A Study o n Formularization of Computer Aided Computational Security

SCIS 2016 2016 Symposium on Cryptography and Information Security
Kumamoto, Japan, Jan. 19- 22, 2016
The Institute of Electronics, Information and Communication Engineers

計算機援用計算量的安全性の定式化に関する一検討
A Study on Formularization of Computer Aided Computational Security

神農 泰圭* Yasuyoshi Jinno　兼子 拓弥** Takuya Kaneko　高橋 健太‡ Kenta Takahashi
尾形 わかは§ Wakaha Ogata　西垣 正勝† Masakatsu Nishigaki

あらまし 計算量的安全性に基づくセキュリティシステムにおいては，CPU の計算機能力が時代とともに進化するという経時的な要因と，攻撃者が正規ユーザと比較してどれだけ大きい計算コストをかけるかという相対的な要因の両者を考慮して秘密情報のエントロピを確保することが必要となる．しかし，CPU の計算機能力の進化は攻撃者だけでなく正規ユーザにも恩恵を与えるものである．そこで本稿では，現在の計算量的安全性の定式化を拡張し，正規ユーザによる CPU の計算機能力の利用に起因するエントロピを，正規ユーザが所持する秘密情報のエントロピに上乗せして扱える計算機援用計算量的安全性の枠組みを提案する．

キーワード 計算量的安全性　計算機援用セキュリティ　ユーザ認証　公開鍵暗号　デジタル署名

1　はじめに

セキュリティシステムが「計算量的に安全」とは，攻撃者によるなりすましや暗号解読に非常に時間がかかるようにすることによって，安全性が確保されていることを言う[4]．なりすましや暗号解読に要する時間は，秘密情報のエントロピに依存するため，これをセキュリティパラメータとして捉え，攻撃成功までに膨大な時間がかかるように，セキュリティシステムのパラメータ（秘密情報）の大きさを適切に設定する．その際，CPU の計算機能力が時代とともに進化するという経時的な要因と，攻撃者が正規ユーザの計算コストと比較してどれだけ大きい計算コストをかけるかという相対的な要因の両方を考慮して秘密情報のエントロピを確保することが必要となる．

しかし，計算機能力の進化は，攻撃者だけでなく正規ユーザにも恩恵を与えるものであり，本来，攻撃者のみが有利となるものではない．実際，正規ユーザの計算機能力を用いて認証や暗号のセキュリティ強化を達成した方式として，計算機援用ユーザ認証[1]およびパスワードベース鍵生成関数（bcrypt [2]，PBKDF2 [3]）などが提案されている．本稿では，このような「正規ユーザの計算機能力を用いて秘密情報のエントロピを上乗せする」というアプローチによるセキュリティ確保の概念を計算機援用セキュリティと呼び，その計算量的安全性の定式化を行う．

2　計算機援用セキュリティシステム

本章では，計算機援用セキュリティシステムの具体的なインスタンスである計算機援用ユーザ認証[1]，パスワードベース鍵生成関数 bcrypt [2]，PBKDF2 [3]について説明し，その安全性について議論する．

2.1 計算機援用ユーザ認証

計算機援用ユーザ認証とは，認証情報の一部が正規ユーザにも未知となっていて，その未知情報を総当たり試行によって求めるというユーザ認証方式である．

* 静岡大学情報学部，〒432-8011 静岡県浜松市中区城北 3-5-1, Faculty of Informatics, Shizuoka University, 3-5-1, Johoku, Naka, Hamamatsu, Shizuoka , 432-8011
** 静岡大学大学院情報学研究科，〒432-8011 静岡県浜松市中区城北 3-5-1, Graduate school of Informatics, Shizuoka University, 3-5-1, Johoku, Naka, Hamamatsu, Shizuoka , 432-8011
‡ 株式会社日立製作所研究開発グループセキュリティ研究部，Hitachi, Ltd.,R&D Group, Security Research Dept.
§ 東京工業大学大学院イノベーションマネージメント研究科，〒152-8552 目黒区大岡山 2-12-1, Graduate school of Innovation Management, Tokyo Institute of Technology, 2-12-1, Ookayama, Meguro, Tokyo, 152-8552
† 静岡大学創造科学技術大学院，〒432-8011 静岡県浜松市中区城北 3-5-1, Graduate School of Science and Technology, Shizuoka University 3-5-1 Johoku, Naka, Hamamatsu, Shizuoka, 432-8011

認証手順：

p_uはユーザが所持すべき秘密情報，p_rは総当り試行によって求める認証情報，$p = p_u|p_r$である．登録フェーズにて，$H(H(p))$が認証サーバに登録されている．

1. サーバはクライアント端末に，$H(H(p))$を送信する
2. ユーザはクライアント端末に，p_uを入力する．
3. クライアント端末は$H(H(p_u|p_r))$と，$H(H(p))$が一致するようなp_rを総当り試行によって探索する．
4. クライアント端末は，$H(p_u|p_r)$をサーバに送信する．
5. サーバは，$H(H(p_u|p_r))$と$H(H(p))$が一致したらユーザを認証する．

ここで，$p_u|p_r$はp_uとp_rのビット列の連結を表し，$H(\cdot)$はハッシュ関数である．p_uとp_rのビット長をそれぞれk_u，k_rとする．

p_uを所持している正規ユーザは，未知であるp_rのみを総当たり試行によって求める形となるため，負担する計算コストは最大で2^{k_r}回である．一方，攻撃者にとってはp_uとp_rの両者が未知であるため，なりすましに必要となる総当たり試行は最大で$2^{k_r+k_r}$回の計算コストとなる．例えば文献[1]では，正規ユーザの計算コストが 1 秒以内，攻撃者の計算コストが 1 年以上となるように，p_uとp_rのエントロピ（ビット長）を決定している．

