



平成 25 年度 卒業論文

Twitter を用いた携帯端末における  
個人認証プロセス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

高田研究室

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、その普及に際しては、利便性の面から問題点がある。

本研究では、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損なうことなく、ユーザに負担の少ない多要素化手法を提案することを目的とした。

本研究では、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) 時系列上における日付と曜日の指定、の 3 つの秘密情報の設定方法を持ったアプリケーションソフトウェアを開発し、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　　のような影響があり、(2) と (3) ではそれぞれ　　のような結果が得られた。更に、考えうる問題点として × × が挙げられ、具体的な解決方法についても考察した。

# 目 次

第 1 章 序論	8
1.1 背景 . . . . .	8
1.2 研究目的 . . . . .	9
1.3 論文の構成 . . . . .	10
第 2 章 個人認証の多要素化への流れ	11
2.1 既存の認証技術 . . . . .	11
2.1.1 知識認証 . . . . .	11
2.1.2 所有物認証 . . . . .	12
2.1.3 生体認証 . . . . .	13
2.2 多要素認証 . . . . .	16
2.3 スマートフォン/タブレットの普及 . . . . .	17
第 3 章 関連研究/製品	19
3.1 多要素認証についての調査 . . . . .	19
3.1.1 二要素認証のユーザビリティに関する比較調査 . . . . .	19
3.2 認証の多要素化手法 . . . . .	19
3.2.1 Google . . . . .	20
3.2.2 PassBan . . . . .	21
3.2.3 Authy . . . . .	22
3.2.4 オンラインゲームにおける多要素化例 . . . . .	23

第 4 章 動機と提案 (仮タイトル)	25
4.1 手軽な多要素化手法の開発 . . . . .	25
4.2 携帯端末への多要素認証の導入 . . . . .	26
4.3 ライフログや SNS の利用 . . . . .	26
4.3.1 既存手法 . . . . .	27
4.4.1 提案手法の概要 . . . . .	30
4.4.1 Twitter についての説明 . . . . .	32
第 5 章 Twitter 上の情報を用いた提案認証システム	34
5.1 システムの概要 . . . . .	34
5.1.1 秘密情報の設定 . . . . .	34
5.1.2 認証操作 . . . . .	36
5.1.3 前提条件 . . . . .	38
5.2 実装の詳細 . . . . .	39
5.3 具体的特徴 . . . . .	41
5.3.1 時間経過による秘密情報の変化 . . . . .	41
第 6 章 検証実験	43
6.1 概要 . . . . .	43
6.1.1 実験手順 . . . . .	43
6.2 SNS の情報を利用することに関する評価実験 . . . . .	45
6.2.1 目的 . . . . .	45
6.2.2 方法 . . . . .	46
6.2.3 結果 . . . . .	46
6.3 時系列における期間を秘密として用いることに関する評価実験 . . . . .	46

6.3.1 目的 . . . . .	46
6.3.2 方法 . . . . .	46
6.3.3 結果 . . . . .	47
6.4 時系列における周期を秘密として用いることに関する評価実験 . . . . .	47
6.4.1 目的 . . . . .	47
6.4.2 方法 . . . . .	47
6.4.3 結果 . . . . .	47
<b>第 7 章 考察</b>	<b>48</b>
7.1 安全性に関する考察 . . . . .	48
7.2 憶えやすさに関する考察 . . . . .	48
7.3 使用継続性に関する考察 . . . . .	49
7.4 他環境における応用に関する考察 . . . . .	49
<b>第 8 章 結論</b>	<b>50</b>
<b>謝辞</b>	<b>51</b>
<b>参考文献</b>	<b>52</b>
<b>付録 A 実装に関する付録</b>	<b>54</b>
A.1 実装コード . . . . .	54
A.2 画面一覧 . . . . .	54
<b>付録 B 実験に関する付録</b>	<b>56</b>
B.1 スケジュール番号 . . . . .	56
B.2 評価実験の概要説明資料 . . . . .	58

B.3 Notifauth 操作マニュアル	60
B.4 評価実験における中間アンケート	63
B.5 評価実験における最終アンケート	68

## 図 目 次

2.1 Google における ID とパスワードの入力画面 . . . . .	13
2.2 Apple iOS におけるタッチパネルによる PIN の入力画面 . . . . .	14
2.3 USB キーの例 . . . . .	14
2.4 静脈を用いた認証のための装置 . . . . .	15
2.5 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	18
3.1 Google Authenticator のワンタイムパスワード表示画面 . . . . .	21
3.2 PassBoard の設定画面 . . . . .	22
3.3 Authy のトークン表示画面 . . . . .	23
3.4 ハードウェアトークンの例 . . . . .	24
3.5 トークン生成アプリケーションの例 . . . . .	24
4.1 Facebook における友人の顔写真を用いた認証画面 . . . . .	30
4.2 Twitter における Timeline 画面 . . . . .	33
5.1 Auto Mode Type Term の設定画面 . . . . .	35
5.2 Auto Mode Type Cycle の設定画面 . . . . .	35
5.3 Manual Mode の設定画面 . . . . .	37
5.4 ロック画面上における通知の選択 (スライド) 動作の例 . . . . .	38
5.5 ロック画面における通知の表示画面を模した認証画面 . . . . .	39
5.6 ロック画面における PIN の入力画面を模した認証画面 . . . . .	39
5.7 Notifauth のクラス図 . . . . .	40
A.1 Notifauth 起動時の画面 . . . . .	55

