Cryptanalysis of an Ultra lightweight Authentication Scheme based on Permutation Matrix Encryption for Internet of Vehicles

Morteza Adeli$^1$, Nasour Bagheri$^2$

$^1$ Department of Sciences, Shahid Rajaee Teacher Training University, Iran, adeli.morteza@gmail.com
$^2$ Electrical Engineering Department, Shahid Rajaee Teacher Training University, Iran, NBagheri@srttu.edu

Abstract. Internet of Things (IoT) has various applications such as healthcare, supply chain, agriculture, etc. Using the Internet of Vehicles (IoV) to control traffic of the cities is one of the IoT applications to construct smart cities. Recently Fan et al. proposed an authentication protocol to provide security of the IoV networks. They claimed that their scheme is secure and can resist against various known attacks. In this paper, we analyze more deeply the proposed scheme and show that their scheme is vulnerable against disclosure and desynchronization attacks. In disclosure attack, we disclose unique identification of the tag $ID$, secret key $S$, encryption matrix $M_2$ and half rows of encryption matrix $M_1$. Furthermore, we proposed an improved authentication scheme based on Maximum Distance Separable (MDS) matrices that is resistance against various attacks while maintaining low computational cost.

Keywords: IoV; security analysis; matrix encryption; MDS matrix

1 Introduction

Internet of Things (IoT) helps us to construct future smart cities. In a smart city, IoT can be used to solve some urban problems such as traffic, air pollution, etc. The goal of a smart city is a tool for improving quality of life. Improving traffic flow reduces air pollution and all while helping us to have a healthy environment. Internet of Vehicles (IoV) is a complex integrated network system that plays an important role in smart cities. An IoV network provides a base to connects different automotives together and to related authorities such as hospital and police in cities. Therefore, for example, the lane is cleared before emergency vehicles such as ambulances or fire engines reach the incident location. An IoV system contains three parts: Radio-frequency identification (RFID) tag, reader, and back-end server. RFID tags are small electronic chips with restricted computational power, memory limitation and low energy that connects to a vehicle to detect and send valuable information such as location, speed, user’s identity to transponder reader. Transponder readers have relatively more resources than
tags and in the IoV networks, display many roles such as pick up signals of environmental tags, their transformed information such as data encrypted in the tag, and their locations. Backend server usually has more powerful computation and storing ability than RFID readers and tags, therefore it stores the related information of tags and readers and calculates the computational processes when authenticates a tag or a reader. In this new emerging technology, the security of the IoV networks is a big challenge as devices or signals that contain important information of vehicle are attractive targets for the attacker to hold them and change it to disrupting the order of traffic of a city or trace path and time of who drives a vehicle. One of the best strategies to achieve data security is authentication before transmission of information in data communication protocols. Depend on network structure, security level requirement and restriction of computational power, until now, numerous authentication protocols have been proposed to respond to this demand. However, unfortunately, most of these protocols are not suitable to use in the IoV networks due to security weakness and/or operational requirements.

Our contribution: In this paper, we take a more detailed look at the Fan et al. scheme and show that the proposed scheme is vulnerable against secret disclosure attack. By using this attack, the attacker can reveal some security parameters of the tag such as $ID$, secret key $S$, encryption matrix $M_2$ and half rows of encryption matrix $M_1$. Furthermore, we demonstrate that the scheme can not resist against desynchronization attack. To overcome the vulnerabilities of the scheme, we propose an improved authentication scheme based on MDS matrices. We analyze the security aspects of our improved scheme through formal and informal analysis. Finally, we implement the improved scheme on FPGA using active HDL coding software tool and compare it with some relevant lightweight authentication schemes.

Paper’s organization: The rest of this paper is organized as follows: in section 2 we look briefly at some authentication protocols for IoV networks and mention their security challenges. Section 3 demonstrates the Fan et al. scheme and in the following, in section 4 we introduce our methods to perform disclosure and desynchronization attacks on the proposed scheme. We proposed an improved authentication scheme in section 5 that is resistance against various known attacks. Next, we evaluate our improved scheme through formal and informal proof in section 6. Implementation result of the improved scheme has been discussed in section 7. At the end, the conclusion of the paper is described in section 8.

