情報処理学会論文誌 Vol.55 No.9 2072–2080 (Sep. 2014)

計算機援用ユーザ認証

兼子 拓弥[1] 本部 栄成[1] 高橋 健太[2] 西垣 正勝[1,a)]

受付日 2013年12月2日, 採録日 2014年6月17日

概要：計算機の能力は日々向上する．総当たり攻撃に耐性を持たせるためには，秘密情報のエントロピを計算機の進歩にともなって増加させる必要がある．すなわち，計算量的安全性に依拠するセキュリティシステムにおいては，秘密情報のエントロピの確保が必須となる．しかし，ユーザ認証においては，現在の暗号化鍵に比べて，人間が覚えることができる秘密情報（パスワードなど）のエントロピが格段に小さい．このため本論文では，正規ユーザの認証においても（不正者と同様に）計算機を利用することによって，ユーザ認証を強化する方法を提案する．具体的には，認証情報を「ユーザ自身が入力する秘密情報」と「計算機の総当たり試行によって入力する補助情報」によって構成する．秘密情報を知っている正規ユーザは，補助情報の分の総当たり試行だけで認証情報の同定が可能であるのに対し，不正者は，秘密情報と補助情報の両方を総当たり試行によって求めなければならず，なりすましが困難となる．不正者の攻撃能力が向上する分，正規ユーザの認証能力も向上するため，計算機の能力がいかに向上したとしても「不正者は，秘密情報の分だけ正規ユーザよりも総当たり試行に要する時間が大きくなる」という正規ユーザ優位の状況が保たれる．

キーワード：ユーザ認証，認証情報，エントロピ強化，総当たり攻撃，記憶負荷

Computer-aided User Authentication

Takuya Kaneko[1] Eisei Honbu[1] Kenta Takahashi[2] Masakatsu Nishigaki[1,a)]

Received: December 2, 2013, Accepted: June 17, 2014

Abstract: Against the brute force attack, the entropy of secret information (passwords, etc.) needs to be big enough. However, human ability to memorize information is limited. On the other hand, adversaries can use the CPU power for the brute force attack and the CPU performance is constantly improving. To tackle this issue, this paper proposes that not only the adversaries but also the legitimate users use the CPU power for the user authentication. Specifically, the user authentication is carried out by using both "the secret information that is supplied by the user" and "the helper information that is found through the brute force search by the CPU". For the adversaries, this brute force search will take a longer time since they have to find both the secret information and the helper information. In contrast, the legitimate users can finish this brute force search in a shorter time since they know the secret information. As the CPU performance improves, the ability of the brute force attack for the adversaries increases. But, at the same time, the ability of the brute force search for the legitimate users increases, too. That is why, it is expected that the brute force attack made by the adversaries is always more difficult than the brute force search made by the legitimate users.

Keywords: user authentication, authentication information, entropy enhancement, brute force attack, memory load

1. はじめに

情報システムにおけるセキュリティ技術の多くは計算量的安全性に依拠する．このようなセキュリティシステムにおいては，不正者の計算量が多項式時間内に収まら

[1] 静岡大学大学院情報学研究科
Faculty of Informatics, Shizuoka University, Hamamatsu, Shizuoka 432–8011, Japan
[2] 株式会社日立製作所横浜研究所
Hitachi, Ltd., Systems Development Laboratory, Yokohama, Kanagawa 244–0817, Japan
[a)] nisigaki@inf.shizuoka.ac.jp

ないようにするために，秘密情報のエントロピの確保が必須となる．たとえば暗号通信などにおいては現在，共通鍵暗号の秘密情報（暗号化鍵）は 256 ビット（最近は 512 ビット），公開鍵暗号は 1,024 ビット（最近は 2,048 ビット）が推奨されている．これに対し，ユーザ認証においては，what-you-know タイプの認証では人間の記憶負荷や利便性などの理由で，who-you-are タイプの認証では認証精度などの理由で，秘密情報（パスワードや生体情報）のエントロピが（暗号化鍵と比べて）格段に小さい．

ユーザ認証の総当たり攻撃に対する耐性を保つためには，秘密情報のエントロピが不正者の持つ計算機能力よりも大きくなるように，秘密情報を設定するべきである．しかし，計算機の能力は日々向上する．これに対し，人間が記憶可能なパスワード長や人体固有の生体情報は，一朝一夕で増えるものではなく，基本的にはつねに一定であると考えられる．したがって，ユーザ認証の秘密情報（パスワードや生体情報）に関しては，秘密情報そのもののエントロピを十分に大きく確保するというコンセプトでは，総当たり攻撃に耐性を持つユーザ認証を実現することは不可能である．

このため本論文では，正規ユーザの認証においても（不正者と同様に）計算機を利用することによって，ユーザ認証を強化する方法を提案する．具体的には，認証情報を「ユーザ自身が入力する秘密情報」と「計算機の総当たり試行によって入力する補助情報」によって構成する．秘密情報を知っている正規ユーザは，補助情報の分の総当たり試行だけで認証情報の同定が可能であるので，短い時間でユーザ認証が終了する．不正者は，秘密情報と補助情報の両方を総当たり試行によって求めなければならず，秘密情報の分だけ正規ユーザよりも総当たり試行に要する時間が大きくなるため，なりすましが困難となる．

総当たり攻撃を行う不正者にとって，計算機能力の向上は攻撃能力の向上に一致する．一方で，補助情報の総当たり試行を行う正規ユーザにとっても，計算機能力の向上は認証能力（補助情報を総当たり試行によって同定する能力）の向上に一致する．このように，不正者の攻撃能力が向上する分，正規ユーザの認証能力も向上するため，将来的に計算機の能力がいかに向上したとしても「不正者は，秘密情報の分だけ正規ユーザよりも総当たり試行に要する時間が大きくなる」という正規ユーザ優位の状況が保たれる．本論文では，計算機の能力をユーザ認証に利用する提案方式を「計算機援用ユーザ認証」と呼ぶ．

2. ユーザ認証におけるエントロピの問題

2.1 パスワード認証

現在最も多く使われているユーザ認証方式は，パスワード認証方式である．パスワード認証では，個人を特定するための ID とユーザがあらかじめ設定したパスワードの入力をユーザに要求し，ID に対するパスワードが正しければ正規ユーザであると認める．

