Vol.2021-DPS-188 No.24
Vol.2021-EIP-93 No.24
2021/9/10

照合可能暗号をもとにしたIoT環境に適した指紋認証システムに関する一考察

木原　眞紀[1,a)]　小野寺　智春[2,b)]　入山　聖史[1,c)]

概要：IT 技術の発展に伴い，指紋を自動で照合する技術が進展してきた．身近な例として，スマートフォンなどの小型デバイスのロック解除や入退出管理などが挙げられる．リモートな環境下で指紋認証を行う場合，通信時における盗聴や，認証サーバ管理者の不正・事故による漏洩が危惧される．本研究では，情報を暗号化したまま処理し復号せずに照合する指紋認証システムを構築する．このシステムでは，攻撃された場合においても情報を復号する鍵はどこにも送信されることがないため，個人情報が漏洩する心配がない．速度の面においても，暗号化の有無による照合時間の差は 1ms と非常に高速な結果を得た．

キーワード：指紋認証，認証，暗号，情報セキュリティ，Internet of Things

1. はじめに

IoT(Internet of Things) 技術の発展に伴い，組織・個人による利活用が増えていることから IoT 機器の台数も年々増加傾向にある．実際，総務省の令和 2 年度版情報通信白書 [1] によれば，2022 年には 340 億台を超える予測がなされている．一方で，IoT 機器がもつ脆弱性も指摘されており，その中でも特に認証時における攻撃として ID，パスワードなどの認証情報に対する攻撃が問題視されている [2]．しかしながら，2019 年 2 月にインターネットサービスを提供している事業者を対象に実施された認証方法に関するアンケートでは，77% の事業者が ID とパスワードのみの認証を採用している [3]．

これに対し ID とパスワードによる個人認証に代わる認証方法として，指紋や顔，静脈をはじめとする生体情報を利用した認証方法（生体認証）がスマートフォンをはじめとする IoT 機器でも多く見受けられるようになってきている．生体認証は本人しか持ち得ない生体情報を利用し本人であることを証明するので，他人の ID とパスワードを利用してなり済ますことができるパスワード認証よりも安全性が高いとされている [4]．また，身体情報を用いることから認証のために必要な物を持ち歩く必要や情報を記憶する必要がなく，紛失や忘失のリスクがなく利便性が高い．こういった点からもスマートフォンや計算機などのデバイスのロック解除をはじめとするユースケースが増加してきている．

生体認証の手法やシステムは多く開発されており，指紋認証は最も一般的な方法の 1 つである [5]．複数の隆線により構成されている指紋の情報は　生涯不変・万人不同と言われ，たとえ一卵性双生児であっても異なる指紋を持っている．指紋認証は，犯罪捜査にも用いられるほど古い歴史のある認証の有効的な手段であると考えられてきた [6]．しかし，生涯不変な情報であるが故に，指紋をはじめとする生体情報を盗まれた場合，パスワードのように変更処理ができない．そのため，指紋などの生体認証に用いる生体情報は慎重に管理する必要があり，指紋情報を保護する研究は広く行われてきた [7], [8], [9]．

計算資源の少ない IoT 機器などの小型デバイスのロック解除といったネットワークを介して行う認証の場合では，安全かつ軽量な認証アルゴリズムが求められる．特に生体認証で用いる生体情報は多次元であり処理に多くの計算資源を必要とするため，ゼロ知識証明をもとにした認証アルゴリズム [10], [11] などの計算量の多い認証アルゴリズムを用いると認証にかかるコストが大きくなってしまう．そこで本稿では 2019 年に提案した木原，入山は IoT 環境に適した認証アルゴリズム [12] を生体認証に適用し，システム構築を試みる．この認証アルゴリズムは，暗号化したまま 2 つの平文の距離を導出できるをもつ照合可能暗号と呼ばれる暗号系クラスをもとにしており，ワンタイムパッド

[1] 東京理科大学 理工学部 情報科学科
〒 278-8510　千葉県野田市山崎 2641
[2] 東京理科大学 理工学研究科 情報科学専攻
〒 278-8510　千葉県野田市山崎 2641
[a)] mkihara@rs.tus.ac.jp
[b)] 6321517@ed.tus.ac.jp
[c)] iriyama@is.noda.tus.ac.jp

を用いた実装では 2048bit のテキスト長の場合においても 0.1ms 以下という非常に高速な結果が得られている．また安全性においても [12], [13] で理論的に安全であることが証明されており，さらに Bruno Blanchet によって開発された暗号化プロトコルに対するセキュリティプロパティ自動検証ツール ProVerif[14] を用いた解析でも安全であることがわかっている．

本稿では，2 章で暗号系・照合可能暗号について定義し，照合可能暗号をもとにしたアルゴリズムを紹介した後に，照合可能暗号をもとにした指紋認証システムを提案する．さらに，3 章で精度・速度の検証を行い，4 章で今後の課題・展望について述べる．

2. 照合可能暗号をもとにした生体認証システム

本章では，数学的準備として暗号系・照合可能暗号・照合可能暗号をもとにした認証アルゴリズムを導入し，照合可能暗号をもとにした生体認証システムについて提案する．

2.1 準備

まず暗号系について定義する．さらに照合可能暗号について定義し，照合可能暗号をもとにした認証アルゴリズムを紹介する．照合可能暗号および照合可能暗号をもとにした認証アルゴリズムは 2019 年に木原，入山により提案されている [12].

