Vol.2012-CSEC-58 No.47
Vol.2012-SPT-4 No.47
2012/7/20

総当り試行を利用した Entropy-Enhanced Password 認証

兼子拓弥† 本部栄成†† 西垣正勝†††

現在，本人確認の手段として広く普及しているパスワード認証には，安全性（攻撃耐性）と利便性（記憶負荷）との間にトレードオフが存在する．ここで，不正者の攻撃能力がパスワード認証の安全性に大きく影響することに鑑み，パスワード認証に対する攻撃手法を「パスワード認証のトレードオフ」をコントロールする手段として逆に利用することができるのではないかと考えた．そこで本稿では，パスワード認証に対する攻撃手法の 1 つである総当り攻撃を利用して，パスワード空間のエントロピを疑似的に増加させる手法を提案する．これにより，不正者の攻撃能力に応じた攻撃耐性を確保しつつ，ユーザの記憶負荷の低減を達成したパスワード認証が実現される．
キーワード: パスワード認証 エントロピ強化 総当たり攻撃 記憶負荷

An Entropy-Enhanced Password Authentication that uses Brute Force Attack

TAKUYA KANEKO† EISEI HONBU†† MASAKATSU NISHIGAKI†††

Password authentication has trade-off between security (attack resistance) and usability (memory load). The security of password authentication is depending a great deal on the adversary's attack ability, and therefore, it is expected that attack technique can be used to control this trade-off. So this paper proposes to effectively increase password entropy by using brute force attack that is one of typical attack techniques against password authentication. The proposed scheme achieves to improve password authentication so that the resistance against attack is ensured even when adversary's attack ability increase and that the burden of memorizing passwords is reduced.
Key words: Password Authentication, Entropy enhancement, Brute force attack, memory load

1. はじめに

現在，本人確認の手段として広く普及しているパスワード認証には，安全性（攻撃耐性）と利便性（記憶負荷）との間にトレードオフが存在する．パスワード認証に対する典型的な攻撃手法である総当り攻撃および辞書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必要がある．しかしながら，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．

パスワード認証の安全性を保つためには，パスワード空間のエントロピが不正者の攻撃能力よりも大きくなるように，パスワードが設定されるべきである．これを実現するにあたっては，パスワード空間を大きくするアプローチと攻撃能力を低下させるアプローチが存在する．

前者は，語呂合わせなどの記憶術を活用して長く複雑なパスワードを覚える方法や，レインボー攻撃[1]対策としてソルトを付加する方法[2]などが，これに当たる．

後者は，パスワードに対する攻撃試行の一回当たりの所要時間を増加させる方法がこれに当たる．オンライン攻撃に対しては，タールピット（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[a][3]）やアカウントロックアウト（規定回数以上の認証失敗の場合にアカウントを無効化する[3]）が有効に機能する．オフライン攻撃には，認証アルゴリズムの性能をあえて劣化させることによって認証アルゴリズムそのものにタールピットを仕掛けるなどの方法が採られる．その典型例である bcrypt [4]では，ハッシュ計算の所要時間を意図的に増大させたハッシュ関数を設計し，このハッシュ関数によって計算したパスワードハッシュを比較することによって認証可否を判定する．

ソルトと bcrypt を併用することによって，オフライン攻撃にもレインボー攻撃にも対抗可能となる．しかし，bcrypt のように暗号関数そのものに手を加える方法の場合は，設計負荷の高さや互換性の欠如が問題となる．そこで本稿では，攻撃手法の 1 つである総当り攻撃を逆に利用することで，簡素かつ汎用的なタールピットを構築し，安全性（攻撃耐性）と利便性（記憶負荷）を両立するパスワード認証方式を実現する．

総当り試行に有する時間は，パスワード長に対して指数関数的に増加する．そのため，ある程度のパスワード長までの試行時間は短く，それ以上のパスワード長に対する試行時間は非常に長くなる性質がある．この性質を利用し，ユーザはパスワードの一部を記憶し，残りの文字列を総当り試行で補完することで，ユーザの記憶負荷の軽減と実用

† 静岡大学情報学部
Faculty of Informatics, Shizuoka University
†† 静岡大学大学院情報学研究科
Faculty of Informatics, Shizuoka University
††† 静岡大学創造科学技術大学院
Graduate School of Science and Technology, Shizuoka University

a 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．本稿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．

時間内での認証完了を同時に達成する．

2. 関連研究

現在までにパスワード認証における安全性と利便性のトレードオフを解消する手法は多く提案されている．ここでは，パスワード管理ツール，ソルト，bcrypt の 3 つを既存方式としてあげる．

