リプレイ攻撃や不正なサーバによる攻撃に耐性のある 秘匿生体認証方式

Sano, Ryosuke / 佐野, 僚哉

(出版者 / Publisher)
法政大学大学院情報科学研究科
(雑誌名 / Journal or Publication Title)
法政大学大学院紀要. 情報科学研究科編 / 法政大学大学院紀要. 情報科学研究科編
(巻 / Volume)
12
(開始ページ / Start Page)
1
(終了ページ / End Page)
6
(発行年 / Year)
2017-03-31
(URL)
https://doi.org/10.15002/00014407

リプレイ攻撃や不正なサーバによる攻撃に耐性のある秘匿生体認証方式
Biometrics Authentication with Template Protection Secure against Replay Attack and Attacks form Server

佐野　僚哉
Ryosuke Sano
法政大学大学院情報科学研究科情報科学専攻
E-mail: ryosuke.sano.7x@stu.hosei.ac.jp

Abstract

Biometrics authentication is attracting rising attention. Because biological information used in authentication contains a lot of information, it is more difficult to mount impersonation attack than ID/Password scheme. Since biological information contains more critical information, it is necessary to manage biological information securely. To resolve this issue, template protection schemes were proposed, where template protection schemes make it possible to authenticate users without revealing biological information of template data. Bringer et al. proposed a biometrics authentication scheme with template protection using error correcting code and homomorphic encryption. The scheme considers the difference between two biometric features as an error, where they are the biometrics information in enrollment and authentication process. Furthermore, the scheme uses a XOR homomorphic encryption to encrypt template data and authenticates users without decrypting their biological information. However, the scheme has two problems. One is nothing that is countermeasure against replay attack, and the other is that selecting parameters is restricted because the scheme uses an error correcting code. In this paper, we propose two schemes that have countermeasures against these problem. One scheme prevents replay attack by adding different values for each session to a query, using Diffie-Hellman key exchange. The other scheme can set parameters more flexibility, by using an additive homomorphic encryption without an error correcting code.

1 はじめに

今日，インターネットの普及に伴い，インターネットを利用したサービスが急増している．それらのサービスの中には, 課金を伴うサービスなど，厳重な個人認証が必要なサービスも少なくない．現在最も利用されている認証方式として，ID/パスワードによる認証がある．ID/パスワード認証は，特別な機器を必要としないという利便性から多くのサービスで採用されている．しかし，多くのユーザは，パスワードの内容を忘れないために，誕生日など安易なパスワードを用いがちである．そのため，攻撃者はパスワードを予想しやすく，なりすましされる危険性が高いのが現状である．

その対策として，生体認証が注目されている．生体認証は，パスワードや物を用いないため，忘却や紛失等によって認証行なえなくなるリスクを軽減できる．また，生体情報は情報量が多いため，ID/パスワード方式に比べ，容易になりすましができない．しかし，生体情報は個人を特定できる情報であり，生涯不変の情報を利用しているので，漏洩した場合に無効化することが困難であるため，生体認証を行う際には生体情報が漏えいしないように厳重な注意を払わなければならない．

上記の課題を解決する方法として，生体情報を秘匿したまま認証処理を行う秘匿生体認証方式が提案された．秘匿生体認証方式とは，オリジナルの生体情報を不可逆な関数や乱数などにより保護し，元の生体情報を秘匿したまま認証を行う技術である．現在，秘匿生体認証方式は多数の方式が提案されており，その中に準同型暗号を用いた方式 [1, 4] が存在する．準同型暗号とは，暗号化したまま演算が行える暗号方式であり，準同型暗号を用いることにより，オリジナルの生体情報を秘匿したまま登録・認証時の生体情報を照合を行うことができる．照合過程で生体情報を復号しないため，認証サーバに登録されている情報が漏洩しても，オリジナルの生体情報を復元することは困難である．

準同型暗号を用いた秘匿生体認証方式の一つとして，準同型暗号と誤り訂正符号を組み合わせた方式が Bringer らによって提案されている [1]．この方式は，Goldwasser-Micali 暗号 (以下，GM 暗号) の XOR 準同型性を利用し，登録・認証時の生体情報のハミング距離を暗号化したままで計算することで認証を行う．2 つの生体情報間の差分を誤りと見立て，誤り訂正符号で訂正可能か否かを基準とした閾値判定を行う．しかし，この方式にはいくつかのつの問題点が存在する．第一の問題は，正規のユーザが認証処理で用いた認証情報を利用することで，第三者が認証されてしまうリプレイ攻撃への耐性がないことである．第二の問題は，誤り訂正符号を利用しているため，認証に用いる閾値やセキュリティパラメータなどの数値が制限されてしまうことである．

