格子マスキング利用のKey Binding技術における テンプレート更新機能

杉村 由花[1,a)] 安田 雅哉[1,b)] 山田 茂史[1,c)] 安部 登樹[1,d)] 新崎 卓[1,e)]

概要：生体認証において，テンプレートと呼ばれる登録生体データを何らかの変換により保護しつつ，照合処理を行うテンプレート保護型生体認証技術が近年盛んに研究されている．特に，テンプレート保護型生体認証技術の一つである key binding は，生体データとユーザ固有の鍵の結合データから補助情報を生成し，登録時と照合時の生体データが十分に近い場合のみユーザ固有の鍵が抽出できる特殊な技術である．Key binding で利用するユーザ固有の鍵として，電子署名や文書などの情報を扱うことで，電子署名や文書暗号化／復号などに応用でき，認証判定だけでなく様々な利用形態の実現が期待されている技術でもある．我々は 2013 年 7 月の情報セキュリティ研究会（ISEC）にて，key binding の新しい実現方式を提案した．この実現方式は，通常の乱数付加アプローチに加え，生体認証特有のデータの揺らぎ吸収のため，格子と呼ばれる数学的概念を利用した格子マスキングを新しく導入したものである．本論文では，格子マスキングによる key binding 技術において，テンプレートの更新処理を行う方法を提案する．

キーワード：生体認証，格子マスキング，テンプレート保護，生体暗号，key binding

Template update method in key binding technology using lattice masking

Yuka SUGIMURA[1,a)] Masaya YASUDA[1,b)] Shigefumi YAMADA[1,c)] Narishige ABE[1,d)] Takashi SHINZAKI[1,e)]

1. はじめに

1.1 生体認証技術

生体認証技術とは，人間の身体的特徴（指紋・静脈・虹彩・顔など）もしくは行動的特徴（筆跡・歩容など）を用いて本人認証を行う技術である．一般的な生体認証方式では，事前に認証に利用する特徴データを本人から採取しデータベースに登録し（登録したデータをテンプレートとよぶ），認証の際には認証対象者から同じ特徴データを採取しテンプレートと比較することで認証する．生体認証技術は，従来から利用されている ID とパスワードによる認証や，トークン（ID カードなど）による認証に比べ優れた点がいくつもあるため，近年金融機関における取引や，高セキュリティエリアにおける入退室管理，また企業における勤怠管理などにおいて利用が進んでいる．

生体認証技術の利点としては，パスワードやトークンを利用者が管理・携帯する必要がなく便利な点がある．また，利用者の身体を利用したものであるため，パスワードやトークンのように盗まれて他者に不正アクセスを許すおそれがないことも利点である．一方，生体認証技術特有の弱点として，テンプレートを変更できない点があげられる．パスワードやトークンは盗難・漏えいした場合，既存のものを無効化し再発行するのが容易である．一方生体認証では，認証に使用する部位は限られている（顔は 1 つ，指は 10 本など）ため，盗まれてもほかのデータに差し替えることはできず，盗まれたテンプレートによるアクセスを許すおそれがある．また認証部位が限られることで，複数のシ

[1] 株式会社富士通研究所
〒 211-8588 神奈川県川崎市中原区上小田中 4-1-1
FUJITSU LABORATORIES LTD.
4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki, Kanagawa 211-8588, Japan
a) sugimura.yuka@jp.fujitsu.com
b) yasuda.masaya@jp.fujitsu.com
c) yamada.shige@jp.fujitsu.com
d) abe.narishige@jp.fujitsu.com
e) shinzaki@jp.fujitsu.com

ステムに同一部位を登録せざるを得ない．そのため，一つのシステムからのテンプレート漏えいの際に他のシステムにもアクセスを許すおそれがあるほか，テンプレートを通じて他のシステムの利用者と名寄せされてしまい利用者のプライバシーが脅かされるおそれもある．

1.2 テンプレート保護型生体認証技術

従来からあるテンプレート保護の方法としては，テンプレートをデータベースに格納する際に暗号化することが挙げられる．この方法では照合時に生体情報を復元する必要があり，その際にテンプレートが漏えいする危険があった．この問題を解決できる技術として，近年ではテンプレート保護型生体認証技術が着目されている．テンプレート保護型生体認証技術では，生体情報そのままではなく，何らかの不可逆変換を施したもの（セキュアテンプレート）を保存する．また認証の際にも，元の生体情報を復元することなく照合する．こうすることで，テンプレート漏えい時には変換方法を変えることでテンプレートの無効化および再発行が行える．また複数システムに同一部位を登録しても，変換方法が異なればあるシステムのテンプレートではほかのシステムにはアクセスできず，名寄せされるおそれもない．セキュアテンプレートから元の生体情報を復元することは困難であるため，物理的な偽造生体を作成しての不正アクセスも防ぐことができる．

テンプレート保護型生体認証技術では，次の要件が求められる（[5], 3 章）：

(a) Diversity: 変換後のテンプレートは複数のデータベース間でクロスマッチングできてはならない．

(b) Revocability: 漏えいしたテンプレートを無効化し，同じ生体データに基づいた新しいテンプレートを容易に発行できなければならない．

(c) Security: 変換後のテンプレートから元のテンプレートを復元することが計算量的に困難でなくてはならない．

(d) Performance: 生体認証システムの認証性能（他人受入率および本人拒否率）を劣化させてはならない．

これまでに提案されているテンプレート保護型生体認証技術は，テンプレート保護の手法により，以下のように大別される（[5] を参照）．

- Feature Transform（特徴変換：以下の 2 つに分類）
  1. Salting（又は Biohashing）：生体データとユーザ固有の乱数やパスワードを利用して，テンプレートを保護する手法．
  2. Noninvertible transform: 不可逆な（一方向性）変換関数によりテンプレートを変換したまま，照合処理を行う手法．
- Biometric cryptosystem（生体暗号：以下の 2 つに分類）
  3. Key binding: 生体データとユーザ固有の鍵の結合データから補助情報を生成し，登録時と照合時の生体データが十分に近い場合のみ，ユーザ固有の鍵を抽出する手法．
  4. Key generation: 生体データから 1 つの鍵を生成し，鍵を用いた照合処理を行う手法．

