# SE-H: Secure and Efficient Hash Protocol for RFID System

Shadi Nashwan

Department of Computer Science and Information, Aljouf University, Saudi Arabia

***Abstract*:** The Radio Frequency Identification (RFID) systems are suffered from the growing of security vulnerabilities. In the RFID Applications, an attackers may exploit these vulnerabilities to break the tag holder privacy and desynchronize the communication messages between system components. The majority of the proposed solutions fail to solve the existing vulnerabilities. Therefore, the authentication protocol is the main challenge in such systems. This paper proposes a secure and efficient hash authentication (SE-H) protocol for RFID system to support attractive security features. Compared with the recent RFID authentication protocols, the SE-H protocol cannot only perform strong security features including mutual authentication and the data secrecy features, but also perform the user anonymity and untraceability features with low cost performance. The security analysis proves that the SE-H protocol is resistant to the current vulnerabilities. Furthermore, the performance analysis in term of the authentication operations cost illustrates that SE-H protocol is more efficient than the recent RFID protocols.

***Keywords*:** Radio Frequency Identification (RFID), Mutual Authentication, Key Derivation Function (KDF), Hash Function.

## 1. Introduction

The RFID technology is used in many applications to identify the objects in numerous fields [2, 3, 12 and 18]. The RFID technology has some advantages over other automatic identification technologies such as the optical barcodes. In RFID systems, millions of objects with the same properties can be distinguished by RFID tag and can identify the objects without the need for line of sight [4]. Due to the lacking of the security mechanisms, the RFID systems are vulnerable for various existing attacks [9, 20 and 21]. Therefore, the security aspects will be the main attribute to grow more and more the demand on these applications [10].

In the RFID systems, the back-end server is responsible for the authentication functions which is connected to a group of the RFID readers. The reader broadcasts the radio frequency (RF) to communicate with of RFID tags where each reader has a unique identity (IDR1, IDR2...IDRn). The RFID reader relays the communication messages between the back-end server and the RFID tag. The latter is an identification device with lower amount of computing capabilities, each tag has a unique identity (IDT1, IDT2... IDTm) and can represent only one object in the system. Usually in the RFID applications, the number RFID tags is much higher than the number of RFID readers [22].

Figure 1 shows the RFID system components. In order to authenticate the RFID tag, the closest RFID reader queries the RFID tag to obtain the tag data through the RF channel. Then the RFID reader relays this data to the back-end server. The latter includes the database that contains the tags and readers information. Upon receiving the tag data, the back-end server decides whether the tag is authorized or not [5, 6 and 8]. In order to achieve acceptable level of security, a lot of security mechanisms have been proposed to offer attractive security features for the RFID systems [5, 10 and 13]. Unfortunately, due to the resources limitations of the RFID tags, always the proposed mechanisms have been suffered from one or more weaknesses. It is plausible to say that, the design majority of the proposed authentication protocols can be classified into two groups: (1) strong authentication protocols with low performance level but cannot be considered applicable to use in the RFID systems; (2) weak authentication protocols with high performance level which are been suffered from different security vulnerabilities. Therefore, the authentication protocol is the main challenge in such systems.

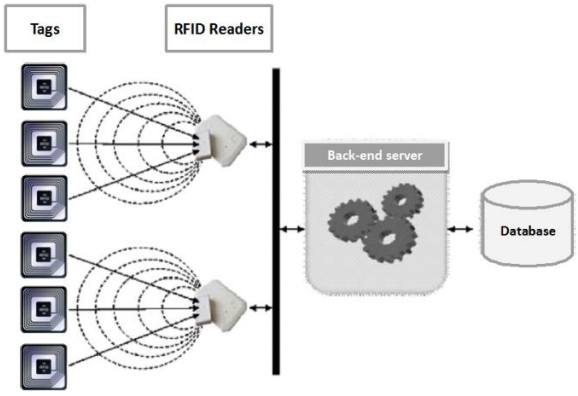



**Figure 1.** RFID System components

The security features and requirements that must be taken in account during the designing of the RFID authentication protocol to resist all expected attacks can be summarized as the following [18, 19 and 20]. The mutual authentication must be between all authentication entities of the RFID system, i.e., the tag, reader and back-end server.

In the same context, the tag anonymity and untraceability features must be achieved to satisfy the tag privacy [15, 21]. The authentication protocol must provide a method to conceal the tag identity in all authentication sessions. Thus, this method can prevent the attackers from exploit the tag data that sent previously. The secrecy is another feature that must be satisfied, the authentication protocols must provide a method to achieve the backward/forward secrecy [10, 23]. Subsequently, an attacker cannot reversely deduce the previous secret session key from the current session key, and cannot deduce the future secret session key from current session key due to using the same key derivation functions [2, 3 and 7].

As mentioned before, the authentication is an essential aspect to defeat the current attacks in the RFID systems. Considering these security vulnerabilities, this paper proposes a secure and efficient hash (SE-H) protocol for

RFID systems to support attractive security features. Comparing with the recent RFID authentication protocols, SE-H cannot perform only mutual authentication feature, but also can perform the user anonymity and untraceability features with low cost performance.

The proposed protocol can defeat the existing attacks such as impersonate attacks, desynchronization attacks, tracking attacks and replay attacks. Moreover, the performance analysis in term of operations cost of authentication illustrates that the proposed protocol is more efficient than the recent mutual authentication protocols.

The remaining sections of this paper are organized as follows: In section 2, the related works is discussed. The proposed protocol is introduced in section 3 .The security and performance analysis of the proposed protocol are illustrated in section 4 and 5, respectively. Finally, this paper will be concluded in section 6.

## 2. Related Works

With increasing the demand on the RFID applications, numerous authentication protocols have been proposed to overcome the security vulnerabilities in RFID systems [24-26].