2 Related Work

The key technology behind the success of the IoV systems is the security and privacy of network and one of the serious requirements of this issue is authentication protocols. Several authentication protocols for the IoV environment have been proposed by authors in the literature. In the year of 2017, Mohit et al. proposed a protocol for authentication and key agreement and claimed that it is
secure against various known attacks. They used lightweight operations such as hash function and XOR (⊕) to reduce the computational cost of their protocol. Later on, Li et al. [6] gave a detailed analysis of Mohit et al. and showed that it has some vulnerabilities such as the absence of session key, user duplication, and impersonation attacks. Wang et al. [10] focused mainly on preserving the privacy of a vehicular ad-hoc network (VANET), so they proposed a self-generated pseudo-identity to guarantee both privacy preservation and conditional traceability. In order for this scheme to operate efficiently, they used a lightweight symmetric encryption and message authentication code (MAC) generation for message signing and a fast MAC re-generation for verification. Liu et al. [7] have introduced an anonymous authentication protocol that provides secure communication between vehicles and roadside units. They use a certificateless short signature scheme combining a regional management strategy in their authentication protocol. Recently, Fan et al. [3] have proposed an authentication protocol for IoV environment and claimed that their scheme is secure against various attacks while has low computational cost. They use permutation matrices to provide security of transmitted data between tag and reader. Unfortunately, in this paper, we show that their scheme is vulnerable against disclosure attack and desynchronization attack.

3 Fan et al. scheme

In this section, we give a brief description of Fan et al. scheme. This scheme uses permutation matrices to encrypt and corresponding transposed matrix to decrypt messages transferred between a reader and a tag. The scheme contains two phases as following: (1) initialization and (2) authentication. Designers of the protocol assume that channel between the reader and the back-end server is secure. We represent the notations used in this article in Table 1 and a brief description of Fan et al. in Fig.1.

Definition: A permutation matrix is a square matrix obtained by permuting the rows of an identity matrix according to some permutation. So every row and column contains precisely a single "1" with "0"s everywhere else, and its inverse is its transpose.

Definition: The Unix timestamp is the number of seconds that have elapsed since 00:00:00 Coordinated Universal Time (UTC) on Thursday, 1 January 1970.

3.1 Initialization

- The back-end server shares secret key $S$ with entire legitimate tags.
- The legitimate reader and tag store corresponding permutation matrices $\{M_1^{-1}, M_2^{-1}\}$ and $\{M_1, M_2\}$ respectively.
- Reader is connected to the Internet to get a real-time Unix timestamp.

3.2 Authentication

- The reader generates random number $R$ and encodes the current network time as $T_1$ of size 128 bits where the first 64 bits are randomly filled and
The latter 64 bits represents the Unix timestamp. The first 64 bits of $T_1$ is generated randomly such that the weight of $T_1$ is always equal to 64. Then the reader computes $H_1 = T_1 \times M_1^{-1}$, $H_2 = (H_1 \oplus R) \times M_2^{-1}$ and sends $\{R, H_2\}$ to tag as challenge.

- Upon receiving, the tag computes $T_1$ by using inverse permutation matrix $M_1, M_2$ and compares $T_1$ with $T_0$ stored in tag. If the last 64 bits of $T_1$ are greater than $T_0$ no more than 48 h, the tag authenticates the reader and the tag updates the value of $T_0$ with $T_1$ and uses the updated $T_0$ to compare with next $T_1$ in next session. Then the tag computes $Y_1 = ID \times M_1$, $Y_2 = (Y_1 \oplus T_1) \times M_2$, $G = (S \oplus R) + ID$ and sends $\{Y_2, G\}$ to the reader.
- The reader computes $ID$ from $Y_2$ and sends $\{G, ID, R\}$ to back-end server through a secure channel. The back-end server computes $ID' = G - (S \oplus R)$ and compares two values $ID$ and $ID'$. If $ID = ID'$, the back-end server responses to the reader that the tag is legitimate.

### Table 1. Notation used in this paper

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>The 64-bits Unix timestamp the reader makes a request to the tag</td>
</tr>
<tr>
<td>$T_0$</td>
<td>Last successful authentication 128-bits encoded time stored in the tag</td>
</tr>
<tr>
<td>$T_1$</td>
<td>The 128-bits encoded time the reader makes a request to the tag</td>
</tr>
<tr>
<td>$R, R_1, R_2$</td>
<td>Random number generated by the reader</td>
</tr>
<tr>
<td>$S$</td>
<td>The secret key shared between back-end server and tag</td>
</tr>
<tr>
<td>$ID$</td>
<td>Unique identification of the specific tag</td>
</tr>
<tr>
<td>$M_1, M_2$</td>
<td>Permutation matrix used in tag</td>
</tr>
<tr>
<td>$M_1^{-1}, M_2^{-1}$</td>
<td>Permutation matrix used in reader</td>
</tr>
<tr>
<td>$M_{MDS}$</td>
<td>Maximum Distance Separable matrix</td>
</tr>
</tbody>
</table>

### 4 Security challenge of the scheme

**Definition:** Let $x_i$ denote the $i$-th bit of $X$, $R = (r_{127}r_{126}...r_0)$, $G = (g_{127}g_{126}...g_0)$ and $ID = (id_{127}id_{126}...id_0)$, where $r_i, g_i, id_i \in \{0, 1\}$.

