Medical information sharing system based on non-repudiation mechanism

T.T. Yang  
Department of Software and Information Science, Iwate Prefectural University, Iwate Ken, Takizawa, Japan

C.L. Chen  
Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung, Taiwan, ROC

G. Chakraborty  
Department of Software and Information Science, Iwate Prefectural University, Iwate Ken, Takizawa, Japan

ABSTRACT: In this era of rapid development of medical, the concerned issue is how to use medical resources effectively at present. Many literatures have been discussed about how to improve medical systems. However, they need to pay attention to security problems. About the other issues, how to protect patients’ privacy and use conveniently that is our concern. In this paper, we design a security scheme for healthcare environments that may be encountered. We also combine mobile device and cloud platform. In our scheme, the patient shares the healthcare information to an agent. Therefore, people can seek medical advice conveniently. Our scheme with the non-repudiation characteristic that can guarantee the information is legal. Our scheme also can defend against some known attacks.

1 INTRODUCTION
In Americans, there are at least 500,000 medical errors annually (Allison, L. 2014). That means even advanced countries have many medical errors, not to mention other countries. The other problem are that many patients concern the medical error when they in emergency department (Kianmehr, N. et al. 2012.). By these reasons, we focus on how to solve these problems and propose a security scheme. Many of researches also discuss medical issues. But, their scheme have some security vulnerability.

In 2012, Wei et al. (Wei, J. et al. 2012.) proposed an improved authentication scheme for telecare medicine information systems. But Zhu et al. (Zhu, Z. et al. 2012.) and Khan et al. (Khan, M. K. et al. 2013.) pointed Wei et al.’s scheme suffer from off-line password guessing attack, and Zhu et, al. also proposed an efficient authentication scheme for telecare medicine information systems. In 2015, Bin Muhaya (Bin Muhaya 2015.) found Zhu et al.’s scheme exists user's impersonation attacks.

Therefore, we should pay attention to avoid similar attacks. In our scheme, the medical information can share with agent who is chose by patient such that the medical information can be used more convenience.

The rest of this paper is organized as follows. Section 2 is our proposed scheme. In Section 3, we analyze the security of this scheme. Finally, we make a conclusion in Section 4.

2 THE PROPOSED SCHEME

2.1 System architecture
Our scheme divide into two phases. First, the initial phase that the patient shares the session key with the agent, uploads the healthcare report to the cloud and saves some information to the emergency reporting center. The architecture of the initial phase is shown in Figure 1.

![Figure 1. The architecture of the initial phase](image)

Then, the emergency reporting center will record the information and tell the medical staff the emergency information. This phase is called the emergency inform phase. The agent downloads the patient’s health report and sends it to the medical staff. Therefore, the medical staff can conduct the health information on the way to hospital. The architecture of the emergency inform phase is shown in Figure 2.
PKR users must register to the key

Figure 2. The architecture of the emergency inform phase

2.2 Notations

In this paper, we use pairing-based cryptography to design the authentication and use advanced encryption standard to encrypt/decrypt the message. The notations of this scheme are shown in Table 1.

Table 1. Notations of this scheme

<table>
<thead>
<tr>
<th>Notations</th>
<th>Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDi</td>
<td>the x’s identity</td>
</tr>
<tr>
<td>HRi</td>
<td>the x’s health reports</td>
</tr>
<tr>
<td>responsesi</td>
<td>the ith responses message</td>
</tr>
<tr>
<td>PSi</td>
<td>the patient’s status</td>
</tr>
<tr>
<td>GPSi</td>
<td>the location information</td>
</tr>
<tr>
<td>certi</td>
<td>the x’s certificate</td>
</tr>
<tr>
<td>ri</td>
<td>the ith random number</td>
</tr>
<tr>
<td>Gi</td>
<td>a cyclic additive group with the order q</td>
</tr>
<tr>
<td>Gj</td>
<td>a cyclic multiplicative group with the order q</td>
</tr>
<tr>
<td>e()</td>
<td>the pairing function e(): G1×G1→G2</td>
</tr>
<tr>
<td>H()</td>
<td>the hash function H: G1→{0,1}l, l=256</td>
</tr>
<tr>
<td>x</td>
<td>the KGC’s private key</td>
</tr>
<tr>
<td>skxy</td>
<td>the session key between x and y</td>
</tr>
<tr>
<td>Ti</td>
<td>the ith timestamp, it is generated by x</td>
</tr>
<tr>
<td>ΔT</td>
<td>the valid transmission time interval</td>
</tr>
<tr>
<td>E,i/IDi(M)</td>
<td>encrypt/decrypt a message M</td>
</tr>
</tbody>
</table>

