On the Security of Permutation Based Authentication Protocols for Internet of Things Applications: The Case of Huang et al.’s Protocol

Samad Rostampour a,*, Nasour Bagheri b, Mehdi Hosseinzadeh c, Ahmad Khademzadeh d

aDepartment of Computer Engineering, Ahvaz Branch, Islamic Azad University, Iran.
bDepartment of Electrical Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran.
cIran University of Medical Sciences, Tehran, Iran.
dIran Telecommunication Research Center, Tehran, Iran.

ARTICLE INFO
Article history:
Received: 20 January 2017
Revised: 11 November 2017
Accepted: 12 December 2017
Published Online: 01 April 2018

Keywords:

ABSTRACT
The Internet of Things (IoT) is a new technology, which enables objects to exchange data via the Internet. Authentication process is a method to prevent an unauthorized access to the IoT systems. The using of bit-wise functions such as XOR, Shift and Rotation could decrease the cost of authentication protocols. On the other hand, the simple operations usually could not provide an acceptable security level. Therefore, the researchers try to improve the security level by creating new permutation functions. In this paper, we evaluate some permutation functions and analyze a protocol which recently has been proposed by Huang et al. We prove that their protocol is vulnerable to the disclosure and the impersonation attacks and an adversary can clone a valid tag easily. The complexity of the proposed attack is low and attack method works efficiently for the secret keys and ID numbers with variable length.

© 2016 JComSec. All rights reserved.

1 Introduction
The Internet of Thing (IoT) is a new concept in the world of information technology. In short, it can be said that the IoT is a modern technology wherein each item (human, animal, object, etc.) is able to exchange data in communication networks [1–4]. In an IoT data exchange process, a unique identification number and an IP address are assigned to each item and its data is sent to a back-end server for processing. This data is then accessible through different devices such as smart phones, tablets, personal computers, etc. The IoT consists of various technologies and services, and concentrates on establishing a connection between appliances that are used every day.

The two main features of an object are communication and identity. In terms of communication, there are a lot of technologies to gather an object’s data such as Radio Frequency Identification (RFID), Bluetooth Low Energy (BLE), and Wi-Fi [5–10]. In an RFID system a tag is assigned to each item and its data is sent to a back-end server for processing. The tag’s information in a wireless environment and then sends it to a back-end database server.

The other main feature is identity which is a factor that identifies an object. An authentication protocol is a process to evaluate identity and causes to validate the objects, fortify the system and prevent it against an unauthorized access. Given that the tag’s weight (The tag’s weight means the number of gate equivalents (GE) [11]) and cost are directly linked, system designers tend to desire lightweight and low-cost...
tags. On the other hand, the use of sophisticated circuits to improve the security level could lead to an increase of accommodated gates on a tag. Therefore, the making a trade-off between the security level and the tag’s weight is necessary. In terms of the tag’s weight, IoT authentication protocols are divided into three categories: Ultra-lightweight, Lightweight, and Non-lightweight.

- Ultra-lightweight: in these kinds of protocols only simple bit-wise functions such as AND, XOR, and Shift are used and the number of gates on the tag is very low.
- Lightweight: in lightweight protocols, the designers can also use one-way or two-way cryptography methods such as Pseudo Random Number Generator (PRNG), Cycle Redundancy Check (CRC) functions, and Data Encryption Standard (DES). These kinds of protocols have resource limitations, and the approximate 3000 gates are accommodated on the tag for the security module [12, 13]. Lightweight protocols are suitable for low-cost tags and can be implemented based on international standards.
- Non-lightweight: in this category, encryption methods such as Hash functions and Elliptic Curve cryptography (ECC) are used to increase the security level. Although these functions can increase the complexity and confidentiality of exchanged messages, they also cause an increase in the number of required gates on the tag beyond that of the lightweight protocols for an equal level of security.

