スマートフォンにおけるフリック操作による画面ロック解除手法の提案

喜多 義弘
長崎県立大学
kita@sun.ac.jp

山下 湧介
長崎県立大学
bs217040@sun.ac.jp

要旨

スマートフォンにおいて，画面ロックを解除するための本人判定手法として，パスワード入力，暗唱番号入力，指紋認証，顔認証が用いられている．これらの手法は，ユーザにとって扱いやすく，かつ一定の安全性を保つことができる．しかし，パスワード入力や暗証番号入力においては覗き見攻撃への耐性が低いという課題が，一方，指紋認証や顔認証においては生体情報の再登録が容易ではないという課題がそれぞれある．そのため，解除に必要なこれらの情報が漏洩した場合，安全性が損なわれてしまうことが懸念される．そこで本研究では，これらの課題へ柔軟に対応するために，フリック操作による画面ロック解除手法を提案する．具体的には，パスワードとしての入力文字と，生体認証としてのフリック操作をそれぞれ判定することで，ユーザ本人かを判定する手法である．これにより，パスワードが漏洩してもフリック操作の生体情報で解除を防ぐことができ，漏洩後にパスワードを変更するだけでフリック操作の生体情報も更新することが容易にできる．

1. はじめに

スマートフォンには，タッチスクリーンの誤操作や他人による操作を防ぐことを目的として，画面ロック機能が搭載されている．画面ロックとは，タッチスクリーンでの操作を受け付けなくする機能であり，その解除のための本人判定手法としてパスワード入力，暗証番号入力，パターン入力，指紋認証，または顔認証が用いられている．

これらの手法は，入力が容易であるため，ユーザにとって扱いやすい．しかし，各手法には安全性に関する課題がある．例えば，パスワード入力，暗証番号入力，およびパターン入力は，解除中に後ろから画面を覗き見られ（この攻撃を以降，覗き見攻撃とする），入力情報が漏洩する．一方，指紋認証および顔認証は，覗き見攻撃への耐性はあるが，指紋や顔の生体情報が何らかの形で漏れた場合，漏れた情報を用いて画面ロックが解除されてしまう．実際に指が写った写真から指紋の情報を抽出し，それを用いて指紋認証が成功している例を報告した研究 [1] がある．そして，指紋認証や顔認証などの身体的な部分を用いた生体認証の場合，その生体情報の再登録には時間が掛かることや，再登録の回数に制限があることなど，生体情報漏洩への柔軟な対応ができていない．

そこで本研究では，覗き見攻撃への対応と，生体認証における情報漏洩への柔軟な対応を備えるため，フリック操作による画面ロック解除手法を提案する．具体的には，パスワードとして任意の文字列をフリック操作にて入力し，さらにフリック操作から動きの特徴を抽出し，それを生体情報としてユーザ本人の識別に用いる．これにより，後ろから画面を覗き見られてパスワードが漏洩し，そのパスワードを入力されても，そのフリック操作入力の動きがユーザ本人であると判定されなければ，ロックが解除されることはない．そして，ユーザ本人が次に使用する際に前回他人が操作したことを提示することで，パスワードが漏洩したと気づき，その文字列をフ

リック操作の生体情報も併せて即座に変更することにより，漏洩への対応を柔軟にかつ迅速に行うことができる．

本論文では，この提案手法をスマートフォン上に実装し，その有用性について評価および考察を行う．

2. 関連研究

スマートフォンを対象とした覗き見攻撃への対策として，フリック操作やキーボード操作の動きを用いた本人識別が複数提案されている．

Shahzad ら [2] は，フリック操作によるジェスチャーを利用した画面ロック解除手法を提案している．この提案では，タッチスクリーン上をフリックしながら特定のジェスチャーを行うと，その動きから，指の速度，デバイスの加速度，ストローク時間などの特徴を抽出し，それらから本人を判定する．これらの特徴は本人特有のものであるため，他人から覗かれ，動きを真似されても再現することができず，ロックは解除されない．

伊藤ら [3] は，スマートフォンにおいて，母音ごとのフリック操作から特徴を抽出し，その母音ごとの特徴から本人を識別する手法を提案している．この手法では，1 クラスサポートベクターマシン [4]（One Class Support Vector Machine, 以降，OCSVM とする）を用いている．