**Definition:** An MDS (Maximum Distance Separable) matrix is a generator matrix of an MDS code and in cryptography is employed in block cipher and hash function as the diffusion layer.

1. **Disclosure attack:** In disclosure attack, an attacker reveals some private information of each parties. In this paper, we perform disclosure attack on Fan et al. protocol and reveal some private information of the scheme such as the $ID$, the secret key $S$, the encryption matrix $M_2$ and half rows of the encryption matrix $M_1$. In the first, we disclose the $ID$ of a tag and
<table>
<thead>
<tr>
<th>Phase</th>
<th>Back-end server</th>
<th>Reader</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Init.</td>
<td>$S$</td>
<td>$(M_1^{-1}, M_2^{-1}, T_1, R)$</td>
<td>$(M_1, M_2, S, ID)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$H_1 = (T_1) \times M_1^{-1}$</td>
<td>$H_2 = (H_1 \oplus R) \times M_2^{-1}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$R, H_2$</td>
<td>$H_1 = (H_2) \times M_2 \oplus R$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T_1 = (H_1) \times M_1$</td>
<td>$i$ if $T_1 &gt; T_0$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$Y_1 = (ID) \times M_1$</td>
<td>$Y_2 = (Y_1 \oplus T_1) \times M_2$</td>
</tr>
<tr>
<td>Aut.</td>
<td></td>
<td>$G = (S \oplus R) + ID$</td>
<td>$G, Y_2$ $\leftarrow Y_1 = (Y_2) \times M_2^{-1} \oplus T_1$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$ID = (Y_1) \times M_1^{-1}$</td>
<td>$ID' = G - R \oplus S$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$if ID' \nmid ID'$</td>
<td>$G, ID, R$ $\leftarrow G, Y_2$</td>
</tr>
</tbody>
</table>

Fig. 1. Fan’s et al. Scheme, where Init. and Auth. denote initialization and authentication retrospectively.

consequently the secret key $S$. In the second, we describe the man-in-the-middle attack that helps us reveal completely the permutation matrix $M_2$ and half rows of the permutation matrix $M_1$.

(a) We try to compute the $ID$ of a tag and the secret key $S$. For this purpose, we need to eavesdrop two sessions information, i.e. $G'$ and $G''$, and corresponding random numbers $R'$ and $R''$ that transferred between the reader and the tag. Let $X = S \oplus R$. We XOR($\oplus$) two sessions information $G'$ and $G''$ such as:

\[ G' \oplus G'' = (X' + ID) \oplus (X'' + ID) \]

We discuss about two sequential bits $id_{i+1}id_i$ such that satisfy the following relation:

\[ g_{i+1}g_i' \oplus g_i'' = (x_{i+1}' x_i' + id_{i+1}id_i) \oplus (x_i'' x_{i+1}'' + id_{i+1}id_i) \]

We explain our method by an example. let $g_{i+1}g_i' \oplus g_i'' = 1$. So there are four states for $x_{i+1}' x_i' \oplus x_{i+1}'' x_i''$. Let it is equal to “01”. So we have

- $g_{i+1}g_i = g_{i+1}g_i' \oplus g_i'' = 01$
- $x_{i+1}' x_i' \oplus x_{i+1}'' x_i'' = 01$
- $g_{i+1}g_i = g_{i+1}g_i' \oplus g_i'' = 01$
- $x_{i+1}' x_i' \oplus x_{i+1}'' x_i'' = 01$
- $x_{i+1}' x_i' \oplus x_{i+1}'' x_i'' = 01$
- $g_{i+1}g_i = g_{i+1}g_i' \oplus g_i'' = 01$
In Table 2 we demonstrate all possible values of two sequential bits \(id_{i+1}id_i\) based on relation between two sequential bits \(g_{i+1}g_i\) and \(r_{i+1}r_i\).