2.2 パスワードベース鍵生成関数

bcrypt および PBKDF2 は，パスワードを入力として暗号鍵を出力する関数であり，入力から出力を求めるに当たっての反復演算回数を可変とすることにより，鍵生成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．鍵生成に時間を要する分，攻撃者の攻撃試行の速度を減速することができ，これによって暗号鍵の生成源（パスワード）のエントロピ不足を補っている．

bcrypt の暗号鍵生成手順

1. パスワードp, ソルトs, 反復演算回数cを入力する．
2. p, sの暗号化を2^c回繰り返す．
3. 暗号化されたp, sを用いてシード文字列を ECB モードで暗号化する作業を 64 回繰り返す．
4. 3.の結果を暗号鍵として出力する．

PBKDF2 の暗号鍵生成手順

1. パスワードp, ソルトs, 反復演算回数cを入力する．
2. p, sを任意のハッシュ関数に入力し，U_1を得る．
3. p, U_1をハッシュ関数に入力し，U_2を得る．
4. p, U_2をハッシュ関数に入力し，U_3を得る．これをU_cを得るまで繰り返す．
5. $U_1 \sim U_c$の排他的論理和を求め，これを暗号鍵として出力する．

pを所持している正規ユーザは，2^c回（bcrypt）あるいはc回（PBKDF2）の反復演算によって暗号鍵が生成される．一方，攻撃者にとってはpが未知であるため，なりすましに必要となる反復演算は2^{k+c}回（bcrypt）あるいは$c \cdot 2^k$回（PBKDF2）となる．ここで，kはpのエントロピ（ビット長）である．

2.3 正規ユーザの計算機能力の利用

計算機能力は日々進化する．セキュリティシステムに対して攻撃を行う攻撃者は，計算機能力を用いて攻撃を行う．そのため，攻撃者からの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能力の進化に伴って増加させる必要がある．また，攻撃者は，正規ユーザよりも大きなコスト（一般的には任意の多項式時間で表される計算コスト）をかけて攻撃を行うため，これに耐え得るだけの秘密情報のエントロピが要求される．

しかし，CPU の計算機能力の進化は攻撃者だけでなく正規ユーザにも恩恵を与えるものであり，本来，攻撃者のみが有利になるものではない．実際，2.1 節，2.2 節で説明した計算機援用セキュリティシステムにおいては，将来的に CPU の計算機能力がn倍となるのに応じて，総当たり試行や反復演算の回数をn倍に増やしたとしてもユーザ認証や鍵生成に要する時間は変化しない．計算機援用ユーザ認証を具体例に採って説明しよう．ある時刻t_1において，正規ユーザの計算コストが 1 秒，攻撃者の計算コストが 1 年となるようにp_uとp_rのビット長k_u，k_rを決定した後，時刻t_2にて CPU の計算機能力が 2 倍になったとする．その時点で，p_rのビット長を$k_r + 1$ビットに増やしてやれば，「正規ユーザの計算コストが 1 秒，攻撃者の計算コストが 1 年」という状況が維持される．bcrypt や PBKDF2 においても同様である．

これに対し，従来の計算量的安全性においては攻撃者側の観点からのみの定式化がなされており，「正規ユーザによる計算機能力の利用」に起因する秘密情報のエントロピについては陽に考慮されていなかった．そこで本稿では，従来の計算量的安全性の定式化を拡張し，「正規ユーザの計算機能力を用いて秘密情報のエントロピを上乗せする」という計算機援用セキュリティの概念を包含した定式化へと変更する．

3 計算機援用計算量的安全性の定式化

本章では，計算機援用セキュリティシステムの計算量的安全性の枠組みを定式化する．ここで，暗号プリミティブに応じて，その安全性は予測困難型（一方向性関数，MAC，署名等）あるいは識別困難型（擬似乱数生成器，擬似ランダム関数，公開鍵暗号等）の評価がなされる[5]ことに鑑み，ここでも攻撃者のアドバンテージの定式化を予測困難型と識別困難型の両者によって行う．

3.1 従来の計算量的安全性の定式化

従来の計算量的安全性の定式化を表 1 に示す．従来の計算量的安全性の定式化におけるセキュリティパラメータはkのみである．すなわち，$k[bit]$の秘密情報pを正規ユーザが所持する形となっている．正規ユーザの計算コ

ストは，セキュリティパラメータkを引数とする多項式時間アルゴリズム$Poly(k)$によりモデル化される．それに対し，攻撃能力が$T[bit]$で制限される任意の攻撃者Aの攻撃コストは，$Poly(k)\cdot 2^T$によりモデル化される．$k[bit]$の秘密情報pに対して，$T[bit]$の攻撃能力を有する任意の攻撃者Aのアドバンテージと計算量的安全性は以下のように定義される．

- 予測困難型：

$$Adv_{A[T]}(k) = \frac{1}{2^{k-T}}$$

- 識別困難型：

$$Adv_{A[T]}(k) = \frac{1}{2^{k-T}} - \frac{1}{2^k}$$

定義 3.1 $k[bit]$の秘密情報を攻撃する，攻撃能力が$T[bit]$で制限されるどのような攻撃者Aに対しても，アドバンテージが，$Adv_{A[T]}(k) < \varepsilon(k)$である時，$T[bit]$計算量的安全であると言う．$\varepsilon$は無視できる（negligible）関数[4]である．

表 1　従来の計算量的安全性の定式化

セキュリティパラメータ	k
秘密情報	$p(k[bit])$
正規ユーザの計算コスト	$Poly(k)$
攻撃者の攻撃コスト	$Poly(k)\cdot 2^T$
攻撃者のアドバンテージ	$Adv_{A[T]}(k)$