A.2 Notifauth ユーザ登録画面 . . . . .	55
A.3 Notifauth 設定時の PIN 登録画面 . . . . .	55
A.4 Notifauth 認証終了時の画面 . . . . .	55

## 表 目 次

5.1 必要環境等 . . . . .	40
---------------------	----

# 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<sup>\*1</sup>(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憶えることが難しく、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]、

---

<sup>\*1</sup> 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 SMS(Short Message Service<sup>\*2</sup>) 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るとなる。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<sup>\*3</sup>(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。

---

<sup>\*2</sup>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

<sup>\*3</sup>Social Networking Service、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 , 近年普及した技術が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 5 章 Twitter 上の情報を用いた提案認証システム : 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少ないため、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。そのため、以下のような欠点が存在する。

- ユーザへ強い記憶負担が大きい
- 認証のための秘密情報入力に際して負担が大きい
- 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、Ashlee Vance[4]によれば、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。

### 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、

- 入力を行うことのユーザの負担が小さい
- 所有物を交換することで秘密情報を容易に変更可能
- 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、

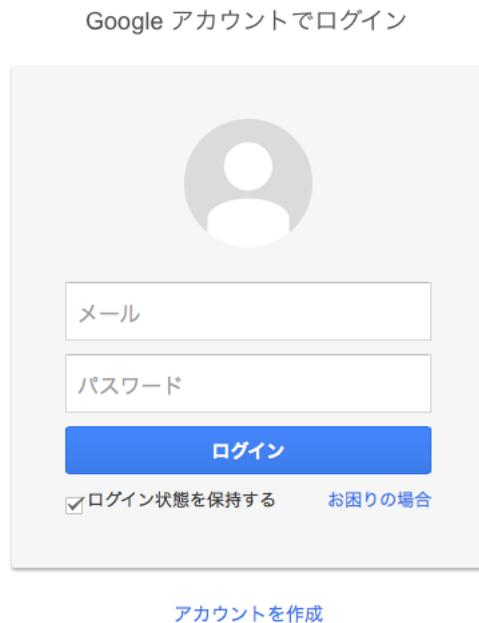


図 2.1: Google における ID とパスワードの入力画面

- 認証の際に手元に所有してい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
- 認証に特殊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 2.3），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 2.2: Apple iOS におけるタッチパネルによる PIN の入力画面



図 2.3: USB キーの例

- 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
- 認証に特殊な機器を必要とするため、導入のコストが高い
- 体質や外部からの影響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。この認証方式の具体例として、指紋、静脈(図2.4)、虹彩を用いたものが挙げられる。



図 2.4: 静脈を用いた認証のための装置

## 2.2 多要素認証

既存の認証手法を複数組み合わせることで、欠点を補い、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して利用している方式が主流である[5]。

銀行(例:ジャパンネット銀行[6])やオンラインゲーム(例:Battle.net[7])などでも多く見られる[5][8]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、多くの金融にかかわらない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。そういうた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<sup>\*1</sup>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乱数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に乱数が表示される。この方式のメリットとして、新たな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。

多要素認証は、中間者攻撃やトロイの木馬を用いた攻撃、フィッシングに対して耐性が強くないことや、サービスプロバイダが負担するコストが大きいといったデメリットも存在する。

<sup>\*1</sup> 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)

実例としての多要素化手法やワンタイムパスワードの生成方式については、第3章で述べる。

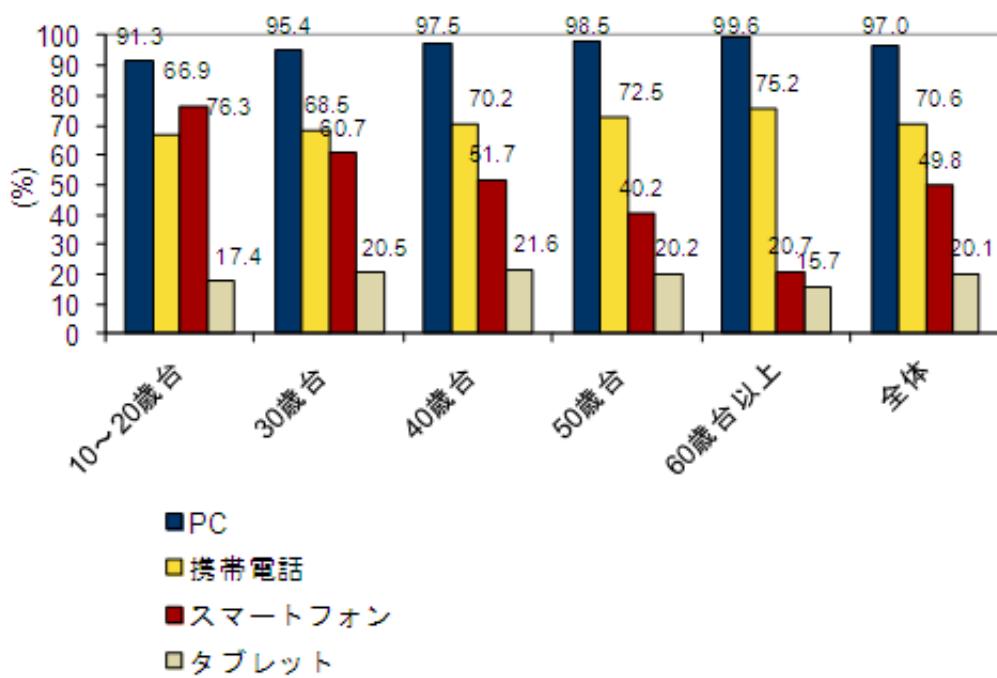
### 2.3 スマートフォン/タブレットの普及

2013年6月に行われたIDC Japanの調査[9]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<sup>\*2</sup>の所有率は20.1%であった(図2.5)。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。