Since the context of this article is about ultra-lightweight protocols, we describe and analyze some of the current presented protocols of this category. One of the first and famous ultra-lightweight protocols is SASI (Strong Authentication and Strong Integrity) [14]. In the SASI protocol, the authors attempted to design a protocol based on bit-wise functions such as XOR and Rotate for low-cost tags. Although SASI had some vulnerabilities and could not satisfy an acceptable level of security, it was employed as the basic theory for ultra-lightweight protocols.

In addition, in recent years many ultra-lightweight protocols have been proposed and most of them had security vulnerability and were not resistant against IoT threats. Because the use of bit-wise functions decreases the complexity of the protocol and increases the probability of a successful attack [15, 19]. Hence, the researchers tried to create new permutation functions based on bit-wise operations.

For example, RAPP (RFID Authentication Protocol with Permutation) is an ultra-lightweight protocol that used a permutation function to improve the security level [20]. The function used the hamming weight and changed the place of the bits. Therefore, it had a simple hardware circuit. Even though RAPP was affordable and low-power, there have been three attack methods presented and proven that RAPP is not secure against traceability, secret disclosure, and desynchronization attacks [21].

Fan et al. proposed another Ultra-Lightweight RFID Mutual Authentication Protocol with Cache (UL-RMAPC) for the IoT applications [22]. The authors created a permutation function based on rotate and reverse operations (RR function). They claimed that the proposed protocol is robust against well-known RFID attacks and they have solved the security problems of previous protocols such as SASI and Gossamer.

In addition, Tewari and Gupta proposed an ultra-lightweight mutual authentication protocol in IoT environments for RFID tags [23]. Their protocol is very efficient since it only utilizes bit-wise operations such as Rotate and XOR. The authors also provided a detailed analysis to demonstrate this protocol is secure against various attacks. Their protocol aims to provide a secure communication with least cost in both storage and computation. Although Tewari and Gupta tried to provide a low-cost and ultra-lightweight protocol based on bit-wise and hamming weight, they could not provide an acceptable security level by these kinds of functions. Because Wang et al. proved that Tewari and Gupta protocol is vulnerable to the key disclosure attack and they could obtain the secret information of the tags [24].

In this direction, Huang et al. also presented an authentication protocol to enhance the security level and reduce the tags’ cost in IoT healthcare applications [25]. They used a permutation function called Con, claiming it to be more lightweight than other encryption functions such as hash functions and elliptic curve based crypto-systems (ECC). In this paper, we argue that Huang et al.’s protocol is not resistant against the disclosure and the impersonation attacks and an adversary can disclose the stored secret parameters of the tag, and clone it.

In the rest of this paper, we review Huang et al.’s protocol in Section 2. Then, we analyze this protocol and implement attacks on it in Section 3. Ultimately, in Section 4 we will conclude.

### 2 Huang et al.’s Protocol

Huang et al. [25] presented a new lightweight grouping proof protocol for medicine distribution applications as it is shown in Figure 1 and claimed that their protocol guarantees the security and privacy in such
applications. When two or more tags participate in an authentication process and the reader has to identify them simultaneously, it is called grouping proof. In the proposed protocol, the reader tries to authenticate two tags and prove the tags are valid. Because the reader has to create a unique message for each tag and provide the integrity in tags’ messages, therefore presenting a secure grouping proof protocol that can satisfy all security features seems difficult. Their protocol is based on ISO-14443 standard for passive tags and uses a permutation function Con. In this section, we review Con function and Huang et al.’s protocol.