OCSVM は，教師なし学習の機械学習法の 1 つであり，すべての学習データを 1 つのクラスにまとめ学習し，その境界を設けることにより，境界から外れたデータは別のクラスとして扱う．これにより，本人ではないデータが入力されたとしても，本人とは異なる別のクラスとして扱うことができる．本研究では，この OCSVM を本人の判別器として用いる．

泉ら [5] は，キーボードを用いたキーストローク認証をスマートフォンに応用することを提案している．この手法では，本人識別に最近傍決定測を用いており，未登録の文字列から特徴を抽出し，予め登録している本人を含む複数人の特徴と比較し，特徴量の差を絶対的な距離に置き換えた場合に，最も距離が近い人物を入力した者として判定している．この手法は，認証の際に入力する文字列は任意のものでよいため，認証時のユーザ負担が小さいという利点がある．しかし，その文字列での特徴と比較するためには，事前にある程度の文字列長を持つ文章を登録する必要があるため，登録時のユーザ負担が大きいことが課題である．

3. フリック操作による画面ロック解除手法の提案

3.1. 提案手法の概要

本研究では，覗き見攻撃や情報漏洩への対応を備えるため，パスワード方式と生体認証方式を組み合わせた，フリック操作による画面ロック解除手法について提案する．具体的には，フリック操作で入力した文字列をパスワードとし，同じ文字列が入力された際にフリック操作の特徴を抽出し，本人の特徴とを比較し本人識別を行う方法である．パスワード方式と生体認証方式を組み合わせることにより，各方式の利点を活かしてもう片方の欠点をフォローし合うことができる．なお，各方式の利点および欠点は以下のとおりである．

- パスワード方式
  - 利点：パスワードの変更が容易である．
  - 欠点：覗き見攻撃への耐性が低い．
- 生体認証方式
  - 利点：覗き見攻撃への耐性が高い．
  - 欠点：生体情報の再登録が容易ではない，または，再登録回数に制限がある．

図 1 に，本提案手法の処理の流れを示す．この手法は，登録フェーズとロック解除フェーズの 2 部で構成する．

登録フェーズは，パスワードとそれを入力する際のフリック操作の特徴をそれぞれ登録するフェーズである．まず，ユーザは任意の文字列をフリック操作にて入力し，登録する．次に，登録した文字列はパスワードとして，ストレージへ保存する．フリック操作の特徴は，OCSVM を用いて学習を行い，ユーザ本人の特徴としてマップを作成する．このマップはパスワードとは別にストレージへ保存する．

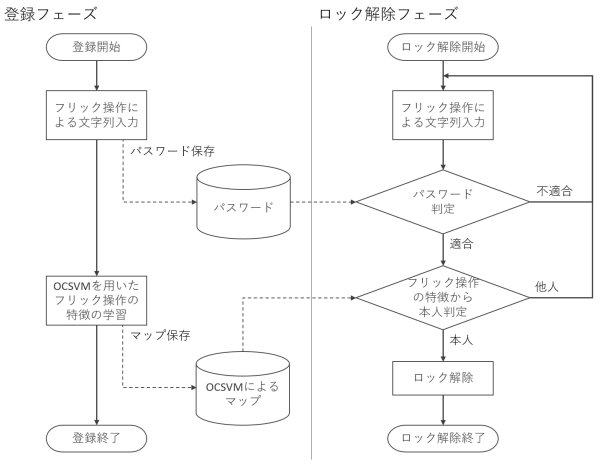

図 1: 本提案手法の各フェーズにおける処理の流れ

一方，ロック解除フェーズは，画面ロックを解除する際に登録時と同様のフリック操作を行い，画面ロックを解除するフェーズである．まず，ユーザは登録した文字列と同様の文字列を，登録時と同様のフリック操作で入力を行う．次にパスワードとして，入力した文字列が登録した文字列と同様かを判定する．この時点で入力した文字列が登録した文字列と異なっていれば，処理を中断し，文字列の入力待機に戻る．