3.2 計算機援用計算量的安全性の定式化

計算機援用セキュリティシステムの計算量的安全性の定式化を表 2 に示す．新定式化では，「正規ユーザによる CPU の計算機能力の利用に起因するエントロピ」に対応する新たなセキュリティパラメータk_{uc}を導入する．すなわち新定式化では，正規ユーザ自身による$k_{uc}[bit]$分の計算コストの負担により，セキュリティシステムの秘密情報は$k + k_{uc}[bit]$に増強される．これは，秘密情報が$p + p_{uc}$に伸張されたことに相当する．正規ユーザが所持する秘密情報は$k[bit]$のpのみであり，$k_{uc}[bit]$分の計算コストは正規ユーザがシステムを使用する際に，その都度，転嫁される形となるため，正規ユーザの計算コストは$Poly(k, k_{uc})\cdot 2^{k_{uc}}$となる．一方，$T[bit]$の攻撃能力を有する任意の攻撃者$A$の攻撃コストは，$Poly(k, k_{uc})\cdot 2^T = Poly(k, k_{uc})\cdot 2^{k_{uc}}\cdot 2^r$によりモデル化される．$T = k_{uc} + r$の記述は，$k_{uc}[bit]$分の計算コストの負担は攻撃者にものしかかることを表しており，$r[bit]$の部分は，「攻撃者が正規ユーザと比較してどれだけ大きい計算コストをかけるか」という相対的な攻撃能力にあたる．

$k + k_{uc}[bit]$の秘密情報$p + p_{uc}$に対して，$T[bit]$の攻撃能力を有する任意の攻撃者Aのアドバンテージと計算量的安全性（計算機援用計算量的安全性）を以下のように定義する．

- 予測困難型：

$$Adv_{A[T]}(k + k_{uc}) = \frac{1}{2^{k-r}}$$

- 識別困難型：

$$Adv_{A[T]}(k + k_{uc}) = \frac{1}{2^{k-r}} - \frac{1}{2^{k+k_{uc}}}$$

定義 3.2 $k + k_{uc}[bit]$の秘密情報を攻撃する，攻撃能力が$T[bit]$で制限されるどのような攻撃者Aに対しても，アドバンテージが，$Adv_{A[T]}(k + k_{uc}) < \varepsilon(k + k_{uc})$である時，$T[bit]$計算機援用計算量的安全であると定義する．

計算機援用計算量的安全の定式化によると，正規ユーザ自身が$k_{uc}[bit]$分の計算コストを負担することによって，セキュリティシステムのセキュリティパラメータが CPU の計算機能力の進化に応じて増加する．そのため，将来計算機能力が向上して，攻撃者の攻撃能力が上昇しようとも，それに伴い，正規ユーザが上乗せできるエントロピも上昇するため，攻撃者の攻撃能力は相対的に一定（$r[bit]$）となる．すなわち，ある時点で，「攻撃者が正規ユーザと比較してどれだけ大きい計算コストをかけるか」という相対コストを基準にして，システムのセキュリティパラメータkを見積もっておけば，将来，計算機能力が向上して攻撃者の攻撃能力が上昇しようとも，それに応じて$k_{uc}[bit]$を増やしてやりさえすればシステムの安全性は保たれる．

新定式化において，$k_{uc} = 0$としたものが従来型の計算量的安全の定式化であることが分かる．すなわち，計算機援用計算量的安全の定式化は，従来の定式化を包含しており，従来の計算量的安全の定式化の自然な拡張となっている．

表 2　計算機援用計算量的安全性の定式化

セキュリティパラメータ	k, k_{uc}
秘密情報	$p + p_{uc}(k + k_{uc}[bit])$
正規ユーザの計算コスト	$Poly(k, k_{uc})\cdot 2^{k_{uc}}$
攻撃者の攻撃コスト	$Poly(k, k_{uc})\cdot 2^{k_{uc}}\cdot 2^r$
攻撃者のアドバンテージ	$Adv_{A[T]}(k + k_{uc})$

4　エントロピ増幅器

本章では，計算機援用計算量的安全性を満たすプリミティブの構成要素となるエントロピ増幅器のインスタンスについて説明する．

4.1 エントロピ増幅器のインスタンス

あるエントロピを持つ入力に対して，計算コストをかけて（入力よりも）高いエントロピを持つ出力を生成す

るシステムを「エントロピ増幅器」と定義する．具体的なインスタンスとしては，ブロックチェーン型エントロピ増幅器，定数値探索型エントロピ増幅器，最小（最大）値探索型エントロピ増幅器，パスワードベース鍵生成関数型エントロピ増幅器などが存在する．

4.2 ブロックチェーン型エントロピ増幅器

ブロックチェーン型エントロピ増幅器とは，「ハッシュ値の下c桁が全て 0 となるハッシュ関数の原像を，総当たり試行によって発見する」という計算コストをかけて，入力のエントロピを増幅させるシステムである．ビットコイン[6]のマイニング（採掘）におけるブロックチェーンが，その具体的なインスタンスである．具体的な手順を以下に示す．

手順：

1. p, cを入力とする．
2. pに$l[bit](l \geq c)$のp_1を連結させて，$p|p_1$としてハッシュ関数に入力し，H_1を得る．H_1の下c桁が全て 0 となったら$p|p_1$を出力して終了する．そうでなければ 3.へ進む．
3. pにp_1とは別の値である$l[bit](l \geq c)$のp_2を連結させて，$p|p_2$としてハッシュ関数に入力し，H_2を得る．H_2の下c桁が全て 0 となったら$p|p_2$を出力して終了する．そうでなければ 4.へ進む．
4. 同様に繰り返し，2^l回繰り返しても，下c桁が全て 0 となるものが見つからなかった場合は失敗となる．

