「マルチメディア，分散，協調とモバイル
(DICOMO2019)シンポジウム」 令和元年7月

パスワードに乱数を組み合わせるユーザ認証方式

渡邉 悠雅[1]　鈴木 秀和[1]　内藤 克浩[2]　渡邊 晃[1]

概要： ユーザ認証を行う際にはパスワードのみでなく別の認証要素を組み合わせることでセキュリティを高めることが多い．しかし，多要素認証では認証専用機器が必要であることや，認証手順が複雑になるという課題がある．また，古いパスワードを利用していると，辞書攻撃により簡単にパスワードを割り出されるという課題もある．本稿では認証要素としてユーザデバイス内に保存した乱数を認証要素として用いる方式を提案する．パスワードと乱数でハッシュ値を算出し、このハッシュ値をサーバに登録するパスワードとして扱わせる．提案方式をセキュリティと使い勝手の両面で既存の多要素認証と比較し評価した．また，提案方式では辞書攻撃耐性が十分高いことを示した．

User authentication Method Combining Random Number with Password

YUGA WATANABE[1]　HIDEKAZU SUZUKI[1]　KATSUHIRO NAITO[2]
AKIRA WATANABE[1]

1. はじめに

インターネットが普及したことにより，ユーザの認証が必要になる機会が増えている．認証時のセキュリティは不正アクセスや情報漏えいから個人を守るために非常に重要である．ユーザ認証には数多くの方法があるが，パスワードを使った認証方式は最も基本となる方式である．しかし，現在ではパスワードのみを認証の鍵とするユーザ認証は，セキュリティが不十分とされており，多要素認証を行うことが多い．多要素認証とは性質の異なる複数の認証要素を組み合わせる認証方式である．認証要素はその性質により知識要素，生体要素，所持要素に分類される．多要素認証で組み合わせる認証要素は，それぞれの弱点を補完しあうことができるため，異なる分類のものを組み合わせたほうがよいとされている， パスワードは知識要素であるため，生体要素や所持認証と組み合わせられることが多い[1]．本稿では，パスワードとの組み合わせ要素として，どのような方式がよいかという観点で検討を進めた．

ユーザ認証ではセキュリティの高さが非常に重要であるが，ユーザの使い勝手も重要である．しかし，ユーザ認証においてセキュリティと使い勝手はトレードオフの関係にあり，両立させることが難しい．多要素認証は認証手順が増える分ユーザに手間と煩わしさを与えやすい．例えば，暗号通貨取引所では，パスワードのほかに OTP(One Time Password)を使う二段階認証を推奨している[2]．しかし，設定や操作の煩わしさから OTP 認証の設定をしないユーザが多く，アカウントが不正にアクセスされる事件が多い．

一方，不正アクセスや情報漏えいを防ぐためには，ユーザサイドのセキュリティだけでなく，サーバサイドのセキュリティを考えることが重要である．サーバサイドから大量の顧客情報が流出する事件は数多く，管理者のミスによる人災や，システムの欠陥が問題視されている[3]．また，管理者側に悪意を持った者が潜んでいる場合もある．セキュリティを考える上では，情報漏えいをさせないことが重要であるが，外部に情報が漏れる前提で，攻撃者に秘匿情報が分からないようにすることも大切である．

辞書攻撃は，辞書に載っている単語をひたすら照合しパスワード解析を行う攻撃であり，最も多く使われる解析攻撃の一つである．一般にユーザが設定するパスワードは不規則な文字列ではなく，モノの名前や人名，有名キャラクター名などユーザが覚えやすいパスワードを設定する傾向がある．辞書攻撃はネット経由で行われるオンライン攻撃と，ローカルで高速にクラッキングを行うオフライン攻撃がある．オンライン攻撃は，Web サイトの認証画面などから，解析する手法である．オンライン攻撃に対しては，認証画面で入力を数回間違えた者を一定時間認証不可能にすることや多要素認証を適応することで対策がなされる．オフライン攻撃はサーバから漏洩したパスワードファイルをコンピュータで解析する手法である．オフライン攻撃による辞書攻撃はソルトやストレッチングに呼ぶ対策が取られることが多い[4]．しかし，それでも辞書攻撃による不正ア

