

# ITPMAP: An Improved Three-Pass Mutual Authentication Protocol for Secure RFID Systems

Mohammed Issam Younis<sup>1</sup> · Mustafa Hashim Abdulkareem<sup>1</sup>

Published online: 17 April 2017  
© Springer Science+Business Media New York 2017

**Abstract** Radio frequency identification (RFID) is a wireless technology used in various applications to minimize the complexity of everyday life. However, it opens a large number of security and privacy issues that require to be addressed before its successful deployment. Many RFID authentication protocols are proposed in recent years to address security and privacy issues, and most of them are based on lightweight cryptographic techniques such as pseudo-random number generators (PRNGs), or bitwise logical operations. However, the existing RFID authentication protocols suffer from security weaknesses, and cannot solve most of the security and privacy problems. A new solution is necessary to address security and privacy issues. In this paper, an improved three-pass mutual authentication protocol (ITPMAP) for low-cost RFID tags is proposed to offer an adequate security level for RFID systems. The proposed ITPMAP protocol uses one PRNG on the tag side and heavy-weighted cryptographic techniques (i.e., digital signature and password-based encryption schemes) on the back-end server side instead of lightweight cryptographic techniques to address the security and privacy issues. The ITPMAP protocol is secure against various attacks such as cloning, spoofing, replay, and desynchronization attacks. Furthermore, as a proof of concept, the ITPMAP protocol is adopted to propose the design of three real-life RFID systems; namely: Signing and Verification of Graduation Certificate System, issuing and verification of e-ticketing system, and charging and discharging of prepaid card system. The Unified Modeling Language is used to demonstrate the design of the proposed ITPMAP protocol and systems. Java language is used for the implementation of the proposed systems. In addition, the “Mifare Classic” tags and readers are used as RFID apparatuses for the proposed systems.

**Keywords** RFID · Security · Privacy · Authentication protocols · Non-repudiation · ECDSA · PBE · UML

---

✉ Mohammed Issam Younis  
younismi@gmail.com; younismi@coeng.uobaghdad.edu.iq

<sup>1</sup> Computer Engineering Department, College of Engineering, University of Baghdad, Jadryah, Baghdad, Iraq

## 1 Introduction

The RFID technology was first employed in the Second World War to recognize and differentiate between the aircraft of friend and foe. Recently, the RFID technology has been considered as the main driver of the future ubiquitous technology. It is also claimed as the core technology to realize the internet of Things (IoT) environment and facilitates the transition to the internet of everything (IoE) environment where all the physical objects are connected anytime and anywhere. It is supposed that RFID will play an important role for future ubiquitous society [1], as well as, in the Internet of Every Thing (IoE) era. Generally, the RFID technology utilizes radio frequency (RF) electromagnetic signals to exchange data between an RFID reader and tag. Ordinarily, RFID tags are used for tracking and identifying what they are embedded into, such as a person, object or animal. Since the RFID tags are small, they can be attached to almost anything including money and clothing. Some RFID tags do not have batteries and they are known as “Passive Tags”. The energy needed by the passive tag to send data is gained from the RF signals that are transmitted by the RFID reader. The passive tags have a sending range of a few meters. Other RFID tags have batteries and they are known as “Active Tags”. The active tags can broadcast data at all times. They normally have a sending range of hundred meters. The RFID tags can keep data about persons, animals, or physical objects to which they are attached, such as personal information, location tracking history, ownership and date of manufacture. Because of its low power requirements and flexibility, the RFID technology is a considerable method to connect the unconnected physical objects to an IoE solution by supplying data by an RFID tag to an RFID reader [2]. The RFID technology is widely used in various areas such as transportation, access control, supply chain management, manufacturing, libraries, automobile security, healthcare, animal tracking, automatic payment, E-passports, etc. Therefore, RFID related business experiences many significant advantages [1]. However, every technology has its problems. Security and privacy of RFID technology are very questionable since RFID is a wireless technology and therefore, it is subject to various attacks such as replay attack, spoofing/cloning attack, and disclosing sensitive information of tags, and hence infringes RFID’s security and privacy. RFID systems need to be designed and implemented with adequate security and privacy protection in order to protect the data on the tag and the data transmitted between the tag and reader and to ensure that the data is accurate and safe from unauthorized access [3]. RFID authentication protocols are considered as a possible solution to secure RFID communications and address the security and privacy issues of RFID systems [4]. Many RFID authentication protocols are proposed to address the security and privacy threats. Some of these protocols are appropriate for only one specific solution, other protocols are found to be incorrect and afterward corrected, and finally some proposals are insignificant and are later disregarded [5]. The rest of this paper is organized as follows. In the next section, related works are reviewed. Section 3 proposes an improved three-pass mutual authentication protocol (ITPMAP) for low-cost RFID tags to address the identified security and privacy issues. Security evaluation of the proposed ITPMAP protocol is presented in Sect. 4. In Sect. 5, as proof of concept, the proposed ITPMAP protocol is used to develop three secure RFID systems. Section 6 shows how to implement the proposed secure RFID systems. Finally, Sect. 7 states the conclusion and gives the direction for future research.

## 2 Related Works

### 2.1 RFID Authentication Protocols

According to our previous survey on the existing RFID authentication protocols [6], several attacks found in recent important research papers [7–34]; namely: desynchronization attack, spoofing/cloning attack, replay attack, man-in-the-middle (MitM) attack, and privacy violation. In desynchronization attack, the shared secret key between a tag and a reader or server is made inconsistent by an adversary. So, the tag and reader cannot recognize each other in the future and the tag becomes disabled [6]. None of the authentication protocols in [7, 9–11, 13–15, 18–20, 22, 23, 25–30, 32, 33] can provide desynchronization attack resistance. In spoofing/cloning attack, an adversary creates a copy of a genuine tag either by using fake tag or using RFID emulation device [6]. All the protocols in [7, 10–12, 14–16, 18–30, 32–34] failed to prevent the spoofing/cloning attack. In replay attack, an attacker replays captured messages, which are exchanged between tag and reader, to communicate with an authorized reader or a genuine tag [6]. All the protocols in [7–12, 14, 16–34] failed to provide replay attack resistance. In MitM attack, an attacker manipulates exchanged messages between tag and reader, by deletion, insertion or modification without being detected by the system [6]. None of the authentication protocols in [7–11, 13, 19, 21–34] can provide MitM attack resistance. Moreover, if the sensitive data is transmitted without encryption, an attacker can simply violate the privacy of the tag's owner by eavesdropping the communications between the RFID readers and tags and in this case, the confidentiality of the RFID system is breached [6]. None of the authentication protocols in [7, 8, 11, 14, 15, 17, 19, 21–28, 30–34] can provide data confidentiality.

As a result, the recent authentication protocols [7–34] have missing features to achieve integrated security and privacy requirements for RFID systems; namely: data confidentiality, non-repudiation, data origin authentication, data integrity, desynchronization attack resistance, spoofing/cloning attack resistance, MitM attack resistance, and replay attack resistance. Therefore, a new solution is necessary to provide the highlighted RFID security and privacy requirements and address the RFID security and privacy. Our survey is summarized and tabulated in the first twenty-eight entries in Table 2. Fix and build from our previous work, this paper proposes a new protocol as a solution to achieve RFID security and privacy requirements based on the elliptic curve digital signature algorithm (ECDSA) and password-based encryption (PBE) scheme.

### 2.2 Digital Signature Algorithms

The ordinary handwriting signature on a document is used to confirm that the signer is liable for the content of the document. The handwriting signature is actually a portion of the document and while forgery is surely feasible, it is hard to do so persuasively. Thus, an approach of signing documents digitally is required to be practically equivalent to the handwriting signatures. Algorithms, which offer this functionality, are named Digital Signature Algorithms (DSAs). DSAs have two components, the private signing algorithm which allows a signer to securely sign a digital document, and the public verification algorithm which allows anybody to verify that the digital signature is reliable. The DSA was suggested by the U.S. National Institute of Standards and Technology (NIST) in

August 1991. In addition, it is set in Federal Information Processing Standard (FIPS) named Digital Signature Standard (DSS) [35].

The ECDSA is elliptic curve counterpart of the DSA. The ECDSA was suggested by Scott Vanstone in 1992 in reply to the NIST demand for public observations on the proposal for DSAs [35]. It was approved as an ISO standard in 1998, and it was approved as an American National Standards Institute (ANSI) standard in 1999. In addition, it was approved as an Institute of Electrical and Electronics Engineers (IEEE) standard and an NIST standard in 2000 [36]. The ECDSA is recommended to be employed in the digital signing of data since the strength-per-key-bit is considerably larger in the elliptic curve system than in the traditional systems. Thus, minor parameters can be used in the ECDSA than in other DSAs but with equal security levels. The benefits that can be obtained from minor parameters include the speed (quicker computations), and minor keys and digital signatures [35] which lead to quicker processing time, and a decrease in the storage space, processing power, and bandwidth. Therefore, the ECDSA is ideal for resource-limited devices like low-cost RFID tags [36].

The ECDSA comprises three phases; namely: the key generation phase, the signing phase, and the verification phase. In the key generation phase, a key pair is generated for each entity. The key pair comprises a signing private key and related verification public key. Each entity preserves the secrecy of its own private key which is used in the process of signing the information. In addition, an entity may create duplications of its own public key accessible by other entities which utilize it to verify the digital signature. The signing phase is performed by a signer (entity) using his own private key. The verification phase is performed by a verifier (entity) using the signer's public key. Both of the signer and verifier may be computers which can carry out the essential operations (i.e., generating key pairs, signing and verification operations). The signing phase is carried out to get a digital signature, including two essential steps. First, a signer computes a hash value "h" of data "d" using one-way hashing function Hash(d). Second, the signer generates a digital signature by using the signing private key to encrypt the hash value "h" as shown in Fig. 1. The verification phase is carried out to verify the digital signature, including three essential steps. First, a verifier calculates a hash value "h1" of data "d" using Hash(d). Second, the verifier computes another hash value "h2" by decrypting the received digital signature using the signer's public key as shown in Fig. 1. Finally, the verifier confirms the validity of received signature by checking whether "h1" equals "h2". If they are equal, the received data is authenticated and the integrity of the data is preserved [35, 36].

### 2.3 Password-Based Encryption Scheme

A basic issue in cryptography is how to communicate safely through an insecure channel such as RF channel, which may be managed by attackers. In this scenario, it is ordinary for two parties to encrypt and authenticate their transmitted information for protecting the confidentiality of the information [37].

To guard the confidentiality of the transmitted sensitive information in the insecure channels, it requires to be encrypted. Therefore, the sensitive information of an RFID tag will be encrypted before storing it on the tag to provide privacy protection for that information. The users need to encrypt and decrypt their sensitive information using a simple-to-recall password (key), and at the same time be assured that the sensitive information is secure from the eavesdroppers. The compromise of sensitive information may be disastrous to the users. The PBE algorithms are used to address issues of the kind portrayed above. An PBE algorithm uses a password to produce a secret key. The password

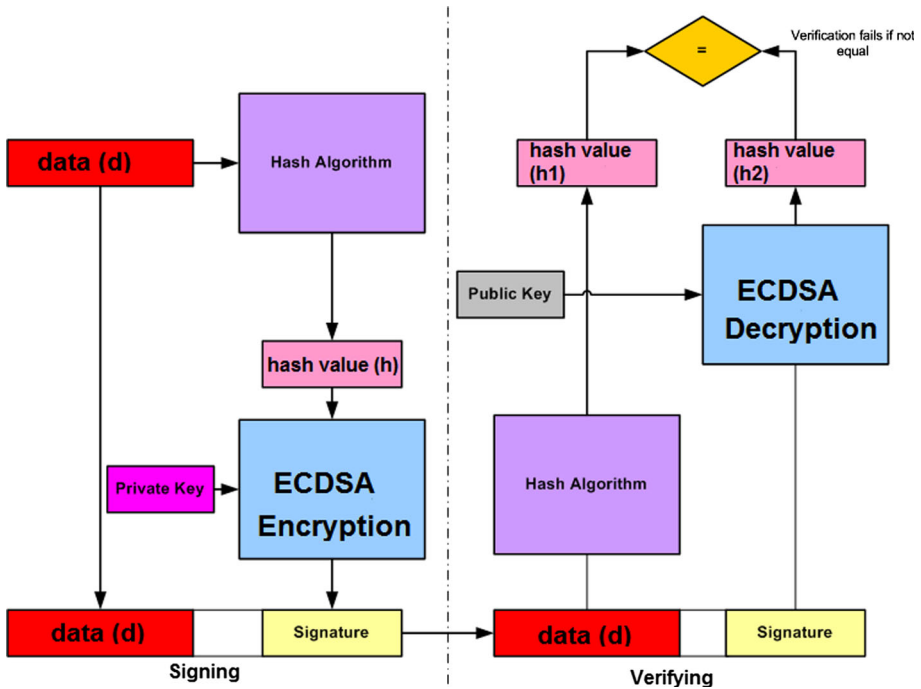


Fig. 1 Signing and verification phases of ECDSA

is supplied by an end user. The strong PBE algorithm blends a random number named “salt” with a password to produce the secret key. Without a salt, the attackers may carry out a brute force attack with relative ease [38].