パスワード認証に対する典型的な攻撃手法に，総当たり攻撃と辞書攻撃が存在する．これらの攻撃に対して十分な安全性を確保するためには，パスワードは長く，かつランダムな文字列とすることが望ましい．しかし，長くランダムな文字列を記憶することはユーザにとって大きな負担となる．そのため，多くのユーザは短いパスワードや記憶しやすいパスワードを使用してしまい，結果としてユーザ認証としての十分な安全性を確保することができていないのが現状である [1]．

2.2 画像認証

人間の画像認識能力の高さを利用し，パスワードの代わりにパス画像を秘密情報として用いることによってユーザの記憶負荷を軽減させる画像認証方式が提案されている．画像認証には，複数の囮画像の中に紛れたパス画像を選択する Cognometric 方式 [2], [3] と，1 枚の画像の中の特定箇所（パスポイント）を選択する Locimetric 方式 [4], [5] に大別される．

しかし，たとえば，Cognometric 方式では，1 画面に表示できる画像数には限度があるため，総当たり数の確保が困難となっている．仮に多数の小さなアイコンを無理矢理一画面に敷き詰めることができたとしても，大量の囮画像の中に紛れるパス画像を発見することは，ユーザにとって容易ではなくなってしまう．また，Locimetric 方式では，たとえば，$1{,}000 \times 800$ 画素の画像に対して正解領域（パスポイント）が 10×10 画素であった場合には，8,000 通りの総当たり数しか確保できないという計算になる．また，人間は画像中の特徴的な点（ホットスポット）をパスポイントとして選択する傾向があることが知られており，パスポイントの実際のエントロピはさらに小さくなると推測されている [6]．

以上のように，秘密情報の記憶に対するユーザの負荷を軽減するために導入された「画像の利用」が，皮肉にも，秘密情報のエントロピを低減させる結果を引き起こしてしまっている．

2.3 CAPTCHA

現在最も広く使用されている文字判読型 CAPTCHA においては，文字種別と文字数を適切に増やすことによって，総当たり攻撃に対して実用的な強度を有するエントロピを確保することは可能である．ただし，解答すべき文字種別や文字数が増えるにつれて，正規ユーザにとって利便性は低下する．

また，最近の光学文字認識（OCR）技術の性能向上により，マルウェアも文字判読型 CAPTCHA の判読が可能になってきている [7]．この問題に対処するため，人間の「よ

り高度な認知処理」に基づく CAPTCHA が種々提案されている [8], [9] が，それらの多くは人間の画像認識能力の高さを利用するものであり，2.2 節で説明した画像認証の場合と同じ理由で，総当たり攻撃に対抗するに足るエントロピの確保に対する課題を残している．

2.4 生体認証

一般的に，生体情報は同一人物であっても入力のたびに誤差が含まれるため，本人拒否率を抑えようとすると，ある程度の他人受入を許容する必要がある．生体情報のエントロピについては，これを正確に評価する方法は現在のところ知られていないが，実用的には「他人受入率の逆数」が生体情報のエントロピ（正確には，当該生体認証装置に対する生体情報のエントロピ）と考えられる [10]．仮に生体認証システムが 99.9999%の精度（他人受入率 0.0001%）を有していたとしても，生体情報のエントロピはたかだか 100 万通りである．したがって，生体認証においても，総当たり攻撃に対する脆弱性は大きなリスクである．

3. 総当たり攻撃に対する既存研究

総当たり攻撃に対してユーザ認証の耐性を保つためには，秘密情報のエントロピが不正者の持つ計算機能力よりも大きくなるように，秘密情報を設定する必要がある．この実現のためには，秘密情報のエントロピを増大させるアプローチと不正者の攻撃効率を低下させるアプローチが存在する．ここでは，これらに関する既存手法をいくつか紹介する．

3.1 秘密情報のエントロピを増大させるアプローチ

3.1.1 語呂合わせ

パスワードは，十分なエントロピを持ち，かつランダムな文字列であることが望ましい．しかしながら，ランダムな文字列をユーザが記憶することは大きな負担となる．そこで，一見ランダムに見えるパスワードに対して語呂合わせなどによって意味づけを行い，長いパスワードを比較的小さな記憶負荷によって覚えるという方法がある [11]．

3.1.2 パスワード管理ツール

ユーザのパスワード管理を支援する方式として，パスワード管理ツール [12] があげられる．パスワード管理ツールは，ユーザの代わりに複数の（ID と）パスワードをユーザ PC 内で管理する．認証情報の記憶が不要となるため，ユーザは，十分に長いランダムなパスワードをサービスごとに個別に設定することが可能となる．

パスワード管理ツールに登録されている認証情報は暗号化によって安全にユーザ PC 内に保管される．ユーザがパスワード管理ツール自体にログインすることで，ツールに登録されているすべての認証情報の使用が許可される．すなわち，ユーザはパスワード管理ツールにログインするための ID とパスワードを 1 組覚えるだけで，サービスごとに異なる認証情報を利用することが可能となる．

この方式では，パスワード管理ツール自体へのログインは，通常のパスワード認証を用いることとなる．すなわち，管理ツールに対するログイン情報に対してはエントロピの課題は解決されず，この部分が脆弱ポイントとして残る．管理ツールにログインするためのパスワードが漏洩してしまうと，ツールに登録されているすべての認証情報が不正者の手にわたるため，被害が深刻となる．

3.1.3 認証を繰り返し行う方法

1 度の認証では十分なエントロピを確保できない場合に，複数回認証を行うことでエントロピの確保を達成することが可能である．たとえば，画像認証においてパス画像を選択するという行為を複数回行う方法や，指紋認証において 2 本以上の指紋を提示する方法がこれにあたる．繰返しの回数が増すごとに，ユーザの利便性が損なわれることになる．

3.1.4 what-you-have タイプの認証

what-you-have タイプの認証であれば，ユーザが所持する IC カードなどのトークンの中に，十分長く，ランダムな認証情報を記録することが可能である．ただし，トークン自体の盗難や紛失に対するリスクがあるため，ユーザにトークンの管理徹底が求められる．このようなリスクに対して，パスワード認証または生体認証によってトークンをアクティベートする対策がとられていることが多いが，その場合は，トークンをアクティベートするための認証情報におけるエントロピの課題が浮上する．