1 名城大学
Meijo University
2 愛知工業大
Aichi Institute of Technology

クセスは不可能になったわけではない．コンピュータの計算速度の進化は非常に速く，ソルトやストレッチング対策を取ったとしても時間と費用をかければ，オフライン攻撃によるパスワード解析が可能である．

そこで，これらの課題を解決するため，ユーザデバイスでのみ保管している乱数とユーザが入力したパスワードのハッシュ値をとり，当該ハッシュ値をパスワードとして扱うユーザ認証方式を提案する．ハッシュ値の算出はユーザデバイス内で自動的に行うので，ユーザはパスワードのみの認証と同じ使い勝手で利用することができる．提案方式は専用機器の用意をする必要がないため，既存のパソコンやスマートフォンで利用することができる．サーバは乱数を知らないので，サーバサイドから元のパスワードが漏れることがない．この方式は，計算量的に無限大の辞書攻撃耐性を持っている．

提案方式として，既存の認証方式を比較し使い勝手とセキュリティの両面で優れていることを示した．また，辞書攻撃耐性を机上で試算し，十分強度が高いことを示した．

以降，第 2 章では既存の多要素認証と辞書攻撃について述べる．第 3 章では，提案方式について述べ，第 4 章では，評価として既存の認証技術との比較と辞書攻撃耐性について述べる．第 5 章では，今後の検討について述べ，第 6 章でまとめる．

2. 既存技術

2.1 多要素認証における認証要素

本節では，多要素認証としてパスワードと組み合わせて使われることが多い認証要素についての概要を述べる．

2.1.1 生体認証

生体認証は個人が持っている身体情報を鍵として認証を行う方法である．生体認証は指紋認証，虹彩認証，静脈認証，顔認証など様々なものがある．ユーザが使用する際の煩わしさは少なく，使用方法が分かりやすい．指紋認証であれば指紋センサー，顔認証であればカメラが必要になるなど，機器の導入が必要になる．身体の変化で認証できないことや，精度によっては別の人を誤認証する問題がある．

2.1.2 IC カード認証

IC カードによる認証は，カードリーダーを使いカード内の秘密鍵を読み取ることで認証を行う．カード内の情報はハードウェアレベルとソフトウェアレベルの両方から守られており，外部から秘密情報への参照を防いでいる．カードの事前発行とカードリーダーの用意が必要であるため，利用者に費用が発生する．

2.1.3 OTP 認証

OTP は一定時間のみ有効なパスワードを鍵とする認証技術である．本稿では Google Authentication などの二段階認証アプリで OTP を生成する方式を想定する．OTP を行うために，ユーザはあらかじめ OTP 生成に使う鍵を共有する．共有した鍵とデバイス内の時刻カウンターを使い 6 桁数字の OTP を生成する．OTP は一定時間ごとに更新されるようになっており，有効時間が決まっている．専用機器の用意が必要ないため，ユーザに金銭的な負担がない．しかし，一定時間内にいくつか操作が必要になり，ユーザに煩わしさを与えることがある．

2.1.4 SMS 認証

SMS(Short Message Service)認証とは，相手の電話番号を指定してメッセージをやり取りするサービスである SMS を利用した認証方式である．SMS で認証の鍵となる動的なパスワードを受け取る認証であり，OTP 認証の一つと分類されることもある．ユーザは SMS で送られてきた 4 桁または 6 桁の数字であるパスワードを，認証時に入力する．使用方法は分かりやすいが，携帯電話やスマートフォンが必要になる．SMS 認証は電話回線を使っているので，認証のたびに電話料金が発生する。また，ユーザの機種によっては，SMS で送られてきたメッセージがロック画面をしていても表示されるため，他人に覗き見される危険性がある．

