



平成 25 年度 卒業論文

Twitter を用いた携帯端末における
個人認証の多要素化に関する研究

電気通信大学 情報理工学部 総合情報学科

1010086 高浪悟

指導教官 高田 哲司 准教授

提出日 平成 26 年 1 月 31 日

概要

個人認証の安全性を高める手法として多要素認証がある。多要素認証は、(1) 知識認証、(2) 所有物認証、(3) 生体認証といった認証要素を複数組み合わせることで、何らかの攻撃により一つが破られても他の認証要素があることで不正利用からアカウントを守る手法である。しかし、導入コストの大きさや利用可能な状況が限られるといった利便性の面から問題がある。

本論文では、個人認証の多要素化について調査を実施した。そこで我々は、多要素認証普及の一要素となっていながらそれ自体については個人認証の多要素化があまり行われていない携帯端末に注目し、安全性と利便性の双方を大きく損なうことのない多要素認証の提案を研究の目的とした。

更に、利便性の向上、特に覚えやすさを改善した認証を目指すべく、近年大きく普及したSNSを、能動的に記録を行うライフログとして利用できないかと考え、既存研究の調査をした上で、それらの問題点の洗い出しと提案を行った。

我々は、秘密情報としてTwitterの投稿を用いた認証システムとして、(1) 特定の一つを自ら選択する、(2) 時系列上における期間の指定、(3) 時系列上における時間と曜日の指定、の3つの秘密情報の設定方法を持つ認証システム Notifauthを開発し、それぞれの設定方法について検証・評価を行った。(1)については、秘密情報でTwitterを用いることで得られる安全性や利便性の向上について主に調査し、(2)と(3)については、ある一定のルールに基づいて秘密情報が変化することで認証の成功率やユーザへの負担がどれほど変化するかを主に調査した。

被験者実験による検証の結果、(1)については、8日間という期間ではあるが、記憶維持が可能であることが明らかになった。(2)と(3)ではそれぞれ、暗証番号(PIN)による認証方法よりも認証成功率が低くなる結果となった。被験者によるアンケートでは、PINによる認証手法よりも(1)の設定方法による認証手法の方が使いやすさと覚えやすさ共に高く評価された。今後の課題としては、条件は覚えているがそれに適合する秘密情報が選べないこと、認証にかかる時間の増加の改善方法について取り組むべきであることを議論した。

目 次

第 1 章 序論	9
1.1 背景	9
1.2 研究目的	10
1.3 論文の構成	11
第 2 章 個人認証の多要素化への流れ	12
2.1 既存の認証技術	12
2.1.1 知識認証	12
2.1.2 所有物認証	13
2.1.3 生体認証	15
2.2 多要素認証	17
2.2.1 概要	17
2.2.2 代表的手法	19
2.2.3 利点と欠点	19
2.2.4 既知の攻撃方法による脆弱性	20
2.3 スマートフォン/タブレットの普及	21
第 3 章 関連研究/製品	23
3.1 多要素認証についての調査	23
3.1.1 二要素認証のユーザビリティに関する比較調査	23
3.2 認証の多要素化手法	25
3.2.1 Google	25

3.2.2 PassBoard	28
3.2.3 Authy	29
3.2.4 オンラインゲームにおける多要素化例	30
第 4 章 動機と提案	33
4.1 動機	33
4.1.1 携帯端末への多要素認証の導入	34
4.2 提案	34
4.2.1 ライフログや SNS を利用した個人認証事例	35
4.2.2 提案手法の概要	39
第 5 章 Twitter 上の情報を用いた認証システム	42
5.1 概要	42
5.1.1 Twitter の使用	42
5.1.2 携帯端末への導入	45
5.1.3 実装の概観	48
5.2 密情報の設定	49
5.2.1 Auto Mode Type Term	49
5.2.2 Auto Mode Type Cycle	51
5.2.3 Manual Mode	55
5.3 認証操作	55
5.4 システムの使用にあたって	57
5.5 開発環境	59
第 6 章 検証実験	60

6.1 概要	60
6.1.1 実験手順	61
6.1.2 被験者	63
6.2 Manual Mode を用いた認証方式の評価実験	64
6.2.1 概要	64
6.2.2 目的	64
6.2.3 方法	65
6.2.4 結果	65
6.3 Auto Mode Type Term を用いた認証方式の評価実験	68
6.3.1 概要	68
6.3.2 目的	69
6.3.3 方法	69
6.3.4 結果	70
6.4 Auto Mode Type Cycle を用いた認証方式の評価実験	73
6.4.1 概要	73
6.4.2 目的	74
6.4.3 方法	74
6.4.4 結果	75
第 7 章 考察	80
7.1 安全性に関する考察	80
7.2 覚えやすさに関する考察	81
7.3 使用継続性に関する考察	82
7.4 他環境における応用に関する考察	82

7.5 今後の課題	83
7.5.1 仕組み	83
7.5.2 実装	84
7.5.3 実験	84
第 8 章 結論	86
謝辞	88
参考文献	89
付録 A 実装に関する付録	93
A.1 実装の詳細	93
A.2 実装コード	94
A.3 画面一覧	94
付録 B 実験に関する付録	96
B.1 スケジュール番号	96
B.2 結果送信の詳細手順	97
B.3 評価実験の概要説明資料	98
B.4 Notifauth 操作マニュアル	100
B.5 評価実験における中間アンケート	109
B.6 評価実験における最終アンケート	114

図 目 次

2.1	Google における ID とパスワードの入力画面	14
2.2	Apple iOS におけるタッチパネルによる PIN の入力画面	15
2.3	Android におけるタッチパネルによるパターンの入力画面	16
2.4	USB キーの例	17
2.5	静脈を用いた認証のための装置	18
2.6	PC , 携帯電話 , スマートフォン , タブレットの年齢層別機器所有率 (IDC Japan の調査結果 [13] から引用)	22
3.1	Google の多要素認証における概要図	26
3.2	Google Authenticator のワンタイムパスワード表示画面	28
3.3	PassBoard の各種設定画面	29
3.4	Authy のトークン表示画面	30
3.5	Authy を用いて二要素認証化した SSH 接続画面	31
3.6	Authy を用いて二要素認証化した WordPress のログイン画面	31
3.7	ハードウェアトークンの例	32
3.8	トークン生成アプリケーションの例	32
4.1	電子メールを用いた認証のシステム	38
4.2	Twitter の Direct Message を用いた認証のシステム	39
4.3	Facebook における友人の顔写真を用いた認証画面	40
5.1	Twitter における公開範囲の概略図	45
5.2	Twitter における Timeline 画面	46

5.3 Twitter の設定画面における公開範囲の設定項目	47
5.4 Notifauth のシステム概略図	49
5.5 Auto Mode Type Term の概略図	50
5.6 Auto Mode Type Term の設定画面	52
5.7 Auto Mode Type Cycle の概略図	53
5.8 Auto Mode Type Cycle の設定画面	54
5.9 Manual Mode の設定画面	56
5.10 ロック画面上における通知の選択(スライド)動作の例	57
5.11 左:ロック画面上における通知の表示画面を模した認証画面, 右:ロック画面上における PIN の入力画面を模した認証画面	58
6.1 実験スケジュール	61
6.2 被験者の特性(左:性別, 中央:年齢, 右:1日あたりのツイート数)	63
6.3 Manual Mode と PIN Mode における設定時からの経過日数ごとの認証成功率	68
6.4 Manual Mode と PIN Mode における設定時からの経過日数ごとの認証時間	69
6.5 Auto Mode Type Term と PIN Mode における設定時からの経過日数ごとの認証成功率	73
6.6 Auto Mode Type Term と PIN Mode における設定時からの経過日数ごとの認証時間	74
6.7 Auto Mode Type Cycle と PIN Mode における設定時からの経過日数ごとの認証成功率	77
6.8 Auto Mode Type Cycle と PIN Mode における設定時からの経過日数ごとの認証時間	78

A.1 Notifauth のクラス図	93
A.2 Notifauth 起動時の画面	95
A.3 Notifauth ユーザ登録画面	95
A.4 Notifauth 設定時の PIN 登録画面	95
A.5 Notifauth 認証終了時の画面	95

表 目 次

5.1 留意事項	59
5.2 開発環境	59
6.1 Manual Mode における各経過日数ごとの認証成功率と認証時間の変化 ($n = 51$)	66
6.2 PIN Mode における各経過日数ごとの認証成功率と認証時間の変化 ($n = 51$)	67
6.3 被験者による Manual Mode に対するアンケート内評価	70
6.4 被験者による PIN Mode に対するアンケート内評価	71
6.5 Auto Mode Type Term における各経過日数ごとの認証成功率と認証時間の変化 ($n = 58$)	72
6.6 被験者による Auto Mode Type Term に対するアンケート内評価	75
6.7 Auto Mode Type Cycle における各経過日数ごとの認証成功率と認証時間の変化 ($n = 58$)	76
6.8 被験者による Auto Mode Type Cycle に対するアンケート内評価	79

第 1 章

序論

1.1 背景

通信網の高速化・大容量化、電子機器の小型化・高性能化などにより、Web サービスで可能なことが多くなった。また、高性能な携帯端末の普及により、個人や決済にかかわる重要な情報を持ち歩くことが一般化しつつあり、必然的に個人認証を行う場面が増えてきている。こういった場面における個人認証では、パスワードや暗証番号^{*1}(英語では Personal Identification Number (略称: PIN))を用いた例をよく見かける。

特にパスワードを用いた認証では、安全性と記憶持続性・利便性に関してはトレードオフの関係が存在する。例えば、辞書攻撃に強い安全なパスワードを用いようとする際には、意味のない文字列にすることが望ましい。しかし、意味のない文字列というのは覚えることが難しく、ユーザがパスワードを他のサービスにおいても使い回してしまう可能性が高まり、どれか一つのサービスからパスワードが流出した際、かえって脆弱になってしまふ恐れがある。現在、こういった問題を防ぐものとして、多要素認証を自由意志で利用できる Web サービス (Google[1]、

^{*1} 本論文において暗証番号認証は、特に指定がない限り 4 枚の数字を秘密情報としたものを想定する。

Dropbox[2] や Evernote[3] など) が増加しつつある。例えば、パスワードの入力が完了し、それが正しいものだと判断された後に、あらかじめ登録された電話番号に Short Message Service^{*2}(以下、SMS) を利用してワンタイムパスワードを送信し、それを入力させるといった方式をとることができる。これにより、覗き見、推測や総当たり攻撃によってパスワードが漏洩した際の不正利用のリスクを減少させることができが可能となる。多要素認証を何らかの方法で適用する行為を個人認証の多要素化と定義する。

また、Social Networking Service^{*3}(以下、SNS) の形態を持つ Web サービスが近年増えてきている。これにより、コミュニケーションの道具やライログとして自分自身の情報を公開することが多くのユーザ間で一般的になりつつある。SNSにおいては、公開範囲をある程度任意に指定できるサービスが多いという特徴がある。

1.2 研究目的

本研究における目的は、SNS の情報を用いて記憶持続性と利便性に考慮しつつ個人認証の安全性を向上させることである。現在行われている個人認証の多要素化は、セキュリティトークンや E メールを用いたものが一般的であり、それにより大きく認証の安全性を高めている。しかし、利便性という点においては、一度認証のための画面から目を逸らす必要がある、特別なハードウェアを持ち歩く必要があるなど、今後の普及に際して改善の余地があると考えられる。

本研究では SNS の情報を用いた個人認証の提案が少ないことに着目し、応用可能な例として携帯端末に搭載することを想定したシステムを考案した。

^{*2}電話番号を利用して短いメッセージを送受信できるサービス

^{*3}社会的ネットワークをインターネット上で構築するサービス。

1.3 論文の構成

本論文は以下の章により構成される .

第 1 章 序論 : この章では , 本研究を行うに至った背景と主たる目的に関する解説を行う .

第 2 章 個人認証の多要素化への流れ : この章では , 認証技術の現状や , 個人認証へ及ぼすと考えられる影響について述べる .

第 3 章 関連研究/製品 : この章では , 前章で述べた内容に関連する , 既存の製品や研究の取り組みを紹介する .

第 4 章 動機と提案 : この章では , 既存の認証における問題点と , それを改善するために近年普及した技術やサービスを用いる理由と既存手法 , 更に提案手法の概要について説明する .

第 5 章 Twitter 上の情報を用いた認証システム : この章では , 本研究で開発したシステムに関する原理と詳細説明を行う .

第 6 章 検証実験 : この章では , 本研究で開発したシステムを用いた実験についての内容と結果の説明を行う .

第 7 章 考察 : この章では , これまでの取り組みと得られた結果から , 本研究の成果と各結果に対する考察 , ならびに今後の課題について考察する .

第 8 章 結論 : この章で本研究について総括する .

第 2 章

個人認証の多要素化への流れ

2.1 既存の認証技術

一般に認証手法は以下の 3 つに大別できる .

- 知識認証
- 所有物認証
- 生体認証

これらの詳細は , 以降の小節で述べる .

2.1.1 知識認証

本人のみが記憶している情報を秘密情報として認証を行う手法 . 主にキーボードやタッチパネルなどの入力インターフェースを用いてアウトプットを行う . この手法は他の認証方式と比較して以下のようなメリットから , 一般の Web サービスやモバイル端末などにおける認証に多く普及している .

- 多くの端末に搭載される汎用的な入力インターフェースを利用できるため , 実装される環境への依存が少ない

- 新たなハードウェアを必要とする場面がないため、低コストで導入できる
- 秘密情報の伝達や保管が容易

秘密情報として、パスワード（図 2.1）や PIN（図 2.2）が用いられることが多い。また、Google 社が開発した携帯端末向けプラットフォームである Android では、 3×3 の点を自由になぞるパターン 2.3 を秘密情報にした認証も存在する。

この認証手法には、以下のような欠点が存在する。

- 秘密情報を記憶保持する必要がある
- 認証のための秘密情報入力に際して身体的負担がある
- 情報量が少なく、総当たり攻撃や辞書攻撃に対して脆弱

推測が難しいパスワードにするには意味を持たせないほうがよいため、記憶するのが難しくなりがちである。しかし、ユーザにそういったパスワードを使用させることは難しく、Rockyou.com^{*1} から大規模漏洩したパスワードの解析を行った Imperva の調査 [4] によれば、ユーザの 50% は氏名やスラングの単語、辞書に載っている単語、平凡なパスワード（連続した数字やキーボードの隣接した文字の組み合わせ等）を使用し、パスワードの 20% がわずか 5000 個のリストで網羅可能であることが明らかになった。

2.1.2 所有物認証

本人のみが所有している物の情報を秘密情報として認証を行う手法。

他の認証手法に対して、

^{*1} ゲーム開発会社である RockYou 社の Web サイト



図 2.1: Google における ID とパスワードの入力画面

- トークンの入力を行わない方式に関しては、入力を行うことのユーザの身体的負担が少ない

- 所有物を交換することで秘密情報を容易に変更可能

- 暗号化方式を変更することで秘密情報の情報量を増やしやすいため、比較的容易に安全性を高められる

- 貸与が可能

などの利点がある。しかしながら、

- 認証の際に手元にあることが求められるため、ユーザが管理するための負担は大きい



図 2.2: Apple iOS におけるタッチパネルによる PIN の入力画面

- 秘密情報の保持や検証に新たな機器を必要とするため，導入のコストが高い
- 盗難・紛失した場合，容易になりすましされる恐れがある

といった欠点も抱えている。

この認証方式の具体例として，物理的なカギ，ID カードや USB キー（図 2.4），ハードウェアトークンを用いたワンタイムパスワードによる認証などが挙げられる。

2.1.3 生体認証

本人の生体情報を秘密情報として認証を行う手法。

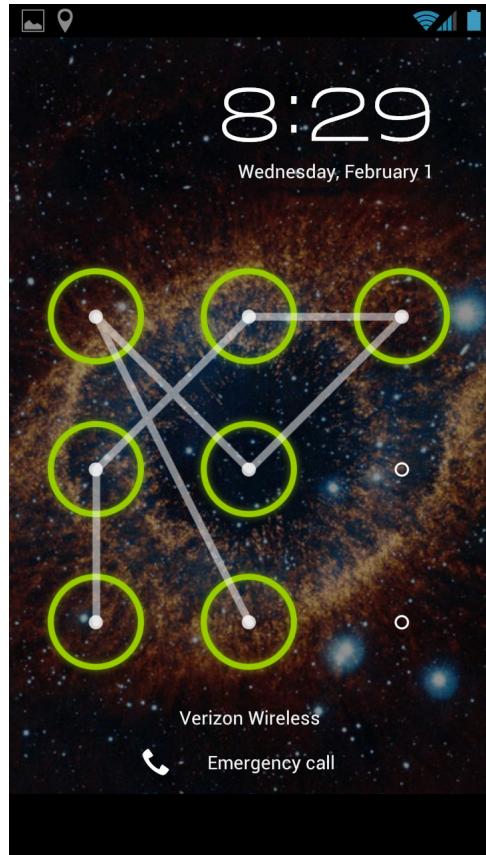


図 2.3: Android におけるタッチパネルによるパターンの入力画面

- 所有物認証のように何かを持ち歩く必要がなく、盗難・紛失の恐れも少ないため、ユーザへの管理負担が少ない
- 入力においてユーザの負担が少ない
- 秘密情報の情報量が大きい

などの利点を持つ反面、

- 秘密情報の変更が困難



図 2.4: USB キーの例

- 身体の情報をスキャンするための特殊な機器を必要とするため，導入のコストが高い
- 身体の状態(例：指の怪我，コンタクトレンズの着用)や外部からの影響(例：光による明暗，騒音)により認証操作を行うことが困難な場合がある

などの欠点が存在する [5] .

この認証方式の具体例として，指紋，静脈(図 2.5)，虹彩を用いたものが挙げられる .

2.2 多要素認証

2.2.1 概要

既存の認証手法を複数組み合わせることで，安全性を高めることができる . これが多要素認証である .