Key binding は上記 4 つのテンプレート保護手法の中でも，ユーザ固有の鍵として電子署名や文書などの情報を扱うことで電子署名や文書暗号化／復号などに応用でき，認証判定だけでなく様々な利用形態の実現が期待されている技術である．さらに，salting や noninvertible transform 手法とは異なり，key binding では照合時に変換のためのパラメータの入力を必要としないため，運用が容易であるというメリットもある．一方，現在 key binding を実現する方法として代表的な fuzzy vault や fuzzy commitment では，誤り訂正符号によって登録・照合データ間の揺らぎを吸収しており，本人間の揺らぎ吸収と他人受け入れの防止を両立するようなパラメータを決定するのが難しいという課題がある．さらに，上に挙げたテンプレート保護要件のうち，(a) Diversity と (b) Revocability を満たすのが難しいという問題も知られている（[5]）．

我々は 2013 年 7 月の情報セキュリティ研究会（ISEC）[13] にて，key binding の新しい実現方式を提案した．我々の実現方式は，通常の乱数付加アプローチに加え，生体認証特有のデータの揺らぎ吸収のために，格子と呼ばれる数学的概念を利用した格子マスキングを新しく導入し，認証に使用する各特徴量について，登録・照合間の揺らぎが格子の大きさ以下であれば正しくマッチングできるものである．具体的な構成は次のとおりである．生体データの登録時，サーバは格子基底を生成しこれを保存するとともに，格子基底からランダム格子元作成用のベクトルの集合をクライアントに送る．クライアントは送られたベクトル集合からランダム格子元を作成し，生体データと鍵に足し合わせてサーバに送る．照合時，サーバは同様のベクトル集合をクライアントに送り，クライアントは同様に作成したランダム格子元を生体データと足し合わせサーバに送る．サーバは mod 演算と呼ばれる演算によりランダム格子元を無視したマッチングを行う．登録・照合データ間の距離が近い場合にのみマッチングが成功し，登録した鍵が正しく抽出される．従来方式に対し，我々の方式では格子基底をシステムごとに変えることでシステム間でのクロスマッチングが行えず，また漏えい時に格子基底を取り換えることで同一生体データから異なるテンプレートを作成できる．すなわち (a) Diversity および (b) Revocability の要件を満たす．また統計的解析にも強く，サーバに生の生体データを晒すことなく登録・照合が行えるという利点ももつ．

1.3 テンプレート更新機能

生体認証に使用する特徴量は変化しにくいものではあるが，年月の経過，周囲の環境変化や認証操作への慣れ等によって，取得される生体データに微小な変化が生じ，長期的に認証精度が低下しうる．これを防ぐため，生体認証システムの中には照合時に取得した生体データを使用してテンプレートを自動更新する機能を有するものがある．しかし key binding 技術においては，テンプレートは生体データと鍵を結合したものであるため，照合に用いる生体データで単純に置き換えることはできなかった．

本論文では，格子マスキング利用の key binding 技術 [13] において，テンプレートの更新処理を行う方法を提案する．提案手法では，認証成功時に受け取った秘匿生体データと抽出した鍵を足し合わせることで，更新用のテンプレートを作成する．この方法により，我々の方式においてもテンプレートの更新を行うことができ，認証精度の低下を防ぐことができる．さらにこの方法は，認証成功していなければ鍵が取り出せず更新不可能であるため，悪意ある者によってテンプレート更新が引き起こされることも防いでいる．また，更新の方法として元のテンプレートと両立するものも提案する．この方法には，季節による生体情報変動への効果や，品質の低い生体データを用いて更新したことによる認証精度低下を防ぐ効果がある．

2. Key binding 技術

Key binding（key-binding biometric cryptosystem）はテンプレート保護型生体認証技術の 1 つで，生体データとユーザ固有の鍵の結合データから補助情報を生成し，登録時と照合時の生体データが十分に近い場合に限り，ユーザ固有の鍵を抽出（復号）可能な scheme である（詳細は [5], 3.3 節を参照）．**図 1** は key binding 技術を用いた認証メカニズムの概要を示しており，その認証手順の概要は以下である：

1. 登録時，生体テンプレート T とユーザ固有の鍵 K の結合データから生成した補助情報 $H = F(T)$（図 1 内では Helper data と表記）を登録しておく．

2. 照合時に，照合用生体データ Q が生体テンプレート T と十分に近かった場合のみ，登録時に結合した鍵 K が抽出でき，この鍵 K を用いて認証チェックを行う．

本技術を用いることで，生体テンプレート T そのものを登録することなく，生体データに基づく認証を実現することが可能である [*1]．また，本技術では Match/Non-match の認証判定だけでなく，ユーザ固有の鍵 K 自体を抽出できるので，鍵 K として ID や電子署名・文書などの情報を扱うことで，ID なし認証や電子署名・文書暗号化/復号などに応用でき，さまざまな利用形態の実現が期待できる技術として注目されている（[5], 4.3 節を参照）．一方で，§1 で説明したテンプレート保護型生体認証技術としてのセキュリティ要件の (c) Security として，

「ユーザの生体情報 Q を知らずに，補助情報 H から，生体テンプレート T またはユーザ固有の鍵 K を求めるのが計算量的に困難」

の性質を満たすことが本技術では強く求められる．現在 key binding 技術を実現する方法としては，以下でそれぞれ説明する fuzzy commitment や fuzzy vault などの誤り訂正符号を利用した方式が代表的である（その他の実現方式として，shielding functions[9] や distributed source coding[2] なども知られている）．

[*1] 認証チェック方法としては，保管しておいたユーザ固有鍵 K のハッシュ値を検証する認証手段や，公開鍵を保管し K を秘密鍵とする PKI に基づく認証手段などの暗号技術に基づく認証方式が可能．

