スマートフォンを「鍵」としたウェブサイトへの自動ログイン機構の提案

藤川真一[†1] 山内正人[†1] 砂原秀樹[†1]

最近，ウェブサイトが不正アクセスされパスワードが流出する事件が多発している．その際，流出したパスワードが他のウェブサイトと同じパスワードを使用している利用者が多いため，連鎖的に他のウェブサイトへも不正ログインされるという被害が頻発している．これは，1)簡単なパスワードが用いられておりクラックを受けやすいこと，2)セキュリティホール等により不正ログインが可能である場合があること，3)複数のウェブサイトで同一のパスワードを使っていることなどが問題である．解決法として，ウェブサイト毎に別のパスワードを使うことや，２段階認証などを啓蒙している．しかし，ユーザの意識に期待する手法は，多様な IT リテラシーの利用者が存在する現状において妥当性のある解決法ではない．また理解が難しいログイン方法はユーザビリティの低下を招き普及の障壁になっている．本稿ではこの問題を改善するために，ユーザビリティを下げることなく，既存システムの改修を最小限にしたログイン管理システム「SmartLogin Key」を提案する．これは，スマートフォンのアプリケーションにパスワードの自動発行管理を行わせるものである．プロトタイプを実装し，ユーザ登録が容易となり，自動ログインが実現可能であることを確認した．

The proposal of the automatic login mechanism to the website which used the smart phone as the "key"

SHINICHI FUJIKAWA[†1] MASATO YAMANOUCHI[†1] HIDEKI SUNAHARA[†1]

Nowadays, many web sites have been cracked and passwords are stolen by cracker. The stolen passwords are used for accessing to the other web sites. The problems are caused by 1) using easy password, 2) using same passwords for the other web sites.
To solve these problems, using different passwords for each web site, 2-Step Verification are said to be effective. However, relying on users themselves for this problem and using a log-in complicated system cannot be said for the best solutions because they push responsibility to the users. In this research, we develop a log-in system using smart phones to purpose for a) not pushing so many tasks to the users, b) redeveloping existed systems minimally. The system is called SmartLogin Key. As the result, comfortable user registrations and automatic login could be showed using a prototype we developed.

1. 背景

昨今，ウェブサービスがクラックされ，パスワードを含むユーザ情報が流出し，不正に取得されたユーザ情報から芋づる式に別のウェブサービスに不正ログインされてしまう問題が頻発している．最初はウェブサービス側のシステム上の不具合がきっかけとして発生することが多い．例として，連続ログインの処理が不適切でブルートフォースアタックによる解析を許してしまう[1]，データベースに保存されているパスワードの不適切なハッシュ化，またウェブアプリケーションやサーバ設定の不備などが挙げられる[2]．そこから流出したログイン情報と同じパスワードを使っている他のウェブサービスに不正ログインされてしまう．これはユーザが同じパスワードを複数のウェブサイトで使用することで，インターネットという分散環境の中で，実質的なシングルサインオンシステムになっていることが問題である．しかし，通常のウェブサービスの利用規約では，ログイン情報の管理はユーザの責任になっていることから，パスワードが他のウェブサービスと共通化されていることについて積極的に解決しようとするウェブサービス事業者は少ない．

各ウェブサービス事業者は，不正ログインの被害を受けた後に，ユーザにパスワードを他のウェブサイトとは別のパスワードに変更するように指導しているが，多くのウェブサイトにてパスワードの入力文字数は 8 文字以上を理想としており[3]，複雑なパスワードの利用をユーザの記憶に頼って運用することは非現実的である．パスワードによるログインを強化する手段として，スマートフォンや携帯電話とを組み合わせた 2 段階認証が存在するが，こちらは多くのユーザにとって概念を理解することが困難であり，普及しているとは言いがたい．

パスワードシステムは，システム組み込みのコストが安価に済む上に，ユーザの認知として十分普及しているために，現時点ではシステム事業者，ユーザ側共に最も使いやすいログインシステムである．そのため現状のパスワードシステムを維持したまま，パスワードの個別化をサポートする仕組みが必要であると考えた．

†1 慶應義塾大学メディアデザイン研究科
Graduate school of media design , Keio University

2. 問題の解決についての提案

2.1 スマートフォンを「鍵」とした自動ログイン機構

既存のパスワードシステムを活用しつつ,ユーザビリティを損なうことなくパスワード個別化を実現するために，スマートフォンを「鍵」としたウェブサービスへの自動ログイン機構を提案する．現在，多くの人が所有するスマートフォンには電子メールや SNS，ネットバンキング等のアクセス情報など, 様々のパーソナルデータが紐付いており，肌身離さず持ち歩くデバイスになっている．スマートフォンを始めとするスマートデバイスが，インターネットを通じて別の端末上のアクセス認証を行えるようになることで,「鍵」のような役割を担えると考える.

