, and intrinsically provides forward secrecy and backward secrecy simply because the random numbers used for each transaction are totally independent and cannot be related across sessions.

5.1.3 Forward Secrecy

An attacker would not succeed in obtaining subsequent outputs RN_1, RN_2, ..., RN_i just by knowing the current or previous states s_1, s_2, ..., s_i and the transitional function, $\partial(RN_i, K')$=RN_{i+1}, which is not invertible but occurs in one way. Because the numbers are generated and used once followed by refreshment, the past outputs are untraceable. In this way, the attacker would not succeed in calculating

or simulating future outputs by knowing the current or previous state.

5.1.4 Desynchronization attack resilience

The proposed protocol avoids the dual usage of RN_3 for authentication from the server to tag and vice-versa, and instead introduces RN_6 to ensure every number is used only once and in a single direction. Therefore the critical pseudo random numbers RN_3, RN_5 and RN_6 cannot be compromised by reflection because they are just meant to be used only once for one-way authentication. Suppose an attacker tries to reflect RN_3 in step 4 before the message reaches the server. If the attacker reflects the messages to the tag, the attack will be detected immediately and the session will discontinue with S and T updating state synchronically in the same step using their currently synchronized state. This is simple because the protocol is designed to accept a unique number only once in a single direction, which is in contrast to the previous protocol FRAP. This means that the protocol is now safe from a desynchronization attack.

5.1.5 Replay Attack

The constant updating and reseeded of the state and the unique usage of every pseudo random number protects the protocol from replay attack because the function, RN_{i+1}, is a one way recursive function that produces pseudo random numbers of state. The one way feature means that once a pseudo random number is used in a session, it will not be re-used owing to the randomness property of the PRNG and the synchronization property of the tag and the server. This eventually protects the scheme from a replay attack because the protocol updates the state when closing of a session to a random value of RN_{i+1}, which is unknown to an attacker. If any replay is attempted, it will definitely differ with the current state and will not be accepted.

5.1.6 Anonymity

Authentication is based on the exchange of variable pseudo random numbers rather than using distinctive identities. The parties in play verify each other by the consistency of the random numbers exchanged. The assumption is that no attacker can deduce the exchanged random numbers correctly within the specific threshold time. By this feature, anonymity can be achieved because no attacker can know the precise identity (ID_{tag}) of the tag by just observing the transmitted pseudo random numbers.

5.2 Performance evaluation

The proposed protocol does not use heavy computational functionality and complies with the EPCGen2 standards. Compared to FRAP in [29] and other related protocols in [26-28], the proposed protocol maintains all the merits and has the same computational load with just a trivial addition in storage memory for RN_6 only. The proposed protocol is complete in four steps in the case of a passive attacker (optimistic property) and takes six steps in the case of an active attacker, as shown in Table 2.

Table 2. Performance comparisons among the related RFID authentication protocols.

Features / Protocols	Tag		Server	
	Hash operation	Random generation	Hash operation	Random generation
YA-TRAP in [26]	2	1	n+1	None
YA-TRAP* in [27]	2+X	3	n+2	None
Chatmon in [28]	3	1	2n+1	None
FRAP in [29]	1	5	1	5
Our protocol	1	6	1	6

6. Conclusion

FRAP, regardless of its security checks per each session, is still vulnerable to reflection attacks that causes desynchronization and eventually leads to denial of service attacks. By avoiding the dual use of RN_3, this paper proposed an equally EPCGen2 compliant protocol called improved flyweight RFID authentication protocol, whose security strength rests in the unique use of random numbers. To avoid the use of a single pseudo random number in two different directions, RN_6 was introduced to ensure unidirectional flow of any particular random number. Another important property is that in case of malicious interception, the server and tag updates their states synchronically using their immediate undisturbed states. All the privacy-preserving security measures of the previous protocol are retained in the proposed protocol while remaining efficient and safe from desynchronization attacks. The mentioned security features satisfied are the mutual authentications, session unlinkability, forward secrecy and anonymity.