2.1.1 暗号系

まず，暗号系について定義する．

定義 1 次の性質を有する 5 つ組 $(\mathcal{P},\mathcal{C},\mathcal{K},\mathcal{E},\mathcal{D})$ を暗号系または暗号化方式と定める:

(1) $\mathcal{P},\mathcal{C},\mathcal{K}$ は集合であり，それぞれ平文空間，暗号文空間，鍵空間という．また，その元をそれぞれ平文，暗号文，鍵という．
(2) $\mathcal{E}=\{E_k; k\in\mathcal{K}\}$ は関数 $E_k:\mathcal{P}\to\mathcal{C}$ の族であり，その元を暗号化関数という．
(3) $\mathcal{D}=\{D_k; k\in\mathcal{K}\}$ は関数 $D_k:\mathcal{C}\to\mathcal{P}$ の族であり，その元を復号関数という．
(4) 任意の平文 $p\in\mathcal{P}$ に対し，$D_k(E_k(p))=p$ が成り立つような鍵 $k\in\mathcal{K}$ が存在する．

本稿では，5 つ組 $(\mathcal{P},\mathcal{C},\mathcal{K},\mathcal{E},\mathcal{D})$ を暗号系と呼ぶ．

暗号系の例として，1917 年に Girbert Vernam によって提案され，1949 年に Claude Shannon によって完全守秘性をもつことが証明されているワンタイムパッド（一回使い捨て暗号）を紹介する．

例 2 ワンタイムパッド暗号系 $C_{otp}=(\mathcal{P},\mathcal{C},\mathcal{K},\mathcal{E},\mathcal{D})$ は次から構成される:

- $\mathcal{P}=\mathcal{C}=\mathcal{K}=\{0,1\}^n$
- $\mathcal{E}=\{E_k; E_k(p)=p\oplus k, k\in\mathcal{K}\}$
- $\mathcal{D}=\{D_k; D_k(c)=c\oplus k, k\in\mathcal{K}\}$

ここで，任意の鍵 $k\in\mathcal{K}$ は一様分布に従い，ランダムに選ばれる．すなわち，平文 $p\in\mathcal{P}=\{0,1\}^n$ を暗号化する場合，鍵 $k\in\mathcal{K}=\{0,1\}^n$ は，$\frac{1}{2^n}$ の確率で選ばれる．また，$\oplus$ は排他的論理和を表す．

2.1.2 照合可能暗号

次に，照合可能暗号について定義する．照合可能暗号とは，暗号文同士の計算を可能とし，かつ結果が 2 つの平文の距離と一致するような暗号系クラスである．上述の例 2 は照合可能暗号クラスに属する暗号系の 1 つである．

定義 3 距離 $V:\mathcal{P}\times\mathcal{P}\to\mathbb{R}_+=[0,+\infty)$ と 2 つの暗号系 $C_1=(\mathcal{P},\mathcal{C},\mathcal{K},\mathcal{E},\mathcal{D})$，$C_2=(\mathcal{P},\mathcal{C},\mathcal{K}',\mathcal{E}',\mathcal{D}')$ が与えられたとき，任意の 2 つの平文 $p_1,p_2\in\mathcal{P}$ に対し，

$$D_{k,k'}(F(E_k(p_1),E'_{k'}(p_2)))=V(p_1,p_2)$$

を満たす 2 つの写像 $F:\mathcal{C}\times\mathcal{C}\to\mathcal{C}, D:\mathcal{C}\to\mathbb{R}_+$ と鍵セット $(k,k')\in\mathcal{K}\times\mathcal{K}'$ が存在するとき，暗号化関数 $(\mathcal{E},\mathcal{E}')$ を照合可能暗号という．ここで，$E\in\mathcal{E}, E'\in\mathcal{E}'$ である．