個人認証の多要素化の実現においては，ワンタイムパスワードを要素の一つと



図 2.5: 静脈を用いた認証のための装置

して利用している方式が主流である [6]。ワンタイムパスワードとは、1度しか利用できないパスワードのことで、事前に手に入れるもしくは認証の際にいくつかの手法により生成するといった方法で使用する。ワンタイムパスワードの生成手法は複数あり、

- 数学的アルゴリズムを用いるもの：一方向性関数に初期シードを与えることで動作、パスワードを生成させる手法
- 時刻同期によるもの：認証サーバの時計と同期させ、その時刻に基づいてパスワードを生成する手法
- トランザクション認証番号を用いるもの：ランダム生成されたパスワードのリストを用意し、それを消費してゆく手法

などが一般的である。

2.2.2 代表的手法

銀行(例: ジャパンネット銀行[7]) やオンラインゲーム(例: Battle.net[8]) などで多く見られる[6][9]のが、ハードウェアトークンと呼ばれる、ワンタイムパスワード生成器を用いた方式である。

さらに近年、Google や Facebook, Apple などの Web サービスでは、パスワードを保持するデータベースの増加とその認証情報の流出による、パスワードリスト型攻撃へのリスクを緩和するために多要素認証を用意している[10][11]。そういういったサービスで利用される方式として、SMS/Eメールやスマートフォン^{*2}用アプリケーションを用いたものがある。SMS/Eメールを用いた際は、手持ちの携帯端末にワンタイムパスワードが記載されたメッセージが送信され、アプリケーションを用いた場合は、アプリケーション上で生成されたワンタイムパスワードが表示される。この方式のメリットとして、新たな専用ハードウェアを持ち歩く必要がなくなることによる利便性の向上と、併せて紛失の危険性も減少するということが挙げられる。

多要素認証に関する既存研究や、具体的な応用例は第3章で述べる。

2.2.3 利点と欠点

多要素認証を導入した際の利点としては、何よりも安全性の向上が大きい。現在多くの多要素認証で導入されているワンタイムパスワードの生成もしくは受信

^{*2} インターネットの利用を前提とした高機能携帯電話。統一された定義はないが、一般社団法人情報通信ネットワーク産業協会によれば「携帯電話・PHS に携帯情報端末(PDA)を融合させた端末で、音声通話機能・ウェブ閲覧機能を有し、仕様が公開された OS を搭載し、利用者が自由にアプリケーションソフトを追加して機能拡張やカスタマイズが可能な製品。」(出展: 通信機器中期需要予測 2010 年度 CIAJ)

を行う方式では、6桁の数字が出力され、それをIDとパスワードに併せて入力する。ここで、従来のIDとパスワードによる認証を1要素目、ワンタイムパスワードを用いた方式を2要素目とする。仮に1要素目が何らかの攻撃手法により突破されたとしても、2要素目が突破できなければ、その場ですぐにアカウントを不正利用されるなどの被害を受けることはない。

欠点として、第一に手間が増えることが挙げられる。具体的には、認証の際にワンタイムパスワードを確認するためにハードウェアトークンや携帯端末を確認したり、認証操作を要素数の数だけ行わなければならない。また、「手間」というユーザが負担するコスト以外にも、新たな機器を導入するなどのコストはサービスプロバイダ側も負担しなければならない。更に、生体認証などユーザの管理負担が少ない認証方式を多要素認証に用いたとしても、その認証方式固有の欠点、例えば生体認証の場合であれば、周囲の環境によってエラー率が増加するといった点は解消できない。サービスプロバイダとユーザ両方に対してかかるコストの増加や、利用可能な状況が限られてしまうといった問題については第4.1節にて詳しく述べる。

2.2.4 既知の攻撃方法による脆弱性

多要素認証は、以下の様な攻撃手法に対して脆弱であることがSchneier[12]によつて指摘されている。

中間者攻撃 通信を行う二者の間に割り込んで、両者が交換する公開情報を自分のものとすりかえることにより、気付かれることなく盗聴したり、通信内容に介入したりする手法。例えば、偽の銀行サイトを作成した後にユーザを誘導し、そこでユーザが入力したIDとパスワードを即座に正式な銀行サイトに入力、更にワンタイムパスワードも同様の手法を用いることで、容易に不正アクセスが可能となる。

トロイの木馬を用いた攻撃 正体を偽ってコンピュータへ侵入し，データ消去やファイルの外部流出，他のコンピュータの攻撃などの破壊活動を行うプログラムを用いた手法．例えば，ユーザが正式な銀行サイトへログインした際に，トロイの木馬経由でセッションを奪い，第三者の口座への送金など，任意の操作を行うことが可能となる．

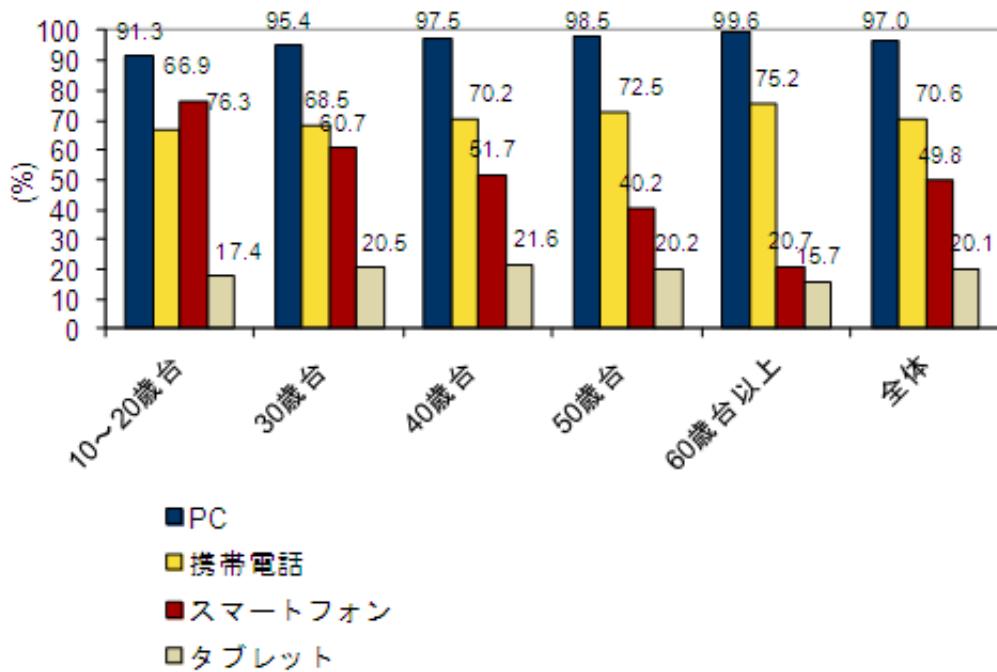
フィッシング攻撃 中間者攻撃でも用いられている，正規のメールを装い偽サイトへユーザを誘導し，秘密情報を獲得する手法．

2.3 スマートフォン/タブレットの普及

2013年6月に行われたIDC Japanの調査[13]によれば，家庭市場におけるスマートフォンの所有率は49.8%，タブレット^{*3}の所有率は20.1%であった(図2.6)．これらの携帯端末の普及により，外出先などからも様々なサービスにアクセスすることが可能になった．しかしその反面，様々なサービスの認証情報や個人情報などのデータを外に持ち出している状態であるため，携帯端末のセキュリティをいかに強化するかが重要になってきている．

第2.2節や第3.2節で述べられているように，携帯端末は近年の普及により，多要素認証における認証要素の一つとして扱われるようになり，サービスプロバイダが従来よりも手軽に認証の多要素化を導入できるようになった．

^{*3}板状のオールインワン・コンピュータやコンピュータ周辺機器の総称．本論文では，特に断りがなければ携帯端末としてのタブレットを指す．



n = 1,136(10~20歳台)、n = 3,758(30歳台)、n = 5,421(40歳台)、n = 3,595(50歳台)、n = 1,583(60歳台以上)、
n = 15,493(全体)

図 2.6: PC , 携帯電話 , スマートフォン , タブレットの年齢層別機器所有率 (IDC Japan の調査結果 [13] から引用)

第3章

関連研究/製品

3.1 多要素認証についての調査

3.1.1 二要素認証のユーザビリティに関する比較調査

Honglu ら [6] は、多要素認証の中でも二要素認証に着目し、主要な二要素認証手法の洗い出しと、それらのユーザビリティの評価を行った。まず最初に予備実験として9人にインタビュー形式で「いつ・どこで・なぜ・どのように二要素認証が使われるのか」を調べた。その結果、よく普及している二要素認証として

- (ハードウェア) セキュリティトークンにより生成
- EメールもしくはSMSを用いて受信
- 専用のアプリケーションを用いて生成

して得られたコードを、ユーザIDとパスワードに加え入力する方式であるという結果を得た。また、その際のアンケートによる調査では、銀行や勤務している会社から強制的にセキュリティトークンを利用させられることや、SMSの送受信がうまく行われずに支払いが遅れることなどへ不満を持つ人の意見も得られた。

次に、予備実験で得られた普及している多要素認証について、(1)それを使用したことのある状況(金融・仕事・個人)と動機(任意・誘因・強制)、(2)15種類にわたる項目のリッカート尺度を用いた評価、についてオンライン調査を用いて219人から回答を得て、(2)に関しては、

使いやすさ 楽しい、便利、簡単、再利用、など

信頼性 セキュア、信頼できる

認識努力の必要度 いらいらする、指導が必要、集中が必要、など

の3つの基準に分類し最終的なスコアを算出した。その結果、(1)については

- 会社などの環境では、ハードウェアのトークンが好まれる、
- 金融や個人による使用では、EメールもしくはSMSを用いた二要素認証方式が多く用いられている
- 既にハードウェアのトークンを利用しているユーザが携帯端末のアプリケーションによる二要素認証に移行することはほとんどない
- 携帯端末のアプリケーションによる二要素認証はごく最近の技術だがセキュリティトークンよりも高い採用率を誇る

という結果が、(2)についてはどの二要素認証技術も高いユーザビリティ評価を獲得しているという結果が明らかになった。様々な相関を調べた結果、どの二要素認証が好まれるかは個人の特徴(年齢や性別)に左右されることが大きく、ターゲットとなるユーザを絞った導入を行わなければならないとした。また、二要素認証同士の比較であれば、安全性と利便性は逆の相関を持たない(安全性が高まれば必ずしも利便性が損なわれるわけではない)ことも明らかにし、先行研究と逆の結果と

なったのは、それらの大部分がパスワードとの比較によるものであったためであると結論づけた。加えて、本実験でのアンケートによる自由回答では、個人認証のプロセスが以下のような点を持つと、ユーザの不満が高まるとした。

失敗しやすい 例：“認証サーバが停止してしまっていた”

厳しすぎる 例：“認証に失敗すると入力を最初からやり直さなければならなかつた”

複雑 例：“3回間違えるとPINのリセットのためにヘルプデスクに連絡しなければならなかつた”

3.2 認証の多要素化手法

3.2.1 Google

Googleでは、アカウントにログインするための認証の多要素化方法を実装している。図3.1にある通り、従来のID/パスワードによる認証方式に加え、以下に紹介する4つの方式のうちいずれか一つを組み合わせた二要素認証をサポートしており、いずれの方式もログインの際にID/パスワードの入力が正しいものであれば入力が可能となる。

SMSを用いてワンタイムパスワードを送信する方式

SMSを用いて、事前に登録された電話番号によりユーザの持っている携帯端末へワンタイムパスワードが送信される。

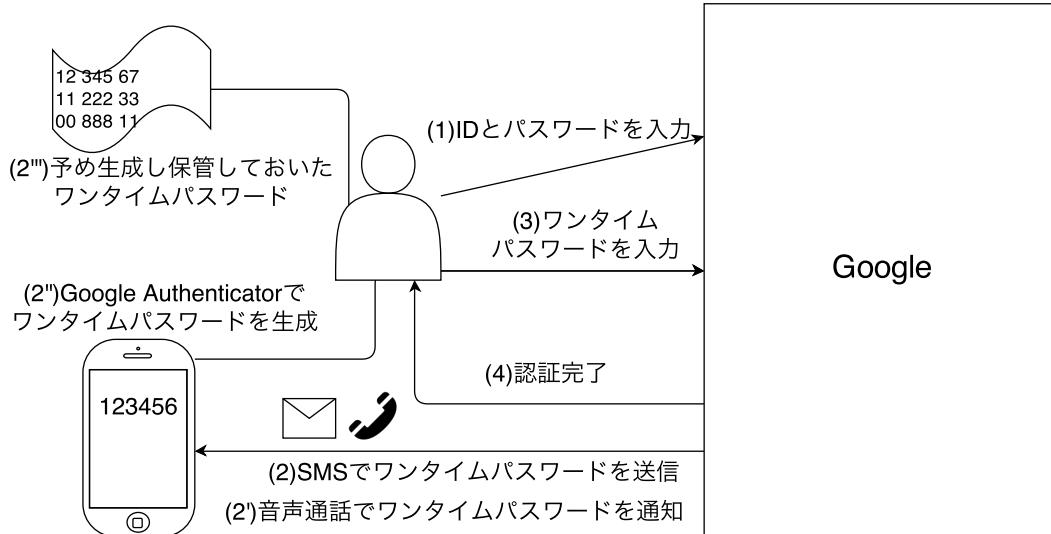


図 3.1: Google の多要素認証における概要図

音声通話によりワンタイムパスワードを確認する方式

SMS ではなく電話を利用して、機械音声でワンタイムパスワードを確認するこ
とが可能となっている。

携帯端末向けアプリケーションでワンタイムパスワードを生成する方式

“Google Authenticator”[14] と呼ばれる携帯端末向けのワンタイムパスワード生
成アプリケーション(図 3.2)を公開しており、実装されている以下の 2 種類のワ
ンタイムパスワード生成アルゴリズムは、どちらも Web の画面に表示された 16
文字の base32 文字列の入力もしくは QR コードを読み込むことで渡されるユーザ
固有の秘密鍵と、特定の変数により SHA1^{*1} を用いた HMAC(Hash-based Message

^{*1} アメリカ国家安全保障局によって設計されたハッシュ関数の一つ。SHA は Secure Hash Algorithm の略

Authentication Code^{*2}) を生成し，6 行の数字コードに変換するものであるが，そこ用いられる変数が異なっている。

HOTP(HMAC-based One-Time Password) 前述の“数学的アルゴリズムを用いる”生成手法であり，ボタンをタップすることで 1 つのワンタイムパスワードが生成されるが，その生成回数を変数として利用し生成する [15]

TOTP(Time-based One-Time Password) 前述の“時間同期による”，HOTP を拡張した手法であり，サーバからのメッセージを用いて 30 秒ごとに変数を決定し生成する(この方式を用いた場合，30 秒毎にワンタイムパスワードは更新される)[16]

なお，いずれの手法も HOTP であれば回数が，TOTP であれば時刻が，クライアント側とサーバ側で同期している必要が存在する。ちなみに，上記の 2 アルゴリズムのいずれかを実装し，更にユーザ固有の秘密鍵を出力できる Web サービスならばこのアプリケーションを用いた認証の二要素化が可能となっている。

バックアップコードを予め保存しておく方式

SMS や音声通話の受信が難しい場合，もしくは Google Authenticator を使えない場合にテキストファイルで 7 行のワンタイムパスワードを出力する機能も実装されており，それらは 10 個を 1 セットとして出力され，1 つにつき 1 回のみ利用できる。また，再発行も可能となっている。

^{*2}暗号ハッシュ関数に基づいたメッセージ認証符号。秘密鍵とメッセージとハッシュ関数により計算される。

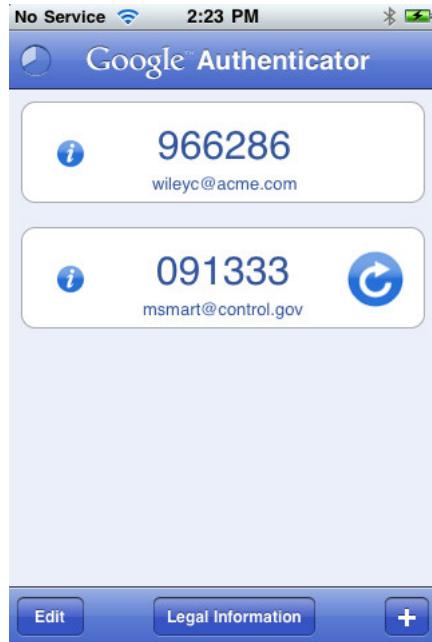


図 3.2: Google Authenticator のワンタイムパスワード表示画面

3.2.2 PassBoard

PassBoard^{*3} というアプリケーションソフトウェア [17] は、スマートフォン上にある各アプリケーションにアクセスする際の認証機能を提供している。このアプリケーションでは、パスワード認証や音声認証、GPS 認証、顔認証などを組み合わせて多要素化が可能となっており、更に認証時の周囲の環境（明るさや騒音）に合わせて、使用する認証方式を自動で設定する機能を持っている。図 3.3 左は、使用する認証方式の画面であり、このようなリストの中からいくつでも組み合わせて使用することができる。図 3.3 右は、認証を付与するアプリケーションに対する個別なセキュリティレベルの設定画面であり、そのソフトウェアの重要度に応じて認証の要素数を変えることができる。

^{*3} 米 PassBan 社により提供

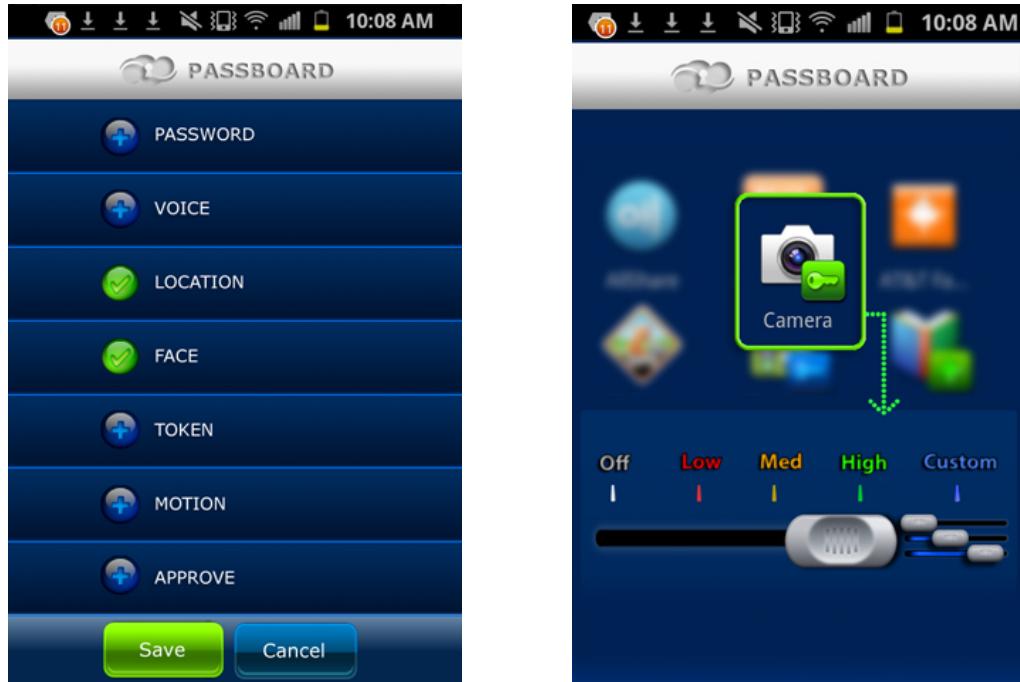


図 3.3: PassBoard の各種設定画面

3.2.3 Authy

Authy[18] というアプリケーションソフトウェアを用いると，Google や Dropbox などの二要素認証に対応しているサービスだけでなく，SSH^{*4}接続や個人のサーバにインストールした WordPress^{*5}へのログインも二要素化が可能となる。Authy に紐付けた Web サービスもしくは WordPress ログインする際は，通常の手順に加え，SMS によって送信されるもしくは Authy のアプリケーション内に表示されているアクセストークン(図 3.4)を入力する(図 3.5)ことで，ログインが完了する。SSH 接続の認証は，ssh コマンドの設定項目に関連付けることで Authy のプラグインを起動させる。この場合も同様に，前述の Authy に紐付けた Web サービスと同じ手順で手に入れたアクセストークンを図 3.5 のようにコマンドラインインター

^{*4}Secure SHell

^{*5}オープンソースのブログソフトウェア

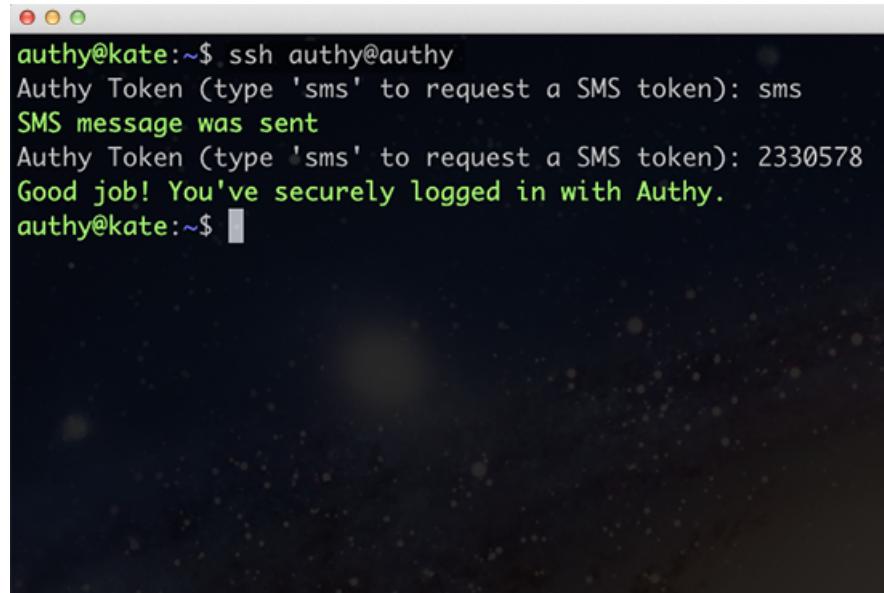
フェイス上で入力することで完了する。



図 3.4: Authy のトークン表示画面

3.2.4 オンラインゲームにおける多要素化例

オンラインゲームにおいては、図 3.7 のようなハードウェアトークンによる認証の多要素化が普及している [6]。2004 年にゲームの限定パッケージにハードウェアトークンが付属した [9] ことがきっかけで現在でも多くのオンラインゲームに二要素認証が導入されている。これらのハードウェアトークンの多くは Google の例 (項) と同様、時刻同期によるワンタイムパスワード生成を行っており、小型の液晶画面にワンタイムパスワードが表示される。ゲームの利用者はゲームにログインする際に、ユーザ ID とパスワードに加えて、ハードウェアトークンに表示されて



```
authy@kate:~$ ssh authy@authy
Authy Token (type 'sms' to request a SMS token): sms
SMS message was sent
Authy Token (type 'sms' to request a SMS token): 2330578
Good job! You've securely logged in with Authy.
authy@kate:~$
```