3.2 不正者の攻撃効率を低下させるアプローチ

3.2.1 タールピット

タールピットとは，認証に失敗した場合，一定時間が経過しないとリトライできない仕組み[*1]のことである [13]．総当たり攻撃は，正しい秘密情報を発見するまで認証を繰り返すことによって行われるため，タールピットを仕掛けて一定時間のリトライを禁止することによって，総当たり攻撃に要する時間を膨大にすることができる．また，規定回数以上の認証失敗の場合には，アカウントを無効化（アカウントのロックアウト）するという方法 [13] もとれる．ただし，タールピット（やアカウントロックアウト）が有効に機能するのは，オンラインで実行される総当たり攻撃に対してのみとなる．

3.2.2 bcrypt

Provos らは，ハッシュ値を求めるための計算量を意図的に多くすることで，ユーザ認証の総当たり攻撃に対する耐性を向上させる方法を提案した [14]．bctypt と名付けられ

[*1] 一般的にはアカウントのロックアウトの一種として知られている．本論文では特にアカウントの無効化との区別を行うために，「タールピット」という言葉で呼称する．

たこの手法は，Blowfish [15] 型のブロック暗号における部分鍵生成演算の繰返し回数を可変とすることで，ハッシュ計算に要する時間をコントロールできるように設計されている．

ハッシュ計算に時間を要するようになれば，認証試行 1 回（認証時に入力された情報のハッシュ値が登録されているハッシュ値と一致するか否かの検査）あたりの所要時間が増加し，その分，不正者が単位時間あたりに実行可能な認証試行回数が減少する．すなわち bcrypt は，認証アルゴリズムそのものにタールピットを仕掛け，不正者の総当たり攻撃能力を減衰させる方式であるといえる．

しかし，bcrypt のように暗号関数そのものに手を加える方法においては，安全性の証明までを考えると，その設計負荷は比較的高いものとなる．また，使用するハッシュ関数が固定されてしまうことは，認証アルゴリズムのバラエティが限定されるという弊害につながるだけでなく，万一このハッシュ関数がブレイクされてしまった場合には代替が効かないという問題をはらむ．

3.2.3 ソルト

通常，パスワード認証システムにおいては，パスワードはハッシュ化された状態で保管されており，認証時に入力されたパスワードのハッシュ値が登録されているハッシュ値と一致するか否かによって認証の可否が判定される．ハッシュ関数の一方向性によって，パスワードのハッシュ値が万一漏洩したとしても，ハッシュ値からパスワードを逆算することは難しい．

しかし，レインボーテーブル（平文のすべての組合せに対するハッシュ値を事前に計算して，平文とハッシュ値の対応を表にしたもの）が用意されていた場合には，不正者はテーブルルックアップによって実時間内にハッシュ値からパスワードを知ることができる [16]．現在，70〜80 ビット程度以上のパスワードでなければ，レインボーアタックに脆弱であるといわれている [17]．しかし，レインボーテーブルの作成は不正者によってアンダーグラウンドでつねに行われており，レインボーアタックに耐性を持たせるために必要となるパスワードのエントロピは絶えず増加していく．

レインボーアタックに対する対策としてソルトが知られている [18]．ソルトとは，パスワードの見かけ上のエントロピを増加させるためにパスワードに付加する乱数のことである．ソルトの付加によって，レインボーテーブルの作成に天文学的な時間を要するようになる．

ソルトの値そのものは，パスワードのハッシュ値とともに平文で保管されることになる．このため，ソルトはパスワードそのもののエントロピを増加させるものではなく，正規ユーザの記憶負荷も増加しない．しかし，その一方で，1 つのハッシュ値とソルトの組を入手した不正者による総当たり攻撃の計算量は，パスワードのエントロピのみにより決定され，ソルトの有無にかかわらず一定である．したがって，そのような総当たり攻撃（ハッシュ値とソルトの組を入手した不正者による総当たり攻撃）に対して十分な安全性を達成しようとするとパスワード自体のエントロピを増加させるほかない．

4. 計算機援用ユーザ認証

計算機の能力は日々向上するため，秘密情報のエントロピを増大させるアプローチには限界がある．また，不正者の攻撃効率を低下させるアプローチに関しては，

- タールピットは，オフラインによる総当たり攻撃には対抗できない．
- bcrypt は，暗号設計の負荷の高さが課題として残る．
- ソルトは，パスワードのエントロピそのものを増加させるものではなく，ハッシュ値とソルトの組に対する総当たり攻撃に対して安全性を保つためには十分長いパスワードを設定する必要がある．

という点が短所として残る．

そこで本論文では，正規ユーザであっても認証の際に（不正者と同様に）計算機を利用し，計算機の能力とユーザ自身が所有する秘密情報を併用して認証を行う「計算機援用ユーザ認証」を提案する．これにより，正規ユーザは「計算機の能力のうえに，さらに秘密情報の分のアドバンテージを加えた情報」を認証に利用できるようになり，計算機の能力のみを利用することしかできない不正者に対してつねに優位を保つことができるようになる[*2]．

4.1 コンセプト

総当たり試行には，全パターンを試行するために必要となる時間は認証情報のビット長に応じて指数関数的に増加するという特徴がある．この特徴を用いて，ユーザに認証情報の一部のみを入力させ，残りの情報を総当たり試行によって補完することで，ユーザ認証におけるエントロピの不足を緩和するとともに，実用時間内での認証完了を達成する．以降，認証情報のうち，ユーザが入力する情報を「秘密情報」，総当たり試行によって補完する情報を「補助情報」と呼ぶ．すなわち認証情報とは，秘密情報と補助情報を連結した情報となる．

不正者が認証情報全体を総当たり攻撃するのに要する時間が $\boldsymbol{T_2}$ 以上であれば安全であり，かつ，正規ユーザが認証に要する時間が $\boldsymbol{T_1}$ 以下ならば利便性が損なわれないとする（$\boldsymbol{T_1} < \boldsymbol{T_2}$）．ここでは，説明を具体的にするために，$\boldsymbol{T_1}$ を 1 秒，$\boldsymbol{T_2}$ を 1 年（$\fallingdotseq 3 \times 10^7$ 秒）と考えることにす

