A Proposal of Efficient Remote Biometric Authentication Protocol

Taiki Sakashita[1], Yoichi Shibata[2], Takumi Yamamoto[2], Kenta Takahashi[3],
Wakaha Ogata[4], Hiroaki Kikuchi[5], and Masakatsu Nishigaki[2]

[1] Graduate School of Systems and Information Engineering, University of Tsukuba,
1-1-1, Tenoudai Tsukuba science city, Ibaraki 305-0006, Japan
sakashita@cipher.risk.tsukuba.ac.jp

[2] Graduate School of Science and Technology, Shizuoka University,
3-5-1 Johoku, Hamamatsu-shi, Naka-ku, Shizuoka-ken, 432-8011, Japan
f5745037@ipc.shizuoka.ac.jp, nisigaki@inf.shizuoka.ac.jp

[3] Hitachi, Ltd., System Development, Lab.,
292, Yoshida-cho, Totsuka-ku, Yokohama-shi, Kanagawa, 244-0817, Japan
kenta.takahashi.bw@hitachi.com

[4] Graduate School of Innovation Management, Tokyo Institute of Technology,
2-12-1 O-okayama, Meguro-ku, Tokyo, 152-8552, Japan
wakaha@mot.titech.ac.jp

[5] School of Information Technology and Electronics, Tokai University,
1117 Kitakaname, Hiratsuka, Kanagawa, 259-1292, Japan
kikn@tokai.ac.jp

Abstract. ZeroBio has been proposed for a secure biometric authentication over the network by conducting secret computing between prover and verifier. The existing ZeroBio are based on zero-knowledge proof that a committed number lies in an interval, or on oblivious neural network evaluation. The purpose of ZeroBio is to give verifier a mean to authenticate provers with perfectly concealing provers'biometric information from verifier. However, these methods need high computational complexity and heavy network traffic. In this paper, we propose another type of ZeroBio protocol that can accomplish remote biometric authentication with lower computational complexity and lighter network traffic by tolerating small decline of security level.

Keywords: biometrics, authentication, zero knowledge interactive proof, secret computing.

1 Introduction

Recently, biometric authentication has been applied to our daily life, and its application range and usage amount have kept growing. In contrast to conventional authentication with password or security token, biometric authentication has an advantage that they don't suffer from forgetting password and loss of token. However, it is needed for biometric authentication to store prover's biometric information to verifier as a template. Since biometric information is unique and

T. Takagi and M. Mambo (Eds.): IWSEC 2009, LNCS 5824, pp. 212–227, 2009.
© Springer-Verlag Berlin Heidelberg 2009

unchangeable over the life time of the individual, it could be a serious problem of privacy if the prover's biometric information and/or templates are compromised. To cope with the leakage of biometric information from the prover's side, the use of biometric information which will not remain and can not be lifted (e.g., veins of the finger or the palm) is recommended. On the other hand, the protection of the templates stored in verifier's side may be more serious. Particularly, when biometric authentication is carried out over the network, verifiers are not always trusted (e.g., phishing site) and giving biometric information as it is to verifiers is not considered to be secure. Therefore protecting biometric templates is an essential issue.

To solve this problem, Ratha et al. introduced the concept of cancelable biometrics in which biometric information in transformed form is stored and verifies it in transformed space [1], and proposed an image block transformation and a minutia nonlinear transformation. Cambier et al. also proposed a method transforming the iris date by rotating and distorting [2]. Hirata et al. proposed a transformation for two-dimensional image matching based biometrics [3]. Cancelable biometrics makes it possible to (i) protect biometric information by storing it in transformed form as a template and (ii) update the template by alternating the transforming function, or the random numbers used in the transforming function. However, there is a concern in cancelable biometrics that the matching score (the difference between biometric information presented at the authentication phase and the template) is not concealed from the verifier, which could be a potential vulnerability such as hill-climbing attack [4].

Nagai et al. proposed the concept of asymmetric biometric authentication, or ZeroBio, where information stored by prover and verifier are asymmetric [5]. They show an authentication method with neural networks that can authenticate prover through zero knowledge interactive proof (ZKIP) without revealing prover's biometric information even to the verifier. Ogata et al. also proposed another ZeroBio which is based on ZKIP to prove that difference between prover's biometric information and stored template is sufficiently small [6]. Both methods above can perfectly conceal provers'biometric information from verifier, but they also have shortcomings that they need high computational complexity and heavy network traffic. Therefore in this paper, we propose a different type of ZeroBio with lower computational complexity and lighter network traffic by tolerating small decline of security level. Our proposed method calculates the difference between the presented biometric information and the enrolled biometric information with secret computing based on the encryption function with a property of homomorphism. Then the significance of the difference is checked secretly and efficiently by using blinded decryption and hash function.

The remainder of this paper is organized as follow. In Section 2, we discuss remote biometric authentication model. In Section 3, we describe related works. In Section 4, we propose an asymmetric biometric authentication protocol based on secure computation, blinded decryption and hash function. In Section 5, we discuss security evaluation, and show our method has superior in computationalcomplexity and network traffic compared to other asymmetric biometric authentication presented in Section 3. In Section 6, we show an improvement to our protocol. Finally, we conclude our study in Section 7.

2 Remote Biometric Authentication Models

Remote biometric authentication model is classified into server (verifier) authentication model and client (prover) authentication model according to where templates are stored. Templates for all clients are managed centrally by a server for the server authentication model, while the template for each client is stored individually in client's smart card for the client authentication model. Although the client authentication model has an advantage of lower privacy concern, it is reported that information stored in a smart card can be revealed with good accuracy by side-channel attack [7]. Therefore this paper targets and discusses the server authentication model.

One of the biggest issues in the server authentication model is privacy. Obviously, it is not desirable in a sense of privacy to store and/or present biometric information to server without encryption. In this paper, we propose a remote biometric authentication method which can verify the authenticity of biometric information by conducting secret computing between prover and verifier. Our proposed method requires clients to have a smart card to carry helper information such as an encryption key to conceal biometric information itself from server. Note that it is impossible to derive biometric information from the helper information stored in the smart card.