In 2014, Soni and Sharma [13] design a mutual authentication protocol using the Elliptic Curve Integrated Encryption System (ECIES) for RFID systems. The proposed protocol is based on the public key technique. Soni and Sharma protocol mainly focuses on defeating the existing attacks without looking to the amount of computing capabilities of the RFID tag. However, the protocol has greatly increased the storage size, processing time and transmission overhead.

In 2014, Chowdhury and Ansary [16] introduce a mutual authentication protocol for RFID systems. This protocol supposes that the communication channel between the back-end server and reader is secure, while the communication channel between the reader and tag is not secure. The authentication parameters, i.e., the timestamp and the random numbers are exchanged between authentication entities to authenticate each other. Chowdhury and Ansary protocol contains three drawbacks: (1) the mutual authentication is not achieved between the reader and tag; (2) the tag cannot compute the hash value without received both of the timestamp and random number as plain text; (3) the back-end server consumes a lot of time to compute the hash value to find the tag identity. In this way, the authentication cannot be secured against the all current attacks.

In 2015, Abughazalah et al. [17] propose a mutual authentication protocol for low-cost RFID tags. This protocol supposes that the communication channel between the back-end server and reader is secure. The proposed protocol uses the Trusted Third Party (TTP) to authenticate the reader through robust authentication protocol. In this protocol, the reader obtains the list of valid tags identities during the initialization process when authenticates itself to the TTP, both of the back-end server and tag can authenticate each other using hash function. The only parameter that is sent to the tag as plain text is the random number that has been generated by reader. In general, the protocol achieves different attractive security features. Unfortunately, this protocol includes three drawbacks: (1) in order to determine the tag identity number, the time of authentication session is consumed by the reader; (2) the tag authenticates the reader indirectly at the end of authentication session; (3) the method that is used to conceal the tag identity consumes the storage space. However, if the system includes a big number of tags then a lot of time will be consumed to compute the hash value that is lead to identity of tag by the reader.

In 2016, Omolola and Osunade [14] present another mutual authentication protocol for low-cost RFID (named SMAP). In this protocol, the back-end server authenticates the reader using a hash function, the Pseudo Random Number Generator (PRNG) function is used to secure the exchanged values between the tag and back-end server though the reader, and the random number that is generated by the tag is not sent as plain to the reader. However, SMAP contains a set of the drawbacks: (1) in order to determine both of the tag and reader identities, the time of authentication session is consumed by the back-end server; (2) the authentication messages that are generated using the same input parameters is sent from tag to the reader; (3) the reader is not authenticated by the tag; (4) the protocol needs extra method for database index replacement. However, if the system include a big number of tags then a lot of time will be consumed to compute the hash value according to the right tag and reader identities by the back-end server. This protocol does not support the mutual authentication between all authentication entities.

In 2017, Zhang et al. [1] design a mutual authentication security RFID protocol based on timestamp. The protocol uses the timestamp and hash function to achieve the mutual authentication between the back-end server, reader and tag. Unfortunately, this protocol includes four drawbacks: (1) the mutual authentication is not achieved between the reader and tag; (2) the mutual authentication is not achieved between the reader and back-end server; (3) the timestamp that is generated by the reader is useless and can be replaced it with any other parameter; (4) as in previous protocols, the back-end server consumes a lot of time to compute the hash function to know the right tag and reader identities. However, this protocol does not resolve the security aspects in defeating the existing attacks and does not support the mutual authentication between all authentication entities.

## 3. Proposed Protocol (SE-H)

In this section, the SE-H protocol assumptions and requirements are listed, respectively. Then the protocol notation is presented. Finally, the details description of the proposed protocol is discussed.

### 3.1 Assumptions

The SE-H protocol is performed according to a set of assumptions: (1) the reader initiates the authentication session with the passive tag; (2) the communication channels are susceptible to various attacks between the authentication entities, i.e. tag, reader and back-end server; (3) the data in the back-end server can be accessed by a secure access control method; (4) the authentication parameters that are stored in the back-end server and tag entities can be updated; 5) the tag cannot perform any operation outside the range of reader signals.

### 3.2 Design Requirements

In order to resist the existing attacks: (1) the reader and tag

produce the timestamps within the range that is determined by the back-end server; (2) both of the back-end server and tag can update the tag identity then save the new and old values in the database after the mutual authentication is satisfied; (3) the mutual authentication must be achieved between all authentication entities; (4) the hash function is used to conceal the tag and reader identities in the whole authentication session; (5) in order to derive a new secret session key ,the KDF function is used by the authentication entities to satisfied the backward/forward secrecy feature; (6) the reader identity is used instead of tag identity to minimize the retrieval time of tag identity.

### 3.3 Notation

**Table 1**. Protocol notation

| Notation | Description |
|---|---|
| **IDTold** | The old tag's identity that is stored in the server. |
| **IDTnew** | The new tag's identity that is stored in the server. |
| **KTnew** | The new tag's secret key that is stored in the server |
| **KTold** | The old tag's secret key that is stored in the server |
| **IDT** | The tag's identity. |
| **IDR** | The reader's identity. |
| **KR** | The secret key of the reader that is shared with the server. |
| **KT** | The secret key of the tag that is shared with the server. |
| **AK** | Anonymity key. |
| **TSR** | Timestamp that is produced by the reader. |
| **TST** | Timestamp that is produced by the tag. |
| **KDF** | Key derivation function. |
| **H** | Hash function. |
| **X⊕Y** | X value is Xored with the Y value. |
| **X←Y** | X value is updated to the Y value. |
| **MAC** | Challenge authentication code. |
| **XMAX** | Expected challenge authentication code. |
| **RES** | Response message. |
| **XRES** | Expected Response message. |
| **j** | The session identity. |
| **n** | Number of tags in the system. |
| **m** | Number of readers in the system. |

Table 1 shows the SE-H protocol notation that will be used to illustrate the authentication operations in RFID tag, RFID reader and back-end server.

