Research Paper

Improvement of the Efficient Mutual Authentication Protocol for Passive RFID Tags (EMAP)

MohammadReza Mehrabani* a, Soosan Sadegha b

a Department of Electrical Engineering, Iran University of science and Technology, Tehran, Iran
b Pharmacy faculty, Tehran university of medical science, Tehran, Iran

ARTICLE INFO
Received: 24 July 2021
Reviewed: 5 August 2021
Revised: 31 August 2021
Accepted: 8 September 2021

ABSTRACT
Radio Frequency Identification (RFID) system is a wireless automatic identification using low-cost RFID tags. Promotion of privacy and security in RFID systems has been considered essential due to their usage in everyday life. In 2012, Guo-Rui Li et al. Proposed a new Efficient Mutual Authentication Protocol for passive RFID (EMAP), that providing confidentiality, untraceability, and efficiency. In addition to proving the security problems of the EMAP method, this paper has presented a novel method to improve this weak point.

1. Introduction
Identification technology through radio frequency (RFID) is a new technology that is used to identify objects and living creatures (Mehrabani & Sadegha, 2021). Due to many advantages such as lower costs, increasing speed and large-scale identification, the use of this technology has been increased day by day. The foundation of this technology is similar to barcode technology with the difference that in RFID systems, identification takes place without any physical contact or direct visibility and recognition perform via radio waves (Abugabah et al., 2020). These advantages make this technology attractive for commercial and industrial users, so that today the deployment of RFID systems in various areas such as credit cards, subways and buses, new Passport cards and new electrical barcodes can be seen (Ranasinghe & Yu, 2017) (Frith, 2019). Due to the use of RFID technology in sensitive areas such as access control systems, supply chain management, new passports, non-contact smart cards, and etc., study on the security aspects of this technology and the deployment of security protocols for authentication in this system, is an urgent need for industries and organizations (Nozari et al., 2021).

* Corresponding author
mrmehrabany@gmail.com
Authentication protocols are the most important tools to provide security in the highest layer, namely the application layer, in RFID system. They run between the tag and card reader, and during their implementation, the parties review the accuracy of each other’s identity to accept. Low cost production in RFID tags is one of the most important features (Namin et al., 2018), so there is no possibility of using encryption separately and the security of the entire system depends directly on the security level of protocols that are used for identification and authentication.

In RFID systems, the most important features of a secure identification and authentication protocol are as follows.

- **Confidentiality**: Ensure that information is only available to those who are authorized to access this information (Kitsos & Zhang, 2008). Confidentiality means that in a secure protocol there shouldn’t be any leakage of information to an attacker or eavesdropper. Information that is stored on the tag such as a unique identifier and secret keys that are employed in reconnaissance operations, should be only disposal to tag, and if it’s necessary, the server should be final. Besides this, another important point about confidentiality, is privacy and the location privacy of the tag or tag’s holder. Location privacy of a tag means that, if an attacker, eavesdrops the session and saves exchanged information during the identification protocol between the tag Ti and an allowed reader at time t, the attacker fails to identify any other transactions related to tag Ti, in the other time.

- **Data integrity**: Maintaining the authenticity and integrity of information and processing methods (Zhuang et al., 2018). It means that no person should be able to alter or manipulate the exchange of information between parties of the protocol. The originality of the message can prevent many attacks that leads to impersonation.

- **Availability**: Assurance that all authorized users can have access to information and other requirements (Burmester & Munilla, 2014). This means that authorized users (tags) at any time and any place need system services, be able to easily use them.

- **Authentication**: The most basic goal that an identifying protocol seeks, is checking the identity of an entity or parties of the protocol (Paise & Vaudenay, 2008).

The new EMAP version, presented in this paper, successfully solved EMAP related data integrity issue (Li et al., 2012). The rest of the article consists of five sections. Section 2 gives a brief introduction of EMAP. Section 3 identifies EMAP security shortcomings. Section 4 is designed to present the improved protocol, and Section 5 evaluates whether or not the protocol is secure and efficient. We give conclusion of the study in section 6.