Using this table, we can discuss about \(i\)-th bit of the ID such as:

- When \(g_{i+1}g_i = 01\) and \(r_{i+1}r_i = 01\), then the \(i\)-th bit of the ID is equal to "0".
- When \(g_{i+1}g_i = 11\) and \(r_{i+1}r_i = 11\), then the \(i\)-th bit of the ID is equal to "0".
- When \(g_{i+1}g_i = 01\) and \(r_{i+1}r_i = 11\), then the \(i\)-th bit of the ID is equal to "1".
- When \(g_{i+1}g_i = 11\) and \(r_{i+1}r_i = 01\), then the \(i\)-th bit of the ID is equal to "1".

### Table 2. Two sequential bits \(id_{i+1}id_i\) based on \(g_{i+1}g_i\) and \(r_{i+1}r_i\)

<table>
<thead>
<tr>
<th>(g_{i+1}g_i)</th>
<th>(id_{i+1}id_i)</th>
<th>(00)</th>
<th>(01)</th>
<th>(10)</th>
<th>(11)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>(id_{i+1}id_i = 00, 01, 10, 11)</td>
<td>impossible</td>
<td>impossible</td>
<td>impossible</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>(id_{i+1}id_i = 00, 10)</td>
<td>impossible</td>
<td>(id_{i+1}id_i = 01, 11)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>impossible</td>
<td>impossible</td>
<td>(id_{i+1}id_i = 00, 01, 10, 11)</td>
<td>impossible</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>impossible</td>
<td>(id_{i+1}id_i = 01, 11)</td>
<td>impossible</td>
<td>(id_{i+1}id_i = 00, 10)</td>
<td></td>
</tr>
</tbody>
</table>

To perform this attack, first, we determine the LSB bit \(id_0\) and in consequence, second, third and so on, until the entire bits of the ID are determined. Two states in Table ?? are not desirable, when

i. \(g_{i+1}g_i = 00\) and \(r_{i+1}r_i = 00\)

ii. \(g_{i+1}g_i = 10\) and \(r_{i+1}r_i = 10\)

In these situations, we must choose another proper pair \((G', R')\) and replace it in relation \(G^0 \oplus G^0 = (X^0 + ID) \oplus (X^0 + ID)\). We keep the pair \((G', R')\) in our data base and if we require a new message in the next steps, we can use this pair again. One-third of these situations are not desirable, so in the worst case, determining each proper bit \(id_i\) requires a different pair \((G', R')\) and so we need \(n\) pairs such that \(n\) satisfies in equation \(\binom{\frac{n}{3}}{3} = 128 \times \frac{3}{2}\). Therefore this method requires at most twenty proper pairs \(\{(G', R')\}_{i=1}^{20}\) to determine all bits of the ID. It should be noted that if the \(id_i\) be equal to "1", in the next step, a carry bit is added to the value of \(r_{i+2}r_{i+1} \oplus r_{i+2}r_{i+1}'\). At the final step, when all bits of the ID was computed, we can compute the secret key \(S\) based...
on relation $S = (G - ID) \oplus R$. The algorithm of the attack is depicted in Algorithm 1.

<table>
<thead>
<tr>
<th>Algorithm 1: Disclosure attack algorithm to find $ID$ and secret key $S$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data:</strong> $(G', R')$ and $(G'', R'')$</td>
</tr>
<tr>
<td><strong>Result:</strong> Value of identity $ID$ and secret key $S$</td>
</tr>
<tr>
<td>1. $G = G' \oplus G''$ and $R = R' \oplus R''$;</td>
</tr>
<tr>
<td>2. for $i=1$ to 128 do</td>
</tr>
<tr>
<td>3. Select two sequential bits $g_{i+1}g_i$ of $G$ and $r_{i+1}r_i$ of $R$;</td>
</tr>
<tr>
<td>4. if $(g_{i+1}g_i = 11$ and $r_{i+1}r_i = 11)$ then</td>
</tr>
<tr>
<td>5. $id_i = 0$;</td>
</tr>
<tr>
<td>6. else if $(g_{i+1}g_i = 01$ and $r_{i+1}r_i = 01)$ then</td>
</tr>
<tr>
<td>7. $id_i = 0$;</td>
</tr>
<tr>
<td>8. else if $(g_{i+1}g_i = 01$ and $r_{i+1}r_i = 11)$ then</td>
</tr>
<tr>
<td>9. $id_i = 1$;</td>
</tr>
<tr>
<td>10. A carry bit add to calculate $r_{i+2}r_{i+1} \oplus r'<em>{i+2}r'</em>{i+1}$ in the next step;</td>
</tr>
<tr>
<td>11. else if $(g_{i+1}g_i = 11$ and $r_{i+1}r_i = 01)$ then</td>
</tr>
<tr>
<td>12. $id_i = 1$;</td>
</tr>
<tr>
<td>13. A carry bit add to calculate $r_{i+2}r_{i+1} \oplus r'<em>{i+2}r'</em>{i+1}$ in the next step;</td>
</tr>
<tr>
<td>14. else</td>
</tr>
<tr>
<td>15. Choose a new pair $(G', R')$;</td>
</tr>
<tr>
<td>16. Go to step one;</td>
</tr>
<tr>
<td>17. $S = (G - ID) \oplus R$;</td>
</tr>
</tbody>
</table>