本研究では，これらの課題の解決に向け，二つの方式を提案する．第一の方式は，Bringer らの方式にリプレイ攻撃の耐性を持たせた方式である．Bringer らの方式をベースとして，認証時に Diffie-Hellman 鍵共有法を利用してチャレンジ・レスポンスを行う．第二の方式は，誤り訂正符号を用いずに加法準同型暗号を用いてパラメータの自由度を高めた方式である．誤り訂正符号を用

Supervisor: Prof. Satoshi Obana

いると，生体情報長や閾値（訂正可能数），セキュリティパラメータを任意に決めることができず，各パラメータが制限されてしまう．しかし，本方式では生体情報の要素ごとに加法準同型暗号を用いて生体情報をマスクして認証を行っているため，各パラメータを自由度を高めることができる．

2 準備

2.1 誤り訂正符号を用いた生体認証方式

Tuyls らによって誤り訂正符号を用いた秘匿生体認証方式 [10] が提案されている．生体認証では，登録時と認証時の生体情報の近似度に基づき本人か否かの認証が行われる．Tuyls らの方式では，生体情報間の差分を誤りと見立て認証を行う．具体的には，誤り訂正符号で符号化した値と 2 つの生体情報の排他的論理和をとり，誤り訂正符号で訂正可能か否か，つまり登録時と認証時の生体情報のハミング距離が閾値以下であるか否かで本人か否か判定を行っている．生体情報を二値 $(0,1)$ のベクトルとし，登録時に生体情報 x を読み取り，乱数 R を生成する．乱数 R を誤り訂正符号で符号化した値と生体情報の排他的論理和 $\mathsf{Encode}_{ECC}(R) \oplus x$ とハッシュ値 $H(R)$ を登録情報とする．認証時は，認証時の生体情報 y を用いて $\mathsf{Encode}_{ECC}(R) \oplus x \oplus y$ を求める．検証者は，$\mathsf{Encode}_{ECC}(R) \oplus x \oplus y$ を復元して R' を求める．2 つの生体情報 x, y が同一人物から生成されている場合は，それぞれの生体情報の差によって生じた誤りを訂正可能なため $H(R) = H(R')$ となり，本人か否かの認証が行える．

2.2 Goldwasser-Micali 暗号

Goldwasser-Micali 暗号は，Goldwasser らによって提案された XOR 準同型性を有した準同型暗号である [3]．この暗号は，平方剰余，平方非剰余を利用した暗号方式であり，平文 {0,1} を平方剰余，平方非剰余で表している．暗号文が平方剰余（もしくは，平方非剰余）の時，平文は 0（もしくは，1）である．

鍵生成アルゴリズム Gen_{GM}

1. 素数 p, q を選び，$N = pq$ とする
2. 法 N において平方非剰余，かつ Jacobi 記号の値が $(\frac{a}{N}) = 1$ である値 $a \in \mathbb{Z}_N^*$ を選ぶ
3. 公開鍵を (N, a)，秘密鍵を (p, q) とする

暗号化アルゴリズム Enc_{GM}

平文をバイナリ列 $b = (b_1, b_2, ..., b_t)$ とする時，

1. 乱数 $r \xleftarrow{u} \mathbb{Z}_N^*$ を決める
2. $b_i (i = 1, ..., t)$ に対し，暗号文 $e = (e_1, ..., e_t)$ を次式により計算する

$$b_i = 0 \text{ の時}: e_i = r^2 \bmod N$$
$$b_i = 1 \text{ の時}: e_i = ar^2 \bmod N$$

復号アルゴリズム Dec_{GM}

暗号文 $e = (e_1, ..., e_t)$ に対して，

1. 秘密鍵を用いて $e_i \ (i = 1, ..., t)$ が平方剰余か平方非剰余か確認する
2. e_i が平方剰余の時，$b_i = 0$ とし，平方非剰余の時，$b_i = 1$ とする

平方剰余同士の積は平方剰余となり，公開鍵 a の 2 乗（平文 1 の暗号文同士の積）も平方剰余である．また，平方剰余と平方非剰余の積は平方非剰余となるため，平文 m, m' の暗号文は $\mathsf{Enc}(m, pk) \cdot \mathsf{Enc}(m', pk) = \mathsf{Enc}(m \oplus m', pk)$ という XOR 準同型性を有している．

2.3 Diffie-Hellman 鍵共有法（DH 鍵共有法）

Diffie-Hellman 鍵共有法は，離散対数を用いて秘密（共通鍵）情報を共有する方法である [2]．十分大きな素数 q に対し，位数 q の巡回群 $\mathbb{G}_q$ の生成元 g を選び，$(\mathbb{G}_q, g, q)$ を公開鍵とする．この時，Alice と Bob の鍵共有は次のように行われる．Alice と Bob はそれぞれ乱数 $a, b \in \{1, ..., q-1\}$ を一様に選ぶ．Alice から Bob へ $A = g^a$ を，Bob から Alice へ $B = g^b$ を送る．Alice は B から $B^a = g^{ab}$ を，Bob は A から $A^b = g^{ab}$ を計算する．上記の処理により，Alice と Bob は g^{ab} を共有することができる．

乱数 a, b を知らない第三者は，通信された情報 A, B から g^{ab} を求めることが困難である．この仮定を CDH 仮定という．

2.4 Paillier 暗号

Paillier 暗号は，Paillier により提案された加法準同型性を有した公開鍵暗号である [9].