この条件式は，関数 F によって 2 つの暗号文 $E_k(p_1),E'_{k'}(p_2)$ に操作を施し，その結果に対して関数 D を作用させることで求めたい距離 $V(p_1,p_2)$ と一致することを意味している．平文 p_1,p_2 はそれぞれ異なる鍵 k,k' で暗号化されているため，安全でないチャネルから 2 つの暗号文を攻撃者が取得したとしても，攻撃者が平文を得ることは困難である．暗号化関数 E,E' の計算と関数 F の計算は同一の場所で行う必要がなく，$F(E_k(p_1),E'_{k'}(p_2))$ も暗号文（暗号化された照合結果）であるため，関数 F の計算をクラウド化し照合結果である距離 $V(p_1,p_2)$ の秘匿計算が可能となる．

2.1.3 照合可能暗号をもとにした認証アルゴリズム

ここでは，IoT デバイスなどのネットワークを介した小型デバイスのロック解除に適した，照合可能暗号をもとにした認証アルゴリズムを紹介する．まず，認証は登録ステップと照合ステップから構成され，アリスを認証される人物，ボブを認証者（認証する人物）とし，アリスはボブを信頼できるパーティであると考える．また実装を考え，サーバーをクラウドサーバーなどの信頼できないサードパーティと仮定する．次に，$C_1=(\mathcal{P},\mathcal{C},\mathcal{K},\mathcal{E},\mathcal{D}), C_2=(\mathcal{P},\mathcal{C},\mathcal{K}',\mathcal{E}',\mathcal{D}')$ を 2 つの暗号系とし，$(\mathcal{E},\mathcal{E}')$ を照合可能暗号とする．また，$p_1,p_2\in\mathcal{P}$ を平文，$(k,k')\in\mathcal{K}\times\mathcal{K}'$ を鍵セット，$c_1=E_k(p_1), c_2=E'_{k'}(p_2)\in\mathcal{C}$ を暗号文とする．

登録ステップ

(1) アリスは p_1 をボブに送信する．
(2) ボブは鍵 k を生成し，$c_1=E_k(p_1)$ を計算する．
(3) ボブは c_1 をサーバーに送信し，サーバーは c_1 をデータベースに格納する．

照合ステップ

(1) アリスは p_2 をボブに送信する．
(2) ボブは鍵 k' を生成し，$c_2=E'_{k'}(p_2)$ を計算する．

(3) ボブは c_2 をサーバーに送信する．
(4) サーバーは $F(c_1, c_2) = c_d$ の計算を行い，c_d をボブへ送る．
(5) ボブは登録時に生成した鍵 k と照合時に新たに生成した k' を用いて $r = D_{k,k'}(F(c_d))$ の計算を行い照合結果を確認する．

このアルゴリズムでは，照合ステップごとにワンタイムキー生成して認証を行い，暗号化に使用した鍵 k, k' はどこにも送信されることはないので，サーバに送信される c_2 から攻撃者がアリスの個人情報を得ることは困難である．さらに，一度暗号文を復号し平文に戻してから照合する必要がないので，アリスの個人情報をサーバーが得ることはない．また，認証のためにゼロ知識証明のようなメッセージのやりとりや計算量の多い鍵配送が必要ないので，計算資源の少ない IoT 環境にも適していると言える．このアルゴリズムにワンタイムパッドを適用して行なった実装では，テキスト長を 128bit から 8192bit まで変えたどの場合においても，暗号化・照合・復号にかかる速度はどれも 0.1ms 以下という高速な結果が得られている．安全性・速度などの詳細な性能評価については [12] を参照されたい．

2.2 システム概要

本システムはユーザー，登録機，照合機，サーバーで構成され，ユーザーをアリス，ボブを登録機/照合機と考えることで 2.1.3 と対応したアルゴリズムとなる．また，2.1.3 で紹介したアルゴリズムと同様に，ユーザーは登録機/照合機を信頼し，サーバーはクラウドサーバーなどの信頼できないものとする．すなわち，図 1 に示すように，ユーザーと登録機/照合機間のチャネルは安全なものであるとし，登録機/照合機とサーバー間は安全でないものとする．

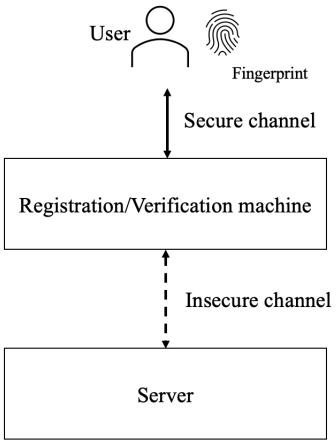

図 1 各パーティ間のチャネル．
Fig. 1 Channels between each party.