2.2章や3.2章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。

---

<sup>\*2</sup>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 2.5: 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

## 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 [5] 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(使いやすさ、信頼性、認識努力)の評価を行った。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設計を行わなければならぬとした。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ことも明らかにした。

#### 3.2 認証の多要素化手法

多要素認証においては、ワンタイムパスワードが多く使われる。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 (RFC 6238<sup>\*1</sup>による)
- 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
- E メールや SMS を使用するもの：E メールや SMS などを経由してワンタイムパスワードを送信する手法

などが一般的である。具体的な応用例は以下に示す。

### 3.2.1 Google

Google では，アカウントにログインする際に複数の多要素化方式を用意している。一つは E メール/SMS を用いてワンタイムパスワードを送信する手法であり，これはログインの際に ID/パスワードの入力が正しいものであれば携帯端末へ送信される。もう一つの方式として，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 3.1)を公開しており，こちらはユーザ固有の秘密鍵とサーバからのメッセージを用いて 30 秒ごとに SHA1<sup>\*2</sup>を用いた HMAC(Hash-based Message Authentication Code<sup>\*3</sup>)を生成し，6 衔の数字コードに変換している。

---

<sup>\*1</sup> Time-Based One-Time Password Algorithm

<sup>\*2</sup> 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

<sup>\*3</sup> 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。

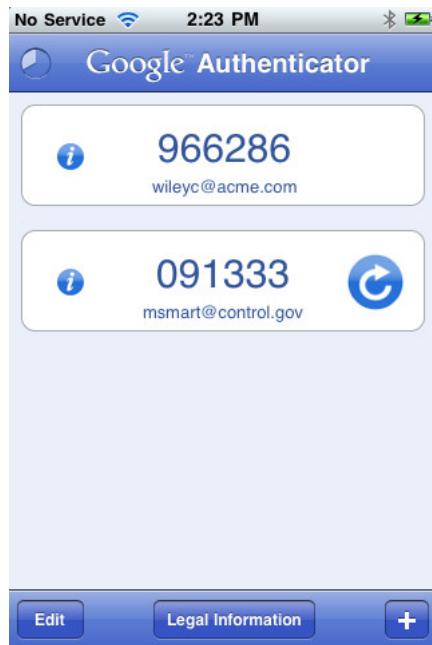


図 3.1: Google Authenticator のワンタイムパスワード表示画面

### 3.2.2 PassBan

PassBoard<sup>\*4</sup> というアプリケーションソフトウェアは、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(図 3.2)。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いる。

---

<sup>\*4</sup> 米 PassBan 社により提供

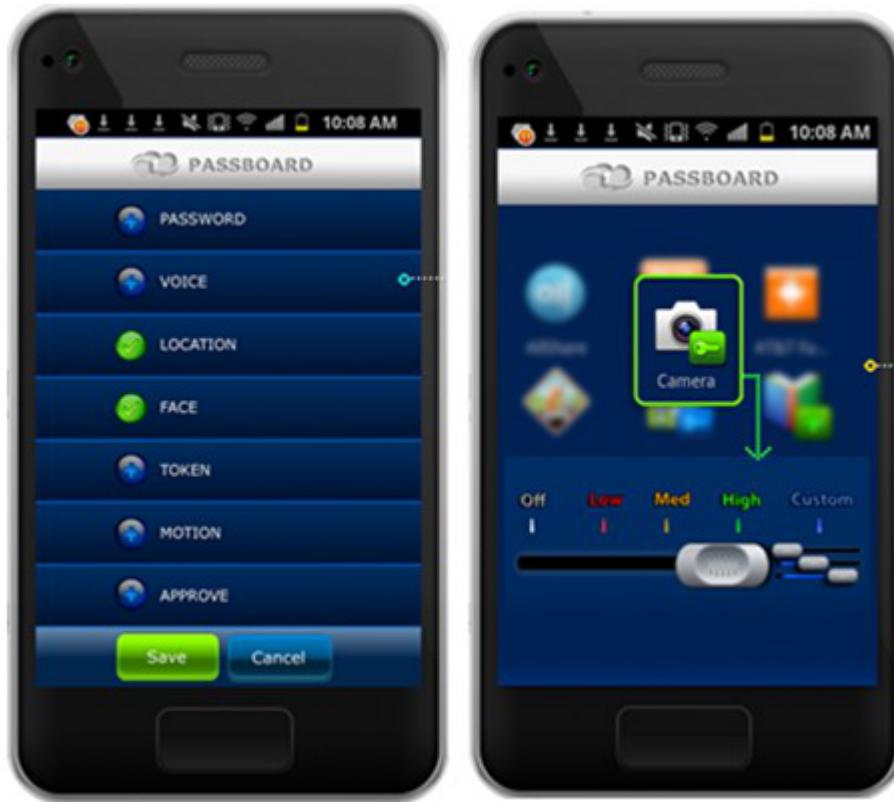


図 3.2: PassBoard の設定画面

### 3.2.3 Authy

Authy<sup>\*5</sup>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<sup>\*6</sup>接続や WordPress<sup>\*7</sup>へのログインも二要素化が可能となる。Authy に紐付けた Web サービスへログインする際は、通常の手順に加え 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 3.3)を入力することで、ログインが完了する。

