A review of authentication protocols for RFID security on smart healthcare

Hanif Restu Dearfian* and Amiruddin Amiruddin

Sekolah Tinggi Sandi Negara, Bogor, Indonesia

*haniest@student.stsn-nci.ac.id

Abstract. Patient safety is a top priority in the medical tasks where patient identification is one of the important things in the process of anesthesia and surgery. Errors in identifying patients can be fatal to the patient's safety. Such identification of patients that still use traditional methods is very susceptible to causing medical faults. To overcome the weaknesses of such traditional identification, researchers have used Radio Frequency Identification (RFID) technology to automatically identify patients. With RFID technology, automatic identification can be done without having to make physical contact with the patient. However, RFID technology still has flaws, including the absence of a protection mechanism for RFID security. This paper presents the results of a literature study on authentication protocols for the security of RFID implementation for smart healthcare.

1. Introduction

Radio Frequency Identification (RFID) is an automatic identification technology without having to make physical contact with the patient. RFID tags [1] are small, inexpensive, and can be combined with a variety of objects such as bracelets, clothes, cards, and others. This technology is also widely applied in human life today [2]. In the research conducted by [3], it is said that common problems that often occur are the misuse of patient identification bracelets as well as medical operators which of course under certain conditions can experience fatigue. Therefore, to deal with these problems, researchers offer RFID technology [4][5][6] that has advantages such as: contactless, practical, inexpensive, durable, and can be paired on various objects.

Every object that will be identified by the RFID system is physically marked using RFID tags. This tag consists of a microchip that functions as a storage and performs logical operations; and an antenna used for wireless communication. RFID reader reads RFID tags through the radio frequency interface. Similar to RFID tags, this reader also has internal storage memory and the ability to perform logical operations. Not only that, this reader also functions as an interface to the back-end database to provide additional functions such as data storage. However, this RFID technology still has flaws [7] including the absence of a protection mechanism for the RFID system, which causes it to be vulnerable to attacks. Therefore, a protection mechanism is required to protect RFID from various types of attacks [8]. This paper discusses the resilience of some RFID authentication protocols against attacks. The result of this study can give an insight for researchers in choosing a suitable protocol for RFID security.
2. Method and Materials

2.1. Review Preparation
For this study, we reviewed 5 (five) authentication protocols for RFID security. We also define several attacks for RFID as in Table 1.

2.2. Method
Review method used in this study is as follows. First, we briefly explain 5 (five) authentication protocols for RFD security found on the recent literature. Second, we analyze the strengths and the weaknesses of the chosen protocols. Lastly, we compare the protocols according to their resistance to several types of attacks previously defined.

Table 1 Several attacks on RFID

| Attacks                  | Meaning                                                                 |
|-------------------------|-------------------------------------------------------------------------|
| Eavesdropping           | Eavesdropping occurs when the attacker "eavesdrops" on a communication channel to get information transmitted on it. |
| Tracking                | This attack is done by utilizing the design error of the RFID tag itself, which always responds to the query from the reader. By sending queries and getting the same response from tags in various locations, information of where the tags are located and what locations they have visited can be obtained. |
| Man-in-the-Middle       | Also know as Relay attack, this attack occurs when the attacker places a device between the reader tags that can block the information sent by the sender, either reader or tag, to be subsequently changed and forwarded to the recipient. |
| Replay                  | An attack on an RFID system where a device repeats key information from communication between the reader and the tag, with the aim of fraud. The device repeats authentication between the reader and tag, deceives the reader or tag to pass the verification process. |
| Forward Security        | If an attacker obtains information from a tag, the attacker can use that information to track future transactions. |
| Denial of Service (DoS) | The purpose of this attack is to make the system malfunction by sending a large number of messages to the server so that the server crashes |
| Cloning                 | In this attack, the attacker clones or makes an imitation of the tag with the information obtained from the tag. |

3. Results and Discussion

3.1. Description of Authentication Protocols For RFID

3.1.1 Time Stamp Based Mutual Authentication Protocol (TSBMA). The steps of this protocol [9] is as follows.

Step 1 - the reader converts the current time to T time stamp and calculates h ( ID _R ⊕ T). Both are sent by the reader to the tag.

Step 2 - After receiving the message from the reader, the tag will not respond until the time stamp used for the communication meets the conditions where T_0 < T < T_max. If these conditions can be met, then the tag will match between h( ID _R ^ ⊕ T) and h( ID _R⊕T). If a match is found, the tag will generate a random value r, calculate h( ID _R⊕T⊕r) then send it to the reader.