### 2. Review EMAP protocol

In this Section, we review the notations used in EMAP protocol (Li et al., 2012). The detailed information is available in EMAP’s protocol.

The notation used in EMAP defined as follows:

- \( pk \) : The public key that is saved in the back end server
- \( r_1 \) / \( r_2 \) : The random numbers
- \( sk \) : The private key that is saved in the back end server
- \( s \) : The shared key for all tags and stores it into the tag and back end
- \( k \) : The random secret key for each tag and stores it into the tag and back end
- \( ID \) : The Tag identification number
- \( A \rightarrow B \) : A forwards a message to B.
- \( A \oplus B \) : Message A is XORed with message B.
- \( E_{pk} (\cdot) \) : Encryption function with the server’s public key \( pk \).
- \( f_s (\cdot) \) : The keyed hash function f with the shared key s
• \( f_k(\cdot) \): The keyed hash function \( f \) with the secret key \( k \)
• \( t_i \): A random value that is chosen with the system

EMAP consists of two phases: the initialization phase, and the authentication phase.

### 2.1 Initialization phase

In this phase, the system produces a pair of public keys \( pk \), private key \( sk \), secret key \( k \), shared key \( s \), corresponding tag identification \( ID \), and the encrypted result \( E_{pk}(ID|t_i) \) and stores them in the back end server. It also stores a random secret key \( k \) for each tag and a shared key \( S \) for all tags and \( E_{pk}(ID|t_i) \) in the tag.

### 2.2 Authentication phase

Fig. 1 illustrates the authentication phase of EMAP protocol. The authentication phase consists of three steps described as follows in detail.

**Round 1.**

Reader → Tag: \( r_1 \)
The tag receives random number \( r_1 \) which is made by the reader.

**Round 2.**

Tag → Reader: \( \langle E_{pk}(ID|t_i), f_s(r_1 | r_2), f_k(E_{pk}(ID|t_i)|r_1 | r_2) \rangle \)
Tag generates random number \( r_2 \), and retrieves the preinstalled encryption information \( E_{pk}(ID|t_i) \). It generates \( f_s(r_1 | r_2) \) and \( f_k(E_{pk}(ID|t_i)|r_1 | r_2) \) respectively. Then XORs \( E_{pk}(ID|t_i) \) with \( f_s(r_1 | r_2) \) and sends \( E_{pk}(ID|t_i) \oplus f_s(r_1 | r_2), f_k(E_{pk}(ID|t_i)|r_1 | r_2) \) to the reader.

The reader forwards the received authentication message with the generated random value \( r_i \) to the server.

The server first computes the hash function value \( f_s(r_1 | r_2) \) and XORs with the first part of received message to get \( E_{pk}(ID|t_i) \). It then decrypts \( E_{pk}(ID|t_i) \) using its own private key \( sk \) to derive the tag’s \( ID \). If the \( ID \) is invalid, the server aborts, otherwise, it finds the tag’s corresponding secret key \( k \) and computes \( f_k(E_{pk}(ID|t_i)|r_1 | r_2) \) to verify the tag. If the hash value equals the last part of the received message, then the tag is authenticated. Otherwise, the server aborts.

**Round 3.**

Database → Reader: \( \langle E_{pk}(ID|t_{i+1}), f_k(r_2), f_k(E_{pk}(ID|t_i)|r_2 | r_1) \rangle \)
After authentication, server chooses new random number \( t_{i+1} \) and computes \( E_{pk}(ID|t_{i+1}) \). After that it generates \( E_{pk}(ID|t_{i+1}) \oplus f_k(r_2), f_k(E_{pk}(ID|t_i)|r_2 | r_1) \) and sends it to the reader.