2.3 The initial phase

In this phase, the patient choose a trust people who is to the agent. All users generate the session key with their agent as follows:

Step 1: The patient chooses a random number r1. The patient uses his/her private key SKp, the random number r1 and the agent’s public key PKA to compute the session key skp,A as follows:

\[ skp,A = H(e(SKp, PKA)^r_1) \] (1)

Then, the patient uses the agent’s public key PKA to encrypt the session key skp,A, the random number r1 and the timestamp Tp1 as follows:

\[ C_1 = E_{PKA}(skp,A, r_1, Tp_1) \] (2)

The patient sends C1 to the agent.

Step 2: After receiving the message, the agent decrypts the message C1 as follows:

\[ (skp,A, r_1, Tp_1) = D_{PKA}(C_1) \] (3)

The agent checks if the timestamp Tp1 is valid or not. If it is valid, the agent uses the random number r1 to compute the session key skp,A as follows:

\[ skp,A = H(e(SKp, PKA)^r_1) \] (4)

The agent checks if the session key skp,A is the same as skp,A or not. If it is true, the agent uses the session key skp,A to encrypt a response responses1 and the timestamp Tp2 as follows:

\[ res_1 = E_{skp,A}(responses_1, Tp_2) \] (5)

And the agent sends the message res1 to the patient.

Step 3: Upon receiving the message, the patient decrypts the message res1 as follows:

\[ (responses_1, Tp_2) = D_{skp,A}(res_1) \] (6)

And the patient checks if the timestamp Tp2 is valid or not. If it is valid, the patient uses the session key skp,A to encrypt a response responses2 and the timestamp Tp3 as follows:

\[ res_2 = E_{skp,A}(responses_2, Tp_3) \] (7)

And the patient sends the message res2 to the agent.

Step 4: After receiving the message, the agent decrypts the message res2 as follows:

\[ (responses_2, Tp_3) = D_{skp,A}(res_2) \] (8)

And the agent checks if the timestamp Tp3 is valid or not. If it is valid, the agent and the patient can use the session key skp,A to encrypt the information.

Then, the patient can upload the health report to the cloud as follows:

Step 5: The patient chooses a random number r2. The user uses his/her private key SKp, the random number r2 and the cloud’s public key PKC to compute the session key skp,C as follows:

\[ skp,C = H(e(SKp, PKC)^r_2) \] (9)

Then, the patient uses the session key skp,C to encrypt the health report HR as follows:

\[ C_{HR} = E_{skp,C}(HR) \] (10)

The patient can use the cloud’s public key PKC to encrypt the random number r2, the session key skp,C and the timestamp Tp4 as follows:

\[ C_{r_2} = E_{PKC}(r_2, skp,C, Tp_4) \] (11)

The patient sends C1 and CHR to the cloud.

Step 6: After receiving the message, the cloud uses its private key to decrypt the message C1 as follows:

\[ (r_2, skp,C, Tp_4) = D_{SKC}(C_{r_2}) \] (12)

The cloud checks if the timestamp Tp4 is valid or not. If it is valid, the cloud uses the random number r2 to
compute the session key \( sk'_{p-C} \) as follows:
\[
    sk'_{p-C} = H(e(SK_C, PK_p)^y)
\]  
(13)
The cloud checks if the session key \( sk_{p-C} \) is the same as \( sk'_{p-C} \) or not. If it is valid, the cloud uploads the message \( C_{HR} \).
The patient also need to submit the agent’s information to the emergency reporting center as follows:
Step 7: The patient uses the emergency reporting center’s public key \( PK_{ERC} \) to encrypt his/her identity \( ID_p \), the timestamp \( T_{ps} \), their agent’s identity \( ID_A \) and the agent’s information and also signs this message as follows:
\[
    C_{check} = E_{PK_{ERC}}(ID_p, T_{ps}, ID_A, Inf_A),
\]
(14)
\[
    Sig_p = S_{sk_{p-C}}(C_{check}).
\]
(15)
Then, the patient sends \( C_{check}, cert_p \) and \( Sig_p \) to the emergency reporting center.
Step 8: After receiving the message, the emergency reporting center decrypts the message \( C_{check} \) as follows:
\[
    (ID_p, ID_A, T_{ps}, Inf_A) = D_{SK_{ERC}}(C_{check}).
\]
(16)
The emergency reporting center checks if the timestamp \( T_{ps} \), the certificate \( cert_p \) and the signature \( Sig_p \) is valid or not. If it is valid, the emergency reporting center stores the agent’s information to record form.
As long as the patient suffers from an accident, the emergency reporting center can use this form to find the agent.