(b) We use man-in-the-middle attack to compute permutation matrix $M_2^{-1}$. When the reader is connected with internet to get Unix timestamp $T$, we alter 64 bits Unix timestamp to "0" and send it to the reader. Therefor when the reader encodes 64-bits $T$ into 128-bits $T_1$, the first 64 bits of $T_1$ are filled by "1" and the later 64 bits are filled by "0" because the weight of the $T_1$ is always 64. We XOR($\oplus$) two such messages $H_2$ and $H_2'$, so we have:

- $H_2 \oplus H_2' = (H_1 \oplus R) \times M_2^{-1} \oplus (H_1 \oplus R') \times M_2^{-1} = (R \oplus R') \times M_2^{-1}$. Therefore the unknown value $H_1$ is eliminated, and the bit positions whose bits are different in $H_2$ and $H_2'$, are the bit positions whose bits are different in $R$ and $R'$ under permutation action $M_2^{-1}$. We collect the set $S$ contain of $n$ pair $\{(H_i', R_i')\}_{i=1}^n$. Let

- $A = H_2 \oplus H_2'$,
- $B = (R \oplus R') \times M_2^{-1}$.

We construct the set $U = \{(A^j, B^j)\}_{j=1}^{\binom{n}2}$, contains of pairwise XOR of all elements of the set $S$. It’s obviously that we can create new member
of the set \( U \) if we need a pair \( (A^1, B^j) \) that has "0" or "1" in specific bit position. We choose a member of set \( U \) like \( (A^1, B^1) \). For bit \( a_0 \in A^1 \), let \( a_0 = 0 \), there are at most 64 positions in \( B^1 \) whose positions are permuted under the permutation action \( M_2^{-1} \). Now we choose another member \( (A^2, B^2) \in U \) such that the number of "0" and "1" are maximum unbiased. Therefor some positions that we guessed for permuted position of \( a_0 \), in previous step, are removed. We continue this approach until all incorrect guesses are removed and first correct row of the permutation matrix \( M_2^{-1} \) was found. For detecting each row of the matrix \( M_2^{-1} \), in the worst case, we require to have 63 appropriate pairs \( (A^j, B^j) \) of the set \( U \). So we can determine the permutation matrix \( M_2^{-1} \) by at most known 63 \times 128 \) proper pairs \( (A^j, B^j) \) of the set \( U \), and in consequence, at most 128 pairs \( (H^i, R^i) \) of set \( S \). The permutation matrix \( M_2 \) is transpose of matrix \( M_2^{-1} \), so we can compute it easily. The algorithm of the attack depicted in Algorithm 2.
Algorithm 2: Disclosure attack algorithm to find the encryption matrix $M_2$

Data: $S = \{(H^i_2, R^i)\}_{i=1}^{128}$

Result: Permutation matrix $M_2^{-1}$

1. Construct the set $U = \{(A^1, B^1)\}_{j=1}^{128}$ contain all pairwise XOR of members of the set $S$;

2. for $i = 1$ to $128$ do
   3. Select $(A^1, B^1) \in U$;
      4. if $a_i = 0$ then
         5. $U_1 = \{t|b_t = 0, 0 \leq t \leq 127\}$;
      else if $a_i = 1$ then
         7. $U_1 = \{t|b_t = 1, 0 \leq t \leq 127\}$;
   8. for $j = 2$ to $\binom{128}{2}$ do
      9. if $a_i = 0$ then
         10. $U_1 = \{t|b_t = 0, 0 \leq t \leq 127\}$;
      else if $a_i = 1$ then
         12. $U_1 = \{t|b_t = 1, 0 \leq t \leq 127\}$;
         13. $U_1 = U_1 \cap U_j$;
       14. if $|U_1| = 1$ then
          15. $(a_i)M_2^{-1} = b_i$;
          16. Break;
       17. else
          18. Continue;
      19. if $|U_1| > 1$ then
         20. Take a new message $(H_2, R)$ and construct a new set $U$;
         21. Go to step 1;