Reader → Tag: \( \langle E_{pk}(ID|t_{i+1}), f_k(r_2), f_k(E_{pk}(ID|t_i)|r_2 | r_1) \rangle \)

The reader forwards the message to the tag, then the tag computes \( f_k(E_{pk}(ID|t_i)|r_2 | r_1) \) and compares it with the last part of the received message. If two values were equal, the server is authenticated. Otherwise, the tag aborts. Finally, the tag calculates \( f_i(r_2) \) and executes exclusive or (XOR) operation on the first part of the received message to get the new encryption information \( E_{pk}(ID|t_{i+1}) \) and replaces the old one.
In this section, we show that EMAP has the following security problem.

3.1 Data integrity problem:
We will show that EMAP is vulnerable to de-synchronization attack. In the third round an attacker can change first part of the message $E_{pk}(ID|t_{i+1})\oplus f_k(r_2)$, by executing XOR operation with arbitrary random number $r'$ to generate $E_{pk}(ID|t_{i+1})\oplus f_k(r_2)\oplus r'$ and send $(E_{pk}(ID|t_{i+1})\oplus f_k(r_2)\oplus r', f_k(E_{pk}(ID|t_1)|r_2|r_3))$ to tag. When tag receives the message, computes $f_k(E_{pk}(ID|t_1)|r_2|r_3)$ and compares it with the last part of the received message. Since second part of the message has not changed, tag will authenticate the server. Now, tag calculates $f_s(r_2)$ and executes XOR operation on the first part of the received message $E_{pk}(ID|t_{i+1})\oplus f_k(r_2)\oplus r'$. It will extract a massage $E_{pk}(ID|t_{i+1})\oplus r'$ and replace the old one. So the De-synchronization between the tag and the server will occur. Even the attacker can send $r'$ instead of $E_{pk}(ID|t_{i+1})\oplus f_k(r_2)$. In this case tag will XOR it with $f_s(r_2)$ and obtains $r'\oplus f_s(r_2)$ and replaces it instead of $E_{pk}(ID|t_{i+1})$. We can see that the XORed result is not equal to the original which is sent by database, therefore, EMAP has DATA integrity problem.

4 Improved EMAP (EMAP+)
This section proposes EMAP* that is an improvement of the EMAP. EMAP* can eliminate the above described security problem. The information kept within respective devices is the same as EMAP. Notation used in EMAP* is just like EMAP. EMAP* consists of two phases: the initialization phase, and the authentication phase.

4.1 Initialization phase
In this phase, the system produces a pair of public keys $pk$, private key $sk$, secret key $k$, corresponding tag identification $ID$, then the encrypted result $E_{pk}(ID|t_i)$ therefore shared key $s$, and stores them in the back-end server. It also stores a random secret key k for each tag and a shared key S for all tags and $E_{pk}(ID||t_i)$ in the tag.

4.2 The authentication phase
In the following, the improved authentication phase steps are explained in detail and indicated in Fig. 2

Round 1.

Reader $\rightarrow$ Tag: $r_1$
The tag receives random number \( r_1 \) which is made by the reader.

**Round 2.**

\[
\text{Tag} \rightarrow \text{Reader}: \langle E_{pk}(ID||t_i)||f_s(r_1||r_2), r_2, f_k(E_{pk}(ID||t_i)||r_1||r_2)\rangle
\]

Tag generates random number \( r_2 \), and retrieves the preinstalled encryption information \( E_{pk}(ID||t_i) \). It generates \( f_s(r_1||r_2) \), and \( f_k(E_{pk}(ID||t_i)||r_1||r_2) \) respectively. Then sends \( \langle E_{pk}(ID||t_i)||f_s(r_1||r_2), r_2, f_k(E_{pk}(ID||t_i)||r_1||r_2)\rangle \) to the reader.