2.2 辞書攻撃と既存の対策

本節では，辞書攻撃の概要について述べ，その対策について説明する．

2.2.1 辞書攻撃とは

辞書攻撃とは，辞書に載っている単語やキャラクターや人物の名前などをひたすら照合することでパスワードを解析する攻撃である．辞書の単語だけではなく数字を加えた文字列や大文字と小文字を混ぜた文字列まで照合する．

辞書攻撃には，ネットワーク経由で解析するオンライン攻撃と，パスワードが保存されているデータファイルなどを入手し，ローカルで解析攻撃を行うオフライン攻撃がある．

オンライン攻撃は，毎回通信シーケンスを通すことにより解析する．通信で発生する処理時間などがあるため，パスワード解析にはそれなりの時間を要する．

オフライン攻撃による辞書攻撃は，オンライン攻撃による辞書攻撃に比べ高速に解析攻撃される危険性がある．パスワードファイルは生で保存されることはなく，必ずハッシュ演算したものが保存されている．しかし，オフライン攻撃による辞書攻撃は通信に関わる処理がないので高速解析が実現できる．また，攻撃者は辞書攻撃を仕掛ける際に，パスワードとそれをハッシュ化した結果をあらかじめ表にしておく．この表を使った辞書攻撃をレインボークラックと呼び，攻撃者は解析時にハッシュ演算を行わずに，非常に高速な解析攻撃が可能である．

2.2.2 辞書攻撃対策

オンライン攻撃による辞書攻撃は，多要素認証を取り入れることや，複数回認証に失敗すると一定時間ログインできなくすることにより対策が取られる．

オフライン攻撃による辞書攻撃対策には，ソルトおよびストレッチングがある．

ソルトとは，サーバサイドでハッシュ値を計算する前にパスワードの前後に付与する文字列のことをいう．ソルトは連結演算子＋を用いて以下のように表すことができる．

ソルト化ハッシュ = ハッシュ(パスワード + ソルト)

ソルトはユーザごとに異なる文字列を使用する．これにより，同じパスワードで保存するユーザがいても，パスワードファイルに保存されるハッシュ値は異なるものになる．ソルトはレインボークラック対策として有用で，20 文字程度の十分長いソルトを設定することが推奨されている．しかし，通常パスワードファイルが漏えいした場合は，ソルトも漏えいすると考えられるため，攻撃者はソルト化ハッシュを毎回計算することにより，時間はかかるが解析が可能である．

ストレッチングとはパスワードを保護するためにハッシュ値計算を繰り返す処理である．ハッシュ値の計算を 1,000 回，1,0000 回と繰り返すことで解析にかかる計算量を増やす処理である．これにより攻撃者が辞書攻撃に要する時間を増やすことができる．しかし，ストレッチングの回数が多いと通常のパスワード処理が増えるため，サーバの負担が増えるうえ，DoS 攻撃に悪用される可能性がある．

3. 提案方式

3.1 提案方式の考え方

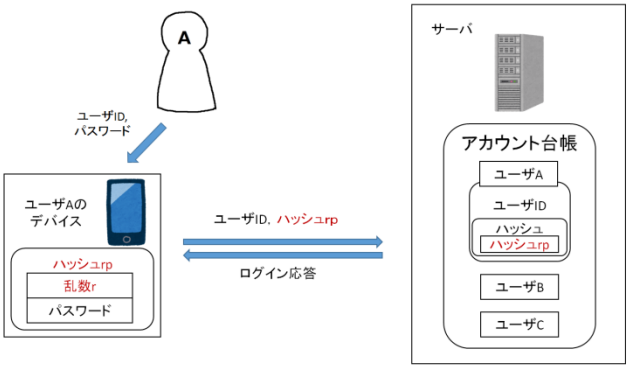

図 1　提案方式の概要

図 1 に提案方式の概要を示す．本提案では，ユーザのデバイスで十分に長い乱数 r を生成し，デバイスの不揮発メモリ内に保存しておく．また，乱数 r とパスワードを組み合わせたハッシュ rp をあらかじめサーバにパスワードとして登録しておく．ログイン時に，乱数 r とユーザが入力したパスワードのハッシュ演算を行い，サーバにはハッシュ rp をパスワードとして送信する．サーバサイドはユーザが保持している乱数 r は分からない点が特徴である．乱数とハッシュ値の生成はユーザデバイスが内部で自動的に処理する．