### 3 The Proposed ITPMAP Protocol

The main idea of the proposed ITPMAP is to use the three-pass mutual authentication protocol [34] which is used in “Mifare Classic” tags; since these tags are widespread, their cost is cheap [39], and their authentication protocol provides desynchronization resistance. However, this protocol has serious problems and can’t provide most of security and privacy requirements [6]. Therefore, an improvement to this authentication protocol is proposed based on the ECDSA and PBE schemes to achieve the security and privacy requirements. The ECDSA is used by the back-end server to supply data origin authentication, data integrity, and non-repudiation [36]. In addition, the PBE scheme is used by the back-end server to encrypt the sensitive information before storing it on RFID tags to provide privacy protection (data confidentiality). Thereby, heavy-weighted cryptographic techniques are used on the back-end server side instead of lightweight cryptographic techniques to provide the security and privacy requirements. The recommended key length for the ECDSA is 256 bit [40, 41]. Therefore, the 256 bits is considered as the default key length value for the ECDSA in this paper. In addition, the 256-bit advanced encryption standard (AES) is used as an PBE scheme to ensure high-security level for accessing the sensitive information of

**Table 1** List of notations

Notations	Description
Ktr	Shared secret key between a genuine tag and authorized reader
Nt	32-bit challenge nonce generated by a tag
PRNG	32-bit pseudo-random number generator
Rr	Reader's response
$eKtr$	CRYPTO1 stream cipher based on Ktr
Nr	32-bit challenge nonce generated by a reader
UID	Unique identifier of a tag
	Concatenation of two inputs
TP	Tag's password inserted by a tag's owner
Ks	Secret key derived from TP
$AES\ PBEe$	$AES\ PBE$ encryption
$ECDSA_s$	$ECDSA$ signing
$ECDSA_v$	$ECDSA$ verifying
$AES\ PBE_d$	$AES\ PBE$ decryption
Kprv	$ECDSA$ private key
Kpub	$ECDSA$ public key
Ds	Sensitive information of tag's owner
Dus	Public (unsecure) information of tag's owner

tags since it is secure and unbroken [40, 41]. The ITPMAP protocol consists of three phases; namely: authentication phase, setting phase, and testing phase. The notations tabulated in Table 1 are used to simplify the description of the proposed ITPMAP protocol.

### 3.1 Authentication Phase

In this phase, the three-pass mutual authentication protocol [34] is performed. Both an RFID reader and an RFID tag authenticates each other by carrying out the three-pass mutual authentication protocol as follows.

1. An RFID reader sends a request R to be authenticated for reading or writing from/to a specific portion of the memory of an RFID tag.
2. The tag uses a secret key Ktr, shared between the genuine tag and authorized reader, to generate a challenge nonce Nt using the pseudo-random number generator (PRNG), and sends it to the reader, and this is the first pass.
3. After receiving the tag's challenge Nt, the reader uses the PRNG to generate a challenge nonce Nr and computes a response Rr to the received Nt using the shared secret key Ktr as follows.

$$Rr = eKtr(Nr||Nt||UID)$$

Next, the reader sends its nonce Nr and response Rr to the tag, and this is the second pass.

- The tag receives the  $N_r$  and  $R_r$  of the reader and verifies reader's response  $R_r$  by comparing it to its own calculation of the response. If the received reader's response is incorrect, the tag does not respond, and the communication is terminated; otherwise, the tag responds with an answer (response)  $R_t$  as follows.

$$R_t = eKtr(N_r || N_r)$$

After receiving the tag's response  $R_t$ , the reader verifies it by comparing it to its own calculation of the response. After transmission of the first random challenge  $N_t$ , the communication between the tag and reader is encrypted using CRYPTO1 stream cipher  $eKtr$  based on the secret key  $Ktr$ .

### 3.2 Setting Phase

In the case of writing data to an RFID tag, the setting phase is carried out after performing the authentication phase. In this phase, the tag's owner must insert a password called a "Tag's Password (TP)". The TP is used to authenticate the tag's holder since only the genuine tag's owner knows the correct TP. The TP is also used to derive a secret key  $Ks$  to encrypt the sensitive information  $Ds$  of tag's owner using the AES PBE algorithm before writing it to the tag as follows.

$$E(Ds) = AESPBEe(Ks, Ds)$$

If the information is not sensitive  $Dus$ , the back-end server writes it to the RFID tag without encryption. After that, the server signs the TP and information of tag's owner (i.e.,  $Ds$  and  $Dus$ ) using an ECDSA private key  $Kprv$  before writing it to the tag as follows.

$$S = ECDSAs(Kprv, TP, E(Ds), Dus)$$

Finally, the server writes the information and digital signature  $S$  to the tag. Figure 2 shows the authentication and setting phases.

### 3.3 Testing Phase

In the case of reading data from an RFID tag, the testing phase is carried out after performing the authentication phase. In this phase, the backend server reads the information (i.e.,  $E(Ds)$  and  $Dus$ ) and digital signature  $S$  from the tag. Next, the password test is performed. The purpose of this test is to authenticate the tag's holder by asking him/her to insert the TP which is only known by the genuine tag's owner. After inserting the TP, the back-end server verifies the digital signature using an ECDSA public key  $Kpub$  as follows.

$$Verify(S) = ECDSAv(Kpub, TP, E(Ds), Dus, S)$$

If the digital signature  $S$  is verified, the tag's holder is authenticated, otherwise, the tag's holder is not authenticated and the communication is terminated. After authenticating the tag's holder, the back-end server derives a secret key  $Ks$  from the inserted TP and uses it to decrypt the sensitive information ( $Ds$ ) using the AES PBE algorithm as follows.

$$Ds = AESPBEd(Ks, E(Ds))$$

Figure 3 shows the authentication and testing phases of the proposed ITPMAP protocol.

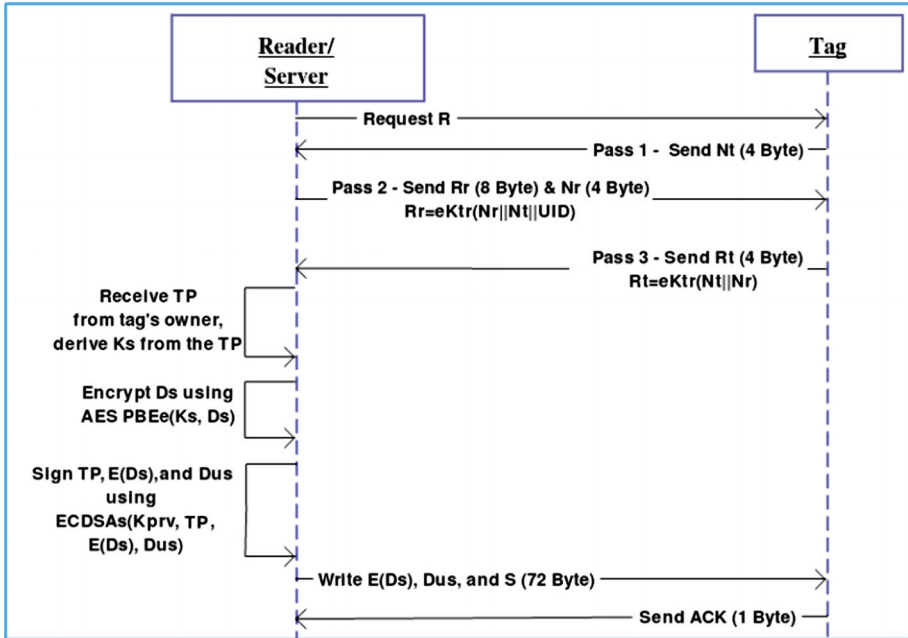


Fig. 2 The authentication and setting phases of ITPMAP

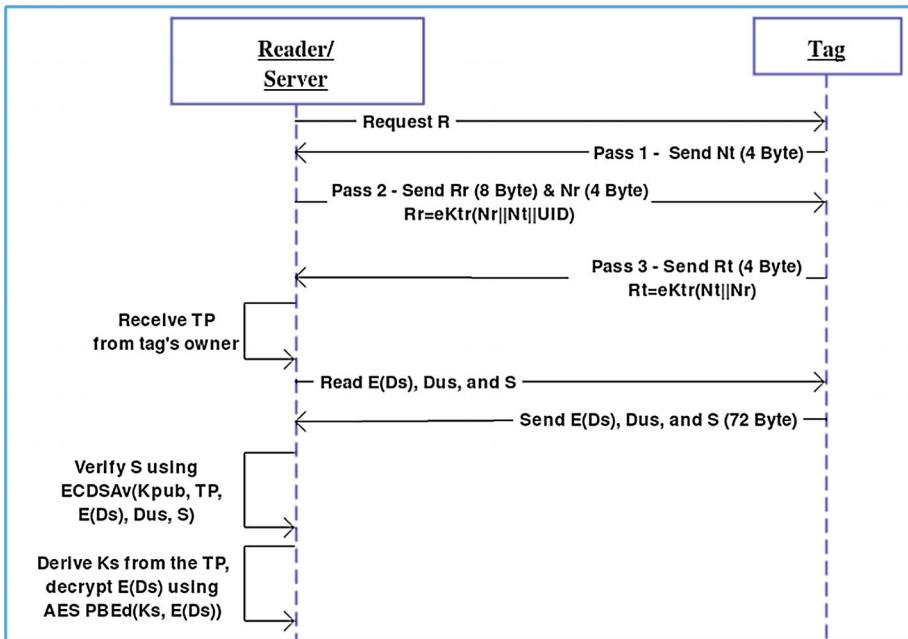


Fig. 3 The authentication and testing phases of ITPMAP



## 4 Security Evaluation of the Proposed ITPMAP Protocol

This section gives the security evaluation of the proposed ITPMAP protocol according to the identified security and privacy requirements in [6] as follows.

- **Desynchronization attack resistance:** the ITPMAP protocol does not need to synchronize the secret key  $K_{tr}$  shared between the genuine tag and authorized reader because the  $K_{tr}$  is fixed and write-protected. Thereby, the attackers cannot desynchronize the genuine tags and the authorized readers.
- **Spoofing/Cloning attack resistance:** an attacker may eavesdrop legitimate communications, but he/she cannot succeed in spoofing or cloning a genuine tag because of the employing of TPs. In the setting phase of the ITPMAP protocol, an TP is requested to be set by the tag's owner. The inserted TP and information related to tag's owner are signed using  $ECDSAs(K_{prv}, TP, E(Ds), Dus)$ . In the testing phase of the ITPMAP protocol, the tag's holder must insert an TP, then the inserted TP, and information of tag's owner, which is stored in the tag, are used to verify the digital signature stored in the tag using  $ECDSAv(K_{pub}, TP, E(Ds), Dus, S)$ . If the digital signature  $S$  is verified, the tag's holder is authenticated, since only the authorized tag's owner knows the correct TP. Getting the TP is impossible because it never appears in any interactions, and it is not stored in clear text neither in the genuine tag, nor in the authorized reader.
- **Replay attack resistance:** an attacker may eavesdrop on the communications between a genuine tag and an authorized reader and store the response messages sent by the genuine tag, and then retransmit the messages to the legitimate reader in order to impersonate the genuine tag. However, the attacker cannot pass the testing phase of the ITPMAP protocol, since only the tag's owner knows the correct TP which is used to verify the digital signature  $S$  using  $ECDSAv(K_{pub}, TP, E(Ds), Dus, S)$ . Another replay attack scenario is that an attacker may replay messages from an authorized reader. But, this scenario does not change secret key  $K_{tr}$  of the genuine tag because it is write-protected, and the attacker gains no sensitive information  $Ds$  of the tag because it is strongly encrypted using  $AES\ PBEe(Ks, Ds)$ . Thereby, the ITPMAP protocol is secure against the replay attack.
- **Resistance to MitM attack:** the ITPMAP protocol can resist the MitM attack because it provides strong integrity using the digital signature technique. Any modification of tag's information (i.e.,  $E(Ds)$  and  $Dus$ ) makes the digital signature  $S$  stored in the tag invalid, which makes the attacker hard to change  $E(Ds)$  or  $Dus$  without being noticed by the RFID system using  $ECDSAv(K_{pub}, TP, E(Ds), Dus, S)$ .
- **Tag information privacy:** in the ITPMAP protocol, the sensitive information  $Ds$  is encrypted by  $AES\ PBEe(Ks, Ds)$  using a secret key  $Ks$  derived from an TP, which is only known by the tag's owner. Thus,  $Ds$  is never transmitted in clear text during the protocol execution. Thereby, the ITPMAP protocol guarantees that the sensitive information  $Ds$  stored in the genuine tag can be only accessed in plaintext form by the authorized entities.
- **Data origin authentication:** the proposed ITPMAP protocol provides data origin authentication since it uses the digital signature technique. The data (i.e.,  $E(Ds)$  and  $Dus$ ) and TP is signed using the signing private key  $K_{prv}$  of a genuine signer by  $ECDSAs(K_{prv}, TP, E(Ds), Dus)$ . Thus, after verifying the digital signature  $S$  using  $ECDSAv(K_{pub}, TP, E(Ds), Dus, S)$ , the verified  $S$  implies that the data was created by the genuine signer.