具体的な例として，以下のものを提案する．PC のウェブサービスにログインする際に，既存のパスワード入力と並行し，ログイン画面に QR コードを表示する（図 1）.この QR コードを読み込んだスマートフォンのアプリケーションが，ウェブサービスのログイン管理サーバと連携し，アプリケーションが自動でパスワードを発行し，ユーザ登録およびログイン処理を実行する．アプリケーションは，ユーザ登録時にランダムにパスワードを発行することで，ユーザが意識することなくウェブサービス毎にパスワードを個別化する．再ログイン時にも，同じ QR コードを読み込みアプリケーションがログイン処理を行う.

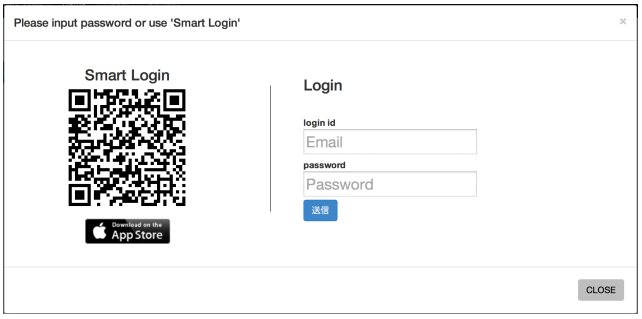

図 1　QR コードを表示したログイン画面の例

2.2 先行事例

スマートフォンのアプリケーションでウェブサイトのログイン情報を管理する仕組みについては，類似の先行事例が存在する.

- 1password
- tiqr
- SQRL(Secure Quick Reliable Login)

1password は, マスターパスワード 1 つ登録することで，複数のパスワードを一括管理するツールである. PC やスマートフォンにパスワードを保存し，パスワードの管理を行う．ランダムパスワードを発行することで，パスワードの個別化が可能である.

tiqr は，QR コードを活用した認証の仕組みで，ユーザ体験およびバックエンドの通信フローは本提案に非常によく似ている．OATH と呼ばれる認証プロトコルに基づいた実装になっている.

SQRL の認証の仕組みも本提案と似たようなユーザ体験で構成されている．QR コードに記載されている URL がワンタイムキーとして認証を行う仕組みである.

2.3 本提案の特徴および解決すること

本提案が先行事例と比べた特徴と，既存のパスワードログインシステムに対して解決することは以下の通りである.

1) 本提案はツールではなく, ログイン機構であり仕組みの提案である．スマートフォン上のアプリケーションを鍵とみなし，他のデバイス上のウェブサイトのログイン情報を関連付ける部分が本質で，スマートフォン上のアプリケーションの実装形態は本論文で実装したアプリケーションに依存しない.

2) ウェブサイト毎に個別のパスワードを自動発行することで，パスワード情報流出の影響範囲を最小化する.

3) ユーザがパスワードを忘れてしまってもスマートフォンさえあればウェブサイトにログインできる.

4) パスワードの管理責任は今までどおりユーザに委ねる.

5) 既存のパスワードログインシステムをそのまま流用する．本提案を導入する際のウェブサービスの改修は最小限で済むように設計されており，データのやりとりもシンプルである．現存するデータベースの構造を変えず，サーバ側にいくつかの機能追加のみで対応可能である.

3. 実現の方法

3.1 本提案の実験，実装について.

本提案を検証するために，以下のシステムを開発した.

- スマートフォンをウェブサイトの鍵とみなし，ログインアカウント, パスワードを管理するアプリケーション「Smartlogin Key（以下，本アプリと記載する)」
- ウェブサービスと連携するためのサーバサイドの仕組み．既存のパスワードによるログインに加えて，ログイン画面に QR コードを表示し，アプリケーションが QR コードを読み込むことで，自動のユーザ登録やログイン処理を行う.