<sup>\*5</sup>[ここに Authy の説明がります]

<sup>\*6</sup>Secure SHell

<sup>\*7</sup>オープンソースのブログソフトウェア

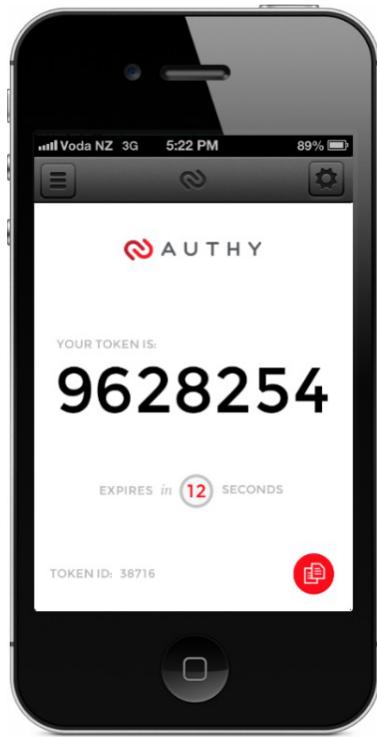


図 3.3: Authy のトークン表示画面

#### 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、ハードウェアトークン(図3.4)による認証の多要素化が普及している[5]。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[8]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、小型の液晶画面にそれを表示したものをログイン時にIDとパスワードの後に入力させることで行っている。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3.5)が用意されていることもある。



図 3.4: ハードウェアトークンの例

図 3.5: トークン生成アプリケーション  
の例

## 第4章

# 動機と提案(仮タイトル)

### 4.1 手軽な多要素化手法の開発

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)コストに関しては、サービスプロバイダが負担するコストがユーザが負担するコストが多要素化手法によって様々に存在する。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、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それぞれ強いられる。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えた。

また、(2)利用可能な状況に関しては、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、そうではない様々な場面でも使える多要素化の方法を模索する必要があるとえた。

## 4.2 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<sup>\*1</sup>が搭載されていることが多い、Webサービスなどにおいても、ブラウザ上からだけでなく、専用のアプリケーションソフトウェアが用意されている場合がある。そういう場面では、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<sup>\*2</sup>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、実際に多要素認証を適用できるのではないかと考えた。

## 4.3 ライフログやSNSの利用

個人認証の方法を提案するにあたって、強度を高めることによって利便性(憶えやすさ、使いやすさ)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<sup>\*3</sup>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、通信の高速化や端末の高速化により、マルチメディアの共有(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おり、その

---

<sup>\*1</sup> 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

<sup>\*2</sup> 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。

<sup>\*3</sup> 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。

中でも特に利用率が高いのは SNS である (ここにそれらしい調査結果への cite) . 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 .

#### 4.3.1 既存手法

ライログや Web サービスを用いた認証では , 以下の様なものが検討・実装されている ,

##### Web 履歴を用いた認証

田村ら [10] 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 [11] 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 [12] 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 [13] 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という工夫がなされた。さらに、基礎実験の後に重要でない故に記憶に残っていないメールをフィルタリングするために曖昧な回答を許可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 [14] らは、Twitter のダイレクトメッセージ<sup>\*4</sup>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？」といった文面で送信された。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<sup>\*5</sup>では、友人の顔写真を表示し本名を回答させることを要求する認証が運用されている。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が存在するため、それ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。

<sup>\*4</sup> 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。

<sup>\*5</sup> 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 4.1: Facebook における友人の顔写真を用いた認証画面

## 4.4 提案手法の概要

前節の各既存手法の問題点を解決するためには、それらを 3 つに大別した上で、それぞれについて以下のような改善策を用意できると考えた。

- 安全性が損なわれる状況が存在する：特定の趣向や環境に依存しにくい情報を利用する
- 認証時に問題が生じる：ある程度公開されている情報を用いたり、イレギュラーをフィルタリングしやすいように文字情報を主として用いる
- 利便性について提案以前の状態から改善できていない：能動的に憶えるのではなく、憶えていることを認証に利用する

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 上にある自分のツイートを利用することで上記の改善策を取り入れることができると考えた．積極的理由として，

1. 能動的な行為によって生成される情報であり，記憶のための負担に配慮可能のこと
2. 生成された日時の詳細が確実に取得でき，時系列を提示することにより記憶を思い出しやすいこと

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．

- 音楽を用いて認証を行う方法
  - 外部の騒音などにより認証を行いにくい場面が存在する
  - 趣味趣向に大きく依存してしまう
- 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
  - 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

また，時系列における情報を保持していることの特徴として，時間情報によって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ものがある．また，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 5.3 節にて示す．

#### 4.4.1 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上の情報はほとんどがタイムライン(図4.2)<sup>\*6</sup>に表示される短文の投稿(「ツイート」と呼ばれる)であり、それら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)に設定されていれば、フォロー<sup>\*7</sup>を許可された人物(フォロワー<sup>\*8</sup>)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。Twitterでは以上のように“public”と“protected”的2つの公開範囲が存在する[15]。

