Vol.2018-GN-105 No.3
Vol.2018-SPT-28 No.3
2018/5/10

安全性と利便性を考慮したチャレンジ・レスポンス分離ユーザ認証に関する提案：Dummy Indicator

登内雅人[†1]　遠藤将[†2]　西垣正勝[†3]

概要：携帯端末におけるユーザ認証の脅威として覗き見攻撃が考えられる．覗き見攻撃に対しては認証方式のチャレンジ&レスポンス化が基本対策であり，既存方式の多くでは，チャレンジを覗き見攻撃者から隠すことで安全性を確保するという方法が採られている．しかし，本来チャレンジ&レスポンス型の認証方式ではチャレンジまたはレスポンスのどちらかを隠すことができれば複数回の覗き見攻撃に対して安全性を確保することができる．本研究では，携帯端末の物理的構造（表裏）を利用することで，攻撃者がチャレンジとレスポンスを同時に盗聴することが困難となる覗き見対策手法を提案する．また，携帯端末の背面という操作部が目視できない入力インタフェースにおいて，ユーザが正確に秘密情報の入力を行えるようなフィードバックの与え方について検討する．

キーワード：ユーザ認証，覗き見攻撃，ユーザインタフェース

1. はじめに

スマートフォンの急速な普及に伴い，携帯端末上でプライバシ情報や機密情報を扱う場面が増加している．プライバシ情報および機密情報保護のため，携帯端末には PIN（Personal Identification Number）やパターンロックなどのユーザ認証技術が導入されている．しかし，これらの認証手法においては，認証行為を覗き見することで認証情報（PIN やパターンロックなどの秘密情報）を不正に取得することが可能である．このような攻撃は覗き見攻撃と呼ばれ，ユーザ認証における脅威の一つとして知られている．

覗き見攻撃を対策するためには，認証情報をワンタイム化することが肝要である．認証情報をワンタイム化する手法としては，認証方式をチャレンジ&レスポンス形式にすることが有効である．現在までに提案されているチャレンジ&レスポンス方式では，チャレンジを覗き見攻撃者の盗聴から隠しながらユーザに受け渡すことによって安全性を確保するという方法が主に採られている．しかし，チャレンジを正規ユーザにのみ秘密に受け渡すタイプの既存方式の多くには，ユーザのメンタルタスクが過大となる問題[1][2]が，また，チャレンジをダミーの中に隠すタイプの既存方式の多くは，マウス操作やクリック音により入力する認証情報が漏れてしまう可能性があるという問題[3]が，それぞれ存在する．

本研究では，携帯端末の画面にチャレンジを表示し，背面からレスポンスを入力することで，攻撃者にチャレンジとレスポンスを同時に盗聴されることを防止する覗き見対策手法を提案する．携帯端末の背面からの入力は，現在普及している多くの携帯端末に設置されている内蔵カメラを用いて実装するため，提案方式は専用デバイスの所持を必要としない．その一方で，携帯端末の背面から入力を行うという方法は，ユーザから操作部が見えない入力インタフェースとなるため，正確な入力を行うことが比較的難しくなり,利便性の低下につながることが想定される.そこで，自分がどのような操作を行っているのかという操作感をユーザに与えつつ，覗き見攻撃者に対しては秘密情報に関する情報を一切与えないようなフィードバックの提示方法について検討する．

本論文の構成は以下のとおりである．2 章で覗き見対策における既存研究とその課題，背面入力インタフェースにおける関連研究について述べる．3 章で提案方式について説明した後，4 章で提案方式に対する基礎実験の報告をする．5 章でまとめと今後の課題を述べる．

2. 関連研究

2.1 チャレンジ&レスポンス方式を利用した覗き見対策

覗き見攻撃を対策するためには，認証情報をワンタイム化することが肝要である．認証情報をワンタイム化するには，チャレンジ&レスポンス型の認証方式にすることが有効である．チャレンジ&レスポンス型の認証方式の一般的な手順は以下のとおりである．認証システムには，あらかじめユーザが秘密情報を登録してあるものとする．

【手順】

① 認証システムは，チャレンジを被認証者（ユーザ）へ提示する．

② ユーザは，自身の有している秘密情報とチャレンジからレスポンス（認証システムへ入力する情報）を計算する．

③ ユーザは，②で計算したレスポンスをシステムへ入力する．

④ システムは，登録されている秘密情報と①で提示したチャレンジから（入力されるであろう）レスポンスを

†1 静岡大学情報学部
Faculty of Informatics, Shizuoka University
†2 静岡大学大学院総合科学技術研究科
Graduate School of Integrated Science and Technology, Shizuoka University
†3 静岡大学創造科学技術大学院
Graduate School of Science and Technology, Shizuoka University