3.2 「Smartlogin key」アプリの機能概要

本アプリは，以下の機能で構成される.

a) ウェブサイトが発行する QR コードを読み込み，QR コードに記載されたログインポリシーの URL から, ログインポリシーを取得し，アプリ内のデータベースに保存する機能

b) 取得したログインポリシーに記載された内容を元に，ウェブサイトに対して，ユーザ登録を行う機能や，自動でログインを行うサーバ連携機能
c) 登録済みのウェブサイトを一覧し，ログイン情報の編集や削除等を行える機能
d) ユーザ登録時に，自動送信するためのプロフィール設定機能
e) ログインポリシーに記載された拡張属性に対応するために，パスワード変更やサービス退会機能や，その他，プライバシーポリシーや利用規約などを表示する機能

なお PC 側の端末を識別するために QR コードに含まれる URL には端末識別情報が追加されている．

3.3 ログインポリシー

各ウェブサイトが発行する QR コードには，ログインポリシーを配信する URL を埋め込む. ログインポリシーとは，ウェブサイト毎のユーザ登録に必要な情報を記載したデータである．本アプリでは，データベースに保存されている過去のログイン情報と照合し，ネームスペースとバージョン番号が一致したログイン情報でログインを試み，情報が見つからない場合は，サービス新規登録を行う．表 1 にログインポリシーのフォーマットを示す.

表 1　ログインポリシーのフォーマット

	名称	データ例
1	ネームスペース	example.com
2	バージョン番号	1.0
3	サービス名称	Example Web Service
4	サービスの説明	自分だけの素敵お店リスト作成サービス
5	サービスのアイコン	http://example.com /icon.png
6	新規登録 API URL	https://api.example.com /user/register
7	ログイン API URL	https://api.example.com /user/login
8	ログイン ID 種別（メールアドレスまたは任意文字列）	“mail_address”（メールアドレス）“any”
9	ログイン ID 利用可能文字列	A-Za-z0-9+-（正規表現）
10	パスワード利用可能文字列	A-Za-z0-9+-（正規表現）
11	パスワード最小文字数	7
13	パスワード最大文字数	13
14	利用規約 URL	http://example.com /terms
15	プライバシーポリシーURL	http://example.com /privacy
16	ログイン情報変更 URL	https://api.exmaple.com/user/edit
17	退会 API URL	https://api.example.com /user/resign

本ログインポリシーの記載内容の詐称を防ぐためにポリシーファイルの配布サーバのドメインとポリシーファイル内のネームスペースが一致しない場合，本ファイルは無効とする．以下にログインポリシーのサンプルを示す.

```
{"policy_version":1.0,
 "lang":"ja",
  "domain":"example.com",
    "service_name":"##name##",
    "service_description":"##description##",
    "service_icon":"http://example.com/icon.png",
    "register_url":"https://api.example.com/user/register",
    "login_url":"http://api.example.com/user/login",
    "id_type":"any",
    "login_usable_character":"A-Za-z+-",
    "passwd_usable_character":"A-Za-z+-",
    "passwd_count_min":6,
    "passwd_count_max":13,
    "term_url":"http://example.com/terms",
    "privacy_url":"http://example.com/privacy"
 }
```

3.4 ウェブサービスに追加する機能

すでに存在するウェブサイト，サービス，アプリケーションにおいて，データベース構造やパスワードの仕様を変更せず，以下の機能を新規に追加するだけで，本システムを実現する．
i) ポリシーファイルの URL を配信する QR コードを生成し，既存のログイン画面に掲載する．
ii) ログインポリシーを配信する API
iii) 新規ユーザ登録を受け付ける API
iv) ID，パスワードを受信し，ログイン処理を受け付ける API
v) ログイン処理が完了した後に，ウェブブラウザ画面をログイン完了画面に導く機能.