[*2] 提案方式は，オフラインでの総当たり攻撃に対する対策となっている点で，既存方式と一線を画す．オンラインでの総当たり攻撃に対しては，本方式は「認証失敗のたびに時間 $\boldsymbol{T_1}$ だけ（4.1 節の例では $\boldsymbol{T_1}$ は 1 秒間）待つ」タイプのタールピットとほぼ同等（ただし，提案方式は，正規ユーザによる 1 回目の認証試行の際にも $\boldsymbol{T_1}$ の待ち時間を要求する）の対策効果を呈す．

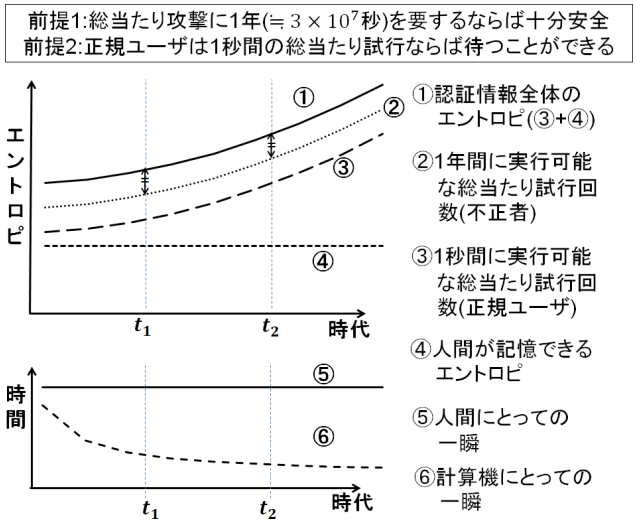


図 1 時代による処理能力（エントロピ）と処理時間の推移
Fig. 1 Transition of computer power and processing time.

る．たとえば，現在の計算機が 1 秒間に 10^8 通りの総当たりが可能だとすると，秘密情報を知らない不正者が総当たり試行に 1 年以上を要するようにするためには，認証情報全体のエントロピとして $3\times10^7\times10^8=3\times10^{15}$ 通りを確保する必要がある．一方で，秘密情報を知っている正規ユーザであれば 1 秒で総当たり試行が終了するためには，補助情報のエントロピを 10^8 通りに抑える必要がある．以上より，秘密情報のエントロピを 3×10^7 通り，補助情報のエントロピを 10^8 通りとして，認証情報全体のエントロピを 3×10^{15} 通りにしてやれば，「不正者による認証情報全体の総当たり攻撃に要する時間は 1 年，かつ，正規ユーザによる認証に要する時間は 1 秒」の制約を満たすことが分かる．

ここで，提案方式においては，補助情報のエントロピを調整することによって，認証情報全体のエントロピが任意に設定できることに注意されたい．これにより，将来，計算機速度が向上し，総当たり試行に要する時間が短縮されたとしても，正規ユーザが入力する秘密情報を増やすことなく，総当たり攻撃に対する耐性を維持することができる．たとえば，上記の例において計算機速度が 10 倍（1 秒間に 10^9 通りの総当たりが可能）になった場合には，補助情報のエントロピを 10 倍（10^9 通り）にして，認証情報のエントロピを 10 倍（3×10^{16} 通り）にする（正規ユーザが入力する秘密情報のエントロピは 3×10^7 通りのままである）．これにより，「不正者による認証情報全体の総当たり攻撃に要する時間は 1 年，かつ，正規ユーザによる認証に要する時間は 1 秒」の制約はそのまま維持される．

この関係を図 **1** に模式的に示した．時代とともに計算機の性能は向上し，不正者が攻撃のために割ける時間（ここでは $T_2=1$ 年と仮定している）の範囲内で実行可能な総当たり攻撃回数もそれに応じて増加する（図 1 の ②）．すなわち，計算機の時間感覚としては，性能の向上とともに「一瞬」という時間が日を追って短くなっていく（図 1 の ⑥）．これに対し，人間の時間感覚は時が移っても大きく変化することはなく，たとえば「1 秒」を感じる時間の長さは，現在（t_1）も次世代（t_2）もほぼ同じ時間を保つ（図 1 の ⑤）．以上より，正規ユーザが一瞬と感じる時間（ここでは $T_1=1$ 秒と仮定している）の範囲内で実行可能な総当たり試行回数は，計算機の性能向上と歩調を合わせて増加することが分かる（図 1 の ③）．よって，「補助情報に対する 1 秒間の総当たり試行」という計算機の援用をユーザ認証に組み込むことにより，認証情報全体のエントロピ（図 1 の ①）は正規ユーザが記憶できる分のエントロピ（図 1 の ④）と 1 秒間に実行可能な総当たりの分のエントロピ（図 1 の ③）を合わせたものとなる．

認証情報のエントロピ（図 1 の ①）が，不正者が実行可能な総当たり試行回数（図 1 の ②）を超えていれば，ユーザ認証の安全性が確保されることになる．人間の記憶力や生体認証装置の精度は時を越えてさほど変化することはない（図 1 の ④）が，提案方式によって「認証情報のエントロピ（図 1 の ①）が不正者の攻撃能力（図 1 の ②）をつねに一定量超える」という状況を実現することが可能である．

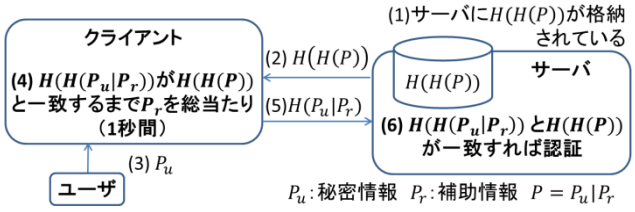


図 2 基本方式
Fig. 2 Basic procedure.

