A Variant-based Biometric Authentication Scheme Based on Rotor Machine for Home Security

Ping Wang*  Chih-Chiang Ku

Department of Management Information System, Kun Shan University, Tainan 710, Taiwan, ROC

Received 6 Oct 2008; Accepted 13 Mar 2009

Abstract

Multifactor authentication methods have been developed to raise the security of user authentication by integrating smart card, fingerprint and password. Nevertheless, currently available authentication schemes cannot effectively avoid specific attacks, such as authentication data being stolen by internal staff. For example, the attacker can get the fingerprints of the user from the imprint cup of biometric capture sensors. Therefore, we propose a variant-based remote authentication scheme using rotor machine concept where the biometric template is encrypted and altered arbitrarily by users to avoid the fingerprint template being stolen. If the encrypted fingerprint template is lost, the user can set a novel string and generate a new encrypted one, and the original template becomes invalid, so hacker or internal staff cannot use it as a legal identity to pass the liveness test. Furthermore, we show that the proposed scheme is a secure scheme and is immune to four types of network attacks via security analysis for smart home applications.

Keywords: Authentication, Remote authentication, Biometrics, Rotator

1. Introduction

The applications of remote real-time monitoring in smart homes offer many benefits to householders; however, they also might bring many security worries. The securing of online accounts against malicious activity, such as identity theft, burglary and counterfeit, is imperative to householders. Due to the non-denial requirements of remote identity authentication for users, multifactor authentication methods have become increasing popular. These methods, which combine biometrics with passwords, storing encrypted authentication value into smart cards or authentication servers, have been employed in many online applications of user authentication, such as two-factor authentication or three-factor authentication. When designing a new authentication mechanism, engineers not only have to take into consideration with rigid security features, but also must defend against attack events, such as biometric features being counterfeited by dishonest internal employees in the registration phase. Obviously, it could be a disaster if biometric data of users were copied that became inactive forever.

In a pioneering work, Lee et al. proposed a remote identity authentication scheme combining fingerprint and smart cards [1]. The major feature of this approach is to increase the security of the ElGamal public key system; encryption parameters are randomly generated by both the user’s fingerprint minutiae and time stamp. Recently, Fan et al. declared a truly three-factor remote authentication scheme that is a technically sound approach for three-factor authentication, as shown in Fig. 1 [2]. Their features of scheme are (i) randomly choose a string $r$ (cannot change after selection) and mix the biometric characteristics of a user with the string by performing the exclusive-or operation ($\text{XOR}$). (ii) The smart card combines the randomly chosen number with the fingerprint minutiae data input in the login phase. Then, the two combined strings are sent to the authentication server for matching.

We questioned some security issues regarding these approaches. Staff person or hacker might copy the fingerprint from imprint cup for each scheme. In Fan’s approach, the fingerprint template will be mixed by a randomized string $r$ with $\text{XOR}$ operation for encryption; however, internal staff might catch this string. The weakness of the scheme is the invariance of features for this randomized string $r$ after users have been selected, such that leads to the consequence that fingerprint minutiae might be recovered if string $r$ is disclosed in the template generation process of the registration phase. For example, in situations of application of user certificate or when the certificate is expired. In other words, Fan et al.’s scheme preserves the privacy of the users in the login and authentication process, whereas staff with bad intentions might take users’ biometrics away in the registration phase [2].

In this research, we enhance the authentication approach of Fan et al. based on a rotor machine whose architecture is
2. A variant biometrics-based authentication scheme

The variant biometric authentication protocol is illustrated in Fig. 3. The proposed protocol consists of three phases: the initialization phase, the registration phase and the authentication phase, where the initialization phase is to set the related parameter for encryption. In the registration phase, the user randomly selects a string $r$ and determines the value of user password $PW_i$. In the following phase, each registered user uses the keypad and fingerprint scanner to generate a fingerprint template and store it into the smart card for matching.

In the authentication phase, each registered user uses the keypad and fingerprint scanner to generate a fingerprint template, as shown in Fig. 4, and then get an extract fingerprint template from the user’s smart card for matching. Finally, the terminal matches two identity templates; the comparison result will be sent to the server to re-verify whether the user is authorized or not, this step can prevent counterfeit attack. The details of the three phases are described below.

2.1 Initialization phase

Step 1.1: Card issuer randomly chooses a large prime number $p$ and its root $\alpha$.
Step 1.2: Randomly generates a 128-bit string $K$ as its key for symmetric encryption, and keeps $(p, \alpha, K)$ secret.