Vol.2018-GN-105 No.3
Vol.2018-SPT-28 No.3
2018/5/10

計算する．この値と③で入力された値が一致していた場合，被認証者を正規ユーザとして認証する．

チャレンジ&レスポンスの本来の目的は，通信路を盗聴する攻撃者に対し，通信路に流れるチャレンジとレスポンスから秘密情報を逆計算することを防ぐことにある．これを達成するには，②における計算の際に暗号演算（典型的にはハッシュ値の計算）が必要となる．しかし，人間は暗号演算のような複雑な計算を行うことはできない．そこで，毎回のチャレンジに対するレスポンスを人間に計算させるタイプのユーザ認証では，複数回の覗き見攻撃に耐性を持たせるために，主にチャレンジを覗き見攻撃者から隠す方法が採られている．

2.1.1 fakePointer

fakePointer は，安全な環境（覗き見が不可能な通信路）を用いてユーザのみにチャレンジを渡す方式である[2]．

認証手順を以下に示す．認証システムには，あらかじめユーザが 4 桁の PIN（各桁は 0～9 のうちいずれかの整数）を登録しているものとする．

【手順】

① ユーザは前もって，人目に晒されない安全な環境下でチャレンジとなる選択シンボル情報を取得しておく．選択シンボル情報は四つの記号のいずれかであり，システムが認証毎にランダムに決定する．

② 覗き見攻撃の危険性がある環境下で認証をする際，ユーザは自身の秘密情報となる PIN と①で取得しておいた選択シンボル情報を利用して，③④のようにレスポンスの入力を行う．

③ 図 1 のように認証画面が表示される．ユーザは左右ボタンによって記号上の数字を左右にシフトさせ，1 桁目の PIN と 1 つ目の選択シンボル情報を重ね合わせて決定ボタンを押す．

④ ③を 4 回繰り返すことで 4 桁の PIN の入力を行う．

fakePointer は，覗き見攻撃に対して安全な認証方式となっている．しかし，認証毎に，事前に安全な環境上でチャレンジ（選択シンボル情報）を取得しておかなければならず，チャレンジをレスポンスの入力まで記憶し続けておかなければならない．これは，ユーザにとって大きな負荷となり得る．

図 1　fakePointer

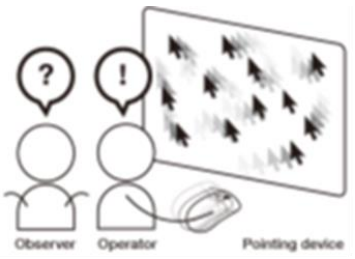

図 2　ダミーカーソル

図 3　Cursor Camouflage の認証画面

2.1.2 ダミー入力の表示による覗き見対策

Cursor Camouflage は，図 2 のように複数の独立に動くダミーカーソルを画面に表示し，ユーザが実際に入力している PIN を覗き見攻撃者によって特定されるのを困難にする方式である[3]．以下に認証手順を示す．認証システムには，あらかじめユーザが 4 桁の PIN（各桁は 0～9 のうちいずれかの整数）を登録しているものとする．

【手順】

① 図 3 のように認証画面が表示される．ユーザはマウスによるカーソル操作を行い，自身の操作に連動して動く本物のカーソルを発見する．

② 発見したカーソルを PIN の上に移動させ，マウスをクリックする．

③ ②を 4 回繰り返すことで PIN の入力をおこなう．

Cursor Camouflage は，①の本物のカーソルの発見をチャレンジの受診，②のマウスクリックをレスポンスの入力と捉えれば，チャレンジ&レスポンス型のユーザ認証の一方式といえる．しかし，攻撃者は認証画面と同時にユーザのマウス操作も盗聴が可能であり，マウス操作（やクリック音）により入力情報が漏れてしまう可能性がある．また，ユーザのカーソル操作の癖により，覗き見攻撃者に本物のカーソルを特定されてしまう危険もある．

2.2 端末背面からの入力を利用した覗き見対策

本来，チャレンジ&レスポンス型の認証では，チャレン

ジまたはレスポンスのどちらかを隠すことができれば覗き見攻撃に対して安全性を確保することができる．そして，その実現には，端末背面からの入力インタフェースが利用可能である．文献[4][5]が端末背面に背面入力用の入力デバイスを追加装備する方式であるのに対し，LensGesture [6]では端末背面の内臓カメラを入力インタフェースとして用いる．本研究では，追加デバイス不要な LensGesture に着目する．