鍵生成アルゴリズム Gen_P

1. 素数 p, q を選び，$N = pq$, $g = 1 + N$ とする
2. 公開鍵を (N, g)，秘密鍵を (p, q) とする

暗号化アルゴリズム Enc_P

平文 $m \in \mathbb{Z}_N$ とし，

1. 乱数 $r \xleftarrow{u} \mathbb{Z}_N$ を選ぶ
2. 暗号文 $c = g^m r^N \bmod N^2$ を計算する

復号アルゴリズム Dec_P

1. $\lambda = lcm(p-1, q-1)$ を求め，関数 L を $L(u) = \frac{u-1}{N}$ と定義する
2. 平文 $m = \frac{L(c^\lambda \bmod N^2)}{L(g^\lambda \bmod N^2)}$ を求める

平文 m_1, m_2 のそれぞれの暗号文は，$\mathsf{Enc}(m_1, pk) \cdot \mathsf{Enc}(m_2, pk) = g^{m_1+m_2}(r_1 r_2)^N \bmod N^2$ という関係があるため，加法準同型性を有している．

本稿では，加法準同型暗号として Paiilier 暗号を用いているが，方式 2 において他の加法準同型暗号を用いても機能する．

3 認証方式のモデル

本稿では，3 者モデルの認証処理を仮定する．伊豆ら [8] はモデルを認証機能を利用する「クライアント」，登録時と認証時の生体情報の暗号文のマッチング処理を行う「認証サーバ」，秘密鍵を用いてマッチング結果を復号して本人か否かの判定を行う「復号センタ」により構成しており，本稿でも文献 [8] に従うこととする．そのため，Bringer らの方式における「ユーザ」を「クライアント」，「データベース」を「認証サーバ」，「サービスプロバイダ」を「復号センタ」と記述する．

また，攻撃者のモデルとして，認証サーバと復号センタはそれぞれ malicious と仮定しているが，それぞれは

結託しないとする．また，第三者の攻撃者と悪意のある認証サーバは，通信路を盗聴できると仮定する．

4 Bringer らの方式

Bringer らは，二値ベクトルで表される生体情報において生体情報を秘匿したまま認証を行う方式を提案している [1]．この方式では，誤り訂正符号で符号化された乱数と登録時の生体方法の XOR と，認証時の生体情報のそれぞれの各ビットを GM 暗号を用いて暗号化する．GM 暗号の XOR 準同型性を利用することにより，Tuyls らの誤り訂正符号を用いた認証を暗号化したままで行うことができる．Tuyls らの方式では，認証処理が Accept された時の処理で求められた乱数を用いることで登録時の生体情報を算出することができた．しかし，Bringer らの方式では GM 暗号を用いているため秘密鍵を知っているエンティティしか Accept 時の乱数を知ることができない．また，秘密鍵を知っている復号センタは登録情報を知らないため，登録時の生体情報を知ることはできない．

Bringer らの方式では，上記の認証方式と PIR（Private Information Retrieval protocol ：プライバシー情報検索プロトコル）を組み合わせ，認証サーバにどの情報にアクセスしているかを秘匿する処理を行っている．ただし，PIR を用いたプライバシー保護の処理は，認証方式に依存しない汎用的な処理であるため，本稿では検討の範囲外とする．

本方式は，セットアップ・登録処理・認証処理の 3 つのプロトコルから構成されている．

■セットアップ

復号センタ：

各パラメータを生成する：セキュリティパラメータ k に対して $(N, a, p, q) \leftarrow \mathsf{Gen}_{GM}(1^k)$，生体情報長 D, 認証閾値 θ に対して符号後の長さを D, 訂正可能数を θ とする誤り訂正符号 ECC を生成する．

また，公開鍵を $pk_{GM} = (N, a)$，秘密鍵を $pk_{GM} = (p, q)$ と設定する．

■登録処理

クライアント：

1. 乱数 R を選び，誤り訂正符号で符号化した値 $C = \mathsf{Encode}_{ECC}(R)$，およびハッシュ値 $H(R)$ を生成する
2. 読み取った生体情報 $x = (x_1, ..., x_D) \in \{0,1\}^D$ に対して，$z = C \oplus x$ を計算する．さらに，GM 暗号を用いて $\mathsf{Enc}_{GM}(z_i)\ (i = 1, ..., D)$ を計算する
3. 認証サーバへ $\mathsf{Enc}_{GM}(z_i)\ (i = 1, ..., D)$ と $H(R)$ を送る