(c) Now, we compute the permutation matrix $M_1$. Suppose that we know $H_2$, R and the permutation matrix $M_2$, $H_1$ is computed such as:

- $H_1 = (H_2 \times M_2) \oplus R$

We know that when we alter the real timestamp to $T = 0$, the reader encodes it to $T_1$ such that the first 64 bits of it will be equal to "1" and later 64 bits will be equal to "0". Now, we alter the real timestamp to $T'$ such that only the $j$-th ($0 \leq j \leq 63$) bit position of $T'$ is equal to "1" and send it to the reader. Upon receiving, the reader encodes it to 128 bits $T'_1$ and $T''_1$ and construct two pairs $(H'_1, T'_1)$ and $(H''_1, T''_1)$. Let

- $H = H_1 \oplus H'_1$ and $T = T_1 \oplus T'_1$.
- $H' = H_1 \oplus H''_1$ and $T' = T_1 \oplus T''_1$.

Common bit position in $H$ and $H'$ that is filled by "1", is permutation of common position in $T$ and $T'$ that is filled by "1" under the permutation
matrix $M_1^{-1}$. So 64 rows of the permutation matrix $M_1$ are computed by at most 128 pairs $(H_j, T_j)$. First 64 bits of $T_1$ are randomly filed, therefore we can’t discuss about these permuted bit positions.

2. **Desynchronization Attack:** In this attack, we show how the penetrator easily destroys the synchronization of the time $T$ updating between the tag and the reader. In the proposed protocol, there are two values for time, i.e. $T_0$ and $T_1$. $T_1$ represents the encoded current time that the reader receives from the internet when the reader makes an authentication request to the tag and $T_0$ represents the encoded time when the reader is successfully identified by the tag in the last session. We change last 64 bits of the current time value $T_1$ to $T'_1$ such that it is greater than $T_1$ no more than 48h and send it to the tag. Upon receiving, the tag check $T'_1 > T_0$, if it holds, it updates the value of $T_0$ to $T'_0$. In the next session, the tag rejects the query request of the reader because the current time value $T_1$ is lower than $T_0$. Also, we know that the binary representation of 48h is consist of 18 bits. So if we change one bit of $R$ or $H_2$, then by probability 18/128, synchronization between the reader and the tag will destroy. In the improved scheme, we eliminate the weakness by using the MDS matrix.

5 **Improved authentication scheme**

In this section, we propose an improved version of Fan et al. scheme that has no security challenges of its predecessor scheme and is resistance against known attacks. We keep the primary structure of their protocol and by made small changes, resolve their security challenges without significantly increasing its computational cost. Our improved scheme, see also Fig 2 like Fan et al. scheme has two phases as following:

5.1 **Initialization**

In the initialization phase

1. The secret key $S$ is shared between the back-end server and the tag.
2. The legitimate reader and tag store corresponding MDS matrices \( \{M_{MDS}^{-1}\} \) and \( \{M_{MDS}\} \) and also permutation matrix \( \{M^{-1}\} \) and \( \{M\} \) respectively.
3. We check real-time Unix timestamp $T$. If $T_{new} < T_{old}$ then $T'_{new} = T_{old} + 1$ and the reader encodes 64 bits $T'_{new}$ to 128 bits $T_1$, otherwise, the reader encodes 64 bits $T_{new}$ to 128 bits $T_1$.

5.2 **Authentication**

In Authentication phase

1. The reader generates two hidden random numbers $R_1, R_2$ and computes
   \[ H_1 = (R_1) \times M^{-1} \]
\[ H_2 = (R_2) \times M^{-1} \]
\[ H_3 = (T_1 \oplus R_1) \times M_{MDS}^{-1} \]
\[ H_4 = (H_3 \oplus R_2) \times M_{MDS}^{-1} \]

and sends \(< H_1, H_2, H_4>\) to the tag.