3.2 アカウント作成手順

ユーザのアカウント作成時に，乱数 r を生成しデバイスの不揮発メモリ内にのみ保存する．アカウント作成は様々な方法があるが，例えば，以下のように既存のメールアドレスを使った方法でアカウント作成を行う．

サーバを TLS で認証した後，ユーザはメールアドレス，ユーザ ID，パスワードを入力する．その際，ユーザデバイスでは十分に長い乱数 r を内部で生成する．ユーザデバイスは乱数 r と入力されたパスワードから，ハッシュ rp を算出する．その後，サーバ登録アカウント情報としてメールアドレス，ユーザ ID，ハッシュ rp を送信する．サーバは登録先メールアドレスに確認を取り，正しい登録であることを確認できると，アカウントを台帳に記録する．サーバはハッシュ rp をパスワードとみなすため，さらにハッシュ値を取ってパスワードファイルに保管する．ユーザデバイスは，乱数 r をユーザが入力したパスワードで暗号化しておく．

3.3 ログイン処理

図 2 に提案方式のログインシーケンスを示す．ユーザは TLS によりサーバを認証した後，ユーザ ID とパスワードを入力する．ユーザデバイスは保存されている乱数 r と入力されたパスワードからハッシュ rp を生成し，ユーザ ID と共にサーバに送信する．サーバは受け取ったログイン情報をデータベースと照合し，正しい情報であると確認されれば，正規ユーザとして認証を完了する。

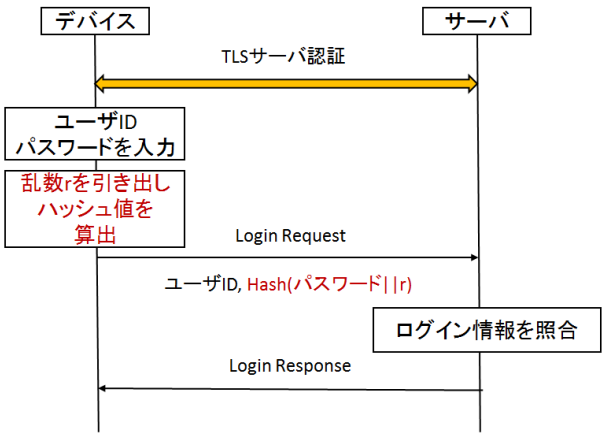

図 2　提案方式のログインシーケンス

表 1　既存の認証方式との比較

	セキュリティ			使い勝手	
	辞書攻撃	類推攻撃	リスト型攻撃	項目 1	項目 2
パスワードのみ	×	×	×	○	○
パスワード+生体認証	△	○	×	×	○
パスワード+IC カード	△	○	○	×	○
パスワード+OTP	△	○	○	○	△
パスワード+SMS	△	○	○	×	○
提案方式	○	○	○	○	○

3.4 提案方式の利点

この認証方式では，ハッシュ rp の生成はデバイス内部で自動処理する．ユーザは，ユーザ ID とパスワード以外の認証要素を入力することなくログイン可能である．また，乱数 r を保存したデバイスを利用しないとログインできないため，実質的な多要素認証であり，セキュリティを向上させている．ユーザの手間が少ないため，他の認証要素や技術を加えた多要素認証にして活用することもできる．サーバサイドに登録するパスワードは，サーバの知らない乱数が含まれるハッシュ値であるため，辞書攻撃をすることは不可能である．また，乱数 r を十分長いものにすることにより，総当たり攻撃をすることも計算量的に実質不可能にできる．サーバ側の処理はパスワードのみを利用したシステムと同じでよい．なお，乱数を保持していないデバイスからログインする方法は，今後の検討として第 6 章に後述する．