### 3.4 Protocol Description

Initially, the authentication entities have the following data: (1) each tag contains the tag identity (IDT), secret key (KT) and the reader's identities (IDR's) with their secret keys (KR's); (2) each reader contains the reader identity (IDR) and its secret key (KR); (3) the database of the back-end server includes the initial data of all readers and tags in the system. To manage the renew process of the tags identities in each authentication session, the back-end server database also contains the old and new tags identities.

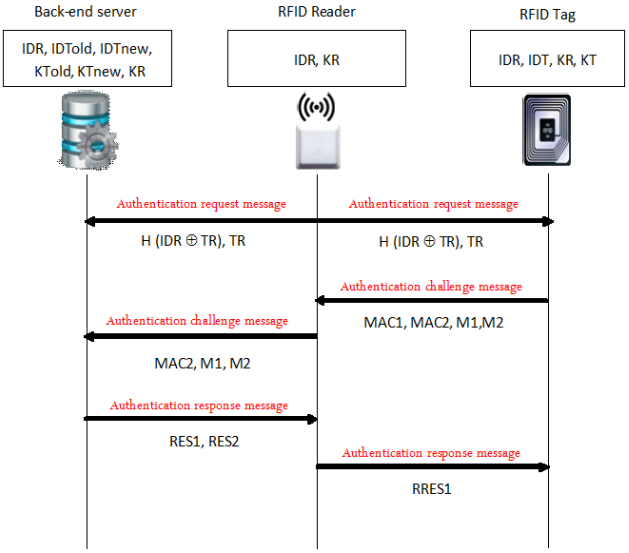



**Figure 2.** SE-H Protocol

Figure .2 illustrates the steps of proposed protocol. In order to initiate the authentication session process, the reader sends the authentication request message for both of the back-end server and tag. This message includes the timestamp (TSR) and the H (IDR⊕TSR) that have been computed by the reader. After that, the reader computes the anonymity key (AK) as AK = KDF (KR⊕TSR).

Upon receiving the authentication request message, the back-end server executes the following steps: (1) in order to determine the IDR, sequentially computes the H (IDR⊕TSR) for all stored IDR's until finds a hash value that is equal to the hash value that have been received from reader;(2) the server retrieves the KR; (3) computes the anonymity key (AK) based on the key derivation function (KDF) as AK = KDF (KR⊕TSR).

The tag computes the AK according the same steps in the back-end server. Then the tag performs the following steps to prepare the challenge message: (1) produces a new timestamp TST; (2) computes the MAC1 and MAC2 as MAC1 = H (TST⊕ KR) and MAC2 = H (TST⊕ KT⊕ IDT), respectively; (3) finally, computes the M1 and M2 as M1 = (TST⊕ AK) and M2 = (IDT⊕ AK), respectively. Consequently, the tag sends the authentication challenge message which includes the MAC1, MAC2, M1 and M2 back to the reader.

In order to authenticate the tag entity when the authentication challenge message is received by the reader, the latter performs the following steps: (1) computes the TST as TST = (M1⊕AK), if the TST is not in the range of the TSR that has been produced previously by the reader, the authentication session is terminated by the reader else; (2) computes the XMAC1 as H (TST⊕KR); (3) compares the MAC1 that has been received from the tag with the computed hash value XMAC1, if the both values are not equal, then the reader terminates the authentication session. Through steps 1 and 2, the reader authenticates the tag. After that, the reader

forwards the authentication challenge message which includes MAC2, M2 and M1 to the back-end server.

Upon receiving the authentication challenge message from the reader, the back-end server performs the following steps to prepare the response messages RES1 and RES2 for the tag and reader, respectively: (1) determines the tag identity as IDT = (AK⊕ M2); (2) computes the TST as TST = (AK⊕M1). In case both of the TST and TSR are not in the system range, the back-end server terminates the authentication session else; (4) retrieves the KT from database; (5) computes the XMAC2 as XMAC2 = H (TST ⊕ IDT⊕ KT); (6) compares the MAC2 that has been received from the reader with computed hash value XMAC2, if the values are not equal, then the back-end server terminates the authentication session. Through step 2 and 6, both of the tag and reader are authenticated by the back-end server. In order to prepare the authentication response message, the back-end server performs the following steps: (1) computes RES1 as RES1 = H (TST⊕TSR⊕KT); (2) computes RES2 as RES2 = H (TST⊕TSR⊕AK); (3) sends the authentication response message which includes RES1 and RES2 back to the reader; (4) updates both of the tag identity and secret key of the tag as ((IDTnew) j+1 = H ((IDT) j), (IDTold) j+1 = (IDT) j) and ((KTnew) j+1 = KDF ((KT) j, TST), (KTold) j+1 = (KT) j), respectively. When the reader receives the authentication response message, the reader performs the following steps: (1) computes the XRES2 as XRES2 = H (TST⊕TSR⊕AK) to verify the back-end server. In case both values, i.e., RES2 and XRES2 are not equal then the reader terminates the authentication session else; (2) relays the authentication response message which includes only the RES1 value to the tag as RES1⊕AK.

Upon receiving the authentication response message, the tag performs the following steps: (1) computes the XRES1 as XRES1 = (H (TST⊕TSR⊕KT) ⊕ AK) to authenticate both of the reader and back-end server. In case both values, i.e., XRES1 and RES1 are not equal then the tag terminates the authentication session else; (2) updates both of tag identity and secret key as (IDT) j+1←H ((IDT) j) and (KT) j+1←KDF ((KT) j, TST).

## 4. Security analysis of SE-H protocol

In this section, the security analysis is performed to illustrate that the SE-H protocol has a high level of security during the authentication session. In addition to discuss how the proposed protocol can prevent the existing attacks, the SE-H is compared with the recent mutual authentication protocols in [1, 14, 16 and 17] in terms of mutual authentication, backward/forward secrecy, anonymity, untraceability and attacks resistant.