ここで，pはパスワード，cはコスト，$p|p_i$はpとp_iの連結を表す．

ブロックチェーン型エントロピ増幅器においては，ハッシュ値の衝突（ハッシュ値の下c桁が全て 0 となる$p|p_i$は 1 つとは限らない）に配慮し，パスワードpに連結させるp_iに対しは，総当たりの順番を決めておく必要がある．また，ハッシュ値の下c桁が全て 0 となるp_iが見つかるまでに要する反復回数は一定ではなく（発見された時点で終了であり，期待値としては2^{c-1}回），かつ，2^l回の総当たり試行以内でハッシュ値の下c桁が全て 0 となる$p|p_i$が見つからない可能性もある．ここで，2^l回の総当たり試行を行なってもハッシュ値の下c桁が全て 0 となる$p|p_i$が見つからない確率は，理想的なハッシュ関数を仮定すると

$$\left(1-\frac{1}{2^c}\right)^{2^l} = \left(1-\frac{1}{2^c}\right)^{2^c \cdot 2^{l-c}} \approx \left(\frac{1}{e}\right)^{2^{l-c}}$$

となる．

4.3 定数値探索型エントロピ増幅器

定数値探索型エントロピ増幅器とは，「暗号関数の原像を，総当たり試行によって発見する」という計算コストをかけて，入力のエントロピを増幅させるシステムである．暗号関数としてハッシュ関数を用いた場合の具体的な手順を以下に示す．

手順：

1. pと$H(p|p_x)$を入力する．ここで，p_xのビット長はc$[bit]$である．
2. pに$l[bit]$のp_1を連結させて，$p|p_1$としてハッシュ関数に入力し，H_1を得る．H_1が$H(p|p_x)$と一致したら$p|p_1$を出力して終了する．そうでなければ 3.へ進む．
3. pにp_1とは別の値である$l[bit]$のp_2を連結させて，$p|p_2$としてハッシュ関数に入力し，H_2を得る．H_2が$H(p|p_x)$と一致したら$p|p_2$を出力して終了する．そうでなければ 4.へ進む．
4. $H(p|p_i)$と$H(p|p_x)$が一致するようなp_iが見つかった時点で，その$p|p_i$を出力する．

ここで，pはパスワード，$H(\cdot)$はハッシュ値，$p|p_i$はpとp_iのビット連結を表す．

ハッシュ関数を用いた定数値探索型エントロピ増幅器においては，ハッシュ値の衝突（$H(p|p_i)$と$H(p|p_x)$が一致する$p|p_i$は 1 つとは限らない）に配慮し，パスワードpに連結させるp_iに対しは，総当たりの順番を決めておく必要がある．また，$H(p|p_i)$と$H(p|p_x)$が一致する$p|p_i$はp_iの総当たりによって必ず見つかるが，発見されるまでに要する反復回数は一定ではない（発見された時点で終了であり，期待値としては2^{c-1}回）．

4.4 最小（最大）値探索演算型エントロピ増幅器

最小（最大）値検索型エントロピ増幅器とは，「暗号関数の最小値（最大値）生成する原像を，総当たり試行によって発見する」という計算コストをかけて，入力のエントロピを増幅させるシステムである．暗号関数としてハッシュ関数を用いた場合の具体的な手順を以下に示す．

手順：

1. p, cを入力とする．
2. pにc$[bit]$のp_1を連結させて$p|p_1$として，$p|p_1$の値をハッシュ関数に入力し，H_1を得る．
3. p_1とは別の値である$c[bit]$の値p_2をpに連結させて，$p|p_2$として，$p|p_2$の値をハッシュ関数に入力し，H_2を得る．
4. 同様に，すべてのp_iを用いて総当りを繰り返し，最終的に$H_1 \sim H_{2^c}$の中で，ハッシュ値が最小値（最大値）H_mとなる$p|p_m$を出力する．

ここで，pはパスワード，cはコスト，$p|p_i$はpとp_iの連結を表す．

ハッシュ関数を用いた最小（最大）値探索型エントロピ増幅器においては，ハッシュ値の衝突（H_mの原像となる$p|p_i$は 1 つとは限らない）に配慮し，パスワードpに連結させるp_iに対しは，総当たりの順番を決めておく必要がある．また，すべてのp_iを用いた総当りが尽くされるため，$p|p_m$を発見するまでに要する反復回数は2^c回で一定であり，H_mの原像となる$p|p_i$は必ず定まる．

4.5 パスワードベース鍵生成関数型エントロピ増幅器

パスワードベース鍵生成関数型エントロピ増幅器とは，「暗号関数の反復演算」という計算コストをかけて，入力のエントロピを増幅させるシステムである．パスワードベース鍵生成関数が，その具体的なインスタンスである．具体的な手順としては，2.2 節の bcrypt や PBKDF2 の手順を参照されたい．

4.6 エントロピ増幅器の安全性の定式化

ブロックチェーン型エントロピ増幅器，定数値探索型エントロピ増幅器，最小（最大）値探索型エントロピ増幅器，パスワードベース鍵生成関数型エントロピ増幅器の安全性を計算機援用計算量的安全性のパラメータによって記述すると次のようになる．

PBKDF2 以外のエントロピ増幅器の定式化：

$$k = n_s\,, k_{uc} = c$$

計算コスト：$Poly(n_s)\cdot 2^c$

PBKDF2 の定式化：

$$k = n_s\,, k_{uc} = log_2\, c$$

計算コスト：$Poly(n_s)\cdot 2^{log_2 c}$

ここで，n_sはパスワードpのビット長，2^cと$2^{log_2 c}$は反復演算回数（ブロックチェーン型および定数値探索型については反復演算回数の期待値）である．

エントロピ増幅器における攻撃者の目標は，ブロックチェーン型エントロピ増幅器，定数値探索型エントロピ増幅器，最小（最大）値探索型エントロピ増幅器に対しては正しい原像を見つけること，パスワードベース鍵生成関数型エントロピ増幅器に対しては，正しい出力の値を見つけることであるので，どちらも，そのアドバンテージは予測困難型の定式化を用いて表現するべきであろう[1]．

正規ユーザは，$k[bit]$の秘密情報（パスワードp）を知っているため，$2^{k_{uc}}$回の反復演算によって$k + k_{uc}[bit]$の出力が得られる．正規ユーザのこの計算コストがリーズナブルとなるように，適正な大きさのセキュリティパラメータk_{uc}を設定することが求められる．