入力した文字列が登録したものと同様であれば，文字列を入力した際のフリック操作の特徴から，ユーザ本人かの判定を行う．このとき，事前に作成していた OCSVM のマップを読み取り，そのマップ上に今回入力したフリック操作の特徴をプロットする．このマップからユーザ本人かの判定を行い，本人と判定されれば画面ロックを解除し，他人と判定されれば処理を中断し，文字列の入力待機に戻る．

3.2. フリック操作における特徴量の定義

フリック操作における特徴量として，以下を抽出する．

- 1 つのフリック操作にかかる時間（timePR）
- フリック操作の間隔時間（timeRP）
- フリック操作の挙動の長さ（lengthF）
- フリック操作の角度（angleF）

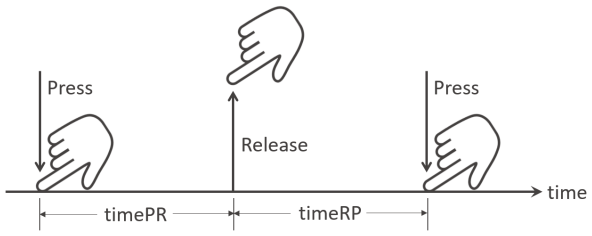

図 2: フリック操作時の指の動作における時間の定義

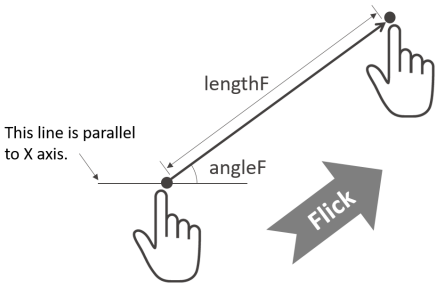

図 3: フリック操作における長さおよび角度の定義

以下では，各特徴量の定義およびその抽出方法について述べる．

図 2 に，フリック操作時の指の動作における時間の定義を，図 3 に，フリック操作における長さおよび角度の定義を，それぞれ示す．

図 2 において，スマートフォンの画面上を指で押してから離すまでの時間を timePR，指を離してから再び画面を押すまでの時間を timeRP と定義する．これらを文字数だけ取得するため，文字数を n としたとき，timePR は n 個，timeRP は $n-1$ 個の特徴をそれぞれ取得することができる．

図 3 において，画面上を指で押した座標（以降，始点とする）から，フリック操作を行った際に指が離れた座標（以降，終点とする）までの直線距離を lengthF，始点を通り X 軸に平行な線を想定したとき，その線と始点と終点を結んだ線から成る角度を angleF と定義する．これらも文字数だけ取得するため，文字数を n としたとき，それぞれ n 個ずつの特徴を取得することができる．

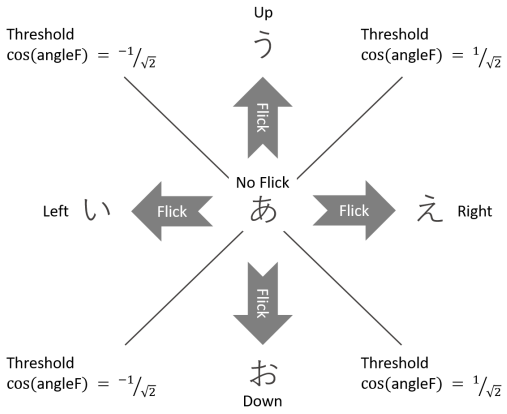

図 4: フリック操作の方向における母音判定の定義

図 4 に，フリック操作の方向における母音判定の定義を示す．フリック操作は,「フリックなし」および上下左右へのフリックの全 5 種類の操作を定めており，5 つの母音に対して図 4 のようにそれぞれ割り当てている．各操作の境界として，angleF の余弦が $1/\sqrt{2}$ または $-1/\sqrt{2}$ を閾値とする．また，子音については，始点の座標を基に判定する．

これらの特徴量をベクトルとし，OCSVM に入力する．このベクトルの次元数は，パスワードの文字数 n としたとき，$4n-1$ 次元となる．

3.3. 提案手法の実装

提案手法の有用性を確認するため，Android スマートフォンに対応するアプリとして実装した．

図 5 に，本提案手法を実装したアプリ画面におけるキーボードの外観を示す．キーボードには各子音を示すボタンを配置し，ボタン上でフリック操作を行うと，その位置に該当する子音とフリック操作に対応した母音を組み合わせて文字を入力する．このキーボードのキー配置は，Android スマートフォンに標準で搭載しているキーボードを参考にしている．