<sup>\*6</sup>投稿が時系列によって表示される画面

<sup>\*7</sup>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

<sup>\*8</sup>自分のことをフォローしている他のユーザ

**Tweets**

---

 **Electroni Kurokawa** @kkshow 3s  
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](#)

---

 **DOT a.k.a pico.RIPE** @DawnSong 5s  
どれや  
[Expand](#) [Reply](#) [Retweet](#) [Favorite](#) [More](#)

---

 **KIZAN518** @KIZAN518 9s  
エリア移動6秒は我慢するかなあ...  
[Expand](#) [Reply](#) [Retweet](#) [Favorite](#) [More](#)

---

 **ふりす** @ind\_fris 11s  
荻窪トマト今まで生きてて一番うまいと思った8800円ぽワイン超えてるうまさだった  
[Expand](#) [Reply](#) [Retweet](#) [Favorite](#) [More](#)

---

 **KUNIO** @kunio9209 18s  
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、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](#)

---

 **igi** @igi 25s  
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 4.2: Twitter における Timeline 画面

## 第 5 章

# Twitter 上の情報を用いた提案認証システム

### 5.1 システムの概要

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て、利便性（憶えやすさ/使いやすさ）と安全性の両立を目指した個人認証手法を実装した（以下 Notifauth）。Notifauth 起動時の画面は図 A.1 のようになっており、この画面から新規登録画面（図 A.2）<sup>\*1</sup>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。

#### 5.1.1 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、

Auto Mode Type Term

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（図 5.1）

---

<sup>\*1</sup> Twitter と連携するため OAuth を用いた

### Auto Mode Type Cycle

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(図 5.2)

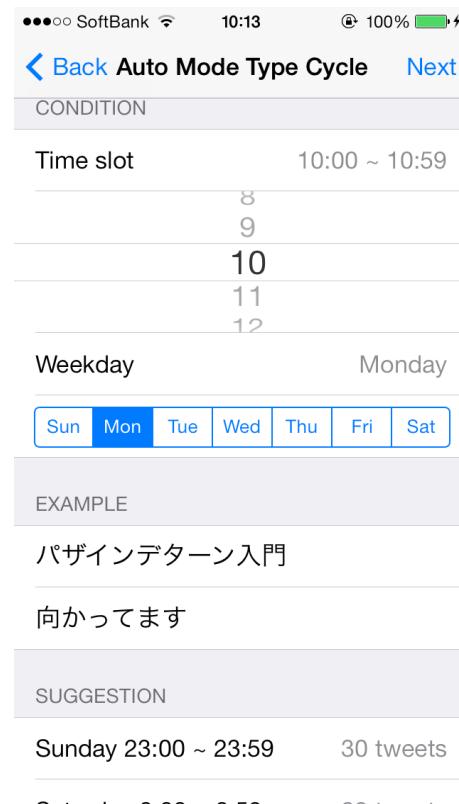
### Manual Mode

自分のツイートから任意に1つ秘密情報となるものを選ぶ(図 5.3)

以上の3つを実装した。



図 5.1: Auto Mode Type Term の設定画面



次に、各設定方法の概要を説明する。

Auto Mode Type Term では、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の中央値のもの)を表示し、ユーザの設定を補助する。

Auto Mode Type Cycle では、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、セレクターを用いて「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、画面中断の「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の中央値のもの)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の設定を補助する。

Manual Mode では、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。

### 5.1.2 認証操作

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 5.4<sup>\*2</sup>)を踏襲したものを探用した。理由として、

1. 本システムは携帯端末における認証の多要素化を目指して実装され、その際

<sup>\*2</sup>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



図 5.3: Manual Mode の設定画面

開発環境である iOS でそういった操作を行えるのはロック画面のみであったため

2. 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、ユーザーへ新たな操作を覚えさせる負担が少ないと考えたため

が挙げられる。また、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 5.5, 図 5.6)をアプリケーション内に実装した。

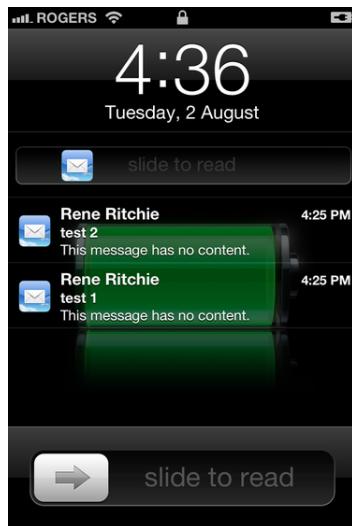


図 5.4: ロック画面上における通知の選択(スライド)動作の例

### 5.1.3 前提条件

システムを利用するのに必要な条件や、実装の際に用いた環境などを表 5.1 に記す。



図 5.5: ロック画面における通知の表示画面  
面を模した認証画面

図 5.6: ロック画面における PIN の入力画面  
面を模した認証画面

## 5.2 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 5.7 の通りである。

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

表 5.1: 必要環境等

必要条件	iOS7 を利用し、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
実装環境	Mac OSX 10.9, Xcode5
動作確認環境	iPhone 4, iPhone 4S, iPhone 5, iPhone 5S, iPod touch