$k + k_{uc}[bit]$のエントロピ増幅器の出力の値を攻撃する，攻撃能力が$T[bit]$で制限される任意の攻撃者 A のアドバンテージは，

$$Adv_{A[T]}(k + k_{uc}) = \frac{1}{2^{k-r}} \qquad (1)$$

と表される．ここで，$r = T - k_{uc}$である．$T[bit]$ の攻撃能力を有するどのような攻撃者 A に対しても，アドバンテージが，$Adv_{A[T]}(k + k_{uc}) < \varepsilon(k + k_{uc})$となるとき，そのエントロピ増幅器は，$T[bit]$計算機援用計算量的安全性を満たす．エントロピ増幅器においては，計算機援用計算量的安全性を満たす程度に大きなkの値を設定することが求められる．

[1] パスワードベース鍵生成関数の安全性について解析した文献[7]においても，その安全性を予測困難型の定式化で評価している．

5 計算機援用計算量的安全性を満たすプリミティブのインスタンス

4 章にて説明したエントロピ増幅器を用いて，計算機援用計算量的安全性を満たすプリミティブのインスタンスを構成可能である．プリミティブのインスタンスはユーザ認証のパスワードや共通鍵暗号の暗号鍵のように，秘密情報の総当たりが最も効率的な攻撃であるとモデル化できるタイプ（セキュリティシステムのエントロピ≒秘密情報のビット長）と，公開鍵暗号の秘密鍵のように，秘密情報の総当たりよりも効率的な攻撃が存在するとモデル化できるタイプ（セキュリティシステムのエントロピ＜秘密情報のビット長）に大別される．本稿では，前者を顕在型プリミティブ，後者を潜在型プリミティブと呼ぶ．

5.1 顕在型プリミティブ

5.1.1. インスタンス

一般的なユーザ認証および共通鍵においては，理想的なパスワードや共通鍵はランダムなビット列である．これに対し，エントロピ増幅器の出力をパスワードあるいは暗号鍵として用いることによって，それぞれエントロピ増幅器とユーザ認証が連結したプリミティブ，エントロピ増幅器と共通鍵暗号が連結したプリミティブを構成することができる．本稿では前者を計算機援用ユーザ認証（Computer Aided User Authentication, 以降 CAUA），後者を計算機援用共通鍵暗号（Computer Aided Symmetric Key Encryption，以降 CASKE）と呼ぶ．

5.1.2. 安全性

ユーザ認証や共通鍵暗号などの顕在型プリミティブに関しては，任意のエントロピ増幅器の出力をそのままパスワードまたは暗号鍵として用いることによって，CAUA および CASKE のインスタンスを構成できる．このため，その安全性は，エントロピ増幅器の安全性に依存する．すなわち，CAUA および CASKE の安全性は，式(1)のエントロピ増幅器の安全性と等価であり，正規ユーザが計算機能力を使用した分だけその安全性が向上することが確認できる．

5.2 潜在型プリミティブ

5.2.1. インスタンス

一般的な公開鍵暗号およびデジタル署名においては，秘密鍵はランダムなビット列である．これに対し，エントロピ増幅器の出力を公開鍵暗号またはデジタル署名の秘密鍵として用いることによって，それぞれ，エントロピ増幅器と公開鍵暗号が連結したプリミティブ，エント

ロピ増幅器とデジタル署名が連結したプリミティブを構成することができる．本稿では，前者を計算機援用公開鍵暗号（Computer Aided Public Key Encryption，以降CAPKE），後者を計算機援用デジタル署名（Computer Aided Digital Signature，以降CADS）と呼ぶ．

CAPKEは，エントロピ増幅器，理想的な決定性抽出器（乱数入力が不要な乱数抽出器）[9]，安全（計算量的に安全）な擬似乱数生成器[8]，選択平文攻撃に対する識別不可能性（IND-CPA）が証明されている公開鍵暗号アルゴリズムによって構成される．CADSは，エントロピ増幅器，理想的な決定性抽出器，安全（計算量的に安全）な擬似乱数生成器，選択文書攻撃に対する存在的偽造困難性（EUF-CMA）が証明されているデジタル署名アルゴリズムによって構成される．

5.2.2. 準備

公開鍵暗号：

公開鍵暗号は，4つのアルゴリズム(Gen1,Gen2,Enc,Dec)からなり，Gen1はセキュリティパラメータ1^{k_l}を入力とし，秘密鍵skを出力するアルゴリズム，Gen2は秘密鍵skを入力とし，公開鍵pkを出力するアルゴリズム，Encは公開鍵pkと平文mを入力とし，暗号文cを出力するアルゴリズム，Decは秘密鍵skと暗号文cを入力とし，平文mあるいは復号不可を表す特別な記号$\perp$を出力するアルゴリズムである．

デジタル署名：

デジタル署名は，4つのアルゴリズム(Gen1,Gen2,Sig,Ver)からなり，Gen1はセキュリティパラメータ1^{k_l}を入力とし，署名鍵$sigk$を出力するアルゴリズム，Gen2は署名鍵$sigk$を入力とし，検証鍵vkを出力するアルゴリズム，Sigは署名鍵$sigk$と平文mを入力とし，署名cを出力するアルゴリズム，Verは検証鍵vk，平文mおよび署名σを入力とし，1（検証成功）または0（検証失敗）を出力するアルゴリズムである．

擬似乱数生成器：

多項式時間で計算可能な確定的関数$PRG:\{0,1\}^k \to \{0,1\}^{l(k)}$は，任意の入力$s \in \{0,1\}^k$に対し，$l(k)$ビットの列を出力する．ここで，$l(k)$は$k$の多項式であり，$l(k) > k$が成立する．