OCSVM を実装するため，機械学習のオープンソースライブラリである scikit-learn[7] を用いる．しかしながら，スマートフォン上で OCSVM のマップを作成するのは，実機のスペックによっては不安定になったり，処理に時間がかかったりと実用性に問題が生じる．

図 5: アプリ画面におけるキーボードの外観

そこで，スマートフォンにて取得したフリック操作の特徴量のデータをサーバに送信し，サーバ上で OCSVM のマップを作成した後，そのマップデータをスマートフォンに送信する手法を採用する．また，複雑な識別にも対応するため，OCSVM に用いるカーネルとして RBF カーネルを採用する．

パスワードの長さは，ひらがな 46 種を用いると想定し，少なくとも 7 文字の長さとする．この長さは，NIST SP800-63[6] において，標的を定めたパスワード推測攻撃に対するレベル 2 の要求事項（攻撃成功確率 2^{-14} 未満）を満たすことを基準にしている．ひらがなに限定した理由として，英数字 36 種（英文字 26 種 + 数字 10 種）の場合では必要になる長さが 8 文字以上になり，また，これらを組み合わせた場合では入力文字種の切り替えによる誤入力が出る可能性があり，入力操作によるユーザの負担が最小になることを考慮したためである．

実装したアプリの使用手順を以下に示す．

1. インストール後の初期実行時，パスワード入力画面を表示し，パスワードの登録をユーザへ促す．
2. ユーザはパスワードを 10 回入力し，登録する．
3. サーバへフリック入力のデータを送信し，OCSVM のマップデータを取得する．
4. 画面点灯時に本アプリが起動するように設定する．

5. 画面点灯し本アプリ起動後、画面ロックとしてパスワード入力画面を表示する．ユーザはパスワードを入力し，画面ロックを外す．以降，画面点灯時，手順 5 のみを行う．

本人判定時に他人であると判断した場合は，入力したのは他人であることを画面に明示し，ロックは解除せずパスワード入力画面に戻る．さらに，その後，本人がロックを解除した際，前回に他人がロック解除を試みた可能性があることを明示し，パスワードの変更を促すようにした．

4. 提案手法における有用性の評価実験

本提案手法の有用性を確認するため，以下の項目を検証する．

- 本人が入力した場合にロックが解除されること
- 他人が本人の真似をした場合はロックが解除されないこと

これらを検証するための実験を行う．以下では実験準備と実施，および，実験結果について述べる．

4.1. 実験準備と実施

実験の被験者は，長崎県立大学の 4 年次学生 10 名（男性 8 名，女性 2 名）とした．この中で，普段からフリック操作で文字を入力している被験者は 8 名であり，残り 2 名（いずれも男性）はボタンを連続でタップして文字を入力する「トグル入力」を普段用いている．今回の実験では，トグル入力を行わず，全被験者はフリック操作による入力で統一する．また，入力に用いる利き手は 10 人全員が右手である．各被験者に実験用のスマートフォンを 1 台ずつ渡し，各自，椅子に座り，スマートフォンを右手に持った状態で入力を行う．

登録するパスワードは，ひらがな 7 文字でかつ 5 種の母音が必ず入る文字列 10 パターンをこちらで事前に用意し，各被験者に示した．これらの文字列はランダムに生成したため，単語としての意味は持たない．文字数は 7 文字であるため，OCSVM に入力するデータのベクトルとしての次元数は，$4 \times 7 - 1 = 27$ 次元となる．

実験手順を以下に示す．今回の実験では，登録または解除のために入力したデータを抽出および分析できるようにパスワードの入力のみを行い，画面ロック解除の機能は外している．そのため，本人判定の成否は，PC 上で判定する．なお，OCSVM のマップはスマートフォンで利用するマップと同じものであるため，PC 上で判定してもその成否は変わらない．