3 Related Works

3.1 Cancelable Biometrics

In cancelable biometrics proposed in [2,3], the biometric information is masked by a random number, and then, the masked information is stored in server as a template. For security reason, the random number used for masking is needed to have a certain level of entropy, and to be stored in a smart card carried by authorized user. Biometric information presented at the authentication phase is also masked by the same random number, and compared with the template (biometric information masked by the random number). Therefore it is important to select proper masking methods appropriate for the comparison of target biometric information.

These methods mask the template by a random number, and thus no biometric information will leak out even if the templates are compromised. Also, in these methods no information except for the random number is stored in a smart card, so biometric information will not leak out even if the smart card is stolen. However, these methods allow server to compute the difference between masked biometric information presented at the authentication phase and the masked template to verify the authenticity of presented biometric information. Therefore, the server can get information of the difference of two biometric information.

3.2 ZeroBio Proposed by Nagai et al.

Nagai et al. proposed a method that can prove the authenticity of user's biometric information while perfectly concealing the biometric information by using oblivious neural networks evaluation [5]. We call the method Nagai scheme.

At enrollment phase, user trains his/her neural network with a set of feature extracted from his/her own biometric information and a set of feature for other users. Throughout the training, the weights of neural network are adjusted so that the neural network can output 1 for authorized user's biometrics information and 0 for unauthorized user's biometric information. After training, the weights of the output layer $\overline{w_j}$ and the commitments of weights of hidden layer $Com(w_{ij})$ in the neural network are enrolled.

At authentication phase, user is authenticated if the user can prove by zero knowledge interactive proof (ZKIP) that the neural network outputs 1 when his/her biometric information are inputted to the neural network without revealing his/her private biometric information. Note that the input biometric features are not exactly identical to that used to train the neural network. The variations will be absorbed by the property that neural networks can accept similar inputs.

3.3 ZeroBio Proposed by Ogata et al.

In cancelable biometrics, biometric information is masked by random number to generate template, while Fuzisaki-Okamoto commitment [8] is used for masking in Ogata et al's method [6]. We call the method Ogata scheme.

At enrollment phase, authorized user computes $E = Com(x, r)$, commitment of biometric information x, and stores it in server as a template. Then, random number r is stored in the user's smart card. From the characteristic of commitment, biometric information will not leak out from the template.

At authentication phase, user computes $E' = Com(x', r')$, commitment of presented biometric information x', and transmits it to the server. The server can calculate the commitment of $x - x'$ by secret computing. Then, the user conducts zero knowledge interactive proof protocol (ZKIP) which proves "difference between two committed biometric information is sufficiently small"

Note that the enrolled biometric information x is not stored in the authorized user's smart card. This means that the authorized user can not calculate the difference between x and x' in the authentication phase. Therefore, in Ogata scheme, the authorized user generates $2\theta + 1$ estimated values $\tilde{x} \in \{x', x' \pm 1, x' \pm 2, \ldots, x' \pm \theta\}$ from the presented biometric information x', and uses $\tilde{x} - x'$ instead $x - x'$ when proving the difference between x (committed in E) and x' (committed in E') calculated by the server is in the (small) interval $[-\theta, \theta]$ using "ZKIP for proving interval."In other words, ZKIP for proving interval composes of $2\theta + 1$ proofs. The server accepts authentication if at least one of $2\theta + 1$ proofs is accepted.

4 Proposed Method

4.1 Elemental Technique

Proposed method uses the (slightly modified) ElGamal encryption which has homomorphism to encrypt biometric information.

Let p be a large prime and g be a primitive element of Z_p^*. The user chooses a random integer s from $1 \leq s \leq p-1$ as a secret key, which is kept secret. Then the user computes $y = g^s$ in Z_p^*. Public key of the user is y, p and g.

The ciphertext $Enc(x)$ of biometric information x is computed as $Enc\,(x) = (g^r, g^x \cdot y^r) = (G, M) \pmod{p}$. Here, $r \in Z_p$ is a random number. The decryption is done by $g^x = M/G^s \pmod{p}$.

It is important to note that the encryption function has a property of homomorphism. For two ciphertexts $Enc(x_1) = (G_1, M_1)$ and $Enc(x_2) = (G_2, M_2)$, let $Enc(x_1) \cdot Enc(x_2)$ be defined as $(G_1 \times G_2, M_1 \times M_2)$. Then we have $Enc(x_1) \cdot Enc(x_2) = Enc(x_1 + x_2)$. Similarly, we have
$Enc(x_1)/Enc(x_2) = (G_1/G_2, M_1/M_2) = Enc(x_1 - x_2)$.

In this way, anyone can compute a ciphertext of difference between two biometric information x_1 and x_2 without decrypting $Enc(x_1)$ nor $Enc(x_2)$.

4.2 Outline

We propose an asymmetric biometric authentication which can be executed with lower computational complexity and lighter network traffic than these Nagai scheme and Ogata scheme.

Our method consists of enrollment phase and authentication phase. Authentic biometric information is encrypted and submitted to the server at the enrollment phase. At the authentication phase, the user encrypts his/her biometric information and sends it to the server. The server computes a ciphertext of difference between enrolled biometric information and presented biometric information using secret computing based on homomorphism of the encryption function. Then the server multiplies the ciphertext by a blind constant and sends it back to the user. The user proves to the server that the decryption of the blinded ciphertext (difference of two biometric information) is smaller than the threshold without disclosing the decryption to the verifier. In this paper, for simplifying explanation, we assume that difference between biometric information is formularized by absolute value of difference.

The outline of the proposed method is shown in Fig. 1.