定義 5.1 PRGに対する攻撃者Bのアドバンテージを

$$Adv_B^{PRG}(k) = |\Pr[B(r) \to 1 | r \leftarrow \{0,1\}^{l(k)}] - \Pr[B(r) \to 1 | r \leftarrow PRG(s)]|$$

と定義し，いかなる確率的多項式時間アルゴリズムBに対しても，$Adv_B^{PRG}(k) < \varepsilon(k)$が成立するとき，$PRG$は計算量的に安全な擬似乱数生成器であると言う．

乱数抽出器：

(k,ε)-抽出器とは，関数$f:\{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$のことで，少なくとも$k$の最小エントロピを持つ（離散）確率変数$X$に対して，$f(X,Y)$が$\{0,1\}^m$上の一様分布と$\varepsilon$-近傍になるものを言う．ただし，$Y$は$\{0,1\}^t$上の一様分布である．また，乱数入力が不要（$t=0$）である乱数抽出器を，決定性抽出器と呼ぶ．

IND-CPA安全：

IND-CPAゲームの挑戦者Bは，公開鍵と秘密鍵のペア$sk \leftarrow Gen1(1^{k_l})$，$pk \leftarrow Gen2(sk)$を生成し，攻撃者$A$に$pk$を入力する．その後，$A$は2つのメッセージ$m_0, m_1$を$B$に返す．$B$はチャレンジビット$b \in \{0,1\}$を0,1のうちどちらか1つ等確率で選び，$c^* \leftarrow Enc(pk, m_b)$を計算して，$A$にチャレンジ暗号文$c^*$を返す．最後に$A$はチャレンジビット$b$の推測値$b^* \in \{0,1\}$を出力する．

公開鍵暗号のIND-CPA安全は以下の定義5.2で定義される．

定義 5.2 公開鍵暗号アルゴリズムがIND-CPA安全であるとは，上記のIND-CPAゲームにおいて，どのような確率的多項式時間アルゴリズムAに対しても，

$$Adv^{IND-CPA}_A(k_l) = \left|\Pr[b = b^*] - \frac{1}{2}\right| \le \varepsilon(k_l)$$

となることをいう．

EUF-CMA安全：

EUF-CMAゲームの挑戦者Bは，検証鍵と署名鍵のペア$sigk \leftarrow Gen1(1^{k_l})$，$vk \leftarrow Gen2(sigk)$を生成し，攻撃者$A$に$vk$を入力する．その後，$A$は自分で選んだメッセージ$m_i$に対する署名$\sigma_i$を返してくれる署名オラクルを適応的に，$T$回利用する．攻撃者$A$はメッセージと署名対（$m^*, \sigma^*$）を$B$に返す．$B$は，$(m^*, \sigma^*) \notin \{m_i\}_{1 \le i \le T}$かつ$Ver(vk, m, \sigma) = 1$であれば，1を出力する．そうでなければ0を出力する．

デジタル署名のEUF-CMA安全は以下の定義5.3で定義される．

定義 5.3 デジタル署名アルゴリズムがEUF-CMA安全であるとは，上記のEUF-CMAゲームにおいて，Bが1を出力することを，$Exp^{EUF-CMA}_A(k_l) \to 1$と記述した時，どのような確率的多項式時間アルゴリズムAに対しても，

$$Adv^{EUF-CMA}_A(k_l) = \Pr[Exp^{EUF-CMA}_A(k_l) \to 1] \le \varepsilon(k_l)$$

となることをいう．

5.2.3. 疑似乱数を秘密鍵として用いる公開暗号アルゴリズムの安全性

IND-CPA安全：

疑似乱数（計算量的に安全な擬似乱数生成器の出力）を「IND-CPA 安全を満たす公開鍵暗号アルゴリズムの秘密鍵」として用いることによって構成した公開鍵暗号アルゴリズム（以下，「疑似乱数型公開鍵暗号」と呼ぶ）も，IND-CPA安全を満たすことを証明する．

定理 5.1 疑似乱数型公開鍵暗号アルゴリズムは，どのような確率的多項式時間アルゴリズム$\tilde{A}$に対しても，

$$Adv^{IND-CPA}_{\Sigma 1,\tilde{A}}(k) = \left|\Pr[b = b^* | sk \leftarrow PRG(k)] - \frac{1}{2}\right| < \varepsilon(k)$$

を満たす．ここで，$\Sigma 1$は，疑似乱数型公開鍵暗号（「k $[bit]$の真性乱数をシードとして生成した疑似乱数」を秘密鍵とした公開鍵暗号）アルゴリズムであることを，$sk \leftarrow PRG(k)$は，擬似乱数生成器にk $[bit]$のエントロピ

を持つシードを入力することによって生成された疑似乱数を秘密鍵skとして用いることを示している.

定理 5.1 の証明

疑似乱数と真性乱数を識別することを目的とする攻撃者Bを以下のように設定する.

・Bに対する次のような挑戦者 C を設定し，Bと C の間のゲーム G を構成する.

C は，$a \in \{0,1\}$を0,1のうちどちらか 1 つを等確率で選び，$a = 0$の時は，skを$\{0,1\}^{k_l}$から一様に選ばれた乱数とし，Bに渡す．$a = 1$の時は，skを$PRG(k)$から生成される疑似乱数として，Bに渡す．Bが 1/2 よりも高い確率でaの値を正答できれば（疑似乱数と真性乱数を識別できれば），Bの勝ち.

・公開鍵暗号アルゴリズムの IND-CPA 攻撃者$\tilde{A}$を仮定する．Bは，C とのゲームを行うにあたり，$\tilde{A}$を利用することができる．$\tilde{A}$を使役しているBを$B^{\tilde{A}}$と記す.

このとき，ゲーム G に対する$B^{\tilde{A}}$の戦略は以下の通りとなる.