認証サーバ：

送られてきた情報 $\mathsf{Enc}_{GM}(z_i)\ (i = 1, ..., D), H(R)$ をテンプレート情報として登録する．

■認証処理

クライアント：

読み取った生体情報 $y = (y_1, ..., y_D) \in \{0,1\}^D$ に対して，GM 暗号で暗号化した暗号文 $\mathsf{Enc}_{GM}(y_i)\ (i = 1, ..., D)$ を認証サーバへ送る

認証サーバ：

1. テンプレート情報と送られてきた値から，$\mathsf{Enc}_{GM}(z_i \oplus y_i)\ (i = 1, ..., D)$ を計算する
2. 計算結果 $\mathsf{Enc}_{GM}(z_i \oplus y_i)\ (i = 1, ..., D)$ と $H(R)$ を復号センタへ送る

復号センタ：

1. $\mathsf{Enc}_{GM}(z_i \oplus y_i)\ (i = 1, ..., D)$ を復号し，$z \oplus y (= C \oplus x \oplus y)$ を求める
2. 乱数 $R' = \mathsf{Decode}_{ECC}(C_i \oplus x_i \oplus y_i)$ を求める．$H(R) \stackrel{?}{=} H(R')$ を確認し，成り立てば Accept，そうでないなら Reject を返す

この方式では，認証サーバが復号センタと結託しないと仮定した場合，悪意のある認証サーバは登録されている暗号文から生体情報を推定することは困難である．また，外部の第三者からの攻撃も考えられるが，登録情報を利用できる悪意のある認証サーバより行えることは少ないため，生体情報を求めることは困難である．

5 提案方式

Bringer らの方式は，(1) リプレイ耐性がない，(2) 誤り訂正符号を用いているためパラメータに制限がかかるという 2 つの問題点があった．

本研究では，それらの問題点を解決する 2 つの方式を提案する．提案方式においても，悪意のある認証サーバによる不正の耐性を持たせた方式となっている．

5.1 Bringer らの方式にリプレイ耐性を持たせた方式（方式 1）

第一の方式は，Bringer らの方式にリプレイ耐性を持たせた方式である．認証時に DH 鍵共有法を用いて，クライアントと認証サーバしか知ることができない値を生成し，平文をマスクする．生成された乱数はセッションごと異なるため，通信を盗聴している攻撃者が以前使用された認証クエリを用いてなりすましを行っても認証を通すことは困難である．

■セットアップ

復号センタ：

Bringer らの方式に加え，DH 鍵共有プロトコルより $(\mathbb{G}_{q'}, g', q')$ を生成し，公開鍵を $pk_{GM} = (N, a)$, $pk_{DH} = (\mathbb{G}_{q'}, g', q')$，秘密鍵を $pk_{GM} = (p, q)$ と設定する．

■登録処理

クライアント：

1. 乱数 R を選び，誤り訂正符号で符号化した値 $C = \mathsf{Encode}_{ECC}(R)$，およびハッシュ値 $H(R)$ を生成する
2. 読み取った生体情報 $x = (x_1, ..., x_D) \in \{0,1\}^D$ に対して，$z = C \oplus x$ を計算する．さらに，GM 暗号を用いて z の各ビット z_i を暗号化する
3. 認証サーバへ $\mathsf{Enc}_{GM}(z_i)\ (i = 1, ..., D)$ と $H(R)$ を送る

認証サーバ：

送られてきた情報 $\mathsf{Enc}_{GM}(z_i)\ (i = 1, ..., D), H(R)$ をテンプレート情報として登録する．

■認証処理

認証サーバ：

乱数 $r_{AS} \in \{1, ..., q' - 1\}$ を選び，$g'^{r_{AS}}$ をクライア

ントへ送る．

クライアント：

1. 乱数 $r_C \in \{1, ..., q'-1\}$ を選び，$g'^{r_{AS}r_C}$ を計算する
2. ハッシュ値 $\mathcal{H}_{DH} = H(g', g'^{r_{AS}}, g'^{r_C}, g'^{r_{AS}r_C})$ を求める．ただし，ハッシュ値 $\mathcal{H}_{DH}$ は長さ D ビットとする
3. 読み取った生体情報 $y = (y_1, ..., y_D) \in \{0,1\}^D$ に対して $y_{DH} = y \oplus \mathcal{H}_{DH}$ を計算し，各ビットの暗号文 $\mathsf{Enc}_{GM}(y_{DH,i})\ (i = 1, ..., D)$ を生成する
4. 認証サーバへ暗号文 $\mathsf{Enc}'_{GM}(y_{DH,i})\ (i = 1, ..., D)$ と g'^{r_C} を送る