4.3 Authentication Method

Authorized user and server share the following common parameters: prime number p, primitive root $g \in Z_p^*$, hash function $Hash()$, threshold θ, set of possible difference $\Delta = \{0, \pm 1, \pm 2, \cdots, \pm\theta\}$. If difference between two biometric information is in Δ, then two biometric information are considered to be sufficiently close. Every calculations in the protocol are computed in Z_p^*.

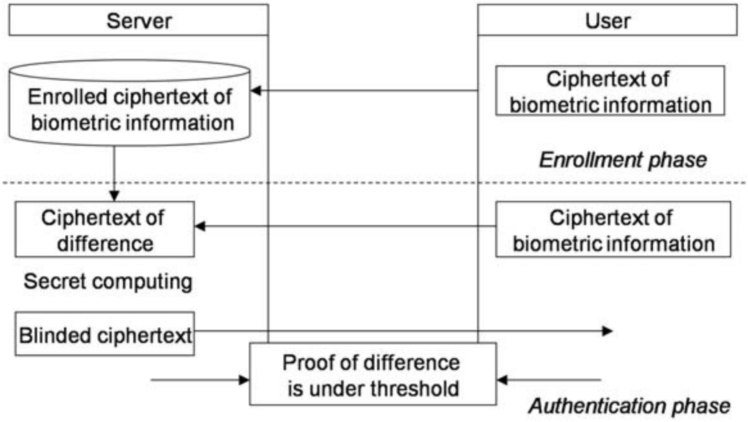

Server
User
Enrolled ciphertext of biometric information
Ciphertext of biometric information
Enrollment phase
Ciphertext of difference
Ciphertext of biometric information
Secret computing
Blinded ciphertext
Proof of difference is under threshold
Authentication phase

Fig. 1. Outline of our proposed method

Enrollment Phase

Step 1-1: Authorized user chooses random integers $k, r \in Z_p$ $(k \neq 0)$.

Step 1-2: The user generates secret key s of ElGamal encryption and corresponding public key $y = g^s$.

Step 1-3: The user puts his/her authentic biometric sample to the biometric reader to obtain biometric information x, and computes the ciphertext $(t_1, t_2) = (g^r, g^{kx} \cdot y^r)$. Note that the user encrypts not the biometric information x but kx. The reason why kx is encrypted will be explained in Sec.5.1.

Step 1-4: The user transmits y and (t_1, t_2) to the server.

Step 1-5: The server stores y and (t_1, t_2)together with the user ID, while the user stores s,y,k in his/her smart card.

Authentication Phase

Step 2-1: A user chooses random integer $r' \in Z_p$.

Step 2-2: The user puts his/her biometric sample to the biometric reader to obtain biometric information x' . Then, the user retrieves s,y,k from his/her smart card, and computes the ciphertext $(t'_1, t'_2) = (g^{r'}, g^{kx'} \cdot y^{r'})$.

Step 2-3: The user transmits (t'_1, t'_2) to the server.

Step 2-4: The server chooses random integers $z, \alpha \in Z_p$ as blind factors. Then the server computes $(w_1, w_2) = (g^z t_1/t'_1, \alpha y^z t_2/t'{}_2)$ and sends back it to the user. Note that (w_1, w_2), the encrypted difference of (t_1, t_2) and (t'_1, t'_2), is concealed from the user by z and R.

Step 2-5: The user decrypts (w_1, w_2) with secret key s to obtain $m = \alpha \cdot g^{k\left(x-x'\right)}$.

Step 2-6: The user chooses a random number $u \in Z_p$ as a blind factor. Then the user computes
$\mathit{\Gamma} = \left\{Hash\left(u \| \, m \cdot g^0\right), Hash\left(u \| \, m \cdot g^{\pm k}\right), \ldots, Hash\left(u \| \, m \cdot g^{\pm k\theta}\right)\right\}$ and transmits $\mathit{\Gamma}$ and u to the server, where '$\|$' denotes concatenation. Here,

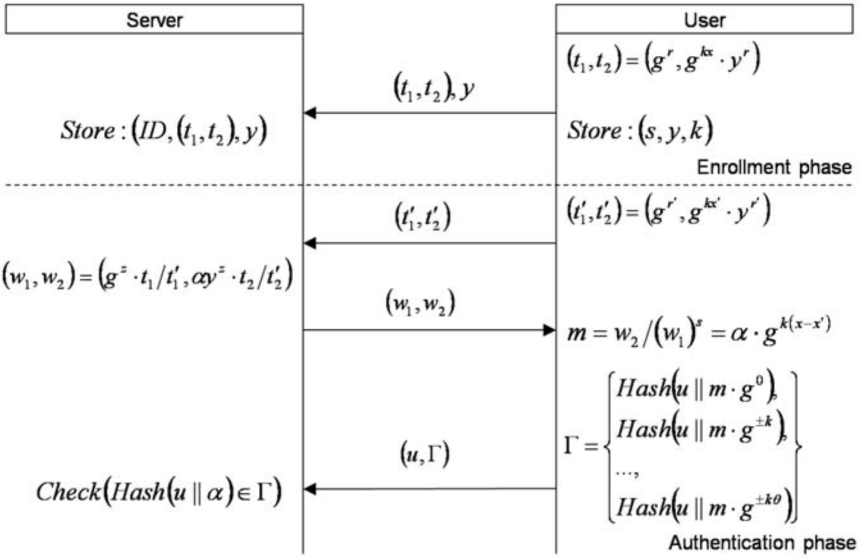

Server
User
$(t_1,t_2)=(g^r,g^{kx}\cdot y^r)$
$(t_1,t_2),y$
$Store:(ID,(t_1,t_2),y)$
$Store:(s,y,k)$
Enrollment phase
(t_1',t_2')
$(t_1',t_2')=(g^{r'},g^{kx'}\cdot y^{r'})$
$(w_1,w_2)=(g^z\cdot t_1/t_1',\alpha y^z\cdot t_2/t_2')$
(w_1,w_2)
$m=w_2/(w_1)^s=\alpha\cdot g^{k(x-x')}$
$\Gamma=\{Hash(u\|m\cdot g^0), Hash(u\|m\cdot g^{\pm k}), \ldots, Hash(u\|m\cdot g^{\pm k\theta})\}$
(u,Γ)
$Check(Hash(u\|\alpha)\in\Gamma)$
Authentication phase

Fig. 2. Proposed protocol

the order of elements of Γ is permuted before transmitting. The reason why the hashing and permutation are needed will be explained in Sec.5.1.