図 3.5: Authy を用いて二要素認証化した SSH 接続画面

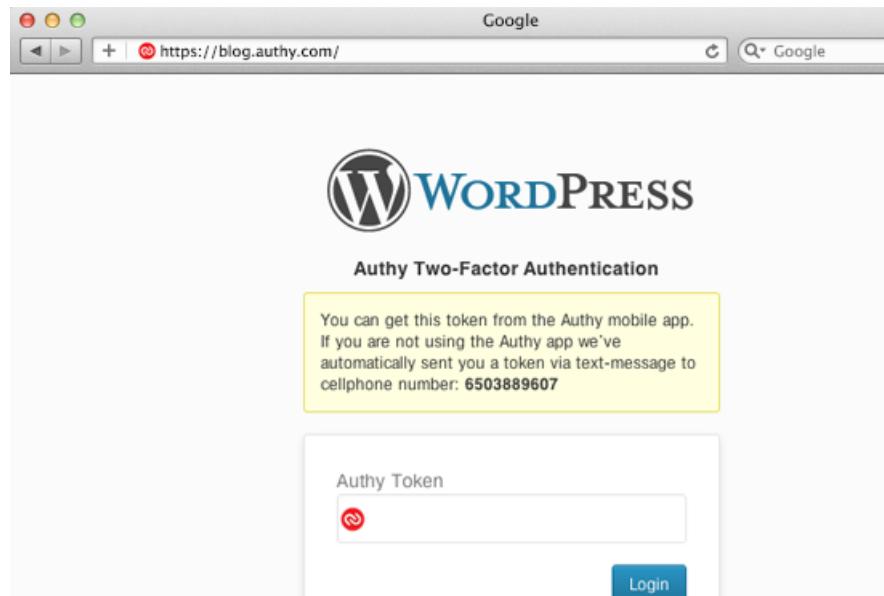


図 3.6: Authy を用いて二要素認証化した WordPress のログイン画面

いるワンタイムパスワードを入力することで、認証が完了し、ゲームをプレイ可能になる。また近年では、他の Web サービスと同様に携帯端末向けの専用トークン生成アプリケーションソフトウェア(図 3.8)が用意されていることもある。



図 3.7: ハードウェアトークンの例

図 3.8: トークン生成アプリケーションの例

第 4 章

動機と提案

4.1 動機

ここまでこの章で多要素認証の現状について述べてきたが、今後の普及に向けて解決しなければならない問題点：(1) コストと(2) 利用可能な状況、がある。

(1) コストについては2つの見方がある。1つはサービス提供側が負担しなければならないコストであり、もう1つは利用者が負担しなければならないコストである。そのそれぞれについて以下に述べる。

- サービスプロバイダは、多要素認証の導入のために新たなハードウェアトークンや認証用機器、新たなシステムを用意する負担を強いられる
- ユーザは、ハードウェアトークンを管理・携帯したり、認証を行う際に携帯端末の画面を確認しなければならないといった負担を強いられる

上記のように、現状では双方にとってあまり手軽とは言えない。そのため、導入を妨げないようなシステムを提案することが普及の鍵になると考えた。

また、(2) 利用可能な状況に関しても、ワンタイムパスワードのSMS/Eメールを用いた送信や携帯端末を用いた生成は、ネットワークに接続していて操作の権限を持つ端末が必要であるし、そもそも個人で使える多要素認証はWebに関わる

ものが多く、そうではない場面、例えば携帯端末そのものの認証やオフラインな状況でも使える多要素化の方法を模索する必要があると考えた。

4.1.1 携帯端末への多要素認証の導入

スマートフォン/タブレットでは、携帯端末専用又はタッチパネルなどによる操作に特化したOS^{*1}が搭載されていることが多く、それらは高機能な開発環境が公開されている。そのため、Webサービスなどにおいても、専用のアプリケーションソフトウェアがサービスプロバイダによって用意され、ブラウザ上からアクセスする必要がなくなりつつある。そういうった場面では、認証情報は端末内に保存され、毎回の個人認証操作を行う必要が省かれていることもあり、端末の画面ロック^{*2}が解除されてしまえば、従来の携帯電話などと比較して多くの操作が可能になってしまう。以上の理由から、携帯端末のセキュリティを向上させることが必要であり、その際に多要素認証を適用できるのではないかと考えた。

更に、それぞれのアプリケーション(例:写真アルバム、メモ、ブラウザなど)に対して、プライバシーを保護するためにロックをかけたいといった需要が存在し、そのためのソフトウェアも既に開発されている(3.2.2など)。そういうった場面で気軽に安全性を強化できる新たなアイデアとしても提案できないかと考えた。

4.2 提案

本論文では、個人認証における利便性を

^{*1} Operating System、基本ソフトとも。ハードウェアを抽象化しインターフェースを提供するソフトウェア

^{*2} 操作を大きく制限されている状態。PIN認証などを行わない限り解除できないことが一般的である。

覚えやすさ 秘密情報を覚えて認証を行うまでの期間記憶保持することの肉体的・精神的・時間的負担の少なさ

使いやすさ 認証及び秘密情報設定の際の肉体的・精神的・時間的負担の少なさ

の2指標で定義する。個人認証の方法を提案するにあたって、認証の強度を高めることによって利便性を損ねてしまうことは避けなければならない。そこでライフログ^{*3}は個人の生活や行動、体験などに基づいているため、個人を特定できる要素が多く、しかも記憶持続性が高いという想定から、個人認証と親和性が高いのではないかと考えた。

また、写真、動画、音楽などの共有(Instagram)や買い物(Amazon.co.jp)などWebサービスで行えることが増えてきているが、その中でもSNSは2011年の総務省の調査[19]では52.9%が1回以上利用したことがあるとされており、若年層ほど利用率が高く、10代と20代ではそれぞれ71.7%と63.9%が現在継続して使用しているという結果が明らかになった。SNS上の情報は、全世界に公開されるパブリックなものから友人のみが閲覧可能な情報や、自分のみが見ることができるプライベートな情報まで、様々な公開範囲を定めて発信できるという特徴を持つ。この特徴は、秘密情報の候補として認証時に表示してもよいものが得られるため、情報漏洩などの被害を抑えることができるのではないかと仮定した。

これらの技術を用いることで、強度と利便性を兼ね備えた認証を提案できないかと考えた。

4.2.1 ライフログやSNSを利用した個人認証事例

ライフログやWebサービスを用いた認証について、5つの既存手法を紹介する。

^{*3}人間の行いをデジタルデータとして記録する技術・行為。ブログやSNSの一部などもライフログだといえる。

Web 履歴を用いた認証

田村ら [20] は、Web に頻繁に接続するユーザである場合、閲覧履歴を用いて“平日の平均 Web 接続時間”，“平日、休日のアクセストドメイン”によってユーザの特徴を抽出できる可能性があるとした。その際は本人認証を Web 閲覧履歴のみによって行えるが、Web に頻繁に接続しないユーザの場合は、ユーザを識別できるほどの特徴が見いだせないという結果が得られている。また、複数のライログルを用いた多要素化についても述べられている。問題点として、本人の趣味趣向を真似ることによってなりすましが行いやすいことが挙げられる。

GPS を用いた認証

長谷ら [21] は、ユーザがあらかじめ予定していた時間に、予定していた場所へ移動したかどうかの情報を個人認証のための特徴量として扱う検討を行った。これによれば、複数のチェックポイントを設け、その場所で送信された GPS データを到着予定場所のものと比較することで、個人認証を行える可能性があるとしたが、GPS データの送信が不可能な場所や、予定時刻へ間に合わない場合が存在するなどの問題点が存在することも示した。

また、今澤ら [22] は、GPS データからユーザが滞在していた場所と時刻の情報を抽出し、ユーザに停留点を回答させる手法で、認証システムを実装した。これによれば、ユーザの 1 週間の停留点数が 10 点以下であった場合に選択肢が減少し安全性が損なわれてしまう可能性があるが、必要操作や依存環境の少なさから様々な場面で応用できるとした。更なる問題点として、GPS のデータを逐一送信できないと認証の安全性が確保しにくくなることが挙げられる。

電子メールを用いた認証

西垣ら [23] は、ユーザの生活履歴を用いて認証を行う手法を提案し、そのプロトタイプとして E メールを用いたシステム（図 4.1）の構築と実験を行った。E メールによる認証は、「最近のメールかどうか」をユーザに回答させるというプロセスで行われた。その際、人間の記憶の曖昧性を取り除くための手法として $(n+1)$ 日前から $(m-1)$ 日前までのメールは認証に使用しないように設定し、最近と過去どちらともいえないような期間のメールを利用しない：例えば $n = 7$, $m = 30$ とすることで「8日前から 29日前までのメールは質問の中に出できません」と明示することでユーザが直感的に回答を行えるようにする、という工夫がなされた。さらに、基礎実験の後に「最近のメール」といった選択肢に加え「曖昧だが最近のメール」といった曖昧な回答の選択肢を追加することで、重要でない故に記憶に残っていないメールを認証に使用しないようにするという改善策をとった結果、最終的に本人による認証では 99% の正答率を得た。問題点として、重要であったりプライベートなメールが認証時に表示されてしまうことで、情報漏洩やプライバシー情報流出の可能性がある。

Twitter の Direct Message を用いた認証

Nemoto ら [24] は、Twitter のダイレクトメッセージ^{*4} (DM) 機能を用いて、定期的に質問を投げかけることでその回答を秘密情報とし、認証を行うシステム、KBA^{*5}（図 4.2）を提案した。質問の内容は「2月 15 日の昼食は？」といった文面で構築され、Twitter のダイレクトメッセージ機能により送信され、回答も同機能を

^{*4} 特定のユーザ宛に、一対一で送信された文章のこと。閲覧可能な人物は、自分と相手のみである。

^{*5} Knowledge-Based Authentication

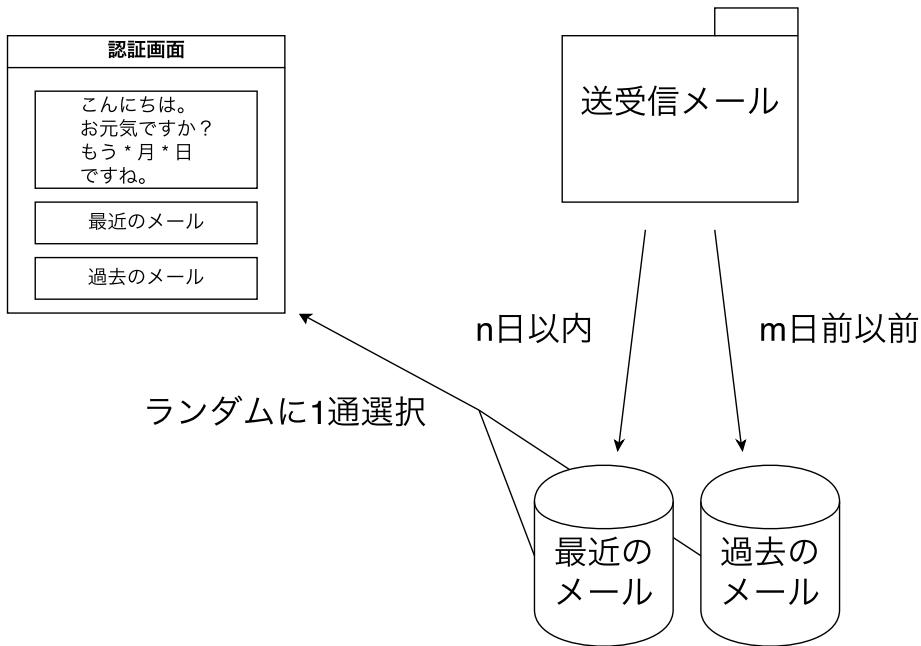


図 4.1: 電子メールを用いた認証のシステム

用いて行う。この手法は、メッセージ機能を用いて秘密の質問を定期的に更新しているだけで、SNS 上でそれを実行する必要性が希薄であると考えられる。

友人の顔写真を用いた認証

Facebook^{*6}では、友人の顔写真を用いた個人認証が運用されている。認証手法は、顔写真が質問として提示され、これに対してその人物の本名を回答する方法である。これはパスワードを忘れてしまった際や、アカウントへの不審なアクセスが確認された場合の本人証明に使われている。Facebook にはユーザから投稿され

^{*6} 米 Facebook 社が提供している SNS である。本名での登録が必須という特徴を持つ。2004 年に学生のみが使用できるサービスであったが、その後一般にも開放され、現在では世界最大のアクセス数を誇る SNS となっている。

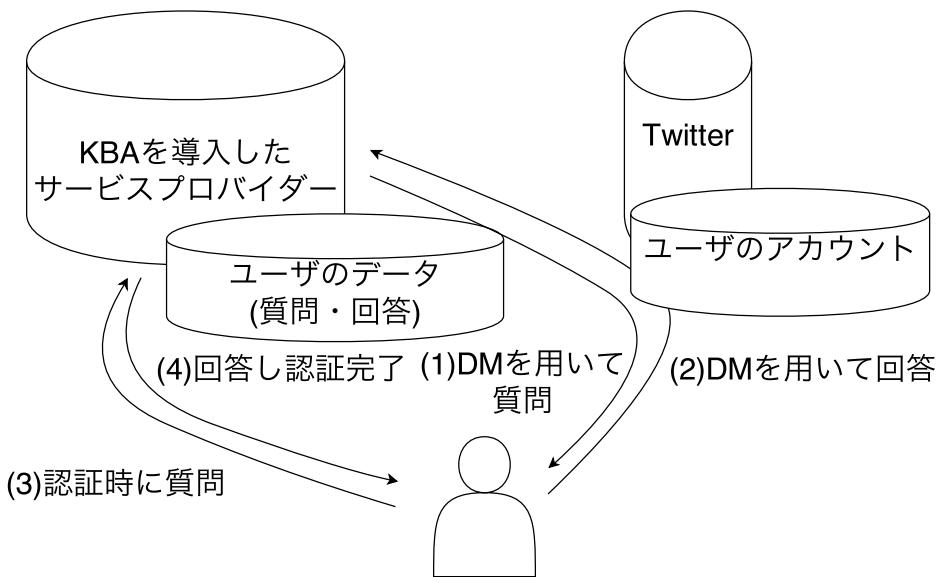


図 4.2: Twitter の Direct Message を用いた認証のシステム

た写真にユーザ名を結びつけることができ、さらに自動で人の顔を抽出しタグ付けを行う機能 [25] も存在するため、そういういった情報を利用していると考えられる。欧州ではプライバシー保護のためこの自動顔認識の機能が無効にされるなどしている。更なる問題点として、友人が自分の顔にのみタグ付けしているという保証がなく（他の動物や物体にも名前のタグ付けが可能）、その場合答えられないという状況が発生し得ることが挙げられる。

4.2.2 提案手法の概要

前節の各既存手法の問題点を、3 つに大別する。

特定の攻撃手法に対して脆弱になりうる状況が存在するもの Web 履歴を用いた認証、GPS を用いた認証



図 4.3: Facebook における友人の顔写真を用いた認証画面

認証時に表示される情報に問題があるもの 電子メールを用いた認証，友人の顔写真を用いた認証

利便性について提案以前の状態から改善できていないもの Twitter の Direct Message を用いた認証

その上で提案手法によって目指すべき点をまとめると以下のようになる．

1. 従来の多要素認証方式に存在した，以下の問題点を解消する
 - (a) 導入に際してかかる金銭的・精神的なコストや負担といった問題点を，既存の入力手法を用いることで解消できる可能性がある
 - (b) ハードウェアやネットワークへの依存し，導入のための状況が限られる

という問題点を、予め保存できる知識情報を用いることで改善できる可能性がある。

2. 能動的に発信した文章を秘密情報として使用することで、従来の知識認証において利用者の負担となっていた記憶負担を低減できる可能性がある
3. 前節で述べた既存の認証方式に存在した、以下の3つの問題点を解消する
 - (a) 特定の趣向や環境に依存してしまうと、一部の攻撃手法に対して脆弱になったり、エントロピーの問題から認証の安全性を担保できないことがあるため、雑多な自分自身の投稿を用いることでそれらの依存を解消できる可能性がある
 - (b) 認証時に表示される情報が、認証とは別に攻撃者に見られては困るものである場合を禦ぐため、既に公開情報となっている自分の投稿を用いることで、安全性の中でも特にプライバシー面での問題を解消する
 - (c) 定期的に秘密情報に関する質問が行われるなど、利便性の面から大きく改善が見られないため、自分が日常的に行っている投稿から秘密情報をつくりだすことで、能動的に覚える作業や定期的な更新の必要性を減らせる可能性がある。

以上の問題点と解決方法によって、従来の方式に比べて

- 利便性
- 安全性

をどちらも損なわず両立させた個人認証手法を目指した。

第 5 章

Twitter 上の情報を用いた認証システム

5.1 概要

本論文における提案システムとして、前章の内容を踏まえ、以下の機能を持つ個人認証手法を実装した（以下 Notifauth）。

- 利便性と安全性を両立させるために、SNS 上に存在する情報を秘密情報として使用する
- 手軽且つ環境へ依存せず導入し安全性を向上させることが可能な多要素認証のモデルケースとして、携帯端末における既存の知識認証に付け加わるように動作する
 - 認証のために新たな操作を覚える負担を考慮し、既存の携帯端末向け OS で既に実装されている画面構成と操作を用いる

5.1.1 Twitter の使用

今回はライログと SNS の両方の特徴を兼ね備えた Web サービスとして、Twitter を選択し、その中でも自分の投稿（ツイート、つぶやきとも呼ばれる。以下ツ

イートと表記する)を秘密情報として利用することで前章で述べた改善策が実現可能になると考えた。積極的理由として、

1. Twitterにおけるツイートは能動的な行為によって生成される情報であり、記憶負担が少なくなる可能性があるため
2. ツイートを書き込んだ日時情報が個々のつぶやきと関連して記憶されている可能性がある。つまり日時情報から特定のつぶやきを想起できる可能性があるため

が挙げられ、他にも考えうる手段としては以下の様なものがあったが、記載の消極的理由により前述の手法をとることにした。

- Twitterのお気に入り情報を用いる手法
 - お気に入りに登録した日時が取得できないため
 - お気に入りに登録したツイートが投稿者により削除される可能性があるため
- Facebookの情報を用いる手法
 - Twitterと違い投稿の文字数制限が緩く、認証時に表示する際に視認性が下がる恐れがあるため
 - マクロミル[26]によると、1日2回以上の頻度で利用するユーザは25.4%であり、更にFacebookの楽しみ方として「自分の近況報告をする」を選択した割合は41.4%と、1日に何度も投稿するユーザはあまり多くない予測されるため

また，ツイートが投稿された日時情報を保持していることにより，時系列上の範囲を指定することで，秘密となる情報群を抽出することができるという特徴を得られると考えた．すなわち，相対的な時間情報の指定を行うことで秘密情報の対象を自動で入れ替えることが可能となる．これによって得られるであろう具体的な利点は第 5.2 節にて示す．

Twitter についての説明

Twitter とは，ユーザが個人で短文(140字以内)を投稿する，ミニブログやマイクロブログといったカテゴリーに分類される SNS である．Twitter では図 5.1 のように“public”と“protected”的 2 つの公開範囲が存在する [27]．Twitter の用語には以下のようなもののが存在する．

ツイート ユーザによる短文の投稿

タイムライン 図 5.2 のように，ツイートを時系列に沿って表示される画面

フォロー 他ユーザの投稿を自分のタイムラインで表示できるよう登録すること

フォロワー 自分のことをフォローしている他のユーザ

ツイートはそれ自体に単独で公開範囲を定めることはできないが，アカウントが“protected”(一般非公開の状態)に設定(図 5.3)されていれば，フォローを許可されたフォロワーのみが閲覧できる状態になる．アカウントが“public”であれば，自分の投稿は他のユーザが自由に閲覧できる．しかし，他人への返信は自分と相手の共通のフォロワーでないとタイムライン上には表示されない．