認証サーバ：

1. 送られてきた値 g'^{r_C} から $g'^{r_{AS}r_C}$ を生成し，ハッシュ値 $\mathcal{H}'_{DH} = H(g', g'^{r_{AS}}, g'^{r_C}, g'^{r_{AS}r_C})$ を計算する（以下では，各ビットを $\mathcal{H}'_{DH,i}$ と記す）
2. ハッシュ値 $\mathcal{H}'_{DH}$ の各ビット $\mathcal{H}'_{DH,i}$ を用いて，暗号文 $\mathsf{Enc}_{GM}(z_i \oplus y_{DH,i} \oplus \mathcal{H}'_{DH,i})\ (i = 1, ..., D)$ を計算する
3. $\mathsf{Enc}_{GM}(z_i \oplus y_{DH,i} \oplus \mathcal{H}'_{DH,i})\ (i = 1, ..., D)$ と $H(R)$ を復号センタへ送る

復号センタ：

1. $Enc_{GM}(z_i \oplus y_{DH,i} \oplus \mathcal{H}'_{DH,i})\ (i = 1, ..., D)$ を復号し，$z_i \oplus y_{DH,i} \oplus \mathcal{H}'_{DH_i}\ (= (C_i \oplus x_i \oplus y_i \oplus \mathcal{H}_{DH,i} \oplus \mathcal{H}'_{DH,i})$ を求める
2. 乱数 $R' = \mathsf{Decode}_{ECC}(C_i \oplus x_i \oplus y_i \oplus \mathcal{H}_{DH,i} \oplus \mathcal{H}'_{DH,i})$ を求める．$H(R) \stackrel{?}{=} H(R')$ を確認し，成り立てば Accept，そうでないなら Reject を返す

通常の処理では，登録時の認証サーバ上で計算されたハッシュ値 H'_{DH} はクライアント上で計算されたハッシュ値 H'_{DH} と同じであるため，認証サーバで計算された暗号文は $\mathsf{Enc}_{GM}(C_i \oplus x_i \oplus y_i \oplus \mathcal{H}_{DH,i} \oplus \mathcal{H}'_{DH,i}) = \mathsf{Enc}_{GM}(C_i \oplus x_i \oplus y_i)$ である．しかし，以前のセッションの認証クエリを利用してリプレイ攻撃を行うと高確率で $H_{DH} \neq H'_{DH}$ となる（詳細は5.3を参照）ため，$\mathsf{Enc}_{GM}(C_i \oplus x_i \oplus y_i \oplus \mathcal{H}_{DH,i} \oplus \mathcal{H}'_{DH,i}) \neq \mathsf{Enc}_{GM}(C_i \oplus x_i \oplus y_i)$ となり高確率で Reject される．よって，本方式はリプレイ攻撃に耐性がある．

また，この対策は，後述の提案方式 2 に対しても適用可能であるが，プロトコルの記述上では省略する．

5.2 加法準同型暗号を用いた方式（方式 2）

第二の方式は，二値ベクトルで表される生体情報において，誤り訂正符号を用いず加法準同型暗号のみを用いて認証を行う方式である．Bringer らの方式は，誤り訂正符号を用いるため各パラメータの取りえる値に制限がある [10]．生体情報のベクトル長や認証の閾値は，生体情報の種類によりベクトル長が変わり，どのエラー率を重要視しているかにより閾値を変えられることが望ましい．しかし，誤り訂正符号を用いているとパラメータの生成や安全性の理由から，それらの値の範囲に制限がかかってしまう．

本方式では，加法準同型暗号を用いているため，マスクに用いる乱数や閾値などのパラメータの自由度を高めることができ，上記の問題を解決することができる．

■セットアップ

復号センタ：

セキュリティパラメータ k に対して $(N, g, p, q) \leftarrow \mathsf{Gen}_P(1^k)$ を生成し，公開鍵を $pk_P = (N, g)$，秘密鍵を $pk_P = (p, q)$ と設定する．また，閾値を θ とする．

■登録処理

クライアント：

1. 乱数 $m_i \in \mathbb{Z}_N\ (i = 1, ..., D)$ を選ぶ
2. 読み取った生体情報 $x = (x_1, ..., x_D) \in \{0,1\}^D$ に対して，Paillier 暗号を用いて以下のように各要素の暗号文 $\mathsf{Enc}_P(z_i)\ (i = 1, ..., D)$ を計算する

$$z_i = \begin{cases} m_i & (x_i = 0 \text{ の時}) \\ 1 - m_i & (x_i = 1 \text{ の時}) \end{cases}$$

3. 認証サーバへ $\mathsf{Enc}_P(z_i)\ (i = 1, ..., D)$ を，復号センタへ $\mathsf{Enc}_P(m_i)\ (i = 1, ..., D)$ を送る