- **Data integrity:** the proposed ITPMAP protocol can provide data integrity since it uses the digital signature technique. In the process of producing a digital signature  $S$  for the data  $E(Ds)$  and  $Dus$  and  $TP$  using  $ECDSAs(K_{prv}, TP, E(Ds), Dus)$ , the hash value of  $E(Ds)$ ,  $Dus$ , and  $TP$  is encrypted with the signing private key  $K_{prv}$ . It is impossible for an adversary to slightly modify  $E(Ds)$  or  $Dus$ , stored in a tag, in such a way that it is valid without the knowledge of  $K_{prv}$  which is encrypted by  $AES\ PBEe(Ks, K_{prv})$  only accessed by the genuine signer. Hence, the integrity of the tag's information is preserved.
- **Non-repudiation:** the proposed ITPMAP protocol can provide non-repudiation feature since it uses the digital signature technique. The private key  $K_{prv}$  and the public key  $K_{pub}$  are mathematically related. Since tag's information (i.e.,  $E(Ds)$  and  $Dus$ ) and  $TP$ , signed with  $K_{prv}$ , can only be verified with the corresponding  $K_{pub}$ , and only the signer can access  $K_{prv}$  since it is encrypted by  $AES\ PBEe(Ks, K_{prv})$  and can only be decrypted using the secret key  $Ks$  derived from the  $TP$ , which is only known by the genuine signer. Thereby, the signer cannot deny having signed  $E(Ds)$ ,  $Dus$  and  $TP$ .
- **Computational cost:** there are only two requirements that the proposed ITPMAP protocol needs for each tag. This includes the ability to compare two numbers, and to generate a random number using PRNG. Comparing two numbers are easy for passive RFID tags. In addition, the PRNG can be effectively implemented on low-cost tags. Thus, the computational cost of the ITPMAP protocol is low.
- **Time complexity:** the proposed ITPMAP protocol is run in a constant time since it requires the same amount of time regardless of the number of tags used in an RFID system, i.e., the time complexity of the proposed ITPMAP protocol is  $O(1)$ .
- **Space complexity:** the space complexity of the proposed ITPMAP protocol is  $O(1)$  since it is only fixed space needed to construct the proposed ITPMAP no matter how many tags that are used in an RFID system.

Table 2 demonstrates the security requirements provided by the ITPMAP protocol compared with other related.

## 5 The Design of Secure RFID Systems

In this section, the proposed ITPMAP protocol is adapted by three RFID systems to make them secure. These systems are Signing and Verification of Graduation Certificate System (SVGCS), Issuing and Verification of E-Ticketing System (IVETS), and Charging and Discharging of Prepaid Card System (CDPCS). Each system has its own database and each user in each of these systems has a record in the system's database that includes information related to the user, a signing private key of the user, and related public key of the user. Before saving the signing private key of each user in the system's database, it should be encrypted by inserting a password by the user then the back-end server derives an encryption key and uses the AES PBE scheme to encrypt the signing private key of the user to keep it securely in the database. The Unified Modeling Language (UML) is used to demonstrate the architectural design of the proposed systems.

### 5.1 The Design of SVGCS

The SVGCS allows the graduate students to issue and verify graduation certificates by using the RFID technology securely. The SVGCS consists of three modules; namely:

**Table 2** Comparison between ITPMAP and related protocols

Implemented requirement (✓) Not supported (×) RFID authentication protocols	Requirements										
	Desynchronization resistance	Spoofing/cloning resistance	Replay attack resistance	MitM attack resistance	Data confidentiality	Non-repudiation	Data origin authentication	Data integrity	Computational cost		
Henrici-Muller [7]	×	×	×	×	×	×	×	×	×	Hash	
OHLCAP [8]	✓	✓	×	×	×	×	×	×	×	Hash	
Osaka et al. [9]	×	✓	×	×	✓	×	×	×	×	Hash	
Ha et al. [10]	×	×	×	×	✓	×	×	×	×	Hash	
Song-Mitchell [11]	×	×	×	×	×	×	×	×	×	Hash	
PAP [12]	✓	×	×	✓	✓	×	×	×	×	Hash	
AFMAP [13]	×	✓	✓	✓	✓	×	×	×	×	Hash	
Cho et al. [14]	×	×	×	×	×	×	×	×	×	Hash	
Srivastava et al. [15]	×	×	✓	✓	×	×	×	×	×	Hash	
Chien-Huang [16]	✓	×	×	✓	×	×	×	×	×	PRNG	
NXP [34]	✓	×	×	×	×	×	×	×	×	PRNG	
Kim et al. [17]	✓	✓	×	✓	×	×	×	×	×	PRNG	
Sun-Ting [18]	×	×	×	✓	×	×	×	×	×	PRNG	
Niu et al. [19]	×	×	×	×	×	×	×	×	×	PRNG	
Burmester-Medeiros [20]	×	×	×	✓	×	×	×	×	×	PRNG	
Qingling et al. [21]	✓	×	×	×	×	×	×	×	×	PRNG	
Deng et al. [22]	×	×	×	×	×	×	×	×	×	PRNG	
Deng et al. [23]	×	×	×	×	×	×	×	×	×	PRNG	
Zhou [24]	✓	×	×	×	×	×	×	×	×	PRNG	
LMAP [25]	×	×	×	×	×	×	×	×	×	Bitwise operations	
M2AP [26]	×	×	×	×	×	×	×	×	×	Bitwise operations	

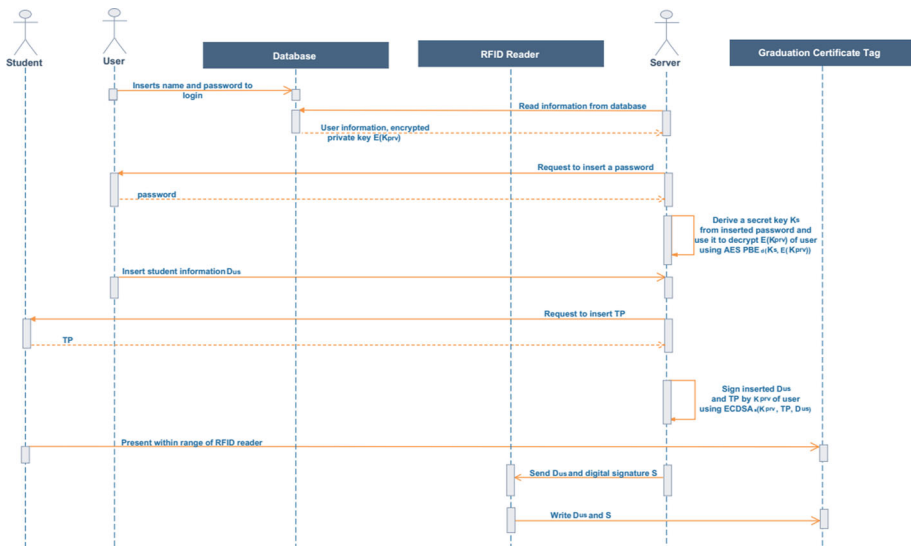
**Table 2** continued

RFID authentication protocols	Requirements									
	Desynchronization resistance	Spoofing/cloning resistance	Replay attack resistance	MitM attack resistance	Data confidentiality	Non-repudiation	Data origin authentication	Data integrity	Computational cost	
SASI [27]	×	×	×	×	×	×	×	×	Bitwise operations	
LMAP++ [28]	×	×	×	×	×	×	×	×	Bitwise operations	
Gossamer et al. [29]	×	×	×	×	✓	×	×	×	Bitwise operations	
Lee et al. [30]	×	×	×	×	×	×	×	×	Bitwise operations	
SULMA [31]	✓	✓	×	×	×	×	×	×	Bitwise operations	
DIDRFID-SIDRFID [32]	×	×	×	×	×	×	×	×	Bitwise operations	
RAPP [33]	×	×	×	×	×	×	×	×	Bitwise operations	
Proposed ITPMAP	✓	✓	✓	✓	✓	✓	✓	✓	PRNG	

Student Information Entry Module, Student Information Updating Module, and Graduation Certificate Verification Module.

### 5.1.1 Student Information Entry Module

This module is used by the system's users to insert information of a graduate student. This information includes student's full name, graduation year, gender, nationality, university name, college name, department name, mobile number, email address, and degrees of the student for each subject, in addition to the attempt in which the student passed those subjects (first or second attempt). In this paper, none of this information is considered as sensitive information. As shown in the sequence diagram in Fig. 4, to insert the student information, a user must login to the SVGCS system by inserting his/her name and password to authenticate himself/herself to the system. If the name and hash of password are found in the database, the user is authenticated and the back-end server uses the inserted name and the hash of inserted password as an index to read the information of the user and his/her encrypted private key from the system's database. Then, the server requests to insert a password from the user to decrypt the encrypted private key. After that, the logged-in user can issue graduation certificates of graduate students. To do so, the authenticated user inserts the information of graduation certificate. After inserting the information of the graduate student, the back-end server requests an TP from the graduate student. Then, the back-end server uses the private key of the logged-in user to sign the information and the inserted TP of the graduate student. Next, the graduate student presents his/her empty tag within the reading range of RFID reader. After that, the server sends the student's information (i.e., the graduation certificate) and the digital signature to the reader to write them to the presented tag of the graduate student.



**Fig. 4** Sequence diagram for the student information entry module

### 5.1.2 Student Information Updating Module

This module is used to review the information of graduate student after reading it from the graduation certificate tag using an RFID reader. If the student’s information needs a modification process due to incorrect entry, or it needs to be updated, it is modified by a system’s user. To review a graduation certificate, a user must login to the SVGCS system by inserting his/her name and password to authenticate himself/herself to the system. If the name and hash of password are found in the database, the user is authenticated and the back-end server uses the inserted name and the hash of inserted password as an index to read the information of the user, his/her encrypted private key, and related ECDSA public key from the system’s database. Then, the server requests to insert a password from the user to decrypt his/her own private key. After that, the logged-in user can update graduation certificates of graduate students. To do so, a graduate student presents his/her graduation certificate tag within the reading range of RFID reader. The back-end server uses the reader to read the graduation certificate information and the digital signature from the presented tag. Then, the server requests an TP from the graduate student to use it with the student’s information to verify the digital signature of the presented tag using the ECDSA public key. If it is verified, the logged-in user reviews and updates the graduation certificate information. After that, the server signs the updated information using the ECDSA signing private key of the user, then the updated information and digital signature are saved into the graduation certificate tag. The details of the module are shown in the sequence diagram in Fig. 5.

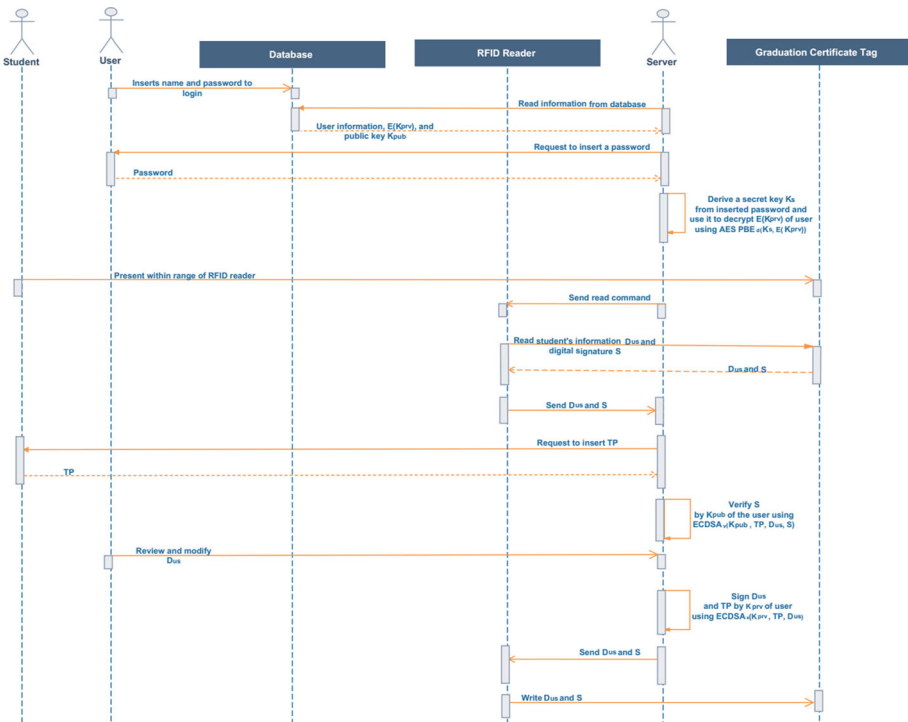


Fig. 5 Sequence diagram for the student information updating module

### 5.1.3 Graduation Certificate Verification Module

This module is used by the logged-in system's users to verify the information of the graduation certificates stored in the graduation certificate tags of graduate students. The graduation certificate verification module is represented by the sequence diagram shown in Fig. 6.

## 5.2 The Design of IVETS

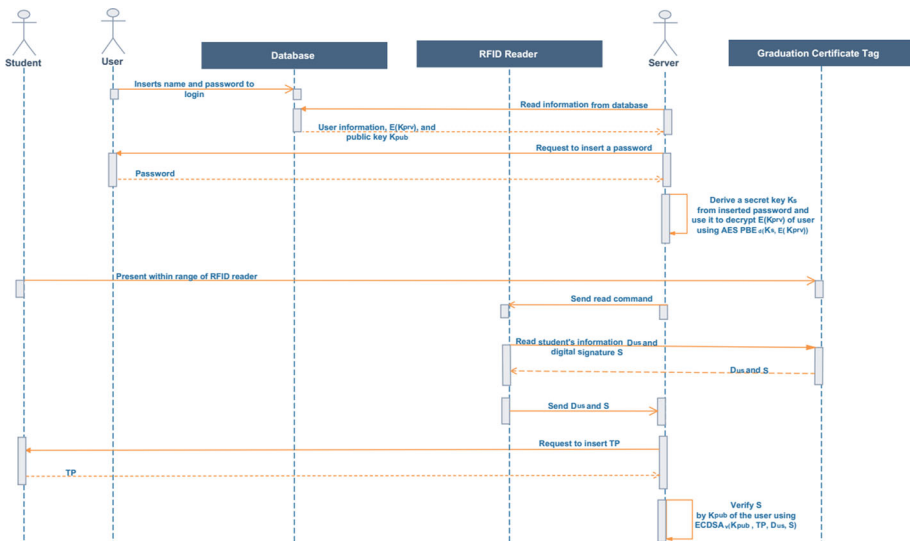
The IVETS allows the customers to issue and verify e-tickets for events (concerts, musicals, operas, sports games, etc.) by using the RFID technology safely. The IVETS consists of three modules; namely: E-Ticket Tags Creating Module, E-Tickets Issuing Module, and E-Tickets Verification Module.