4. 評価

4.1 既存の多要素認証との比較

提案方式をセキュリティと使い勝手で既存の認証方式と比較した．比較対象はパスワードのみの場合，多要素認証として，パスワードに IC カード認証，OTP 認証，SMS 認証を組み合わせた場合とした．表 1 に既存の認証方法との比較を示す．評価項目と比較結果は以下のとおりである．

4.1.1 評価項目

セキュリティで評価する項目は辞書攻撃，類推攻撃，リスト型攻撃とした．

辞書攻撃は辞書に載っている単語をひたすら照合してパスワードを解析する攻撃である．辞書攻撃はオンライン攻撃およびオフライン攻撃によるものがあり，ここでは双方の攻撃に対する評価をしている．

類推攻撃は，不正アクセスを目的としてターゲットが設定していると考えられるパスワードを推測してクラッキングする手法である．パスワードには名前や誕生日，出身地などが設定されることが多いため、すぐに不正アクセスされてしまう場合もある．

リスト型攻撃とは他サービスなどから流出したアカウント情報を利用して，不正アクセスを試みる攻撃である．ユーザが流出先のログイン情報と同じものを使用している場合は簡単に不正アクセスされてしまう場合がある．

使い勝手の評価は，導入費および認証利用時にかかる費用の安さ(項目 1 とする)と，認証の分かりやすさや煩わしさの少なさ(項目 2 とする)で評価した．

4.1.2 認証技術の比較結果

以下の比較では辞書攻撃に△の評価をしているものがある．これは，オフライン攻撃による辞書攻撃でパスワードが解析される可能性があるが，多要素認証であるため不正アクセスまではされないということで△と評価している．

パスワードは，ユーザが覚えやすい単語やキャラクター名などを使われる傾向があるため，パスワードのみの認証では辞書攻撃や類推攻撃にあう危険性がある．また，パスワードを使いまわしているのと，リスト型攻撃にも弱いためセキュリティ全般を×と評価した．一方，ユーザに発生する費用と煩わしさは最も少ないので○と評価した．

パスワードと生体認証の組み合わせは，生体情報を辞書攻撃や類推攻撃することはできないため，辞書攻撃を△，類推攻撃を○と評価した．漏洩した身体情報を使ったリスト型攻撃がされる可能性があるため×と評価した．認証時に指紋センサーやカメラなどのデバイスが必要になるため，費用を×と評価した．ユーザの操作の手間や煩わしさは小さいので○と評価した．

パスワードと IC カード認証の組み合わせは辞書攻撃，類推攻撃，リスト型攻撃に対する耐性をもっている．パスワードと組み合わせた場合でも類推攻撃とリスト型攻撃に強いので○と評価した．IC カード認証には，カードの事前準備とカードリーダーの用意が必要であり費用が発生するので，使い勝手の項目 1 を×と評価した．使用方法がわかりやすくユーザに与える操作の負担は小さいので○と評価した．

パスワードと OTP 認証の組み合わせは，一定時間有効な動的パスワードを使用しているため，類推攻撃，リスト型攻撃に耐性を持つ．OTP 認証を取り入れるのに，専用デバイスは必要ない．発生する費用は小さいので項目 1 は○とした．しかし，使用方法と設定がやや複雑であり，一定時間の間に，ユーザはいくつか操作を要求され，少し煩わし

さを与えやすいため，項目 2 を△と評価した．

SMS 認証では OTP 認証と同様に，一定時間有効な一度きりの動的パスワードを使用する．類推攻撃とリスト型攻撃を〇に評価した．認証を行うたびに電話回線を利用し，電話料金が発生するため項目 1 の評価を×とした．使用方法はサーバから SMS で送られてきた数字を入力だけなので，項目 2 は〇と評価した．