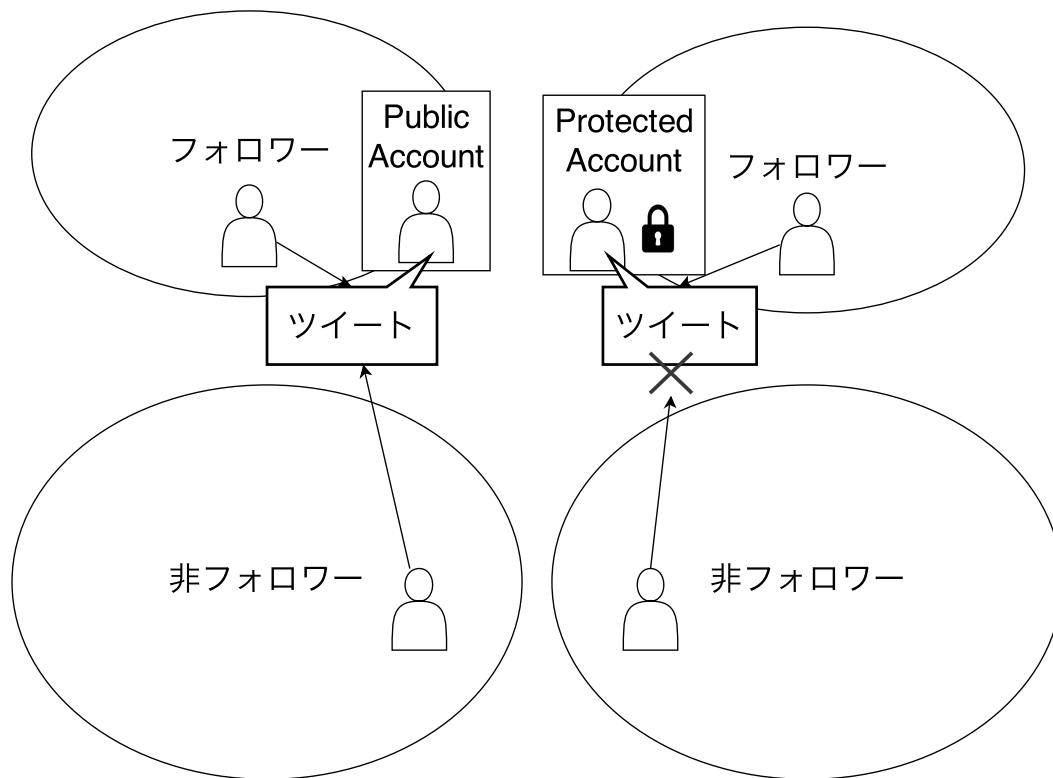


図 5.1: Twitter における公開範囲の概略図

5.1.2 携帯端末への導入

コストや制約の面で手軽な多要素認証として、携帯端末への導入を目指した。具体的には以下のことが理由として挙げられる。

- 携帯端末においては、従来の E メールやトークンを利用した認証の多要素化事例がみられなかったことから、改善の余地があると判断したため
- 携帯端末は持ち歩き様々な環境で使うことが想定され、本認証を利用可能な状況に関して具体的に改善すべき点が得られやすいと予測したため
- 総務省の調査 [28] では、スマートフォンの利用者中、サービス別利用率にお

The screenshot shows a Twitter timeline with the following tweets:

- User 1 (3s ago)**: 夢まるっきり覚えてないというか覚えてた気がするけど結局寝坊と変わらん行動をしてしまったことに対する嫌悪感で覚えていようとしなかった
Expand Reply Retweet Favorite More
- User 2 (5s ago)**: どれや
Expand Reply Retweet Favorite More
- User 3 (9s ago)**: エリア移動6秒は我慢するかなあ...
Expand Reply Retweet Favorite More
- User 4 (11s ago)**: 荘重トマト今まで生きてて一番うまいと思った8800円ばワイン超えるうまさだった
Expand Reply Retweet Favorite More
- User 5 (18s ago)**: FFのドット絵描いてた女性が「ドット絵に大事なのは愛を注ぐ事」とインタビューで仰られてて、素人の僕ですが凄く共感しました。わかるでわかるで素人だけど！
Expand Reply Retweet Favorite More
- User 6 (25s ago)**: 話がシビアになればなるほど握る可能性は高くなるけどホントにそれでいいのか？って判断に悩ましさが伴ってくるよね...
Expand Reply Retweet Favorite More

図 5.2: Twitter における Timeline 画面

いて 54.1% が SNS を利用しており、パーソナルコンピュータでの SNS 利用率の 57.1% を上回っていることから、SNS 情報を利用することとの親和性もあると考えたため

セキュリティとプライバシー
セキュリティとプライバシーの設定を変更します

セキュリティ

ログイン認証 ログインリクエストを利用しない
 携帯電話にログイン認証リクエストを送信
メッセージを送信できませんでした。(こちらの機能は日本国内では利用できません)
 ログイン認証リクエストをTwitterアプリに送信する
設定をしたTwitter for iPhone、またはTwitter for Androidでタップするだけでリクエストの承認ができます。 [詳しい説明](#)

パスワード パスワードのリセットに個人情報を使う
初期設定ではユーザー名を入力するだけでパスワードをリセットできるようになっています。このチェックボックスをチェックすると、パスワードを忘れたときにメールアドレスや電話番号を入力するよう促されます。

プライバシー

ツイートの公開設定 ツイートを非公開にする。
「ツイートを非公開にする」を選択すると、今後のツイートは一般に公開されず、承認したユーザーのみが閲覧できます。非公開設定以前のツイートは、一般に公開されている場合があります。 [詳細はこちら](#)

位置情報をツイート ツイートに位置情報を追加
ツイートに付与された位置情報はTwitterに保存されます。位置情報を付加するかどうかツイートごとに設定できます。 [詳しい説明](#)

[すべての位置情報を削除](#)

過去のツイートからすべての位置情報を削除します。この処理には30分程かかります。

見つけやすさ 他のユーザーがメールアドレスから検索可能にする

図 5.3: Twitter の設定画面における公開範囲の設定項目

更に，実装は Apple 社の携帯端末向け組み込みプラットフォームである iOS に向けて行った。その理由としては，Kantar Worldpanel Comtech の調査 [29] によると，スマートフォンプラットフォームの日本国内における iOS のシェアは 66.2% であり，最も普及していると考えられるから，というのが挙げられる。また，認証操作として iOS に実装されているロック画面上の通知とその選択操作（図 5.10^{*1}）を踏襲したものを探用した。理由として，

- 開発環境である iOS 上でロック中に情報の表示や選択といった操作を行えるのは，開発を開始した当初の OS のバージョンではロック画面のみであったため
 - ロック画面で通知をスライドし選択する動作は iOS 標準の機能であり，ユーザへ新たな操作を覚えさせる負担を与えない目的と合致するため
- という点が挙げられる。

5.1.3 実装の概観

システムの概略図は図 5.4 のようになっている。Notifauth では 4 種類の認証方法が用意されており，それぞれの設定はアプリ内の保存領域に保存されるほか，ツイートを用いる認証方法では，Twitter の公式 API^{*2}へとリクエストを送り，レスポンスとして得られたツイートのデータはデータベースに保持される。認証時には保存された設定情報を用いて認証を行う。各認証方式の詳細は第 5.2 節で述べる。

Notifauth 起動時の画面は付録 A.3 - 図 A.2 のようになっており，この画面から

^{*1} この場面ではスライドすることでロック解除後に受信したメールをすぐに読むことができる

^{*2} Application Programming Interface，ソフトウェアの機能やデータなどを外部のプログラムから呼び出して利用するための手順や形式を定めた規約

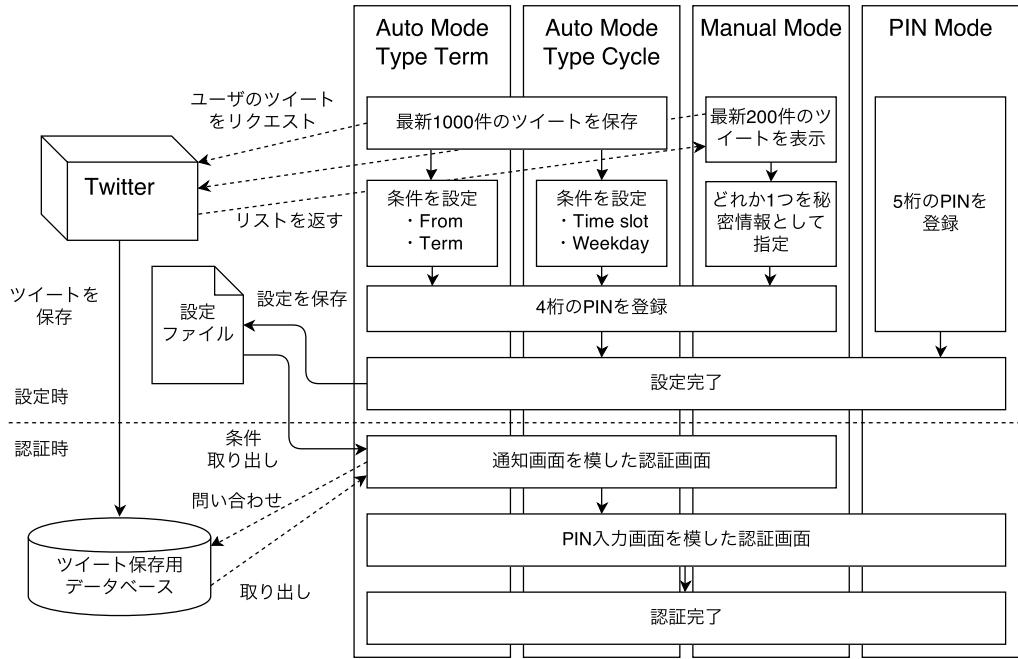


図 5.4: Notifauth のシステム概略図

新規登録画面(付録 A.3 内 - 図 A.3)^{*3}への遷移，設定画面への遷移，実験の試行を開始，実験結果の送信を行うことが可能となっている。

5.2 秘密情報の設定

この手法を用いた秘密の設定方法として，以下の 3 つを実装した。

5.2.1 Auto Mode Type Term

この認証方式は，図 5.5 のように，日/週/月/年前から日年間を指定し，認証時点にその範囲に当てはまるツイートが秘密情報となる。直近の約 1000 件のツ

^{*3} Twitter と連携するため OAuth を用いた

イートを取得し、その中で最も古いものから 12 時間前までの範囲が選択可能である。

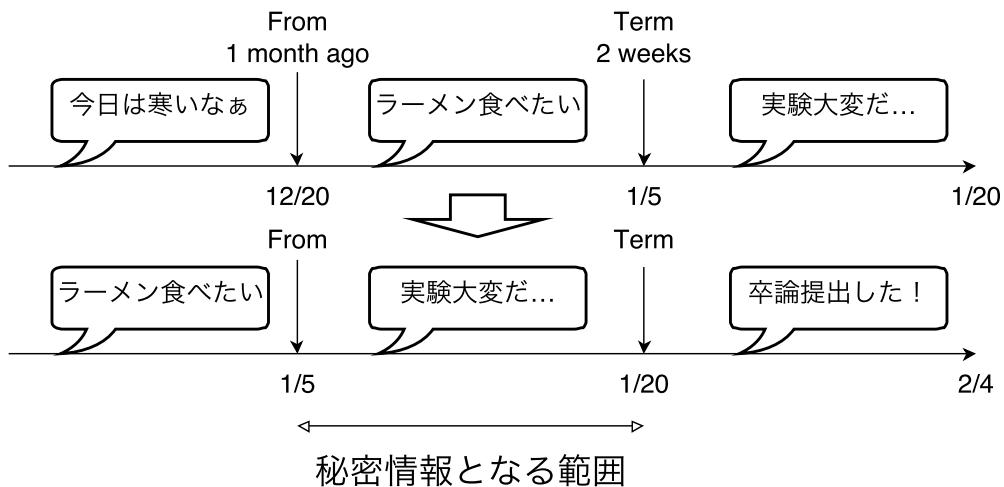


図 5.5: Auto Mode Type Term の概略図

意図

設定を行った時から時間が経過すると秘密情報とするツイートが入れ替わる場合がある。これが成立することの利点としては、

- 定期的な秘密情報の変更を能動的に行う必要が低減される
- 設定した期間等が秘匿されている限り、統計を用いた出現頻度による攻撃がしにくくなる可能性がある

が挙げられる。欠点として以下の点が挙げられる

- ユーザの本人認証率が下がる可能性がある

- 期間の設定やツイートの頻度によっては、秘密情報の数が減りすぎることで、統計的手法を用いた攻撃に脆弱になる恐れがある

設定方法

図 5.6 画面上段の「CONDITION」においてスライダーを用いて「From」(どのくらい前のツイートから秘密情報とするか)と「Term」(From からどのくらいの期間のツイートを秘密情報とするか)を設定する。各スライダーの最大値は、Notifauth によって取得しデータベースに保持されているツイートの中から最も古いものを基準として用いる。また、画面下段の「EXAMPLE」に、秘密情報として該当するツイートの一部(1行目が最古のもの、3行目が最新のもの、2行目はツイート群の配列における要素数を 2 で割った値をインデックスとして取り出したもの)を表示し、ユーザが設定を簡単に行えるための指標とする。

例えば、図 5.6 に表示されているのと同じ、1ヶ月前から 2 週間の期間のツイートを秘密情報とするように設定すると、図 5.5 のように、1/20 時点で認証操作を行う際には、「ラーメン食べたい」が秘密情報となるが、その 2 週間後である 2/4 に認証を行った時には「実験大変だ...」が正しい秘密情報として扱われ、「ラーメン食べたい」を選択しても認証は失敗する、という状況になる。

5.2.2 Auto Mode Type Cycle

この認証方式は、図 5.7 のように、曜日の 時台という条件に当てはまるツイートが秘密情報となる。

意図

この方式を採用することで、



図 5.6: Auto Mode Type Term の設定画面

- 定期的な秘密情報の変更を能動的に行う必要が低減される
- 新たな秘密情報の候補が出現することで、統計的手法を用いた攻撃に対し強度が高くなる可能性がある

ということを従来の方式と比べた利点として予想した。また、考えられる欠点として以下のものが挙げられる。

- ユーザの本人認証率が下がる可能性がある

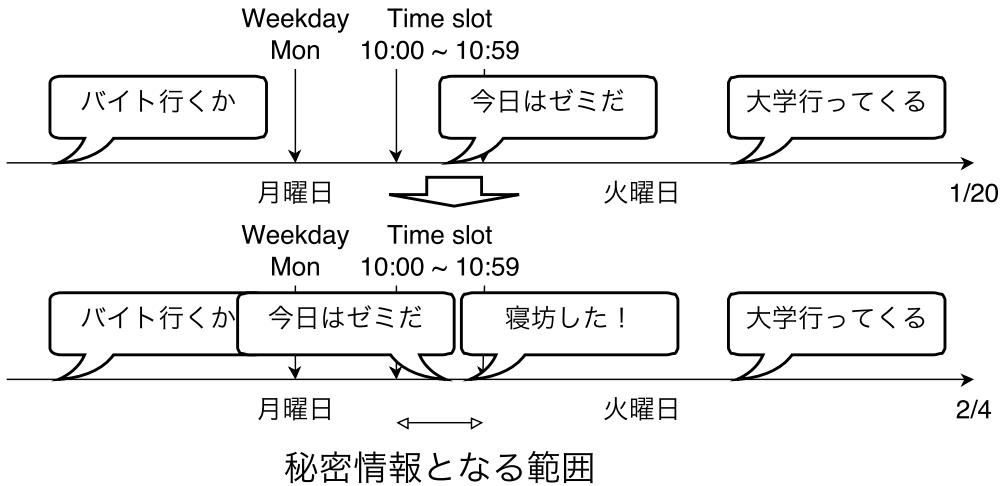


図 5.7: Auto Mode Type Cycle の概略図

設定方法

図 5.8において、画面上段の「CONDITION」においてピッカーを用いて「Time slot」(1時間単位で、何時のツイートを秘密情報とするか)を、選択式のボタンを用いて「Weekday」(何曜日のツイートを秘密情報とするか)を設定する。また、画面中段の「EXAMPLE」に、秘密情報として該当するツイートの一部(1行目が最古のもの、3行目が最新のもの、2行目はツイート群の配列における要素数を2で割った値をインデックスとして取り出したもの)を表示し、画面下段の「SUGGESTION」には Notifauth によって取得しデータベースに保持されているツイートの中で投稿回数が多い曜日・時間の組み合わせを上位3つ表示する。これらを参考にすることでユーザが設定を簡単に行えると考えられる。

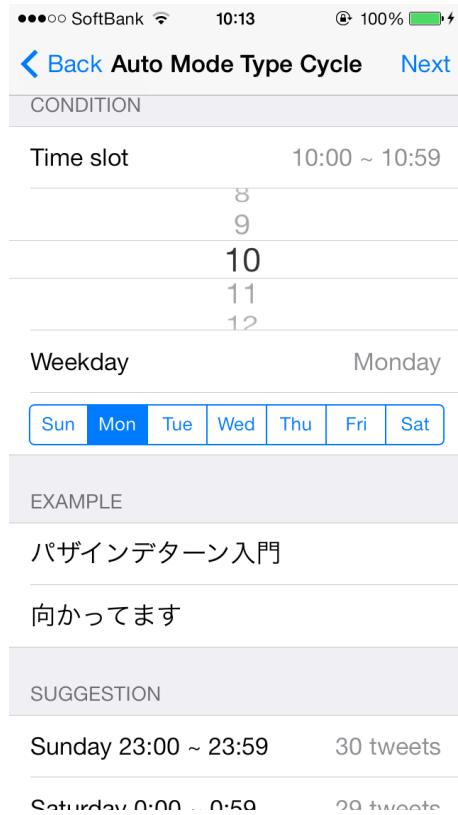


図 5.8: Auto Mode Type Cycle の設定画面

具体例

例えば、図 5.8 に表示されているのと同じ、月曜日の 10:00 ~ 10:59 に投稿されたツイートを秘密情報とするように設定すると、図 5.7 のように、1/20 時点で認証操作を行う際には「今日はゼミだ」が秘密情報となるが、その 2 週間後である 2/4 に認証を行った時には「寝坊した！」も正しい秘密情報の一つとして追加され、「今日はゼミだ」と「寝坊した！」のどちらを選択しても認証は失敗する、という状況になる。

5.2.3 Manual Mode

この認証方式は，自分のツイート最新 200 件を取得し，その中から任意に 1 つ秘密情報となるものを選ぶ．この方式では，認証時に新たなツイートの取得を行わないため，ダミーの選択肢は秘密情報を設定した時の群から選びぬかれる．

意図

当方式は 3 つの手法の中で最も単純であり，他の 2 つの方式のような特徴を持たない．そのため，以下の欠点を抱えている．

- 認証中の画面において，選択肢の中に必ず選択肢の一つとして表示されるため，総当たり攻撃や統計的手法を用いた攻撃に対して強度をもたない

設定方法

直近のツイートを最大 200 件取得し，これのうちどれを秘密情報とするかを図 5.9 のように手動で選択し設定する．ここで設定したツイートは，もう一度設定しない限りは実験終了まで固定されたままである．

5.3 認証操作

第 5.1.2 節で述べたように，本システムでは認証操作にロック画面中の通知機能を利用し開発を行った．実験を行いやすくするために本論文中の実装では，上記のロック画面を模した環境(図 5.11)をアプリケーション内に実装した．

通知の表示画面を模した認証画面(図 5.11 左)では，10 個のツイートの本文と，当てはまるものがなかった場合に選択する「No match」の 11 つの候補を表示している．その中から，正解だと思われるものを，指でタップし，そのまま右にスライ



図 5.9: Manual Mode の設定画面

ドすることで、次の画面に遷移する。なお、正解でないツイートの群をこれ以降、ダミーと呼ぶ。

PIN の入力画面を模した認証画面(図 5.11 右)では、0 から 9 までのボタンと「キャンセル」ボタンが存在し、要求される桁数の入力が完了した段階で、結果画面に遷移する。また、入力の途中で間違えた数字を選択してしまった場合は「キャンセル」ボタンをタップすることで、前画面に戻る。その際、表示される候補の内容や順番は新たに更新される。

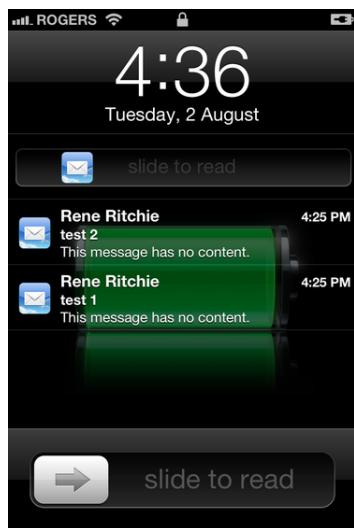


図 5.10: ロック画面上における通知の選択(スライド)動作の例

5.4 システムの使用にあたって

本システムを利用するための留意事項を表 5.1 に記す。本システムは、Apple 社が開発した携帯端末向けオペレーティングシステムである iOS 7 を搭載している携帯端末を利用し、Twitter アカウントを保持していることが利用可能な最低条件と