### 5.2.1 E-Ticket Tags Creating Module

In this module, the system's users are responsible for creating new e-ticket tags for new customers. All that is asked from the user to create a new e-ticket tag is to login using his/her name and password and insert the required information related to the new customer. The information includes the full name of the customer, age, gender, mobile number, and email address. In this paper, none of this information is considered as sensitive information. The sequence diagram, shown in Fig. 7, is constructed to show all the messages needed for this module.

### 5.2.2 E-Tickets Issuing Module

In this module, the users are responsible for issuing new e-tickets for the customers. A customer can simply purchase an e-ticket by going to certain places which sell the tickets.



**Fig. 6** Sequence diagram for the graduation certificate verification module

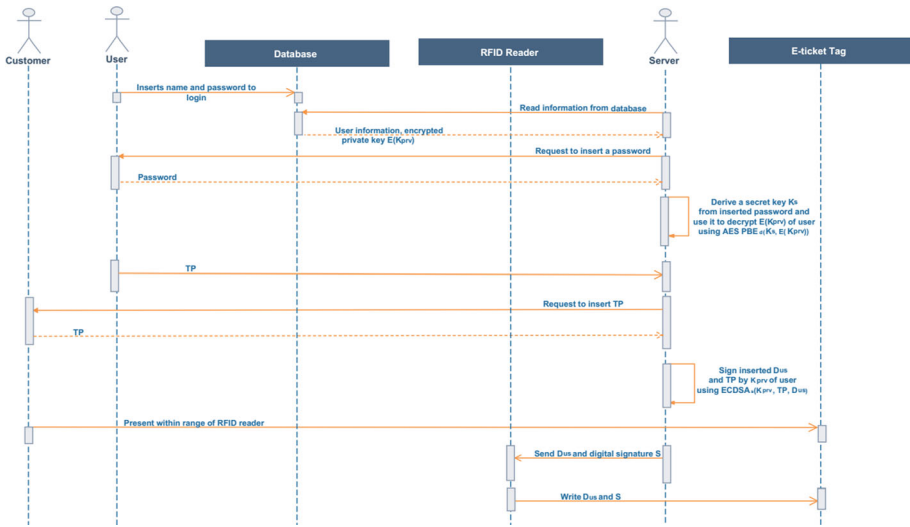


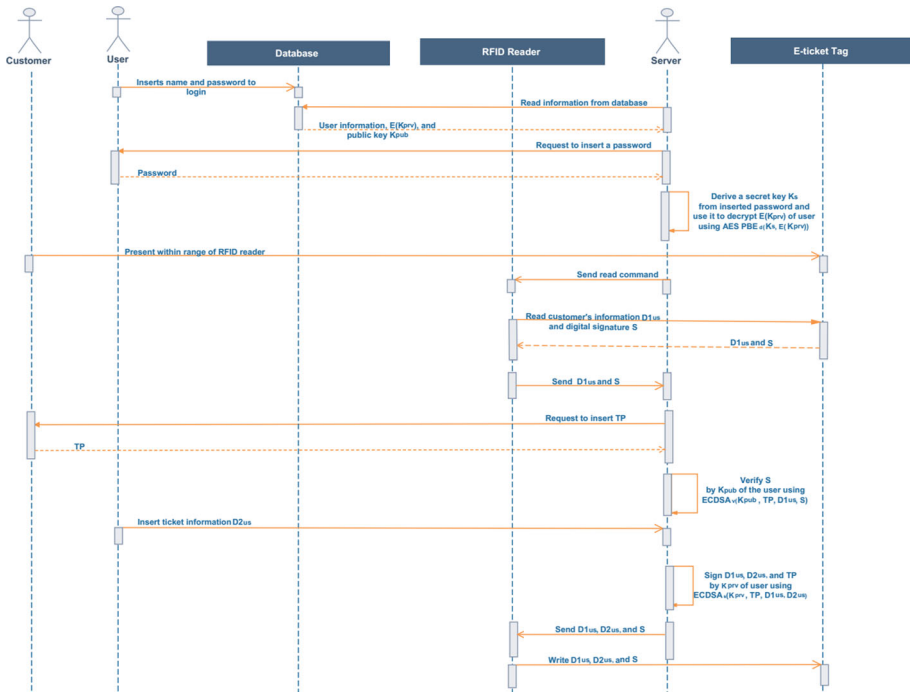
Fig. 7 Sequence diagram for creating a new e-ticket tag

The customer selects an event, buy and pay for the event, and presents his/her e-ticket tag within the reading range of the RFID reader. The back-end server reads the customer’s information and the digital signature from the presented e-ticket tag of the customer using the RFID reader. After that, the back-end server demands to insert an TP by the customer to authenticate him/her. After inserting the TP, the server determines whether the presented e-ticket tag is valid (i.e. the digital signature stored in the presented tag is verified) using the ECDSA public key of the logged-in user. If the e-ticket tag is verified, the server signs the information of the selected e-ticket and the TP of the customer using the signing private key of the logged-in user. The ticket information includes the event type, event name, event location, event date, start time of the event, end time of the event, ticket price, and validity of the ticket. After signing the information, the back-end server sends the ticket information and the digital signature to the reader to save them into the presented e-ticket tag of the customer. The details of E-Tickets Issuing Module is represented by the sequence diagram shown in Fig. 8.

### 5.2.3 E-Tickets Verification Module

In this module, the users are responsible for verifying the issued e-tickets stored in the e-ticket tags of the customers. To do so, a user uses his/her name and password to login to the IVETS system. Then, the customer can present his/her e-ticket tag within the reading range of the RFID reader at the event place. The back-end server uses the reader to read the ticket information and digital signature from the presented e-ticket tag of the customer. After that, the back-end server demands to insert an TP by the customer to authenticate him/her. After inserting the TP, the server checks the validity of e-ticket by verifying the digital signature of presented e-ticket tag using the ECDSA public key of the logged-in user. If the digital signature is verified, the e-ticket is valid and the customer can pass to the event. The details of E-Tickets Verification Module is shown in the sequence diagram in Fig. 9.





**Fig. 8** Sequence diagram for the e-tickets issuing module

### 5.3 The Design of CDPCS

The CDPCS allows the customers to charge and discharge the balance of their prepaid cards by using the RFID technology safely. The CDPCS consists of three modules; namely: Prepaid Cards Creating Module, Prepaid Cards Charging Module, and Prepaid Cards Discharging Module.

#### 5.3.1 Prepaid Cards Creating Module

In this module, the system's users are responsible for creating new prepaid cards for new customers. All that is asked from the users to create a new customer(s) is to login and insert the required information related to the customers. The information includes the full name of customer, age, gender, mobile number, email address, and initial balance. In this paper, none of this information is considered as sensitive information. The sequence diagram, shown in Fig. 10, is constructed to show all the messages needed for this module.

#### 5.3.2 Prepaid Cards Charging Module

In this module, the users are responsible for charging balance of the prepaid cards of customers. To do so, the user should login to the CDPCS system. Then, a customer presents his/her prepaid card (tag) within the reading range of the RFID reader. After presenting the prepaid card, the back-end server sends a "read" command to the RFID reader to read the customer's information and the digital signature from the presented card.

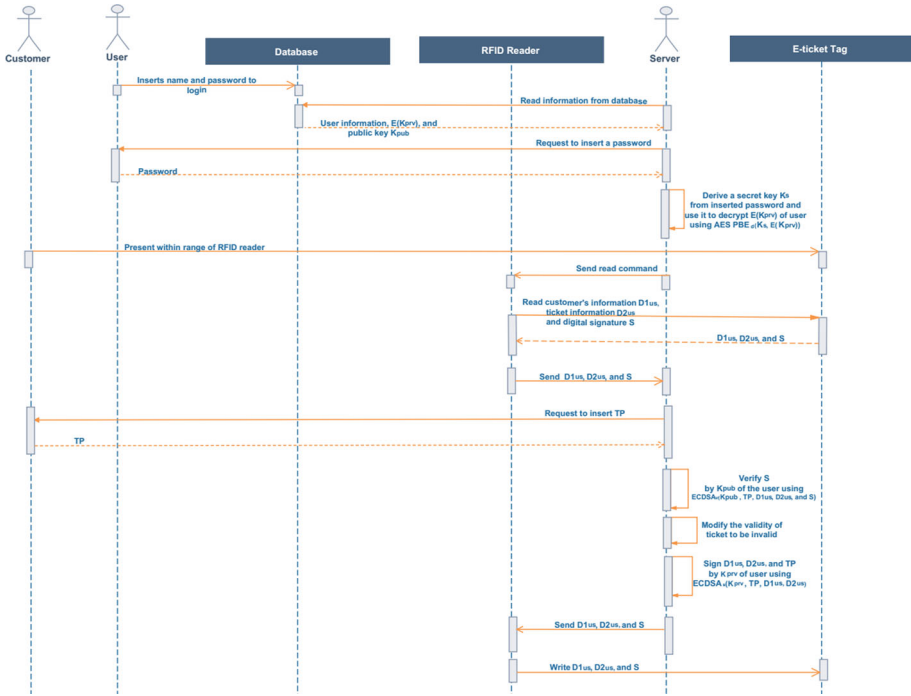


Fig. 9 Sequence diagram for the e-tickets verification module

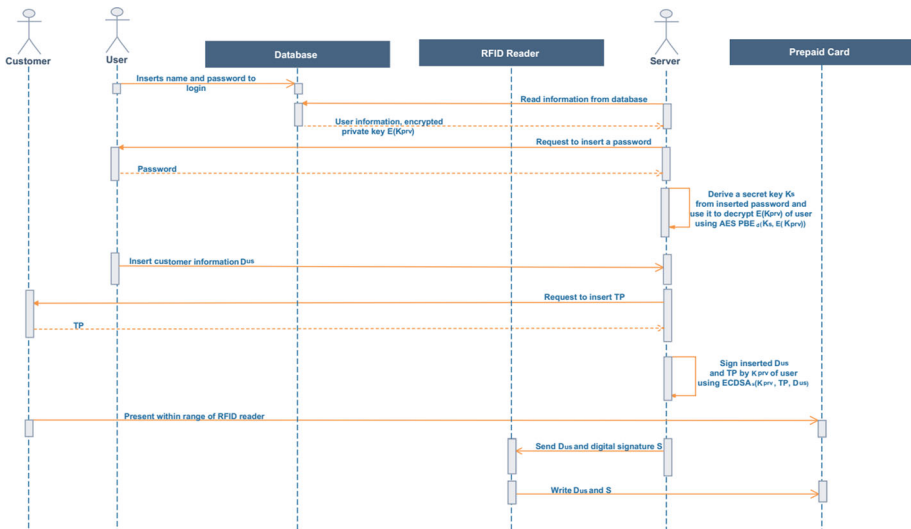


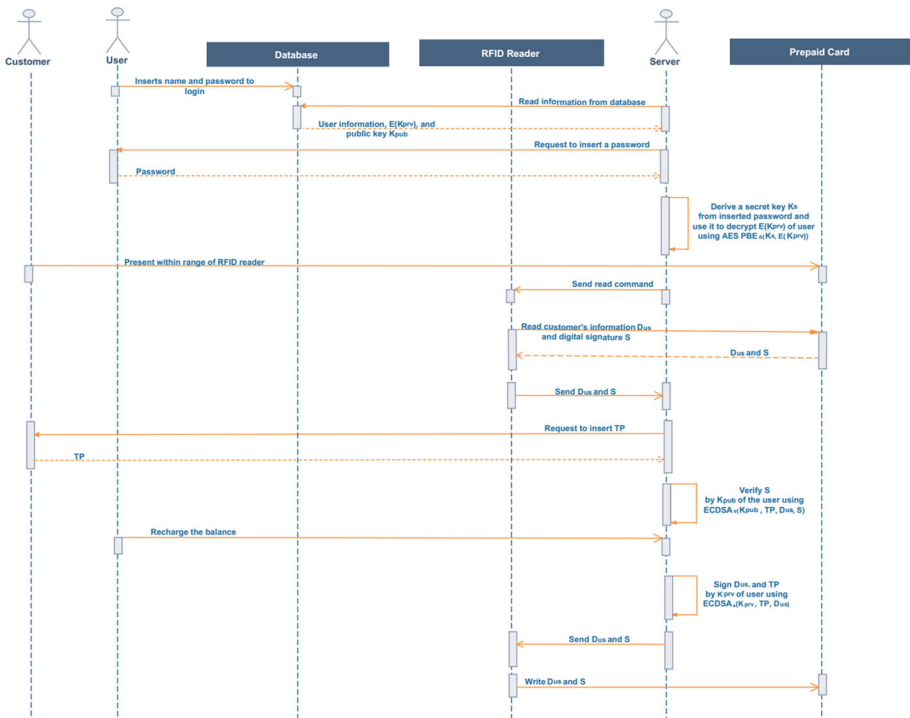
Fig. 10 Sequence diagram for creating new prepaid card

After that, the back-end server demands to insert an TP by the card’s owner to authenticate him/her. After inserting the TP, the server determines whether the presented card is valid (i.e. the digital signature read from the presented card is verified) using the ECDSA public

key of the user. If the prepaid card is verified, the back-end server recharges the balance of the presented prepaid card. Then, the server signs the TP of customer and the information of the prepaid card, after recharging its balance, using the signing private key of the logged-in user. After signing the information and TP, the back-end server sends the information and digital signature to the reader to write them to the presented prepaid card of the customer. The prepaid card charging module is represented by the sequence diagram shown in Fig. 11.