4.2 基本的な認証手順（基本方式）

提案方式による具体的な認証の手順を以下に記す（図 **2**）．なお，提案方式における秘密情報は，パスワード認証においてはパスワード，CAPTCHA においては提示された問題に対してユーザが入力する解答，生体認証においては生体情報である．

(1) サーバに，認証情報 $\boldsymbol{P}$ $(=\boldsymbol{P_u}|\boldsymbol{P_r})$ の二重ハッシュ値 $\mathbf{H}(\mathbf{H}(\boldsymbol{P}))$ が格納されている．ここで，$\boldsymbol{P_u}$ はユーザが入力する秘密情報，$\boldsymbol{P_r}$ は総当たり試行によって同定する補助情報を表す．
(2) サーバは，$\mathbf{H}(\mathbf{H}(\boldsymbol{P}))$ をクライアントに送信する．
(3) ユーザは，$\boldsymbol{P_u}$ をクライアント端末に入力する．
(4) クライアント端末は，$\mathbf{H}(\mathbf{H}(\boldsymbol{P_u}\,|\,\boldsymbol{P_r}))$ が $\mathbf{H}(\mathbf{H}(\boldsymbol{P}))$ と一致する $\boldsymbol{P_r}$ を総当たり試行によって求める．
(5) クライアント端末は，$\mathbf{H}(\boldsymbol{P_u}|\boldsymbol{P_r})$ をサーバに送信する．
(6) サーバは，受け取った $\mathbf{H}(\boldsymbol{P_u}|\boldsymbol{P_r})$ のハッシュ値 $\mathbf{H}(\mathbf{H}(\boldsymbol{P_u}\,|\,\boldsymbol{P_r}))$ が，$\mathbf{H}(\mathbf{H}(\boldsymbol{P}))$ と同一であれば認証成功とし，一致しなければ認証失敗とする．

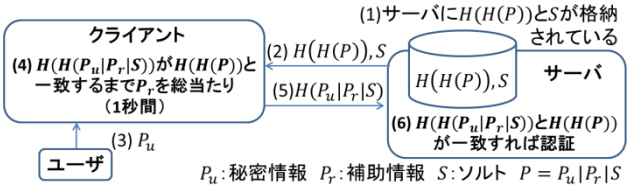


図 3 拡張方式
Fig. 3 Extended procedure.

4.3 ソルトを加えた認証手順（拡張方式）

レインボーアタック [16] への攻撃耐性を考慮し，4.2 節の認証手順を，ソルトを加えた手順へと拡張する．ソルトの導入によってレインボーアタック（レインボーテーブルを用いたテーブルルックアップ）への耐性を強化するとともに，ソルトだけでは実現しえなかった「ハッシュ値とソルトの組を入手した不正者による総当たり攻撃」に対する安全性を向上させることが可能となる．ソルトを加えた拡張方式の認証手順を以下に記す（図 3）．

(1) サーバに，認証情報 $\boldsymbol{P}$ $(=\boldsymbol{P_u}|\boldsymbol{P_r}|\boldsymbol{S})$ の二重ハッシュ値 $\mathbf{H}(\mathbf{H}(\boldsymbol{P}))$ およびソルト $\boldsymbol{S}$ が格納されている．ここで，$\boldsymbol{P_u}$ はユーザが入力する秘密情報，$\boldsymbol{P_r}$ は総当たり試行によって同定する補助情報を表す．
(2) サーバは，$\mathbf{H}(\mathbf{H}(\boldsymbol{P}))$ と $\boldsymbol{S}$ をクライアントに送信する．
(3) ユーザは，$\boldsymbol{P_u}$ をクライアント端末に入力する．
(4) クライアント端末は，$\mathbf{H}(\mathbf{H}(\boldsymbol{P_u}|\boldsymbol{P_r}|\boldsymbol{S}))$ が $\mathbf{H}(\mathbf{H}(\boldsymbol{P}))$ と一致する $\boldsymbol{P_r}$ を総当たり試行によって求める．
(5) クライアント端末は，$\mathbf{H}(\boldsymbol{P_u}|\boldsymbol{P_r}|\boldsymbol{S})$ をサーバに送信する．
(6) サーバは，受け取った $\mathbf{H}(\boldsymbol{P_u}|\boldsymbol{P_r}|\boldsymbol{S})$ のハッシュ値 $\mathbf{H}(\mathbf{H}(\boldsymbol{P_u}|\boldsymbol{P_r}|\boldsymbol{S}))$ が，$\mathbf{H}(\mathbf{H}(\boldsymbol{P}))$ と同一であれば認証成功とし，一致しなければ認証失敗とする．

5. 運用面に関する考察

5.1 秘密情報の使い回し

ユーザが複数のサービスを利用する場合，パスワードの使い回しが問題となる．たとえば，ユーザ A がサーバ 1 とサーバ 2 の 2 つの WEB サービスを利用するケースを考えた場合，利便性の観点からは，1 つのパスワードでどちらのサービスも利用できたほうが便利であろう．しかし，サーバ 1 とサーバ 2 のパスワードを同一にしてしまうと，万一サーバ 1 が不正者の侵入を許してしまった場合に，不正者はサーバ 1 から盗んだユーザ A のパスワードをそのまま使ってサーバ 2 にもログインすることが可能となってしまう．すなわち，安全性の観点からは，サーバ 1 とサーバ 2 に登録される認証情報は互いに異なるものにすべきである．

提案方式であれば，同一の秘密情報（パスワード）$\boldsymbol{P_u}$ に対して，異なる 2 つの補助情報 $\boldsymbol{P_{r1}}$，$\boldsymbol{P_{r2}}$ を用意して，2 つの異なる認証情報 $\boldsymbol{P}_1$ $(=\boldsymbol{P_u}|\boldsymbol{P_{r1}})$，$\boldsymbol{P}_2$ $(=\boldsymbol{P_u}|\boldsymbol{P_{r2}})$ を生成することができる．よって，サーバ 1 の認証情報として $\boldsymbol{P}_1$ $(=\boldsymbol{P_u}|\boldsymbol{P_{r1}})$ を，サーバ 2 の認証情報として $\boldsymbol{P}_2$ $(=\boldsymbol{P_u}|\boldsymbol{P_{r2}})$ を登録すれば，ユーザが入力するパスワード $\boldsymbol{P_u}$ を同一としたままで，サーバ 1 とサーバ 2 に登録される認証情報を異なるものにすることができる．

5.2 総当たり試行を実行するエンティティ

4 章で示した基本方式，拡張方式は，総当たり試行をクライアント側で行う形式となっている．しかし，認証サーバ側で総当たり試行を行う形式や，総当たり試行を第三者機関に委託する形式も考えられる．本節では，これらの比較を行う．