1. $B^{\tilde{A}}$は，C からskを受け取る.
2. $B^{\tilde{A}}$は，skを公開鍵暗号の秘密鍵とし，アルゴリズム Gen2 を用いて，skに対する公開鍵pkをskから生成する.
3. $B^{\tilde{A}}$は，$\tilde{A}$にpkを渡す.
4. IND-CPA 攻撃者$\tilde{A}$は，$B^{\tilde{A}}$にm_0, m_1を返す.
5. $B^{\tilde{A}}$は，コイントスによってチャレンジビット$b \in \{0,1\}$の値（0 または 1）を選び，$c^* \leftarrow Enc(pk, m_b)$を計算して，$\tilde{A}$にチャレンジ暗号文$c^*$を返す.
6. $\tilde{A}$は, チャレンジビットbの推測値$b^* \in \{0,1\}$を$B^{\tilde{A}}$に返す.
7. $B^{\tilde{A}}$は，$b \neq b^*$の時，0を C に回答し，$b = b^*$の時，1を C に回答する.

手順 6 において，$b = b^*$であった場合，$B^{\tilde{A}}$は「$\tilde{A}$が公開鍵暗号に対する IND-CPA 攻撃に成功した」ことが分かる．IND-CPA 安全な公開鍵暗号アルゴリズムを用いているので真性乱数であるskの時，$\tilde{A}$は IND-CPA 攻撃に成功しない. $\tilde{A}$がIND-CPA 攻撃に成功したということは，手順 3 にて$B^{\tilde{A}}$から$\tilde{A}$に渡されたpkが，擬似乱数であるskから生成されたものであるということが演繹できる.

一方，安全な疑似乱数の定義より，以下の補題 5.1 が成り立つ.

補題 5.1 挑戦者 C が計算量的に安全な擬似乱数生成器を用いれば，どのような確率的多項式時間アルゴリズム$\tilde{A}$に対しても,

$$\Pr[b = b^* | sk \leftarrow PRG(k)] < \Pr[b = b^* | sk \leftarrow \{0,1\}^{k_l}] + \varepsilon(k)$$

が成り立つ．ここで，$sk \leftarrow \{0,1\}^{k_l}$は，$\{0,1\}^{k_l}$から一様ランダムに選ばれた真性乱数を$sk$として用いることを示している.

補題 5.1 の証明

定義 5.1 より，いかなる確率的多項式時間アルゴリズムBに対しても,

$$Adv_B^{PRG}(k) = |\Pr[B(sk) \to 1 | sk \leftarrow \{0,1\}^{k_l}] - \Pr[B(sk) \to 1 | sk \leftarrow PRG(k)]| < \varepsilon(k)$$

が成り立つ. ここで, 確率的多項式時間アルゴリズムBは, 入力が真性乱数であるか擬似乱数であるかを識別するアルゴリズムであり，$B(sk) \to 1$は，Bが「入力値skを疑似乱数であると判定した」ことを示している.

上式は任意のBに対して成立するため, $\tilde{A}$を使役している$B^{\tilde{A}}$に対しても上式が成り立つ．ゲーム G に対する$B^{\tilde{A}}$の戦略から明らかなように, 手順 7 において$B^{\tilde{A}}$が 1 を出力するのは,「手順 6 にて$\tilde{A}$が返してくるb^*」が$b = b^*$となる場合であるので，上式は式(2)のように変形できる.

$$\max_{\tilde{A}} Adv_{B^{\tilde{A}}}^{PRG}(k) = \max_{\tilde{A}} |\Pr[b = b^* | sk \leftarrow \{0,1\}^{k_l}] - \Pr[b = b^* | sk \leftarrow PRG(k)]| < \varepsilon(k) \quad (2)$$

式(2)は，いかなる確率的多項式時間アルゴリズム$\tilde{A}$に対しても「$\Pr[b = b^* | sk \leftarrow \{0,1\}^{k_l}]$と$\Pr[b = b^* | sk \leftarrow PRG(k)]$の差は$\varepsilon(k)$未満である」ことを意味している．よって，$\Pr[b = b^* | sk \leftarrow \{0,1\}^{k_l}] + \varepsilon(k)$は必ず，$\Pr[b = b^* | sk \leftarrow PRG(k)]$よりも大きいことが言え，補題 5.1 が証明された.

補題 5.1 を用いることによって，定理 5.1 を証明することができる. 補題 5.1 の式を変形していくことにより,

$$\max_{\tilde{A}} \left\{ \left| \Pr[b = b^* | sk \leftarrow PRG(k)] - \frac{1}{2} \right| \right\} < \max_{\tilde{A}} \left\{ \left| \Pr[b = b^* | sk \leftarrow \{0,1\}^{k_l}] + \varepsilon(k) - \frac{1}{2} \right| \right\},$$

$$\max_{\tilde{A}} \left\{ \left| \Pr[b = b^* | sk \leftarrow PRG(k)] - \frac{1}{2} \right| \right\} < \max_{\tilde{A}} \left\{ \left| \Pr[b = b^* | sk \leftarrow \{0,1\}^{k_l}] - \frac{1}{2} \right| \right\} + \varepsilon(k) \quad (3)$$

が得られる.

ここで, 式(3)の左辺が疑似乱数型公開鍵暗号（「k $[bit]$の真性乱数をシードとして生成した疑似乱数」を秘密鍵とした公開鍵暗号）アルゴリズムに対する IND-CPA 攻撃者$\tilde{A}$のアドバンテージを表す式そのものであり，式(3)の右辺第 1 項が通常の公開鍵暗号（k_l $[bit]$の真性乱数を秘密鍵とした公開鍵暗号）アルゴリズムに対する IND-CPA 攻撃者$\tilde{A}$のアドバンテージの式そのものであることに気付くと,

$$Adv_{\Sigma 1,\tilde{A}}^{IND-CPA}(k) < Adv_{\Sigma 0,\tilde{A}}^{IND-CPA}(k_l) + \varepsilon(k) \quad (4)$$

であることが分かる．ここで，Σ0は，通常の公開鍵暗号（k_l $[bit]$の真性乱数を秘密鍵とした公開鍵暗号）アルゴ

リズムであることを示している.