また，登録機と照合機は同一の事業者が提供するものとし，登録機と照合機間のチャネルは安全なものとする．表記と実装のために登録機と照合機を分けたシステムを記述するが，登録機と照合機は同一デバイスでも構わない．

本システムは登録ステップと照合ステップに分かれ，両ステップともに「指紋画像からバイナリ列への変換ステップ」が各ステップのの最初に組み込まれる．また，利用する暗号系としてワンタイムパッドを採用する．ここでは，2 章 3 節で指紋画像をバイナリ列へ変換する手法について説明し，4 節で認証システムの全体の流れを説明する．本稿では生体情報として指紋を利用することとするが，本システムは指紋画像からバイナリ列への変換 (2 章 3 節) をそのほかの生体情報からバイナリ列への変換ステップに置き換えることで，指紋以外の生体認証にも応用可能なシステムである．

2.3 指紋画像の変換

ここでは，特定の指紋画像をバイナリ列に変換する方法について説明する．$I(i,j)$ を高さ H ピクセル幅 W ピクセルの 2 次元 256 レベルのグレースケールの指紋画像とし，$I(i,j)$ を $a \times a$ ピクセルごとに分割したブロックを $B_{x,y}$ とする．この手法は，次の 3 つのステップで構成される：

(1) 指紋画像の隆線方向の推定 [15]
(2) 特異点の位置の特定 [15], [16]
(3) 点周辺の隆線方向からバイナリ列の生成

2.3.1 隆線方向の推定

各ブロックの隆線方向 $\theta_{x,y}$ を推定するために，ブロック $B_{x,y}$ の異方性 $\theta'_{x,y}$ を考える．
計算に用いられる Sobel フィルター Sx, Sy[17] は次の通りである：

$$Sx = (sx_{i,j}) = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

$$Sy = (sy_{i,j}) = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

各 $i, j (0 \le i \le H, 0 \le j \le W)$ に対し，$I(i,j)$ の勾配 $Gx_{i,j}, Gy_{i,j}$ を以下のように計算する：

$$Gx_{i,j} = \sum_{m=-1}^{1} \sum_{n=-1}^{1} I(i+n, j+m) sx_{m+1,n+1}$$

$$Gy_{i,j} = \sum_{m=-1}^{1} \sum_{n=-1}^{1} I(i+n, j+m) sy_{m+1,n+1}$$

ここで，$sx_{i,j}, sy_{i,j}$ はそれぞれ Sx, Sy の i, j 成分である．

次に $\theta'_{x,y}$ を求めていくが，$B_{x,y}$ の平均勾配 $\overline{Gx_{x,y}}, \overline{Gy_{x,y}}$ を用いて $\theta'_{x,y}$ の計算を行うとブロック内の勾配は互いに打ち消し合ってしまう．そのため，代わりに 2 倍角 $2\theta'_{x,y}$，勾配の 2 乗平均 $\overline{Gx^2_{x,y}}, \overline{Gy^2_{x,y}}$，勾配の積 $\overline{Gx_{x,y}Gy_{x,y}}$ をを使用

して $\theta'_{x,y}$ の導出を行う．ここで，$\overline{Gx^2_{x,y}}, \overline{Gy^2_{x,y}}, \overline{Gx_{x,y}Gy_{x,y}}$ はそれぞれ以下の通りである:

$$\overline{Gx^2_{x,y}} = \sum_{i,j\in B_{x,y}} Gx^2_{i,j}$$
$$\overline{Gy^2_{x,y}} = \sum_{i,j\in B_{x,y}} Gy^2_{i,j}$$
$$\overline{Gx_{x,y}Gy_{x,y}} = \sum_{i,j\in B_{x,y}} Gx_{i,j}Gy_{i,j}$$

よって，$\tan 2\theta'_{x,y}$ は以下のように求められる．

$$\begin{aligned}\tan 2\theta'_{x,y} &= \frac{2\tan\theta'_{x,y}}{1-\tan^2\theta'_{x,y}} \\ &= \frac{\frac{2\overline{Gy_{x,y}}}{\overline{Gx_{x,y}}}}{1-\frac{\overline{Gy^2_{x,y}}}{\overline{Gx^2_{x,y}}}} \\ &= \frac{2\overline{Gx_{x,y}Gy_{x,y}}}{\overline{Gx^2_{x,y}}-\overline{Gy^2_{x,y}}}\end{aligned}$$

したがって，$\theta'_{x,y}, \theta_{x,y}$ は以下のように求められる;

$$\theta'_{x,y} = \frac{1}{2}\arctan\frac{2\overline{Gx_{x,y}Gy_{x,y}}}{\overline{Gx^2_{x,y}}-\overline{Gy^2_{x,y}}}$$
$$\theta_{x,y} = \theta'_{x,y} + \frac{\pi}{2}$$