2.2 Registration phase

Step 2.1: Let user $U_i$ with identity $ID_i$ be about to register with the server. The user inputs a random string $r$ via keypad and determines the value of a card password $PW_i$ and then $r, PW_i$ are stored in the smart card.

Step 2.2: The user $U_i$ obtains her/his fingerprint via a sensor and then extracts the minutiae from the fingerprint image to form a fingerprint template $F$, representing the fingerprint template...
of $U_i$.

Step 2.3: The terminal computes $EF_i^* = F_i^* \oplus r$, and $HEF_i = h(EF_i^*)$, where $h(\cdot)$ is a public one-way hash function.

Step 2.4: The terminal sends $(ID, p, a, K, HEF_i)$ back to the server via a secure channel.

Step 2.5: The terminal keeps $(ID, PW_i, EF_i)$ secret.

2.3 Authentication phase

The user inputs her/his fingerprint and password, and then a login request is sent to the authentication server. The fingerprint will be checked according to the following seven steps.

Step 3.1: $U_i$ inputs her/his password $PW_i^*$ into the terminal; if correct, then extract $K$, else reject the request.

Step 3.2: Employ Diffie-Hellman key exchange algorithm [3] to form a session key $SK$, where $SK$ is a shared secret between terminal and server. The detail of Diffie-Hellman key exchange algorithm is depicted as [6]:

Terminal randomly selects a number $X_i$ such that $X_i < p$, then computes $Y_i = a^{X_i} \mod p$ and $EY_i = E_a(ID, || Y_i)$, where $p$ and $a$ are stored in the card, then sends $EY_i$ to server. Similarly, server also randomly selects a number $X_\theta$ such that $X_\theta < p$, then computes $Y_\theta = a^{X_\theta} \mod p$ and $EY_\theta = E_a(Y_\theta)$, then sends $EY_\theta$ to terminal. The server decrypts by performing $D_h(EY_\theta)$, and obtains $Y_\theta$ from the decryption result with a prefix $ID$, then computes the session key $SK = (Y_\theta)^{X_i} \mod p$; similarly, the terminal also decrypts by performing $D_h(EY_i)$, and obtains $Y_i$ from the decryption result, then computes the corresponding session key $SK = (Y_\theta)^{X_i} \mod p$, where $SK$ is a shared secret between terminal and server.

Step 3.3: Let $F_t^*$ represent the input fingerprint data generated by performing the process of minutiae extraction via the sensor. The terminal computes $EF_t^* = F_t^* \oplus r$, where $EF_t^*$ represents a biometrics template of a user.

Step 3.4: The server generates a one-time symmetric key $RK$, then computes $M = E_{SK}(RK)$ and sends $M$ to the terminal, where $E_{SK}(\cdot)$ denotes a symmetric encryption function, such as AES method, using the session key $SK$ generated from Step 3.2.

Step 3.5: The terminal extracts $EF_i$ from the smart card and acquires $RK$ by performing decryption process $D_{SK}(M)$, where $D_{SK}(\cdot)$ denotes the symmetric decryption function using $SK$.

Step 3.6: The terminal matches between $EF_i^*$ and $EF_i$, if successful, then an authorized user is identified, else not. The terminal sends $RM = E_{SK}(h(EF_i^*)||CM)$ to the server for reconfirmation, where $E_{SK}(\cdot)$ denotes a decryption function using the key $RK$, and $CM$ denotes the message of the matching result.

Step 3.7: Finally, the server will re-verify whether $h(EF_i^*) = HEF_i$, or not. If true, the server accepts the login request of $U_i$, otherwise it rejects.

3. Discussion

3.1 Security analysis

To inspect the risk of resisting attacks by internal staff or external networks, four types of network attacks on our schemes were analyzed according to previous studies.

Suppose hackers intend to attack the communication sessions between a client and server by means of interception, fabrication, modification, and masquerade the packet data on the Internet. There are four formal types of network attacks: (i) man in the middle attack; (ii) offline dictionary attack; (iii) eavesdrop attack; (iv) replay attack.

When analyzing type (i), (ii) and (iii) attacks, we assumed that hackers might intercept the encrypted data in the communication session to acquire meaningful content, such as parameter $EF_i$ and $RK$, to carry on a user authentication illegally. However, the data transmission is protected by session key generated from the Diffie-Hellman key exchange algorithm (see Step 3.2). Therefore, hackers cannot successfully steal the complete biometric data. Actually, whatever hackers want to decrypt the encrypted packet data, they have to solve the Discrete-Logarithm Problem (DLP), which Diffie and Hellman proved is hard to accomplish in a finite period of time [4]. In addition, the session key will change once in a while, such that it adds difficulty to decrypting it. If the packet data is decrypted, hackers only acquire an encrypted fingerprint template, because they do not have string ($r$) and password of user ($PW_i$), to reproduce the original fingerprint template. Hence our scheme is safe even if they have the superior skill to solve the encryption transmission packet data on the Internet, where the key varies in each session.