5.2.1 クライアント側で総当たり試行を行う形式

クライアント側で総当たり試行を行う場合，認証サーバに大きな負荷をかけることなく認証を行うことが可能である．しかし，ユーザごとに利用する PC は異なるため，クライアントマシンの計算機能力がユーザによって異なることになる．正規ユーザの PC（クライアントマシン）の演算性能に合わせて補助情報のエントロピを設定すると，不正者が演算能力の高い計算機を利用して総当たり攻撃を仕掛けた場合に，期待される安全性を保つことができない恐れがある．

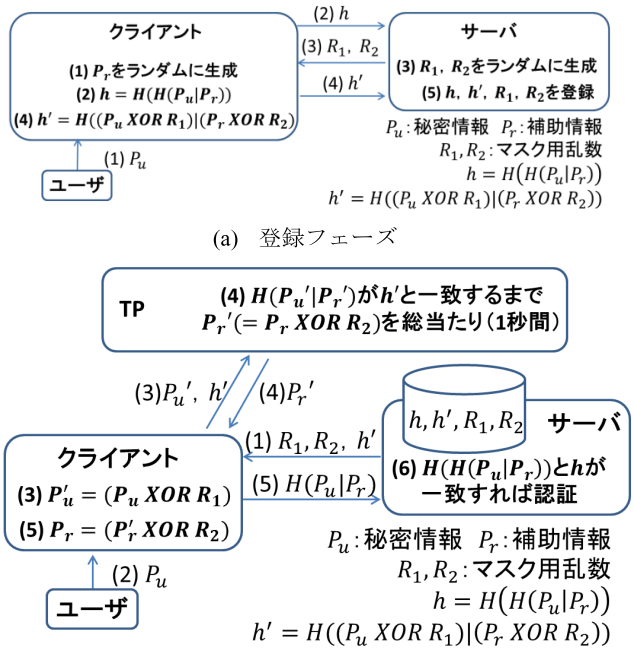
5.2.2 認証サーバ側で総当たり試行を行う形式

認証サーバ側で総当たり試行を行う場合，すべてのユーザに対して同一の演算能力が提供されるため，5.2.1 項で示した「クライアントマシンの演算能力の差を利用した攻撃」の問題は解消される．一方で，同時に多数のユーザが認証を行うことによりサーバに負荷がかかることが考えられ，これを利用した DoS 攻撃のリスクが増加する．また，認証サーバに総当たり試行を代行してもらうためには，認証サーバに秘密情報 $\boldsymbol{P_u}$ を通知しなければならず，サーバが不正者に乗っ取られてしまった場合や，サーバが悪意を持っていた場合には，秘密情報 $\boldsymbol{P_u}$ そのものが不正者に奪われてしまうことになる．

5.2.3 総当たり試行を第三者機関に委託する形式

総当たり試行を代行する第三者機関（TP）を用意することができれば，5.2.1 項で示した「クライアントマシンの演算能力の差を利用した攻撃」の問題と，5.2.2 項で示した「サーバの高負荷，および認証情報の漏洩リスク」の問題が解決される．この TP は総当たり試行のための演算パワーのみを提供すればよいため，クラウドサービスとして実運用できると考えられる．TP を用いた場合の認証の手順を以下に示す（図 4）．

【登録フェーズ】
(1) ユーザは $\boldsymbol{P_u}$ をクライアント端末に入力し，クライアント端末は $\boldsymbol{P_r}$ を生成する．ここで，$\boldsymbol{P_u}$ はユーザが入力する秘密情報，$\boldsymbol{P_r}$ は総当たり試行によって同定する補助情報を表す．
(2) クライアント端末は，$\boldsymbol{h}=\mathbf{H}(\mathbf{H}(\boldsymbol{P_u}|\boldsymbol{P_r}))$ をサーバに



(a) 登録フェーズ

(b) 認証フェーズ

図 4 第三者機関を利用した方式
Fig. 4 Procedure using Third Party.

送信する．

(3) サーバは，マスク用乱数 R_1，R_2 を生成し，クライアント端末に送信する．ここで，R_1，R_2 はそれぞれ P_u，P_r と同じビット長である．

(4) クライアント端末は，$h' = \mathrm{H}((P_u\ XOR\ R_1)|(P_r\ XOR\ R_2))$ をサーバに送信する．

(5) サーバは，h，h'，R_1，R_2 を，ユーザの認証情報として登録する．

【認証フェーズ】

(1) サーバは，R_1，R_2，h' をクライアント端末に送信する．

(2) ユーザは，P_u をクライアント端末に入力する．

(3) クライアント端末は，$P_u' = (P_u\ XOR\ R_1)$，h' を TP に送信する．

(4) TP は，$\mathrm{H}(P_u'|P_r')$ が h' と一致する P_r' $(= P_r\ XOR\ R_2)$ を総当たり試行によって求め，クライアント端末に送信する．

(5) クライアント端末は，P_r $(= P_r'\ XOR\ R_2)$ を計算し，$P = P_u|P_r$ を復元し，そのハッシュ値 $\mathrm{H}(P_u|P_r)$ をサーバに送信する．

(6) サーバは，受け取った $\mathrm{H}(P_u|P_r)$ のハッシュ値 $\mathrm{H}(\mathrm{H}(P_u|P_r))$ が，h と同一であれば認証成功とし，一致しなければ認証失敗とする．

マスク用乱数 R_1，R_2 によって，認証情報 P_u および P_r の TP への漏洩が防がれていることに注意されたい．なお，認証サーバと TP は結託しないという前提を置いている．また，この方式に，さらにソルトを加えることも可能である．

6. 安全性に関する考察

6.1 総当たり攻撃耐性

4.1 節で説明したように，提案方式の総当たり攻撃に対する耐性は，将来計算機速度が向上した場合にも保たれる．ここで，計算機性能の向上に合わせてエントロピを大きくすべき情報は P_r だけであるため，ユーザが記憶する P_u のエントロピは時代が変化しても一定であることに注意されたい．これは，秘密情報 P_u を知っている正規ユーザと知らない不正者の間における P の総当たり試行に要する時間の差は，計算機速度に関係なく，P_u のエントロピに依存するためである．このため，提案方式においては，十分なエントロピを持つ秘密情報 P_u を一度設定すれば，その後の計算機速度の向上に左右されず，安全にユーザ認証を行うことができる．