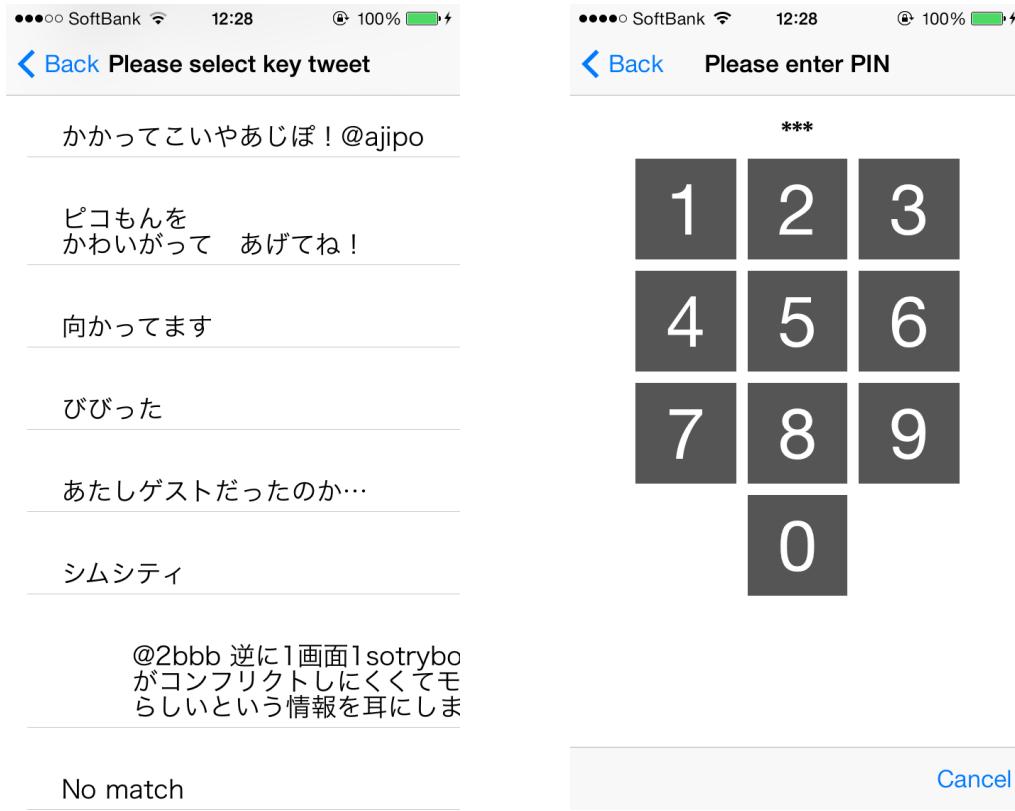


図 5.11: 左：ロック画面における通知の表示画面を模した認証画面，右：ロック画面における PIN の入力画面を模した認証画面

なる。加えて、1日あたりのツイートの数が極端に少ないと最低限度の安全性を保てない恐れが生じるため、定期的に複数のツイートを行っていることを推奨条件とした。また、本アプリケーションは、OAuth^{*4}を用いた Twitter との連携を行わなければ利用することができない。

^{*4} デスクトップ、モバイル、Web アプリケーションなどにセキュアな API 認可の標準的手段を提供するためのオープンなプロトコル

表 5.1: 留意事項

必要条件	(1)iOS7 を利用していること (2)Twitter アカウントを保持していること
推奨条件	定期的に複数のツイートを行っていること
事前準備	Twitter の OAuth を用いて本ソフトウェアと連携する

5.5 開発環境

本システムの開発環境を表 5.2 に示す。本システムは Apple 社のパーソナルコンピュータ用 OS である Mac OS X と同社の総合開発環境である Xcode を用いて開発を行った。また、動作に必要なプラットフォームとして同社の iOS バージョン 7.0 以降を搭載している端末を要求し、サポート対象となっている現行機種の 9 割以上で動作の確認を行っている。

表 5.2: 開発環境

プラットフォーム	Apple 社 iOS バージョン 7.0 以上
開発言語	Objective-C
実装環境	Mac OS X 10.9, Xcode5
動作確認環境	iPhone 4, iPhone 4S, iPhone 5, iPhone 5S, iPod touch

第 6 章

検証実験

6.1 概要

本論文で提案する個人認証システムについて、以下の 3 つの評価実験を行った。

Manual Mode を用いた認証方式の評価実験 SNS の情報を利用することで、従来の PIN を一桁増やした認証と比較し、どれだけ利便性と安全性を向上させることができるかを検証する

Auto Mode Type Term を用いた認証方式の評価実験 一定のルール（期間）に基づいて秘密情報が変化することが認証の成功率やユーザへの負担にどう影響を与えるかについて検証する

Auto Mode Type Cycle を用いた認証方式の評価実験 一定のルール（周期）に基づいて秘密情報が変化することが認証の成功率やユーザへの負担にどう影響を与えるかについて検証する

それぞれの実験は、時間的な制約から予備実験、本実験などの形式で行うことせず、一度に行った。また、ユーザからみた利便性に対する評価を得るために、2 回の選択式（一部自由記述含む）アンケートを実施した。

6.1.1 実験手順

以降の節のそれぞれの実験は第??節にて挙げた3つの実装(以降「パターン」と記載する)に対応しており、それぞれのパターンは多要素化手法として評価するために認証操作の後に4桁のPINによる認証操作を追加した。そこに「PINの桁数を一桁増やし、5桁にしたものを作秘密情報とする」パターンを追加し、計4パターンで相互に比較を行った。各パターンの実験は一つにつき8日間にわたって実施、その間に設定した日から数えて、0日目(設定直後)、1日目、3日目、8日目の4回の認証試行を行った(図6.1)。それぞれのパターンで実験中の期間は重複せず、順番は偏りのないように設定し、そのスケジュールにそって全実験を実施した。また、アンケートに関しては、16日目が経過した段階で、2種類のパターンを比較するための中間アンケート(付録B.5)を、32日目が経過した段階で最終アンケート(付録B.6)を行った。^{*1}

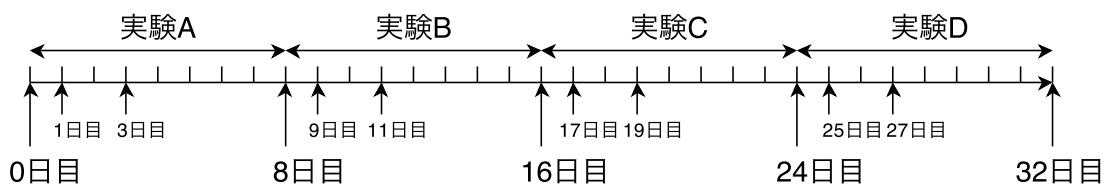


図 6.1: 実験スケジュール

^{*1}スケジュールは4つのパターンの組み合わせであり、その総数は ${}_4P_4$ の式で表される。本実験ではこれら全てに固有の番号(以降、「スケジュール番号」と記載する)を付録B.1の通り割り振つて管理する。

初回実験説明・導入

1. 実験担当者が実験の目的・注意事項・免責事項を説明する。この手順は付録Bの実験説明資料と操作説明資料を用いて行い、不明な点があれば質問してもらった。
2. 被験者のスケジュールを決定し、それに合わせて提案システムを実装したアプリケーションソフトウェア(以降「Notifauth」と記載する)のソースコードにスケジュール番号を登録した。
3. 実験担当者の開発用端末と被験者の携帯端末を接続し、Notifauthをインストールする^{*2}。
4. 実際にNotifauthを操作し、全てのパターンでひと通りの秘密情報設定と認証操作を行ってもらった。
5. その後、Notifauth内の全ての保存されたデータを初期化し、スケジュールに沿ったパターンのみ設定を行ってもらうことで実験開始とした。
6. 上記手順で設定したパターンについて認証操作を行ってもらった。
7. この段階で実験データを送信してもらい、該当データの受信を実験担当者が確認を行った。

試行手順

1. トップ画面(付録A.3内の図A.2)で、試行したいパターンをセレクタで選択し、「Test」をタップする。

^{*2}ここでAppleの開発者用アカウントと被験者の端末の紐付けを行う

2. “PIN Mode”以外の場合，ロック画面を模した画面(第5.3内の図5.11左図)が表示され，秘密情報に当てはまると思われるツイートを見つけ，そのセルをスライドする。
3. PINの入力画面(第5.3内の図5.11右図)が表示され，“PIN Mode”であれば5桁，それ以外のパターンであれば4桁のPINを入力する。
4. 結果画面が表示されるので，「Home」ボタンを押す。

なお，試行手順の一つとして，実験結果は認証操作直後にメールで送信して頂くかたちで収集した。

6.1.2 被験者

男性12名，女性3名の計15名が実験を行った。うち本学の学生は5名であった。性別や年齢は表6.2の通りである。実験の開始時期の関係から，全員が実験を完了しているわけではなく，全ての検証実験を終了したのは10人で，そのうち最終アンケートに答えたのは7人である。中間アンケートには12人が回答した。

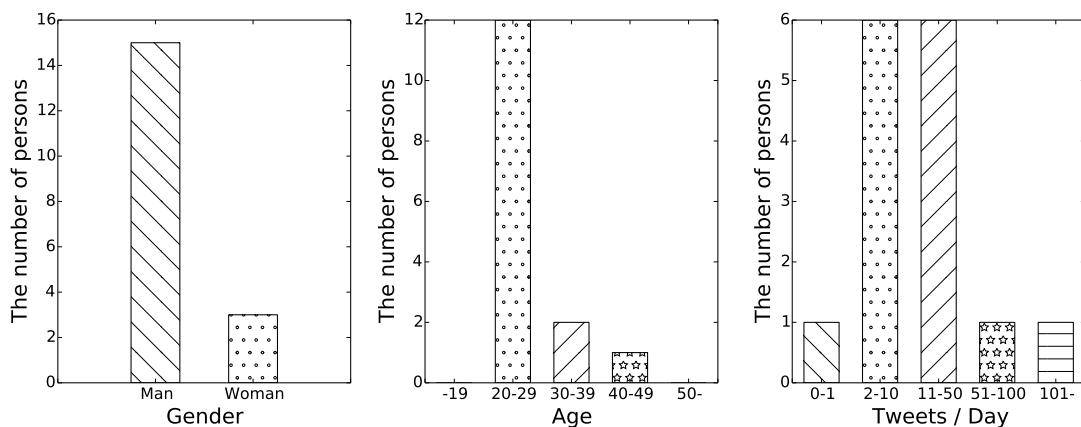


図6.2: 被験者の特性(左:性別，中央:年齢，右:1日あたりのツイート数)

6.2 Manual Mode を用いた認証方式の評価実験

6.2.1 概要

本実験では、SNS の情報を利用することで、従来の PIN を一桁増やした認証と比較し、どれだけ利便性と安全性を向上させることができるかの評価を行う。本実験で評価対象とするパターンとして，“Manual Mode”を採用する。

6.2.2 目的

アプリケーションを用いた実験では、測定した結果から以下の 3 指標にもとづいて相関や有意差をみた。

- 短期の記憶保持

目的 秘密情報の短期記憶が可能かどうかの検証

仮説 日数をおかない試行において 5 衔の PIN 認証よりも認証成功率が高い、つまり、短期の記憶保持について暗証番号認証よりも当方式を用いた認証の方が容易

測定方法 0 日目、1 日目、3 日目の認証成功率を比較し、相関をみる

- 長期の記憶保持

目的 秘密情報の長期記憶が可能かどうかの検証

仮説 日数をおく試行において 5 衔の PIN 認証よりも認証成功率が高い、つまり、長期の記憶保持について暗証番号認証よりも当方式を用いた認証の方が容易

測定方法 3 日目と 8 日目の認証成功率を比較し、相関を見る。

- 認証時間

目的 利便性の検証

仮説 5 衝の PIN 認証よりも認証時間が短い、つまり、暗証番号認証よりも当方式を用いた認証の方が利便性が高い

測定方法 認証操作の画面が表示されてから、認証を終えるまでの時間を計測する。認証の成否は問わないものとする。

6.2.3 方法

被験者実験により各試行の成功と失敗、認証にかかった時間を収集し、事後アンケートを実施する。また、有意差は Welch の t 検定を用いる（この場合、(1) サンプルサイズが 30 以上で十分大きいこと、(2) 検定を複数回繰り返すことで帰無仮説全体を通しての有意水準が不当に上昇してしまう、(3)t 検定は母分布が正規分布でないときにも頑健性を持つ（妥当な結果を与える）ことから、標本が正規分布であることは検証をせずに自明とした）。

6.2.4 結果

本実験の結果を表 6.2 に示す。試行のタイミングが 1 日程度前後した被験者が存在したため、1-2 日目を 1 日目、3-6 日目におこなったものを 3 日目、7-8 日目に行つたものを 8 日目の試行とした。比較のための PIN Mode における認証成功率と認証時間は表 6.2 に示す。

表 6.1: Manual Mode における各経過日数ごとの認証成功率と認証時間の変化 ($n = 51$)

経過日数	認証成功率 (%)	認証時間 (秒)
0	92.9	14.2
1-2	91.7	10.7
3-6	91.7	8.9
7-8	100.0	8.7
<hr/>		
平均	94.12	10.74
標準偏差	3.47	8.61
中央値		7.86
最大値		41.84
最小値		3.29

記憶保持

表 6.2 に示した通り、経過日数と認証成功率におけるピアソンの相関係数は 0.144 で、標本数による限界値 [30] を考慮しても有意な差ではないと考えられる。また、図 6.3 に PIN Mode との認証率の比較を示した。検定を行った結果、PIN Mode の認証成功率とは有意差がある (Welch の t 検定, $p = 0.012 < 0.05$) ことが明らかになった。

- 短期の記憶保持

0日目，1日目，3日目にかけての Manual Mode での認証成功 rate は 92.9%，91.7%，91.7% と推移しており，ピアソン相関係数は 0.031 で，有意な相関はみられなかった。

- 長期の記憶保持

3日目の認証成功 rate は 91.7% で，8日目には 100% であった。3日目から 8日目にかけてのピアソン相関係数は 0.261 であり，有意な相関はみられなかった。

認証時間

図 6.4 に Manual Mode と PIN Mode との認証時間の比較を示した。PIN Mode とは数値上 4 倍程度の差があり，検定を行った結果，有意差がある (Welch の t 検定， $p = 0$) といえた。

表 6.2: PIN Mode における各経過日数ごとの認証成功 rate と認証時間の変化 ($n = 51$)

経過日数	認証成功 rate (%)	認証時間 (秒)
0	100.0	2.17
1-2	92.3	2.81
3-6	90.9	2.82
7-8	100.0	2.51
平均	96.08	2.55
標準偏差	4.22	1.60
中央値		1.93
最大値		8.40
最小値		1.11

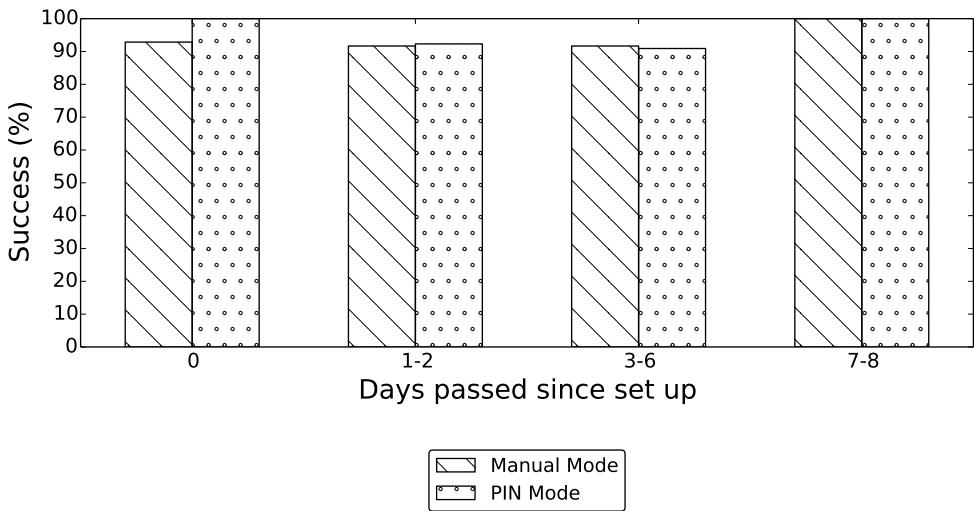


図 6.3: Manual Mode と PIN Mode における設定時からの経過日数ごとの認証成功率

アンケート結果

表 6.6 に被験者によるアンケート結果を示す。また、PIN Mode に対するアンケート結果も併せて表 6.4 に示す。

6.3 Auto Mode Type Term を用いた認証方式の評価実験

6.3.1 概要

本実験では、SNS の情報の特性を利用した認証システムとして “Auto Mode Type Term” を採用し、記憶持続性と利便性の評価を行う。更に、ある一定のルールに基づいて秘密情報が変化することが認証の成功率やユーザへの負担がどう影響を与えるかについても検証する。また、他の実験で用いたパターンとの比較も行う。

6.3.2 目的

アプリケーションを用いた実験で測定した結果から相関や有意差をみた指標は第6.2節に準じ、(1)短期の記憶保持、(2)長期の記憶保持、(3)認証時間とした。

6.3.3 方法

被験者実験により各試行の成功と失敗、認証にかかった時間を収集し、事後アンケートを実施する。平均値を比較する際の検定方法は6.2節に準じ、Welchのt検定を利用した。

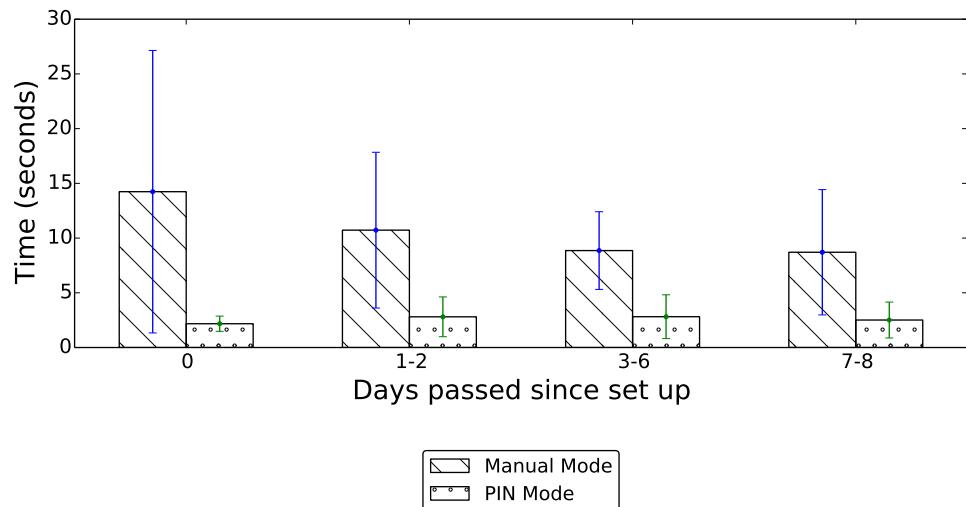


図 6.4: Manual Mode と PIN Mode における設定時からの経過日数ごとの認証時間

表 6.3: 被験者による Manual Mode に対するアンケート内評価

項目名	平均値	回答者数
秘密情報の記憶保持にかかる負担はどのくらい感じますか？ (とても小さい：1-とても大きい：5)	1.13	8
認証にかかる時間はどのように感じましたか？ (とても短い：1-とても長い：5)	1.13	8
認証を成功させるために必要な操作負担はどの程度でしたか？ (とても小さい：1-とても大きい：5)	1.25	8
認証を行うのにどれくらいフラストレーションを感じましたか？ (とても小さい：1-とても大きい：5)	1.5	8
タイピングしたりタッチパネルをスライドしたりする作業の 負担はどの程度でしたか？(とても小さい：1-とても大きい：5)	1.43	7

6.3.4 結果

本実験の結果を表 6.5 に示す。試行のタイミングが 1 日程度前後した被験者が存在したため、1-2 日目を 1 日目、3-6 日目におこなったものを 3 日目、7-8 日目に行つたものを 8 日目の試行とした。比較のための PIN Mode における認証成功率と認証時間は第??節と同じく表 6.2 に示す。

記憶保持

表 6.5 に示した通り、経過日数と認証成功率におけるピアソンの相関係数は 0.234 で、標本数による限界値を考慮すると有意ではないと考えられる。また、図 6.5 に

PIN Modeとの認証率の比較を示した。検定を行った結果、PIN Modeの認証成功率とは有意差がある (Welch の t 検定, $p = 0$) ことが明らかになった。