2.1. パスワード管理ツール

複数のパスワードの管理を支援する方式としてパスワード管理ツール[5]が挙げられる．パスワード管理ツールとは，複数の ID とパスワードの組合せ（以下，「認証情報」と呼ぶ）をクライアント PC 内で管理することで，ユーザがそれらの認証情報を記憶することなくパスワード認証を行うためのツールである．この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クライアント 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，すべての認証情報を使用することができる．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覚えることなく各サービスへログインすることができる．

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,既存のパスワード認証を用いることとなる.すなわち，既存のパスワード認証の課題である安全性と利便性のトレードオフを解決する方法ではない．また，外部サービスにログインする際のパスワードなど，通常であればサーバ側にて管理されるべき認証情報についても，その全てがクライアント PC にて保管されることになるため，不正者による PC への侵入には注意が必要である．認証情報は暗号化されているため，認証情報の漏洩にはある程度の耐性があると考えられるが，認証情報が破壊・削除されるとサービス不能に陥る．

2.2. 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（平文のすべての組み合わ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[1]．現在，70～80 ビット程度以上のパスワードでなければ，レインボー攻撃に脆弱であるといわれている[6].

レインボー攻撃に対する対策としてソルトが知られている[2]．ソルトとは，パスワード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パスワードのエントロピが増加し，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．なお，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，正規ユーザの記憶負荷は増加しない．すなわちソルトは，パスワードのハッシュ値の逆計算に対するエントロピを増加させているだけであり，パスワードの総当り攻撃に対するエントロピを増加させているわけではないことに留意されたい．

2.3. bcrypt

パスワード認証の安全性を保つためには，パスワード空間のエントロピが不正者の攻撃能力よりも大きくなるように，パスワードが設定されるべきである．ソルトは，パスワード空間を大きくすることによってこれを達成する方法である．一方，攻撃能力を低下させることによって，これを達成する方法も存在する．

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，パスワード認証の総当り攻撃に対する耐性を向上させる手法を提案した[4]．bcrypt と名付けられたこの手法は，Blowfish [7]型のブロック暗号における部分鍵生成演算の繰り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の検査）当たりの所要時間が増加し，その分，不正者が単位時間当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る方法であるといえる．これによって不正者の総当り攻撃能力が減衰し，パスワード空間のエントロピが同じであっても総当り攻撃耐性が向上する．

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. 提案方式

不正者の攻撃能力がパスワード認証の安全性に大きく影響することに鑑み，パスワード認証に対する攻撃手法を「パスワード認証における安全性（攻撃耐性）と利便性（記憶負荷）のトレードオフ」をコントロールする手段として逆に利用することができるのではないかと考えた．そこで本稿では，パスワード認証に対する攻撃手法の 1 つである総当り攻撃を利用して，簡素かつ汎用的なタールピットを構築し，安全性（攻撃耐性）と利便性（記憶負荷）を両立

Vol.2012-CSEC-58 No.47
Vol.2012-SPT-4 No.47
2012/7/20

するパスワード認証方式を提案する．

3.1. コンセプト

総当り試行には，全文字列を試行するために必要となる時間は文字列長に応じて指数関数的に増加するという特徴がある．そのため，ある文字列長までの試行にかかる時間は短いが，それ以上の文字列長の試行にかかる時間は膨大となる．この特徴を用いて，ユーザはパスワードの一部を記憶し，残りの文字列を総当り試行で補完することで，ユーザの記憶負荷の軽減と実用時間内での認証完了を同時に達成する．

例えば，現在の計算機が 1 秒間に10^8通りの総当りが可能だとすると，英数記号 95 種の 8 文字のパスワードを総当りするには$95^8 \fallingdotseq 6.6 \times 10^{15}$通りの試行をする必要があるため約 2 年半かかるが，4 文字の総当りは$95^4 \fallingdotseq 8.1 \times 10^7$通りの試行で済むため 1 秒未満で終了する．これを利用し，8 文字のパスワードの内，4 文字だけを正規ユーザに記憶してもらう．正規ユーザは残る 4 文字を総当りで同定するだけで良いので，総当り試行は許容時間内（1 秒未満）で終了する．一方，攻撃者は 8 文字のパスワードすべてを同定する必要があるため，総当り試行に要する時間が膨大（約 2 年半）になり，パスワードは十分な総当り攻撃耐性を有する．