### 4.1 Mutual authentication

The proposed protocol assumes the communication channels between all authentication entities are not secure. Therefore, the SE-H protocol uses a set of authentication parameters to achieve the mutual authentication between all system entities. The reader checks whether the TST in the range of TRT or not, then the reader verifies whether XMAC1 equals to MAC1 or not. If TT is not in the range or XMAC1 is not the same as MAC1, the tag is not legitimate. Therefore, the reader terminates the authentication session. The tag verifies that the reader and back-end server have generated correct response, if XRES1 is not equal to RES1, it terminates the authentication session, and else the tag authenticates both of reader and back-end server together. The back-end server verifies whether TST and TRT in same range or not, then computes the XMAX2 to verify whether the received XMAC2 equals to MAX2 or not. If TST and TSR are not in the same range or MAC2 is not equal to XMAX2, the back-end server terminates the authentication session. If both conditions are satisfied, the back-end server authenticates the tag and reader. In the same context, when the reader receives the RES1, it also computes the XRES1, then it checks whether the back-end server has generated the correct response or not. If XRES1 is not equal to RES1, it terminates the authentication session, else the reader authenticates the back-end server.

**Table 2**. Mutual authentication in the authentication entities.

| Authentication protocols | tag-reader | tag- back-end server | reader- back-end Server |
|---|---|---|---|
| [1] | ≠ | ≡ | ≠ |
| [14] | ≠ | ≡ | ≡ |
| [16] | ≠ | ≡ | ≠ |
| [17] | ≠ | ≡ | ≠ |
| SE-H | ≡ | ≡ | ≡ |

Table 2 shows that the mutual authentication between all authentication entities is achieved only in the proposed protocol, meanwhile is achieved partially in the other mutual authentication protocols, the notation (≡) and (≠) denote that the mutual authentication is achieved or is not achieved, respectively.

### 4.2 Backward and forward secrecy

In the proposed protocol, an adversary cannot deduce the session keys due to using one time functions, i.e., the hash and KDF functions. The KT that is stored in the both of the tag and back-end server is not transmitted as plain message, it is protected by the hash function with the TST that is generated by the tag. In the same manner, the secret key of reader that is stored in all authentication entities is not sent as plain text between authentication entities, it is protected by the hash function with the TSR that is generated by the reader. In addition to, the remaining authentication parameters are protected by the AK that has been derived by the KDF function based on the TST and the KR. Both of the TST and TSR can change the hash KDF values in each authentication session. Therefore, only the legitimate entities of the RFID system can retrieve and use the authentication session keys.

### 4.3 Anonymity and untraceability

In order to achieve the anonymity and untraceability features, the proposed protocol protects the tag and reader identities within the challenge messages either by the hash function or by the AK. In general, for all authentication parameters that include the identity and key of the tag are updated after each successful authentication session. In despite of the back-end server renews the IDTnew and stores the IDTold but before that, the back-end server checks whether TST and TSR are in the correct range or not.

### 4.4 Resistance to attacks

Assume that the attackers can catch and eavesdrop the authentication messages between the RFID system entities, and also suppose the attacker can transmit these messages to impersonate either the RFID reader or RFID tag.

The SE-H protocol has many strength properties that can be summarized as the following: (1) all parameters are protected either by hash function or by the KDF function. An attacker cannot obtain the session keys or the authentication parameters that are exchanged between the authentication entities; (2) The IDT is renewed after each successful authentication by the back-end server and the tag itself; (3) the IDT is concealed during the transmission by the AK that is changed in each authentication session according to the TST; (4) the KT is renewed after each successful authentication; (5) the IDR is protected by the hash function during the transmission where the hash value is changed according to the value of TRT that is generated by the reader itself; (6) if the authentication is fail, the existing identities and keys of the tag and reader will be used for next authentication session with fresh authentication parameters such as the TST, TRT and AK;(7) the mutual authentication must be achieved between all authentication entities.

Therefore, the proposed protocol can prevent the following: (1) unauthorized reader cannot replay the tag identity; (2) the unauthorized tag cannot replay the reader identity; (3) the attacker cannot track the tag holder; (4) the tag and reader identities or the secret keys of unsuccessful authentication session cannot be used by the attacker; (5) the authentication messages of the previous authentication session cannot be resent by the attacker; (6) the attacker cannot force the back-end server and tag to renew the tag identity when the authentication is fail; (7) the attacker cannot impersonate the tag, reader and server due to cannot retrieve the reader identity or compute the anonymity key. Consequently, the proposed protocol can resist all current attacks such as impersonate, desynchronization, tracking and replay attacks.

**Table 3.** Security properties of authentication protocol

| Security properties | [1] | [14] | [16] | [17] | SE-H |
|---|---|---|---|---|---|
| **Mutual Authentication** | ∓ | ∓ | ∓ | ∓ | = |
| **Anonymity** | = | = | = | = | = |
| **Backward and forward secrecy** | = | = | = | = | = |
| **Impersonate attack** | ∔ | = | ∓ | ∓ | = |
| **Desynchronization attack** | ∔ | ∔ | ∔ | = | = |
| **Replay attack** | = | = | = | = | = |
| **Tracking attack** | ∔ | = | ∔ | = | = |

### 4.5 Comparisons

Table 3 shows that the SE-H protocol achieves the highest level of security among the other authentication protocols, the [14] and [17] come in the middle level of security while the [1] and [16] in the last level of security. The notation (=), (∓) and (∔) denote that the security property is fully satisfied, partially satisfied and is not satisfied, respectively.

## 5. Performance analysis of SE-H protocol

This section conducts the performance analysis to observe the effect of security level that is satisfied in the SE-H protocol during the mutual authentication session. The analysis in term of operations cost is performed by comparing the SE-H protocol with the recent mutual authentication protocols for RFID systems [1, 14, 16, and 17].

The basic operations of the authentication protocols can be determined with assess how many units will be consumed by each operation. Due to the expected execution time, the basic operations are classified into three levels: (1) the Xor, concatenation, replacement and verification operations consume one unit; (2) the operations that is preformed to generate random numbers and produce timestamps consume two units; and (3) the hash and key derivation functions consume three units after excluding the inner operations.