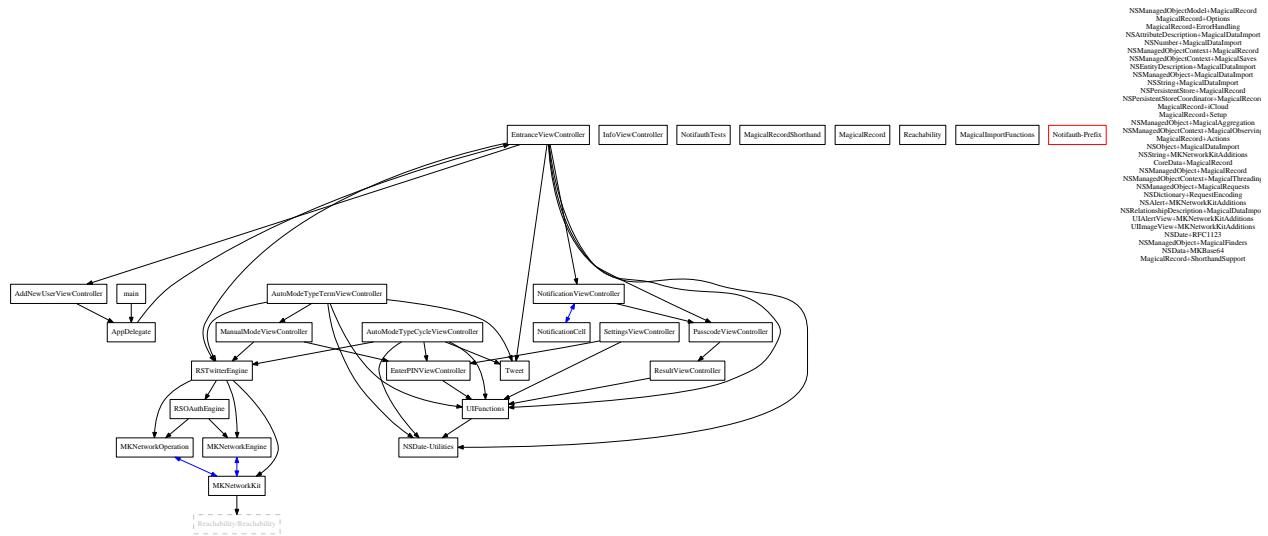


図 5.7: Notifauth のクラス図

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.1 にある通り、Web で一般公開されている。

## 5.3 具体的特徴

### 5.3.1 時間経過による秘密情報の変化

Auto Mode Type Termにおいて、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、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。

また、Auto Mode Type Cycleにおいては、

- 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

ということも利点として考えられる。

欠点としては以下のようなものが挙げられる。

- ユーザの本人認証率が下がる可能性がある
- 期間の設定やツイートの頻度によっては、ダミー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

本研究では、以上の利点が本当に作用するかどうかの検証実験も行った。

## 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、3つの評価実験を行った。それらの実験は、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をせず、一度に行った。

##### 6.1.1 実験手順

以降の節のそれぞれの実験は第 5.1.1 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 衔の 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 衔にしたものと秘密情報をとする」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。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。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

通り割り振って管理する。

#### 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う。
2. 不明な点があれば質問してもらう。
3. 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する。
4. 実験担当者の開発用端末と被験者の携帯端末を接続し、Appleの開発者用アカウントと被験者の端末の紐付けを行った上でNotifauthをインストールする。
5. 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う。
6. 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する。
7. 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う。
8. 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ののち、初回実験説明・導入の終了とする。

#### 試行手順

1. トップ画面で、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. “PIN Mode” 以外の場合，ロック画面を模した画面が表示され，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 の入力画面が表示され，“PIN Mode” であれば 5 行，それ以外のパターンであれば 4 行の PIN を入力する．
4. 結果画面が表示されるので，「Home」をタップする．

#### 結果送信手順

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．

## 6.2 SNS の情報を利用することに関する評価実験

### 6.2.1 目的

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 方法

付録の B.4 や B.5 にある通り，被験者には使用パターンについて 5 段階のリッカート尺度を使った質問に答えてもらい，さらに既に 8 日間の試行が終了している他パターンとの比較もしてもらう。

### 6.2.3 結果

(ここに最高の結果が入る)

## 6.3 時系列における期間を秘密として用いることに関する評価実験

### 6.3.1 目的

本実験では，SNS の情報の特性を利用した認証システムの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。本実験で評価対象とするパターンとして，“Auto Mode Type Term” を採用する。

### 6.3.2 方法

付録の B.4 や B.5 にある通り，被験者には使用パターンについて 5 段階のリッカート尺度を使った質問に答えてもらい，さらに既に 8 日間の試行が終了している他パターンとの比較もしてもらう。

### 6.3.3 結果

(ここに最高の結果が入る)

## 6.4 時系列における周期を秘密として用いることに関する評価実験

### 6.4.1 目的

本実験では、SNS の情報の特性を利用した認証システムの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。本実験で評価対象とするパターンとして，“Auto Mode Type Cycle” を採用する。

### 6.4.2 方法

付録の B.4 や B.5 にある通り、被験者には使用パターンについて 5 段階のリッカート尺度を使った質問に答えてもらい、さらに既に 8 日間の試行が終了している他パターンとの比較もしてもらう。

### 6.4.3 結果

(ここに最高の結果が入る)