Step 2-7: The server computes $Hash\,(u||\alpha)$. The server authenticates the user if $Hash\,(u||\alpha) \in \Gamma$ is satisfied. If $x - x' \in \Delta$, then there exists $d \in \Delta$ such that $(x - x') + d = 0$. That is why, the server can understand that the presented biometric information x' is sufficiently close to the enrolled biometric information x if there exists $d \in \Delta$ such that $m \cdot g^{kd} = \alpha$.

Fig. 2 illustrates the above protocol.

5 Property of Proposed Protocol

5.1 Remarks

The proposed protocol uses some techniques to prevent attack. In this section, we explain how our techniques work.

Necessity of Hash Function. In Step 2-6 in the authentication phase, every elements of Γ are hashed. If they were not hashed, then the server could abuse the authentication protocol as a decryption machine in the following way.

Assume that the server has a ciphertext (w_1, w_2) and wants to know the plaintext. Then the protocol would be the followings: The server sends (w_1, w_2) back to the user in Step 2-4. The user decrypts (w_1, w_2) to obtain plaintext m in Step 2-5, and generate $u \in Z_p$ to compute
$\Gamma = \left\{ \left(u \| \, m \cdot g^0 \right), \left(u \| \, m \cdot g^{\pm k} \right), \ldots, \left(u \| \, m \cdot g^{\pm k\theta} \right) \right\}$ in Step 2-6.

In this case, all the values $\{\ldots, m \cdot g^{-2k}, m \cdot g^{-k}, m \cdot g^{k}, m \cdot g^{2k}, \ldots\}$ are disclosed to the server since Γ and u are transmitted to the server in Step 2-6. Here, the server knows a "knowledge about Γ"that one of the elements of Γ is $m(= m \cdot g^0)$ and Γ forms $\left\{..., u\| \, m \cdot g^{-2k}, \, u\| \, m \cdot g^{-k}, \, u\| \, m \cdot g^{k}, \, u\| \, m \cdot g^{2k}, ...\right\}$. Such knowledge helps the server to decide which one in Γ is m, even if the order of elements is permuted.

When the value of each element in Γ is not disclosed, the server can't deduce m even if the server knows the "knowledge about Γ."Thus, this attack can be prevented by hashing each element in Γ, in this case, server no longer treat the user as decryption machine.

Necessity of permutation of elements of *Γ*. In Step 2-6 in the authentication phase, the elements of Γ are permuted and then transmitted to the server. If elements of Γ were not permuted, then the server could derive the difference between the enrolled biometric information and the presented biometric information $x - x'$ in the following way.

Assume that the lth element of Γ is equal to $Hash\,(u||\alpha)$. This means that $Hash(\,u\| \, \alpha \cdot g^{k(x-x')} \cdot g^{kl}) = Hash(\,u\| \, \alpha)$, i.e., $x - x' + l = 0$. Therefore the server can derive $x - x'$.

This is caused by the fact that the server can deduce the preimage of the hashed value from the order of elements of Γ. Therefore, this attack can be prevented by the permutation of elements of Γ in every authentication.

Necessity of *k*. In Step 1-3 in the enrollment phase and Step 2-2 in the authentication phase, the user encrypts not the biometric information x or x', but kx or kx'. If ciphertext of x or x' were used, then the server could derive the difference between the enrolled biometric information and the presented biometric information $x - x'$ in the following way.

Assume that we do not use k, that is, $k = 1$ in our protocol. In this case, the elements in Γ transmitted from the user to the server in Step 2-6 are $\left\{Hash\left(u||\alpha \cdot g^{\left(x-x'\right)\pm 0}\right), Hash\left(u||\alpha \cdot g^{\left(x-x'\right)\pm 1}\right), ..., Hash\left(u||\alpha \cdot g^{\left(x-x'\right)\pm \theta}\right)\right\}$. Here, the server who knows g, α and u could try to compute

$$\left\{Hash\left(u||\alpha \cdot g^{\left(x-x'\right)\pm 0}\right), Hash\left(u||\alpha \cdot g^{\left(x-x'\right)\pm 1}\right), ..., Hash\left(u||\alpha \cdot g^{\left(x-x'\right)\pm \theta}\right)\right\}$$

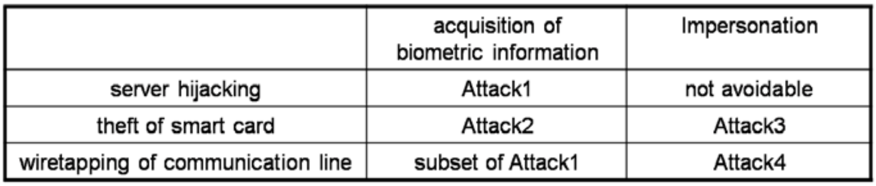
by guessing $x - x'$. This means that the server can know the guess is correct when the server's calculation is identical to Γ transmitted from the user.

This attack is caused by the fact that the server also can compute the elements of Γ. Therefore, this attack can be prevented by introducing a random number k which is secret from the server.

5.2 Security

In this section, we evaluate the security of our proposed protocol. Here we first define the attack model to derive security requirements for biometric authentication system, and then confirm that proposed protocol satisfies these requirements.

Table 1. List of attacks



	acquisition of biometric information	Impersonation
server hijacking	Attack1	not avoidable
theft of smart card	Attack2	Attack3
wiretapping of communication line	subset of Attack1	Attack4

Attack model and requirements for biometric authentication system Attack model can be divided into attacker's objectives and means of attack.