ただし，ある時点でのハッシュ値 $\mathrm{H}(\mathrm{H}(P))$ が不正者の手にわたってしまった場合，計算機性能の向上によって，そのハッシュ値に対する総当たり攻撃はいずれ必ず成功し，P $(= P_u|P_r)$ が不正者に漏れることになる．このため，P_u については，（エントロピの大きさはそのままでよいが）計算機性能の向上に応じてその値を更新する必要がある．

計算機性能の向上が今後もムーアの法則にほぼ従うと仮定するならば，計算機能力は 18 カ月で約 2 倍となる．つまり，攻撃能力が 2 倍になっても安全性を保てる程度に P_r のエントロピを確保しておけば，P_u の更新は 18 カ月ごとに行えばよいということになる．また，CPU のスペックは各 CPU メーカが公開しているため，その世代の計算機性能を定期的に調査して，必要に応じて認証情報を再登録するという方法をとることも可能であろう．なお，P_u のエントロピは P よりも小さいので，P_u の更新に関するユーザの負担は，P の更新と比べて小さい．

6.2 各種認証方式における秘密情報

4.1 節の試算によれば，提案方式において「不正者による認証情報全体の総当たり攻撃に要する時間は 1 年，かつ，正規ユーザによる認証に要する時間は 1 秒」の制約を満たすために必要となる秘密情報 P_u のエントロピ E_u は，3×10^7 通りである．本節では，各種のユーザ認証における E_u に対して，その妥当性を考察する．

6.2.1 パスワード認証

一般に，パスワードとして使用可能な文字 95 種から秘密情報を作成する場合，$95^4 \fallingdotseq 8.1 \times 10^7$ より，ユーザが P_u として 4 文字以上を記憶することで，「不正者による認証情報全体の総当たり攻撃に要する時間は 1 年，かつ，正規ユーザによる認証に要する時間は 1 秒」の制約が満たされることが分かる．

6.2.2 画像認証

たとえば，Cognometric 方式において，15 枚の画像の中から 7 枚の画像を正しい順序で選択する場合，認証試行 1 回あたりのエントロピは ${}_{15}\mathrm{P}_7 \fallingdotseq 3.2 \times 10^7$ となる．しかし，この場合は，ユーザは秘密情報として 7 枚の画像とその順番を記憶しなければならず，このままでは現実的とはいえない．画像認証において，ユーザが覚えるべき秘密情報に対して 3×10^7 通りのエントロピを確保するためには，何らかの工夫が必要である．

6.2.3 CAPTCHA

画像ベースの CAPTCHA においては，15 枚の画像の中から 7 枚を正しい順序で選択するような問題を生成することができれば，認証試行 1 回あたりのエントロピは ${}_{15}\mathrm{P}_7 \fallingdotseq 3.2 \times 10^7$ となる．たとえば，15 種類の動物の画像を表示し，その中から「干支の順（子，丑，寅，…）に動物を 7 匹クリックせよ」というような CAPTCHA の問題を構成してやれば，秘密情報（CAPTCHA の問題に対するユーザの解答）として 3×10^7 通りのエントロピを確保できることが分かる．

6.2.4 生体認証

たとえば，指紋 1 指を用いて 99.99%の精度（他人受入率 0.01%）で本人認証を行うことができる指紋認証システムがあったとする．このシステムに指紋 2 指を登録し，AND 型の認証を行った場合，その他人受入率は 1×10^{-8} となるため，指紋 2 指で 1×10^8 通りのエントロピを確保できることになる．したがって，提案方式は，生体認証においても，実用的な運用範囲（2 指の利用）内で「不正者による認証情報全体の総当たり攻撃に要する時間は 1 年，かつ，正規ユーザによる認証に要する時間は 1 秒」の制約を満たすことができると期待される．

また，現在，理論的な観点からも生体情報のエントロピの評価が進められており，たとえば文献 [19] では，虹彩認証においては，虹彩情報のエントロピが最低でも 100 ビット程度（$\fallingdotseq 1.2 \times 10^{30}$ 通り）であることが報告されている．この点からも，提案方式を利用するにあたっての要件となる 3×10^7 通りのエントロピの確保は可能であると考えられる．

6.3 秘密情報のエントロピの設定

提案方式では，認証情報全体 $\boldsymbol{P}$ $(= \boldsymbol{P_u}|\boldsymbol{P_r})$ の総当たりに対しては膨大な時間を要し，補助情報 $\boldsymbol{P_r}$ のみの総当たりに対しては正規ユーザがストレスなく待てる時間内で終了するように，$\boldsymbol{P_u}$ および $\boldsymbol{P_r}$ のエントロピを設定する．

本論文では，「不正者による認証情報全体の総当たり攻撃に要する時間は 1 年，かつ，正規ユーザによる認証に要する時間は 1 秒」という制約を例にとって，$\boldsymbol{P_u}$ および $\boldsymbol{P_r}$ のエントロピを具体的に算定したが，ここまでの議論は，正規ユーザと不正者の計算機能力が同程度であることを前提としていた．しかし，5.2.1 項のようにクライアント端末で総当たり試行を行う場合などには，正規ユーザと不正者では使用する計算機の性能は非対称となることが考えられる．

一般的には，正規ユーザはスマートフォンなどの計算能力の小さな端末を利用することが期待されるのに対し，不正者はハイスペックな計算機を用意したり，クラウドコンピューティングサービスを利用したりするであろう．このような場合は，正規ユーザと不正者の計算機能力の比についても考慮した上で，$\boldsymbol{P_u}$ や $\boldsymbol{P_r}$ のエントロピを算定する必要がある．たとえば，4.1 節の試算（$\boldsymbol{P_u}$ のエントロピは 3×10^7 通り）のケースにおいて，正規ユーザの 100 倍の計算能力を持つ不正者が総当たりに 1 年間を要するようにしたい場合は，$\boldsymbol{P_u}$ のエントロピが 3×10^9 通り以上となるように $\boldsymbol{P_u}$ を設定することとなる．

なお，5.2.3 項のように第三者機関を用いる場合であれば，不正者が用意できる計算機資源と同程度以上の計算能力を所有する第三者機関を運用してやることにより，正規ユーザと不正者の計算能力比の考慮は無用となると考えられる．