1. 各被験者が 10 回ずつパスワードを入力する．このとき入力したデータを「登録用データ」とする．
2. 30 分間のブランク（休憩）を設け，パスワード登録時における被験者が持つ入力操作の感覚をリセットさせる．
3. 各被験者は登録したパスワードを再び 10 回入力する．このとき入力したデータを「解除用データ」とする．
4. 被験者間でスマートフォンを交換し，各被験者はそのスマートフォンで登録したパスワードを他人として 1 回入力する．これを本人以外のスマートフォンすべてが回るように 9 回繰り返す．このとき入力したデータを「模倣用データ」とする．
5. 登録用データを用いて被験者ごとの OCSVM を作成し，解除用データおよび模倣用データをそれぞれ入力して本人判定の成否を調べる．

手順 4 にて，被験者間でスマートフォンを交換する理由として，端末の違いによるフリック操作感への影響を軽減するためである．また，ユーザ本人が所有するスマートフォンを他人が操作することを考慮すると，パスワードを登録したスマートフォンに他人も同じパスワードを入力することが望ましいと考えられる．

4.2. 実験結果

表 1 に各被験者の本人判定成功数を，表 2 に各スマートフォンでの本人判定数を，それぞれ示す．各表中の値は，10 回または 9 回の入力のうちに判定成功または誤判定した数であり，末尾の値は平均回数を示す．被験者 I および被験者 J は，普段トグル入力を用いている者である．

表 1: 各被験者の本人判定成功数（10 回中）

被験者	A	B	C	D	E	
成功数	8	9	9	8	8	
被験者	F	G	H	I	J	平均
成功数	10	10	8	8	7	8.5

表 2: 各スマートフォンでの本人誤判定数（9 回中）

被験者	A	B	C	D	E	
誤判定数	1	1	3	2	1	
被験者	F	G	H	I	J	平均
誤判定数	2	1	2	0	1	1.4

表 1 より，本人判定成功率は被験者によって 10〜7 回となり，平均は 8.5 回であった．被験者 J は 7 回であったが，序盤の回に失敗が集中していたことから，操作に慣れていないことが考えられる．その他の被験者も前半 5 回に失敗が偏っていたため，実験による緊張や慣れが影響していると考える．

一方，表 2 より，他人が同じパスワードを入力した際にユーザ本人と誤判定する回数は，スマートフォンの端末によって 0〜3 回となり，平均は 1.4 回であった．誤判定が起こる理由として，本人と他人のフリック操作が似ていることが考えられる．そのため，登録用データに対する，解除用データおよび模倣用データの各 cos 類似度を計測し，定量的に比較する．

表 3 に，各被験者における登録用データと各データとの cos 類似度の平均値と分散を示す．これらの値は，登録データにおけるすべての被験者および回ごとのデータ（データ数：100）に対して，解除用データおよび模倣用データのすべての被験者および回ごとのデータ（データ数：100 および 90）をすべての組み合わせで cos 類似度を求め，それぞれの平均を算出したものである．

cos 類似度の各平均値は，模倣用データの方が解除用データに比べて低い傾向にあるため，本人と他人とを区別することは可能であると考えられるが，分散を考慮すると類似度の高低が逆転することもあり，明確に区別することは難しいことが表 3 より読み取ることができる．

表 3: 各被験者における登録用データと各データとの cos 類似度の平均値と分散

被験者	A	B	C	D	E
解除用データ					
平均値	0.962	0.985	0.982	0.977	0.973
分散	0.031	0.012	0.012	0.014	0.022
模倣用データ					
平均値	0.943	0.952	0.971	0.969	0.955
分散	0.025	0.023	0.006	0.020	0.032
被験者	F	G	H	I	J
解除用データ					
平均値	0.998	0.989	0.988	0.958	0.966
分散	0.001	0.003	0.007	0.033	0.031
模倣用データ					
平均値	0.967	0.951	0.972	0.891	0.903
分散	0.021	0.034	0.021	0.032	0.042

表 4: 母音別の登録用データと各データとの cos 類似度の平均値と分散

母音	あ	い	う	え	お
解除用データ					
平均値 (A)	0.998	0.975	0.978	0.982	0.988
分散	0.001	0.011	**0.019**	0.010	0.007
模倣用データ					
平均値 (B)	0.996	0.962	0.953	0.974	0.980
分散	0.001	0.026	**0.032**	0.012	0.010
平均値の差					
(A)−(B)	**0.002**	0.013	**0.025**	0.008	0.008