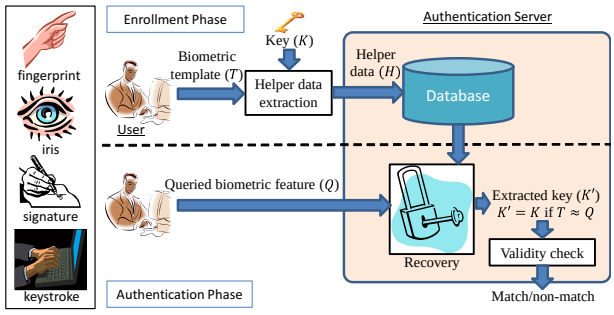

図 1 Key binding 技術を用いた認証メカニズムの概要

2.1 Fuzzy commitment

Fuzzy commitment は，1999 年 Juels-Wattenberg[7] によって提案された手法である．Fuzzy commitment では，登録時，誤り訂正符号 C の符号語 w（=key binding の枠組みのユーザ固有の鍵 K に対応）と生体テンプレート x に対して，組 $(x-w, h(w))$ を補助情報として登録する（ただし，h はハッシュ関数とする）．照合時，ユーザの生体特徴ベクトル x' が与えられたとき，まず $w' = x' - (x-w) = w + \delta$ $(\delta = x' - x)$ を計算する．2 つの生体特徴ベクトル x, x' の揺らぎが十分に小さいとき（つまり，誤り訂正符号の復号可能範囲内であれば），w' は符号語 w に復号でき，ハッシュ値を比較することで認証チェックが行える仕組みである．

2.2 Fuzzy vault

一方，fuzzy vault は，2002 年に Juel-Sudan[6] によって提案された手法である．2003 年には，Clancy-Lin-Kiyavash[1] によって指紋認証への応用方式が提案されており，特に fingerprint vault と呼ばれている．また，顔認証 [3]，虹彩認証 [8]，署名認識 [4] などへの応用方式も数多く提案されており，fuzzy vault は現在最も代表的な key binding 実現方式である．Fuzzy vault では，まず秘密情報 s（=key binding の枠組みのユーザ固有の鍵 K に対応）を任意の情

報 A を用いて暗号化（Lock 処理）する．復号（Unlock 処理）には，A と同じ形式の情報 B を用いて，2 つの情報 A と B の大部分が一致した場合のみ，誤り訂正符号により秘密鍵 s を復号することができる．以下で，fuzzy vault の処理手順の概要を紹介しておく：

- Lock 処理
  秘密情報 s と情報 $A = \{a_1, a_2, \ldots, a_n\}$ に対し，s から生成されたランダムな多項式 $p(X)$ を用意し，$(x_i, y_i) = (a_i, p(a_i))$ とおく（ここで，$p(0) = s$ とする）．さらに，$i = n{+}1, \ldots, r$ に対し，$x_i \notin A, y_i \neq p(x_i)$ となるような疑似データ群 (x_i, y_i)（chaff 集合と呼ばれる）を追加する．最後に，A から生成されたデータと疑似データの判別を困難にするよう，r 個のデータ (x_i, y_i) の順番をシャッフルし，これを Lock 情報 R（Vault と呼ばれる．Key binding の枠組みでいう補助情報）とする．
- Unlock 処理
  A と同様の形式情報 $B = \{b_1, b_2, \ldots, b_n\}$ を用いて，Lock 情報 R から b_i と一致するデータ x_j を探索し，一致集合 $Q = \{(b_i = x_j, y_j)\}$ を生成する．このとき，Lock 情報 R の部分集合 Q から誤り訂正復号により多項式 $p(X)$ を復元できた場合のみ，秘密情報 $s = p(0)$ を得ることができる．

Fuzzy vault は，Lock 情報 R から多項式 $p(X)$ を求める多項式復元問題と chaff 集合の付加によって，テンプレート保護に関するセキュリティ要件 (c) Security が保たれている．

2.3 既存 key binding 実現方式の課題

上記で説明した fuzzy commitment や fuzzy vault による key binding 実現方式では，方式独自のパラメータに加えて誤り訂正符号を併用しているため，データの揺らぎ吸収と他人受け入れ率を両立させるパラメータ設定が非常に難しいという課題がある（[12], 3.3 節参考）．つまり，§1 で説明したテンプレート保護に関する要件 (d) Performance を満たすことが困難である．さらに，テンプレート保護型生体認証のセキュリティ要件である (a) Diversity と (b) Revocability の 2 条件を満たすことが一般的に難しいという課題も知られている（詳細は [5], 3.3 節と 4.3 節を参照）．例えば，fuzzy vault において，同じ秘密情報 s と情報 A から生成された 2 つの Lock 情報 R と R' があったとする．このとき，R と R' に含まれる疑似データである chaff 情報がどんなに異なっていても，情報 $A = \{a_1, \ldots, a_n\}$ から生成された正規データ $(a_i, p(a_i))$ の集合はすべて一致するため，R と R' は同一の Lock 情報とみなすことができ，(a) Diversity と (b) Revocability の 2 条件を満たさないことが分かる（fuzzy commitment においても，§2.1 で説明した方式の構成から，(a), (b) の 2 条件を満たさないことが明らかに分かる）．

3. 格子マスキング利用の key binding 実現手法

本節では，[13] で提案された key binding に適用可能な格子マスキング，および格子マスキングを利用した key binding 技術の特長について紹介する．

3.1 プライバシー保護利活用技術と乱数マスク化

近年，プライバシー情報の保護とその利活用のバランスを適切に管理しながら，利用価値の高い情報を安全かつ有効に活用するプライバシー保護データマイニングの研究が，生体認証も含めたさまざまな分野で盛んに行われている．現在知られているプライバシー保護データマイニング手法には，大きく分けて匿名化・乱数マスク化（ランダム化）・MPC（=Multi-Party Computation）・準同型暗号の 4 つのアプローチがある（文献 [10], [11] 参考）．

ここでは，乱数マスク化について着目する．プライバシー保護データマイニングにおける乱数マスク化によるアプローチでは，データにランダムな摂動を加えることによって，データのプライバシーを保護しつつデータの大域的な統計情報を算出する方法である．しかし，通常の乱数マスク化アプローチは処理性能に優れているものの，データに乱数を加えるため厳密な計算には向かず，生体情報固有のデータ揺らぎ吸収という特殊な処理が求められる key binding 技術にそのまま適用することは難しい．