また，提案方式においては，正規ユーザが記憶しない文字数を変えることによって，パスワード全体のエントロピを任意に設定できる．よって，将来，計算機速度が向上し，総当り試行に要する時間が短縮されたとしても，正規ユーザが記憶する文字数を増やすことなく，総当り攻撃に対する耐性を維持することができる．例えば，上記の例において計算機速度が 95 倍になった場合には，パスワードを 9 文字に増やして，その内の 4 文字をユーザに記憶してもらうようにする．この結果，正規ユーザが行う「残る 5 文字の総当り試行」と不正者に求められる「全 9 文字の総当り攻撃」の所要時間はそれぞれ 1 秒未満，2 年半のまま維持される．

この関係を図 1 と図 2 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（ここでは仮に 1 年と仮定する）の範囲内で実行可能な総当り攻撃回数もそれに応じて増加する（図 1 の①）．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 2 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，例えば「1 秒」を感じる時間の長さは，現在（図 1，図 2 の t1）も次世代（図 1，図 2 の t2）もほぼ同じ時間を保つ（図 2 の⑤）．以上より，正規ユーザが一瞬と感じる時間（ここでは仮に 1 秒と仮定する）の範囲内で実行可能な総当り試行回数は，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②）．よって，「1 秒間の総当り試行」というタールピットを採用することにより，不正者の攻撃能力（図 1 の①）は 1 秒間の総当り試行（図 1 の②）の分だけ減じられ，不正者の実効的な攻撃能力は時代によらず一定値になる（図 1 の④）．パスワードのエントロピがこの値（図 1 の④）を超えていれば，パスワード認証の安全性が確保されることになる．人間の記憶力も時を越えてさほど変化することはない（図 1 の③）が，タールピットによって不正者の実効的な攻撃能力（図 1 の④）が一定となるため，常にパスワード認証の安全性を確保することが可能である．

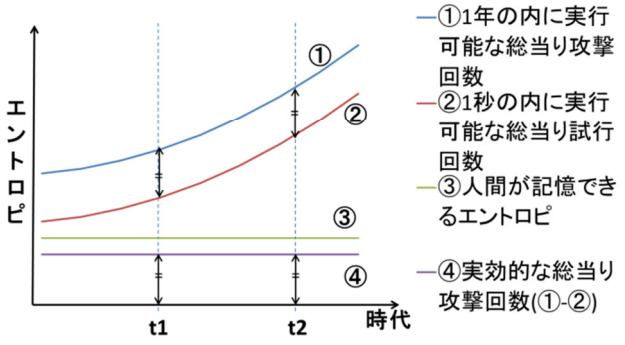

図 1　時代による処理能力（エントロピ）の推移

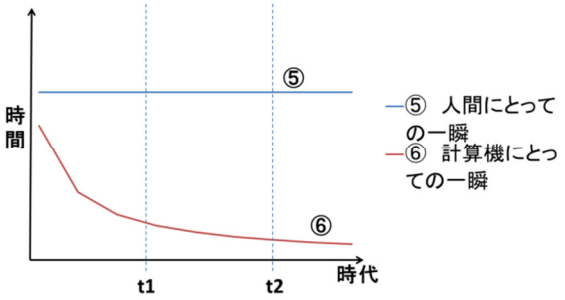

図 2　時代による処理時間の推移

3.2. 認証手順

提案方式における，具体的な登録と認証の手順を以下に記す．また，それぞれを図 3 と図 4 に図示する．

(1) **登録フェーズ**

I. ユーザ（被認証者）は，端末に ID と文字列P_uを入力する．P_uが，ユーザが記憶すべき情報となる．3.1 節で説明した例の場合，P_uの文字列長は 4 文字となる．

II. 端末は，乱数P_rとソルト S を生成し，P_u，P_r，S の連結$P_u|P_r|S$ をパスワード P とする．3.1 節で説明した例の場合，P_rのエントロピは95^4となる．S のエントロピは，レインボー攻撃に十分対抗できる大きさとする．