IND-CPA 安全が証明されている公開鍵暗号アルゴリズムに対しては，その公開鍵暗号アルゴリズムを IND-CPA 攻撃するどのような確率的多項式時間アルゴリズム$\tilde{A}$に対しても，定義 5.2 の不等式が成立する．疑似乱数を「IND-CPA 安全を満たす公開鍵暗号アルゴリズムの秘密鍵」として用いることによって構成されている疑似乱数型公開鍵暗号アルゴリズムにおいても，これは成立し，以下が成り立つ.

$$Adv^{IND-CPA}_{\Sigma 0,\tilde{A}}(k_l) = \left|\Pr[b = b^* | sk \leftarrow \{0,1\}^{k_l}] - \frac{1}{2}\right|$$

$$\leq \varepsilon(k_l)$$

よって，式(4)は以下のようになる.

$$Adv^{IND-CPA}_{\Sigma 1,\tilde{A}}(k) < Adv^{IND-CPA}_{\Sigma 0,\tilde{A}}(k_l)$$

$$\leq \varepsilon(k_l) + \varepsilon(k) < \varepsilon(k) + \varepsilon(k) = \varepsilon(k)$$

以上より，定理 5.1 が証明された.

CADS の安全性証明：

紙面の都合で割愛する.

5.2.4. CAPKE および CADS の構成

CAPKE の手順：

1. エントロピ増幅器に，k' $[bit]$のパスワードpを入力する．一般的にパスワードは完全な乱数ではないため，その真のエントロピをk $[bit]$ $(k < k')$ とする.
2. エントロピ増幅器より$k' + k_{uc}$ $[bit]$の$p + p_{uc}$が出力される．ここで，$p|p_{uc}$の真のエントロピは $k + k_{uc}$ $[bit]$である.
3. $p|p_{uc}$を決定性抽出器に入力する.
4. 決定性抽出器より$k + k_{uc}$ $[bit]$の$p' + p'_{uc}$が出力される．ここで，$p' + p'_{uc}$の真のエントロピは $k + k_{uc}$ $[bit]$である.
5. $p' + p'_{uc}$を擬似乱数生成器のシードとして入力する.
6. 擬似乱数生成器よりk_l $[bit]$の擬似乱数が出力される．ここで，k_lは，公開鍵暗号アルゴリズムのセキュリティパラメータである.
7. k_lを秘密鍵として，公開鍵暗号の 3 つのアルゴリズム Gen2，Enc，Dec を使用する.

CADS の手順：

手順 1～6 は，CAPKE の手順と同じである.

7. k_lを署名鍵として，デジタル署名の 3 つのアルゴリズム Gen2，Sig，Ver を使用する.

5.2.5. 計算機援用計算量的安全性の証明

CAPKE の安全性が識別困難型の計算機援用計算量的安全性を満たすインスタンスであること，および，CADS が予測困難型の計算機援用計算量的安全性を満たすインスタンスであることを証明する.

CAPKE の手順 4～6 から，エントロピ増幅器によって生成される CAPKE の秘密鍵，$k + k_{uc}$ $[bit]$のエントロピを有するシードから生成される疑似乱数となっていることが分かる．このため，定理 5.1 が成り立ち，それ故に次の定理 5.2 が成立する.

定理 5.2 任意のエントロピ増幅器，理想的な決定性抽出器，計算量的に安全な擬似乱数生成器，IND-CPA 安全が証明されている公開鍵暗号アルゴリズムを用いて構成した CAPKE はどのような確率的多項式時間アルゴリズム$\tilde{A}$に対しても，

$$Adv^{IND-CPA}_{\Sigma 1,\tilde{A}}(k + k_{uc}) < \varepsilon(k + k_{uc})$$

が成り立つ．すなわち，識別困難型の$T[bit]$計算機援用計算量的安全性（正規ユーザによる CPU の計算機能力の利用に起因するエントロピがセキュリティパラメータに上乗せされる）を満たす.

CADS の証明については，紙面の都合で割愛する.

6 まとめ

本稿では，従来の計算量的安全性の定式化を拡張し，正規ユーザによる CPU の計算機能力の利用に起因するエントロピに対応するセキュリティパラメータを導入した．そして，相対コスト要因に関する秘密情報のみを「保持すべき秘密情報」として切り出して扱える計算機援用計算量的安全性の定式化を行った．また，計算機援用計算量的安全性は，従来の計算量的安全性の定式化を包含していることを示した.

参考文献

[1] 兼子拓弥，本部栄成，高橋健太，西垣正勝，“計算機援用ユーザ認証，” 情報処理学会論文誌，Vol.55，No.9，pp.2072-2080，2014.

[2]Niels Provos and David Mazieres, “A Future-Adaptable Password Scheme,” USENEX Annual Technical Conference，1999.

[3]Meltem Sonmez Turan,Elaine Barker,William Burr, and Lily Chen, “NIST Special Publication 800-132 Recommendation for Password-Based Key Derivation,” 2010.

[4]森山大輔，西巻陵，岡本龍明，「公開鍵暗号の数理」，共立出版株式会社，2011.

[5]高橋健太，松田隆宏，村上隆夫，“暗号プリミティブにおける秘密情報の分布と安全性の関係，” 2015 年暗号と情報セキュリティシンポジウム予稿集，2015.

[6]Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2009.

[7]Frances F. Yao and Yiqun Lisa Yin, “Design and Analysis of Password-Based Key Derivation Functions,” CT-RSA 2005, Vol.3376, pp245-261, 2005.

[8]Andrew C. Yao, “Theory and Applications of Trapdoor Functions(Extended Abstract),” FOCS，pp80-91，1982.

[9]小柴健史，“乱数抽出の基礎，” 第 2 回横幹連合コンファレンス， 2007.