Step 3 - after receiving the response from the tag, the reader sends h ( ID _R⊕T⊕r), r, h( ID _R⊕T), and T to the database for authentication
3. Step 4 - after receiving a message from the reader, the server searches in the database and matches h(\[ID\] _R'' ⊕ T) with h(\[ID\] _R ⊕ T). If a match is found, the tag is valid, and the server will generate a new random value r_2 and calculate h(ID' ⊕ r_2). The server sends h(ID' ⊕ r_2) and r_2 to the reader. Conversely, if no match is found, the reader or tag is considered non-genuine and communication is stopped.

3. Step 5 - the reader passes the information sent by the server to the tag. The tag then calculates h(ID' ⊕ r_2) and matches it with h(ID' ⊕ r_2). If a match is found, the tag updates the time stamp T₀ to T. If no match is found, then the database or reader is considered not original and T₀ does not change.

3.1.2 Semi-Randomized Access Control Protocol (SRAC). The steps of this SRAC protocol [10] is as follows.

Step 1 - the reader sends the query to the tag.

Step 2 - the tag sends MetaID = H(Key) to the reader.

Step 3 - the server looks for Key using MetaID, generates RS random numbers, and checks whether H(Key ⊕ R_S) is unique when compared to other MetaIDs. If not unique, the server re-generates RS until H(Key ⊕ R_S) becomes unique. The server updates the Key as follows.

\[
\begin{align*}
\text{If } H(\text{Key}_{\text{curr}}) &= \text{MetaID, then Key}_{\text{prev}} &\leftarrow \text{Key}_{\text{curr}}, \text{Key}_{\text{curr}} &\leftarrow H(\text{Key}_{\text{curr}} \oplus R_S) \\
\text{If } H(\text{Key}_{\text{prev}}) &= \text{MetaID, then Key}_{\text{curr}} &\leftarrow H(\text{Key}_{\text{prev}} \oplus R_S) \\
\text{Key} &\leftarrow \text{Key}_{\text{prev}}
\end{align*}
\]

The Server sends R_S and H(Key || R_S) to the tag through the reader.

Step 4 - the tag verify if H(Key || R_S) is true. If true, the tag will update Key ← H(Key ⊕ R_S).

The need for R_S in this hash function is that the tag need to verify the integrity of R_S. The Server authenticates the tag by verifying the presence of the MetaID in the database server, while teh tag authenticates the server by verifying H(Key || R_S). As the protection against the DoS attack, the key update of the server must be faster than that of the tag. The Server store two keys, Key_{curr} and Key_{prev}.

3.1.3 Two-Way Authentication Protocol (TWAP). TWAP [11] This is a hash function-based authentication protocol for RFID. The following explains the workflow of this protocol.

Step 1 - Reader reads the local time of the T system and sends T and the query to the tag.

Step 2 - The tag generates H(T || S) using the received T and the random value S, then the tag generates M1 = H(T || S) ⊕ H(TID), M1 is sent to the reader along with S.

Step 3 - After receiving (M1, S), the reader checks the T value and extracts the new time Tnew, ΔT' = Tnew - T, if ΔT' is too large compared to ΔT, it can be concluded that there has been an attack in communication so interfere with the authentication process. But if it's still within the specified limit, then the reader calculates M2 = H(RID) ⊕ H(Tnew || S), and sends (M1, M2, T, Tnew, S) to the database.

Step 4 - After receiving a message from the reader, the database extracts the local T2 time and calculates TDT' = T2 - Tnew. If TDT' is too large compared to ΔDT, then it can be concluded that there has been an attack in the communication process so that the communication process is immediately ended. If the
TDT’s value is still within the specified limit, the database will generate $H(T_{\text{new}} \oplus S)$ and XORATE it with $M2$, and search in the database to find the same $H(RID)$ value, if a match is found, then the reader considered original. To check the authenticity of the tag, the database generates $H(T \oplus S)$ and XORES it with $M1$, then the database matches it with $H(TID)$ to check the authenticity of the tag.

**Step 5** - The database calculates $H(TID') \oplus H(RID')$ and sends it to the reader

**Step 6** - Reader calculates $H(TID')$ based on $H(RID')$ and sends it to the tag

**Step 7** - Tags compare the results of calculations with $H(TID')$ which they got themselves. If a match is found, the authentication process is successful.

3.1.4 *Mutual Authentication Protocol (MAP).* MAP [12] is an RFID authentication protocol based on a lightweight hash function that provides RFID authentication services between tags and databases. The following is the steps of this protocol.

**Step 1** – the reader generates random number $N_R$ and send request and $N_R$ to the tag. After receiving the message, the tag generates random number $N_T$ and calculates $M1 = h(S \oplus N_R \oplus N_T)$. Next, the tag sends $N_T$ and $M1$ to the reader.