We next assumed that hackers directly send an authentication outcome——“legal user”——to the server, bypassing all proper procedures, to make an identity counterfeit for deceiving the server. Due to the reason that the terminal has to transmit back two parameters, $h(EF_i)$ and $CM$, to the server in step 3.7, it is intractable for the hackers to decrypt $M= E_{SK}(RK)$ without acquiring the session key $SK$, therefore this attempt does not work. Furthermore, it is assumed that hackers have attained the formation of the terminal in steps 3.1-3.3 and employed a replay attack (Type iv) to counterfeit an legal identity in the authentication process. In step 3.4, one-time private key with 128-bit length, $R_{sk}$, will be randomly generated by the server for each authentication. It will become invalid in next session such that hackers cannot re-submit the previous message to the server by intimating the user to log in the server legally.

<table>
<thead>
<tr>
<th>Step</th>
<th>Executor</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3</td>
<td>Terminal</td>
<td>Compute $EF_i^* = F_i^* \oplus r$</td>
</tr>
<tr>
<td>3.4</td>
<td>Server</td>
<td>Generate a one-time private key $RK$</td>
</tr>
<tr>
<td>3.5</td>
<td>Terminal</td>
<td>Compute $M = E_{SK}(RK)$ and send $M$</td>
</tr>
<tr>
<td>3.6</td>
<td>Terminal</td>
<td>Decrypt $D_{SK}(M)$ to obtain $RK$ and extract $EF_i$ from smart card</td>
</tr>
<tr>
<td>3.7</td>
<td>Server</td>
<td>Verifies $(h(EF_i^*)) = HEF_i$</td>
</tr>
</tbody>
</table>
Table 2. Comparison with the related schemes (revised from Lee et al. [1]).

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Computational cost of operations during login and authentication phase</td>
</tr>
<tr>
<td></td>
<td>Store complete biometric template in either side</td>
</tr>
<tr>
<td></td>
<td>Clock synchronization</td>
</tr>
<tr>
<td>Our scheme</td>
<td>4E+4SE+4SD+2H+2S</td>
</tr>
<tr>
<td>Fan et al. [2]</td>
<td>E+3SE+4SD+4H+3X</td>
</tr>
<tr>
<td>Kim et al. [5]</td>
<td>4E+2H</td>
</tr>
<tr>
<td>Timestamp-based</td>
<td>Server NO</td>
</tr>
<tr>
<td>Kim et al. [5]</td>
<td>4E+1H</td>
</tr>
<tr>
<td>Nonce-based</td>
<td>Server NO</td>
</tr>
<tr>
<td>Lee et al. [1]</td>
<td>7E+2H+2X</td>
</tr>
</tbody>
</table>

*Notes: E represents computation time performing a modular exponentiation; SE denotes computation time to execute a modular symmetric encryption; SD indicates the calculation time carrying out a modular symmetric decryption computation; H is the time to perform a one-way hash function; X is the time accomplishing a modular XOR operation; S is the time to perform a string-mapping operation of a rotor machine.

3.2 Efficiency analysis

To examine the execution efficiency of our proposed scheme, computational complexity was analyzed by comparing with Fan et al. [2], Kim et al. [5] and Lee et al. [1]. The analysis of execution efficiency of all schemes in the authentication phase is shown in Table 2. From Table 2, we know that the computational complexity of our scheme is a little bit greater than that of Fan et al. Notably, our scheme adds three exponential operations, one symmetric encryption operations and two string-mapping operation, but eliminates two hash operations and three XOR operation compared to Fan et al.’s scheme. In summary, our scheme designs a security approach comparable with Fan et al.’s approach, while needing the extra computations for string-mapping with authentication data.

5. Conclusions

In this paper, we have proposed a secure biometric authentication protocol via a variant string mixed with biometric data based on a rotor machine in remote authentication for smart home security. From the analysis results, our scheme is capable of decreasing the risk of fingerprint template being stolen and counterfeited; effectively preventing four types of the familiar network attacks.

Acknowledgements

This research was partially supported by grant No. 098-N-570-EDU-T-036 from the Ministry of Education and National Science Council project: TWISC@NCKU, NSC 98-2219-E-006-001 and NSC 98-2219-H-168-001.

References