2.3.2 特異点の位置の特定

すべての隆線方向に基づいてバイナリ列を生成する場合，特定の画像内の指紋の位置が固定されていないため同じ指に対して異なるバイナリ列を生成する可能性がある．この問題を解決するために指紋の参照点 (c_x, c_y) を決定し，その点を中心とする特定の範囲内の隆線方向を使用する．指紋は「ループ」または「ホール」をはじめとする特徴的な形状がいくつか存在する．一般にこれらは特異点と呼ばれ，指紋照合の手がかりとして使用される．本手法では，特異点が存在する位置をバイナリ列を生成するための参照点として使用する．

ブロック $B_{x,y}$ が特異点であるかどうかを判断するため，Kawagoe, Tojo によって提案されたポアンカレ指数法 [15] を用いる．$B_{x,y}$ のポアンカレ指数 $P(x,y)$ は次のように定義される;

$$\begin{aligned}P(x,y) &= \sum_{i=0}^{7}\delta_i \\ &= \begin{cases}\delta'_i+\pi & (\delta' \le -\frac{\pi}{2}) \\ \delta'_i & (|\delta'| < \frac{\pi}{2}) \\ \delta'_i & (\frac{\pi}{2} \le \delta')\end{cases}\end{aligned}$$

ここで，$\delta'_i = \theta_{i \mod 8} - \theta_{i+1 \mod 8}$ である．

ナンバリングは $B_{x,y}$ の左上 $(x-1, y-1)$ から始まり，以下のように時計周りに座標に向かって進んでいく．

$(x-1,y-1)=0$	$(x,y-1)=1$	$(x+1,y-1)=2$
$(x-1,y)=7$	(x,y)	$(x+1,y)=3$
$(x-1,y+1)=6$	$(x,y+1)=5$	$(x+1,y+1)=4$

$B_{x,y}$ は $P(x,y) \approx \pi$ のとき loop，$P(x,y) \approx 2\pi$ のとき whorl と呼ばれる．本提案では，各 $B_{x,y}$ に対して loop，whorl となる点 (x,y) を参照点 (c_x, c_y) として定める．

2.3.3 バイナリ列の生成

最後に，2.3.1 で求めた $\theta_{x,y}$ および 2.3.2 で求めた (c_x, c_y) を用いてバイナリ列を生成する．$t_{x,y} = k(\{k \in \mathbb{N} | 0 \le k \le q\}, \frac{2^k\pi}{2^q} \le \theta_{x,y} < \frac{2^{k+1}\pi}{2^q})$ をビット深度 q でにおける $\theta_{x,y}$ の量子化とし，(c_x, c_y) を中心とした $(2w+1)\times(2w+1)$ ブロックを取得してバイナリ列 p を以下のように生成する：

$$\begin{aligned}p &= p_1 p_2 \cdots p_{(2w+1)^2} \\ &= t_{c_x-w,c_y-w}t_{c_x-w+1,c_y-w} \\ &\quad \cdots t_{c_x+w,c_y-w}t_{c_x-w,c_y-a+w}\cdots t_{c_x+w,c_y+w}\end{aligned}$$

なお，w は任意に選択できるものとする．

2.4 照合可能暗号を用いた指紋認証システム

ここでは，照合可能暗号を用いた指紋に指紋認証システムを提案する．システムは登録ステップおよび照合ステップで構成され，フローはそれぞれの図 2，3 に示す．なお，各ステップのステップ（2）では，2 章 3 節の行程を用いて指紋画像をバイナリ列へ変換する．

2.4.1 登録ステップ

(1) ユーザーは 指紋情報 fp_1 を登録機に送信する．
(2) 登録機は fp_1 を p_1 に変換する．
(3) 登録機は鍵 k を生成し，$c_1 = p_1 \oplus k$ を計算する．
(4) 登録機は c_1 をサーバー，に送信し，k を照合機へ送る．

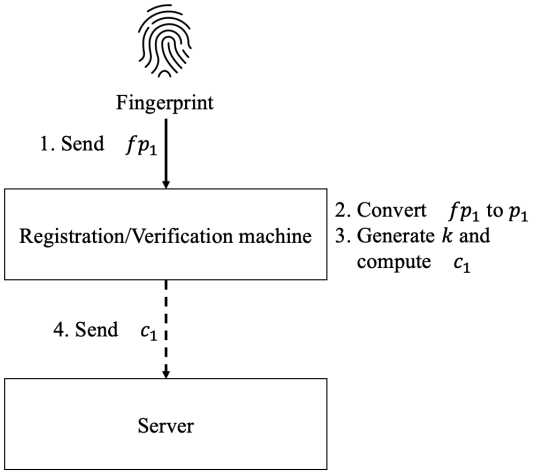

図 2 登録ステップ.
Fig. 2 Registration step.

