計算機能力のエイジングと不正者のエフォートによる安全性への影響を考慮した計算量的安全性の定式化

兼子 拓弥† 本部 栄成† 高橋 健太‡ 西垣 正勝†

†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
nisigaki@inf.shizuoka.ac.jp

‡(株)日立製作所 横浜研究所
244-0817 神奈川県横浜市戸塚区吉田町 292
kenta.takahashi.bw@hitachi.com

あらまし 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，CPUの計算機能力の時間的変化（エイジング要因）と不正者の計算コスト（エフォート要因）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須となる．しかし，CPUの進化は不正者だけでなく正規ユーザにも恩恵を与えるものであるため、エフォート要因こそが本質的であるという考え方が成り立つであろう．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，エイジング/エフォートの各要因に対応する安全性を切り分けて扱える枠組みを提案する．具体的には各要因に対応する2つのセキュリティパラメータを導入した定式化を行うとともに，秘密情報のエントロピがエフォート要因のみに依存する仕組みを定式化する．この結果，ある時点で「不正者が負担する計算コストがどれくらい大きければ安全性が担保されるか」を見積もった上で秘密情報のエントロピを決定してやれば，将来計算機能力が向上しても秘密情報のエントロピを一定に保ったまま安全性を維持することが可能となる．

Formularization of Computational Security with consideration of Aging of CPU Performance and Effort of Attackers

Takuya Kaneko† Eisei Honbu† Kenta Takahashi‡ Masakatsu Nishigaki†

†Graduate school of Informatics, Shizuoka University
3-5-1 Johoku, Naka, Hamamatsu, Shizuoka 432-8011, JAPAN
nisigaki@inf.shizuoka.ac.jp
‡Yokohama Research Laboratory, Hitachi, Ltd.,
292 Yoshida, Totsuka, Yokohama, Kanagawa 244-0817, JAPAN
kenta.takahashi.bw@hitachi.com

Abstract In the security system which is based on computational security, it is necessary to ensure the entropy of secret information in consideration of both the improvement of CPU performance (aging factor) and the computational cost of attacker (effort factor).

However, because the evolution of CPU performance benefits not only attackers but also legitimate users, it is expected that the effort factor takes more important role in security system. So, in this paper, we extend the conventional formularization of computational security so that we can consider the security relating to the aging factor and that relating to the effort factor independently. More specifically, we formularize computational security with two security parameters for both factors, and then propose a new formulation in which the entropy of secret information is determined only by the effort factor. As a result, once the required entropy of secret information is assessed based on "how much computational cost is needed to protect the security system at some time point", the security of the system will be ensured without increase in the entropy of secret information even as CPU performance increases.

1 はじめに

CPU の計算機能力は日々向上する. 計算機を利用した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 CPU の進化に伴って増加させる必要がある. また, 不正者は, 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. 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, CPU の計算機能力の時間的変化(エイジング要因)と不正者の計算コスト(エフォート要因)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須となる.

しかし, CPU の進化は不正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, 不正者のみが有利になるものではない. この視点に立てば, 時間的に変化することのない安全性の決定要因として, エフォート要因こそが本質的に重要であるという考え方が成り立つであろう. 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, エイジング/エフォートの各要因に対応する安全性を切り分けて扱える枠組みを提案する.

現在の計算量的安全性の定式化では, エイジング要因とエフォート要因の 2 つを 1 つのセキュリティパラメータだけで評価している. これに対し, 本稿では, 各要因に対応する 2 つのセキュリティパラメータを導入するとともに, 秘密情報のエントロピがエフォート要因のみに依存し, エイジング要因に依存しない仕組みを定式化する. 具体的には, 不正者と正規ユーザの両者に関与するエイジング要因については, その時代の計算機能力に応じた計算コストの負担を不正者にも正規ユーザにも一律に要求することによって必要なエントロピを補填する方式に変更する.

この方式であれば, 正規ユーザも計算コストを支払ってエイジング要因に関するエントロピを確保する形となるため, 正規ユーザがこれを秘密情報として覚える必要がなくなる. この結果, 正規ユーザが所持すべき秘密情報のエントロピはエフォート要因のみに依存することになり, ある時点で「不正者が負担する計算コストがどれくらい大きければ安全性が担保されるか」を見積もった上で秘密情報のエントロピを決定してやれば, 将来計算機能力が向上しても秘密情報のエントロピを一定に保ったまま安全性を維持することが可能となる. 秘密情報の情報源が限られている場合(パスワード, 生体情報, PUF など)や, 秘密情報の記録容量が限られる場合(軽量チップなど)など, CPU の進化に従って秘密情報のエントロピを増加させることが困難であるアプリケーションにおいては, 本定式化が特に有効であると期待される.