### 5.3.3 Prepaid Cards Discharging Module

In this module, the users are responsible for discharging the balance of the prepaid cards of the customers. First, a user uses his/her name and password to login to the system. Then, a customer presents his/her prepaid card (tag) within the reading range of the RFID reader. After presenting the prepaid card, the back-end server sends a “read” command to the reader to read the customer’s information and the digital signature from the presented card. After that, the back-end server demands to insert an TP by the card’s owner to authenticate him/her. After inserting the TP, the server checks the validity of the presented card by verifying the digital signature using the ECDSA public key of the logged-in user. If the prepaid card is verified, its balance is discharged by the back-end server. Then, the server signs the customer’s TP and the information of the prepaid card, after discharging its balance, using the signing private key of the logged-in user. After signing the TP and information, the back-end server sends the information and the digital signature to the



**Fig. 11** Sequence diagram for the prepaid card charging module

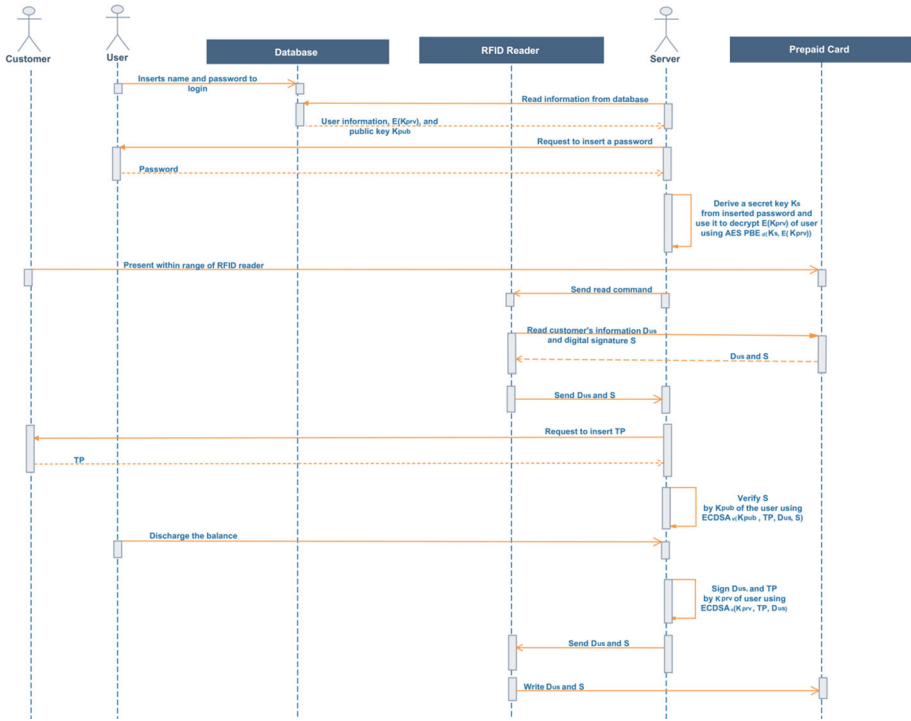


Fig. 12 Sequence diagram for the prepaid card discharging module

reader to write them to the presented prepaid card of the customer. The details of the prepaid card discharging module is shown in the sequence diagram in Fig. 12.

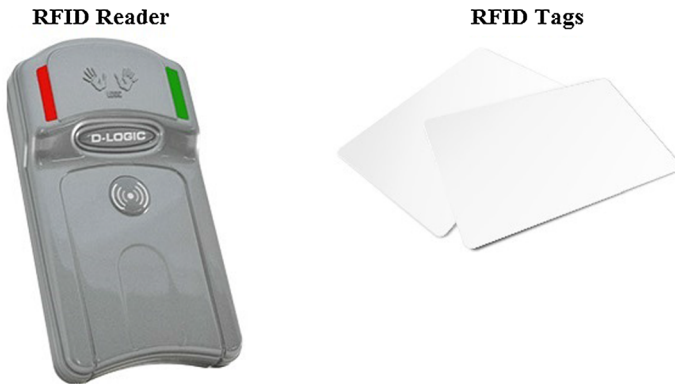
### 6 The Implementation of Secure RFID Systems

The implementation of the proposed secure RFID systems (i.e., the SVGCS, IVETS, and CDPCS) consists of the following: RFID apparatuses, software requirements that need to be set in the back-end server, and graphical user interfaces (GUIs) covering the needed activities of the proposed systems.

RFID apparatuses required by the proposed systems include two components; namely: tags, and readers. The “Mifare Classic” RFID tags are employed to save the information of individuals concerned with each proposed system. Every tag contains one 32-bit unchangeable UID, representing a single individual. On the other hand, “Mifare Classic” RFID readers are employed by the proposed systems to read and write from/to the tags. The readers have built-in antennas for tags access, and they can detect tags within a reading range of 2–8 cm [42]. The used RFID apparatuses are shown in Fig. 13.

Software requirements needed by the proposed systems consist of two items as follows.

1. Java Runtime Environment (JRE) File—the Java programming language is employed for the implementation of proposed systems according to the following causes:



**Fig. 13** Mifare classic RFID apparatuses

- i. Java programming language aids the communications between the nonhomogeneous hardware and software platforms (i.e., readers, databases, and operating systems) [43].
  - ii. Java is consistent with various well-known operating systems (i.e., Windows, Linux and Macintosh) so that there is no restriction for the back-end servers employed in the implementation of proposed systems, i.e., Java provides cross-platform functionality [43].
2. Java DB (also known as Apache Derby) Server—the Java DB server is employed for the database according to the following causes:
- i. Java DB is implemented in Java, as a result, it is a platform-independent database [44].
  - ii. Java DB can be given directly with its application since there are no further actions or special installations to operate the database are necessary [44].

It should be mentioned that the proposed three secure systems (i.e., the SVGCS, IVETS, and CDPCS) are implemented in one RFID tag per individual. The byte map of tag is shown in Fig. 14.

Based on the proposed design in Sect. 5, the implementation details of the SVGCS, IVETS, and CDPCS are presented in this section.

## 6.1 The Implementation of SVGCS

This section demonstrates the concrete implementation of the SVGCS. For the SVGCS implementation, constructed Java classes and GUIs are classified into four modules; namely: User Login Module, Graduation Certificate Creating Module, Student Information Updating Module, and Graduation Certificate Verifying Module. Java codes for the SVGCS System are given in Appendix I. SVGCS modules are discussed in detail below.

### 6.1.1 User Login Module

This module is used by the users to login to the SVGCS system. First, the back-end server demands from a user to present his/her name and password to authenticate himself/herself

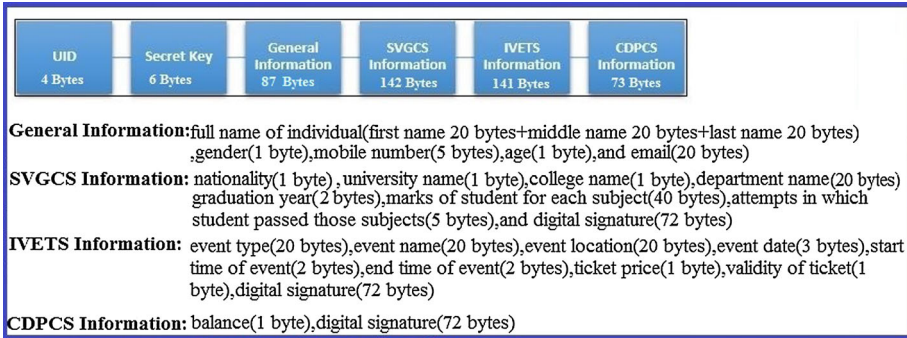


Fig. 14 Byte map of an RFID tag

to the system. If the name and hash of password are found in the system’s database, the user is authenticated and the back-end server uses the inserted name and the hash of inserted password as an index to read the information of the user and his/her public key and encrypted private key from the system’s database. Then, the server requests to insert a password from the user to decrypt the encrypted private key. After that, the following GUI is displayed (Fig. 15).

The displayed GUI (Fig. 15) is used by the logged-in user to select one of three modules: “Graduation Certificate Creating”, “Student Information Updating”, and “Graduation Certificate Verifying”.

### 6.1.2 Graduation Certificate Creating Module

In this module, the logged-in users are responsible for inserting the information of graduate students to create their graduation certificates. First, a graduate student presents his/her empty graduation certificate tag within the reading range of the RFID reader. Then, an GUI

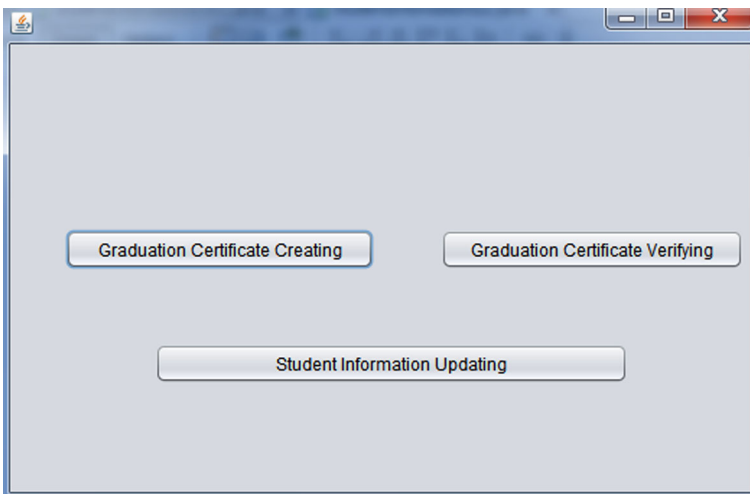


Fig. 15 GUI for selecting a module by the logged-in user of SVGCS

is displayed (Fig. 16) to insert the required information related to the graduate student by the logged-in user. After inserting the information, the user clicks “Sign and Save” button. After clicking the button, the back-end server requests to insert an TP by the graduate student. Then, the back-end server signs the information and TP of the graduate student using the signing private key of the logged-in user. Then, the server writes the digital signature and student’s information to the presented graduation certificate tag of the graduate student using the RFID reader.

### 6.1.3 Student Information Updating Module

In this module, the logged-in user reviews the student’s information after reading it from the graduation certificate tag of the graduate student to check its correctness. First, the graduated student presents his/her graduation certificate tag within the reading range of the RFID reader. Then, the backend server reads student’s information from the presented tag using the RFID reader, and the server displays the information in an GUI (Fig. 17) to the logged-in user. The user checks the displayed information. If there is an incorrect entry or the information needs to be updated, the user modifies the student’s information. After modifying the information, the user clicks “Sign and Save” button to sign the modified information using the signing private key of the logged-in user. Then, the server saves the signed information and digital signature in the presented graduation certificate tag.

### 6.1.4 Graduation Certificate Verifying Module

This module is responsible for the verification of the graduation certificates of the graduate students. First, the graduated student presents his/her graduation certificate tag within the reading range of the RFID reader. Then, the back-end server reads the student’s information and the digital signature from the presented tag using the reader. Then, the server requests to insert an TP by the graduate student. The server verifies the graduation certificate by verifying the digital signature using the ECDSA public key of the logged-in

First Name	Mustafa	<b>First Year</b>	Mathematics I	76	<input type="checkbox"/> 2nd Attempt	<b>Second Year</b>	Mathematics II	81	<input type="checkbox"/> 2nd Attempt
Middle Name	Hashim		Electrotechnique & Energy Conversion	82	<input type="checkbox"/> 2nd Attempt		Electronics & Communications Lab.I	87	<input type="checkbox"/> 2nd Attempt
Last Name	Abdulkareem		Electrical Circuits I	58	<input type="checkbox"/> 2nd Attempt		Computer Architecture I	88	<input type="checkbox"/> 2nd Attempt
Graduation Year	2014		Electronics I	79	<input type="checkbox"/> 2nd Attempt		Electronics II	62	<input type="checkbox"/> 2nd Attempt
Gender	Male		Fundamentals of Digital Systems	82	<input type="checkbox"/> 2nd Attempt		Microprocessor & Microcomputer I	75	<input type="checkbox"/> 2nd Attempt
Nationality	Iraq		Computer Science & Prog. Methodology	99	<input type="checkbox"/> 2nd Attempt		Data Structures	93	<input type="checkbox"/> 2nd Attempt
University Name	University of Baghdad		Electricity Lab.	64	<input type="checkbox"/> 2nd Attempt		Digital Systems Design	79	<input type="checkbox"/> 2nd Attempt
College Name	College of Engineering		Engineering Workshop	83	<input type="checkbox"/> 2nd Attempt		Electrical Circuits II	64	<input type="checkbox"/> 2nd Attempt
Department Name	Computer		Human Rights	98	<input type="checkbox"/> 2nd Attempt		Arabic Language	98	<input type="checkbox"/> 2nd Attempt
Email	cannocan10@gmail.com	<b>Third Year</b>	Communications	83	<input type="checkbox"/> 2nd Attempt	<b>Fourth Year</b>	Computer Control	83	<input type="checkbox"/> 2nd Attempt
Mobile number	7701234568		Electronics III	94	<input type="checkbox"/> 2nd Attempt		Control & Computer Lab.	98	<input type="checkbox"/> 2nd Attempt
Card ID	0xf789CC69		Engineering & Numerical Analysis	95	<input type="checkbox"/> 2nd Attempt		Internet Technology	91	<input type="checkbox"/> 2nd Attempt
			Computer Architecture II	92	<input type="checkbox"/> 2nd Attempt		Software Engineering	86	<input type="checkbox"/> 2nd Attempt
			Microprocessor & Microcomputer II	99	<input type="checkbox"/> 2nd Attempt		Cryptography & Computer Security	96	<input type="checkbox"/> 2nd Attempt
			Operating Systems	83	<input type="checkbox"/> 2nd Attempt		Computer Networks	84	<input type="checkbox"/> 2nd Attempt
			Electronics & Communications Lab.II	92	<input type="checkbox"/> 2nd Attempt		Interfacing I/O Devices	88	<input type="checkbox"/> 2nd Attempt
			Database Systems	96	<input type="checkbox"/> 2nd Attempt		Digital Signal Processing	97	<input type="checkbox"/> 2nd Attempt
							Engineering Project	94	<input type="checkbox"/> 2nd Attempt