さらにパスワードに使用する文字により本人判定の精度へ影響を考慮するため，使用したパスワードの文字を各母音ごとに分類し，それぞれの文字について上記と同様に cos 類似度を算出した結果を，表 4 に示す．

表 4 より，解除用データと模倣用データの平均値の差が最も小さいのは母音「あ」であり，最も大きいのは母音「う」であることが確認できる．平均値の差が小さいほど互いのデータは似ているため，本人判定の精度が低くなり，差が大きいほど他人は模倣しづらいということになり，本人判定の精度が高まると予想される．また，両データにおける各母音の分散に注目すると，母音「う」の分散が大きいことも確認できる．

5. 考察

5.1. 実験での検証項目についての考察

今回の実験の検証項目について考察する．まず，項目「本人が入力した場合にロックが解除されること」について，表 1 より，本人判定成功数の平均は 10 回中 8.5 回であった．連続で失敗することを考慮した場合，2 回連続で失敗する確率は約 2.3%，3 回連続で失敗する確率は約 0.3%になり，実用的な回数のうちにロックを解除できることが見込まれる．また，最も成功数が少ない被験者 J は，普段トグル入力を用いてたことから操作に不慣れでいたことが考えられることから，フリック操作に慣れることによってその動きが安定し，失敗しにくくなると考える．

次に，項目「他人が入力した場合にロックが解除されないこと」について，表 2 より，I 以外の被験者は何れかのユーザ本人として 1 回以上は誤判定されていることや，被験者 C においては 10 回中 3 回と最も多い誤判定があることから，本項目については達成できていないと考えられる．この要因としては，OCSVM において学習に用いる登録データの数が少ないことも挙げられるが，同じパスワードを打つことでフリック操作も同じ操作となるため，図 3 で示すとおり，本人と他人との差が出にくいことが最も大きく影響していると考えられる．

他人の場合，自身が普段から扱うスマートフォンと異なる場合もあり，持ち方やパスワードの文字入力間隔に本人との違いが出てくるため，それらが関わる特徴量を分析する必要がある．図 4 において，母音ごとの本人と他人との差を cos 類似度で算出した結果，前述したとおり，母音「あ」では cos 類似度の差はなく，母音「う」が cos 類似度の差が最も出ることが確認できた．母音「あ」の場合は，フリック操作はなくタップのみで入力する形になるため，他の母音よりも差が出る特徴量が少ないことが考えられる．一方，母音「う」の場合は，上方向へのフリック操作であり，操作する指は伸ばす形になる．この指の長さや，スマートフォンを持つ位置によってはフリックの角度 angleF が変わるため，個人差が表れやすいと考えられる．同様に，左方向へ指を伸ばす操作になる母音「い」も，母音「う」に次いで cos 類似度の平均の差が出ていることから，操作する指の動きが本人判定へ影響を与えると考えられる．母音「え」や母音「お」の場合，指を曲げながらのフリック操作になり，フリックの長さは短く，角度も差が出にくい．そのため，個人差が出にくいと考えられる．

5.2. 既存手法との比較

本提案は，パスワード方式とフリック操作による生体認証方式を組み合わせることにより，覗き見攻撃への耐性と，パスワードを容易に変更できる柔軟性を有している．これは既存手法にはない，本提案独自のものであり，新規性であると考える．

本提案に近い既存手法として，Shahzad ら [2] の研究があるが，彼らの手法はスマートフォンの画面上でジェスチャーを表すためのフリック操作をしている．ジェスチャーを表すことにより，その様子を後ろから覗き込む攻撃者にとっては，そのジェスチャーが画面ロック解除に直接結びついた鍵であることが明らかになり，そのジェスチャーを模倣することに注力する．一方，本提案においては，パスワードと同時に，そのパスワードを入力するフリック操作も画面ロック解除の鍵としているため，攻撃者にとっては両方を同時に覚える必要がある．そのため，本提案は既存手法よりも攻撃しにくい手法であると考える．