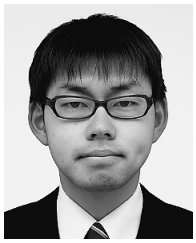
7. まとめと今後の課題

本論文では，ユーザ認証に対する攻撃手法である総当たり攻撃を逆に利用し，計算機を使って認証情報の一部を補完することによって，安全性と利便性を両立するユーザ認証方式を提案した．提案方式の場合は，不正者の攻撃能力が向上する分，正規ユーザの認証能力も向上する．このため，将来的に計算機の能力がいかに向上したとしても，「不正者は，秘密情報の分だけ正規ユーザよりも総当たり試行に要する時間が大きくなる」という正規ユーザ優位の状況が保たれる．

提案方式におけるユーザ認証の秘密情報のエントロピに関して考察し，ユーザが入力する秘密情報としては 3×10^7 通りのエントロピが確保できれば，「不正者による認証情報全体の総当たり攻撃に要する時間は 1 年，かつ，正規ユーザによる認証に要する時間は 1 秒」という制約が満たされることを示した．ただし，これは，総当たり攻撃を想定した場合の評価であることに注意しなければならない．たとえばパスワード認証において，ユーザが秘密情報として推測しやすい文字列を設定してしまうと，辞書攻撃などによる脆弱性が顕在化する．このため，提案方式においても，既存のパスワード認証と同様に，推測されにくい秘密情報を設定することがユーザに求められる．また，ソーシャルエンジニアリングなどをはじめとした不正者による「計算機能力を利用する以外の攻撃」に対する耐性についても検討する必要がある．

参考文献

[1] Scary Logins: Worst Passwords of 2012 – and How to Fix Them, available from ⟨http://splashdata.com/press/PR121023.htm⟩.
[2] Takada, T., Onuki, T. and Koike, H.: Awase-E: Recognition-based Image Authentication Scheme Using Users' Personal Photographs, Innovation in Information Technology (2006).
[3] Two Factor Authentication, Graphical Passwords - Passfaces, available from ⟨http://www.realuser.com/⟩.
[4] Wiedenbeck, S., Waters, J., Birget, J.-C., Brodskiy, A. and Memon, N.: PassPoints: Design and longitudinal evaluation of a graphical password system, *International Journal of Human-Computer Studies*, Vol.63, pp.102–127 (2005).
[5] ピクチャパスワードでサインインする：Microsoft Windows，入手先 ⟨http://windows.microsoft.com/ja-jp/windows-8/personalize-pc-tutorial#1TC=t1⟩.
[6] Chiasson, S., Forget, A., Biddle, R. and Oorschot, V.P.C.: User interface design affects security: Patterns in click-based graphical passwords, *International Journal of Information Security*, Vol.8, No.6, pp.387–398 (2009).
[7] PWNtcha – Caca Labs, available from ⟨http://caca.zoy.org/wiki/PWNtcha⟩.
[8] ASIRRA – Microsoft Research, available from ⟨https://research.microsoft.com/en-us/um/redmond/projects/asirra/⟩.
[9] Yamamoto, T., Suzuki, T. and Nishigaki, M.: A proposal of Four-panel cartoon CAPTCHA, *Proc. IEEE International Conference on Advanced Information Networking and Applications 2011*, pp.159–166 (2011).
[10] 高橋健太：テンプレート保護と生体認証基盤，電子情報通信学会ソサイエティ大会講演論文集 2012 年（基礎・境界），pp.SS-53–SS-54 (2012).
[11] 佐藤優人，加藤貴司，ベッド B. ビスタ，高田豊雄：画像連想語呂合わせパスワードを利用したパスワード作成支援システムの改良手法の提案，2010 年暗号と情報セキュリティシンポジウム予稿集，1E 2-4 (2010).
[12] パスワード管理ソフト ID Manager，入手先 ⟨http://www.woodensoldier.info/soft/idm.htm⟩.
[13] IPA ISEC セキュア・プログラミング講座：Web アプリケーション編第 2 章アクセス制限対策：ユーザー認証，入手先 ⟨https://www.ipa.go.jp/security/awareness/vendor/programmingv2/contents/101.html⟩.
[14] Provos, N. and Mazieres, D.: A Future-Adaptable Password Scheme, USENIX Annual Technical Conference (1999).
[15] Schneier, B.: Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish), *Proc. Cambridge Security Workshop Fast Software Encryption*, pp.191–204, Springer Verlag (1994).
[16] How Rainbow Tables work, available from ⟨http://kestas.kuliukas.com/RainbowTables/⟩.
[17] ニーモニックニュース：ニーモニックニュース 2012 年 6 月第 2 号，入手先 ⟨http://mneme.blog.eonet.jp/default/2012/06/post-b03a.html⟩.
[18] Morris, R. and Thompson, K.: Password Security: A Case History, *Comm. ACM*, Vol.22, No.11, pp.594–597 (1979).
[19] 赤尾直彦，披田野清良，小松尚久：最小距離エントロピーを用いた虹彩情報の情報量推定に関する一考察，2011 年暗号と情報セキュリティシンポジウム予稿集，3E 1-4 (2011).



兼子 拓弥

2013 年静岡大学情報学部情報科学科卒業．現在，同大学大学院修士課程．情報セキュリティに関する研究に従事．

本部 栄成

2011 年静岡大学情報学部情報科学科卒業．2013 年同大学大学院修士課程修了．在学中，情報セキュリティの研究に従事．

高橋 健太 （正会員）

1998 年東京大学理学部情報科学科卒業．2000 年同大学大学院修士課程修了．2012 年同大学院情報理工学研究科博士課程修了．博士（情報理工学）．2000 年（株）日立製作所入社．以来，バイオメトリクスおよび情報セキュリティの研究開発に従事．2001 年情報処理学会高度交通システム研究会優秀論文賞受賞．平成 20 年度情報処理学会論文賞受賞．電子情報通信学会会員．

西垣 正勝 （正会員）

1990 年静岡大学工学部光電機械工学科卒業．1992 年同大学大学院修士課程修了．1995 年同博士課程修了．日本学術振興会特別研究員（PD）を経て，1996 年静岡大学情報学部助手．同講師，助教授の後，2006 年より同創造科学技術大学院助教授．2007 年同准教授，2010 年同教授，2013 年同大学院情報学研究科教授．博士（工学）．情報セキュリティ全般，特にヒューマニクスセキュリティ，メディアセキュリティ，ネットワークセキュリティ等に関する研究に従事．2013 年より情報処理学会コンピュータセキュリティ研究会主査．