## 第 7 章

### 考察

#### 7.1 安全性に関する考察

安全性に関しては，単純な組み合わせにおいては PIN による方式を上回り，さらに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。更に，hoge により hage といったことが考えられる。しかし，設定情報をいかに秘匿するかといった面では，暗号化などの改善を行う必要がある。

#### 7.2 憶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憶えやすさを向上させることが主たる目的として存在した。しかし，被験者実験において，秘密情報の設定方法によっては憶えやすさが低下するという結果が得られたため，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。

### 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、被験者実験で得られた感想などから見ても、利便性についての評価が高いので、使用継続性が高いと考えられる。

### 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、設定の柔軟さから、携帯端末以外の環境でも応用が可能だと考えられる。被験者実験にて実施したアンケートでは、「　などに導入したい」といった意見を得ることができた。

## 第 8 章

### 結論

本論文では、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では、従来の知識認証のメリットを生かしつつ、新たな特徴を併せ持った認証要素を、様々な部分で応用できると考えている。また、被験者実験によって問題点の洗い出しと、今後の方向性の手がかりを得ることができた。しかしながら、実験に関しては手法などに問題点が多かったため、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、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] 2段階認証プロセスについて - google アカウント ヘルプ. <https://support.google.com/accounts/answer/180744>, 2014-01-15.
- [2] Dropbox - アカウントで 2段階認証を有効にするには。. <https://www.dropbox.com/help/363>, 2014-01-15.
- [3] 2段階認証を全ユーザが利用可能に — evernote 日本語版ブログ. <http://blog.evernote.com/jp/2013/10/05/15717>, 2014-01-15.
- [4] Ashlee Vance. If your password is 123456, just make it hackme. <http://www.nytimes.com/2010/01/21/technology/21password.html>, 2010.
- [5] 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.
- [6] ワンタイムパスワードのご案内|ジャパンネット銀行. <http://www.japannetbank.co.jp/security/security/otp.html>, 2014-01-15.
- [7] Battle.net authenticator - battle.net support. <https://us.battle.net/support/en/article/battlenet-authenticator>, 2014-01-15.
- [8] 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.

- [9] 浅野 浩寿 and 木村 融人. 2013 年国内モバイル／クライアントコンピューティング市場家庭ユーザー利用実態調査：ブランド認知度と購買行動の変化.  
<http://www.idcjapan.co.jp/Report/Pc/j13180103.html>, 2013.
- [10] 健範 田村, 和宏 鶴丸, 将嗣 市野, and 尚久 小松. Web 閲覧履歴情報に着目したライフ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.
- [11] 容子 長谷, 輝勝 青木, and 浩 安田. M-068 スケジュールと gps 情報を利用した認証方法の検討 (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.
- [12] 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.
- [13] 正勝 西垣 and 誠 小池. 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.
- [14] Tomofumi Nemoto, Kyohei Furukawa, and Manabu Okamoto. Poster: Knowledge-based authentication using twitter. Symposium On Usable Privacy and Security 2011, 2011.
- [15] Twitter help center — 公開と非公開ツイートについて. <https://support.twitter.com/articles/243055>, 2014-01-15.

## 付録 A

# 実装に関する付録

### A.1 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.2 画面一覧



図 A.1: Notifauth 起動時の画面



図 A.2: Notifauth ユーザ登録画面

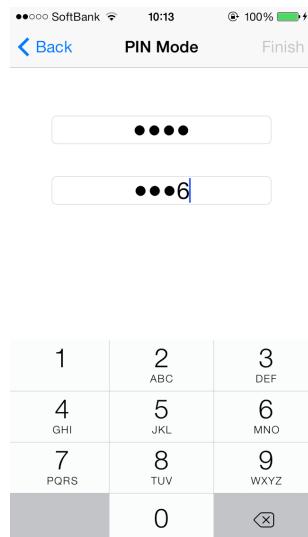


図 A.3: Notifauth 設定時の PIN 登録画面



図 A.4: 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 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

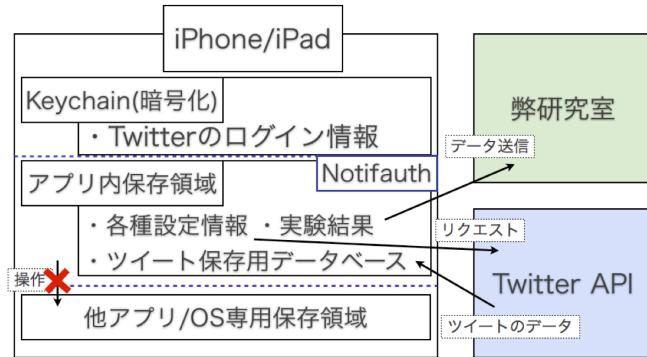
#### ・ご協力頂ける方は

[satorutakanami@gmail.com](mailto: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