**Table 4.** Operations cost notation and units.

| Notation | Description | Cost |
|---|---|---|
| **Gr** | Genrate a random number | 2 units |
| **Pt** | Produce a timestamp | 2 units |
| **Co** | Concatenation $\|\|$ operation | 1 unit |
| **Xo** | Xor $\oplus$ operation | 1 unit |
| **Hf** | Hash function H | 3 units |
| **Io** | If equal = = operation | 1 unit |
| **Kd** | Key dervation function | 3 units |
| **Ro** | Replacment = operation | 1 unit |

Table 4 illustrates the notation of operations in the authentication protocol and the operations cost that are performed during the authentication session in all authentication entities.

Suppose that, the operations vector (Vo) in each authentication entity can be described as Vo = [Gr, Pt, Co, Xo, Hf, Io, Kd, Ro], the weight of the Vo elements represents how many times each operations are executed in each authentication entity.

In this context, some of the operations are repeated according to the number of readers and tags in the RFID system, especially when the authentication entity retrieves the tag or reader identity from its database. In addition to, all authentication parameters in either the proposed protocol or another mutual authentication protocols have the same size; and assume the system includes (n) readers and (m) tags where m >> n.

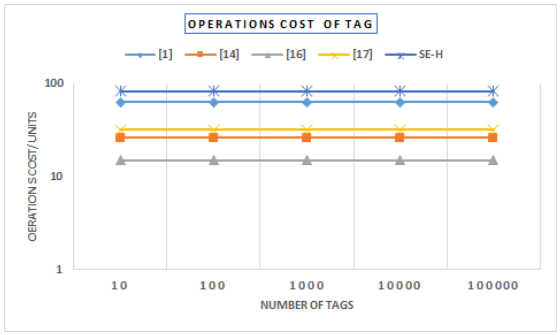
Table 5 illustrates that the maximum number of operations that are performed in each successful authentication session through the Vo's of all authentication entities (see Appendix A).

The Xo, Hf and Io operations are used to authenticate and conceal the tag identity or the reader identity; therefore the weights of these operations are proportionally increased with the increasing of n and m, respectively.

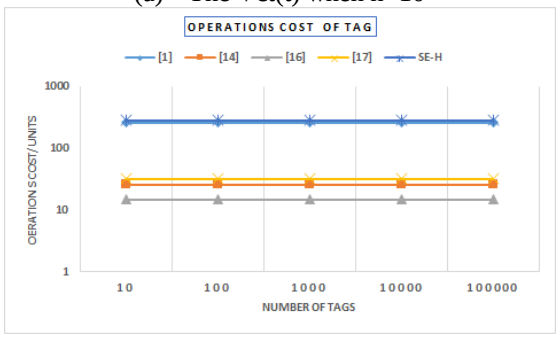
Table 6 illustrates the vectors cost for all authentication entities where the elements of the vector cost (Vc) is equal to

the operation cost multiplied by the elements weight of the Vo as table 5 (see Appendix A). .

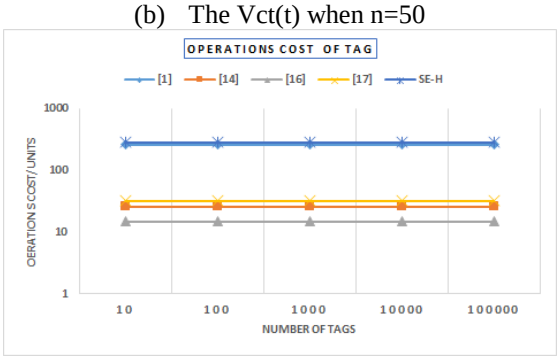
Consequently, the total vector cost (Vct) is equal to the total of elements weight of Vc for each authentication entities. Particularly, the Vct (t), Vct (r) and Vct (s) represent the total vectors cost of tag, total vector cost of reader and total vector cost of back-end server, respectively. The comparison in term of Vct for each authentication entity among RFID authentication protocols is shown in table7 (see Appendix A). In order to compare the SE-H protocol with the mutual protocols in [1, 14, 16 and 17] in terms of the total operations cost in each authentication entities and the overall operations cost in one successful authentication session, the Vct (t), Vct (r) and Vct (s) are calculated when the m = 10, 100, 1000, 10000 and 100000 with n = 10, 50 and 100, respectively.




(a) The Vct(t) when n=10




(b) The Vct(t) when n=50

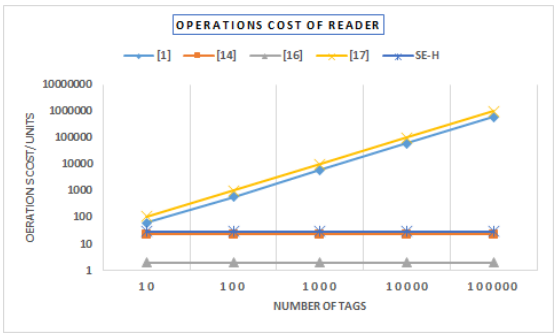



(c) The Vct(t) when n=100

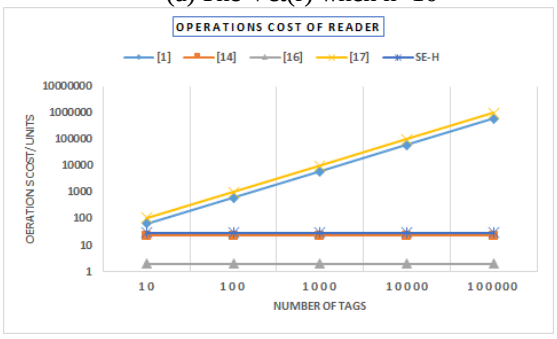
**Figure 3.** The operations cost of RFID tag

According to the value of Vct (t) in the table 7, figure 3. (a)-(c) shows the total vector cost Vct (t) among the mutual protocols when the n = 10, 50 and 100, respectively. From these figures, the SE-H and [1] protocols have the highest operations cost than others.