しかしながら，本提案における本人判定の精度については，既存手法と比較して低いことが懸念される．伊藤ら [3] や泉ら [5] の研究では，認証精度は 9 割台に達しているが，本提案では実験において 10 回中平均 8.5 回の成功数に留まっている．その要因として，OVSVM における学習させるデータやその特徴量が少ないことが挙げられるが，画面ロックを解除する度にその入力データを蓄積しておき，定期的に学習することでその差を補うことができると考える．特にスマートフォンはユーザ本人のみが扱うため，本人に特化した OCSVM のマップを作成することが可能であり，他人からの攻撃を判別しやすくなると考える．

5.3. 入力の慣れへの対応についての考察

今回の実験を通して被験者を観察していると，回を重ねるごとに入力がスムーズになり，文字入力の間隔が短くなっている傾向にあった．本実験では，事前に作成

したランダムなパスワードを用いたため，被験者は当初慣れない文字列で入力が遅かった．このことが実験結果に影響していることは多少懸念されるが，実用性を考慮した場合，画面ロック解除の際に入力したデータを蓄積しておき，定期的に学習を行い，OCSVM のマップを更新する必要があると考える．

OCSVM のマップを作成するには時間を要するため，例えば，サーバと連携し，サーバで OCSVM のマップを作成しておき，その間にスマートフォンでは画面ロック解除時の入力データを蓄積しておく．そして，OCSVM のマップを更新するため，サーバからそのマップを受け取り，それと同時にサーバへ今まで蓄積したデータを渡す．これらの繰り返しにより，定期的に OCSVM のマップを更新することが可能になる．

このような入力の慣れに対応することにより，ユーザ本人に特化した OCSVM のマップへ更新でき，本人判定の精度も向上できることが見込まれる．

6. おわりに

本研究では，覗き見攻撃への対応と，生体認証における情報漏洩への柔軟な対応を備えるため，フリック操作による画面ロック解除手法を提案した．本提案をスマートフォン上に実装し，被験者 10 人に対して試用実験を行った．その結果，本人判定の際に他人からの入力をユーザ本人として誤判定する可能性があることを確認したが，その解決方法として，個人差を判定しやすい特徴量を用いることや，ユーザ本人に特化した OCSVM のマップへと更新し続けることにより，対処していく方法を考察し，今後検討していく方針を定めた．

これらの課題を解決することにより，本人判定の精度を向上することができ，本提案手法の有用性が認められれば，画面ロック解除時における覗き見攻撃の対策や，生体情報漏洩による再登録の柔軟性に寄与できることが見込まれる．

今後の課題として，以下を挙げる．

- フリック操作から抽出できる特徴量の再検討
- ユーザの「慣れ」への対応
- 安全なサーバ運営を含む実用性を考慮したシステムの検討

参考文献

[1] 越前功, 大金建夫, “写真からの指紋復元の脅威とその対策技術”, 情報処理学会論文誌, Vol.58, No.9, pp.824-829, 2017.

[2] Shahzad, M., Liu, X.A., Samuel, A., “Secure unlocking of mobile touch screen devices by simple gestures: you can see it but you can not do it”, Proceedings of the 19th Annual International Conference on Mobile Computing & Networking, pp.39-50, 2013.

[3] 伊藤駿吾, 白石陽, “スマートフォンのフリック入力方式の特徴に着目した継続認証手法の提案”, 第 25 回マルチメディア通信と分散処理ワークショップ論文集, pp.1-8, 2017.

[4] Schölkof, B., Smmola, A.J., Williamson, R.C., and Bartlett, P.L., “New Support Vector Algorithms”, *Neural Computation*, Vol.12, No.5, pp.1207-1245, 2000.

[5] 泉将之, 佐村敏治, 西村治彦, “フリック入力による日本語非定型文のキーストローク認証”, 情報処理学会関西支部 支部大会, E-15, pp.1-8, 2013.

[6] National Institute of Standards and Technology, “Special Publication 800-63: Digital Identity Guidelines, Version 1.0.2”, 2006.

[7] David Cournapeau, “scikit-learn Machine Learning in Python”, https://scikit-learn.org/stable/ (最終閲覧：2021 年 3 月 22 日)