2 従来の計算量的安全性の定式化における課題

現在，認証や暗号等様々なセキュリティシステムにおいて，正規ユーザ以外が不正にシステムを利用することを防ぐため，計算量的安全性に基づいたセキュリティ設計がなされている．セキュリティシステムが「計算量的に安全である」とは，現状の計算機能力を考慮した上で，不正者による暗号解読やなりすまし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[5]．暗号解読やなりすましに要する時間（期待値）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．

このセキュリティパラメータを設定するに当たっては，「時代と共に進化していくCPUの計算機能力（エイジング要因）」と「不正者が攻撃のためにかける時間的・金銭的コスト（エフォート要因）」を考慮する必要がある．つまり，「不正者が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）をかけて攻撃しても安全」かつ，「将来，ある程度まで計算機能力が向上することによって不正者の攻撃能力が向上しても安全」となるようにセキュリティパラメータを設定する．

ここで，従来の計算量的安全性の定式化では，エイジング要因とエフォート要因を 1 つのセキュリティパラメータ（秘密情報）で表現している．すなわち，計算機能力の向上に対して安全性を保つためにセキュリティパラメータを大きくすることは，秘密情報自体が大きくなることに直結する．このため，パスワード認証，生体認証，PUFなど秘密情報の情報源が限られる場合や，軽量チップ実装など秘密情報の記録容量が限られる場合など，CPU の進化に従って秘密情報のエントロピを増加させることが困難であるアプリケーションにおいては，秘密情報のエントロピを十分に確保することができず，将来的に安全性を維持できなくなることが考えられる．

しかし，CPU の進化は不正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，不正者のみが有利になるものではない．すなわち，エイジング要因に関するセキュリティパラメータは，その時代の計算機能力に応じた計算コストの負担を不正者にも正規ユーザにも一律に要求するものであると考えることができよう．一方，エフォート要因に関するセキュリティパラメータは，不正者のみに一定量以上の計算コストの負担を要求するものである．この視点に立てば，時間的に変化することのない安全性の決定要因として，エフォート要因こそが本質的に重要であるという考え方が成り立つであろう．

事実，「その時代の計算機能力に応じた計算コストの負担を不正者にも正規ユーザにも一律に要求する」というコンセプトに基づいて設計されたセキュリティシステムは既に存在しており，今までに bcrypt [2]や PBKDF2 [3]，計算機援用ユーザ認証[4]などが提案されている．しかし，エイジング要因とエフォート要因を 1 つのセキュリティパラメータ（秘密情報）で表現する従来の計算量的安全性の定式化では，これら「エイジング要因に関する計算コストを正規ユーザに負担させるセキュリティシステム」の安全性を適切に表現することができない[1]．

3 新しい計算量的安全性の定式化

3.1 計算量的安全性の相対的な定式化

本稿では，従来の計算量的安全性の定式化を拡張し，エイジング/エフォートの各要因に対応する安全性を切り分けて扱える枠組みを提案する．従来の計算量的安全性の定式化と，本稿で

[1] このため、たとえば，bcrypt や PBKDF2 では，「不正者の攻撃能力に応じて適切な回数だけ計算を繰り返す」というような曖昧な表現で説明がなされている．

提案する新しい計算量的安全性の定式化をまとめたものを表 1 に示す.

従来の定式化(表 1 の「従来型」)では, セキュリティパラメータはkのみであり, 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. 正規ユーザと不正者の計算能力はいずれも, セキュリティパラメータkを引数とする任意の多項式時間アルゴリズムPoly(k)によりモデル化される. ただし, 正規ユーザは秘密情報pを既知であり, 不正者は未知である. 不正者がいかなる多項式時間アルゴリズムによる攻撃を行ったとしても, 成功確率Adv(k)が無視できる大きさである(ε(k)よりも小さい)場合に, 計算量的安全性が保証される. より正確には, Adv(k)は「当て推量で攻撃した場合の成功確率に対して, Poly(k)で努力する不正者の成功確率がどれだけ高まるか」の差分(アドバンテージ)である. 当然, 秘密情報pのエントロピ(セキュリティパラメータk)が大きくなるにつれて, Adv(k)は小さくなる(攻撃困難になる)ので, Adv(k)がkに対して無視できる(Adv(k) < ε(k))ように, kの値を十分に大きくすることが必要となる.