2.4.2 照合ステップ

(1) ユーザーは 指紋情報 fp_2 を照合機に送信する．
(2) 照合機は fp_2 を p_2 に変換する．
(3) 照合機は鍵 k' を生成し，$c_2 = p_1 \oplus k$ を計算する．
(4) 照合機は c_2 をサーバーに送信する．

(5) サーバーは $c_d = F(c_1, c_2) = c_1 \oplus c_2$ を計算し，c_d を照合機へ送る
(6) 照合機は鍵 k，k' を用いて $r = D_{k,k'}(c_d)) = \sum c_d \oplus k \oplus k'$ の計算を行い，$r \leq s$ を確認する．

ここで，s はしきい値とし，照合結果の計算は

$$\begin{aligned} r &= \sum c_d \oplus k \oplus k' \\ &= \sum c_1 \oplus c_2 \oplus k \oplus k' \\ &= \sum p_1 \oplus k \oplus p_2 \oplus k' \oplus k \oplus k' \\ &= \sum p_1 \oplus p_2 \end{aligned}$$

となる．すなわち，照合結果 r は，登録に使用した指紋情報 fp_1 を変換した個人情報 p_1 と照合時に送られてきた指紋情報 fp_2 を変換した個人情報 p_2 のハミング距離となる．

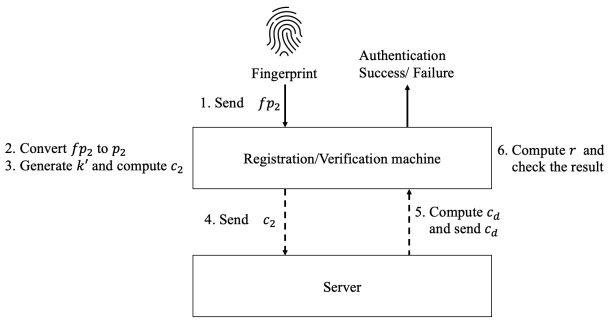

図 3 照合ステップ．
Fig. 3 Verification step.

3. 実験

本章では，提案した認証システムに対し認証にかかる速度・認証精度の面から性能評価を行う．

3.1 実験環境

認証速度の実験の環境は以下の通りである．
OS: Windows 10
CPU: AMD Ryzen 5 5600X
RAM: 16GB
認証精度の実験の環境は以下の通りである．
DEVICE: MacBook Pro
OS: Big Sur
CPU: Intel Corei5, 1.4GHz
RAM: 16GB
認証速度・認証精度の 2 つの実験に共通する実験環境は以下の通りである．
開発言語 Python3.8.6
指紋認証モジュール FPM10A
使用した画像サイズ $(H, W) = (288, 256)$
バイナリ列生成パラメータ w $w = 5$
また，使用した外部ライブラリを以下に示す．

- pyfingerprint [18]: 指紋画像の取得
- fingerprint_recognition [19]: 指紋画像から特徴量の抽出
- OpenCV[20]: fingerprint_recognition の依存ライブラリ

なお，指紋認証モジュールについては指紋画像の取得に使用し，fingerprint_recognition に関しては改変を行っている（MIT ライセンスのため，プログラムの改変は許可されている）．

3.2 認証速度

ここでは，暗号化処理の有無による登録指紋の認証速度の差の比較を行い，指紋を提示してから照合が終了するまでにかかる時間に有意差が生じるかを検証する．生体テンプレートとしては，両手の指 10 本分の指紋画像 $(fp_{1,1}, fp_{1,2}, \cdots, fp_{1,10})$ 及び右人差し指の指紋画像計 5 枚 $(fp_{2,1}, fp_{2,2}, fp_{2,3}, fp_{2,4}, fp_{2,5})$ を使用する．ここで，$fp_{1,i}\,(i = 1, \cdots, 10)$ は登録用の指紋画像とし，2 章 4 節の登録ステップ fp_1 と対応する．また，$fp_{2,j}\,(j = 1, \cdots, 5)$ は照合用の指紋画像とし，2 章 4 節の照合ステップ fp_2 と対応する．実験手順は以下の通りである．
暗号化なし (1)，(2) にかかる時間を計測：
(1) $fp_{1,i}, fp_{2,j}$ をそれぞれ $p_{1,i}, p_{2,j}$ に変換する．
(2) $V(p_{1,i}, p_{2,j})$ を計算する．
暗号化あり (1)-(4) にかかる時間を計測：
(1) $fp_{1,i}, fp_{2,j}$ をそれぞれ $p_{1,i}, p_{2,j}$ に変換する．
(2) $c_{1,i} = E_k(p_{1,i}), c_{2,j} = E'_k(p_{2,j})$ を計算する（鍵生成も含む）．
(3) $c_d = F(c_{1,i}, c_{2,j})$ を計算する．
(4) $D_{k,k'}(c_d)$ を計算する．
暗号化処理あり/なしの実験に対し，テストデータとしてそれぞれ各 $(fp_{1,i}, fp_{2,j})$ を登録/照合のペア 50 組分として測定を行い，平均で評価した結果を図 4 に示す．