Reader forwards the received authentication message with the generated random value \( r_1 \) to server. The server first computes hash function value \( f_s(r_1||r_2) \) and XORS it with the first part of received message to get \( E_{pk}(ID||t_i) \). It then decrypts \( E_{pk}(ID||t_i) \) by using its own private key \( sk \) to derive tag’s \( ID \). If the \( ID \) is invalid, the server aborts, otherwise, it finds the tag’s corresponding secret key \( k \) and computes \( f_k(E_{pk}(ID||t_i)||r_1||r_2) \) to verify the tag. If the hash value equals to the last part of the received message, then the tag is authenticated. Otherwise, the server aborts.

**Round 3.**

\[
\text{Database} \rightarrow \text{Reader}: \langle E_{pk}(ID||t_{i+1}), f_k(E_{pk}(ID||t_i)||E_{pk}(ID||t_{i+1}||r_2)\rangle
\]

After authentication, server chooses a new random number \( t_{i+1} \) and computes \( E_{pk}(ID||t_{i+1}) \). After that it generates \( \langle E_{pk}(ID||t_{i+1}), f_k(E_{pk}(ID||t_i)||E_{pk}(ID||t_{i+1}||r_2)\rangle \) and send it to the reader.

\[
\text{Reader} \rightarrow \text{Tag}: \langle E_{pk}(ID||t_{i+1}), f_k(E_{pk}(ID||t_i)||E_{pk}(ID||t_{i+1}||r_2)\rangle
\]

The reader forwards the message to the tag, and tag gets the first part of the message to computes \( f_k(E_{pk}(ID||t_i)||E_{pk}(ID||t_{i+1}||r_2) \) and compares it with the last part of the received message. If two values are equal, the server is authenticated. Otherwise, the tag aborts. Finally, the first part of message \( E_{pk}(ID||t_{i+1}) \) is replaced with old one.

**Fig. 2. The execution phase of EMAP**
Analysis

5.1 Confidentiality
The ID information of RFID in this protocol will remain secret just like the EMAP protocol. In fact, the only difference between our protocol and EMAP is in the third round where in our protocol the reader sends $E_{pk}(ID||t_{i+1})$, instead of $E_{pk}(ID||t_i+1)\oplus f_k(r_2)$. In fact, adversary would never be able to obtain the ID information because it is protected by encryption.

5.2 Untraceability
EMAP* can provide untraceability just like EMAP, due to $E_{pk}(ID||t_{i+1})$ in the third round, which is changing every phase.

5.3 Data integrity
The EMAP* Protocol unlike EMAP that was vulnerable to de-synchronization attack supplies data integrity and this is because of changes in the last round of the EMAP protocol. Suppose that in the third round the adversary $A$ can change $E_{pk}(ID||t_{i+1})$ and sends it to tag. When Tag receives the message, starts to compute the hash function $(E_{pk}(ID||t_i)||E_{pk}(ID||t_{i+1}))||r_2)$ and compare it with the last part of the received message. Because $E_{pk}(ID||t_{i+1})$ has been changed by adversary, the tag will abort.

5.4 Mutual authentication
EMAP* provides two-way mutual authentication just like EMAP. The tag is authenticated in the second round by $f_k(E_{pk}(ID||t_i)||r_1||r_2)$ . On the other side the reader proves its identity to the tag by $f_k(E_{pk}(ID||t_i)||E_{pk}(ID||t_{i+1})||r_2)$. So EMAP* can prevent the impersonation attack and authenticate reader or tag and vice versa.

5.5 Efficiency
EMAP* is exactly the same as EMAP. On the reader side, The time of finding tag’s $ID$, and the space requirement is equal to $O(1)$. On tag side, the two factors mentioned above are equals to $O(1)$. As claimed in (Li et al., 2012) these features relieve the constraints on the RFID tag in real applications in comparison with the other protocols.

6 Conclusions
It observed that with a little change in the EMAP we could promote this protocol and have an efficient mutual authentication protocol for passive RFID tags that provides data integrity, confidentiality, un-traceability, mutual authentication, and efficiency, while EMAP* has all features of the EMAP in time complexity, space complexity, and communication cost.
References:


This work is licensed under a Creative Commons Attribution 4.0 International License.