2.1 Con Function

To reduce the cost and computational load of passive tags, Huang et al. utilize a new permutation function that has two input parameters and one output, Con (X,Y). It rearranges the first input (X) based on the hamming weight of the second input (Y) that just uses XOR and Rot functions to calculate the output as follows:

1. Receives two inputs X and Y.
   a. X = x1x2...xi; where xi ∈ {0,1} and i = 1;2;...;l;
   b. Y = y1y2...yi; where yj ∈ {0,1} and j = 1;2;...;l;
2. Calculates the hamming weight of Y that is called (m) that is the number of ones in Y, such that yk1 = yk2 = ... = ykm = 1 and ykm+1 = ykm+2 = ... = ykl = 0; is discontinuous, i.e. yk1 just presents one of all the yj, that equals to 0 or 1.
3. Divides ones to even-group and odd-group and rearranges X based on the position of ones in Y to compute the output of Con as below: Con = xk1xk2...xkm+2xkm+3...xkl; where m = 4, as a result Con(X,Y) = 10101010

In addition, Huang et al.’s uses Rot(a,b) function. The first parameter of the function is the main parameter that we want to rotate it and the second parameter indicates the number of rotations. Rot function rotates the first input h times where h is the hamming weight of the second parameter.

2.2 Huang et al.’s Method

In this section, we explain how Huang et al.’s protocol works based on Con function. The used notations of the proposed protocol have been listed in Table 1. The protocol has two phases: Initialization and Authentication.

### Table 1. Notations Used in the Huang et al.'s Protocol

<table>
<thead>
<tr>
<th>Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>First input of Con function</td>
</tr>
<tr>
<td>Y</td>
<td>Second input of Con function</td>
</tr>
<tr>
<td>TA and TB</td>
<td>The participating tags in the authentication process</td>
</tr>
<tr>
<td>T</td>
<td>Output of Con function</td>
</tr>
<tr>
<td>m</td>
<td>Hamming weight</td>
</tr>
<tr>
<td>r</td>
<td>A random number</td>
</tr>
<tr>
<td>l</td>
<td>Data length bit</td>
</tr>
<tr>
<td>k</td>
<td>Shared secret bit of each tag</td>
</tr>
<tr>
<td>ID</td>
<td>ID number of each tag</td>
</tr>
<tr>
<td>Rot(a,b)</td>
<td>Rotate a based on hamming weight of b</td>
</tr>
<tr>
<td>⊕</td>
<td>XOR function</td>
</tr>
</tbody>
</table>

(1) Initialization phase
In this phase, the shared secret key k and the identification number ID are stored in the tags.

(2) Implementation Phase

(a) The reader generates two random numbers r1 and r2 and sends them with a request message to TA and TB.
(b) Upon receiving r1, TA generates a random number ra, calculates A = Con(Rot(ra; r1) ⊕ IDA; rA) and sends A to the reader.
(c) After receiving A and ra, the reader calculates VRa = Con(ra; IDA) and transmits it to TB.
(d) When TB receives VRa, generates rb randomly and computes B = Con(Rot(VRB; rB) ⊕ IDB; rb ⊕ r2) and VRB = Con(rot(B; r2); k2). Then, TB transfers B, rB and VB to the reader.
(e) Afterwards, the reader computes VRB = Con(rB; IDB) and transmits it to TA.
(f) TA calculates VA = Con(VRB; Rot(ra; k1)) and responds to the reader.
(g) The reader calculates proof PAB with ra, rB, A, B, VA and VB and transfers it to the server.
(h) Upon receiving PAB, the server computes PAB using the information located in the database. If PAB = PAB, the authentication process is successful; otherwise, it is aborted.
3 Security Analysis of Huang et al.’s Protocol

Huang et al. claimed that their protocol is robust against well-known RFID attacks based on the following reasons:

- The protocol is based on ISO-14443 standard and the user can adjust the transmission power of the tag under 4 inches. Therefore, the adversary cannot eavesdrop or intercept the information in the communication channel.
- Each tag has a shared secret key that the information is encrypted by it. Because the secret key is not transferred in the channel, the adversary cannot access to it. Accordingly, the protocol is resistant against the disclosure attack.
- Information is computed by two tags, which are related to each other. Hence, if one of them is absent, the authentication process is not completed and adversary needs both of them simultaneously.