LensGesture の操作イメージを図 4 に示す. LensGesture の操作方法としては，Static LensGesture と Dynamic LensGesture という二つのアプローチが提案されている．それぞれ図 5 のように背面カメラに映る映像から入力を判別する．

Static LensGesture は，ユーザの指によって背面カメラ全体が覆われている状態（全覆），部分的に覆われている状態（左覆，右覆，下覆）の 4 状態の動作によって入力操作を行う（図 6 上）．4 状態を識別するために，背面カメラから取得した映像をリアルタイムで解析し，輝度によって背面カメラがどのように覆われているかどうかの判定を行っている．

Dynamic LensGesture は，ユーザの指が背面カメラの前を左右あるいは上下方向に通過する動作によって入力操作を行う（図 6 下）．上下左右の 4 動作を識別するために，背面カメラで取得した映像をリアルタイムで解析し，前後のフレームを比較することで指がどの方向からどの方向へ動いたか判定する．

図 4　LensGesture


図 5　操作時のカメラ映像の例

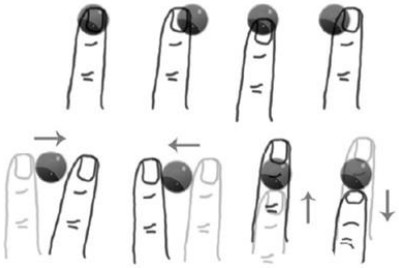
図 6　操作方法

3. Dummy Indicator

本稿では，2.1.2 節の Cursor Camouflage と 2.2 節の Static LensGesture を融合させた入力インタフェースを実装することによって，攻撃者がチャレンジとレスポンスを同時に盗聴することが困難なチャレンジ&レスポンス型ユーザ認証の実現を目指す．

チャレンジとレスポンスを分離するためには，携帯端末の画面（表）にチャレンジを表示するようにした上で，端末の背面に設置された内臓カメラ（裏）を用いてレスポンスを入力させる必要がある．ここに Static LensGesture のコンセプトが有効に働く．現在普及している携帯端末には背面にカメラが設置されていることが一般的であるため，Static LensGesture は多くの携帯端末に適用可能であると考えられる．

インタフェースデザインの分野において，ユーザに適切なフィードバックを与えることが利便性の面で重要であることが知られている．背面カメラを使う LensGesture においては，ユーザは操作部を目視できないため，レスポンスを正確に入力することが難しくなると想定される. そこで，自分がどのような操作を行っているのかという操作感をユーザに与えつつ，攻撃者に対しては入力内容に関する情報を与えないようなフィードバックの提示方法を検討する必要がある．ここに Cursor Camouflage のコンセプトが有効に働く．正規のフィードバックの提示と同時に，ダミーとなるフィードバックを提示することで，操作をしているユーザには正規のフィードバックを特定できるが，攻撃者には正規のフィードバックを特定できないようになることが期待される．

Cursor Camouflage と Static LensGesture を融合させた提案方式を「Dummy Indicator」と呼ぶ．なお，Static LensGesture では全覆，左覆，右覆，下覆の 4 状態を識別しているが，提案方式では，全方位の入力操作が可能となるように拡張することを試みる．具体的には，ユーザが背面カメラを指で覆う際の画像を機械学習によって処理し，背面カメラをポインティングスティック[7]のように扱う仕組みを実現する．携帯端末の画面上に表示されたマウスカーソル（インジケータ）を，背面カメラを使って自在に操作するユーザインタフェースを提供することによって，ユーザにより柔軟な入力操作とより的確なフィードバックを与えることができる．

Vol.2018-GN-105 No.3
Vol.2018-SPT-28 No.3
2018/5/10

4. 基礎実験

4.1 実験目的

本研究の基礎実験として，提案方式の操作性（カーソル操作の利便性）と視認性（ダミーを含むインジケータの中から本物を見つけることが容易いか）を調査する．

4.2 実験システム

提案方式の基礎実験を行うため，提案方式の実装を行った．実験用携帯端末の諸元は次のとおりである．機種：ARROWS NX F-01F，サイズ：約 140mm×約 70mm×約 10mm，重量：約 150g，OS：Android 4.2.2，開発言語：java．実験用携帯端末の画面を図 7 に示す．背面カメラに映るユーザの指の映像をリアルタイムで 2 層の NN（ニューラルネットワーク）に入力し，インジケータの進む方向を自動判定する．NN は scikit-learn[8]を利用した．NN の学習を行うに当たり,「画面上のランダムな点にカーソルを表示させ,実験実施者（著者）がその点を入力しているつもりで背面カメラに対して指を置く」という操作を 1000 回繰り返すことによって，教師データ（「ユーザの意図した入力点」と「その際の背面カメラに写る指の画像」のペア）を収集した．提案方式を実装した携帯端末は 1 台であり，全被験者がこの端末を使って実験を行う．