3.2 格子マスキング技術の導入

そこで，データに乱数を付加しつつ，データの揺らぎ吸収を行うために，格子と呼ばれる数学的概念を用いた格子マスキングという技術を新たに導入した．格子とは無限の規則的な網目の交点の集合のことで，数学的には線形独立なベクトルの整数結合のなす集合である．具体的には，n 個の線形独立なベクトル $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n \in \mathbb{R}^n$（簡単のため，実数ベクトル空間 $\mathbb{R}^n$ 上で考える）に対して，これら n 個のベクトルの整数結合の集合

$$
\begin{aligned}
L &= \mathcal{L}(\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n) \\
&= \left\{ \sum_{i=1}^{n} n_i \vec{v}_i \ (n_i\text{:整数}, \forall i) \right\}
\end{aligned}
$$

のことを格子と呼ぶ．秘匿したいデータ $\vec{A} \in \mathbb{R}^n$ に対して，通常の乱数マスク化とは異なり，ランダムな格子 L の元 $\vec{r} = \sum_{i=1}^{n} n_i \vec{v}_i \in L$（整数 n_i をランダムに選ぶ）を付加した秘匿情報

$$\vec{c} = \vec{A} + \vec{r} \in \mathbb{R}^n$$

を考える．さらに，2 つのデータの揺らぎ吸収のために，以下で定義される格子理論特有の写像を用いる：格子 L の基

底 $\{\vec{v}_1,\ldots,\vec{v}_n\}$ から得られる $n\times n$-行列 $V=(\vec{v}_1,\ldots,\vec{v}_n)^T$ を考える（各ベクトル $\vec{v}_i$ を行ベクトルとして表現し，V は行ベクトル $\vec{v}_i$ を縦に n 個並べた行列である．右肩の ^{T} は行列の転置操作を表す）．この行列 V を格子 L の基底行列と呼び，その基底行列 V から一意的に定まる格子 L の基本領域を

$$P=\mathcal{P}(V)=\mathcal{P}(\vec{v}_1,\ldots,\vec{v}_n)\subset\mathbb{R}^n$$
$$=\left\{\sum_{i=1}^{n}a_i\vec{v}_i,\ \left(-\frac{1}{2}\le a_i<\frac{1}{2},\forall i\right)\right\}$$

と定義する．このとき，基底行列 V から定まる写像を

$$\mathbb{R}^n\to P,\quad \vec{z}\mapsto \vec{z}\bmod V:=\vec{z}-\lceil\vec{z}\times V^{-1}\rfloor\times V$$

と定義する．ただし，$\lceil\vec{v}\rfloor$ はベクトル $\vec{v}\in\mathbb{R}^n$ の各実数係数に対する最近似整数ベクトルを表す．よって，$\lceil\vec{z}\times V^{-1}\rfloor\times V$ は格子 L の元であり，特にその格子元はベクトル $\vec{z}$ の最近似格子元となり，mod$V$ 写像とは元ベクトル $\vec{z}$ から最近似格子元を差し引いた写像である．mod$V$ 写像の定義から，特に

(i) $\vec{a}+\vec{z}\bmod V=\vec{a}\bmod V\quad(\vec{z}\in L)$

(ii) $\vec{z}\bmod V=\vec{z}\quad(\vec{z}\in P)$

という性質が成り立つ（線形性は成り立たないことに注意）．

3.3 Key binding への適用

ここでは，§3.2 で導入した格子マスキング技術を key binding へ適用する方法を述べる．適用アイデアの概要としては，

- ランダムな格子元を付加することによりデータを秘匿し，
- データを秘匿したままデータの揺らぎ吸収のために，格子から定まる mod$V$ 写像を利用して行う

の 2 点である．

3.3.1 処理手順の概要

格子マスキングを利用した key binding 処理手順の概要を以下で示す（key binding の認証メカニズムとして，図 1 も参照）：簡単のため，ここでは n 次元ベクトルを平文データとして扱い，ユーザ固有の鍵 K として 1 つの実数を扱うことにする．特に照合後にテンプレート更新を行う手順について**図 2** に示す（クライアントとデータベースを持つサーバの 2 者間認証モデルで説明する）．

準備： サーバは，まず $(n+2)$-個の線形独立なベクトル $\vec{v}_1,\vec{v}_2,\ldots,\vec{v}_{n+2}\in\mathbb{R}^{n+2}$ を生成する．そこで，$(n+2)$-個のベクトル $\vec{v}_i$ から生成される格子を L とし，その $(n+2)\times(n+2)$-基底行列を $V=(\vec{v}_1,\ldots,\vec{v}_{n+2})^T$ とする（ここで，サーバのみが基底行列 V を保持していることに注意しておく）．

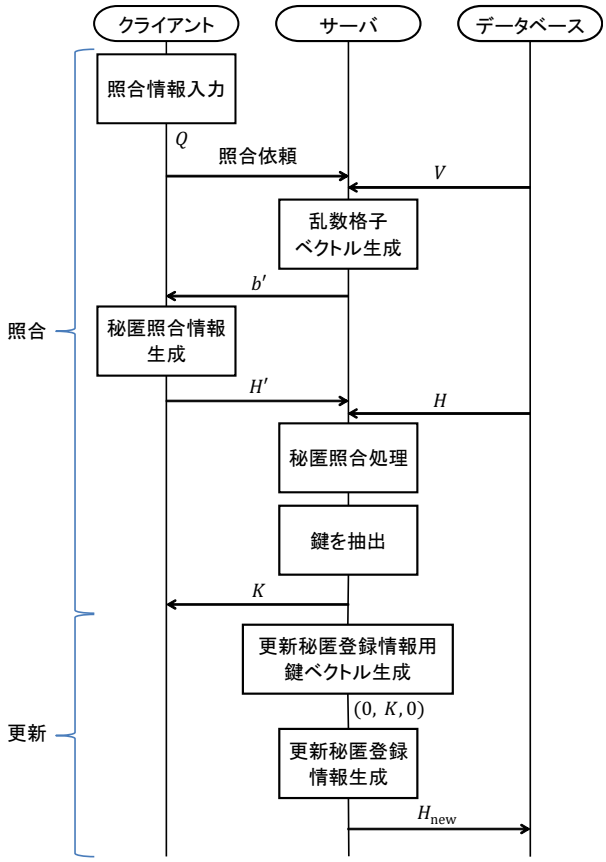

図 2 照合および更新手順

Fig. 2 Verification and template update flow

登録時：

1) まず，サーバは適当な格子元 $\vec{b}_1\in\mathbb{R}^{n+2}$ を作成し，クライアントに送信する．
2) クライアントは，n 次元テンプレート情報 $\vec{T}\in\mathbb{R}^n$ と鍵 $K\in\mathbb{R}$ に対して，テンプレート情報 $\vec{T}$ と鍵 K の結合データの最終成分に 0 を追加した $(n+2)$-次元ベクトル $(\vec{T},K,0)$ を生成する．次に，整数 r をランダムに生成し，サーバから受信した格子元 $\vec{b}_1\in\mathbb{R}^{n+2}$ を元に，格子マスキングによる秘匿情報