従来の定式化(表 1 の「従来型」)に対し, セキュリティパラメータを 2 つに分離した定式化が表 1 の「パラメータ分離型」である. ここで, エイジング要因に対応したセキュリティパラメータをk_r(および, k_rに関する秘密情報をp_r), エフォート要因に対応したセキュリティパラメータをk_u(および, k_uに関する秘密情報をp_u)としている. 単にkがk_rとk_uに分かれただけ($k = k_u|k_r$)であり, 「従来型」と「パラメータ分離型」の定式化の意味は本質的には同じである.

パラメータ分離型の定式化に対して, エイジング要因に関するセキュリティパラメータk_rについては「その時代の計算機能力に応じた計算コストの負担を不正者にも正規ユーザにも一律に要求する」という仕組みを導入することによって, 計算量的安全性の定式化は表 1 の「相対型」の形となる. エイジング要因(セキュリティパラメータk_r)に関する秘密情報p_rについては, 正規ユーザもセキュリティシステムを使用する際にその都度相応の計算コストを払ってその「正しい値」を発見するという形式となるため, 正規ユーザはp_rを覚える必要はない. すなわち, エフォート要因(セキュリティパラメータk_u)に関する秘密情報p_uのみが「正規ユーザが所持すべき秘密情報」となる. これに伴い, 安全性の定式化もAdv(k) < ε(k)からAdv(k_u) < ε(k_u)という形に変わる.

セキュリティシステムの使用の度に正規ユーザおよび不正者が新たに負担することになるp_rの探索は, 正規ユーザおよび不正者の計算能力に重畳されることになるため, 両者の計算能力がPoly(k_u, k_r)からPoly(k_u, k_r) · SPoly(k_r)に変わる. ここで, エイジング要因に関するセキュリティパラメータk_rは計算機能力の進化にともなって増加させる必要があるため, それに応じてp_r探索の計算コストも増加することになる. Moore の法則[1]に従えば計算機能力の進化は指数関数的であることに鑑み, 今回の定式化ではp_r探索の計算コストをSPoly(k_r)(セキュリティパラメータk_rに対する任意の多項式f(k_r)より漸近的に大きな関数)時間アルゴリズムによって定式化している.

表 1 計算量的安全性の定式化のまとめ

	従来型	パラメータ分離型	相対型
セキュリティパラメータ	k	k_u, k_r	k_u, k_r
秘密情報	p (k[bit])	p_u, p_r (それぞれk_u, k_r[bit])	p_u (k_u[bit])
正規ユーザの計算能力	Poly(k)	Poly(k_u, k_r)	Poly(k_u, k_r) · SPoly(k_r)
不正者の計算能力	Poly(k)	Poly(k_u, k_r)	Poly(k_u, k_r) · SPoly(k_r)
攻撃成功確率	Adv(k) < ε(k)	Adv(k_u, k_r) < ε(k_u, k_r)	Adv(k_u) < ε(k_u)

3.2　セキュリティパラメータの決定

本定式化によって，セキュリティシステムの安全性は，エフォート要因に関する秘密情報p_uのみによって担保される形となる．エイジング要因については，その時代の計算機能力に応じた大きさを持つ秘密情報p_rの探索によって必要なエントロピが補填される．このため，ある時点で「不正者が負担する計算コストがどれくらい大きければ安全性が担保されるか」を見積もった上で，2つのセキュリティパラメータk_uおよびk_rの大きさを決定してやれば，将来計算機能力が向上した場合も，エイジング要因に関するセキュリティパラメータk_rの大きさだけを適切に設定し直すことで，エフォート要因に関するセキュリティパラメータk_uの大きさには変更を加えないまま[2]，セキュリティシステムの計算量的安全性を維持することが可能である．このような観点から，今回の定式化を「相対型」計算量的安全性と呼ぶこととする．

相対型の定式化においては，正規ユーザは「計算機能力（エイジング要因）により補填できる情報量のうえに，さらに秘密情報（エフォート要因）の分を加えた情報量」を利用できるため，計算機能力（エイジング要因）のみを利用することしかできない不正者に対して常に優位を保つことができる．ここで，セキュリティシステムの利便性の観点から，正規ユーザの計算コスト（p_rの探索）は十分小さくなくてはならず，一方，セキュリティシステムの安全性の観点からは，不正者の計算コスト（p_uとp_rの探索）は十分大きくなくてはならない．この条件を鑑みてセキュリティパラメータk_uおよびk_rの大きさを決定することが肝要となる．