4.2 オフライン攻撃における辞書攻撃

本節では辞書攻撃に着目し，提案方式が辞書攻撃に強いことを示す．パスワードファイルが漏洩したものとし，既存の辞書攻撃対策では，近年の技術により短時間で解析されることを示す．

一般的によく知られる SHA-256 や MD5 といった一方向性ハッシュ関数は，非力なパソコンなどでも高速に計算することが可能である．GPU や専用ハードウェアである ASIC などを用いることでさらに高速にハッシュ関数の計算が可能である[7]．今日では，サーバがパスワードをハッシュ関数に入れる際には，ハードウェアによる高速計算対策を行ったハッシュ関数を使用することが推奨されている．推奨されるハッシュ関数は時代とともに変化しており，2019 年現在では BCrypt が推奨されている[8]．BCrypt はハッシュ関数の機能として，ソルトとストレッチング機能が内蔵されており，デフォルトではストレッチングを約 1,000 回行う．表 2 にオフライン攻撃の辞書攻撃所要時間を示す．

表 2　オフライン攻撃の辞書攻撃所要時間

	辞書攻撃にかかる時間
SHA-256	8.7 ミリ秒
BCrypt	30 分
提案方式	不可

(1)　SHA-256 に対する辞書攻撃

BCrypt が一般になる前は SHA-256 や SCrypt などのハッシュ関数が使われていた．そこで，SHA-256 を GTX 1080 Ti のハッシュレートを用いて，オフライン攻撃における 1 人を狙った辞書攻撃が，どれくらいの時間で完了するのかを試算した．

GTX 1080 Ti のハッシュレートは 4,600 MH/s とした(1 秒間に 46 億回 SHA-256 を計算することができる)[9]．辞書攻撃の単語数は，Openwall のワードリスト数である 4,000 万単語と仮定し[10]，ソルトとストレッチング 1,000 回の処理を行うものとした．解析時のハッシュ計算以外にかかる時間は考慮していない．以上より，1 人のパスワードに辞書攻撃を完了する時間は，

$$4.0 \times 10^7 \div (4.6 \times 10^9) \fallingdotseq 8.7 \times 10^{-3}$$

となり，8.7 ミリ秒で 1 人のパスワードを辞書攻撃が完了する．SHA-256 は高速解析に対する耐性を持っていないため，ソルトとストレッチング処理を行っていても一瞬でパスワード解析されることがわかる．

(2)　BCrypt に対する辞書攻撃

GTX 1080 Ti の BCrypt に対するハッシュレートを利用して，1 つのパスワードを解析するのにかかる時間を試算した．辞書攻撃の単語数は，(1)と同様に 4,000 万単語で仮定した．ハッシュレートは 22,000 H/s で試算した[9]．1 人のパスワードを解析するのに要する時間は，

$$4.0 \times 10^7 \div (2.2 \times 10^4) \div (6.0 \times 10) \fallingdotseq 3.0 \times 10$$

BCrypt は高速解析を行うハードウェアに対する攻撃耐性を持っているが，それでも狙った一人を辞書攻撃する際は 30 分ほどで完了する．実際は，ハッシュ計算以外の処理も行うのでさらに時間がかかるが，レインボークラックを行わなくても，オフライン攻撃で高速に辞書攻撃ができることが分かる．

(3)　提案方式

提案方式では，サーバに登録されるパスワードは，ユーザデバイスで生成したハッシュ rp である．ハッシュ rp は辞書にないため，辞書攻撃をすること自体が不可能である．また，ハッシュ rp を十分大きくすれば(例えば 128bit)，総当たり攻撃も不可能である．

5. 今後の検討

提案方式では，乱数 r を保持していないデバイスからはログインすることができない．この特性は，多要素認証であることからオンライン攻撃耐性を高めている．しかし，ユーザによっては，別のデバイスから同じアカウントにログインしたい者がいると考えられる．そこで，別デバイスからログインできる方式を検討している．