これら 5 点がウェブサービスに必要な機能である．その他，実用段階においては登録内容の変更機能や，退会機能

等を提供する API が必要と考えられるが，本論文では言及しない．

3.5 自動ログインの処理フロー

本アプリによる自動ログインは，図 2 に示すシーケンスでログイン処理を行う．本アプリは QR コードからログインポリシーを読み込み，API を通じて自動で認証を行い，認証完了時にウェブサーバからブラウザにログイン通知を送る流れで構成している．

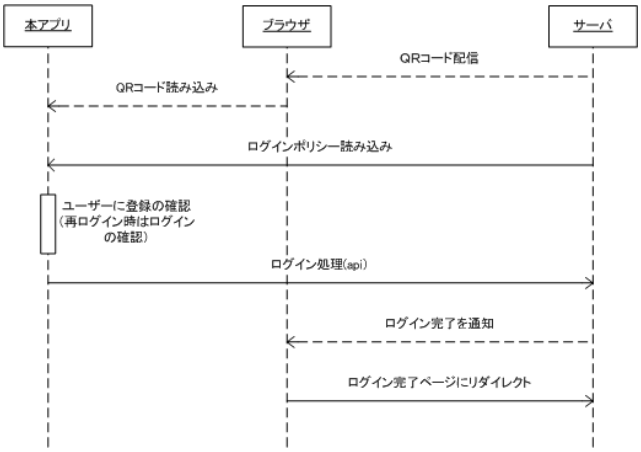

図 2　ログインシーケンス

ユーザ新規登録の際には，パスワード文字列をログインポリシーに定められた仕様に沿ったランダム値を自動生成する．この情報をアプリ側が管理することで，ユーザをパスワード管理の煩わしさから解放されると共に，特定のウェブサービスがクラックされた際の被害を最小化する．

4. 結果，実装について

本研究の実装環境を表 2 に示す．

表 2　動作確認環境

種別	名称
PC	MacBook Pro Intel Core i7 2.4GHz メモリ 16GB
OS	Mac OSX 10.8.4
Web サーバ	Ruby on rails 3.1.3
DB	MySQL 5.5.27
Socket サーバ	Node.js v 0.8.6 [4]
ブラウザ	Google Chrome 31.0.1632.4
スマートフォン	iPod touch (iOS7.0.1)

ログイン処理を行うウェブシステムは，ログイン画面を表示するウェブサーバと，ログイン処理完了後に完了画面を表示する Node.js サーバで構成されている．
ウェブブラウザが読み込んでいる HTML5 [5]上で動作している WebSocket が，Node.js サーバと接続しているため，ログイン完了時にサーバからプッシュ通知を行うことができ，それを検知したウェブブラウザが自動的にログイン完了画面に導いている．

これら機能を用いて，ログイン画面（図 3）に表示されている QR コードをアプリで読み込み（図 4），自動ログイン処理を行うことで，ウェブページを自動，かつ速やかにログイン完了画面（図 5）に遷移させることに成功した．

図 3　ログイン画面

図 4　スマートフォンアプリによる
QR コード読み込み

図 5　ログイン完了画面

各機器間の構成を図 6 に示す．

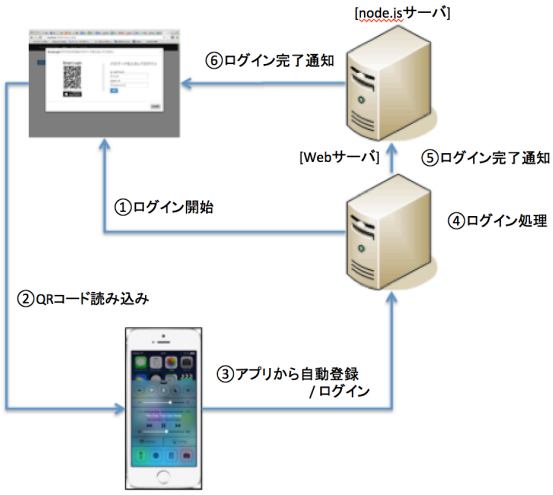

図 6　サーバ構成

5. 考察

本提案のシステムを使うことで，ウェブサイト毎にパスワードを自動生成しながらも，ユーザに負担をかけることなくウェブサービスへのログインが可能になった．ユーザの利便性という観点において，スマートフォンアプリケーションから QR コードを読み込むだけでログインできるログイン管理ツールとして機能しており，ウェブサイト毎に異なるパスワードを覚えることに比べ，簡便な方法で日常のログイン機能が実現できた．