In this section, we prove that mentioned reasons cannot guarantee the security of the proposed protocol and present an attack model to penetrate to the system. The attack method does not depend on power transmission of the tag. Given a valid tag and an illegitimate reader, the adversary can send a request to the tag and collects its responses. Afterwards, she processes the stored tag responses and obtains the secret values.

In addition, a user of the system can obtain the secret values of tags. For instance, one can put a sniffer between a tag and a reader and eavesdrops the messages.

The aim of this attack is to disclose the secret parameters of the tag. In each tag, there are two stored parameters, secret key \((k)\) and ID-number \((ID)\). The security method of the Huang et al.’s protocol is based on a permutation function that is called \(Con\) that was explained in Section 2. \(Con\) function consists of three parameters \(X\) and \(Y\) as input and \(T\) as output, \(T = Con(X, Y)\). We show that if the adversary has just two parameters of \(Con\) function, can also get the third parameter. In this attack, we assume that the adversary has an illegitimate reader and sends messages to the tag and collects its responses. In two separate processes, we will disclose \(k\) and ID-number.

### 3.1 ID Disclosure Attack

Given, \(Y\), in \(T = Con(X, Y)\) function, the mapping from \(T\) to \(X\) is an invertible permutation. Hence, given \(Y\) and \(T\), the adversary can compute \(X\) easily, because \(X\) is rearranged based on \(Y\). We explain ID disclosure attack as follow, which is shown in Figure 3:

1. The adversary divides ones of \(Y\) to two groups, odd (gray cells) and even (blue cells). Odd ones imply left side bits of \(T\) and even ones point to right side bits as it is shown in Figure 3.
2. Puts the first left bit of \(T\) in the position of the first one of \(Y\).
3. Puts the first right bit of \(T\) in the position of...
the second one of Y. 

(4) She repeats steps 2 and 3, (m) times for odd group and even group respectively.

(5) There are (l − m) zeros in Y. The adversary arranges remaining bits of T (red cells) based of zeros of Y. As it is shown in Figure 3c, the adversary selects the first bit after even group (gray cells) and puts it in the position of the first left zero bit of Y. She repeats this operation for (l − m) times.

Hence, to disclose ID_A and ID_B the adversary impersonates a legitimate reader and does as follows:

(1) The adversary generates a random number r_1 and sends it with a request message to T_A.

(2) Upon receiving r_1, T_A generates a random number r_A, calculates \( A = Con(\text{Rot}(r_A; r_1) \oplus ID_A; r_A) \) and sends A to the reader (impersonated by the adversary).

(3) After receiving A and r_A, adversary can determine \( \text{Rot}(r_A; r_1) \oplus ID_A \).

(4) Adversary determines ID_A as \( ID_A = \text{Rot}(r_A; r_1) \oplus ID_A \oplus \text{Rot}(r_A; r_1) \).

(5) The Adversary also calculates \( V_{RA} = Con(r_A; ID_A) \) and transmits it to T_B.

(6) When T_B receives \( V_{RA} \), generates \( r_B \) randomly and computes \( B = \text{Con}(\text{Rot}(V_{RA}; r_B) \oplus ID_B; r_B) \) and \( V_{RB} = \text{Con}(\text{rot}(B; r_2); k_2) \).

Then, T_B transfers B, r_B and V_B to the reader.

(7) Given B, \( V_{RA}, r_B \) and r_2 the adversary can determine \( \text{Rot}(V_{RA}; r_B) \oplus ID_B \) and disclose ID_B as \( ID_B = \text{Rot}(V_{RA}; r_B) \oplus ID_B \oplus \text{Rot}(V_{RA}; r_B) \).

The success probability of the presented attack is ‘1’ while the attack complexity is only impersonating the legitimate reader in a single session between the tags and the reader. On the other hand, given that ID-number is a static value for each tag, the adversary can use the extracted values to trace the tag holder at any time. Since the attack procedure is trivial, we omit more details. The success probability of the adversary to determine the tag holder correctly would be \( 1 - 2^{-n} \), where \( n \) is the bit length of ID, and the complexity for each attack session would be just impersonating the reader once.