1 つのアカウントに対して複数のデバイスを登録できるように拡張する．なお，この方法を利用する場合は，サーバの拡張が必要になる．図 3 に 1 アカウントに複数ハッシュ値を登録した認証を示す．

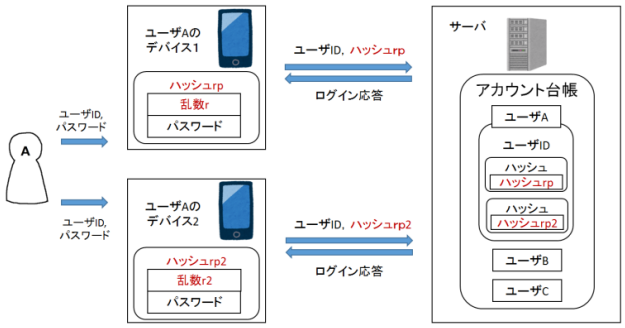

図 3　1 アカウントに複数ハッシュ値を登録した認証

ユーザは，新規デバイス用に独自の乱数 r2 を生成する．この方式では，ユーザは同じパスワードを利用して複数のデバイスを利用できる．

6. まとめ

乱数とパスワードを組み合わせたユーザ認証方式を提案した．乱数とパスワードでハッシュ値をとり，算出したハッシュ値をパスワードとしてサーバに登録する．乱数はユーザデバイス内の不揮発メモリ内に保存されており，サーバは知ることができない．提案方式は，辞書攻撃，類推攻撃，リスト型攻撃などに耐性を持ちセキュリティが高い．ユーザは二段階要素の入力をする必要がなく，既存のパスワード認証と同じ使い勝手で利用することができる．サーバはユーザが設定した元のパスワードを知るすべがなく，サーバサイドのパスワードファイルが流出した場合にも辞書攻撃，総当たり攻撃で解析されることがない．

参考文献

[1]　鈴木 宏哉，山口 利恵：研究報告コンピュータセキュリティ（CSEC），Vol.2016-CSEC-73，No.13，pp.1-8(オンライン)，入手先〈https://ipsj.ixsq.nii.ac.jp/ej/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=160458&item_no=1&page_id=13&block_id=8〉（2019-01-09）.
[2]　多要素認証の種類と方法(多要素認証第 2 回)，TrustLogin by GMO，入手先<https://sku.id/catalog/skuid_whitepaper02.pdf> (参照 2019-01-29).
[3]　二段階認証とは，bitFlyer，入手先 <https://bitflyer.com/ja-jp/glossary/two_factor_authentication> (参照 2019-05-10).
[4]　実際にあった情報漏えいの被害事例まとめ，CyberSecurityTIMES，入手先 <https://www.shadan-kun.com/blog/measure/2763/> (参照 2019-05-10).
[5]　ハッキング被害に遭った 2300 万人以上が使っていた最も危険なパスワードは「123456」，Gigazine，入手先 <https://gigazine.net/news/20190422-most-used-dangerous-password/> (参照 2019-05-12).
[6]　本当は怖いパスワードの話，ITmedia，入手先 <https://www.atmarkit.co.jp/ait/articles/1110/06/news154.html> (参照 2019-04-14).
[7]　scrypt が GPU に破られる時，びりあるの研究ノート，入手先 <https://blog.visvirial.com/articles/519> (参照 2019-04-28).
[8]　Password_hash，php[world] 2019 Call for Speakers，入手先 <https://php.net/manual/ja/function.password-hash.php> (参照 2019-05-12).
[9]　Password Cracking with 8x NVIDIA GTX 1080 Ti GPUs，STH，入手先 <https://www.servethehome.com/password-cracking-with-8x-nvidia-gtx-1080-ti-gpus/> (参照 2019-04-29).
[10]　Openwall wordlists collection，Openwall，入手先 <https://www.openwall.com/wordlists/> (参照 2019-04-28).