III. 端末は，H(P)を計算する．

IV. 端末は，ID，H(P)，S をサーバ（認証者）に送信する

V. サーバは，受け取った ID，H(P)，S を関連付けてデータベースに保存する．

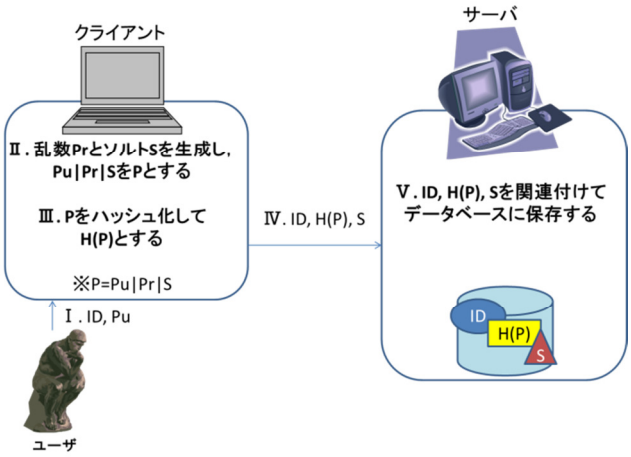

図 3　登録フェーズ

(2)　認証フェーズ

I. ユーザは，端末に ID とP_uを入力する．

II. 端末は，ID をサーバに送信する．

III. サーバは，受け取った ID に関連付けられている H(P)と S を取り出す．

IV. サーバは，2 つの乱数R_1，R_2を生成する．R_1およびR_2のエントロピは，レインボー攻撃に十分対抗できる大きさとする．

V. サーバは，H(H(P)|R_1),S,R_1,R_2を端末に送信する．

VI. 端末は，H(H(P_u|P_r|S)|R_1)が H(H(P)|R_1)と一致するまでP_rを総当りする．3.1 節で説明した例の場合，P_uが正しければ約 1 秒で正しいP_rが見つかる．

VII.　端末は，H(H(P_u|P_r|S)|R_2)をサーバに送信する．

VIII.　サーバは，端末から受け取った H(H(P_u|P_r|S)|R_2)と自らが算出した H(H(P)|R_2)が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．

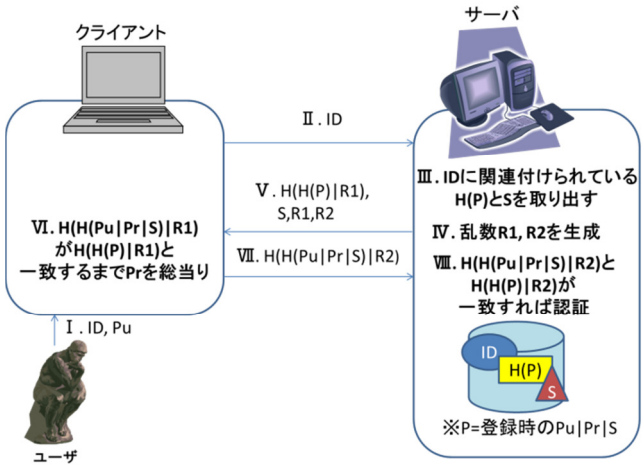

図 4　認証フェーズ

4.　安全性評価

提案方式の安全性を，パスワード認証に対する攻撃方法への耐性の面から考察する．

4.1.　リプレイ攻撃に対する耐性

3.2 節で示した認証手順の認証フェーズにおいて，乱数R_1，R_2が導入されており，認証手順全体がチャレンジ&レスポンス型のパスワード認証の構成となっている．チャレンジは認証の度にサーバにて新たに生成されるものであり，毎回変更される．そのため，同じパスワードを用いる場合でも端末からのレスポンスは毎回変わることになる．以上より，不正者が提案方式の認証フェーズを盗聴し，ある時点における端末からのレスポンスを入手できたとしても，それ以降の認証試行の際にそのレスポンスを利用することはできない．

4.2.　レインボー攻撃に対する耐性

3.2 節で示した認証手順においては，提案方式の本質である「1 秒間の総当り試行」というタールピットに加え，レインボー攻撃対策としてのソルト S が併用されている．また，3.2 節で示した認証手順の認証フェーズにおいては，乱数R_1，R_2が導入されており，R_1が H(H(P)|R_1)の H(P)に対するソルトとして，R_2が H(H(P_u|P_r|S)|R_2)の H(P_u|P_r|S)に対するソルトとして，それぞれ利用されている．以上より，十分大きなエントロピを持つ S，R_1，R_2を用いることによって，提案方式のレインボー攻撃に対する耐性は保証される．

4.3.　総当り攻撃に対する耐性

提案方式では，$P_u|P_r$の総当りに対しては膨大な時間を要し，P_rの総当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，P_uおよびP_rのエントロピを設定する．このため，提案方式は$P_u|P_r$の総当り攻撃に対する耐性を持っていると言える．