Fig. 16 GUI for student information entry

Fig. 17 GUI for reviewing student information

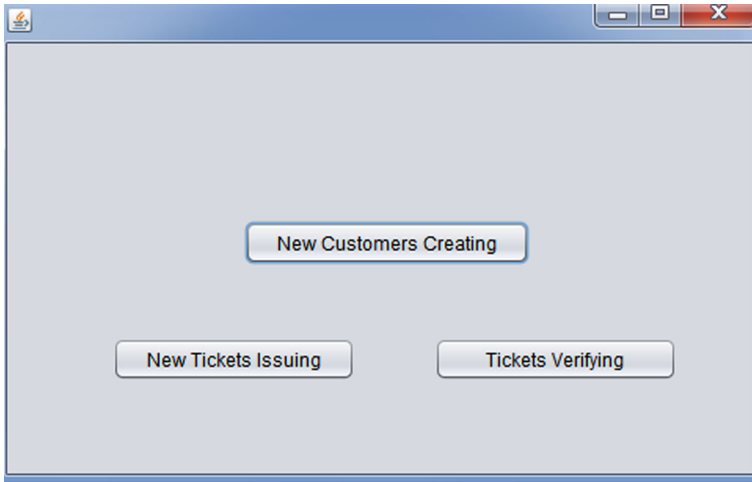
Fig. 18 GUI for displaying graduation certificate

user. If the signature is verified, the graduation certificate is valid, and the information of graduation certificate is displayed in a GUI shown in Fig. 18.

### 6.2 The Implementation of IVETS

This section demonstrates the concrete implementation of the IVETS. For the IVETS implementation, constructed Java classes and GUIs are classified into four modules; namely: User Login Module, New Customer Creating Module, New Tickets Issuing Module, and Tickets Verifying Module. Java codes for the IVETS System are given in Appendix II. IVETS modules are discussed in detail below.





**Fig. 19** GUI for selecting a module by the logged-in user of IVETS

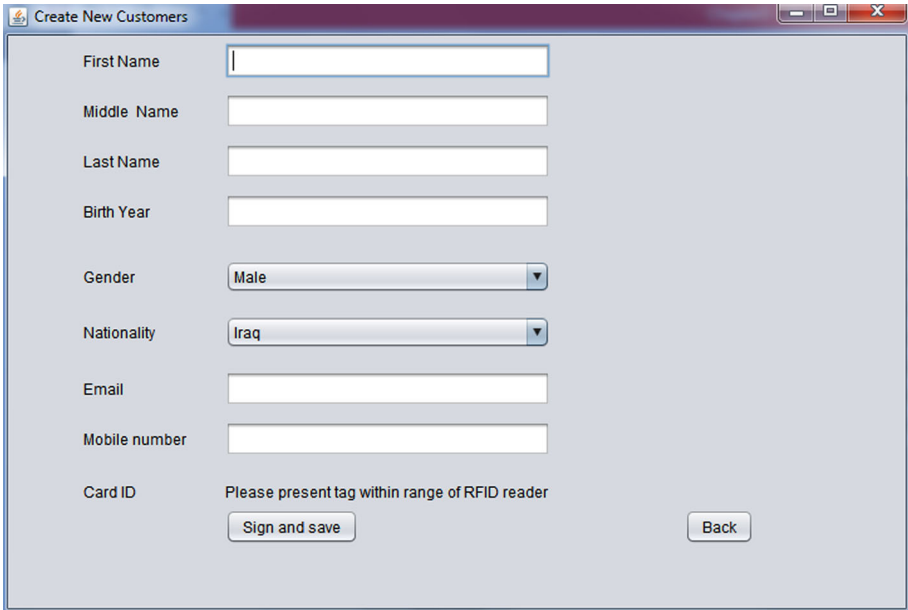
### 6.2.1 User Login Module

This module is used by the users to login to the IVETS system. First, the back-end server demands from a user to present his/her name and password to authenticate himself/herself to the system. If the name and hash of password are found in the system's database, the user is authenticated and the back-end server uses the inserted name and the hash of inserted password as an index to read the information of the user and his/her encrypted private key from the system's database. Then, the server requests to insert a password from the user to decrypt the encrypted private key. After that, the following GUI is displayed (Fig. 19).

The displayed GUI (Fig. 19) is used by the logged-in user to select a one of three modules: "New Customers Creating", "New Tickets Issuing", and "Tickets Verifying".

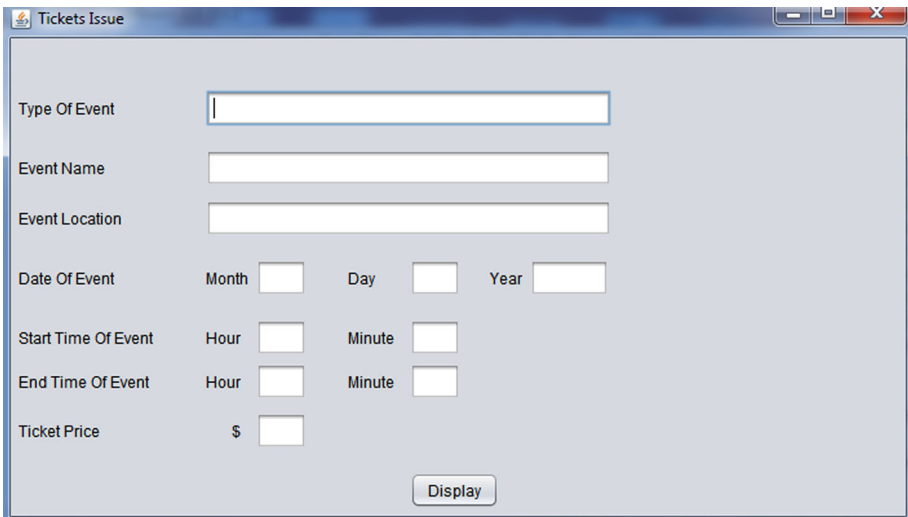
### 6.2.2 New Customer Creating Module

In this module, the logged-in users are responsible for inserting the information related to new customers. First, the new customer presents his/her empty e-ticket tag within the reading range of the RFID reader. Then, an GUI is displayed (Fig. 20) and the logged-in user inserts the required information related to the customer. After inserting the information, the user clicks "Sign and Save" button. After clicking the button, the back-end server requests to insert an TP by the new customer. Then, the server employs the signing private key of logged-in user to sign the information and TP of the new customer. Then, the back-end server stores the customer's information and the digital signature in the presented e-ticket tag of the customer.



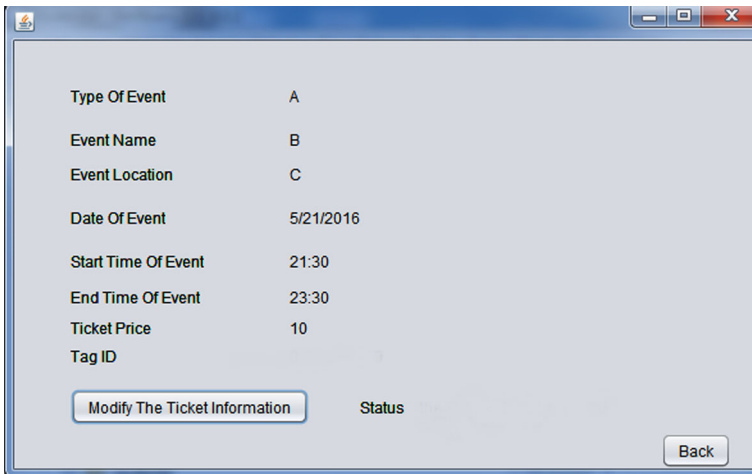
The screenshot shows a window titled "Create New Customers" with a light gray background. It contains several input fields and two buttons. The fields are: "First Name" (a single-line text box with a cursor), "Middle Name" (a single-line text box), "Last Name" (a single-line text box), "Birth Year" (a single-line text box), "Gender" (a dropdown menu with "Male" selected), "Nationality" (a dropdown menu with "Iraq" selected), "Email" (a single-line text box), and "Mobile number" (a single-line text box). Below these is a "Card ID" label and the instruction "Please present tag within range of RFID reader". At the bottom are two buttons: "Sign and save" and "Back".

Fig. 20 GUI for IVETS’s customer information entry with empty fields

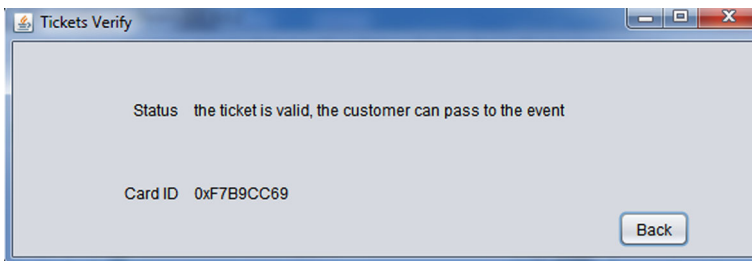


The screenshot shows a window titled "Tickets Issue" with a light gray background. It contains several input fields and one button. The fields are: "Type Of Event" (a single-line text box), "Event Name" (a single-line text box), "Event Location" (a single-line text box), "Date Of Event" (three separate single-line text boxes for "Month", "Day", and "Year"), "Start Time Of Event" (two separate single-line text boxes for "Hour" and "Minute"), "End Time Of Event" (two separate single-line text boxes for "Hour" and "Minute"), and "Ticket Price" (a single-line text box with a "\$" symbol to its left). At the bottom center is a "Display" button.

Fig. 21 GUI for inserting ticket information



**Fig. 22** GUI for issuing e-tickets

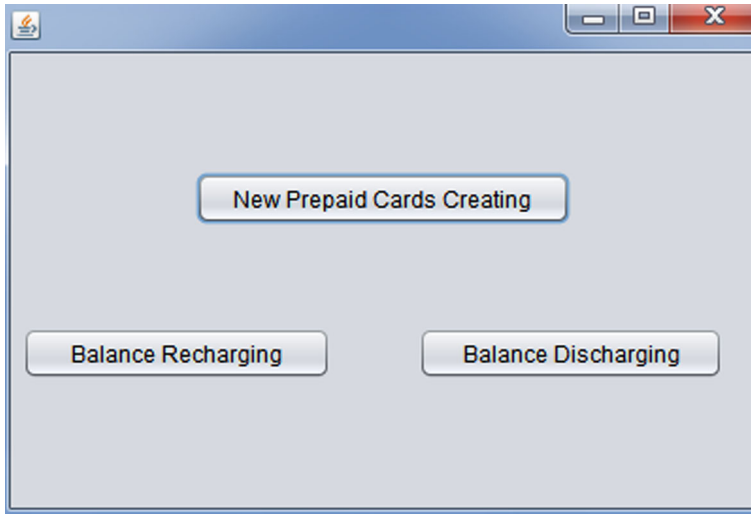


**Fig. 23** GUI for verifying e-ticket

### 6.2.3 New Tickets Issuing Module

In this module, the users issue new e-tickets to the customers. First, the logged-in user inserts the ticket's information using an GUI shown in Fig. 21.

After inserting the ticket's information, the user clicks the "Display" button to display the ticket's information in another GUI (Fig. 22). After that, if a customer decides to select the displayed ticket, the customer presents his/her e-ticket tag within the reading range of the RFID reader. The server reads the information of customer and digital signature from the presented tag using the RFID reader. Then, the server requests to insert an TP by the customer. The server uses the inserted TP and the customer's information to verify the digital signature. If presented tag is valid (i.e., the digital signature is verified), the back-end server signs the displayed ticket's information and TP using the signing private key of the logged-in user. Then, the back-end server writes the ticket's information and the digital signature into the presented e-ticket tag using the RFID reader.



**Fig. 24** GUI for selecting a module by the logged-in user of CDPCS

#### 6.2.4 Tickets Verifying Module

This module is responsible for the verification of issued e-tickets in the e-ticket tags of the customers. First, a customer presents his/her e-ticket tag within the reading range of the reader. Then, the server reads the ticket's information and the digital signature from the presented tag using the RFID reader. The server verifies the issued e-ticket by verifying the digital signature using the ECDSA public key of the logged-in user. If the digital signature is verified, the e-ticket is valid and the customer can pass to the event as shown in Fig. 23.

### 6.3 The Implementation of CDPCS

This section demonstrates the concrete implementation of the CDPCS. For the CDPCS implementation, constructed Java classes and GUIs are classified into four modules; namely: User Login Module, New Prepaid Cards Creating Module, Balance Recharging Module, and Balance Discharging Module. Java codes for the CDPCS System are given in Appendix III. CDPCS modules are discussed in detail below.