References

[1] D. N. Duc and K. Kim, "Defending RFID authentication protocol against DoS attacks", *Journal of Computer Communication*, Vol. 34, No. 3, pp. 384-390, 2011. Article (CrossRef Link)

[2] D. Han, "Gröbner Basis Attacks on Lightweight Authentication Protocols RFID", *Journal of Information Processing Systems*, Vol. 7, No. 4, pp. 691-706, 2011. Article (CrossRef Link)

[3] T. Vallent and H. Kim "Cryptanalysis of Flyweight RFID Authentication Protocol", 2012 *IEEEK Summer conference*, 2012. Article (CrossRef Link)

[4] H. Jannali and A. Falahat, "Cryptanalysis and Enhancement of Low cost RFID Authentication protocol", *International Journal of UbiComp*., Vol. 3, pp. 1-9, 2012. Article (CrossRef Link)

[5] J. Munilla and A. Peinado, "Distance Bound Protocol with void-challenge for RFID," *Proc. of workshop on RFID Security*, 2006. Article (CrossRef Link)

[6] http://www.smartcardalliance.org/pages/publications-epc-gen2-faq. Article (CrossRef Link)
[7] P. Kitsos and Y. Zhang, *RFID Security-techniques, protocols and system-on-chip design*, Springer, 2008. Article (CrossRef Link)
[8] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. E Tapiadar, A. Ribagorda, "LAMED-A PPRNG for EPC Class-1Generation2 RFID Specification", *Journal of Computer Standard and Interface*, Vol. 31, No. 1, pp. 88-97, 2009. Article (CrossRef Link)
[9] G. Avoine, X. Carpent, B. Martin, "Privacy-Friendly Synchronized Ultra-lightweight authentication Protocols in Storm", *Journal of Network and Computer Application*, Vol. 35, No. 2, pp. 826- 843, 2012. Article (CrossRef Link)
[10] K. Ahsan, H. Shan, P. Kingston, "RFID application: Introduction and exploratory study", *International Journal of Computer Science Issues*, Vol. 3, No. 3, pp. 1-7, 2010. Article (CrossRef Link)
[11] P. Peris-Lopez, Hernandez-Castro, J. M. E Tapiadar and J. C. A van der Lubbe, "A Cryptanalysis of EPC Class-1 Generation-2 Standard Compliant Authentication Protocols", *Engineering Applications of Artificial Intelligence*, Vol. 24, pp. 1061–1069, 2011. Article (CrossRef Link)
[12] H.-M. Sun and W.-C. Ting, "A Gen2 Based RFID Authentication Protocol for Security and Privacy", *IEEE Transaction on mobile Computing*, Vol. 8, No. 8, pp. 1052-1062, 2009. Article (CrossRef Link)
[13] X. Yi, L. Wang, D. Mao, Y. Zhan, "An Gen2 Based Security Authentication protocol for RFID system", *2012 International Conference on Applied Physics and Industrial Engineering*, Vol. 24, pp. 1385-1395, 2012. Article (CrossRef Link)
[14] M. Burmester and M. R. "Robust, Anonymous RFID Authentication with Constant Key-Lookup", *Proc. of ACM Symposium on Information, Computer and Communication Security*, pp. 283-291, 2008. Article (CrossRef Link)
[15] C.-L. Chen and Y.-Y. Deng, "Conformation of EPC Class 1 Generation 2 standards RFID system with mutual authentication and privacy protection", *Engineering Application of Artificial Intelligence,* Vol. 22, No. 8, pp. 1284-1291, 2009. Article (CrossRef Link)
[16] H.-M. Sun and W.-C Ting, "A gen2-based rfid authentication protocol for security and privacy", *IEEE Transaction on Mobile Computing*, Vol. 99, No. 1, pp. 1052-1062, 2009. Article (CrossRef Link)
[17] C. Qingling, C .Z. Yiju, and W. Yonghua, "A minialist mutual authentication protocol for RFID systems and BAN logic analysis", *Computing, Communication, Control and Manage*, *ISECS International Colloquium*, pp.449-453, 2008. Article (CrossRef Link)
[18] D. Sae, J. Baek and D. Cho, "Secure RFID Authentication Scheme for EPC Class Gen2", *Proc. of 3rd Int. Conf. on Ubiquitous Information management and Communication (ICUIMC-2009)*, pp. 221-227, 2009. Article (CrossRef Link)
[19] A. Juels, "Strengthening EPC tags against cloning", *Proc. of 4th ACM Workshop on Wireless security*, pp. 67-76, 2005. Article (CrossRef Link)
[20] C. Chatmaon, T. Le, M. Burmester, "Secure Anonymous RFID Authentication Protocols", *Technical Report*, pp. 1-10, 2006. Article (CrossRef Link)
[21] S. Boyean and C. J. Mitchell, "RFID Authentication Protocol for Low Cost tags", *Proc. of ACM Conference on Wireless Network Security,* pp. 140-147, 2008. Article (CrossRef Link)
[22] G. R. T. White, "A comparison of Bar coding and RFID Technologies in Practice", *Journal of Information Technology and Organizations*, Vol. 2, pp. 119-132, 2007. Article (CrossRef Link)
[23] W. Killmann and W. schindler, *A Proposal for: Functionality class for random number generators,version2.0*, 18 September 2011. Article (CrossRef Link)
[24] B. Barak and S. Halevi, "A model and architecture for Pseudo-Random Generator and applications to Dev/Random", *Proc. of the 12th ACM conference on Computer and communications security*, pp. 203-212, 2005. Article (CrossRef Link)
[25] M. Moessner and G. N. Khan, "Secure authentication scheme for passive C1G2 RFID tags", *Computer Networks*, Vol. 56, pp. 273-286, 2012. Article (CrossRef Link)
[26] G. Tsudik, "YA-TRAP: Yet another trivial RFID authentication protocol", *Proc. of the 4th annual IEEE International Conference on Pervasive Comput. Commun. Workshops*, pp. 640–643, 2006. Article (CrossRef Link)
[27] G. Tsudik, "A family of dunces: trivial RFID identification and authentication protocols", *Proc. of the Symposium on Privacy-Enhancing Technologies*, 2007. Article (CrossRef Link)
[28] C. Chatmon, T. van Le and M. Burmester, "Secure Anonymous RFID Authentication Protocols", *Technical Report TR-060112*, Florida State University, 2006. Article (CrossRef Link)
[29] M. Burmester and J. Munilla, "A Flyweight RFID Authentication Protocol", *Proc. of Workshop on RFID Security*, 2009. Article (CrossRef Link)