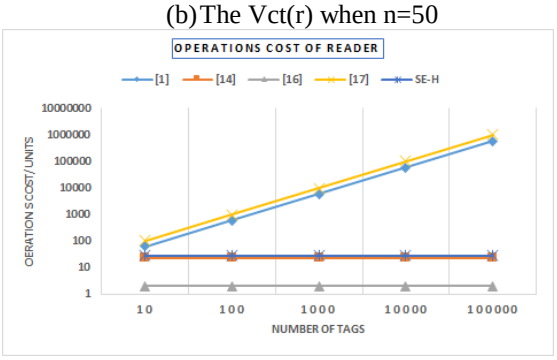
The reason for that, the Vct (t) value of the SE-H protocol is changed according the n value. The proposed protocol unlike the other protocols, the SE-H protocol shifts the impact of the number of readers to the impact of the number of tags. However, the SE-H protocol operations cost remains within the limit of the passive tag computation capabilities.




(a) The Vct(r) when n=10




(b) The Vct(r) when n=50

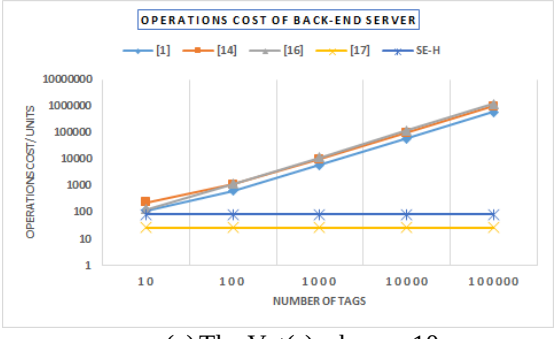



(c) The Vct(r) when n=100

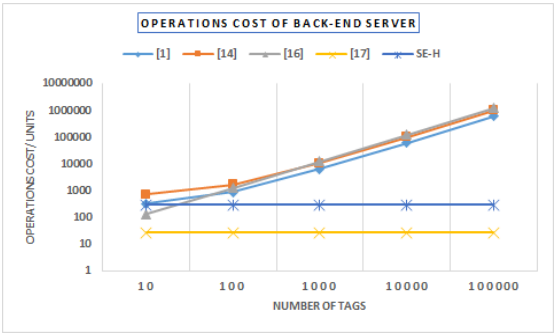
**Figure 4.** The operations cost of reader.

Based on the value of Vct (r) in the table 7, figure 4. (a)-(c) shows the total vector cost Vct (r) among the mutual protocols when the n = 10, 50 and 100, respectively. From these figures, the SE-H, [14] and [16] protocols have smallest Vct (r) value than others. The reason for that, the Vct (r) value of SE-H protocol does not change according to m value while in the [1] and [17] protocols, the value of Vct (r) is increases greatly when the value of m increases slightly.

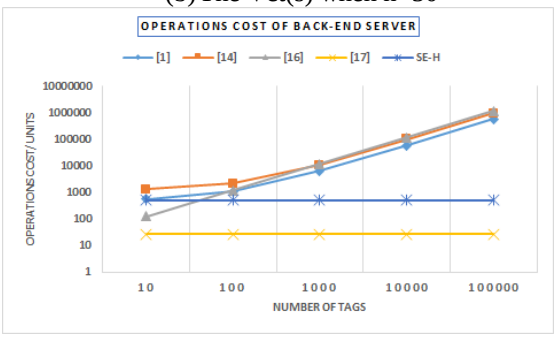
The readers should be served more than one tag in the same time, it is important to have a very high efficiency. Although the SE-H protocol is only which the reader authenticates both of the tag and back-end server among the other mutual authentication protocols, it can also achieve lowest cost of the performance.




(a) The Vct(s) when n=10




(b) The Vct(s) when n=50



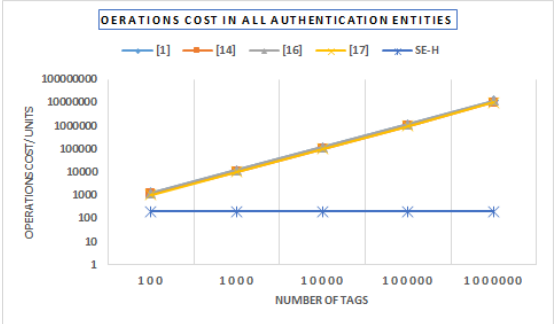

(c) The Vct(s) when n=100

**Figure 5.**The operations cost of back-end server.

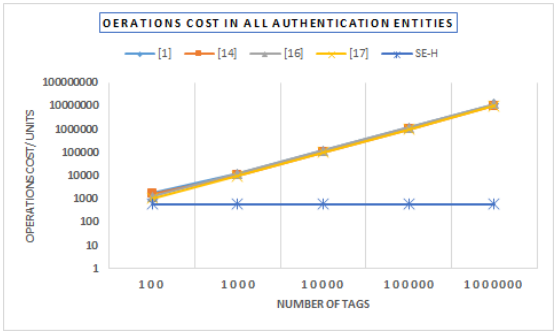
According to the value of Vct (s) in the table 7, figure 5. (a)-(c) shows the total vector cost Vct (s) among the mutual protocols when the n = 10, 50 and 100, respectively. From these figures, the SE-H and [17] protocols have smaller Vct (s) value than others. The reason for that, the Vct (s) value of SE-H protocol does not change according to m value while in the [17] protocol, the reader sends the tag identity as clear text to the back-end server. In the other protocol the value of Vct (s) is increases greatly when the value of m increases slightly. It is worth mentioning here, the back-end server authenticates both of the tag and reader and renews the tag identity after each successful authentication session in the SE-H protocol.