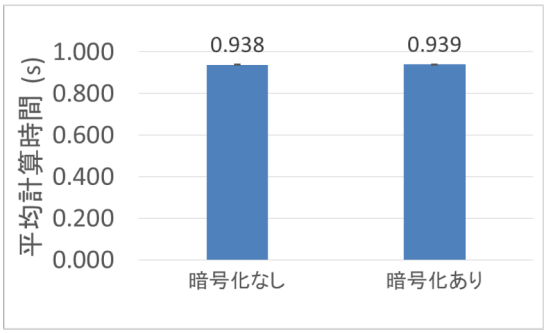

図 4 処理速度の比較．
Fig. 4 Comparison of of processing speed

図 4 のグラフからわかるように暗号化の有無で有意差は生じなかった．

3.3 認証精度

同じ指同士で照合した場合と異なる指同士で照合した場合で距離の差異があるかの測定，閾値を変えた場合に認証率がどのように変化するかを検証する．実験手順は，各ペアについて，距離を計算・記録し，閾値 $s = 200, s = 150$ の場合において照合の可否を検証し，照合率を測定する．2.4.2 にあるように平文間の距離をハミング距離，すなわちバイナリ列 r の 1 の総数を距離として考えるため，閾値 s は，小さくなればなるほど許容する差異が小さくなる．今回の実験で閾値として採用した $s = 200, s = 150$ の場合，$s = 150$ が許容範囲が小さい，

生体テンプレートとしては, 以下を使用して測定を行った．

右人差し指同士　右人差し指の指紋画像計 10 枚分の中から 2 枚分を取り出した対：45 ペア

左人差し指同士　左人差し指の指紋画像計 10 枚分の中から 2 枚分を取り出した対：45 ペア

異なる指同士　右人差し指の指紋画像計 10 枚分及び左人差し指の指紋画像計 10 枚分，それぞれをペアにした計 100 ペア

認証精度に関する結果として，それぞれの場面における認証成功率を以下の表 1 に示す．

表 1　認証精度
Table 1　Authentication accuracy.

	$s = 200$	$s = 150$
右人差し指同士（同一の指）	0.489	0.289
左人差し指同士（同一の指）	0.378	0.289
右人差し指，左人差し指（異なる指）	0.34	0

もっとも認証成功率が高かったのは，閾値が 200 の場合における同一の右人差し指でありおよそ $\frac{1}{2}$ の確率で認証が成功する．閾値が 200 の場合における同一の左人差し指は 4 割に少々満たない程度成功率であるが，異なる指であっても 3 割強の割合で認証が成功してしまうことがわかった．また，閾値を 150 にした場合は，同一の指は (左右問わず)3 割に満たない認証成功率となったが，異なる指の認証成功率は 0 であった．

4. 考察と課題

3 章の実験を通して今後の課題としては，認証速度の向上及び精度の向上を図る必要がある．

速度の面において，照合可能暗号を用いた場合と用いない場合で有意差がなく，用いない場合における時間も平均 0.938 秒かかっていることから暗号化の有無よりも指紋からの平文生成時間が照合時間に大きく寄与していることを示唆している．IoT 機器は本実験環境よりも計算資源が少ないことが容易に考えられるため，指紋変換アルゴリズム及び開発言語の検討が必要であると言える．

次に精度の面において，認証精度としては良好であるとはいえない結果を得た．同一の指での実験結果から，同じ指で撮影した画像 2 枚で異なるバイナリ列を生成してしまう確率が高いことが容易にわかる．このことから指紋変換アルゴリズムの調整・検討が必要であると言える．今回は筆者らで考案したバイナリ列への変換アルゴリズムを用いたが，今後の方向性としては一方向性関数を用いたキャンセラブルバイオメトリクスやバイオメトリクス暗号などの中から，バイナリ列を生成が可能なものを吟味し実験を行っていきたいと考える．また，異なる指での実験結果から閾値を 200 と定めると，3 割強の確率で他人を本人と認識してしまうことがわかる．すなわち，閾値の設定を改善する必要がある．

よって，速度・精度のどちらの面においても指紋画像の変換アルゴリズムの早急な変更・改善が必要であると言える．

照合可能暗号をもとにしたアルゴリズムは，任意の個人情報に応用可能であり，既にクレジットカード番号や NFC 型 IC カードなどの情報が変動しない個人情報を用いた実装が行われている．3 章の実験でわかったように，ユニークな数字配列だけでなく生体情報の場合においても照合可能暗号の有無で有意差がなかったことから非常に高速に照合できると言える．今後，指紋だけでなく顔や静脈などのあらゆる生体情報にも対応した認証システムを開発していくことで，IoT 環境の認証の安全性の向上に繋がることが期待できる．また，照合可能暗号をもとにした認証アルゴリズムの応用として，複数のインターネットサービスなどのアクセス管理が可能となるシングルサインオンアルゴリズムも提案されている [13]．本提案が完成することで，[13] のアルゴリズムにも指紋をはじめとする生体情報を適用することができ，生体情報を用いたシングルサインオンシステムを実現できる．