#### 6.3.1 User Login Module

This module is used by the users to login to the CDPCS system. First, the back-end server demands from a user to present his/her name and password to authenticate himself/herself to the system. If the name and hash of password are found in the system's database, the user is authenticated and the back-end server uses the inserted name and the hash of inserted password as an index to read the information of the user and his/her encrypted private key from the system's database. Then, the server requests to insert a password from the user to decrypt the encrypted private key. After that, the following GUI is displayed (Fig. 24).

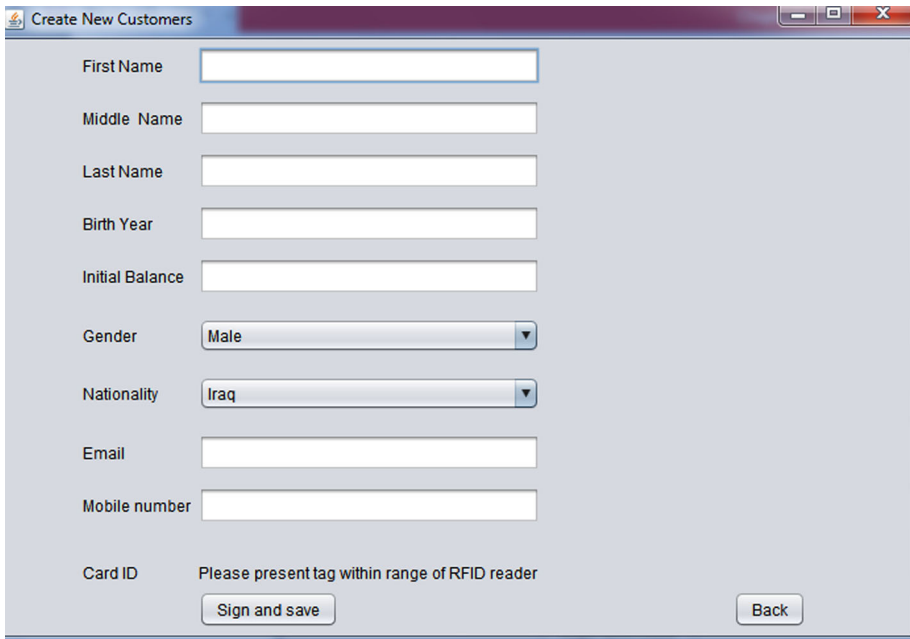


Fig. 25 GUI for CDPCS’s customer information entry

The displayed GUI (Fig. 24) is used by the logged-in user to select a one of three modules: “New Prepaid Cards Creating”, “Balance Recharging”, and “Balance Discharging”.

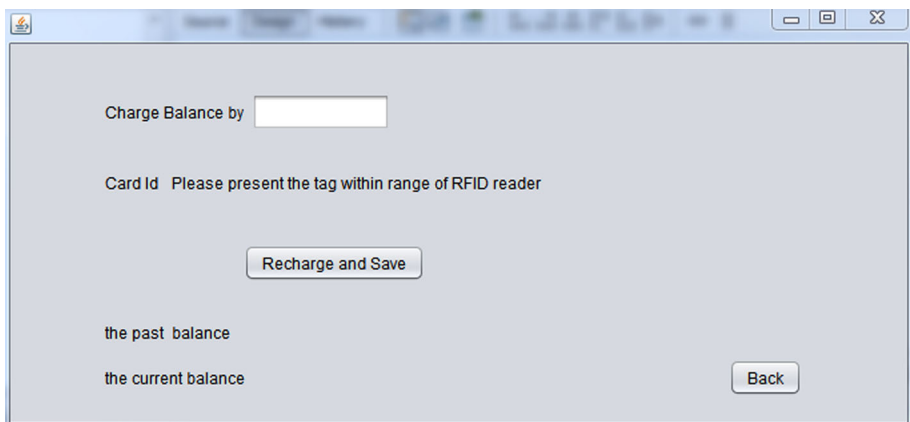
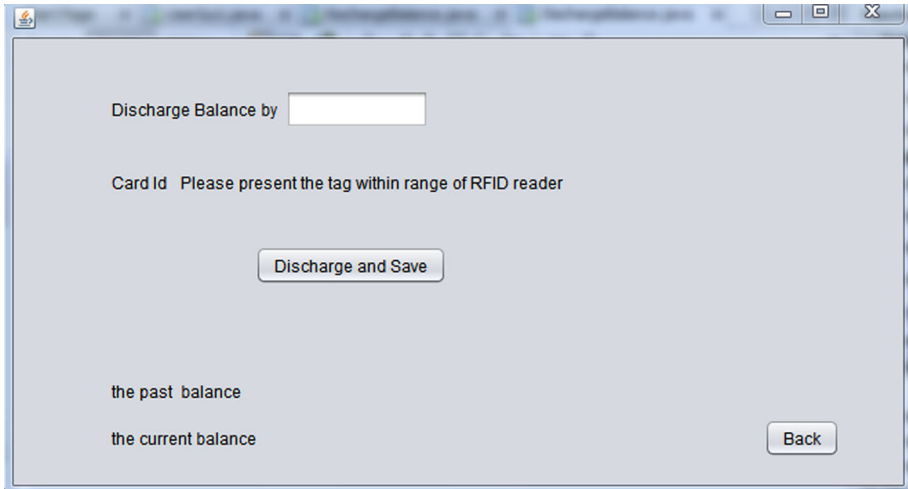


Fig. 26 GUI for recharging balance



**Fig. 27** GUI for discharging balance

### 6.3.2 New Prepaid Cards Creating Module

In this module, the users are responsible for inserting the information of new prepaid cards for new customers. First, a new customer presents his/her empty prepaid card within the reading range of the RFID reader. Then, an GUI is displayed (Fig. 25), and the logged-in user inserts the required information related to the new customer. Then, the user clicks the “Sign and Save” button. After clicking the button, the back-end server requests to insert an TP by the customer. Then, the information and TP of the customer are signed using the signing private key of the logged-in user. Then, the customer’s information and the digital signature are stored in the presented prepaid card by the back-end server.

### 6.3.3 Balance Recharging Module

In this module, the logged-in users are responsible for recharging the balances of prepaid cards of the customers. First, a customer presents his/her prepaid card within the reading range of the RFID reader. Then, an GUI is displayed (Fig. 26), and the logged-in user inserts the amount of increasing of balance to be added to the current balance of the presented prepaid card. After that, the user clicks the “Recharge and Save” button. After the clicking, the back-end server reads the customer’s information and the digital signature from the presented card. Then, the server requests to insert an TP by the customer. After that, the back-end server verifies the prepaid card by verifying the digital signature. If presented card is valid (i.e., the digital signature is verified), the back-end server recharges the balance of the prepaid card and signs the information and the TP of the customer using the private key of the logged-in user. Then, the server writes the information and the digital signature to the presented card of the customer.

### 6.3.4 Balance Discharging Module

In this module, the users are responsible for discharging the balances of prepaid cards of the customers. Firstly, a customer presents his/her prepaid card within the reading range of the RFID reader. Then, a GUI is displayed (Fig. 27), and the logged-in user inserts the amount of decreasing of balance to be reduced from the current balance of the presented prepaid card. After that, the user clicks the “Discharge and Save” button. Then, the back-end server reads the customer’s information and the digital signature from the presented card tag. After that, the back-end server requests to insert an TP by the customer. After inserting the TP, the back-end server verifies the presented card by verifying the digital signature. If presented card is valid, the back-end server discharges the balance of the prepaid card and signs the information and TP of the customer. Then, the server stores the information and digital signature in the presented prepaid card of the customer.

## 7 Conclusion

The security and privacy are very important requirements of RFID systems. The typical low-cost RFID tags can carry out lightweight operations like the bitwise logical operations, CRC and PRNG to address security and privacy issues. But these limited-resource tags cannot perform heavy-weighted cryptographic operations, which are needed for constructing secure protocols. This paper proposed the ITPMAP protocol to overcome the drawbacks in the existing authentication protocols and provides the missing features in the related works. The proposed ITPMAP protocol is designed by using heavy-weighted cryptographic techniques (the ECDSA and AES PBE schemes) on the back-end server side instead of lightweight cryptographic techniques to provide the security and privacy requirements. As proof of concept, the proposed ITPMAP protocol is adapted by three RFID systems; namely: the SVGCS, IVETS, and CDPCS. It should be mentioned that the proposed ITPMAP protocol is reusable by the three systems. The proposed ITPMAP protocol is analyzed and compared with the other authentication protocols in terms of the elected requirements (Sect. 4). As a result, the proposed ITPMAP achieves the nine identified security and privacy requirements; namely: data confidentiality, non-repudiation, data origin authentication, data integrity, desynchronization attack resistance, spoofing/cloning attack resistance, MitM attack resistance, replay attack resistance, and low computational cost on the tag-side. The proposed ITPMAP protocol can be used in various RFID and IoE applications at different sectors that require sensitive security and privacy requirements. Besides the RFID technology, the proposed ITPMAP protocol may be implemented using other wireless technologies like the NFC, Wi-Fi, Bluetooth, and Cellular. In addition, the proposed ITPMAP protocol can be integrated with physical identification systems (e.g., face recognition, and finger print).

## Appendix I: Java Codes for the SVGCS System

### 1 User Login Module

```
void SVGCS_User_Login_Module() {
    getLoginNameAndPasswordFromUser();
    if(isUserNameAndHashedPasswordInDatabase==true)
        readUserInformationAndPublicKeyAndEncryptedPrivateKeyFromDatabase();
    password=getPasswordFromUser();
    secretKey=deriveSecretKeyFromInsertedPassword(password);
    privateKey=decryptUsingAESPBE(secretKey,encryptedPrivateKey);
    displaySelectingModuleGui(privateKey,publicKey,userInformation);
    return;
}
```

### 2 Graduation Certificate Creating Module

```
void SVGCS_Graduation_Certificate_Creating_Module() {
    getNationalityFromTextField();
    getGenderFromTextField();
    getUniversityNameFromTextField();
    getCollegeNameFromTextField();
    getDepartmentNameFromTextField();
    getFirstNameFromTextField();
    getMiddleNameFromTextField();
    getLastNameFromTextField();
    getGraduationYearFromTextField();
    getMobileNumberFromTextField();
    getEmailFromTextField();
    getStudentDegreesAndAttemptsFromTextFields();
    getTagPassword();
    signStudentInformationAndTP(userPrivateKey,studentInformation,TP);
    writeStudentInformationAndDigitalSignatureToTag(studentInformation,digitalSignature);
    writeStudentInformationToDatabase(studentInformation);
    return;
}
```

### 3 Student Information Updating Module

```
void SVGCS_Student_Information_Updating_Module() {
    readStudentInformationAndDigitalSignatureFromTag();
    getTagPasswordFromGraduateStudent();
    verifyStudentInformationAndTP(userPublicKey,digitalSignature,studentInformation,TP);
    displayStudentInformationInGui(studentInformation);
    if(isStudentInformationIncorrect==true)
        correctStudentInformation(studentInformation);
    else if(isStudentInformationNeedUpdate==true)
        updateStudentInformation(studentInformation);
    signStudentInformationAndTP(userPrivateKey,studentInformation,TP);
    writeStudentInformationAndDigitalSignatureToTag(studentInformation,digitalSignature);
    writeStudentInformationToDatabase(studentInformation);
    return;
}
```

### 4 Graduation Certificate Verifying Module

```
void SVGCS_Graduation_Certificate_Verifying_Module() {
    readStudentInformationAndDigitalSignatureFromTag();
    getTagPasswordFromGraduateStudent();
    verifyStudentInformationAndTP(userPublicKey,digitalSignature,studentInformation,TP);
    displayStudentInformationInGui(studentInformation);
    return;
}
```



## Appendix II: Java Codes for the IVETS System

### 1 User Login Module

```

void IVETS_User_Login_Module() {
    getLoginNameAndPasswordFromUser();
    if(isUserNameAndHashedPasswordInDatabase==true)
        readUserInformationAndPublicKeyAndEncryptedPrivateKeyFromDatabase();
    password=getPasswordFromUser();
    secretKey=deriveSecretKeyFromInsertedPassword(password);
    privateKey=decryptUsingAESPBE(secretKey,encryptedPrivateKey);
    displaySelectingModuleGui(privateKey,publicKey,userInformation);
    return;
}

```

### 2 New Customer Creating Module

```

void IVETS_New_Customer_Creating_Module() {
    getNationalityFromTextField();
    getGenderFromTextField();
    getFirstNameFromTextField();
    getMiddleNameFromTextField();
    getLastNameFromTextField();
    getBirthYearFromTextField();
    getMobileNumberFromTextField();
    getEmailFromTextField();
    getTagPassword();
    signCustomerInformationAndTP(userPrivateKey,customerInformation,TP);
    writeCustomerInformationAndDigitalSignatureToTag(customerInformation,digitalSignature);
    writeCustomerInformationToDatabase(customerInformation);
    return;
}

```

### 3 New Tickets Issuing Module

```

void IVETS_New_Tickets_Issuing_Module() {
    getTicketInformationFromTextFields();
    displayTicketInformationInGui();
    readCustomerInformationAndDigitalSignatureFromTag();
    getTagPasswordFromCustomer();
    verifyCustomerInformationAndTP(userPublicKey,digitalSignature,customerInformation,TP);
    signTicketInformationAndCustomerInformationAndTP(userPrivateKey,customerInformation,
    ticketInformation,TP);
    writeTicketInformationAndCustomerInformationAndDigitalSignatureToTag
    (customerInformation,ticketInformation,digitalSignature);
    writeTicketInformationAndCustomerInformationToDatabase
    (customerInformation,ticketInformation);
    return;
}