2. Upon receiving, first the tag computes \(R_1, R_2\) by inverse matrix \(M\), then the tag computes \(T_1\) as following:
\[ R_1 = (H_1) \times M \]
\[ R_2 = (H_2) \times M \]
\[ H_3 = (H_4) \times M_{MDS} \oplus R_2 \]
\[ T_1 = (H_3) \times M_{MDS} \oplus R_1 \]

If the last 64 bits of \(T_1\) are greater than \(T_0\) no more than 48h, the tag authenticates the reader and the tag updates the value of \(T_0\) with \(T_1\) and uses the updated \(T_0\) to compare with next \(T_1\) in next session. Then the tag computes
\[ Y_1 = (ID \oplus R_2) \times M_{MDS} \]
\[ Y_2 = (Y_1 \oplus T_1) \times M_{MDS} \]
\[ G = (S \oplus H_3) + ID \]

and sends \(< Y_2, G >\) to the reader.

3. The reader computes
\[ Y_1 = (Y_2) \times M_{MDS}^{-1} + T_1 \]
\[ ID = (Y_1) \times M_{MDS}^{-1} \oplus R_2 \]

and sends \(< ID, G, H_3 >\) to the back-end server.

4. The back-end server computes \(ID' = G - (S \oplus H_3)\). If \(ID = ID'\) the back-end server responds to the reader that the tag is legitimate.

6 Security analysis of improved protocol

In this section, we prove that the improved protocol is secure against the attacks proposed in this paper and next, by scyther tool, we show that the improved protocol is secure against any attacks.

6.1 Informal security proof

1. **Resistance against disclosure attack:** In improved scheme, we make two changes in Fan et al. scheme to prevent the attack.
   (a) First in authentication phase, we compute \(H_4 = (H_3 \oplus R_2) \times M_{MDS}^{-1}\), so \(H_4\) is depended on hidden random values \(R_1, R_2\) and alters in each session even if the adversary can insert a failure value \(T_1\). Therefore the adversary can not compute MDS matrix \(M_{MDS}^{-1}\).
   (b) Second, we replace \(R\) with \(H_3\) in relation \(G = (S \oplus R) + ID\). Therefor the adversary can not compute the \(ID\) from \(G = (S \oplus H_3) + ID\), because \(H_3\) is unknown for him. Also we depend \(Y_1\) to the random value \(R_2\), so \(Y_1\) changes in each session.
2. Resistance against desynchronization attack: This attack occurs when an adversary changes the real-time Unix timestamp or $H_4$ by man-in-the-middle attack. So in the improved scheme, in initialization phase, reader compares current timestamp $T_{\text{new}}$ with old value $T_{\text{old}}$ that is already saved in its memory. If $T_{\text{new}} < T_{\text{old}}$ then it computes $T_{\text{new}}' = T_{\text{old}} + 1$ and encodes it to new $T_1$. Also we use an MDS matrix to provide diffusion property. Therefor if an adversary changes a bit of $R_1$, $R_2$ or $H_4$, this alteration propagates to some other bits of $H_3$.

6.2 Formal security proof

Scyther is one of the types of software tools that can be used for formal analysis of the cryptographic protocols. Scyther supports Security Protocol Description

<table>
<thead>
<tr>
<th>Phase</th>
<th>Back-end server</th>
<th>Reader</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Init.</td>
<td>$S$</td>
<td>$(M_{\text{MDS}}^{-1}, T_1, R_1, R_2)$</td>
<td>$(M_{\text{MDS}}, M, T_0, S, ID)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>if $T_{\text{new}} &lt; T_{\text{old}}$ then $T_{\text{new}}' = T_{\text{old}} + 1$ and encode $T_{\text{new}}'$ to $T_1$ else encode $T_{\text{new}}$ to $T_1$</td>
<td></td>
</tr>
<tr>
<td>Aut.</td>
<td></td>
<td>$H_1 = (R_1) \times M^{-1}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$H_2 = (R_2) \times M^{-1}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$H_3 = (T_1 \oplus R_1) \times M_{\text{MDS}}^{-1}$</td>
<td>$H_1, H_2, H_3$ $R_1 = (H_1) \times M$ $R_2 = (H_2) \times M$ $H_3 = (H_4) \times M_{\text{MDS}} \oplus R_2$ $T_1 = (H_3) \times M_{\text{MDS}} \oplus R_1$ if $T_1 &gt; T_0$ update $T_0 = T_1$ $Y_1 = (ID \oplus R_2) \times M_{\text{MDS}}$ $Y_2 = (Y_1 \oplus T_1) \times M_{\text{MDS}}$ $G = (S \oplus H_3) + ID$ G,$Y_2$ $Y_1 = (Y_2) \times M_{\text{MDS}}^{-1} \oplus T_1$ $ID = (Y_1) \times M_{\text{MDS}}^{-1} \oplus R_2$ $\leftarrow G, ID, H_3$ $ID' = G - (H_3 \oplus S)$ $\text{if } ID \neq ID'$</td>
</tr>
</tbody>
</table>