$$\vec{H}=\left(\vec{T},K,0\right)+r\vec{b}_1\in\mathbb{R}^{n+2}\qquad(1)$$

を生成し，サーバのデータベースに登録しておく．

照合時：

1) 登録時と同様，サーバは適当な格子元 $\vec{b}_2\in\mathbb{R}^{n+2}$ を作成し，クライアントに送信する（登録時と照合時に送信する格子元 $\vec{b}_1$ と $\vec{b}_2$ は全く異なるものを生成してよい）．
2) クライアントは，n 次元照合データ $\vec{Q}\in\mathbb{R}^n$ に対し，整数乱数 r' を生成し，サーバから受信した格子元 $\vec{b}_2\in\mathbb{R}^{n+2}$ から，格子マスキングによる秘匿情報

$$\vec{H}'=\left(\vec{Q},0,0\right)+r'\vec{b}_2\in\mathbb{R}^{n+2}$$

を生成し，サーバに送信する．

3) サーバは，鍵 K の抽出のため，登録秘匿情報 $\vec{H}$ と照合秘匿情報 $\vec{H}'$ の差分ベクトル $\vec{z} = \vec{H} - \vec{H}' \in \mathbb{R}^{n+2}$ を算出し，$\vec{z} \bmod V$ を計算する．計算結果 $\vec{z} \bmod V \in \mathbb{R}^{n+2}$ の最終成分が 0 の場合（つまり，登録データ $\vec{T}$ と照合データ $\vec{Q}$ の揺らぎが小さい場合），$\vec{z} \bmod V$ の第 $(n+1)$-成分として鍵 K が（高い確率で）抽出可能で，これを用いて認証チェックを行うことができる．

ここで，いくつかの注意点を述べておく．上記の構成では，ユーザ固有の鍵として実数値のみを考えていたが，複数の実数値で構成されるベクトル情報でも同様の処理が可能である．さらに，上記では，認証判定用に，ベクトルの最終成分に 0 を付加した場合を説明したが，この最終成分の個数を増やすことで認証判定成功の確率を格段に増大させることが可能である．

3.3.2 照合時におけるデータ揺らぎ吸収の原理

照合時の手順 3) における登録データ $\vec{T}$ と照合データ $\vec{Q}$ の揺らぎ吸収の原理を簡単に説明しておく．登録秘匿データ $\vec{H}$ と照合秘匿データ $\vec{H}'$ の差分ベクトル $\vec{z}$ は，

$$\vec{z} = \Big(\vec{T} - \vec{Q}, K, 0\Big) + \underbrace{r\vec{b}_1 - r'\vec{b}_2}_{\text{格子 } L \text{ の元}} \in \mathbb{R}^{n+2}$$

となることが分かる．次に，mod$V$ 写像の性質 (i) から，

$$\vec{z} \bmod V = \Big(\vec{T} - \vec{Q}, K, 0\Big) \bmod V$$

が成り立つ（右辺にも mod$V$ が必要なことに注意）．ここで，ベクトル $(\vec{T} - \vec{Q}, K, 0)$ が格子 L の基本領域 $P = \mathcal{P}(V)$ に含まれている場合（鍵 K もある程度小さい値を利用），mod$V$ 写像の性質 (ii) から，

$$\vec{z} \bmod V = \Big(\vec{T} - \vec{Q}, K, 0\Big) \in \mathbb{R}^{n+2}$$

となり，このとき最終成分が 0 により認証判定できると共に，第 $(n+1)$-成分にユーザ固有の鍵 K が抽出可能となる仕組みである（前述したように，最終成分の個数を増やせば認証判定の成功確率を増大させることが可能）．つまり，本方式では，2 つのデータの揺らぎ吸収可能範囲を，格子の基本領域 P の大きさでコントロールすることが可能である．つまり，基底行列 $V = (\vec{v}_1, \ldots, \vec{v}_{n+2})^T$ の選び方のみでコントロールが可能である．

3.3.3 提案方式の特長

ここでは，提案方式の特長をいくつか挙げておく：

(1) 提案方式においては，システム間で異なる格子を使えば，秘匿テンプレート同士はクロスマッチングできない．具体的には，同じテンプレート $\vec{T}$ と鍵 $\vec{K}$ から作られた以下の 2 つの秘匿テンプレート

$$\vec{H}_1 = (\vec{T}, \vec{K}, \vec{0}) + \vec{q}_1 \text{ および } \vec{H}_2 = (\vec{T}, \vec{K}, \vec{0}) + \vec{q}_2$$

が，異なる格子点 $\vec{q}_1 \in L_1$ と $\vec{q}_2 \in L_2$ でマスクされているとする．このとき，2 つの秘匿テンプレートの差分 $\vec{H}_1 - \vec{H}_2 = \vec{q}_1 - \vec{q}_2$ は $\mathbb{R}^{n+2}$ 上の乱数のように分布するため，攻撃者は $\vec{q}_1$ も $\vec{q}_2$ も入手することはできない．したがって，提案方式は (a) Diversity を満たしている．さらに，もし秘匿テンプレートが漏えいした場合，システムでは格子を変更することで漏えいしたテンプレートを無効化し，新しい秘匿テンプレートを元のテンプレートと新しい格子から作ることができる．これは提案方式が (b) Revocability を満たしていることを意味する．

(2) Fuzzy vault では生体特徴を一様ランダムな chaff 集合の中に隠すが，生体特徴は一様に分布しないため統計解析により生体特徴と chaff を見分ける攻撃に弱い．これに対し，提案方式は生体特徴を直接ランダムな要素に変換するため，提案方式における秘匿テンプレートは生体特徴の分布によらず一様性をもつ．したがって，提案方式は統計解析による攻撃に強い．

