Digital Content Protection in Cloud Environments using Near Field Communication

D.Jothipriya¹, A.Akila²
¹MPhil Scholar, Vels University, Chennai
²Assistant Professor, Department of CS., Vels University, Chennai
Email Id: jothi.gopal@gmail.com

Abstract: In the cloud environment, the digital content accessed by the user is mostly stored in the server. Thus, it needs to prevent malicious persons from modifying information and faking the user's identity to obtain the digital content. In addition, in the cloud computing environment, it is required to consider the storage space and computing performance problems for the mobile device. In this paper, a secure and efficient NFC identity authentication and digital content protection scheme is proposed. We use an NFC tag to store the security token and other related parameters, and put the tags in a protected environment in order to prevent it from being damaged or tampered by malicious persons. Through the NFC side-channel, a user needs to access the NFC tag to access the security token. In addition, we use the challenge-response protocol and the message authentication code to achieve mutual authentication, and generate the session key to protect the transmitted digital content.

Keywords: Cloud Computing, NFC Tag, Identity Authentication, Digital Content Protection

I. INTRODUCTION

NFC[1],[7],[8] consists of three operating modes: Card Emulation mode, Peer-to-Peer mode, and Read/Write mode [8]. Card emulation mode in the application can be used as a digital wallet [9], travel cards and access control cards. Peer-to-Peer mode NFC devices can do the cross-platform data transmission, such as pictures, videos, music, contacts data transfer. In Read/Write mode, the data can be read from NFC tag. In this paper, for solving these problems, we propose a secure and efficient NFC identity authentication and digital content protection scheme in the cloud environment. We use the NFC side-channel approach [7] to store the security token and the related parameters in an NFC tag. In addition, in order to ensure the communication security, the NFC tag will be placed in the secure environment to prevent malicious persons from destroying the tag or modifying the tag [8]. When the user uses an NFC-enabled mobile device to read tag and obtain the security token, then the user can use this security token to communicate with the cloud server and to do the mutual authentication. In the authentication process, we use a challenge-response protocol to authenticate each other and generate the session key.

II. RELATED WORK

This section will review the related works.

A) Manfred Aigner et al.’s scheme
Manfred Aigner et al.’s [1] proposed an NFC security mechanism. By the NFC tags, it has cheaper cost and can use NFC target as coupons to use. The user can read an NFC tag to obtain coupons (pick-up of an mCoupon), and use coupons to buy the products or services at the cashier (cash-in of an mCoupon). The following is the used symbols in their method.
R: A random number.
K: A symmetric key.
E: The symmetric encryption function.
C: The client.
T: The target (issuer).
PK: An asymmetric public key.
SK: An asymmetric private key.
ClientAuth: The authentication message for the client.

B) Alfredo Matos et al.’s scheme
Alfredo Matos et al.’s [10] proposed a security mechanism to authenticate Wi-Fi hotspot by NFC side-channel. In public places, there would have some public Wi-Fi hotspot, but these hotspots may be fake and malicious. If the user does not validate the Wi-Fi hotspot, it will likely cause the user's information to be stolen. The following is the used symbols in their scheme.
KAP: The asymmetric public key used by the access point.
KAP: The asymmetric private key used by the access point.
KTPP: An asymmetric public key generated by TTP/CA.
KTPP: An asymmetric private key generated by TTP/CA.
SIGKTPP: The signed public key certificate by TTP/CA.
ESSID: Extended service set identification (ESSID) used by the access point.
PASSWPA2: Access point password using the WPA2 standard.
MACAP: Access point media access control address.
IPAP: Access point IP address.
R: A random number generated by the user.
x: A timestamp generated by the user.

III. METHODOLOGY

We propose an NFC security scheme based on the challenge/response method. This scheme can be used in cloud environment for identity authentication and digital content protection. Our scheme can be used for variety of mobile devices with limited storage and computation capacity. Our proposed scheme consists of four phases: the initialization phase, the reading phase, authentication phase, the access phase. The following is the used symbols in our method.

x: The master key generated by the server.
KI: The i-th security token generated by the server.
HMAC(): The hash message authentication code function.[5].
h(): The one-way hash function.
ID: The i-th identity of the tag generated by the server.
R: A random number generated by the user.
Ri: A random number generated by the server.  
Lifetimei: The valid period of the i-th security token.  
SK: The session key shared between the NFC-enabled mobile device and the server  
Esk(): The symmetric encryption function using the shared key sk[6].  
∥: The string concatenation operator.

A) Initialization phase
In the initialization phase, the cloud server generates the parameters and computes the security token. These parameters will be stored in NFC tag by NFC communication channel. The cloud server generates the parameters as follows:  
1) The cloud server generates the parameters (x,IDi,Lifetimei), and computes the security token $K_i = h(ID_i)II\|\text{Lifetime}_i$.
2) The cloud server stores the parameters (x,IDi,Lifetimei) and stores the parameters (IDi,Ki,Lifetimei) into the NFC tag by the NFC communication channel.