Fig. 2. Our improved Scheme, where Init. and Auth. denote initialization and authentication respectively
Language (SPDL) to implement a protocol. We must write all events of each part of the protocol in the set of roles. Roles are defined by a sequence of events like computing, sending or receiving of terms that carry out in each part of a protocol. In this protocol, we have two roles. Report of the scyther tool shows that our improved protocol is safe against all threats. Security analysis result of the improved scheme is presented in Table 3.

Table 3. Security analysis result of the improved scheme with Scyther

<table>
<thead>
<tr>
<th>Claim</th>
<th>Status</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secret ID</td>
<td>Ok</td>
<td>No attacks within bounds</td>
</tr>
<tr>
<td>Secret S</td>
<td>Ok</td>
<td>No attacks within bounds</td>
</tr>
<tr>
<td>Niagree</td>
<td>Ok</td>
<td>No attacks within bounds</td>
</tr>
<tr>
<td>Nisynch</td>
<td>Ok</td>
<td>No attacks within bounds</td>
</tr>
<tr>
<td>Alive</td>
<td>Ok</td>
<td>No attacks within bounds</td>
</tr>
<tr>
<td>Weakagree</td>
<td>Ok</td>
<td>No attacks within bounds</td>
</tr>
</tbody>
</table>

7 Implementation

Fan et al. scheme involves low cost operation such as XOR (⊕), ADD (+) and permutation. We keep primary structure of the protocol and by adding a MDS matrix to it, eliminate its security weaknesses. MDS matrices have maximum branch number and used in block cipher to provide diffusion property. We implement our improved protocol in ISE 14.7 environment for Virtex-7 FPGAs with two different MDS matrices. First, we use MDS matrix of the encryption algorithm AES [1]. It has branch number "5" and can be efficiently implemented in hardware. Next, we use MDS matrix of encryption algorithm ARIA [5]. It is a 16 × 16 binary matrix of branch number "8" and can be efficiently implemented in hardware too. We compared resource consumption of improved protocol and some other lightweight authentication protocols [10,12,3] in Table 4.

Table 4. Resource used in the tag, where scheme 1 and scheme 2 denote our improved scheme with MDS matrix of algorithm AES and ARIA respectively

<table>
<thead>
<tr>
<th>Protocol</th>
<th>[3]</th>
<th>[12]</th>
<th>[10]</th>
<th>Improved scheme 1</th>
<th>Improved scheme 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Slice LUTs</td>
<td>197</td>
<td>426</td>
<td>1126</td>
<td>1077</td>
<td>1026</td>
</tr>
<tr>
<td>Number of Slice Registers</td>
<td>384</td>
<td>32</td>
<td>879</td>
<td>258</td>
<td>261</td>
</tr>
</tbody>
</table>

Our improved scheme has not security weakness of Fan’s scheme, even though, based on Table 4 its computational cost is slightly higher than some other
schemes. The implementation cost of the improved scheme depends on chosen MDS matrix. We show this issue in cost difference between improved scheme 1 and 2.

8 Conclusion

In this paper, we have analyzed more deeply the Fan et al. scheme and have shown that their scheme is vulnerable against disclosure attack. We have performed a disclosure attack on the scheme in three different ways. In the first, the ID and the secret key $S$ have been revealed by at most twenty session information transferred between the tag and the reader. This attack is passive so it can be performed easily by eavesdropping communicated messages between the tag and the reader. In the following, using a man-in-the-middle attack, we have computed all rows of the encryption matrix $M_2$ and half rows of the encryption matrix $M_1$. Computing all rows of the permutation matrix $M_2$ and half rows of the permutation matrix $M_1$ requires at most 128 proper session information. Furthermore, we have shown that the proposed scheme is also vulnerable to a desynchronization attack. To overcome this weakness, we proposed an improved version of the scheme that used MDS matrices. Next, we evaluate the security of our improved scheme by schyter tool, and in the end, we implement the improved scheme on Virtex-7 FPGAs using VHDL language and compare the implementation cost with some relative protocols.

References