3.1 節で説明したように，提案方式の総当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．コンピュータの誕生以来，CPU 性能は飛躍的に向上しており，計算機速度は 10 年で約 9 倍になっている[8]．計算機速度が向上すると，$P_u|P_r$の総当りに要する時間は短くなるため，その分だけ総当り攻撃に対する攻撃耐性が低下してしまう．$P_u|P_r$のエントロピを E，P_uのエントロピをE_u，P_rのエントロピをE_rとし，仮に計算機速度が 2 倍になった場合を想定する．このとき，計算機速度に伴い，P_rの総当りに要する時間は$\frac{1}{2}$となる．このため，P_rのビット数を増加させてE_rのエントロピを 2 倍にしても，正規ユーザが行うP_rの総当り試行に要する時間は維持される．また，$E = E_u \times E_r$なので，E_rを 2 倍にすることにより，E も 2 倍となり，不正者が実行すべき$P_u|P_r$全体の総当りに要する時間も維持される．

4.4. パスワード長の評価

まず，不正者による$P_u|P_r$に対する総当り攻撃について考える．ここでは，総当りに 1 年（約 3200 万秒）を必要とするとき総当り攻撃に対する十分な耐性を有すると定義する．現在の計算機が 1 秒間に X 通りの総当りが可能であるとすると，E としては約 3200 万×X 通り以上のエントロピを持てば安全といえる．

次に，正規ユーザによるP_rに対する総当り試行について考える．E_rは，総当りが終了するまでユーザがストレスなく待てる範囲で，なるべく大きくするべきである．ここでは，約 1 秒で総当りが終了するならばユーザの利便性は損なわれないと定義する．現在の計算機が 1 秒間に X 通りの総当りが可能であるとすると，E_rとしては X 通りのエントロピとなるようにE_rを設定する．

最後に，正規ユーザが記憶すべきP_uについて考える．$P_u|P_r$のエントロピ E が約 3200 万×X 通り，P_rのエントロピE_rが X 通りになるようにするため，P_uのエントロピE_uは約 3200 万通り以上にすれば良いということになる．文字列P_uの各 1 文字を英数記号 95 種とすると，P_uを 4 文字に設定したときに，総当り試行数は$95^4 \fallingdotseq 8.1 \times 10^7$となる．よって，ユーザが記憶すべき文字列$P_u$は最低 4 文字とすることで，総当り攻撃に対する攻撃耐性を十分に確保できると考えられる．

将来計算機速度が向上し，1 秒間に実行可能な総当り試行回数 X が増加したとしても，それに応じてP_rのエントロピE_rを大きくすることによって，文字列P_uの文字数（最低 4 文字）を変更することなく，総当り攻撃耐性を維持することができることに留意されたい．

5. まとめと今後の課題

本稿では，総当り攻撃を逆に利用してパスワードの一部を補完することによって，簡素かつ汎用的なタールピットを構築し，安全性と利便性を両立するパスワード認証方式を実現した．

提案方式であれば，ユーザが最低 4 文字の文字列P_uを記憶するだけで，総当り攻撃に十分な耐性が確保できることを示した．しかし，ユーザがP_uとして推測しやすい文字列を設定してしまうと，辞書攻撃による脆弱性が顕在化する．このため，既存のパスワード認証と同様に，ユーザには推測されにくいパスワードP_uを設定することが求められる．

現段階ではユーザが記憶すべき文字列P_uは 4 文字であるが，1 文字当たりの文字種類を増やすことによって，この文字数を減らすことができると考えられる．今後は，ひらがな，カタカナ，漢字といった，2 バイト文字を使用することを可能にすることにより，提案方式の記憶負荷を更に低減することについて検討していく．

謝辞　本稿を執筆するにあたり，株式会社日立製作所横浜研究所高橋健太様から，有益なるご助言を頂きました．心より感謝申し上げます．

参考文献

1) How Rainbow Tables work
http://kestas.kuliukas.com/RainbowTables/
2) Robert Morris and Ken Thompson: Password Security: A Case History, Communications of the ACM, 22(11): 594-597, November (1979).
3) 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 : ユーザー認証対策
http://www.ipa.go.jp/security/awareness/vendor/programmingv2/contents/101.html
4) Niels Provos and David Mazieres: A Future-Adaptable Password Scheme, USENIX Annual Technical Conference (1999)
5) パスワード管理ソフト ID Manager
http://www.woodensoldier.info/soft/idm.htm
6) 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
http://mneme.blog.eonet.jp/default/2012/06/post-b03a.html
7) Bruce Schneier: Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish), Fast Software Encryption, Cambridge Security Workshop Proceedings, pp. 191-204. Springer-Verlag, December (1993).
8) 青山貞一, 鷹取敦: 究極のパソコン活用 ～3 次元流体計算の可能性と課題～, 武蔵工業大学 環境情報学部情報メディアセンタージャーナル, 第 8 号, pp.95-100 (2007)