このシステムの問題点は，スマートフォンと Web サーバの間には，インターネットを介して接続する以外にシステム上の繋がりがないことである．また QR コードは，配信するサーバのドメイン等に依存せず単体で配布可能な媒体のため，悪意のある攻撃者によって，全く違う場所にある他人の PC にログイン情報を送信してしまう問題も想定できる．また，ログインポリシー自体が詐称される懸念においても議論が必要だと考える．

6. 課題

今後の課題として，QR コードに含まれる端末識別情報が詐称された時に，悪意のある利用者の端末にログインしてしまうことを防ぐ検証が必要と考える．また，ログインポリシーの詐称可能性についても対応していく必要がある．

スマートフォンが壊れてしまった時やアプリを削除してしまった時に，もとに戻す仕組みが別途必要になる．クラウドストレージ上に暗号化したデータでバックアップをとっておくことが有力な方法だが，情報を復元するためにパスワードを使用してしまうと，１つのマスターパスワードですべてのログイン情報を管理しているのと同じ状態になってしまうため，パスワード文字列に委ねるのは避けるべきである．

代替となる手段としては，秘密の質問を活用する方法[6]や，記憶からパスワードを生成できる EpisoPass[7]と連携するなどし，特定のパスワードに依存しない情報から復元させる仕組みの採用が必要と考えている．

7. まとめ

今回は，日本人に親しまれている QR コードを用いて，スマートフォンによる自動ログインシステムのプロトタイプを作成した．パスワードは多くのユーザに定着しており，QR コードは，iOS の Passbook や，アジアの E コマースで活用されるなど認知が高まっていることから[8]，シンプルでユーザ認知が高い仕組みのため普及させやすいインターフェスと考えられる．

スマートフォン専用のウェブサービスについては，本アプリから直接ログインしてしまう仕組みを追加すれば対応可能である．

スマートフォンをウェブサイトの「鍵」となる重要な情報を保存するという行為に対するセキュリティの懸念においては，スマートフォンに今後，指紋認証等が搭載されることでより解決することが期待される．

また前述した通り，QR コードは配信するドメインと QR コードに掲載されているウェブサービスのドメインとは紐付かないことが懸念材料ではあるが，セキュリティの懸念が解決してさえいれば，電信柱の交通広告，イベントのポスター，雑誌やショッピングバッグに印刷した QR コードからウェブサービスに簡単にアクセスできるように発展可能なため，今後出てくる Google Glass 等のスマートデバイスなどの画像入力が可能で，入力に制約があるインターネット接続デバイスとも親和性が高い仕組みだと考える[9]．今後，詳細な課題を解決した上で，本仕様を公開し，ウェブサービスを運営する企業に提案していきたいと考えている．

謝辞

本研究は JSPS 科研費 24650031 の助成を受けたものです．

参考文献

1) Yan, J and Blackwell, A. and Anderson, R. ,Grant, A.: Password memorability and security: Empirical results, Security and Privacy, IEEE, Vol. 2, No. 5, PP. 25-31 (2004)
2) 八津川 直伸，石野 貴子：重大な脅威に対するセキュリティ設計手法の考察 UNISYS TECHNOLOGY REVIEW”，ユニシス技報，日本ユニシス, Vol. 98, No.3 (2008)
3) 山田 純一 :パスワードの安全性,富山大学総合情報基盤センター広報, Vol.10: pp.44-47 (2013)
4) Node.js
http://nodejs.jp/ (2013 年 9 月現在)
5) HTML5
http://www.whatwg.org/specs/web-apps/current-work/multipage/ (2013 現在)
6) 平野 亮,森井 昌克: パスワード運用管理に関する考察および提案とその開発 ，信学技報, Vol. 111, no. 286, LOIS2011-47, pp. 129-134 (2011)
7) EpisoPass - 記憶からパスワードを生成
http://episopass.com/ (2013 年 9 月現在)
8) 『QR コードに関する調査』を東アジア主要 4 カ国・地域でスマートフォン利用者を対象に実施
http://www.gmo.jp/news/article/?id=3953(2013 年 9 月現在)
9) QR コードで「Google Glass」にハッキング
http://wired.jp/2013/07/19/glass-new-vulnerabilities/ (2013 年 9 月現在)