(3) Fuzzy vault では，正規ユーザの秘匿テンプレートの chaff の一部を攻撃者の生体特徴に置き換えることで，正規ユーザと攻撃者の両方が認証に成功するような秘匿テンプレートを作ること（blended substitution attack）ができ，これは正規ユーザが気づかないまま悪意を持つものがログインできる状態となる（すなわち他人受入率の増大）ことから (d) Performance の面における脆弱性である．一方提案方式では，秘匿テンプレートの作り方 (1) から，複数のユーザが認証に成功するような秘匿テンプレートを作ることはできず，(d) Performance においてより優れている．

(4) Fuzzy commitment や fuzzy vault の既存実現方式とは異なり，提案方式では，照合データそのものではなく秘匿照合データをサーバに送信するだけで照合処理が可能である．加えて，システムは秘匿登録情報 $\vec{H}$ や秘匿照合情報 $\vec{H}'$ からは，生の照合データ $\vec{Q}$ も生の登録データ $\vec{T}$ も復元することはできない．よって，§3.3.1 に示した認証手順の中でこれら生のデータを悪意を持つものに盗み見られるおそれがない．

4. テンプレート更新方法

4.1 更新方法とその原理

生体認証システムを長期にわたり安定運用するためには，テンプレートの更新機能が重要である．テンプレートの更新を単に登録手順の繰り返しにより行うことは可能だが，より簡便な更新方法，たとえば照合時に自動的に更新する方法を備えることが望ましい．テンプレート保護型でない生体認証システムにおいては，照合時にサーバに送られた生体データを新たなテンプレートとして登録すること

が可能であるが，テンプレート保護型のシステムでは更新方法に工夫が必要である．以下では，格子マスキングによる key binding 技術において，秘匿されていない生体情報をサーバに晒すことなく，照合時にサーバに送られる秘匿照合情報のみから更新テンプレートを作成する方法について述べる．

提案するテンプレート更新方法の原理を以下に説明する．認証成功した場合，サーバは $\vec{H}' = (\vec{Q}, 0, 0) + r'\vec{b}_2$ および鍵 K を持っている．鍵 K を用いて，$(n+2)-$ 次元の更新用鍵ベクトル $(\vec{0}, K, 0)$ を生成する．ここで $\vec{0}$ は n 次元（生体情報と同じデータ構造）の零ベクトルである．ここから次の数式により更新秘匿登録情報 $\vec{H}_{\rm new}$ を生成する．

$$\vec{H}_{\rm new} = \Big(\vec{0}, K, 0\Big) + \vec{H}' \tag{2}$$

これを変形すると以下のようになる．

$$\begin{aligned}\vec{H}_{\rm new} &= \Big(\vec{0}, K, 0\Big) + \Big\{\Big(\vec{Q}, 0, 0\Big) + r'\vec{b}_2\Big\} \\ &= \Big(\vec{Q}, K, 0\Big) + r'\vec{b}_2 \end{aligned} \tag{3}$$

$\vec{Q}$ が（照合時に取得した）生体情報であり，r' が整数乱数であり，$\vec{b}_2$ が格子元であることに注意して，秘匿登録情報 $\vec{H}$ の式と (3) を比較すると，$\vec{H}_{\rm new}$ は照合生体情報 $\vec{Q}$ と鍵 K から作られた秘匿登録情報であることがわかる．よって，$\vec{H}_{\rm new}$ を $\vec{H}$ に代えてデータベースに登録することで，テンプレートの更新が行える．

テンプレート更新手順は照合成功後に行う手順で，以下の通りである．

1) サーバは照合により抽出した鍵 K を用いて，最初に n 次元の零ベクトルおよび最後に 0 を付け加えて拡張した更新用鍵ベクトル $(\vec{0}, K, 0)$ を生成する．

2) サーバは照合時に受け取った秘匿情報 $\vec{H}'$ を用いて

   $$\vec{H}_{\rm new} = (\vec{0}, K, 0) + \vec{H}'$$

   を生成する．

3) もともと登録されている $\vec{H}$ に代えて，サーバのデータベースに $\vec{H}_{\rm new}$ を登録する．

このテンプレート更新手順において，新しい登録データ $\vec{Q}$ は常に格子元 $r'\vec{b}_2$ との和としてのみ扱われるため，生の $\vec{Q}$ がサーバ（ひいてはサーバ内のデータを盗み見ることができる攻撃者）に晒されることはない．また古い登録データ $\vec{T}$ についても，テンプレート更新手順内にこれを復元する操作は含まれないし，また原理的にサーバは $\vec{T}$ を復元できない．

更新秘匿登録情報の生成および置き換えは，照合のたびに行う必要はなく，置き換えの必要が生じた場合に行えばよい．生体認証システムの実装においては，照合を行って本人を確認したのち本手順によって置き換えを行う再登録手順を手動で起動できるようにしてもよいし，照合時に過去の認証からの経過時間や過去の認証結果履歴などから，更新を行うか自動的に判別するようにしてもよい．

更新秘匿登録情報は，上記のように既存の登録情報と置き換えるほかに，既存の登録情報に追加して複数の登録情報を持つようにすることも可能である．このようにすると，生体情報が季節により変動する場合などに各季節に応じた登録情報を持っておくことができる．また，更新時の生体情報の品質が高くなかった場合に，既存の登録情報を置き換えてしまい認証精度が低下する事態も防ぐことができる．

4.2 具体的な適用例と数値例

ここでは，§3.3 で述べた手法を，具体的なシステムへ適用した例について述べる．特に，データの揺らぎを吸収したい範囲を基底行列 V のパラメータに反映する方法を例示し，登録・照合時に行う計算の例を数値により示す．詳細は [13] を参照のこと．

4.2.1 システムの概要

試作したシステムでは，鍵として 8 文字以下の ASCII 文字列，生体情報としてアプリで指定した領域内の 3 点に接触する方法（接触位置，および接触時間）を用いた．システム内部では，鍵 K は各次元が ASCII 文字一文字を表す 8 次元のベクトルとして記録している．また登録時の平文データは，触れた点の (x, y) 座標および触れている時間 t を 3 点それぞれについて取得した，$n = 3 \times 3 = 9$ 次元のベクトルとして記録している．登録・照合間の距離判定に関しては，登録時の x, y, t の各次元に対し，照合時の値の揺らぎが以下の範囲内であればその次元について一致と判定した．