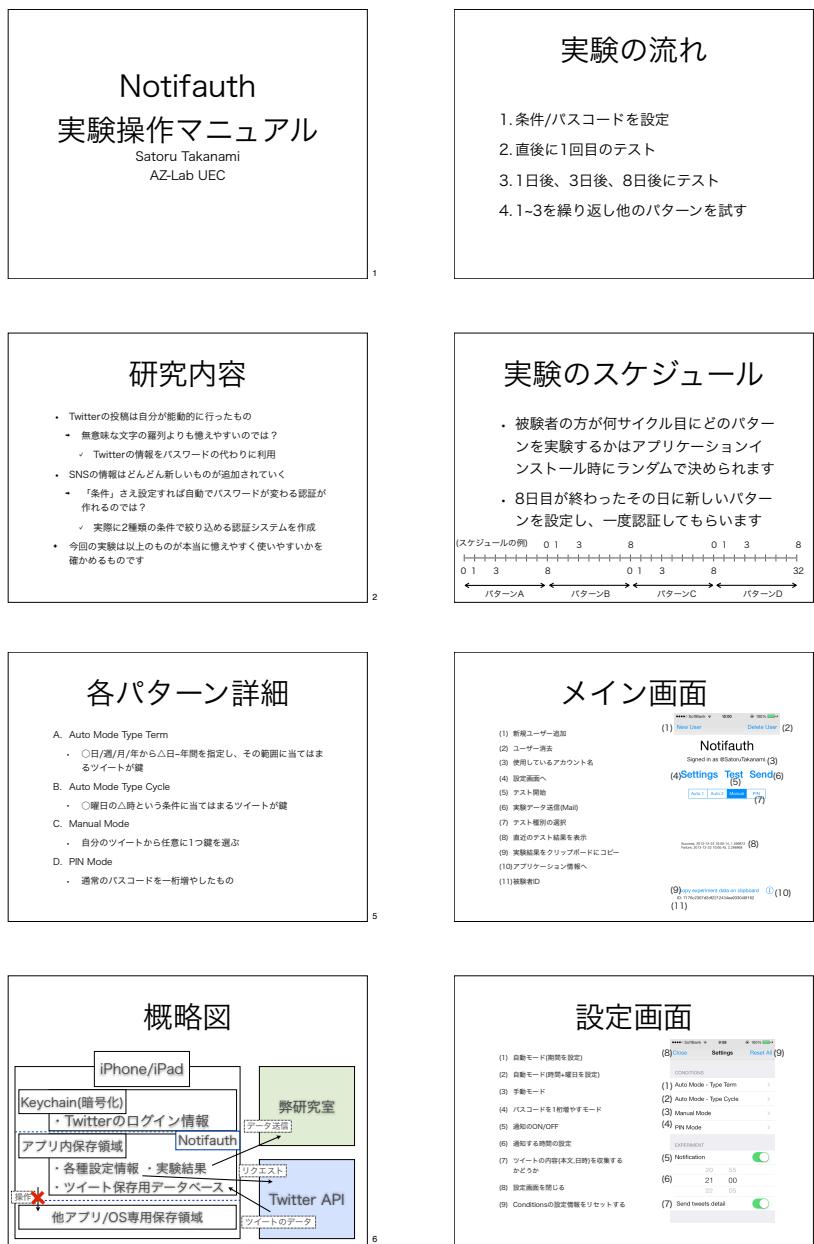
## ・ 実験の順番/スケジュール

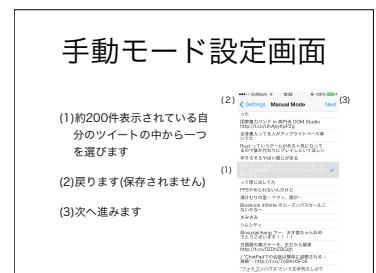
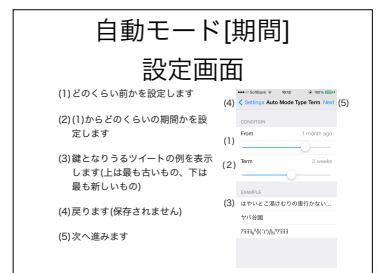
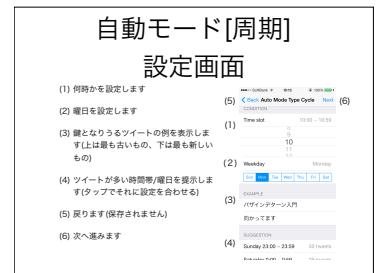
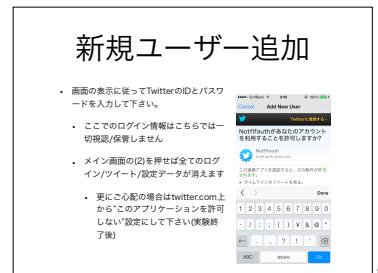


## ・ 実施日程詳細

	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				

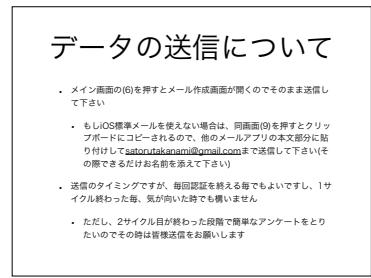
## B.3 Notifauth 操作マニュアル







17



18

## B.4 評価実験における中間アンケート

Notfiauth評価アンケート（中間）

2014/01/15 14:32

### Notfiauth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らせていただきます。

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たしません。不明な点がありましたら、高浪悟([satorutakanami@gmail.com](mailto: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(朝屋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

## B.4 評価実験における中間アンケート

65

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？

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

Notifiauth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	<input type="radio"/> 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	<input type="radio"/> 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	<input type="radio"/> 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	<input type="radio"/> 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.5 評価実験における最終アンケート