2.4 The emergency inform phase

In this phase, the case reporter can call the emergency reporting center when he/she finds the patient. The emergency reporting center also can find the agent, if the patient submitted the agent’s information to the emergency reporting center.
Step 1: The case reporter meets an accident and he/she calls the emergency reporting center. The case reporter uses the emergency reporting center’s public key \( PK_{ERC} \) to encrypt his/her identity \( ID_C \), location information GPS, the patient’s status \( PS \) and the timestamp \( T_{CR1} \) and also signs as follows:
\[
    C_{help} = E_{PK_{ERC}}(ID_C, GPS, PS, T_{CR1}).
\]
(17)
\[
    Sig_{CR} = S_{sk_{CR}}(C_{help}).
\]
(18)
Then, the case reporter sends \( C_{help}, cert_{CR} \) and \( Sig_{CR} \) to the emergency reporting center.
Step 2: After receiving the message, the emergency reporting center decrypts the message \( C_{help} \) as follows:
\[
    (ID_C, GPS, PS, T_{CR1}) = D_{SK_{ERC}}(C_{help}).
\]
(19)
The emergency reporting center checks if the timestamp \( T_{CR1} \), the certificate \( cert_{CR} \) and the signature \( Sig_{CR} \) is valid or not. If it is valid, the emergency reporting center uses record form to find the agent.
After the case was established, the emergency reporting center informs the medical staff to help the patient.
Step 3: The emergency reporting center uses the medical staff’s public key \( PK_{MC} \) to encrypt identity \( ID_{ERC} \), the case information \( Inf_{case} \) and the timestamp \( T_{ERC2} \) and also signs as follows:
\[
    C_{case} = E_{PK_{MC}}(ID_{ERC}, Inf_{case}, T_{ERC2}).
\]
(20)
\[
    Sig_{ERC} = S_{sk_{ERC}}(C_{case}).
\]
(21)
Then, the emergency reporting center sends \( C_{case}, cert_{ERC} \) and \( Sig_{ERC} \) to the medical staff.
Step 4: After receiving the message, the medical staff decrypts the message \( C_{case} \) as follows:
\[
    (ID_{ERC}, Inf_{case}, T_{ERC2}) = D_{SK_{MC}}(C_{case}).
\]
(22)
The medical staff checks if the timestamp \( T_{ERC2} \), the certificate \( cert_{ERC} \) and the signature \( Sig_{ERC} \) is valid or not. If it is valid, the medical staff goes to the incident’s location according the case information.
The medical staff requests the agent to upload the patient’s health report and the agent must verify the medical staff’s identity and his/her certificate.
Step 5: The medical staff chooses a random number \( r_3 \). The medical staff uses his/her private key \( SK_{MS} \), the agent’s public key \( PK_{A} \) and the random number \( r_3 \) to compute the session key \( sk_{MS-A} \) as follows:
\[
    sk_{MS-A} = H(e(SK_{MS}, PK_{A})^y).
\]
(23)
Then, the medical staff uses the agent’s public key \( PK_{A} \) to encrypt identity \( ID_{MS} \), the case information \( Inf_{case} \), the request \( request \), the session key \( sk_{MS-A} \), the random number \( r_3 \) and the timestamp \( T_{MS2} \) and also signs as follows:
\[
    C_{req} = E_{PK_{A}}(ID_{MS}, Inf_{case}, request, sk_{MS-A}, r_3, T_{MS2}).
\]
(24)
\[
    Sig_{MS} = S_{sk_{MS}}(C_{req}).
\]
(25)
Then, the medical staff sends \( C_{req} \) and \( Sig_{MS} \) to the agent.
Step 6: After receiving the message, the agent decrypts the message \( C_{req} \) as follows:
\[
    (ID_{MS}, Inf_{case}, request, sk_{MS-A}, r_3, T_{MS2}) = D_{SK_{A}}(C_{req}).
\]
(26)
The agent checks if the timestamp \( T_{MS2} \) and the signature \( Sig_{MS} \) is valid or not. If it is valid, the agent uses the random number \( r_3 \) to compute the session key \( sk'_{MS-A} \) as follows:
\[
    sk'_{MS-A} = H(e(SK_{A}, PK_{MS})^y).
\]
(27)
The agent also checks if the session key \( sk_{MS-A} \) is the same as \( sk'_{MS-A} \) or not. If it is valid, the agent logs in.
to the cloud according the request. The agent and the medical staff can use the session key for the future communication. The agent need download the patient’s health information. However, the agent can use the session key to decrypt the health information and send it to the medical staff.