4.3 実験方法

情報系学部の大学生 3 名に実験被験者を依頼した．被験者には，まず，提案方式における背面カメラでの入力操作に慣れてもらうため，自身が十分と思えるまで練習を行ってもらった．その後，ダミーのインジケータの数が 4，9，19 個（本物のインジケータを含めた数は 5，10，20 個）の各状況において，背面カメラによりインジケータを操作しながら本物のインジケータを発見できるかどうかの試行を,それぞれ 10 回行ってもらった．被験者には本物のインジケータを発見した時点で画面をタップしてもらい，特定までにかかった所要時間を計測する. なお, 今回の実験では，各インジケータはアルファベットで分けて表示しており，毎回の試行の後に被験者に発見したインジケータのアルファベットを回答してもらい，成否を判定する．

4.4 実験結果

実験結果を表 1 に示す．被験者がインジケータを発見するまでの平均所要時間は，インジケータが 5 個の際に 5.28 秒，10 個の際に 7.15 秒，20 個の際に 12.56 秒，正答率はそれぞれ 96.7%，96.7%，100%であった．誤回答はインジケータが 5 個の際と 10 個の際に 1 人の被験者に 1 回ずつ発生したのみであった．

表 1 から，背面カメラという入力インタフェースによって，ダミーを含む中でも自分のインジケータを発見することが可能であると言える．ただし，所要時間については改善が必要であり，今後，操作性の向上等により本物のインジケータを発見するまでの所要時間を減らすことを目指す．

表 1　実験結果の集計

インジケータ数	平均所要時間[s]	精度[%]
5	5.2799	96.7
10	7.1519	96.7
20	12.5644	100

また，今回の実験結果をインジケータ数ごとに集計したデータを表 2 から表 4 に示す．この結果から，被験者ごとで所要時間にばらつきがあることが分かる．これは，人により操作に対する慣れの度合いや操作の仕方及び手の大きさ等の違いによるものだと思われる．今後，認証システムの初回利用時にユーザの指の動かし方を登録しておき，それによってユーザごとに NN の学習結果性を微調整する方法などの適用を検討する．

4.5 安全性に関する考察

提案方式は，チャレンジとレスポンスの同時盗聴を防ぐことにより，単発の覗き見攻撃者に対して安全な認証方式の実現を期待できるが，複数回の覗き見を行うことができる攻撃者に対しては，ユーザのカーソル操作の癖などにより，本物のインジケータが特定されてしまう恐れがある．具体的には，ユーザは毎回同じようなカーソル操作（円を描く，上下左右繰り返し等）でダミーの中から本物のインジケータを発見しようとするため，攻撃者がその動きから本物のインジケータを容易に発見してしまうことが考えられる．これは覗き見攻撃に対する安全性上の問題となる．今後はその検討を行う必要がある．