- 短期の記憶保持

0日目から3日目までの Manual Modeでの認証成功率は 38.5%から 75.0%とばらつきがみられ、ピアソン相関係数は 0.233 で有意な相関ではない。

- 長期の記憶保持

3日目の認証成功率は 75.0%, 8日目の認証成功率は 41.7%で、期間が空くと認証成功率が下がってしまった。3日目から8日目の期間におけるピアソン相関係数

表 6.4: 被験者による PIN Mode に対するアンケート内評価

項目名	平均値	回答者数
秘密情報の記憶保持にかかる負担はどのくらい感じますか? (とても小さい : 1-とても大きい : 5)	1.13	8
認証にかかる時間はどのように感じましたか? (とても短い : 1-とても長い : 5)	1.13	8
認証を成功させるために必要な操作負担はどの程度でしたか? (とても小さい : 1-とても大きい : 5)	1.13	8
認証を行うのにどれくらいフラストレーションを感じましたか? (とても小さい : 1-とても大きい : 5)	1.5	8
タイピングしたりタッチパネルをスライドしたりする作業の 負担はどの程度でしたか? (とても小さい : 1-とても大きい : 5)	1.43	7

表 6.5: Auto Mode Type Term における各経過日数ごとの認証成功率と認証時間の変化 ($n = 58$)

経過日数	認証成功率 (%)	認証時間
0	40.0	21.52
1-2	38.5	21.67
3-6	75.0	20.30
7-8	41.7	19.09
<hr/>		
平均	51.79	22.14
標準偏差	14.39	14.50
中央値		19.24
最大値		65.0
最小値		5.62

は-0.347で、相関はみられなかった。

認証時間

図 6.6 に PIN Mode との認証時間の比較を示した。こちらも Manual Mode 同様、PIN Mode とは大きく差があり、検定を行った結果、有意差がある (Welch の t 検定, $p = 0$) ことが判明した。

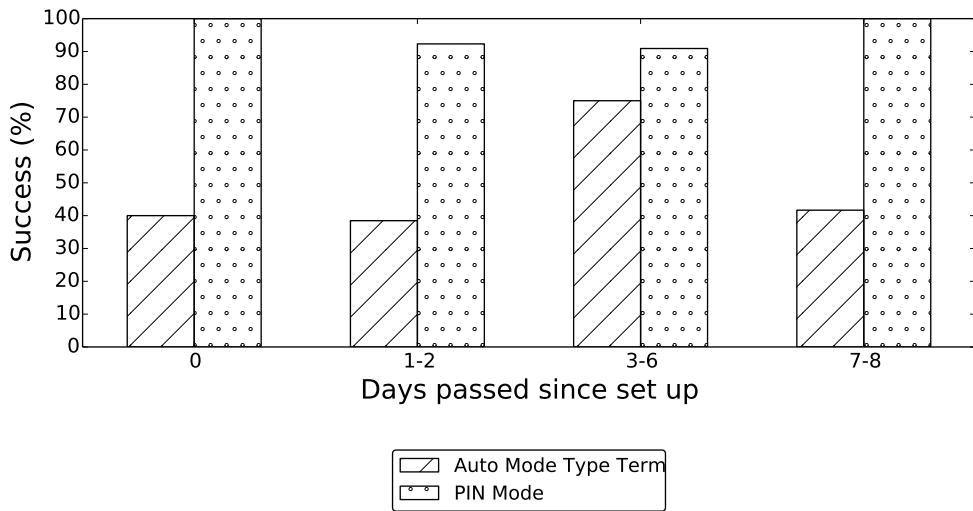


図 6.5: Auto Mode Type Term と PIN Mode における設定時からの経過日数ごとの認証成功率

アンケート結果

被験者によるアンケート結果を表 6.6 に記す。この認証方法で失敗したことがあると答えた人の中で、設定内容を覚えていた人は 7 人中 4 人で、設定内容を忘れてしまった人は 7 人中 3 人であった。

6.4 Auto Mode Type Cycle を用いた認証方式の評価実験

6.4.1 概要

本実験では、SNS の情報の特性を利用した認証システムとして “Auto Mode Type Cycle” を採用し、記憶持続性と利便性の評価を行う。更に、ある一定のルールに基づいて秘密情報が変化することが認証の成功率やユーザへの負担がどう影響を

与えるかについても検証する。また、他の実験で用いたパターンとの比較も行う。

6.4.2 目的

アプリケーションを用いた実験で測定した結果から相関や有意差をみた指標は第6.2節に準じ、(1) 短期の記憶保持、(2) 長期の記憶保持、(3) 認証時間とした。

6.4.3 方法

被験者実験により各試行の成功と失敗、認証にかかった時間を収集し、事後アンケートを実施する。平均値を比較する際の検定方法は6.2節に準じ、Welch の t 検定を利用した。

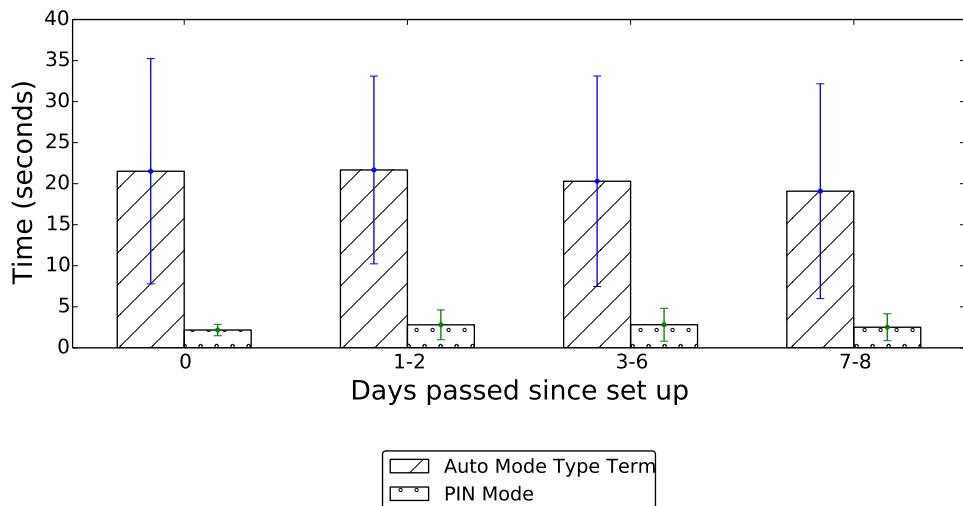


図 6.6: Auto Mode Type Term と PIN Mode における設定時からの経過日数ごとの認証時間

表 6.6: 被験者による Auto Mode Type Term に対するアンケート内評価

項目名	平均値	回答者数
秘密情報の記憶保持にかかる負担はどのくらい感じますか？ (とても小さい：1-とても大きい：5)	4.09	11
認証にかかる時間はどのように感じましたか？ (とても短い：1-とても長い：5)	3.73	11
認証を成功させるために必要な操作負担はどの程度でしたか？ (とても小さい：1-とても大きい：5)	3.18	11
認証を行うのにどれくらいフラストレーションを感じましたか？ (とても小さい：1-とても大きい：5)	3.45	11
タイピングしたりタッチパネルをスライドしたりする作業の 負担はどの程度でしたか？(とても小さい：1-とても大きい：5)	3.14	7

6.4.4 結果

本実験の結果を表 6.7 に示す。試行のタイミングが 1 日程度前後した被験者が存在したため、1-2 日目を 1 日目、3-6 日目におこなったものを 3 日目、7-8 日目に行つたものを 8 日目の試行とした。比較のための PIN Mode における認証成功率と認証時間は第??節と同じく表 6.2 に示す。

記憶保持

表 6.7 に示した通り、経過日数と認証成功率におけるピアソンの相関係数は 0.083 で、標本数による限界値を考慮すると有意ではないと考えられる。また、図 6.7 に

表 6.7: Auto Mode Type Cycle における各経過日数ごとの認証成功率と認証時間の変化 ($n = 58$)

経過日数	認証成功率 (%)	認証時間
0	35.7	21.7
1-2	25.0	21.5
3-6	14.3	26.2
7-8	30.8	22.9
平均	27.59	22.95
標準偏差	7.98	15.27
中央値		20.1
最大値		91.0
最小値		4.99

PIN Mode との認証率の比較を示した。検定を行った結果、PIN Mode の認証成功率とは有意差がある (Welch の t 検定, $p = 0$) ことが明らかになった。

- 短期の記憶保持

0 日目から 3 日目までの Auto Mode Type Cycle における認証成功率は 35.7% から 14.3% まで落ちた。この期間でのピアソン相関係数は -0.149 であり、有意な相関はみられなかった。

- 長期の記憶保持

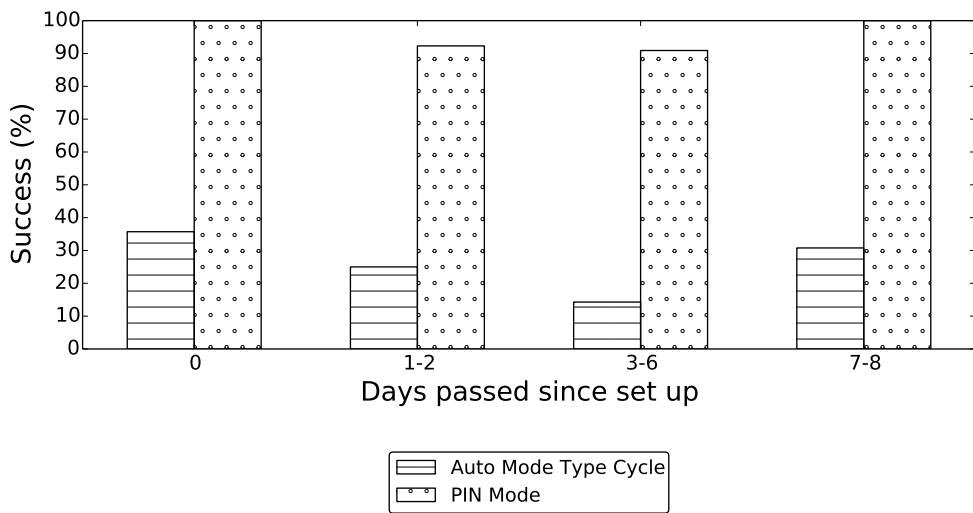


図 6.7: Auto Mode Type Cycle と PIN Mode における設定時からの経過日数ごとの認証成功率

3 日目の認証成功率は 14.3% , 8 日目の認証成功率は 30.8% であり , ピアソン相関係数 0.243 と有意な相関はみられなかった .

認証時間

図 6.8 に PIN Mode との認証時間の比較を示した . こちらも他の Mode 同様 , PIN Mode とは大きく差があり , 検定を行った結果 , 有意差がある (Welch の t 検定 , $p = 0$) ことが判明した .

アンケート結果

被験者によるアンケート結果を表 6.8 に記す . この認証方法で失敗したことがあると答えた人の中で , 設定内容を覚えていた人は 7 人中 5 人で , 設定内容を忘れて

しまった人は7人中2人であった。

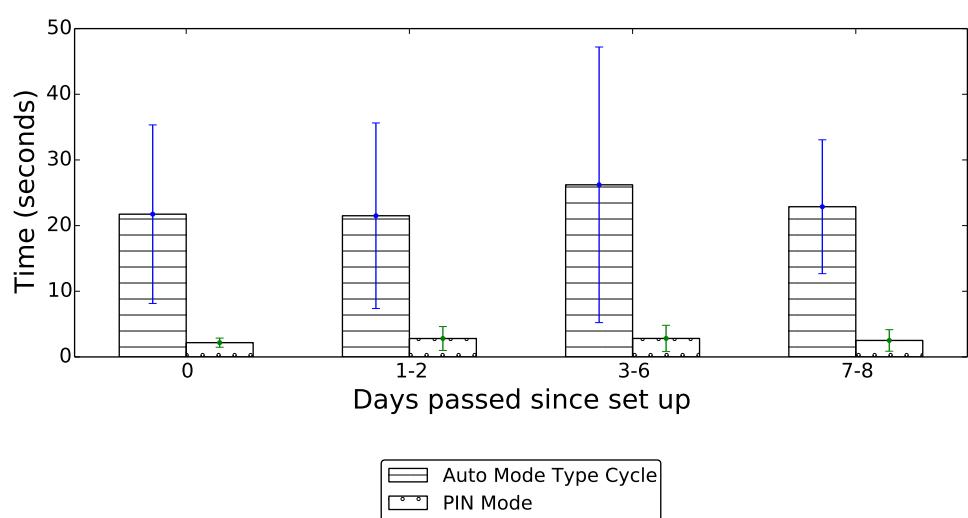


図 6.8: Auto Mode Type Cycle と PIN Mode における設定時からの経過日数ごとの認証時間

表 6.8: 被験者による Auto Mode Type Cycle に対するアンケート内評価

項目名	平均値	回答者数
秘密情報の記憶保持にかかる負担はどのくらい感じますか? (とても小さい : 1-とても大きい : 5)	3.56	11
認証にかかる時間はどのように感じましたか? (とても短い : 1-とても長い : 5)	3.44	11
認証を成功させるために必要な操作負担はどの程度でしたか? (とても小さい : 1-とても大きい : 5)	2.44	11
認証を行うのにどれくらいフラストレーションを感じましたか? (とても小さい : 1-とても大きい : 5)	3.44	11
タイピングしたりタッチパネルをスライドしたりする作業の 負担はどの程度でしたか?(とても小さい : 1-とても大きい : 5)	2.57	7

第 7 章

考察

7.1 安全性に関する考察

安全性に関しては，単純な場合の数においては 5 行の PIN による認証方式と同等の 10^5 通りにして実験を行い，認証要素を増やす前と比べて安全性が向上したと言える。また，本システムでは，ダミーの数を増やすことで柔軟に安全性を高めることができる。しかし，設定情報が漏洩してしまえば，あとは使用しているアカウントの投稿データを取得するだけで簡単に攻撃が可能となってしまう。加えて，期間や周期を秘密情報として設定を行う場合，設定情報のエントロピーは投稿の頻度に依存し，従来の方式に比べて高いとはいえない。

更に，認証のエラー率を下げるために秘密情報が極端に多いまたは小さくなるよう設定した場合にも統計を用いた攻撃に脆弱となってしまう恐れがある。そのため，設定時の因子を増やしたり，認証操作をリストからの選択式にせず 2 抹を用いた上で回答回数を増やすなどの改善によってエントロピーを減少させず [31] にエラー率を低下させる方法を検討する必要があると考えられる。

7.2 覚えやすさに関する考察

本システムの認証方式では、Twitter の投稿を用いることによって覚えやすさを向上させることが主たる目的として存在し、被験者実験において、Manual Mode による手動での秘密情報の設定の場合は 5 行の PIN による認証と比べても同等の認証成功率を得ることができたと考えられる。

Auto Mode Type Term において、0 日目から 3 日目までの Manual Mode での認証成功率において、その中で 0 日目と 1 日目の認証成功率が低いにも関わらず 3 日目で上昇しており、設定した条件は記憶していたもののうまく候補の中から当てることが出来なかった可能性がある。Auto Mode Type Cycle においては、8 日間通しての平均認証成功率は 27.59% と、PIN による認証と比べてかなり劣ることが明らかになったが、この結果に関しても上記の理由によるものだと推測できる。それらの条件設定により秘密情報が変化する手法では、設定は覚えているがその条件に当てはまる秘密情報を選ぶことができない場合が多かったことは第 6.3 節と第 6.4 節のアンケートの結果からもみることができた。そのため、ユーザの記憶が曖昧になってしまふと考えられる情報を排除するなどの対策をとる必要があると考えられる。

長期間における記憶に関しては、Auto Mode Type Term に関しては 75.0% から 41.7% と認証成功率の大幅な下降がみられたが、それ以外の認証方式では大きく下降しないもしくは大きく上昇し、Auto Mode Type Term に関しても有意な相関ではないことと前段落の推測から、いずれの方式も覚えやすさに大きな差はないと考えられる。

3 つの提案手法と 1 つの既存手法を比較すると、42.9% の被験者が Manual Mode が最も覚えやすいと答え、同じく 28.6% の被験者は PIN Mode が最も覚えやすい

と答えた。そのため、Manual Mode に関しては PIN よりも覚えやすさにおいて優れていると言える。

7.3 使用継続性に関する考察

本システムの認証方式では、設定方法によっては長期間使用することにより、秘密情報のエントロピーが上昇したり、自動的に秘密情報が入れ替わることで定期的な秘密情報変更をする必要が小さくなるなどの利点が存在する。また、被験者アンケートで得られた感想などを見ても、利便性についての評価が高かった。しかしながら、認証操作にかかる時間に関しては、いずれの方式も PIN による認証よりも平均で 10 倍以上多くかかっているという結果が得られているため、日常的に頻繁に使用する携帯端末においては、ユーザの利便性を著しく下げ、利用継続性を損ねてしまう可能性が存在する。この問題点の解決方法として、既存研究のように認証手法を 2 択にして複数回回答させるといったものが挙げられる。

3 つの提案手法と 1 つの既存手法を比較すると、57.1% の被験者が Manual Mode が最も「今後日常的に使いたい」と答え、PIN Mode でそう答えた被験者は 28.6% であった。そのため、Manual Mode に関しては PIN よりも使用継続性において優れていると言える。

7.4 他環境における応用に関する考察

本システムの考え方は、ハードウェアへの依存の少なさや、設定方法の単純さから、携帯端末以外の環境でも応用が可能だと考えられる。

また、他の SNS を使用可能かという点に関しては、OAuth などのセキュアな認証プロトコルと Web API が提供されていればデータベース構造などを大きく変え

る必要もなく導入できる。これにより、様々なSNSの情報を組み合わせることでエントロピーの上昇や、新たな秘密情報の設定が可能になることで安全性の向上にも繋がることが考えられる。

7.5 今後の課題

7.5.1 仕組み

認証時に表示する秘密情報の候補数は、総当たり攻撃に対する頑強さに直結する。そのため、どれくらいの数まで候補を表示しても、ユーザが負担だと感じないかを調べる必要があり、それによってユーザの利便性を可能な限り損なわない形で簡単に安全性を強化できると考えられる。

秘密情報の設定において、アンケートで得られた回答として多かったのは、「設定情報は覚えているがそれに当てはまるツイートを選べない」という問題である。これが原因で、自動で設定する2手法(Auto Mode Type TermとAuto Mode Type Cycle)では、大きく認証成功率が落ちていると考えられる。また、本手法においては、被験者が任意に条件を設定することが可能なので、認証成功率が低くなれば自然に簡単な設定、例えば「Fromを1年前、Termを1年間に設定する(直近1年間のツイート全てが秘密情報となる)」や「明け方など、1-2件しかツイートしていない時間帯にTime slotを設定(そのツイートの内容 자체を覚えてしまう)」などにしがちであると予測できる。これは被験者実験によって得られた設定内容からも読み取ることができ、これにより利便性の高低が安全性にも関わっているといえる。この問題については、秘密情報の回答方法を根本的に変えることで改善できる可能性がある。具体的には、既存手法(第4.2.1項)で行っているように、独立した一つの情報、本システムの場合は1ツイートに対して2~4択で正解の選択肢を答えさ

せ，それを複数回繰り返すというものである．既存手法ではこれに加え，曖昧な記憶による認証の失敗を防ぐため，はっきりと覚えている情報に対してのみ正しい回答どうかを検証した．この2手法を導入することで，利便性と安全性について，どちらも改善できる可能性がある．

また，Twitterのアカウントが使用不可になってしまった場合(例えば，利用規約に反するアカウントであると誤って判断され，凍結されてしまう例が存在する)に，この認証が使用不可能になってしまう．その場合に，復旧のための手段を用意する必要があると考えられる．

7.5.2 実装

Manual Modeにおいて，現状の実装では秘密情報となるツイートを1つ選ぶという設定しか行えない．そのため，認証時に(1)毎回必ず候補として表示されるツイートがあり，それが正解と予測できてしまうこと，(2)“Protected”に設定してあるアカウントを利用している場合などで，他人に見られては困る発言が候補として表示されてしまうこと，の2つの安全性に関する問題が起こり得てしまう．そのため，(1)複数の秘密情報を設定可能にする，(2)認証に利用したくない情報は予め利用者の判断で弾けるようにする，といった解決策を実装する必要があると考えられる．

7.5.3 実験

アンケートでは携帯端末は毎日使うというユーザがほとんどであった．そのため1日に何度も認証を行うという予測のもと実験を行うことで，使用継続性についてより細かく改善すべき点が見えてくるのではないかと感じた．