認証サーバ：

送られてきた情報 $\mathsf{Enc}_P(z_i)\ (i = 1, ..., D)$ をテンプレート情報として登録する．

復号センタ：

送られてきた情報を復号し，$m_i\ (i = 1, ..., D)$ を登録する．

■認証処理

認証サーバ：

クライアントへテンプレート情報 $\mathsf{Enc}_P(z_i)\ (i = 1, ..., D)$ を送る

クライアント：

読み取った生体情報 $y = (y_1, ..., y_D) \in \{0,1\}^D$ に対して，Paillier 暗号を用いて以下のように暗号文 $\mathsf{Enc}_P(z'_i)\ (i = 1, ..., D)$ を生成し，認証サーバへ送る

$$z'_i = \begin{cases} z_i & (y_i = 0 \text{ の時}) \\ 1 - z_i & (y_i = 1 \text{ の時}) \end{cases}$$

認証サーバ：

$\mathsf{Enc}_P(z'_i)\ (i = 1, ..., D)$ を復号センタへ送る

復号センタ：

登録されている情報 $m_i\ (i = 1, ..., D)$ をもとに，送られてきた暗号文 $\mathsf{Enc}_P(z'_i)\ (i = 1, ..., D)$ の平文 z'_i に対して，$z'_i \stackrel{?}{=} m_i$ を確認する．各暗号文 $i(= 1, ..., D)$ において成り立たない暗号文の個数 e を閾値と比較し，$e \leq \theta$ なら Accept，$e > \theta$ なら Reject とする．

本方式では，二値であることを活かし，生体情報の値によって処理を変えることで登録・認証時の生体情報が同じか否か表している．上記のプロトコルにおいて，$x_i = y_i$ の時 $z'_i = m_i$ となり，$x_i \neq y_i$ の時 $z'_i = 1 - m_i$ となるため，復号センタは登録している情報をもとに認証が行える．

また，マスクに用いる乱数を復号センタに持たせることで，悪意のある認証サーバによる不正を防ぐことができる．詳細は，5.3節で記述する．

5.3 提案方式の安全性考察

本節では，提案方式の安全性について記述する．

提案方式 1 では，Bringer らの方式に DH 鍵共有法を適用してリプレイ耐性を持たせている．そのため，安全性は DH 鍵共有法と GM 暗号に依存する．また，以前に用いられた特定のセッションの暗号文を用いてリプレイ攻撃を成功させるためには，以前のセッション $\mathcal{S}$ 時の共通鍵情報 $g^{r_A s_T c}$ と攻撃時のセッション $\mathcal{S}'$ の共通鍵情報 $g'^{r_A s_T c}$ の差分を予想し，セッション $\mathcal{S}$ 時の認証クエリに付与する必要がある．そのため，複数ビットの $\{0,1\}$ を同時に予想するのが困難である必要がある．DH 鍵共有法の安全性から，それぞれの共通鍵情報を予想することは困難であるが，本方式の生体情報は二値ベクトルであるため，2 パターンに限られ，マスクにはマスク値の各ビットを用いているため，計 4 パターンに限られる．そのため，通信を盗聴している攻撃者に，暗号文のパターンを識別されない必要がある．そこで，それらのパターンを識別されないために，提案方式では IND-CPA 安全な暗号方式である GM 暗号や Paillier 暗号を用いている．また，リプレイ攻撃として，以前の複数のセッションで用いられた認証クエリを組み合わせて攻撃を行うことが考えられる．しかし，DH 鍵共有法の安全性により共有鍵情報の暗号文から，IND-CPA 安全な暗号方式の識別不可能性から，どのセッションの暗号文の要素を用いれば認証されるか予想するための情報を得られないため，この攻撃方法も防ぐことができる．

提案方式 2 において，悪意のある認証サーバが登録情報を用いて生体情報を知ることができるか考察する．まず，Paillier 暗号の安全性により登録されている暗号文から平文を求めることは困難である．また，登録情報を用いて認証処理を行い，その認証結果から生体情報を推定することも困難であると考えられる．認証結果から生体情報を推定するには，認証結果が Accept（もしくは，Reject）の時，生体情報のベクトルのある要素 x_j が 0(もしくは，1) であるとわかるように関連付ける必要がある．そのためには，要素 j 番目以外の $(D-\theta-1)$ 個の要素だけに対して，マッチング結果としてマスクに用いている乱数の暗号文（ユークリッド距離が 0 の暗号文）を送る必要がある．しかし，認証サーバはマスク用の乱数を知らないため，意図してマスク用乱数の暗号文を送ることは困難である．生体情報 $\{0,1\}$ の出現頻度が等確率であると仮定した時，ある要素のマッチング結果がマスクしている乱数の暗号文である確率は $\frac{1}{2}$ であり，一つの要素のみを予想できる確率は $\frac{1}{2}$ ある．ただし，攻撃を成功させるには複数要素（生体情報長）に対して同時に生体情報を予想しなければならない．そのため，認証の結果から生体情報を推定するのは困難だと考えられる．考えられる攻撃として，認証時の生体情報を任意に決めて認証処理を行う方法がある．この方法からもとの生体情報を推定する困難さは，第三者が任意のユーザになりすまして認証を行い，もとの生体情報を推定する困難さと同等なので生体情報を知るのは困難であると考えられる．