**Table 8.** The overall operations cost per successful authentication session.

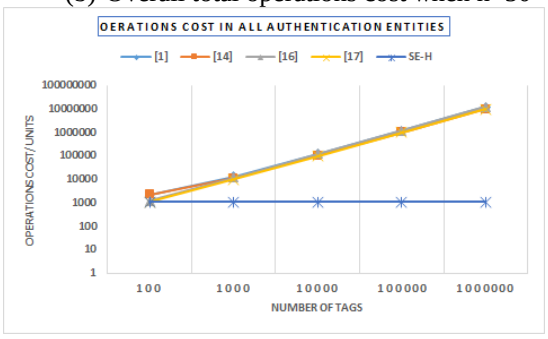
| Authentication protocols | Total operations cost |
|---|---|
| **[1]** | 10n + 12m + 26 units |
| **[14]** | 12n + 10m + 72 units |
| **[16]** | 12m + 26 units |
| **[17]** | 10m + 66 units |
| **SE-H** | 10n +98 units |




(a) Overall total operations cost when n=10




(b) Overall total operations cost when n=50




(c) Overall total operations cost when n=100

**Figure 6**. Overall operations cost of authentication session.

The overall operations cost during the successful authentication session can be calculated as Vct (t) + Vct (r) + Vct (s). The overall operations cost is represented in terms of the n and m values where obviously the m >> n. The comparison in term of overall operations cost among the recent authentication protocols is shown in table 8. Figure 6. (a)-(c) shows overall operations cost among the mutual protocols when the n = 10, 50 and 100, respectively. The overall operations cost of the proposed authentication protocol is outperformed the other mutual authentication protocols. In the SE-H, the computation overhead does not increase rapidly, the overall operations cost is based on the n value. Despite of the SE-H protocol is more secure than the other mutual authentication protocols, it can provide also the lowest cost performance.

## 6. Conclusions

This paper proposes a secure and efficient hash protocol (SE-H) for RFID systems to support attractive security features for the RFID system. Comparing with the existing RFID authentication protocols, the SE-H can achieve the mutual authentication, the data secrecy, tag holder anonymity and untraceability features. The fully mutual authentication between all authentication entities is achieved based on a set of timestamps and hash values, the forward/backward secrecy are satisfied by using the KDF functions to derive the secret and anonymity keys. The identities of the authentication entities are completely concealed using the hash function and anonymity key where the identity and the secret key of the tag are renewed in each successful authentication session. The security analysis proves that the SE-H protocol can defeat the existing attacks such as impersonate attack, tracking attack, desynchronization attack and replay attack. Moreover, the performance analysis in terms of operations cost of authentication illustrates that the proposed protocol satisfies the highest level of security with lowest cost performance comparing with existing mutual authentication protocol for RFID systems.

## 7. Acknowledgement

The author would like to express his gratitude to all members of Computer Science and Information College, Aljouf University for their support.

## References

[1] C. Zhang, N. Ning, W. Zhang, and H. Mu, "A mutual authentication security RFID protocol based on time stamp," Journal of Computers, Vol. 28, No. 2, pp. 223-229, 2017.

[2] E. Ahmed, E. Shaaban, M. Hashem, "Lightweight mutual authentication protocol for low cost RFID tags," International journal of network security & its applications (IJNSA), Vol. 2, No. 2, pp. 27-37, 2010.

[3] G. Jin, B. Li, J. Mou, P. Li and X. Zhao, "A secure mutual authentication protocol to maintain synchrony conforming to EPC Gen2V2 standard," International conference on materials, information, mechanical, electronic and computer engineering, USA, pp. 165-171. 2016.

[4] I. Jeon, E. Yoon, "A new ultra-lightweight RFID authentication protocol using merge and separation operations," International Journal of Math. Analysis, Vol. 7, No. 52, pp. 2583-2593, 2013.

[5] J. Shen, H. Tan, Y. Wang, J. Wang, "A novel RFID authentication protocol for multiple readers and tag groups," Advanced Science and Technology Letters, Vol. 50, pp. 55-61, 2014.

[6] K. Hong-yan, "Design of a Mutual Authentication Protocol for RFID Based on ECC," The Open Automation and Control Systems Journal, Vol. 7, pp. 1532-1536, 2015.

[7] M. Habibi, M. Gardeshi, M. Alaghband, "Cryptanalysis of two mutual authentication protocols for low-cost RFID," International Journal of Distributed and Parallel Systems, Vol. 2, No. 1, pp. 103-114, 2011.

[8] M. Morshed, A. Atkins, H. Yu, "An efficient and secure authentication protocol for RFID systems," International Journal of Automation and Computing, Vol. 9, No. 3, pp. 257-265, 2012.

[9] N. Chikouche, F. Cherif, P. Cayrel, M. Benmohammed, "Improved RFID authentication protocol based on randomized McEliece cryptosystem," International journal of network security, Vol. 17, No. 4, pp. 413-422, 2015.

[10] N. Chikouche, F. Cherif, M. Benmohammed, "An authentication protocol based on combined RFID-biometric System," International journal of advanced computer science and applications, Vol. 3, No. 4, pp. 62-67, 2012.

[11] N. Lo, K. Yeh, "Mutual RFID authentication scheme for resource-constrained tags," Journal of information science and engineering, Vol. 26, No. 5, pp. 1875-1889. 2010.

[12] N. Soni, S. Sharma, "A survey of RFID authentication protocols & encryption techniques," International journal of advancements in research & technology, Vol. 2, No. 5, pp. 465-468, 2013.

[13] N. Soni, S. Sharma, "An RFID mutual authentication protocol using ECIES," International journal of emerging technology and advanced engineering, Vol. 4, No. 1, pp. 380-384, 2014.

[14] O. Omolola, O. Osunade, "Secure mutual authentication protocol for Low-cost RFID," International journal of computer and information technology, Vol. 5, No. 1, pp. 25-32, 2016.