今回はiOS用のアプリケーションソフトウェアとして実装を行ったが、結果として被験者の端末にインストールするための制約が大きく、被験者の数を十分に集めることができなかった。したがって、Webサービスとしてブラウザから利用可能な形で実装することで、より多くの実験データを集めたり、更に長期にわたつて実験を行うことが可能になると考えられる。

第 8 章

結論

本論文では、現在の多要素認証における現状の確認、ライログや SNS の情報を認証に使うことの有用性などを検討し、既存手法の問題点を洗いだした上で、Twitter の情報を用いた携帯端末向け個人認証の多要素化手法の提案、実験と結果の解析を行った。

本論文で提案した 3 種類の個人認証手法と各手法に対する被験者実験では、一つは自分の直近 200 件のツイートの中から手動で設定した秘密情報に関して記憶維持できるかを、残りの二つは期間や時間曜日を設定することで自分の直近 1000 件のツイートの中から自動で決定された秘密情報に関して記憶維持できるかをそれぞれ調査した。更に、覚えやすさと使いやすさに関して 5 段階での評価と自由記述を含むアンケートを被験者に回答してもらうことで、使用継続性などを評価・比較した。

被験者実験によって、Twitter の情報を認証に用いることで、記憶持続性を高めることができると考えられる。しかし、条件設定により秘密情報が変化する手法を用いた場合には認証成功率が低く、設定を覚えているにもかかわらず秘密情報として正解となるものを選ぶことがユーザにとって難しいという予測がたてられた。利便性の面からみると、手動で秘密情報を設定する手法では、既存の PIN を用いた認証と比べて認証にかかる時間は劣っているものの、アンケートの結果では優

れていると回答する割合が多かった。

今後の課題として、条件設定により秘密情報を決定する手法に関して、記憶の曖昧さに配慮する認証操作の開発や、認証の際に正しい秘密情報であるツイートの選択にかかる時間を減らす工夫の導入、推測による攻撃に対する脆弱性を解消するため設定方法を柔軟にすることなどを検討する必要があると結論づけられた。加えて、実験に関しても試行を行う頻度や対象とするプラットフォームなどに改善すべき点がみられたため、計画を見直した上で更なる検証が必要だと感じた。

謝辞

本研究を進めるにあたって、1年間を通して丁寧な御指導、数々の御助言をしてくださいました高田哲司准教授に厚く御礼申し上げます。

また、研究について数々の知識やアドバイスをいただいた、高田研究室の皆様に深く感謝いたします。

加えて、実装や実験について数多くの知見を与えて下さり、本論文についても様々なご指摘を下さいました石井通人さんと原田陽紗子さん、更に、本論文の校正をして下さいました安部草麻生さんと実験に協力して下さった方々に深く感謝の意を申し上げます。

最後に、不自由ない学生生活を支援してくれた両親に心から感謝致します。

参考文献

- [1] Google 2-Step Verification, 2014-01-18. <http://www.google.com/landing/2step/>.
- [2] How do I enable two-step verification on my account? - Dropbox, 2014-01-15.
<https://www.dropbox.com/help/363>.
- [3] Two-Step Verification Available to All Users — Evernote BlogEvernote Blog, 2014-01-15. <http://blog.evernote.com/blog/2013/10/04/two-step-verification-available-to-all-users/>.
- [4] Imperva Releases Detailed Analysis of 32 f Breached Consumer Passwords, 2014-01-28. http://www.imperva.com/news/press/2010/01_21_imperva_releases_detailed_analysis_of_32_million_passwords.html.
- [5] Do We Have To Tust Biometric Authentication -Kaspersky Daily — We use words to save the world — Kaspersky Lab Official Blog, 2014-01-28. <http://blog.kaspersky.com/biometric-authentication/>.
- [6] Emiliano De Cristofaro, Honglu Du, Julien Freudiger, and Gregory Norcie. Two-factor or not two-factor? a comparative usability study of two-factor authentication. *CoRR*, abs/1309.5344, 2013.
- [7] ワンタイムパスワードのご案内 | ジャパンネット銀行, 2014-01-15. <http://www.japannetbank.co.jp/security/security/otp.html>.
- [8] Battle.net Authenticator - Battle.net Support, 2014-01-15. <https://us.battle.net/support/en/article/battlenet-authenticator>.

- [9] Shinji R. Yamane. Secure online game play with token: A case study in the design of multi-factor authentication device. In *Proceedings of the 2Nd International Conference on Human Centered Design*, HCD'11, pages 597–605, Berlin, Heidelberg, 2011. Springer-Verlag.
- [10] IPA 独立行政法人情報処理推進機構: コンピュータウイルス・不正アクセスの届出状況 [6月分および上半期] について, 2014-01-26. <http://www.ipa.go.jp/security/txt/2012/07outline.html>.
- [11] Here's Everywhere You Should Enable Two-Factor Authentication Right Now, 2014-01-26. <http://lifehacker.com/5938565/heres-everywhere-you-should-enable-two-factor-authentication-right-now>.
- [12] Bruce Schneier. Two-factor authentication: Too little, too late. *Commun. ACM*, 48(4):136–, April 2005.
- [13] IDC Japan. 国内モバイル / クライアントコンピューティング機器家庭ユーザー利用実態調査結果を発表, 2013. <http://www.idcjapan.co.jp/Press/Current/20131003Apr.html>.
- [14] google-authenticator - Two-step verification - Google Project Hosting, 2014-01-28. <https://code.google.com/p/google-authenticator/>.
- [15] RFC 4226 - HOTP: An HMAC-Based One-Time Password Algorithm, 2014-01-26. <http://tools.ietf.org/html/rfc4226>.
- [16] RFC 6238 - TOTP: Time-Based One-Time Password Algorithm, 2014-01-26. <http://tools.ietf.org/html/rfc6238>.

- [17] PASSBAN, 2014-01-25. <http://www.passban.com/>.
- [18] Authy, 2014-01-25. <https://www.authy.com/>.
- [19] ソーシャルメディアの利用状況：平成23年版情報通信白書, 2014-01-25.
<http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h23/html/nc232310.html>.
- [20] 田村 健範, 鶴丸 和宏, 市野 将嗣, and 小松 尚久. Web閲覧履歴情報に着目したログによる本人認証に関する一考察(デジタルドキュメント, ライフログ活用技術, オフィス情報システム, 一般). 電子情報通信学会技術研究報告. *LOIS, ライフインテリジェンスとオフィス情報システム*, 111(152):19–24, jul 2011.
- [21] 長谷 容子, 青木 輝勝, and 安田 浩. M-068 スケジュールとGPS情報を用いた認証方法の検討(M.ネットワーク・モバイルコンピューティング). 情報科学技術フォーラム一般講演論文集, 3(4):235–236, aug 2004.
- [22] 今澤 貴夫, 小池 英樹, and 高田 哲司. GPSデータを用いた位置認証システムとその停留点算出方式. 情報処理学会シンポジウム論文集, 2008(8):707–712, 2008-10-08.
- [23] 西垣 正勝 and 小池 誠. ユーザの生活履歴を用いた認証方式:電子メール履歴認証システム(ネットワークセキュリティ). 情報処理学会論文誌, 47(3):945–956, mar 2006.
- [24] Tomofumi Nemoto, Kyohei Furukawa, and Manabu Okamoto. Poster: Knowledge-Based Authentication using Twitter. Symposium On Usable Privacy and Security 2011, 2011.

- [25] Making Photo Tagging Easier - Facebook, 2014-01-28. <https://www.facebook.com/notes/facebook/making-photo-tagging-easier/467145887130>.
- [26] 2012年 Facebook ユーザ500人 利用実態調査|マクロミル, 2014-01-28. http://www.macromill.com/r_data/20120315facebook/.
- [27] Twitter Help Center — About public and protected Tweets, 2014-01-18. <https://support.twitter.com/entries/14016>.
- [28] 総務省. 情報通信白書平成24年版, 2013. <http://www.soumu.go.jp/johotsusintokei/whitepaper/h24.html>.
- [29] News - Windows sees strong European growth - Kantar Worldpanel, 2014-01-28. <http://www.kantarworldpanel.com/global/News/news-articles/Windows-sees-strong-European-growth>.
- [30] 南風原 朝和. 心理統計学の基礎 統合的理解のために, 2002-08.
- [31] 兼子 拓弥, 本部 栄成, and 西垣 正勝. ユーザ認証においてユーザが覚えるべき秘密情報のエントロピに関する一考察. Symposium On Cryptography and Information Security 2013, 2013.

付録 A

実装に関する付録

A.1 実装の詳細

Notifauth は、iOS 用アプリケーションとして実装された。クラス図は図 A.1 の通りである。

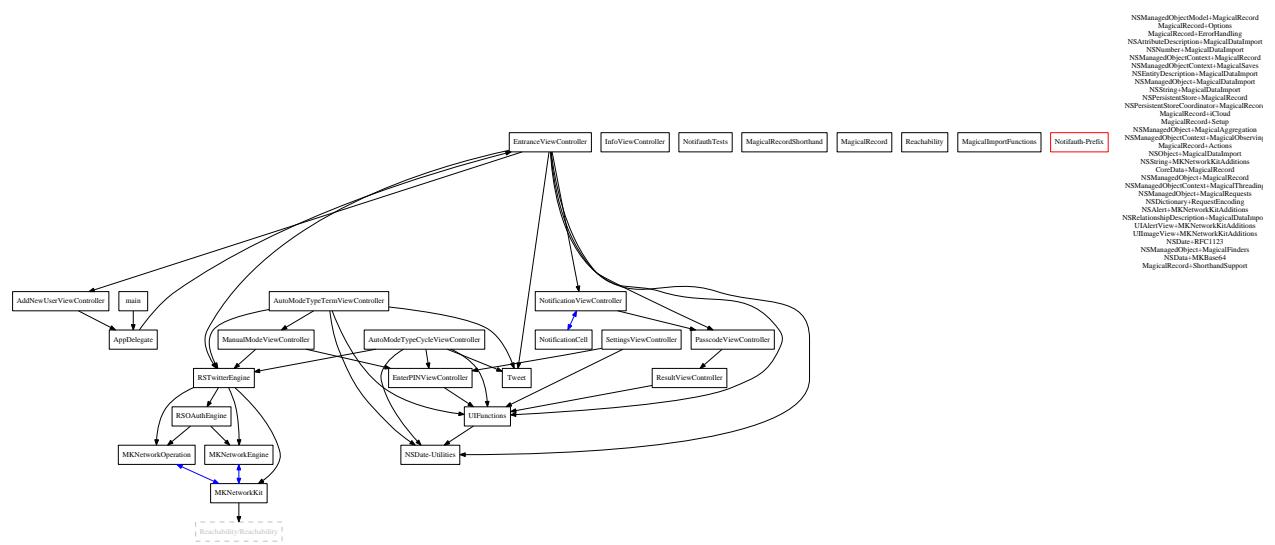


図 A.1: Notifauth のクラス図

Twitter の認証情報 (OAuth の認証トークン) は、Apple 社の “Keychain” により、暗号化し保存されている。ツイートのデータは、Object-relational mapping(以降

ORM) フレームワークである CoreData を用いて SQLite ファイルに保存されている。また、Notifauth 内の様々な設定情報は iOS 標準の NSUserDefaults オブジェクトを利用しアプリケーションソフトウェア内の専用領域に保存されており。今回は特に暗号化は行っていない。

使用したサードパーティ製ライブラリは

- MagicalRecord
- MKNetworkKit
- RSOAuthEngine
- RSTwitterEngine
- NSDate-Utilities

である。ソースコードは付録 A.2 にある通り、Web で一般公開されている。

A.2 実装コード

Mac OSX の Xcode 5 上にて、Objective-C を用いて実装した。ソースコード等を含めた Xcode プロジェクトの各ファイルは、<https://github.com/storz/Notifauth> へ設置し、MIT ライセンスにより配布している。

A.3 画面一覧



図 A.2: Notifauth 起動時の画面



図 A.3: Notifauth ユーザ登録画面

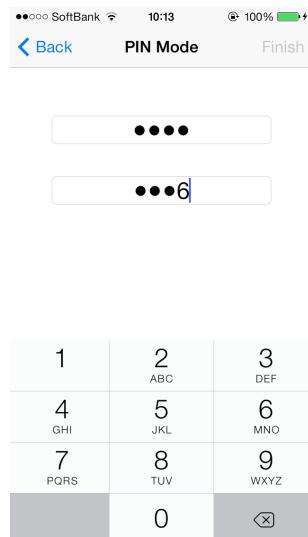


図 A.4: Notifauth 設定時の PIN 登録画面



図 A.5: Notifauth 認証終了時の画面

付録B

実験に関する付録

B.1 スケジュール番号

スケジュール番号	順番	スケジュール番号	順番
0	A B C D	12	C A B D
1	A B D C	13	C A D B
2	A C B D	14	C B A D
3	A C D B	15	C B D A
4	A D B C	16	C D A B
5	A D C B	17	C D B A
6	B A C D	18	D A B C
7	B A D C	19	D A C B
8	B C A D	20	D B A C
9	B C D A	21	D B C A
10	B D A C	22	D C A B
11	B D C A	23	D C B A

A : Auto Mode Type Term

B : Auto Mode Type Cycle

C : Manual Mode

D : PIN Mode

B.2 結果送信の詳細手順

1. トップ画面で「Send」をタップすると、iOS 標準のメール送信画面が開くので、何も編集を行わずに送信する。
2. ここで仮に iOS へ自分のメール情報(送信サーバ、アカウントなど)が登録されていない場合以下の手順を行う
 - (a) 「Send」をタップせず、トップ画面下部の「copy experiment data on clipboard」をタップする。
 - (b) クリップボードにデータがコピーされているので、メールアプリに貼り付けて実験担当者のメールアドレスへ送信する。

B.3 評価実験の概要説明資料

「Notifauth: Twitter の情報を利用した携帯端末の多要素化方式に関する提案」実験について

電気通信大学 情報理工学部

高田研究室 高浪 悟

・ 本実験の概要

本実験は「Twitter の情報を利用した携帯端末の多要素化方式に関する提案」の一環として行われるもので、被験者の方には、自身の Twitter アカウントを利用し、

1. 該当する期間を設定し自動で秘密の情報となる自分の投稿(以下ツイート)を絞り込む
 2. 該当する曜日・時間を設定し自動で秘密の情報となるツイートを絞り込む
 3. ツイートの一覧の中から手動で秘密の情報となるものを設定する
 4. パスワードの桁数を従来の 4 桁から 1 桁増やす
- の 4 つのパターンにおいて各 8 日の間に 4 回(0 日目、1 日目、3 日目、8 日目)、iOS のロック解除に似た操作を行っていただきます。想定される所要時間は合計およそ 20 分です。2 パターンが終了した時点と 4 パターンが終了した時点でアンケートにお答えいただきます。

・ 本実験の被験者に対する要件

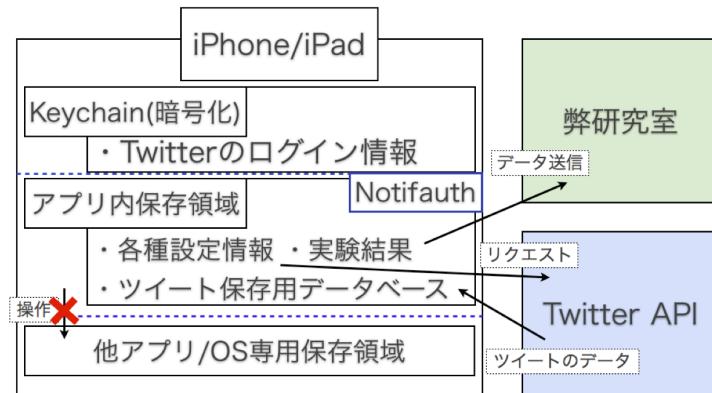
1. iOS 7 を搭載している端末を利用していること
2. Twitter アカウントを所持し、1 件以上投稿を行っていること
3. 1 月前半までに都内でお会いでき、アプリのインストール作業(20 分ほど)を行えること

・ ご協力頂ける方は

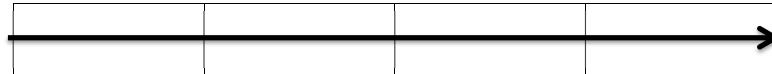
satorutakanami@gmail.comまでご連絡ください。
直近のスケジュールをお伺いします。

高浪 悟

・ 概略図



・ 実験の順番/スケジュール



・ 実施日程詳細

	1回目	2回目	3回目	4回目
Auto Mode Type Term (Auto 1)				
Auto Mode Type Cycle (Auto 2)				
Manual Mode				
PIN Mode				

B.4 Notifauth 操作マニュアル

Notifauth 実験操作マニュアル

Satoru Takanami
AZ-Lab UEC

研究内容

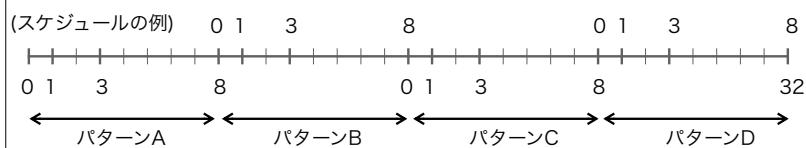
- Twitterの投稿は自分が能動的に行ったもの
 - ↳ 無意味な文字の羅列よりも憶えやすいのでは？
 - ✓ Twitterの情報をパスワードの代わりに利用
- SNSの情報はどんどん新しいものが追加されていく
 - ↳ 「条件」さえ設定すれば自動でパスワードが変わる認証が作れるのでは？
 - ✓ 実際に2種類の条件で絞り込める認証システムを作成
- 今回の実験は以上のものが本当に憶えやすく使いやすいかを確かめるものです

実験の流れ

1. 条件/パスコードを設定
2. 直後に1回目のテスト
3. 1日後、3日後、8日後にテスト
4. 1~3を繰り返し他のパターンを試す

実験のスケジュール

- ・ 被験者の方が何サイクル目にどのパターンを実験するかはアプリケーションインストール時にランダムで決められます
- ・ 8日目が終わったその日に新しいパターンを設定し、一度認証してもらいます



各パターン詳細

A.Auto Mode Type Term

- 日/週/月/年から△日~年間を指定し、その範囲に当てはまるツイートが鍵

B.Auto Mode Type Cycle

- 曜日の△時という条件に当てはまるツイートが鍵

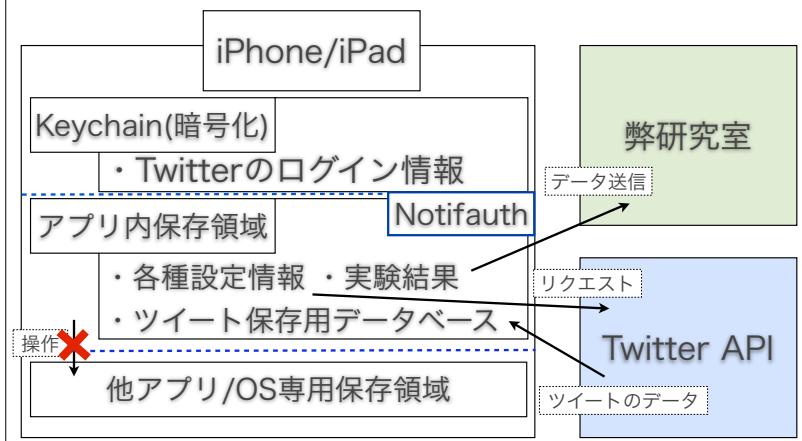
C.Manual Mode

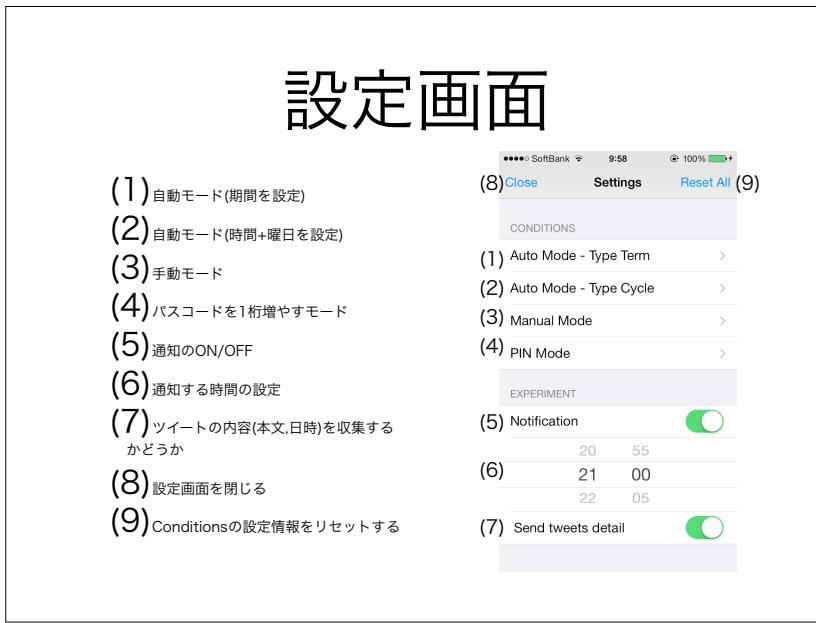
- 自分のツイートから任意に1つ鍵を選ぶ

D.PIN Mode

- 通常のパスコードを一桁増やしたもの

概略図





新規ユーザー追加

- 画面の表示に従ってTwitterのIDとパスワードを入力して下さい。
- ここでログイン情報はこちらでは一切視認/保管しません
- メイン画面の(2)を押せば全てのログイン/ツイート/設定データが消えます
- 更にご心配の場合はtwitter.com上から”このアプリケーションを許可しない”設定にして下さい(実験終了後)