```

### 4 Tickets Verifying Module

```

void IVETS_New_Tickets_Verifying_Module() {
    readTicketInformationAndCustomerInformationAndDigitalSignatureFromTag();
    getTagPasswordFromCustomer();
    verifyCustomerInformationAndTP (userPublicKey,digitalSignature,ticketInformation,
    customerInformation,TP);
    modifyTicketInformationToBeInvalid();
    signTicketInformationAndCustomerInformationAndTP(userPrivateKey,customerInformation,
    ticketInformation,TP);
    writeTicketInformationAndCustomerInformationAndDigitalSignatureToTag
    (customerInformation,ticketInformation,digitalSignature);
    writeTicketInformationAndCustomerInformationToDatabase
    (customerInformation,ticketInformation);
    return;
}

```

## Appendix III: Java Codes for the CDPCS System

### 1 User Login Module

```
void CDPCS_User_Login_Module() {
    getLoginNameAndPasswordFromUser();
    if(isUserNameAndHashedPasswordInDatabase==true)
        readUserInformationAndPublicKeyAndEncryptedPrivateKeyFromDatabase();
    password=getPasswordFromUser();
    secretKey=deriveSecretKeyFromInsertedPassword(password);
    privateKey=decryptUsingAESPBE(secretKey,encryptedPrivateKey);
    displaySelectingModuleGui(privateKey,publicKey,userInformation);
    return;
}
```

### 2 New Prepaid Cards Creating Module

```
void CDPCS_New_Customer_Creating_Module() {
    getNationalityFromTextField();
    getGenderFromTextField();
    getFirstNameFromTextField();
    getMiddleNameFromTextField();
    getLastNameFromTextField();
    getBirthYearFromTextField();
    getInitialBalanceFromTextField();
    getMobileNumberFromTextField();
    getEmailFromTextField();
    getTagPassword();
    signCustomerInformationAndTP(userPrivateKey,customerInformation,TP);
    writeCustomerInformationAndDigitalSignatureToTag(customerInformation,digitalSignature);
    writeCustomerInformationToDatabase(customerInformation);
    return;
}
```

### 3 Balance Recharging Module

```
void CDPCS_Balance_Recharging_Module() {
    readCustomerInformationAndDigitalSignatureFromTag();
    getTagPasswordFromCustomer();
    verifyCustomerInformationAndTP(userPublicKey,digitalSignature,customerInformation,TP);
    rechargeBalance(customerInformation,amountOfIncreasingInBalance);
    signCustomerInformationAndTP(userPrivateKey,customerInformation,TP);
    writeCustomerInformationAndDigitalSignatureToTag(customerInformation,digitalSignature);
    writeCustomerInformationToDatabase(customerInformation);
    return;
}
```

### 4 Balance Discharging Module

```
void CDPCS_Balance_Discharging_Module() {
    readCustomerInformationAndDigitalSignatureFromTag();
    getTagPasswordFromCustomer();
    verifyCustomerInformationAndTP(userPublicKey,digitalSignature,customerInformation,TP);
    dischargeBalance(customerInformation,amountOfDecreasingInBalance);
    signCustomerInformationAndTP(userPrivateKey,customerInformation,TP);
    writeCustomerInformationAndDigitalSignatureToTag(customerInformation,digitalSignature);
    writeCustomerInformationToDatabase(customerInformation);
    return;
}
```

## References

1. Syamsuddin, I., Dillon, T., Chang, E., & Han, S. (2008). A survey of RFID authentication protocols based on hash-chain method. In *Proceedings of Third International Conference on Convergence and Hybrid Information Technology* (pp. 559–564), USA.
2. Chaouchi, H. (2010). *The internet of things: Connecting objects*. Hoboken: Wiley.
3. Muhic, I., & Hodzic, M. (2014). Internet of things: Current technological review and new low power wireless sensor network protocol proposal. *Southeast Europe Journal of Soft Computing*, 3(2), 46–57.
4. Bilal, Z. (2015). Addressing security and privacy issues in low-cost RFID systems. Ph.D. Thesis, University of London, England.
5. Yousuf, Y., & Potdar, V. (2008). A survey of RFID authentication protocols. In *Proceedings of 22nd International Conference on Advanced Information Networking and Applications* (pp. 1346–1350), Japan.
6. Younis, M. I., & Abdulkareem, M. H. (2017). A survey of RFID authentication protocols. *Inventi Impact: Information Security*, 2017(1), 1–12.
7. Henrici, D., & Muller, P. (2004). Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers. In *Proceedings of Second IEEE Annual Conference on Pervasive Computing and Communications Workshops* (pp. 149–153), USA.
8. Choi, E. Y., Lee, S. M., & Lee, D. H. (2005). Efficient RFID authentication protocol for ubiquitous computing environment. *Lecture Notes in Computer Science*, 3823, 945–954.
9. Osaka, K., Takagi, T., Yamazaki, K., & Takahashi, O. (2006). An efficient and secure RFID security method with ownership transfer. In *Proceedings of 2006 IEEE International Conference on Computational Intelligence and Security* (pp. 1090–1095), China.
10. Ha, J., Moon, S., Nieto, J. M. G., & Boyd, C. (2007). Low-cost and strong-security RFID authentication protocol. *Lecture Notes in Computer Science*, 4809, 795–807.
11. Song, B., & Mitchell, C. J., (2008). RFID authentication protocol for low-cost tags. In *Proceedings of First ACM Conference on Wireless Network Security* (pp. 140–147), USA.
12. Liu, A. X., & Bailey, L. A. (2009). PAP: A privacy and authentication protocol for passive RFID tags. *Computer Communications*, 32(7), 1194–1199.
13. Sadighian, A., & Jalili, R. (2009). AFMAP: Anonymous forward-secure mutual authentication protocols for RFID systems. In *Proceedings of Third International Conference on Emerging Security Information, Systems and Technologies* (pp. 31–36), Greece.
14. Cho, J., Jeong, Y., & Park, S. O. (2012). Consideration on the brute-force attack cost and retrieval cost: a hash-based radio-frequency identification (RFID) tag mutual authentication protocol. *Computers & Mathematics with Applications*, 69(1), 58–65.
15. Srivastava, K., Awasthi, A. K., Kaul, S. D., & Mittal, R. C. (2015). A hash based mutual RFID tag authentication protocol in telecare medicine information system. *Journal of Medical Systems*, 39(1), 1–5.
16. Chien, H., & Huang, C. (2007). A lightweight RFID protocol using substring. *Lecture Notes in Computer Science*, 4808, 422–431.
17. Kim, K. H., Choi, E. Y., Lee, S. M., & Lee, D. H. (2006). Secure EPCglobal class-1 gen-2 RFID system against security and privacy problems. *Lecture Notes in Computer Science*, 4277, 362–371.
18. Sun, H., & Ting, W. (2009). A Gen2-based RFID authentication protocol for security and privacy. *IEEE Transactions on Mobile Computing*, 8(8), 1052–1062.
19. Niu, H., Taqieddin, E., & Jagannathan, S. (2015). EPC Gen2v2 RFID standard authentication and ownership management protocol. *IEEE Transactions on Mobile Computing*, 15(1), 137–149.
20. Burmester, M., & Medeiros, B. (2008). The security of EPC Gen2 compliant RFID protocols. *Lecture Notes in Computer Science*, 5037, 490–506.
21. Qingling, C., Yiju, Z., & Yonghua, W. (2008). A minimalist mutual authentication protocol for RFID system & ban logic analysis. In *ISECS International Colloquium on Computing, Communication, Control, and Management* (pp. 449–453), China.
22. Yeh, T., Wang, Y., Kuo, T., & Wang, S. (2010). Securing RFID systems conforming to EPC class 1 generation 2 standard. *Expert Systems with Applications*, 8(12), 7678–7683.
23. Deng, G., Li, H., Zhang, Y., & Wang, J. (2013). Tree-LSHB+: An LPN-based lightweight mutual authentication RFID protocol. *Wireless Personal Communications*, 72(1), 159–174.
24. Zhou, J. (2015). A quadratic residue-based lightweight RFID mutual authentication protocol with constant-time identification. *Journal of Communications*, 10(2), 117–123.

25. Peris-Lopez, P., Hernandez-Castro, J. C., Estevez-Tapiador, J. M., & Ribagorda, A. (2006). LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags. In *Proceedings of Second Workshop on RFID Security*, Austria.
26. Peris-Lopez, P., Hernandez-Castro, J. C., Estevez-Tapiador, J. M., & Ribagorda, A. (2006). M2AP: A minimalist mutual-authentication protocol for low-cost RFID tags. *Lecture Notes in Computer Science*, 4159, 912–923.
27. Chien, H. (2007). SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. *IEEE Transactions on Dependable and Secure Computing*, 4(4), 337–340.
28. Li, T. (2008). Employing lightweight primitives on low-cost RFID tags for authentication. In *Proceedings of 2008 IEEE vehicular technology conference* (pp. 1–5), Canada.
29. Peris-Lopez, P., Hernandez-Castro, J. C., Estevez-Tapiador, J. M., & Ribagorda, A. (2009). Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol. *Lecture Notes in Computer Science*, 5379, 56–68.
30. Lee, Y., Hsieh, Y., You, P., & Chen, T. (2009). A new ultralightweight RFID protocol with mutual authentication. In *Proceedings of 2009 WASE International Conference on Information Engineering* (pp. 58–61), China.
31. Kianersi, M., Gardeshi, M., & Arjmand, M. (2011). SULMA: A secure ultra light-weight mutual authentication protocol for low cost RFID tags. *International Journal of UbiComp*, 2(2), 17–24.
32. Lee, Y. (2012). Two ultralightweight authentication protocols for low-cost RFID tags. *Applied Mathematics and Information Sciences*, 6, 425–431.
33. Tian, Y. (2012). A new ultralightweight RFID authentication protocol with permutation. *IEEE Communications Letters*, 16(5), 702–705.
34. NXP Semiconductors (2014). Mainstream contactless smart card IC for fast and easy solution development. *Product Data Sheet*, Rev. 3.0. Available at [https://www.nxp.com/documents/data\\_sheet/MFIS50YYX.pdf](https://www.nxp.com/documents/data_sheet/MFIS50YYX.pdf).
35. Liao, H., & Shen, Y. (2006). On the elliptic curve digital signature algorithm. *Tunghai Science*, 8, 109–126.
36. Khalique, A., Singh, K., & Sood, S. (2010). Implementation of elliptic curve digital signature algorithm. *International Journal of Computer Applications*, 2(2), 21–27.
37. Abdalla, M., Fouque, P. A., & Pointcheval, D. (2006). Password-based authenticated key exchange in the three-party setting. *IEE Information Security*, 153(1), 27–39.
38. Atreya, M. (2004). Password based encryption. [https://web.cs.ship.edu/~cdgira/courses/CSC434/Fall2004/docs/course\\_docs/Article3-PBE.pdf](https://web.cs.ship.edu/~cdgira/courses/CSC434/Fall2004/docs/course_docs/Article3-PBE.pdf). Accessed Oct 22, 2016.
39. Jacobs, B. (2009). Architecture is politics: Security and privacy issues in transport and beyond. In *Proceedings of Second International Conference on Computers, Privacy and Data Protection* (pp. 289–299), Belgium.
40. Barker, E., & Roginsky, A. (2015). Transitions: Recommendation for transitioning the use of cryptographic algorithms and key lengths. *NIST Special Publication*. 800-131A Rev. 1. <http://dx.doi.org/10.6028/NIST.SP.800-131Ar1>
41. Smart, N. (2012). Ecrypt II yearly report on algorithms and key sizes (2011–2012). *Technical Report*. <http://cordis.europa.eu/docs/projects/cnect/6/216676/080/deliverables/002-DSPA20.pdf>
42. D-Logic. uFR classic NFC RFID reader. <http://www.d-logic.net/nfc-rfid-reader-sdk/products/ufr-classic>
43. Al-Tameemi, Z. F. A. (2010). Design and implementation of a scalable automated RFID-based attendance system with scheduling technique. M.Sc. Thesis, Universiti Sains Malaysia (USM), Malaysia.
44. Bock, H. (2011). *The definitive guide to NetBeans platform 7 (expert's voice in Java)*, (1st edn.). CA: Apress Berkely.



**Mohammed Issam Younis** obtained his Doctorate in Computer Engineering from Universiti Sains Malaysia in 2011. He had done the M.Sc. and B.Sc. in Computer Engineering from University of Baghdad in 2001 and 1997 respectively. His research interests are: Distributed System, Information Security and Cryptography, Digital Signature and non-repudiation Protocols, Algorithms, Computer Networking, Software Engineering, RFID, and IoT. He has various publications as books, thesis, journals, Invited IEEE Tutorials. He is associated with various committee like: Iraqi Union of Engineers, Cisco Networking Academy, Software Engineering Research Groups, AIDL Research Groups. He honored by different awards, medals, patents, and grants. Assoc. Prof. Dr. Younis is currently a faculty member and Cisco Instructor at the Computer Engineering Department, College of Engineering, University of Baghdad.



**Mustafa Hashim Abdulkareem** received the M.Sc. and B.Sc. in Computer Engineering from the University of Baghdad in 2017 and 2014 respectively. He is associated with various committee like: Iraqi Union of Engineers, and Cisco Networking Academy. His research interests involve: wireless network security, authentication protocols, RFID development, computer networks, and IoE.