Thokozani Felix Vallent received his B.Ed. in Mathematics from University of Malawi-Chancellor College, Malawi, in 2007. Currently he is a graduate student of Department of IT Convergence at Kyungil University, Republic of Korea. His current research interests are cryptography, authentication technologies, RFID security, cognitive radio network security and u-healthcare security.

Eun-Jun Yoon received his MSc degree in computer engineering from Kyungil University in 2002 and the PhD degree in computer science from Kyungpook National University in 2006, Republic of Korea. From 2007 to 2008, he was a full-time lecturer at Faculty of Computer Information, Daegu Polytechnic College, Republic of Korea. From 2009 to 2011, he was a 2nd BK21 contract professor at the School of Electrical Engineering and Computer Science, Kyungpook National University, Republic of Korea. Currently, he is a professor at the Department of Cyber Security, Kyungil University, Republic of Korea. His current research interests are cryptography, authentication technologies, smart card security, network security, mobile communications security, and steganography.

Hyunsung Kim received his M.E. and Ph.D. degrees in Computer Engineering from Kyungpook National University, Republic of Korea, in 1998 and 2002, respectively. From 2000 to 2002, he worked as a senior researcher at Ditto Technology. He had been an associate professor from 2002 to 2011 with the Department of Computer Engineering, Kyungil University. Currently, he is an associate professor at the Department of Cyber Security, Kyungil University, Republic of Korea. His current research interests are cryptography, VLSI, authentication technologies, network security and ubiquitous computing security.

Copyrights © 2012 The Institute of Electronics Engineers of Korea