B) Reading phase
In the reading phase, we assume the NFC tag is placed in a secure and protected environment. The user uses a mobile device to near the NFC tag to obtain the parameters. The user uses an NFC-enabled mobile device to near the NFC tag to obtain the parameters (IDi,Ki,Lifetimei) and store the parameters in a mobile device.

C) Authentication phase
In this phase, the user uses the parameters obtained from the NFC tag to do mutual authentication with the cloud server. The user and the server will generate their random numbers in the authentication phase, and compute the hash message authentication codes to authenticate each other. The user generates a random number $R_1$, and sends (IDi,R1) to the cloud server. The cloud server receives the IDi to compute the security token $K_i$. The cloud server generates a random number $R_2$ and computes $v = \text{HMAC}(K_i,R_2)$. The cloud server sends (R2,v) to the user. The user computes $v' = \text{HMAC}(K_i,R_2)$, and checks if $v'$ is equal to $v$. If it is not equal, the user ends the authentication process. If it is equal, the user computes $w = \text{HMAC}(K_i,R_2)$, and send $w$ to the cloud server. The cloud server computes $w' = \text{HMAC}(K_i,R_2)$, and checks if $w'$ is equal to $w$. If it is equal, it accepts the authentication.

D) Access phase
In this phase, the user and the cloud server will compute the session key, and use the symmetric encryption to achieve digital content protection. When the user and the cloud server finish mutual authentication, the cloud server will compute the session key $SK = h(K_i)\|R_1\|R_2$. The cloud server can use the session key SK to encrypt the digital content, and sends $E_{SK}[E \text{–content}]$ to the user. After the user receives $E_{SK}[E \text{–content}]$, she/he computes the session key $SK = h(K_i)\|R_1\|R_2$ and decrypts $E_{SK}[E \text{–content}]$ to obtain the digital content $E \text{–content}$.

IV. PERFORMANCE AND SECURITY ANALYSIS

A) Security analysis
In this section, we will discuss the security and functionality of our proposed scheme. The security analysis consists of mutual authentication, being suitable for diverse mobile devices in cloud environment, no password table, no time-synchronization problem, preventing the replay attack, session key agreement, preventing the dictionary attack and the brute force attack, and the user's identity privacy protection. Table 1 presents the functionality comparison between our scheme and Alfredo Matos et al.'s scheme[10].

Table 1: The functionality comparison

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our scheme</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Matos et al.'s</td>
<td>Yes</td>
<td>N/A</td>
<td>N/A</td>
<td>Yes</td>
<td>N/A</td>
</tr>
</tbody>
</table>

P1: Mutual authentication  
P2: No password table  
P3: No time-synchronization problem  
P4: Preventing the replay attack  
P5: Session key agreement  
N/A: Not available

a) Mutual authentication
In order to prevent the attacker from faking a user to log on the server to access sensitive information, the user and the cloud server must do mutual authentication to ensure their real identities. In Section 3.3.4, after the user receives (R2,v) from the cloud server, the user checks that $v$ is equal to $\text{HMAC}(K_i,R_2)$. If it is true, the user authenticates the cloud server since the authentication tag $\text{HMAC}(K_i,R_2)$ is computed from the key $K_i$ shared between the cloud server and the user. In Section 3.3.5, after the cloud server receives $w$ from the user, the cloud server computes $w'$ and then checks that $w'$ is equal to $\text{HMAC}(K_i,R_2)$. If it is true, the cloud server authenticates the user since the authentication tag $\text{HMAC}(K_i,R_2)$ is computed from the key $K_i$ shared between the user and the cloud server. Use an NFC tag to store the related parameters. Thus we can reduce the cost of storage and computing to suitable for the diverse mobile devices on cloud environments.

b) No password table
Typically, the cloud server needed to create a password table to store the user's password for authentication. Thus, when the cloud server has a password table, it will increase the storage cost on the server's database and the risk of the insider attack. In order to prevent these problems, the cloud server does not require a password table to store the user's password in our proposed scheme since ever NFC tag can get the security token generated by the server. Thus, the cloud server and the server can use the security token to do authentication each other.

In Section 3.3, after the cloud server receives (IDi,R2) from the user, the cloud server computes the security token $K_i = h(ID_i)\|\text{Lifetime}_i$ and the user can use a random number $R_2$ and the security token $K_i$ to compute the message authentication code and sends it to the cloud server. At this point, the cloud server can compute $w' = \text{HMAC}(K_i,R_2)$ and check if $w'$ is equal to $w$ for the user identity authentication. It replaces the use of a password to verify the identity. Thus, our proposed scheme does not require a password table to store passwords.
In our proposed scheme, we use nonce to complete authentication phase since we want to avoid the time-synchronization problem. In Section, the cloud server and user generates the random number R1 and R2. Then, the cloud server and user uses these random numbers and the security token Kt to compute the hash message authentication codes to do the mutual authentication. Based on this operation, our proposed scheme can solve the time-synchronization problem since our proposed scheme is a nonce-based authentication scheme.