Attacker's objectives : One of the biggest attacker's objectives is “impersonatio”of a certain system itself. Also, attackers try “acquisition of biometric information”of authorized users in a certain system to use them for further frauds such as impersonation of the other system or trace of the users.

Means of attack : We consider that attackers can attack the server, the authorized user, or the communication line. The first type of attack is a kind of “server hijacking.”If an attacker succeeds in hijacking a server, the attacker can access all information in the server. Note that hijack means impersonation of the server administrator (or, maybe cheating by the server administrator). So, impersonation has no meaning for hijackers. The second type of attack is done by “theft of a smart card.”We assume that the attacker can extract the information stored in the smart card that he/she steals. The third type of attack is “wiretapping of communication line.”In remote authentication protocols, every information transmitted in the communication line are received or generated by the server. Therefore, if an authentication system can protect biometric information from the server hijackers, the system is robust also against the acquisition of biometric information by the wiretapping of communication line.
The combination of attacker's objectives and means of attack indicates us that we have to consider security against 4 types of attacks showed in Table1. Here, we assume that more than one of the above attacks are not conducted by attackers at the same time. Note that the security analysis described here is a kind of informal analysis. For instance, if the server is malicious, the attacker (server) who get the information stored in the smart card will be able to impersonate. Therefore, to be precise, the formal security analysis should be conducted.

From Table1, biometric authentication system has to satisfy following requirements.

Requirement 1 (against attack 1) : The server can not obtain any information about biometric information of authorized users from the enrolled data and/or through authentication protocol.

Requirement 2 (against attack 2) : Anyone can not derive any information about biometric information from data stored in a smart card.

Requirement 3 (against attack 3) : Even if attacker gets a smart card of an authorized user, it is impossible to impersonate the user without knowing biometric information sufficiently close to the enrolled biometric information.

Requirement 4 (against attack 4) : Even if attacker uses information obtained by wiretapping the communication line, it is impossible to impersonate anyone.

Security Evaluation. We show the proposed protocol in Section 4 satisfies the above requirements.

Requirement 1. Information obtained by the server through protocol are only ciphertexts of biometric information and hash values of difference between biometric information concatenated with a random number.

The server does not have a secret key. Also, the server can not abuse the authentication protocol as a decryption machine, as described in Sec.5.1. Therefore the server can not decrypt any ciphertext. In addition, the server can not estimate the preimage of hash values because of onewayness of the hash function.

Therefore, even if an attacker can hijack the server, the attacker can not obtain information about biometric information.

Requirement 2. Information stored in a smart card is only secret key s and random number k. Therefore it is impossible for an attacker to obtain information about biometric information by theft of a smart card.

Requirement 3. An attacker with a user's smart card can retrieve the user's secret key s and random number k. However, we can show that even if an attacker can obtain a smart card, the attacker can not succeed impersonation without knowing the user's biometric information:

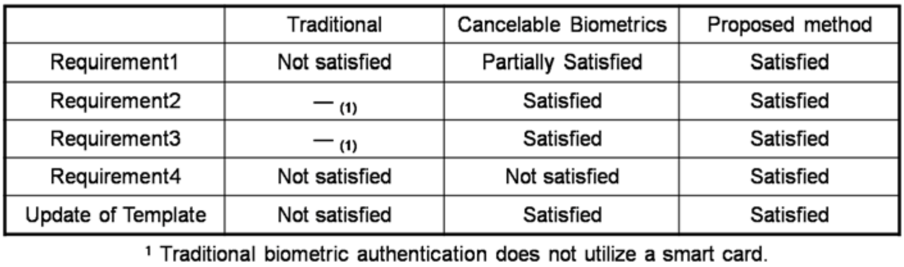
To succeed in impersonating, the attacker has to transmit to the server a set of hash values Γ which contains $Hash\,(u||\alpha)$ in Step 2-6. This means that the attacker is required to guess α with high probability. This is, however, proved to be impossible, as explained as explained below.

The attacker can present an arbitrary data $\hat{x}$ to the server, instead of the attacker's biometric information, since the attacker knows the secret key. That is, the attacker encrypts $\hat{x}$ to obtain the ciphertext (t'_1, t'_2), and transmits it to the server in Step 2-3. In this case, (w_1, w_2) calculated by the server using (t_1, t_2) and (t'_1, t'_2) in Step 2-4 is a ciphertext of $\alpha \cdot g^{k(x-\hat{x})}$. Therefore, after receiving (w_1, w_2), all information the attacker knows is $\left(k, \hat{x}, \alpha \cdot g^{k(x-\hat{x})}\right)$.

To guess the value R from $\left(k, \hat{x}, \alpha \cdot g^{k(x-\hat{x})}\right)$, it is necessary for the attacker to know x or $x - \hat{x}$ with high probability. This means that the attacker who succeeds impersonation can estimate the user's biometric information x with high probability before the start of the protocol. - Q.E.D.-

More preciously, if the attacker could only get the amount of the difference $x - \hat{x}$, the attacker can calculate R without knowing the user's biometric information x itself. In practical sense, however, we can understand that this is not a critical issue, since the attacker who knows $x - \hat{x}$ is almost equivalent to the attacker who knows x.

Table 2. Security comparison



	Traditional	Cancelable Biometrics	Proposed method
Requirement1	Not satisfied	Partially Satisfied	Satisfied
Requirement2	— [1]	Satisfied	Satisfied
Requirement3	— [1]	Satisfied	Satisfied
Requirement4	Not satisfied	Not satisfied	Satisfied
Update of Template	Not satisfied	Satisfied	Satisfied

[1] Traditional biometric authentication does not utilize a smart card.

Requirement 4. An attacker can retransmit (t'_1, t'_2) that an authorized user transmitted in Step 2-3. However, the server generates different random numbers z, α each time to compute (w_1, w_2) in Step 2-4, therefore the attacker without knowledge of the secret key s can not decrypt (w_1, w_2), and thus impersonation will fail.