- x, y: ± 32（単位はピクセル）
- t: ± 200（単位はミリ秒）

提案方式では，すべての次元について一致した場合に照合全体が成功となり，登録された鍵を出力する．失敗した場合にも，どの次元が一致しなかったのか等はサーバにもわからないため，登録情報の秘密は守られる．

4.2.2 基底行列 V のパラメータ設定

登録時の平文データ $\vec{T}$ を n 次元ベクトル，鍵 K を k 次元ベクトルとする．また，$\vec{T}$ の各次元について揺らぎ吸収可能範囲を $\pm\sigma_i$, K の各次元について取り得る値の範囲を 0 以上 ρ_j 以下とする．

基底行列 V（$(n+k+1)$-次元正方行列）を以下のように作成することで，指定した揺らぎ吸収可能範囲に従った照合が実現できる．

(1) 各 $1 \le i \le n$ に対して，閾値 σ_i より小さい適当な乱数の組 $(s_{i1}, \ldots, s_{ik}, r_i)$ を生成．各 $1 \le j \le k$ に対して，

閾値 ρ_j より小さい適当な乱数 r_{n+j} を生成し，さらに適当な乱数 r_{n+k+1} を生成する．

(2) 以下で定義される $(n+k+1)$-次元正方行列 V を作成する:

$$V = \begin{pmatrix} 2\sigma_1 & \cdots & 0 & s_{11} & \cdots & s_{1k} & r_1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 2\sigma_n & s_{n1} & \cdots & s_{nk} & r_n \\ 0 & \cdots & 0 & 2\rho_1 & \cdots & 0 & r_{n+1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 2\rho_k & r_{n+k} \\ 0 & \cdots & 0 & 0 & \cdots & 0 & r_{n+k+1} \end{pmatrix}$$

4.2.3 数値例

ここでは簡単のため，領域内の 1 点について (x, y, t) を取得し，鍵 K は ASCII 文字 1 文字とする場合の数値例を示す．登録時の x, y, t の各次元に対する揺らぎ吸収範囲は §4.2.1 に示した通りとする．

準備： 次元数 $n = 3$, 揺らぎ吸収範囲から $\sigma_1 = 32$, $\sigma_2 = 32$, $\sigma_3 = 200$ とおく．また，鍵は ASCII 文字 1 文字とするため $k = 1$, $\rho_1 = 256$ とおく．サーバは §4.2.2 の設定方法に従い，必要な部分には乱数を設定して基底行列

$$V = \begin{pmatrix} 64 & 0 & 0 & 7 & 5 \\ 0 & 64 & 0 & 4 & 3 \\ 0 & 0 & 400 & 5 & -2 \\ 0 & 0 & 0 & 512 & -42 \\ 0 & 0 & 0 & 0 & 123 \end{pmatrix}$$

を生成し，データベースに登録する．

登録時：

1) まず $\vec{v}_1 = (64, 0, 0, 7, 5)$, $\vec{v}_2 = (0, 64, 0, 4, 3)$, $\vec{v}_3 = (0, 0, 400, 5, -2)$, $\vec{v}_4 = (0, 0, 0, 512, -42)$, $\vec{v}_5 = (0, 0, 0, 0, 123)$ に対して，サーバは適当な整数乱数を生成し，乱数格子ベクトル

$$\begin{aligned} \vec{b}_1 &= 2\vec{v}_1 + 3\vec{v}_2 - 5\vec{v}_3 - \vec{v}_4 + 5\vec{v}_5 \\ &= (128, 192, -2000, -511, 686) \end{aligned}$$

を生成し，クライアントに送信する．

2) 鍵が文字"A"（$K = 65$），テンプレートが $\vec{T} = (x, y, t) = (237, 178, 120)$ の場合，クライアントは $(\vec{T}, K, 0) = (237, 178, 120, 65, 0)$ を生成する．次に整数乱数 r（例として $r = 7$）を生成し，格子元 $\vec{b}_1$ を元に秘匿登録情報

$$\begin{aligned} \vec{H} &= (237, 178, 120, 65, 0) + r\vec{b}_1 \\ &= (1133, 1522, -13880, -3512, 4802) \end{aligned}$$

を生成し，サーバのデータベースに登録しておく．

照合時：

1) 登録時と同様，サーバは適当な整数乱数を生成し，乱数格子ベクトル

$$\begin{aligned} \vec{b}_2 &= 5\vec{v}_1 - 2\vec{v}_2 + 7\vec{v}_3 + \vec{v}_5 \\ &= (320, -128, 2800, 62, 128) \end{aligned}$$

を生成し，クライアントに送信する．

2) 照合データ $\vec{Q}_1 = (250, 190, 100)$（$\vec{T}$ との差が揺らぎ吸収範囲内の例）に対し，クライアントは整数乱数 $r' = 123$ を生成し，秘匿照合情報

$$\begin{aligned} \vec{H}_1 &= (250, 190, 100, 0, 0) + r'\vec{b}_2 \\ &= (39610, -15554, 344500, 7626, 15744) \end{aligned}$$

を生成し，サーバに送信する．
もし照合データが $\vec{Q}_2 = (250, 250, 100)$（$\vec{T}$ との差が揺らぎ吸収範囲外の例）であったとすると，秘匿照合情報

$$\begin{aligned} \vec{H}_2 &= (250, 250, 100, 0, 0) + r'\vec{b}_2 \\ &= (39610, -15494, 344500, 7626, 15744) \end{aligned}$$

を生成する．

3) サーバは，秘匿照合情報 $\vec{H}_1$ に対して以下の計算を行う：

$$\begin{aligned} \vec{d}_1 &= \vec{H} - \vec{H}_1 \\ &= (-38477, 17076, -358380, -11138, -10942), \\ \vec{z}_1 &= \vec{d}_1 \bmod V \\ &= (-13, -12, 20, 65, 0). \end{aligned}$$