3.2 Secret Key Disclosure Attack

In Huang et al.’s protocol, a transferred value from the tag to the reader is \( V_{RB} = Con(\text{Rot}(B; r_2); k_2) \), where \( k_2 \) is a static value for each tag and is not updated. On the other hand, assuming that the adversary follows the attack procedure that presented in Section 3.1 we can assume that the adversary knows B and r_2 and can calculate \( \text{Rot}(B; r_2) \). Thus, extracting \( k_2 \) is reduced to extract \( Y \) from \( T = Con(X; Y) \) given \( T \) and X. Unlike the mapping T to X, for fixed Y, which is an invertible permutation, the mapping T to \( Y \), for fixed \( X \), is not a permutation necessarily. However, given several pairs \( T^i = Con(X^i; Y) \), the adversary can determine Y uniquely. Hence, to determine \( k_2 \) in Huang et al.’s protocol, the adversary impersonates the reader and initiates \( t \) sessions with the legitimate tags T_A and T_B. In each session \( i \), the adversary stores \( B^i, r_{B^i} \) and \( V_{RB^i} \) for \( 1 \leq i \leq t \), that are transferred from the adversary to the tag or from the reader to the tag in the \( i \)th session. To obtain the value of \( k_2 \), the adversary stores B, \( V_{RB} \) and \( r_B \) as Table 2 to find appropriate \( Y = k_2 \). In each row of Table 2 examples of the calculated values by the tag for each adversary’s query are shown. Given \( T^i = (Con(X^i; k_2)) \), for \( 1 \leq i \leq t \), where \( X^i = \text{Rot}(B^i; r_{B^i}) \) and assuming that \( (Z_j) \) determines the \( j \)th bit of string \( Z \), \( (Z_0) \) is the most left bit, the adversary can extract \( k_2 \) bit by bit as follows:

- If \( (T^i)_0 = (X^i)_0 \) for \( 1 \leq i \leq t \), set \( (k_2)_0 = 1 \) otherwise \( (k_2)_0 = 0 \).
- Given, \( (k_2)_0 \), the adversary determines \( (k_2)_1 \) in the next step. There could be two cases for \( (k_2)_0 \), i.e. \( (k_2)_0 = 0 \) and \( (k_2)_0 = 1 \) respectively and adversary does as follows in each case:
  - When \( (k_2)_0 = 0 \), if \( (T^i)_1 = (X^i)_1 \) for \( 1 \leq i \leq t \), set \( (k_2)_1 = 1 \) otherwise \( (k_2)_0 = 1 \).
  - When \( (k_2)_0 = 1 \), if \( (T^i)_1 = (X^i)_n-1 \) for \( 1 \leq i \leq t \), set \( (k_2)_1 = 1 \) otherwise \( (k_2)_1 = 1 \).
Table 2: Examples of Con Function Based on Different Inputs

<table>
<thead>
<tr>
<th>X = Rot(B; r2)</th>
<th>Y = k</th>
<th>T = Con(X, Y) = V_Rb</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 0 1 0 0 0</td>
<td>1 0 0 1 1 1 1 1</td>
<td>1 0 1 0 1 0 0 1</td>
</tr>
<tr>
<td>0 1 0 0 1 1 1</td>
<td>1 0 0 1 1 0 1 1</td>
<td>1 1 1 0 0 1 0 1</td>
</tr>
<tr>
<td>1 0 1 0 0 1 0</td>
<td>1 0 0 0 1 1 1 1</td>
<td>1 1 1 0 0 1 0 1</td>
</tr>
</tbody>
</table>