Step 7: The agent uses the cloud’s public key \( PK_C \) to encrypt identity \( ID_A \), the case information \( Inf_{case} \) and the timestamp \( T_{AS} \) and signs as follows:
\[
C_{down} = E_{PK_C} \left( ID_A, Inf_{case}, T_{AS} \right).
\]

(28)

\[
\text{Sig}_A = S_{sk_A} \left( C_{down} \right).
\]

(29)

Then, the agent sends \( C_{down} \) and \( \text{Sig}_{MS} \) to the cloud.

Step 8: After receiving the message, the cloud decrypts the message \( C_{down} \) as follows:
\[
\left( ID_A, Inf_{case}, T_{AS} \right) = D_{sk_C} \left( C_{down} \right).
\]

(30)

The cloud checks if the timestamp \( T_{AS} \) and the signature \( \text{Sig}_A \) is valid or not. If it is valid, the cloud sends \( C_{HR} \) to the agent and records this download. The agent can use the session key with the patient to decrypt the patient’s health information. Afterward, the agent encrypts the patient’s health information.

Step 9: The agent uses the session key \( sk'_{PA} \) to decrypt the message \( C_{HR} \) as follows:
\[
HR = D_{sk'_{PA}} \left( C_{HR} \right).
\]

(31)

Then, the agent uses the session key \( sk'_{MS-A} \) to encrypt \( HR \) and the timestamp \( T_{AS} \) as follows:
\[
C_{Inf} = E_{sk'_{MS-A}} \left( HR, T_{AS} \right).
\]

(32)

Then, the agent sends \( C_{Inf} \) to the medical staff.

Step 10: After receiving the message, the medical staff decrypts the message \( C_{Inf} \) as follows:
\[
\left( HR, T_{AS} \right) = E_{sk_{MS-A}} \left( C_{Inf} \right).
\]

(33)

The medical staff checks if the timestamp \( T_{AS} \) is valid or not. If it is valid, the medical staff can use this report to help the patient before they arrive in hospital.

3 SECURITY ANALYSIS

3.1 Impersonation attack

In our scheme, one party use their private key and the other party’s public key to generate a authentication value as following Eq. (1), (4), (9), (13), (23) and (27). The attacker cannot get the private key to generate the valid authentication. So, our scheme can avoid the impersonation attack.

3.2 Man-in-middle Attack

That is means the attacker want to disguise communication sides each other. In our scheme, the attacker cannot disguise the second communication side. Because the first communication side knows who is the second communication side also and can get the valid second communication side’s public key. So the attacker cannot achieve the man-in-middle attack.

3.3 Replay attack

In our scheme, the messages are included the timestamps as following Eq. (2), (5), (7), (11), (14), (17), (20), (24), (28), and (32). The receiver also checks if the timestamp is valid or not. If it is invalid, the receiver found the message was replayed and rejected this communication. So, our scheme can avoid the replay attack.

3.4 Non-repudiation

The non-repudiation means someone cannot deny they send the message and the receiver can find who the sender is. In the emergency inform phase, the receiver gets the signature and also checks if the signature is valid or not. Because of the signature is included the information. The sender cannot deny they are the sender. So, our scheme have the non-repudiation mechanism.

4 CONCLUSION

In this paper, we proposed a medical information sharing with agent based non-repudiation mechanism. The patient can share the medical information to their trust people. We think that it conforms to reality life. Our scheme can avoid impersonation attack, man-in-middle attack and reply attack. Our scheme also have the non-repudiation mechanism.

ACKNOWLEDGEMENT

This paper was supported by the Ministry of Science and Technology of Taiwan under Grant MOST 103-2622-E-212-009-CC2, MOST 103-2632-E-324-001-MY3 and MOST 104-2221-E-324-012.

REFERENCES

Allison, L. 2014. Doctor errors kill 500,000 Americans a year. Newsmax health (newsmaxhealth.com).