d) Preventing the replay attack
In our proposed scheme, we use the challenge-response authentication method to prevent the replay attack. In Section after the cloud server receives the parameters (ID1, R1), the cloud server returns (R2, HMAC(K, R2)) to the user. When the user verify the response HMAC(K, R2) received from the cloud server, the user return the response HMAC(K, R2), to the cloud server in Section 3.3.4. Thus when the cloud server verify the response HMAC(K, R2), the authentication phase will completed. Based on this processes, reusing the random number R1 and R2 will not pass the authentication since the random number R1 and R2 can only be used once. Thus, our proposed scheme can prevent the replay attack.

e) Session key agreement
In our proposed scheme, after completing the authentication, the user and the cloud server can compute the session key SK = h(K||R1||R2) as the digital content encryption and decryption key. Every session key of the digital content will be different in a different authentication phase. In our proposed scheme, the session key SK contains the parameter h(K||R1||R2) where K = h(ID||x||Lifetime,) shared with the cloud server and the NFC tag in Initialization phase, and R1, R2 is a random number chosen by the cloud server and user in Authentication phase.

B) The communication and computation cost
In this section, we will discuss the communication, storage and computation cost of our scheme.

a) The storage cost and communication cost comparison
In this section, we compare the communication cost and the storage cost. The detail comparison is illustrated in Table 2. In P1 of Table 2, our scheme has the parameters(ID, Kt, Lifetime). The memory length is 32 + 128 + 32 = 192 bits. Alfredo et al.’s scheme [18] has the parameters (KAP, Cert(KAP, SgK), ESSID, PASS, MACAP, IPAP). The memory length is 1024 + 1024 + 256 + 63 + 48 + 32 = 2447 bits.

In P2 of Table 2, our proposed scheme has the parameters (ID, R1, R2, HMAC(K, R1), HMAC(K, R2)). The memory length is 32 + 64 + 64 + 128 + 128 = 416 bits. Alfredo et al.’s scheme [18] has the parameters (E(KAP (nonce, TS)), E(nonce, TS)). The memory length is 1024 + 64 + 32 = 1120 bits. In P3 of Table 2, the user’s device memory has the parameter SK. The memory length is 128 bits. Alfredo et al.’s scheme [18] has the parameter KAP. The memory length is 1024 bits.

Table 2: The communication cost and the storage cost comparison

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our scheme</td>
<td>192 bits</td>
<td>416 bits</td>
<td>128 bits</td>
</tr>
<tr>
<td>Alfredo Matos et al.’s</td>
<td>2447 bits</td>
<td>1120 bits</td>
<td>1024 bits</td>
</tr>
</tbody>
</table>

P1: Communication cost of the reading phase for related parameters
P2: Communication cost of the authentication phase for related parameters
P3: Memory size for storing the cryptographic parameters in the user’s mobile device

b) The computation cost comparison
For comparing the computation cost, we quotes [14, 15, 24] as the references and assume that Hash=0.4M, Sym = 0.4M, and EXP= 240M, where Hash is defined as the time cost of the hash function operation, Sym is defined as the time cost of the symmetric encryption or decryption operation, and EXP is defined as the time cost of the exponential operation. The detail comparison is illustrated in Table 3. In P1 of Table 3, our scheme needs 1 Hash operation. Alfredo et al.’s scheme [10] needs 1 exponential operation.

In P2 of Table 3, our scheme needs 5 Hash operations. Alfredo et al.’s scheme [10] needs 2 exponential operations. In P3 of Table 3, our scheme needs 2 Hash operations and 2 Sym operations. Alfredo et al.’s scheme [1] do not have the access phase, so it is not available.

Table 3: The computation cost comparison

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our scheme</td>
<td>1Hash =0.4M</td>
<td>5Hash =2M</td>
<td>2Hash+2 Sym=1.6M</td>
</tr>
<tr>
<td>Alfredo et al.’s</td>
<td>1E=240M</td>
<td>2E=480M</td>
<td>N/A</td>
</tr>
</tbody>
</table>

P1: The communication cost of the initialization phase for the cloud server
P2: The communication cost of the authentication phase for the user’s device and the cloud server
P3: The communication cost of the access phase for the user’s device and the cloud server
Hash: Hashing operation
Sym: The symmetric encryption or decryption operation
E: The exponential operation
M: The modular multiplication operation
N/A: Not available

V. CONCLUSION
In this paper, we have proposed a secure and efficient NFC identity authentication and digital content protection mechanism in the cloud environment. Our proposed scheme uses the NFC as a side-channel to store security token in order to ensure the physical security for the authentication. In addition, we use a challenge-response authentication mechanism to compute the hash message authentication codes to achieve the mutual authentication, and prevent the message from being forged and replayed. In our proposed scheme, the cloud server does not need a password table, thus our proposed
scheme can prevent the dictionary attack and the insider attack. In addition, the user and the cloud server can generate a session key to access the digital content. Our proposed scheme can reduce the computation cost and the communication cost and provide an NFC identity authentication and digital content protection in the cloud environment.

References
[9] Innovation Research and Technology (No Date), "NFC in real world: Turning the NFC promise into profitable, everyday applications” available at: http://innovation-group.com (Access date: May 27, 2011)