- Following this procedure, the adversary can determine whole \( k_2 \) bit by bit. For example assuming that the adversary already has determined the first \( r \) bits of \( k_2 \) and the hamming weight of these \( r \) bits is \( m \). To extract \((r+1)^{th}\)-bit of \( k_2 \), \( i.e. \ (k_2)_r \), the adversary does as follows:
  - When \( m \) is even, if \((T^r)_{m+1} = (X^r)_{m+1} \) for \( 1 \leq i \leq t \), set \((k_2)_{m+1} = 1 \) otherwise \((k_2)_{m+1} = 0 \).
  - When \( m \) is odd, if \((T^r)_{m+1} = (X^r)_{m+1} \) for \( 1 \leq i \leq t \), set \((k_2)_{m+1} = 1 \) otherwise \((k_2)_{m+1} = 0 \).

Following the above attack, the adversary can determine whole bits of \( k_2 \). To determine the success probability of the given attack, omit the case where \( k_2 = \{0\}^n \), a \( n \)-bit string for which all bits are zero, where the above-mentioned attack returns \( k_2 = \{1\}^n \), and can be fixed easily. For \((k_2)_{0} \) the attack may return a wrong guess if there exist \( 1 \leq j \leq n - 1 \) such that \((X^r)_0 = (X^r)_j \) which happens with the probability \( 1 - (1 - 2^{t-j})^{n-1-j} \). Assuming that, the adversary already has determined the first \( r \) bits of \( k_2 \) correctly, it may return a wrong guess for \((k_2)_{r} \) if there exist \( r + 1 \leq j \leq n - 1 \) such that \((X^r)_{j} = (X^r)_j \) which happens with the probability of \( 1 - (1 - 2^{t-j})^{n-1-r} \). Hence the total success probability of the given attack is lower bounded by \( \Pi_{j=0}^{r} (1 - 2^{t-j})^{n-1-j} \).

Another approach to determine \( Y = k_2 \) is as follows:

Step 1  - The adversary divides \( Y \) into 4-bit blocks and analyzes the first left-side block as it is shown in Figure 4.
  - Based on \( X \) and \( T \), she assumes the value of the block from \( (0000) \) to \( (1111) \) and checks that which of 16 possible states can be true as it is shown in Table 3. In each row of Table 3, the adversary finds the true possible states for \( k_2 \). For example, in step 3.2 for \( X = 1 1 0 1 0 0 1 0 \) and \( T = 1 0 1 0 1 0 1 0 \), she analyzes the first 4-bit block of \( Y \) and finds the seven values can be true possible state. In step 2, she changes \( X \) and \( T \), repeats the comparisons and continues this operation until finding the appropriate states.
  - For instance, as it is shown in Figure 5 (0001) can be true because the forth bit of \( X \) equals to the first bit of \( T \) and there are not any dissimilar bits.

- On the other hand, as it is shown in Figure 6 (0010) is not true, because the third bit of \( X \) is zero, though the first bit of \( T \) is one.
- The adversary continues this comparison for all numbers to (1111) and understands 0100; 0101; 0110; 1000; 1001; 1100; 1110 can be true and stores acceptable numbers. It must be noted that (0000) will be checked at the end because this number could be true for each state. If any possible state is not found, the adversary checks (0000) and merge it to the next 4-bit block.

Step 2  - The adversary checks remaining numbers by the second row of Table 3 that \( X = (01001110) \) and \( T = (01100101) \).
  - Again, (0001) is acceptable but (0100) is not.