5. おわりに

本稿では，IoT 環境における安全な認証を実装することを目的として，照合可能暗号をもとにした指紋認証システムを提案した．また，性能評価として認証精度と暗号化した場合と暗号化なしの場合で速度の比較を示し，その可能性について議論した．ID・パスワードや任意のカード番号などの不変な数字配列だけでなく，生体情報を使用した場合においても照合処理は非常に高速に処理できることがわかったため，今後は実際の認証場面で利用できるよう速度・精度の向上を目指していきたい．

参考文献

[1] 総務省：情報通信白書令和 2 年版 (online)，入手先 ⟨https://www.soumu.go.jp/johotsusintokei /whitepaper/ja/r02/html/nd114120.html⟩ (参照 2021.08.05).
[2] 独立行政法人情報処理推進機構 (IPA): 情報セキュリティ白書 2020，独立行政法人情報処理推進機構 (2020).
[3] フィッシング対策委員会: インターネットサービス提供事業者に対する「認証方法」に関するアンケート調査結果 (online)，入手先 ⟨https://www.antiphishing.jp/news/info/wg_auth_report_20190516.html⟩ (参照 2021.08.05).
[4] 笹川耕一: 指紋による個人認証, 生体医工学, 44.1, pp. 15–19 (2006).
[5] AN, Kataria., et al.: *A survey of automated biometric authentication techniques*, 2013 Nirma university international conference on engineering (NUiCONE), IEEE, pp. 1-6 (2013).
[6] 溝口正典，原雅範: 指紋掌紋の照合技術 (online)，入手先 ⟨https://jpn.nec.com/techrep/journal/g10/n03/pdf/100304.pdf⟩ (参照 2021.08.09).
[7] Rathgeb, C., Uhl, A.: *A survey on biometric cryptosystems and cancelable biometrics*, EURASIP Journal on Information Security, 2011.1, pp. 1–25 (2011).
[8] S, Shukla., SJ,Patel: *Securing fingerprint templates by enhanced minutiae - based encoding scheme in Fuzzy Commitment*, IET Information Security, 15.3, pp.256–266(2021).
[9] VS, Baghel., SS, Ali., S, Prakash.: *A non - invertible transformation based technique to protect a fingerprint template*, IET Image Processing, (2021).
[10] A, Fiat., A, Shamir.: *A. How to prove yourself: Practical solutions to identification and signature problems*, Conference on the theory and application of cryptographic techniques, Springer, Berlin, Heidelberg, pp. 186–194 (1986).
[11] CP, Schnorr.: *Efficient identification and signatures for smart cards*, Conference on the Theory and Application of Cryptology, Springer, New York, NY, pp. 239–252 (1989).
[12] Kihara, M., Iriyama, S.: *New authentication algorithm based on verifiable encryption with digital identity*, Cryptography 2019, 3, 19 (2019).
[13] Kihara, M., Iriyama, S.: *Security and Performance of Single Sign-On Based on One-Time Pad Algorithm*, Cryptography 2020, 4, 16 (2020).
[14] Bruno Blanchet(developper): ProVerif:Cryptographic protocol verifier in the formal model(online), 入手先 ⟨https://prosecco.gforge.inria.fr/personal/bblanche/proverif/⟩ (参照 2021.08.09).
[15] Kawagoe,M., Tojo.A.: *Fingerprint pattern classification*, Pattern Recognition, 17.3, pp. 295–303 (1984).
[16] Levi, G., Sirovich,F.: *Structural descriptions of fingerprint images*, Information Sciences 4.3–4, pp. 327-355 (1972).
[17] FG, Irwin.: *An Isotropic 3x3 Image Gradient Operator*, Presentation at Stanford AI Project 2014.02 (1968).
[18] Bastian Raschke: bastianraschke/pyfingerprint(online), 入手先 ⟨https://github.com/bastianraschke/pyfingerprint⟩ (参照 2021.08.09).
[19] Manuel Cuevas: cuevas1208/fingerprint_recognition(online), 入手先 ⟨https://github.com/cuevas1208 /fingerprint_recognition⟩ (参照 2021.08.09).
[20] OpenCV team: opencv(online), 入手先 ⟨https://opencv.org/⟩ (参照 2021.08.09).