Finally we compare security issues of our protocol with the traditional biometric authentication protocol and the cancelable biometric authentication protocol. Table 2 shows the result of comparison.

Cancelable biometrics do not fully satisfy Requirement 1, because difference between the enrolled biometric information and the presented biometric information is leaked out to the server hijacker. Also, Requirement 4 is not satisfied, because it is possible to succeed replay attack by retransmitting the information derived from wiretapping of the communication line.

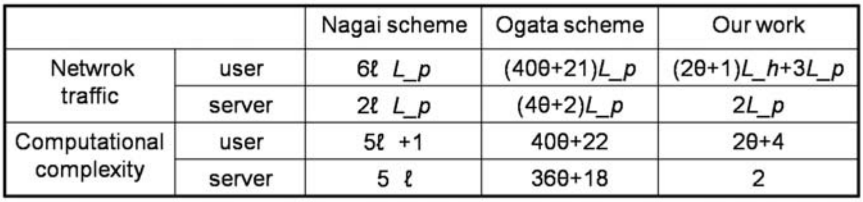
On the other hand, our protocol as well as Nagai scheme and Ogata scheme satisfies all requirements. However, as described in Requirement 3, our protocol will allow an attacker who knows the amount of the difference $x - \hat{x}$ to impersonate an authorized user without knowing the user's biometric information x. In practical sense, the attacker who knows $x - \hat{x}$ is almost equivalent to the attacker who knows x. So, we can understand that this is not a critical problem. But, it is small decline of security level compared to Nagai scheme and Ogata scheme. We will give an improvement of our protocol against this issue in Sec.6.

5.3 Comparison of Efficiency

Here, we compare the efficiency of our protocol with other ZeroBio protocols such as Nagai scheme and Ogata scheme presented in Section 3.

We compare computational complexity by the number of exponentiation operation needed for one authentication phase, and network traffic by the number of data transmitted during one authentication phase. Ogata et al. improved their result in [9] by storing additional information in a smart card to reduce both computational complexity and network traffic without declining any security. However, as the same improvement as [9] can be applied also to our protocol, we compare here Ogata scheme and our scheme without utilizing the improvement proposed in [9].

Table 3. Comparison of proposed protocol and other ZeroBio protocol



		Nagai scheme	Ogata scheme	Our work
Netwrok traffic	user	6ℓ *L_p*	(40θ+21)*L_p*	(2θ+1)*L_h*+3*L_p*
	server	2ℓ *L_p*	(4θ+2)*L_p*	2*L_p*
Computational complexity	user	5ℓ +1	40θ+22	2θ+4
	server	5 ℓ	36θ+18	2

Let L_p be the size of the transmitted data packets (typically, $L\_p = 1024$ bits) and L_h be the length of a hash value (typically, $L\_h = 160$ bits). Let ℓ be the number of hidden layer unit in neural network. And θ be a security parameter used in Ogata scheme and our scheme to define the authentic interval $[-\theta, \theta$ $]$ for the difference between biometric information. Then, we can summarize the estimates of computational complexity and network traffic needed for each protocol in Table 3.

We first compare our protocol with Nagai scheme. Although the biometric information fed to neural network is n-dimensional vector in Nagai scheme, we assume $n = 1$ here for simplicity of estimation. We can see from Table 3 that computational complexity and network traffic are proportional to the number of hidden layer unit ℓ. As ℓ and θ are different parameter, we can not directly compare with Nagai scheme. However Nagai scheme at least needs additional computational cost to train the weight of the connection in the neural network.

Next, we compare our protocol with Ogata scheme. Our protocol achieves improvement in both computational complexity and network traffic needed for the server and the user. For the user, our protocol needs only 1/20 of computational complexity and network traffic needed for Ogata scheme. For the server, our protocol needs $1/(18\theta + 9)$ of computational complexity, and $1/(2\theta + 1)$ of network traffic needed for Ogata scheme. Therefore, we can confirm that our protocol achieves the performance improvement compared to Nagai scheme and Ogata scheme.

6 Improvement of Our Protocol

In the security evaluation with respect to Requirement 3 in Section 5.2, we described that if an attacker knows $x - \hat{x}$, the attacker can impersonate without knowing the enrolled biometric information x. This means that even if $\hat{x}$ is not close to x, an attacker who presents an arbitrary data $\hat{x}$ will be authenticated in the case that the attacker knows the difference $x - \hat{x}$. In practical sense, we can understand that this is not a critical issue, since the attacker who knows $x - \hat{x}$ is almost equivalent to the attacker who knows x. However, it is more preferable if our protocol can prove that the user indeed possesses x' such that sufficiently close to the enrolled biometric information x. Therefore, in this section, we try to improve our protocol.

More concretely, the user transmits every ciphertext of integers $d \in \mathit{\Delta}$ at the enrolled phase. The server generates random bit b at the authentication phase. If $b = 0$, then the server transmits the ciphertext of αd to the user and checks the reply from the user satisfies $Hash\,(u||\alpha) \in \mathit{\Gamma}$ to confirm that the user compute with the proper d. If $b = 1$, the regular authentication phase (namely, the Authentication Phase described in Section 4.3) is conducted. Note that the procedures for the user are the same regardless of whether b is 1 or 0.

Detailed explanation of our improved protocol is described as follow.

6.1 Authentication Method

Enrollment phase

Step 3-1: Authorized user chooses random integers $k, r \in Z_p$ $(k \neq 0)$. In addition, the user chooses random integers $r[d] \in Z_p$ for each $d \in \mathit{\Delta}$, where $r[d]$ is used for encrypting each d in Step 3-4.

Step 3-2: The user generates secret key s of ElGamal encryption and corresponding public key $y = g^s$.