表 2　被験者毎の結果（インジケータ 5 個）

	平均所要時間[s]	精度
被験者1	6.4393	100%
被験者2	4.2214	100%
被験者3	5.179	90%

表 3　被験者毎の結果（インジケータ 10 個）

	平均所要時間[s]	精度
被験者1	8.2938	100%
被験者2	6.3159	100%
被験者3	6.8461	90%

表 4　被験者毎の結果（インジケータ 20 個）

	平均所要時間[s]	精度
被験者1	12.7011	100%
被験者2	6.173	100%
被験者3	18.8193	100%

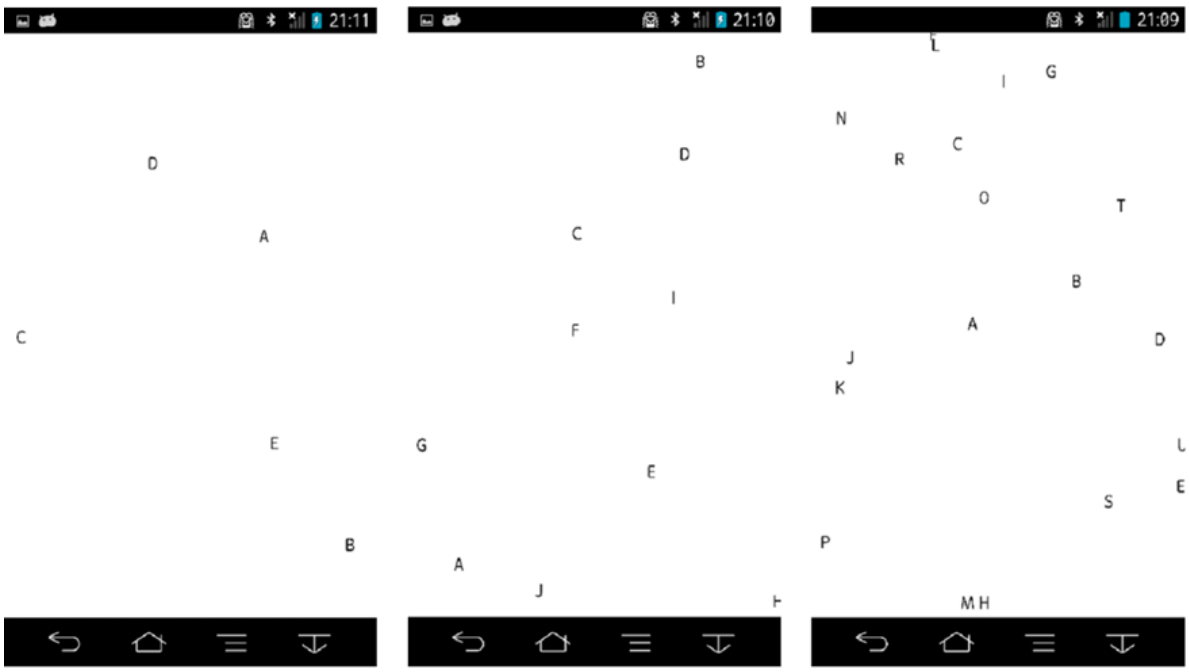

図 7　実験画面（左からインジケータ 5 個，10 個，20 個）

5. まとめと今後の課題

本稿では，携帯端末の物理的構造（表裏）によりチャレンジとレスポンスの同時盗聴を防止することで，ユーザ認証の覗き見対策を行う方式の実現に向けて，Cursor Camouflage と Static LensGesture を融合させた入力インタフェースを提案した．本研究の基礎実験結果より，提案方式は多数のダミーを含んでいても本物のインジケータを十分操作可能であることが確認された．しかし，本物のインジケータを発見するのにかかる所要時間は改善の必要があり，そのために背面カメラという入力インタフェースでの操作性の向上が当面の課題となる．今後は上記に加え，提案方式の覗き見に対する安全性の検証を行い，安全で利便性の高い認証方式の実現を目指す．

参考文献

[1] 徐強，西垣正勝，“ニーモニックに基づくワンタイムパスワード型画像認証の実現可能性に関する検討”，情報処理学会研究報告 Vol.2006-CSEC-32, pp.317-322, 2006.
[2] 高田哲司，“fakePointer:映像記録による覗き見攻撃にも安全な認証手法”，情報処理学会論文誌，Vol.49. No.9, pp. 3051-3061, 2008.
[3] Keita Watanabe, Fumito Higuchi, Masahiko Inami, Takeo Igarashi, “CursorCamouflage: Multiple Dummy Cursors as A Defense against Shoulder Surfing”, SIGGRAPH Asia 2012 Emerging Technologies, 2012.
[4] 平岡茂夫ら，“Behind Touch : 携帯電話のための背面・触覚操作インタフェース”，情報処理学会論文誌，Vol.44. No.11, pp. 2520-2527, 2003.
[5] 岡田直之ら，“背面タッチパッドを用いた片手ポインティング”，研究報告ヒューマンコンピュータインタラクション（HCI），Vol.2009-28(2009-HCI-132), pp.25-32, 2009.
[6] X. Xiao, T. Han, J. Wang, “LensGesture: augmenting mobile interactions with back-of-device finger gestures”, ICMI 2013, pp.287-294, 2013.
[7] Pointing stick (April. 3, 2018, 13:19 UTC). In Wikipedia: The Free Encyclopedia. Retrieved from https://en.wikipedia.org/wiki/Pointing_stick
[8] David Cournapeau, “scikit-learn: machine learning in Python — scikit-learn 0.19.1 documentation”, Retrieved from http://scikit-learn.org