Step 3  - In step three, the adversary checks remaining number for new \( X \) and \( T \). Finally, she understands just (1000) can be true answer and other numbers are not acceptable.
Table 3. Steps of the Finding K and Reducing False Values in Each Step

<table>
<thead>
<tr>
<th>Step</th>
<th>X</th>
<th>T</th>
<th>True Possible States</th>
<th>False States</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 1 0 1 0 1 1 0</td>
<td>1 0 1 1 0 0 1</td>
<td>0001, 0100, 0101, 1000, 1001, 1100, 1110</td>
<td>0010, 0111, 1010, 1011, 1101, 1111</td>
</tr>
<tr>
<td>3</td>
<td>0 0 1 1 0 1 0 0</td>
<td>1 0 1 1 0 0 1</td>
<td>1000</td>
<td>0001, 1100</td>
</tr>
</tbody>
</table>

Hence, she repeats these steps for the next 4-bit block and checks all numbers based on (1000) in the first block for X and T in Table 2. Ultimately, she finds (0111) is the correct answer and Y = (1000111). The number of comparisons is 42 that it is much smaller than the number of comparisons in the comprehensive search for 8-bit data ($2^8$). Therefore, in Con function, if the second parameter be fixed, the adversary can obtain it and this function is insecure. We evaluated this attack method for different lengths of parameters such as $k$ from 16-bit to 128-bit. For each length, we used five different $k$ and calculated the average number of comparisons. For 16-bit, 32-bit, 64-bit and 128-bit length, the average number of comparisons consequently were 180, 330, 780 and 1624 as it is shown in Figure 7. The complexity of this attack is less than using the comprehensive search for finding $k$, though the comprehensive search is not practicable for data length more than 32-bit.

3.3 Permanent Impersonation Attack

For permanent impersonation attack, the adversary needs to obtain all stored parameters in the tag. Because just two parameters ($k, ID$) are stored in the tag, if the adversary computes them, she can clone the tag.

In Section 3, the adversary obtained $k$ and ID of the tag; therefore, she can make a hard copy and clone $T_B$. For $T_A$, the adversary repeats the proposed attack in Section 3 and computes all stored parameters. Hence, the disclosure attack causes to the permanent impersonation attack easily and adversary impersonates all the tags in the system.

The using of only one secret parameter in each variable that is created by the tag is the main reason of the weakness of the protocol. To generate $B$, Because $T_D$ just uses $ID_B$ as a secret parameter, the attacker can compromise the protocol. If we use both $ID_B$ and $k_2$ to generate tags’ messages, the security level of the protocol will be increased and this method will not be applicable. Generally, since in ultra-lightweight protocols only simple operations such as XOR, Rotate, Shift are utilized, providing an acceptable level of the security is very difficult. The researchers should generate messages carefully and obsessively and increase the complexity of messages. Therefore, to present a
secure and ultra-lightweight protocol, it must be analyzed based on formal and informal security method completely.

4 Conclusion

In this paper, we discussed about security of authentication protocols which have used permutation function in IoT technology. We reviewed some protocols and analyzed one of them that recently has been presented by Huang et al. for medical healthcare applications. The authors presented an authentication protocol and utilized a new permutation function that is called Con for rearranging data before sending. The Con function had two input parameters and rearranged the first input based on hamming weight of the second input. We proved that Con function was vulnerable to the disclosure and the impersonation attacks, and the attack method is efficient for data with any length. Therefore, it proves that the permutation function is not suitable solution to improve the security of ultra-lightweight and lightweight protocols in IoT systems and researchers have to improve the security level of the protocols that use permutation functions.

References


Samad Rostampour is a Lecturer and Faculty member of Department of Computer in Ahvaz Branch, Islamic Azad University. His research interests include Computer Networks, Information Security and Cryptography and System Design.

Nasour Bagheri is an assistant professor at Electrical Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran. He is the author of over 50 articles in information security and cryptography. Homepage of the author is available at: http://n-bagheri.srttu.ir.

Mehdi Hosseinzadeh is an assistant professor at Department of Computer Engineering, Science and Research branch, Islamic Azad University, Tehran, Iran. His research interests are Computer Arithmetic with emphasis on Residue Number System, Cryptography, Network Security and E-Commerce.

Ahmad Khademzadeh is a professor at Iran Telecommunication Research Center, Tehran, Iran. His research interests are Cryptography, Network Security and Telecommunication.