**Step 2** – the reader calculates $M2 = h(RID \oplus N_R)$ and sends the message $(M1, M2, N_R, N_T)$ to the database.

**Step 3** – After receiving $M1$, $M2$, $N_R$ and $N_T$ from the reader, the database do the following: The database compares whether $N_R$ matches $N_R$ (old). If it matches then the check fails. If $N_R$ and $N_R$ (old) do not match, then the database calculates $h(S \oplus N_R \oplus N_T)$ and receives random numbers $N_R$ and $N_T$. Next the database checks $M1 = h(S \oplus N_R \oplus N_T)$ to verify whether the tag is truly genuine. The database calculates $h(RID \oplus N_R)$ and receives a random number $N_R$. Reader is considered original if the ratio $M2 = h(RID \oplus N_R)$ is successful. The database generates $N_{DB}$ random numbers and calculates $M3 = h(ID \oplus N_{DB})$ and $S_{new} = h(S \oplus N_{DB} \oplus N_T)$. Then $M3$ and $N_{DB}$ are sent to the reader. The database updates $S_{new} = h(S \oplus N_{DB} \oplus N_T)$.

**Step 4** - after the reader receives the $M3$ and $N_{DB}$ messages, it passes it to the tag

**Step 5** - after receiving $M3$ and $N_{DB}$, the tag calculates $M3 = h(ID \oplus N_{DB})$. Bidirectional authentication occurs when tags successfully verify this message. When the transaction is complete, the tag will update the $S_{new} = h(ID \oplus N_{DB} \oplus N_T)$.

3.2 *Comparison of protocols*

Table 2 summarizes several authentication protocols describe in this study. The analysis of each protocol is given subsequently.

| Protocol(s) | Forwards Security | Tag Tracking | Cloning Attack | Replay Attack | Eavesdropping | Denial of Service | Denial of Service |
|-------------|-------------------|--------------|----------------|---------------|---------------|-------------------|-------------------|
| TSBMA       | √                 | √             | √              | √             | √             | √                 | -                 |
| SRAC        | √                 | √             | √              | -             | -             | -                 | √                 |
| TWAP        | √                 | √             | √              | -             | -             | -                 | √                 |
3.2.1 Time Stamp Based Mutual Authentication Protocol (TSBMA). This protocol is resistant to the Man in the Middle (MITM) attack, because in this protocol the reader has a unique identifier, IDr, and the message sent is entered hashed first. When the attacker pretends to be the original reader who will communicate with the tag, the tag will verify the authenticity of the reader. Because the information sent is in the form of a hash, even if the attacker manages to get the information, the attacker cannot obtain the identifier value. So, the attacker cannot be authenticated by the tag and fails to pretend to be the original reader. Time stamp T on this protocol is updated in each communication session and the response of the tag is different for each random number r that is generated. If the attacker tries to do a replay attack, then this protocol can prevent it because the T value always changes so that it does not meet \( h(ID^r \oplus T^r) = h(ID^r \oplus T^r) \). This protocol is also resistant to De-synchronization attack because the last step in this protocol is to update the \( T_0 \) communication timestamp to T. If the attacker gets the authentication information that the reader sends to the database, the \( T_0 \) in the tag will not be updated.

3.2.2 Two-Way Authentication Protocol (TWAP). This protocol is resistant to forward security because in addition to using the hash function in which the characteristics of the hash function are one way or one way, random characteristics of random numbers generated by tags and differences in timestamp generated between the reader and database in each communication make the attacker unable to know previous tag information through the information it has now. The attacker also cannot perform replay attacks because the value of the T timestamp generated by the reader is always changing in each communication. With the ID of the tag and reader encrypted with the hash function as well, even though the attacker managed to get information from the ID, the attacker cannot know the original value of the ID, so it will be difficult for the attacker to fake the reader or tag. By changing the value of random numbers and timestamp in each communication also makes the attacker difficult to tag tracking by identifying the specific location of the tag through information between the reader and tag because in this protocol the reader does not send information related to the tag to the identity. The use of static IDs also makes this protocol resistant to De-synchronization attacks because there is no dynamic update mechanism from the content inside the tag.

3.2.3 Semi-Randomized Access Control (SRAC). This protocol is resistant to tag tracking because confidential information in tags always changes every time a communication is successful. This protocol also provides protection against forward security, because the reader and tags update their confidential information every time they communicate using a hash function that has one-way characteristics. So even if the attacker manages to get confidential information from the communication between the reader and the tag, the attacker cannot find out the previous confidential information from the information he obtained. This protocol can prevent Denial of Service (DoS) attacks because in this protocol the server only needs to store previous confidential information for each tag. If the server fails to find MetaID, the server can look for it through the previous MetaID. However, this protocol is vulnerable to replay attacks, because the attacker can pretend to be an original reader or tag by repeating the previous message.