$\vec{z}_1$ の最終成分が 0 であるので照合成功と判定し，第 4 成分から鍵 $K = 65$（登録した文字"A"）をクライアントに送信する．
もし秘匿照合情報 $\vec{H}_2$ が送られてきた場合，

$$\begin{aligned} \vec{d}_2 &= \vec{H} - \vec{H}_2 \\ &= (-38477, 17016, -358380, -11138, -10942) \\ \vec{z}_2 &= \vec{d}_2 \bmod V \\ &= (-13, -8, 20, 69, 3) \end{aligned}$$

より $\vec{z}_2$ の最終成分が 0 でないので照合失敗と判定し，第 4 成分から鍵 $K = 69$（登録したのと異なる文字"E"）をクライアントに送信する．

更新時： テンプレート更新が行えるのは照合成功の場合のみである．すなわちこの例においては秘匿照合情報 $\vec{H}_1$ が送られた場合にあたる．

1) 照合成功後，サーバはまず鍵 $K = 65$ を元に更新用鍵ベクトル $(\vec{0}, K, 0) = (0, 0, 0, 65, 0)$ を作成する．

2) 更新用鍵ベクトル $(\vec{0}, K, 0)$ と秘匿照合情報 $\vec{H}_1$ から，

サーバは以下の計算により更新秘匿登録情報 $\vec{H}_{\mathrm{new}}$ を生成する．

$$\begin{aligned}\vec{H}_{\mathrm{new}} &= (\vec{0}, K, 0) + \vec{H}_1 \\ &= (39610, -15554, 344500, 7691, 15744)\end{aligned}$$

$\vec{H}_{\mathrm{new}} = (\vec{Q}_1, K, 0) + r'\vec{b}_2$ であり，すなわち $\vec{H}_{\mathrm{new}}$ は鍵 $K = 65$ と照合データ $\vec{Q}_1 = (250, 190, 100)$ を格子元 $r'\vec{b}_2$ でマスクした新たな秘匿テンプレートである．

3) もともと登録されている $\vec{H}$ に代えて，サーバのデータベースに $\vec{H}_{\mathrm{new}}$ を登録する．

5. まとめと今後の課題

今回我々は，格子マスキング利用の key binding 技術 [13] において，テンプレートの更新処理を行う方法を提案した．提案手法は，認証成功時に受け取った秘匿生体データと抽出した鍵を足し合わせることで，更新用のテンプレートを作成するものである．この方法により，格子マスキング利用の key binding 技術においてもテンプレートの更新を行うことができ，認証精度の低下を防ぐことができる．また認証成功していなければ鍵が取り出せず更新不可能であるため，悪意ある者によってテンプレート更新が引き起こされることのない方法である．さらに作成した更新テンプレートを元のテンプレートと両立することで，季節による生体情報変動に対応すること，品質の低い生体データを用いて更新したことによる認証精度低下を防ぎつつ認証精度の低下に対応することができるようになった．

今後の課題としては，本論文で取り上げたテンプレートの更新処理を含む格子マスキング利用の key binding 技術全体について，指紋認証や手のひら静脈認証など実用的な生体認証技術への適用を検討することを考えている．

参考文献

[1] Clancy, T. C., Kiyavash, N. and Lin, D. J.: Secure smartcardbased fingerprint authentication, Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications, ACM, pp. 45–52 (2003).

[2] Draper, S. C., Khisti, A., Martinian, E., Vetro, A. and Yedidia, J. S.: Using distributed source coding to secure fingerprint biometrics, IEEE International Conference on Acoustics, Speech and Signal Processing–ICASSP 2007, Vol. 2, IEEE, pp. 129–132 (2007).

[3] Feng, Y. C. and Yuen, P. C.: Protecting face biometric data on smartcard with reed-solomon code, IEEE International Conference on Computer Vision and Pattern Recognition Workshop–CVPRW'06, IEEE, p. 29 (2006).

[4] Freire-Santos, M., Fierrez-Aguilar, J. and Ortega-Garcia, J.: Cryptographic key generation using handwritten signature, Biometric Technology for Human Identification III, Vol. 6202, pp. 225–231 (online), DOI: 10.1117/12.665875 (2006).

[5] Jain, A. K., Nandakumar, K. and Nagar, A.: Biometric Template Security, EURASIP Journal on Advances in Signal Processing, Vol. 2008, pp. 113:1–113:17 (online), DOI: 10.1155/2008/579416 (2008).

[6] Juels, A. and Sudan, M.: A fuzzy vault scheme, Designs, Codes and Cryptography, Vol. 38, No. 2, pp. 237–257 (2006 (a preliminary version was presented at ISIT 2002)).

[7] Juels, A. and Wattenberg, M.: A fuzzy commitment scheme, Proceedings of the 6th ACM conference on Computer and communications security–ACM CCS'99, ACM, pp. 28–36 (1999).

[8] Lee, Y. J., Bae, K., Lee, S. J., Park, K. R. and Kim, J.: Biometric key binding: Fuzzy vault based on iris images, Advances in Biometrics, Springer, pp. 800–808 (2007).

[9] Tuyls, P., Akkermans, A. H., Kevenaar, T. A., Schrijen, G.-J., Bazen, A. M. and Veldhuis, R. N.: Practical biometric authentication with template protection, Audio- and Video-Based Biometric Person Authentication–AVBPA'05, Lecture Notes in Computer Science, Vol. 3546, Springer, pp. 436–446 (2005).

[10] 佐久間淳，小林重信：プライバシ保護データマイニング，人工知能学会誌， Vol. 24, No. 2, pp. 283–294 (2009).

[11] 佐久間淳，高橋克巳：クラウドストレージにおける個人情報の利活用とプライバシー保護 (特集 クラウドを支えるデータストレージ技術)，情報処理， Vol. 52, No. 6, pp. 706–715 (2011).

[12] 清水將吾，瀬戸洋一：国際標準化に向けたテンプレート保護技術の体系化，産業技術大学院大学紀要，Vol. 1, pp. 93–104 (2007).

[13] 杉村由花，安田雅哉，山田茂史，安部登樹，新崎　卓：格子マスキングを利用した Key Binding 技術の提案，電子情報通信学会技術研究報告， Vol. 113, No. 135, pp. 297–304 (2013).