Step 3-3: The user puts his/her authentic biometric sample to the biometric reader to obtain biometric information x, and computes the ciphertext $(t_1, t_2) = (g^r, g^{kx} \cdot y^r)$.

Step 3-4: The user computes $E(\mathit{\Delta}) = \{(g^{r[d]}, g^{kd} \cdot y^{r[d]})|d \in \mathit{\Delta}\}$, the set of ciphertexts of kd for all $d \in \mathit{\Delta}$.

Step 3-5: The user transmits y, (t_1, t_2) and $E(\mathit{\Delta})$. Note that the elements of $E(\mathit{\Delta})$ should be permuted before sending to server. Otherwise, attackers may guess the relationship between ciphertexts and plaintexts from the order of elements of $E(\mathit{\Delta})$.

Step 3-6: The server stores y, (t_1, t_2) and $E(\mathit{\Delta})$ together with the user ID, while the user stores s,y,k in his/her smart card.

Authentication phase

Step 4-1: A user chooses random integer $r' \in Z_p$.

Step 4-2: The user puts his/her biometric sample to the biometric reader to obtain biometric information x'. Then, the user retrieves s,y,k from his/her smart card, and computes the ciphertext $(t'_1, t'_2) = (g^{r'}, g^{kx'} \cdot y^{r'})$.

Step 4-3: The user transmits (t'_1, t'_2) to the server. Step 4-4 to Step 4-8 are independently conducted L times at the same time.

Step 4-4: The server generates random bit b. If $b = 0$, then, the server randomly chooses $z, \alpha \in Z_p$ and $(e_1, e_2) \in E\,(\mathit{\Delta})$ as blind factors, and transmits $(w_1, w_2) = (e_1 g^z, e_2 \alpha y^z)$ to the user. If $b = 1$, then, the server randomly chooses $z, \alpha \in Z_p$ as blind factors, and transmits (w_1, w_2)=$(g^z t_1/t'_1, \alpha y^z t_2/t'_2)$ to the user.

Step 4-5: The user decrypts (w_1, w_2) with secret key s to obtain $m = \alpha \cdot g^{k\left(x-x'\right)}$.

Step 4-7: The user chooses a random number $u \in Z_p$ as a blind factor. Then the user computes $\mathit{\Gamma} = \left\{Hash\left(u||m \cdot g^{kd}\right)|d \in \mathit{\Delta}\right\}$, and transmits $\mathit{\Gamma}$ and u to the server. Here, the order of elements of $\mathit{\Gamma}$ is permuted before transmitting.

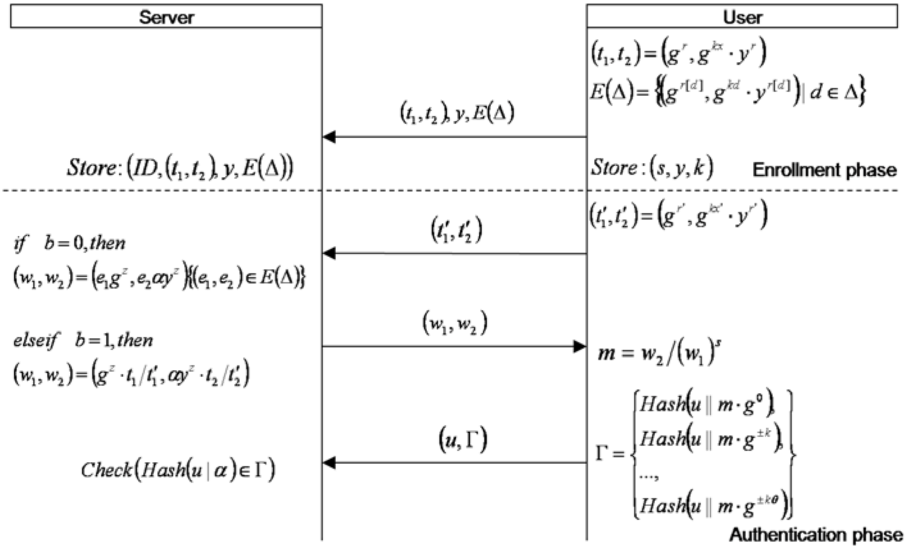

Server
User
$(t_1,t_2)=(g^r, g^{kx}\cdot y^r)$
$E(\Delta)=\{(g^{r[d]}, g^{kd}\cdot y^{r[d]}) \mid d\in\Delta\}$
$(t_1,t_2), y, E(\Delta)$
$Store:(ID,(t_1,t_2), y, E(\Delta))$
$Store:(s,y,k)$
Enrollment phase
$(t'_1,t'_2)=(g^{r'}, g^{kx'}\cdot y^{r'})$
(t'_1,t'_2)
$if\ \ b=0, then$
$(w_1,w_2)=(e_1g^z, e_2\alpha y^z)\}(e_1,e_2)\in E(\Delta)\}$
(w_1,w_2)
$elseif\ \ b=1, then$
$(w_1,w_2)=(g^z\cdot t_1/t'_1, \alpha y^z\cdot t_2/t'_2)$
$m=w_2/(w_1)^s$
$\Gamma=\{Hash(u\,\|\,m\cdot g^0), Hash(u\,\|\,m\cdot g^{\pm k}), \ldots, Hash(u\,\|\,m\cdot g^{\pm k\theta})\}$
(u,Γ)
$Check(Hash(u\,|\,\alpha)\in\Gamma)$
Authentication phase

Fig. 3. Protocol with higher security

Step 4-8: The server computes $Hash\,(u||\alpha)$, then checks whether it satisfies $Hash\,(u||\alpha) \in \Gamma$.

Step 4-9: If Step 4-8 is always true, then the user is authenticated.

We illustrate the above protocol in Figure 3.