3.2.4 Mutual Authentication Protocol (MAP). This protocol is resistant to tag tracking during each communication session provided that the secret value S, \( N_r \), and \( N_T \) are random numbers. Even though the attacker gets \( M_1 \), he will find it difficult to track because the secret value has changed. Cloning attacks can be prevented by using a secret value and a unique identifier. To make a cloning of a tag, the attacker must fake \( M_1 \) and \( M_2 \). However, the attacker cannot calculate \( M_1 \) and \( M_2 \) because the attacker cannot obtain the secret value of S and the RFID reader. If the attacker becomes man in the middle and manages to get \( M_1, M_2, N_r \) and \( N_T \) sent by the reader to the database and attempts to do a replay attack by resending the message, then authentication will fail because what was sent back by the attacker is the previous \( N_r \) value, which is stored in the database as \( N_r \) (old). This protocol is also resistant to forward
security, because the database and tags store the secret value of S and update it in each communication session. So the forward security attack will fail because the secret value of S has no connection with the previous session. To prevent DoS attacks, this protocol can handle the problem of a constant S value when the attacker performs a DoS attack, by checking the next session that $S = S_{old}$ and fixing the S value so that it is not constant.

4. Conclusion
This study has examined several authentication protocols for RFID security in smart healthcare applications. The results of the analysis show that the reviewed protocols have different mechanisms for protecting RFID. From the results of our study, based on vulnerabilities/attacks, there is no protocol that is resistant to all attacks. The Two-Way Authentication Protocol, even though it can overcome DoS attacks, is not resistant to eavesdropping and desynchronization attacks. The Time Stamp Based Mutual Authentication protocol, although most can overcome attacks, is not resistant to DoS attacks. It can also be seen that all proposed protocols use the hash function in their operations. The hash function which is a one-way function does not require expensive computing costs, thus it is suitable to be applied in RFID which does have limitations in memory and computational ability.

The future work of this study is to choose and implement the right authentication protocol for RFID security in smart healthcare applications.

5. References
[1] Chunhui Piao ; Zhenjiang Fan ; Chunyan Yang ; Xufang Han; Research on RFID security protocol based on grouped tags and re-encryption scheme; 2010 IEEE International Conference on Wireless Communications, Networking and Information Security; pp. 568 - 572, 2010
[2] Min-Shiang Hwang, Chia-Hui Wei, and Cheng-Yee Lee; Privacy and security requirements for RFID applications, Journal of Computers, vol 20 no.3, 2009
[3] Zhen-Yu Wu, Tzer-Long Chen, Sung-Chiang Lin, and Charlotte Wang; A Secure RFID Authentication Scheme for Medicine Applications; Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), Volume: 1, 2013, pp. 175-181.
[4] Hanshen Gu ; Dong Wang; A Content-aware Fridge based on RFID in smart home for home-healthcare; 2009 11th International Conference on Advanced Communication Technology; Volume: 02; pp. 987 - 990, 2009
[5] B. Venkatalakshmi ; A. Pravin Renold ; R. S. Lysa Packiam; Smart RFID care [SRC] for pervasive health care system; 2011 IEEE 3rd International Conference on Communication Software and Networks; pp.: 650 - 653, 2011
[6] Luca Catarinucci ; Danilo de Donno ; Luca Mainetti ; Luca Palano ; Luigi Patrono ; Maria Laura Stefanuzzi ; Luciano TarriconeAn IoT-Aware Architecture for Smart Healthcare Systems; IEEE Internet of Things Journal; Volume: 2 , Issue: 6; pp. 515 - 526, 2015
[7] Masoumeh Safkhani ; Nasour Bagheri ; Majid Naderi ; Yiyuan Luo ; Qi Chai; Tag Impersonation Attack on Two RFID Mutual Authentication Protocols; 2011 Sixth International Conference on Availability, Reliability and Security; pp. 581 - 584, 2011
[8] R. Priya ; S. Sivasankaran ; P. Ravisasthiri ; S. Sivachandiran; A survey on security attacks in electronic healthcare systems; 2017 International Conference on Communication and Signal Processing (ICCSP); pp. 0691 - 0694; 2017
[9] Zhang, C., Zhang, W. and Mu, H.; A Mutual Authentication Security RFID Protocol Based on Time Stamp; Journal of Computers; vol. 28, No. 2, 2017, pp. 223-229.
[10] Yong Ki Lee and Ingrid Verbauwhede; Secure and Low-Cost RFID Authentication Protocols; Los Angeles, 2005
[11] Wenjin Yu and Yixiang Jiang; Mobile RFID Mutual Authentication Protocol Based on Hash Function; International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Nanjing, China, 2017

[12] Wei, C. H., Hwang, M. S. and, Chin, A. Y.-H.; A Mutual Authentication Protocol for RFID, IEEE Computer Society, 2011