また，伊豆らにより，悪意のある認証サーバが登録されている情報を用いて認証処理を行うことで，その認証の結果からもとの生体情報が推定できる攻撃 [7] が報告されている．認証サーバ上に登録されている任意のユーザのテンプレート情報において，暗号方式の準同型性を用いて暗号文の生体情報に閾値以上の値をかけてマッチング結果を偽造することで，その認証結果から生体情報が 0 であるか 1 であるか推定することができる．しかし，提案方式において，上記の攻撃を行うのは困難である．提案方式のテンプレート情報の平文は，乱数によりマスクされており，生体情報だけに対して値を掛けることができない．攻撃を行うには，乱数を打ち消さなければならないが，認証サーバは乱数の値を知ることができないため攻撃を成功させることができないと考えられる．ただし，伊豆らの攻撃は多値ベクトルで表現される生体情報に対しても有効であったが，本方式は二値ベクトルしか扱えない．

6 提案方式の発展

提案方式 2 は，Bringer らの方式とは異なり，復号センタがユーザ固有の秘密情報を持っている必要がある．そのため，復号センタに負担がかかってしまう．本節では，この欠点を改善する方法を提案する．

■乱数を認証サーバに持たせる工夫

Bringer らの方式では，復号センタは暗号方式の秘密鍵しか登録していなかった．それに対し，提案方式 2 では，ユーザごとのマスク用の乱数を復号センタが持っており，復号センタへの負担が大きくなっている．そこで，マスク用乱数を認証サーバへ持たせる改善を図る．

5.2 節のプロトコルでは，復号センタに生体情報の要素数だけの乱数を持たせていたが，本改良方式では認証サーバが乱数を登録する．この際，マスク値の選び方には注意が必要である．仮に，生体情報長 D 個の乱数をマスク値として認証サーバに持たせると，悪意のある認証サーバは任意のマスク値を正規のマスク値として復号センタへ送ることが可能となる．そのため，認証サーバは以下のような方法で任意のユーザのもとの生体情報を推定することが可能となってしまう．

▼提案方式 2 に対する悪意のある認証サーバの攻撃例

任意のユーザの登録情報 $\mathsf{Enc}_P(z_i), \mathsf{Enc}_P(m_i)$ $(i = 1, ..., D)$，閾値 θ とする．j 番目の要素 x_j に対して，任意の乱数 m'_i としてマッチング結果を $\mathsf{Enc}_P(z'_i)$: $z' = (1 - m'_1, ..., 1 - m'_\theta, m'_{\theta+1}, ..., m'_{j-1}, z_j, m'_{j+1}, ..., m'_D)$，認証判定用乱数を $\mathsf{Enc}_P(m'_i)$: $m' = (m'_1, ..., m'_{k-1}, m_j, m'_{k+1}, ..., m'_D)$ と偽造して復号センタへ送る．この時，$z_j = m_i$ ならマッチング結果と認証判定用乱数が異なる要素数は閾値 θ 個となり Accept が返され，$z_j = 1 - m_i$ なら Reject が返される．よって，認証の結果からもとの生体情報が推定できる．

従って，上記のような攻撃を防ぐために，乱数 $m_1 \in \mathbb{Z}_N$ を一つだけ選び，他のマスク値を $m_i = m_1^i$ $(i = 2, ..., D)$ によって生成する．さらに，ブロック暗号など乗法準同型性のない暗号方式で暗号化し，認証サーバ上で記憶する．そのため，秘密鍵を保持している復号センタのみが m_1 を知ることができ，$m_2, ..., m_D$ を計算できる．この改良プロトコルにおいて，悪意のある認証サーバが生体情報を推定するには，乱数 m_1 を予想するか，生体情報全てを同時に予想するしかないが，それを行え

表 1　既存方式との比較

方式	リプレイ耐性	認証サーバによる不正耐性	パラメータの自由度	多値	通信量		ユーザの秘密情報
					$\mathcal{C} \leftrightarrow \mathcal{AS}$	$\mathcal{AS} \leftrightarrow \mathcal{DC}$	
Bringer らの方式 [1]	×	○	△	×	$D\|GM\|$	$D\|GM\|$	必要なし
服部らの方式 [4]	×	×	○	○	$D\|BGN_{\mathbb{G}}\|$	$\|BGN_{\mathbb{G}_T}\|$	必要なし
伊豆らの方式 [8]	×	○	○	○	$D\|BGN_{\mathbb{G}}\|$	$\|BGN_{\mathbb{G}_T}\|$	必要
提案方式 1	○	○	△	×	$D\|GM\| + 2\|DH\|$	$D\|GM\|$	必要なし
提案方式 2	○	○	○	×	$2D\|P\| + 2\|DH\|$	$D\|P\|$	必要なし