6.2 Discussion

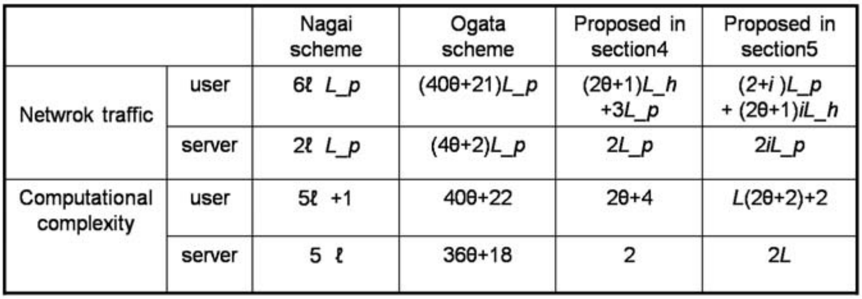
In our regular protocol shown in Sec.4, (t'_1, t'_2) does not have to be a ciphertext of kx' where x' is sufficiently close to x. We first show that in our improved protocol, the user (or attacker) is not authenticated with high probability if x' is not close to x.

Assume that (t'_1, t'_2) is a ciphertext of $k\hat{x}$ generated by an attacker in Step 4-2, where $x - \hat{x}$ is not small enough and the attacker knows the amount of $x - \hat{x}$. In the case of $b = 1$, Step 4-4 to Step 4-8 are the same as the regular authentication phase. Therefore, the attacker can impersonate using the knowledge of the difference $x-\hat{x}$. More concretely, the attacker generates $Hash\,(u||\alpha)$ by calculating $Hash\,(u||m \cdot g^{kd})$ with $d = -(x - \hat{x})$, mixes it into Γ, and transmits Γ to the server in Step 4-7.

On the other hand, if $b = 0$, the attacker has to calculating $Hash\,(u||m \cdot g^{kd})$ with every $d \in \Delta$ to compute Γ in Step 4-7 so that the attacker can obtain Γ which includes $Hash\,(u||\alpha)$ in it.

That is, to succeed impersonation, the attacker needs to use $d = -(x - \hat{x})$ for $b = 1$ and $d \in \Delta$ for $b = 0$ when calculating $Hash\,(u||m \cdot g^{kd})$, However (w_1, w_2) transmitted from the server in Step 4-4 is concealed by the random number R, and the attacker has no way to know the value b. This means that

Table 4. Computational complexity and network traffic



		Nagai scheme	Ogata scheme	Proposed in section4	Proposed in section5
Netwrok traffic	user	$6\ell\ L\_p$	$(40\theta+21)L\_p$	$(2\theta+1)L\_h$ $+3L\_p$	$(2+i\ )L\_p$ $+ (2\theta+1)iL\_h$
	server	$2\ell\ L\_p$	$(4\theta+2)L\_p$	$2L\_p$	$2iL\_p$
Computational complexity	user	$5\ell\ +1$	$40\theta+22$	$2\theta+4$	$L(2\theta+2)+2$
	server	$5\ \ell$	$36\theta+18$	2	$2L$

the probability that attacker passes the test of Step 2-8 is 1/2. Therefore, the probability that attacker succeeds impersonation is $(1/2)^i$, which is negligible for sufficiently large i.

Next, we discuss computational complexity and network traffic. In the improved protocol, the user conducts Step 4-4 to Step 4-8 i times. This means that computational complexity and network traffic depend not only on θ but also on security parameter i. There is the tradeoff between security and computational complexity and network traffic. Therefore it is important to set proper i which satisfies the required security level.

Table 4 shows the estimate of network traffic and computational complexity needed for our protocol, where L_p, L_h, ℓ and are the same definition as used in Table 3. We can find that the improvement in the security level of our protocol is accompanied by an increase in its computational complexity and network traffic.

7 Conclusion

In this paper, we proposed a secure remote biometric authentication system which has a certain level of resistance against impersonation and biometric information disclosure. We also compared our method with other asymmetric biometric authentication, and found that our method achieves the asymmetric biometric authentication with comparatively smaller computational complexity and network traffic.

References

1. Ratha, N.K., Connell, J.H., Bolle, R.M.: Enhancing Security and Privacy in Biometrics-based Authentication Systems. IBM Systems Journal 40(3) (2001)
2. Cambier, J.L., Cahn von Seelen, U., Glass, R., Moore, R., Scott, I., Braithwaite, M., Daugman, J.: Application-Specific Biometric Templates. In: IEEE Workshop on Automatic Identification Advanced Technologies, Tarrytown, NY, March 14-15, pp. 167–171 (2002)
3. Hirata, S., Takahashi, K.: Cancelable Biometrics with Perfect Secrecy for Correlation-based Matching. In: Tistarelli, M., Nixon, M.S. (eds.) ICB 2009. LNCS, vol. 5558, pp. 875–885. Springer, Heidelberg (2009)

4. Hill, C.J.: Risk of masquerade arising from the storage of biometrics, Bachelor thesis, Dept. of CS, Australian National University (2002)
5. Nagai, K., Kikuchi, H., Ogata, W., Nishigaki, M.: ZeroBio - Evaluation and Development of Asymmetric Fingerprint Authentication System Using Oblivious Neural Network Evaluation Protocol. In: Proceedings of 2007 International Conference on Availability, Reliability and Security, pp. 1155–1159 (2007)
6. Ogata, W., Kikuchi, H., Nishigaki, M.: Zero-knowledge interactive proofs for proving nearness of biometrics and its application. In: Symposium on Information Theory and its Applications, SITA2006, pp. 319–322 (2006)(in Japanese)
7. Paul, K., Joshua, J., Benjamin, J.: Differential Power Analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
8. Fujisaki, E., Okamoto, T.: Statistical Zero-Knowledge Protocols to Prove Modular Polynomial Relations. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 413–430. Springer, Heidelberg (1997)
9. Ogata, W., Kikuchi, H., Nishigaki, M.: Improvement of the biometric authentication system using ZKIP. In: Symposium on Information Theory and its Applications, SITA2007, pp. 689–693 (2007)(in Japanese)