[15] P. Lopez, J. Castro, J. Tapiador AND A. Ribagorda, "An ultra-light authentication protocol resistant to passive attacks under the Gen-2 specification," Journal of information science and engineering, Vol. 25, No. 1, pp. 33-57, 2009.

[16] R. Chowdhury, M. Ansary, "A secured mutual authentication protocol for RFID system," international journal of scientific & technology research, Vol. 3, No. 5, pp. 52-56, 2014.

[17] S. Abughazalah, K. Markantonakis, K. Mayes "A formally verified mutual authentication protocol for low-cost RFID tags," International Journal of RFID Security and Cryptography, Vol. 3, No. 2, pp. 156-169, 2015.

[18] S. Anand, B. Santhi, "A review on efficient mutual authentication RFID system security analysis," International Journal of Engineering and Technology, Vol. 5, No. 1, pp. 381-384, 2013.

[19] S. Nashwan, B. Alshammari, "Formal analysis of MCAP protocol against replay attack," British Journal of Mathematics & Computer Science, Vol. 22, No. 1, pp. 1-14, may 2017.

[20] S. Nashwan, B. Alshammari, "Mutual Chain authentication protocol for SPAN transactions in Saudi Arabian banking," International journal of computer and communication engineering, Vol. 3, No. 5, pp. 326- 333, 2014.

[21] S. Rostampour, M. Namin, "A new efficient and secure mutual authentication protocol for RFID systems," IEEE European Modelling Symposium, pp. 383-388, 2015.

[22] Z. Cheng, Y. Liu, C. Chang and S. Chang, "Advanced constantly updated RFID access control protocol using challenge-response and indefinite-index," International journal of innovative computing, information and control, Vol. 8, No. 12, pp. 8341-8354, 2012.

[23] S. Nashwan, "SAK-AKA: A Secure Anonymity Key of Authentication and Key Agreement protocol for LTE network," The International Arab Journal of Information Technology, Vol. 14, No. 5, pp. 790-801, 2017.

[24] U. Mujahid, M. Najam-ul-islam, "Pitfalls in Ultralightweight RFID Authentication Protocol," International Journal of Communication Networks and Information Security, Vol. 7, No. 3, pp. 169-176, 2015.

[25] E. Taqieddin, "On the Improper Use of CRC for Cryptographic Purposes in RFID Mutual Authentication Protocols," International Journal of Communication Networks and Information Security, Vol. 9, No. 2, pp. 230-240, 2017.

[26] U. Mujahid, M. Najam-ul-islam, "Ultralightweight Cryptography for Passive RFID Systems," International Journal of Communication Networks and Information Security, Vol. 6, No. 3, pp. 173-181, 2014.

# Appendix A

**Table 5.** Vectors of operations of authentication entities.

| Authentication protocols | Vo of (tag) | Vo of (reader) | Vo of (back-end server) |
|---|---|---|---|
| **[1]** | [1, 0, 0, n+3, n+2, n, 0, 2] | [0, 1, 0, 2m+1, 3m, m, 0, 1] | [1, 0, 0, 2m+n+1, m+n+1, m+n+1, 0, 2} |
| **[14]** | [6, 0, 0, 6, 0, 2, 0, 6] | [1, 0, 0, 5, 4, 3, 0, 2] | [6, 0, 0, 4n+4m+9, 2n, 2m+2n+1, 0, 4m+2] |
| **[16]** | [1, 0, 1, 3, 2, 1, 0, 2] | [1, 0, 0, 0, 0, 0, 0, 0] | [0, 1, 2m, 2m+2, 6m+3, 2m, 0, 2] |
| **[17]** | [1, 0, 5, 2, 5, 1, 0, 7] | [1, 0, 2m, 0, 2m+1, 2m, 0, 2] | [0, 0, 4, 1, 4, 1, 0, 8] |
| **SE-H** | [0, 1, 0, n+8, n+4, n+1, 2, 4] | [0, 1, 0, 6, 3, 3, 1, 6] | [0, 0, 0, n+9, n+4, n+2, 2, 7] |

**Table 6.** Vectors cost of authentication entities.

| Authentication protocols | Vc of (tag) | Vc of (reader) | Vc of (back-end server) |
|---|---|---|---|
| **[1]** | [2, 0, 0, n+3, 3n+6, n, 0, 2] | [0, 2, 0, 2m+1, 3m, m, 0, 1] | [2, 0, 0, 2m+n+1, 3m+3n+3, m+n+1, 0, 2} |
| **[14]** | [12, 0, 0, 6, 0, 2, 0, 6] | [2, 0, 0, 5, 12, 2, 0, 2] | [12, 0, 0, 4n+4m+9, 6n, 2m+2n+1, 0, 4m+2] |
| **[16]** | [2, 0, 1, 3, 6, 1, 0, 2] | [2 0, 0, 0, 0, 0, 0, 0] | [0, 2, 2m, 2m+2, 6m+3, 2m, 0, 2] |
| **[17]** | [2, 0, 5, 2, 15, 1, 0, 7] | [2, 0, 2m, 0, 6m+3, 2m, 0, 2] | [0, 0, 4, 1, 12, 2, 0, 8] |
| **SE-H** | [0, 2, 0, n+8, 3n+12, n+1, 6, 4] | [0, 2, 0, 6, 9, 3, 3, 6] | [0, 0, 0, n+9, 3n+12, n+2, 6, 7] |

**Table7.** The total vector cost of each authentication entities.

| Authentication protocols | Vct (t) | Vct (r) | Vct(s) |
|---|---|---|---|
| **[1]** | 5n+13 units | 6m+4 units | 5n+6m+9 units |
| **[14]** | 26 units | 23 units | 12n + 10m + 23 units |
| **[16]** | 15 units | 2 units | 12m+9 units |
| **[17]** | 32 units | 10m+7 units | 27 units |
| **SE-H** | 5n+33 units | 29 units | 5n+36 units |