る可能性は低いと考えられる．

7 既存方式との比較

本節では，既存方式である Bringer らの方式 [1]，服部らの方式 [4]，服部らの方式を基本とした伊豆らの方式 [8] と，2 つの提案方式の比較を表 1に示す．服部らの方式は，任意回の加算と 1 回の乗算を行える準同型性を有した BGN 暗号を用いて登録・認証時のユークリッド距離を求めて認証を行う秘匿認証方式であり，伊豆らの方式は，服部らの方式に認証サーバによる不正の耐性を持たせた方式である．伊豆らの方式は，登録時の乱数をユーザに持たせ，登録時に用いた乱数を知っているユーザ以外は正しい認証処理を行うことができないという特徴がある．

表 1の通信量において，$|GM|, |DH|, |P|, |BGN_{\mathbb{G}}|, |BGN_{\mathbb{G}_T}|$ はそれぞれの暗号方式の暗号文の大きさ，また，$\mathcal{C} \leftrightarrow \mathcal{AS}$ はクライアントー認証サーバ間，$\mathcal{AS} \leftrightarrow \mathcal{DC}$ は認証サーバー復号センタ間の通信を表しており，写像前の暗号文を $BGN_{\mathbb{G}}$，写像後の暗号文を $BGN_{\mathbb{G}_T}$ とする．また，Bringer らの方式と提案方式においては PIR の処理を除いた通信量となっている．

提案方式 1 は，Bringer らの方式と比べ，リプレイ耐性があり，通信量がほとんど同じである．また，提案方式 2 は，パラメータの自由度が高く，リプレイ耐性と悪意のある認証サーバによる不正への耐性がある．ただし，伊豆らの方式と比較すると，提案方式 2 は多値ベクトルに対応しておらず，通信量も多いという欠点がある．しかし，伊豆らの方式は，リプレイ耐性がなく，ユーザが秘密情報を保持しておかなければならないためユーザへ負担がかかってしまう．

8 まとめ

Bringer らの方式が有していた 2 つの問題点において，それぞれの問題を解決した方式を提案した．また，提案方式 2 は，悪意のある認証サーバによる不正にも耐性があり，伊豆らが報告した悪意のある認証サーバによる攻撃手法も防ぐことができる．ただし，他の既存方式に比べ通信量が多いといったデメリットが存在する．

今後の課題としては，処理の効率化と通信量の削減，およびベクトルの多値化が挙げられる．また，本方式では，本稿のモデル外の攻撃が可能となっているため，悪意のある復号センタなどを仮定した他のモデルにおける不正にも耐性のある方式への拡張も挙げられる．

参考文献

[1] J. Bringer and H. Chabanne, “An Authentication Protocol with Encrypted Biometric Data”, AFRICACRYOT 2008, LNCS 5023, pp.109-124, Springer-Verlag, 2008.

[2] W. Diffie and M. Hellman, “New directions in cryptography”, IEEE Transactions on Information Theory-22, pp.644-654, 1976.

[3] S. Goldwasser and S. Micali,“Probabilistic Encryption”, J. Computer and System Sciences, vol.28, pp. 270-299, 1984.

[4] M. Hattori, N. Matsuda, T. Ito, Y. Shibata, K. Takashima, and T. Yoneda, “Provably-secure Cancelable Biometrics Using 2-DNF Evaluation”, Journal of Information Processing, Vol.20, No.2, pp.496-507, IPSJ, 2012.

[5] T. Hirano, T. Ito, Y. Kawai, N. Matsuda, T. Yamamoto and T. Munaka, “A Practical Attack to AINA2014’s Countermeasure for Cancelable Biometric Authentication Protocols”, ISITA2016, Monterey, California, USA, October 30-November 2, 2016

[6] 伊豆, 酒見, 武仲, 鳥居, “キャンセラブル生体認証方式の安全性について (その 1)”, 第 3 回バイオメトリクスと認識・認証シンポジウム (SBRA 2013), IEICE, 2013.

[7] 伊豆, 酒見, 武仲, 鳥居, “キャンセラブル生体認証方式の安全性について (その 2)”, 信学技報 ISEC2013-79, IEICE, 2013.

[8] 伊豆, 酒見, 武仲, 鳥居, “キャンセラブル生体認証方式の安全性について (その 3)”, 第 31 回 暗号と情報セキュリティシンポジウム (SCIS 2014), IEICE, 2014.

[9] P. Pallier,“Public-key cryptosystems based on composite degree residuosity classes”, EUROCRYPT 1999, J. Stern (Ed.), Vol.1592 of LNCS, pp.223238, 1999.

[10] P. Tuyls, A. H. M. Akkermans, T. A. M. Kevenaar, G. J. Schrijen, A. M. Bazen, and R. N. J. Veldhuis, “Practical biometric authentication with template protection,” Proceedings of the 5th International Conference, AVBPA 2005, pp.436446, 2005.