自動モード[期間] 設定画面

- (1)どのくらい前かを設定します
- (2)(1)からどのくらいの期間かを設定します
- (3)鍵となりうるツイートの例を表示します(上は最も古いもの、下は最も新しいもの)
- (4)戻ります(保存されません)
- (5)次へ進みます



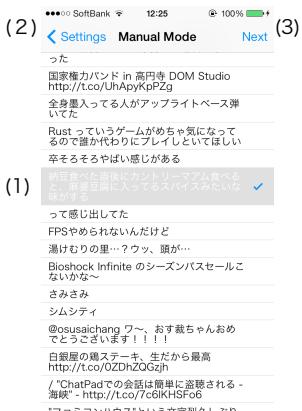
自動モード[周期] 設定画面

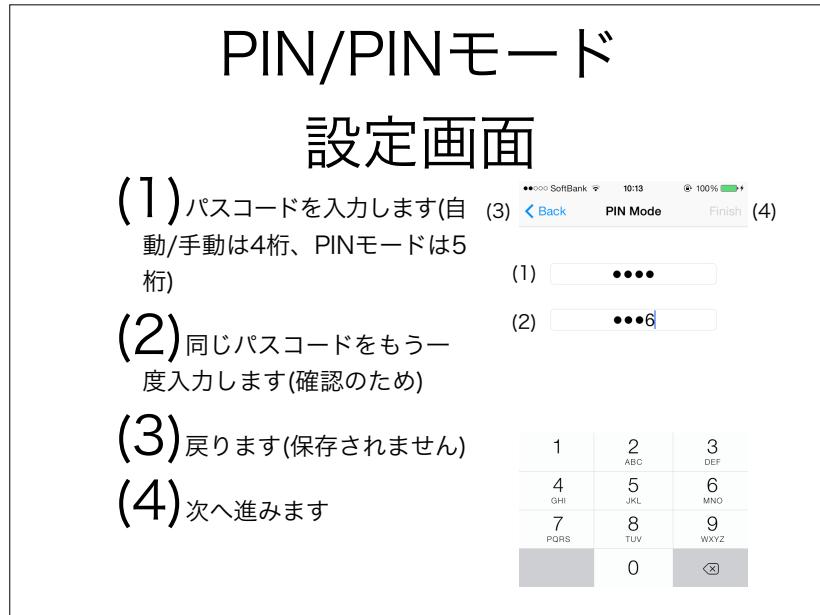
(1) 何時かを設定します
 (2) 曜日を設定します
 (3) 鍵となりうるツイートの例を表示します(上は最も古いもの、下は最も新しいもの)
 (4) ツイートが多い時間帯/曜日を提示します(タップでそれに設定を合わせる)
 (5) 戻ります(保存されません)
 (6) 次へ進みます



手動モード設定画面

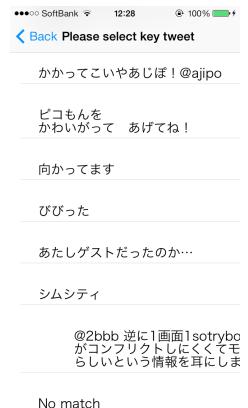
(1) 約200件表示されている自分のツイートの中から一つを選びます
 (2) 戻ります(保存されません)
 (3) 次へ進みます





実験[ツイート選択]

- iOSロック時の通知画面の
ように該当のツイートをス
ライドします
- 当てはまらない場合は
「No match」を選択し
てください



実験[パスコード入力]

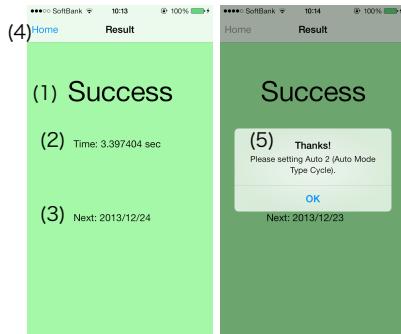
- iOSロック時のパスコー
ド入力画面のように入
力します
- タイプミスなどでや
り直す場合はBack
で戻り、ツイートの
選択部分からとなり
ます

1	2	3
4	5	6
7	8	9
0		

Cancel

実験[結果画面]

- (1) 結果が表示されます
- (2) 実験にかかった時間が表示されます
- (3) 次の実験日が表示されます
- (4) メイン画面に戻ります
- (5) 1サイクルが終わった時は右図のように次の実験パターンの指示が表示されます



データの送信について

- メイン画面の(6)を押すとメール作成画面が開くのでそのまま送信して下さい
 - もしiOS標準メールを使えない場合は、同画面(9)を押すとクリップボードにコピーされるので、他のメールアプリの本文部分に貼り付けして satorutakanami@gmail.comまで送信して下さい(その際できるだけお名前を添えて下さい)
- 送信のタイミングですが、毎回認証を終える毎でもよいですし、1サイクル終わった毎、気が向いた時でも構いません
 - ただし、2サイクル目が終わった段階で簡単なアンケートをとりたいのでその時は皆様送信をお願いします

B.5 評価実験における中間アンケート

Notifiauth評価アンケート（中間）

2014/01/15 14:32

Notifiauth評価アンケート（中間）

今回は、本実験にご協力いただき誠にありがとうございます。

既に2つのパターン(Auto Mode Type Term, PIN Modeなど)をやっていただいた方に、中間アンケートをとさせていただきます。

また本アンケートに入力・回答いただいた内容は、研究内容の向上と論文執筆以外の目的には使用いたしません。また氏名を回答頂いておりますが、個人を特定可能にしうる形でアンケート情報を公表することは決していません。不明な点がありましたら、高浪悟(satorutakanami@gmail.com)まで問い合わせ下さい。

*必須

1. お名前を教えて下さい *

実験の通知メールの冒頭に書かれているお名前(苗字など)で大丈夫です。

2. 性別を教えて下さい。 *

1つだけマークしてください。

- 男性
- 女性
- その他

3. 年齢を教えて下さい。 *

1つだけマークしてください。

- 10代
- 20代
- 30代
- 40代
- 50代以上

Notifiauth評価アンケート（中間）

2014/01/15 14:32

4. 携帯端末の利用状況について、過去一年間の利用状況を振り返り、携帯端末を利用しなかった間隔が最も長かったのはどのくらいですか？*

普段使っているものを全て合わせた回数をお答え下さい。およそで結構です。

1つだけマークしてください。

- 1～4時間 (ヒマさえあれば使っている)
- 12時間 (半日ぐらいは使わなかったことがある)
- 24時間 (丸一日使わなかったことがある)
- 2～5日間 (数日使わなかったことがある)
- 1週間 (1週間程度使わなかったことがある)
- 2週間 (2週間ぐらい携帯電話を使わなかったことがある)
- 1ヶ月以上 (1ヶ月くらい携帯端末を使わなかったことがある)

5. 普段最も使用頻度の高い携帯端末のロック解除方法は何ですか？*

1つだけマークしてください。

- なし (ロックしていない)
- PIN (数字のみのパスコード)
- 英数字のパスワード
- パターン (点をなぞるもの)
- 指紋認証
- その他:

6. 認証に使用したTwitterのスクリーンネーム(@○○の部分)を教えて下さい*

もしアカウントを教えてたくない場合は、Twitterの投稿頻度(1日あたり)をおおよそでいいのでお答え下さい。

7. Twitterはどのくらいの頻度で閲覧していますか？*

1つだけマークしてください。

- 1回未満 / 1日 (毎日必ずは見ていないが、思いつくと見る程度)
- 1～2回 / 1日 (毎日1回程度)
- 3～6回 / 1日 (朝昼晩とか決まったタイミングで見ている)
- 7～20回 / 1日 (1時間に1回前後は見ている)
- とにかくよく見ている、常時見ている / 1日

1週目に実験したパターンについて質問です

Notifiauth評価アンケート（中間）

2014/01/15 14:32

本アンケートを送った際のメールに記載してある、1週目に実験したパターンについての操作性・利便性・記憶持続性などについてのアンケートです。

8. 秘密情報の記憶保持にかかる負担はどのくらい感じますか？*

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい

9. 認証にかかる時間はどのように感じましたか？*

1つだけマークしてください。

1 2 3 4 5

とても短い とても長い

10. 認証を成功させるために必要な操作負担はどの程度でしたか？*

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい

11. 認証を行うのにどれくらいフラストレーションを感じましたか？*

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい

12. 設定画面について、使いづらさを感じましたか？

.....
.....
.....
.....
.....

Notifauth評価アンケート（中間）

2014/01/15 14:32

13. もしこのパターンを日常的に使用しなければならぬとしたらあなたはどんな改良を望みますか？

2週目に行ったパターンについて質問です

本アンケートを送った際のメールに記載してある、2週目に実験したパターンについての操作性・利便性・記憶持続性などについてのアンケートです。

14. 秘密情報の記憶保持にかかる負担はどのくらい感じますか？*

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい

15. 認証にかかる時間はどのように感じましたか？*

1つだけマークしてください。

1 2 3 4 5

とても短い とても長い

16. 認証を成功させるために必要な操作負担はどの程度でしたか？*

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい

17. 認証を行うのにどれくらいフラストレーションを感じましたか？*

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい

Notifiauth評価アンケート（中間）

2014/01/15 14:32

18. 設定画面について、使いづらさを感じましたか？

19. もしこのパターンを日常的に使用しなければならぬとしたらあなたはどんな改良を望みますか？

1週目で行ったパターンと2週目で行ったパターンの比較について質問です

20. どちらのパターンの方が認証操作が楽でしたか？ *

1つだけマークしてください。

- 1週目
 2週目

21. どちらのパターンの方が秘密保持しやすかったですか？ *

1つだけマークしてください。

- 1週目
 2週目

22. 今後日常的に使わなければならぬとすれば、どちらのパターンを使いたいですか？ *

1つだけマークしてください。

- 1週目
 2週目

以上でアンケートは終了です。フォームを送信して下さい。

B.6 Notifiauth評価実験における最終アンケート

2014/01/29 15:52

Notifiauth評価アンケート（最終）

今回は、本実験にご協力いただき誠にありがとうございました。

4つ全てのパターン(Auto Mode Type Term/Auto Mode Type Cycle/Manual Mode/PIN Mode)をテストして32日間にわたる実験を終了した方に最終アンケートをとさせていただきます。

また本アンケートに入力・回答いただいた内容は、研究内容の向上と論文執筆以外の目的には使用いたしません。また氏名を回答頂いておりますが、個人を特定可能にしうる形でアンケート情報を公表することは決していません。不明な点がありましたら、高浪悟(satorutakanami@gmail.com)まで問い合わせ下さい。

*必須

1. お名前を教えて下さい *

実験の通知メールの冒頭に書かれているお名前(苗字など)で大丈夫です。

ここからの6問は中間アンケートで答えていただいた方は回答不要です

その場合は次ページの「Auto Mode Type Term(Auto 1)について質問です」まで進んで下さい。

2. 性別を教えて下さい。

1つだけマークしてください。

- 男性
- 女性
- その他

3. 年齢を教えて下さい。

1つだけマークしてください。

- 10代
- 20代
- 30代
- 40代
- 50代以上

Notifiauth評価アンケート（最終）

2014/01/29 15:52

4. 携帯端末の利用状況について、過去一年間の利用状況を振り返り、携帯端末を利用しなかった間隔が最も長かったのはどのくらいですか？

普段使っているものを全て合わせた回数をお答え下さい。およそで結構です。

1つだけマークしてください。

- 1～4時間 (ヒマさえあれば使っている)
- 12時間 (半日ぐらいは使わなかったことがある)
- 24時間 (丸一日使わなかったことがある)
- 2～5日間 (数日使わなかったことがある)
- 1週間 (1週間程度使わなかったことがある)
- 2週間 (2週間ぐらい携帯電話を使わなかったことがある)
- 1ヶ月以上 (1ヶ月くらい携帯端末を使わなかったことがある)

5. 普段最も使用頻度の高い携帯端末のロック解除方法は何ですか？

1つだけマークしてください。

- なし (ロックしていない)
- PIN (数字のみのパスコード)
- 英数字のパスワード
- パターン (点をなぞるもの)
- 指紋認証
- その他:

6. 認証に使用したTwitterのスクリーンネーム(@○○
の部分)を教えて下さい

もしアカウントを教えてくれない場合は、Twitterの
投稿頻度(1日あたり)をおおよそいいでお答え
下さい。

7. Twitterはどのくらいの頻度で閲覧していますか？

1つだけマークしてください。

- 1回未満 / 1日 (毎日必ずは見ていないが、思いつくと見る程度)
- 1～2回 / 1日 (毎日1回程度)
- 3～6回 / 1日 (朝昼晩とか決まったタイミングで見ている)
- 7～20回 / 1日 (1時間に1回前後は見ている)
- とにかくよく見ている、常時見ている / 1日

*これらの質問は、一部中間アンケートでも回答して頂いておりますが、すべての認証手法の実験を
終わった上で、皆様の意見をあらためてお聞かせ頂きたく、全ての手法に対するアンケートをとらせて頂
いております。

Notifiauth評価アンケート（最終）

2014/01/29 15:52

自由記述に関しては、中間アンケートと同じ内容であればその旨を記載していただければ大丈夫です。
ご協力のほどよろしくお願いします。

Auto Mode Type Term(Auto 1)について質問です

Auto Mode Type Termの操作性・利便性・記憶持続性などについてのアンケートです。

秘密情報の記憶保持にかかる負担はどのくらい感じますか？*

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい

認証にかかる時間はどのように感じましたか？*

1つだけマークしてください。

1 2 3 4 5

とても短い とても長い

認証を成功させるために必要な操作負担はどの程度でしたか？*

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい

認証を行うのにどれくらいフラストレーションを感じましたか？*

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい

Notifiauth評価アンケート（最終）

2014/01/29 15:52

タイピングしたりタッチパネルをスライドしたりする作業の負担はどの程度でしたか？ *

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい

13. このパターンで一度でも認証に失敗しましたか？そのときルール(期間や時間曜日)を憶えていましたか？ *

1つだけマークしてください。

- 失敗したことがある、その時ルールを憶えていなかった
- 失敗したことがある、その時ルールは憶えていた
- 一度も失敗しなかった

14. 設定画面について、使いづらさを感じましたか？

15. もしこのパターンを日常的に使用しなければならぬとしたらあなたはどんな改良を望みますか？

Auto Mode Type Cycle(Auto 2)について質問です

Auto Mode Type Cycleの操作性・利便性・記憶持続性などについてのアンケートです。

Notifiauth評価アンケート（最終）

2014/01/29 15:52

秘密情報の記憶保持にかかる負担はどのくらい感じますか？*

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい**認証にかかる時間はどのように感じましたか？***

1つだけマークしてください。

1 2 3 4 5

とても短い とても長い**認証を成功させるために必要な操作負担はどの程度でしたか？***

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい**認証を行うのにどれくらいフラストレーションを感じましたか？***

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい**タイピングしたりタッチパネルをスライドしたりする作業の負担はどの程度でしたか？***

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい

Notifiauth評価アンケート（最終）

2014/01/29 15:52

21. このパターンで一度でも認証に失敗しましたか？そのときルール(期間や時間曜日)を憶えていましたか？*

1つだけマークしてください。

- 失敗したことがある、その時ルールを憶えていなかった
 失敗したことがある、その時ルールは憶えていた
 一度も失敗しなかった

22. 設定画面について、使いづらさを感じましたか？

.....
.....
.....
.....
.....
.....

23. もしこのパターンを日常的に使用しなければならぬとしたらあなたはどんな改良を望みますか？

.....
.....
.....
.....
.....

Manual Modeについて質問です

Manual Modeの操作性・利便性・記憶持続性などについてのアンケートです。

秘密情報の記憶保持にかかる負担はどのくらい感じますか？*

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい

Notifiauth評価アンケート（最終）

2014/01/29 15:52

認証にかかる時間はどのように感じましたか？*

1つだけマークしてください。

1 2 3 4 5

とても短い とても長い

認証を成功させるために必要な操作負担はどの程度でしたか？*

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい

認証を行うのにどれくらいフラストレーションを感じましたか？*

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい

タイピングしたりタッチパネルをスライドしたりする作業の負担はどの程度でしたか？*

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい

29. 設定画面について、使いづらさを感じましたか？

Notifiauth評価アンケート（最終）

2014/01/29 15:52

30. もしこのパターンを日常的に使用しなければならぬとしたらあなたはどんな改良を望みますか？

PIN Modeについて質問です

PIN Modeの操作性・利便性・記憶持続性などについてのアンケートです。

秘密情報の記憶保持にかかる負担はどのくらい感じますか？*

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい

認証にかかる時間はどのように感じましたか？*

1つだけマークしてください。

1 2 3 4 5

とても短い とても長い

認証を成功させるために必要な操作負担はどの程度でしたか？*

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい

Notifiauth評価アンケート（最終）

2014/01/29 15:52

認証を行うのにどれくらいフラストレーションを感じましたか？*

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい

タイピングしたりタッチパネルをスライドしたりする作業の負担はどの程度でしたか？*

1つだけマークしてください。

1 2 3 4 5

とても小さい とても大きい

36. 設定画面について、使いづらさを感じましたか？

37. もしこのパターンを日常的に使用しなければならないとしたらあなたはどんな改良を望みますか？

全パターンにおける比較について質問です

38. どの認証操作が楽でしたか？ *

楽な順にランクをつけて下さい。

1 行につき 1 つだけマークしてください。

1位 2位 3位 4位

Auto Mode Type Term	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Auto Mode Type Cycle	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Manual Mode	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
PIN Mode	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

39. どのパターンが秘密保持しやすかったですか？ *

しやすかった順にランクをつけて下さい。

1 行につき 1 つだけマークしてください。

1位 2位 3位 4位

Auto Mode Type Term	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Auto Mode Type Cycle	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Manual Mode	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
PIN Mode	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

40. 今後日常的に使わなければならぬとすれば、どのパターンを使いたいですか？ *

使いたい順にランクをつけて下さい。

1 行につき 1 つだけマークしてください。

1位 2位 3位 4位

Auto Mode Type Term	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Auto Mode Type Cycle	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Manual Mode	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
PIN Mode	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

認証操作についての評価について質問です

Notifiauth評価アンケート（最終）

2014/01/29 15:52

ツイートを選ぶ際はスライドではなくタップ(タッチするだけ)のほうがいいと思う *

1つだけマークしてください。

1 2 3 4 5

全くそう思わない とてもそう思う**ツイートを全文表示しない方がいいと思う ***

なぜ？：ツイートを全文表示すると一度に表示できる量が減り、探す手間が増えたりしますが、そうしないときちゃんと認識できないツイート(アスキーアートなど)も存在します。そういう点を問題ないと感じている人がどれくらいいらっしゃるかを調べるために質問です。

1つだけマークしてください。

1 2 3 4 5

全くそう思わない とてもそう思う**表示するツイートの数はもっと多いほうがいいと思う ***

なぜ？：表示するツイートの数が多い方が安全性が高まりますが、その分探す手間やハッキリと思い出せなかった場合の負担が増えます。そこを問題だと感じている人がどれくらいいらっしゃるかを調べるために質問です。

1つだけマークしてください。

1 2 3 4 5

全くそう思わない とてもそう思う**この認証方法についての質問です。(自由回答)**

44. ツイートを秘密情報として使用することをどう思いますか？

Notifiauth評価アンケート（最終）

2014/01/29 15:52

45. それぞれのパターンについて感じた印象はありますか？

以上でアンケートは終了です。フォームを送信して下さい。

Powered by
 Drive