4　新定式化の具体例

4.1　従来の計算量的安全性

3 章では，従来の計算量的安全性におけるセキュリティパラメータkをk_rとk_uに分離する（$k = k_u|k_r$）ことによって相対型の定式化を導いたが，「相対型の定式化において、$k_u = k$，$k_r = 0$としたものが従来型の定式化である」という捉え方も可能である．すなわち，相対型の定式化は，従来の計算量的安全性を包含した定式化となっていることが分かる．

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のイメージを図 1 に示す．k[bit]の秘密情報に対し，不正者がA[bit]分の情報を分析する能力を有している場合，不正者の攻撃成功確率（不正者がk[bit]の秘密情報を言い当てる確率）は$1/2^{k-A}$である．一方で，不正者がk[bit]の秘密情報を完全に当て推量で言い当てる確率は$1/2^k$である．したがって，$\mathrm{Adv}(k) = 1/2^{k-A} - 1/2^k$であり，この値が negligible となる程度に十分大きなセキュリティパラメータkを設定することが求められる．

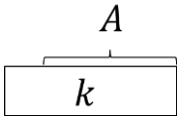

図 1　従来の計算量的安全性における セキュリティパラメータ

4.2　パスワードベース鍵生成関数

bcrypt [2]および PBKDF2 [3]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あたっての反復演算回数を

[2] ただし，たとえば，ある時点でのハッシュ値$\mathrm{H}(p_u|p_r)$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，そのハッシュ値の原像を求める攻撃はいずれ必ず成功し，p_uおよびp_rが不正者に漏れることになる．このため，秘密情報p_uは（エントロピの大きさはそのままで良いが）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．

可変とすることによって暗号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，不正者の攻撃試行の速度を減速させることができ，これによって入力（パスワード）のエントロピの不足を補っている．

bcrypt の計算手順を要約すると，

1. P, S, 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.の結果を鍵として出力する

となる．ここで，Pはパスワード，Sはソルト，cはコストである．bcrypt におけるパスワードPのビット長をn_pとして相対型の定式化に当てはめると，

$k_u = n_p$, $k_r = c$,
計算能力$= \mathrm{Poly}(n_p) \cdot 2^c$,
攻撃成功確率$\mathrm{Adv}(n_p) < \varepsilon(n_p)$

となる．

一方，PBKDF2 の計算手順を要約すると，

1. P, S, 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 2.と同じハッシュ関数に入力し，U_2を得る
4. P, U_2を 2.や 3.と同じ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～U_cの排他的論理和を求め，これを鍵として出力する

となる．ここで，Pはパスワード，Sはソルト，cは反復回数，U_iは第iラウンドのハッシュ値である．PBKDF2 におけるパスワードPのビット長をn_p，反復回数cについては$n_c = \log c$として相対型の定式化に当てはめると，

$k_u = n_p$, $k_r = n_c$,
計算能力$= \mathrm{Poly}(n_p) \cdot 2^{n_c}$,
攻撃成功確率$\mathrm{Adv}(n_p) < \varepsilon(n_p)$

となる．

bcrypt および PBKDF2 のセキュリティパラメータのイメージを図 2 に示す．

正規ユーザはk_u[bit]の秘密情報（パスワードP）を知っているため，2^{k_r}回の反復演算を 1 回行えば正しい暗号鍵を生成できる．正規ユーザのこの計算コストがリーズナブルになるように，適正な大きさのセキュリティパラメータk_rを設定することが求められる．

k_u[bit]の秘密情報（パスワードP）に対し，不正者が（2^{k_r}回の反復演算を繰り返しながら）A[bit]分の情報を分析する能力を有している場合，不正者の攻撃成功確率（不正者がk_u[bit]の秘密情報を言い当てる確率）は$1/2^{k_u - A}$である．一方で，不正者がk_u[bit]の秘密情報を完全に当て推量で言い当てる確率は$1/2^{k_u}$である．したがって，$\mathrm{Adv}(k) = 1/2^{k_u - A} - 1/2^{k_u}$であり，この値が negligible となる程度に十分大きなセキュリティパラメータk_uを設定することが求められる．

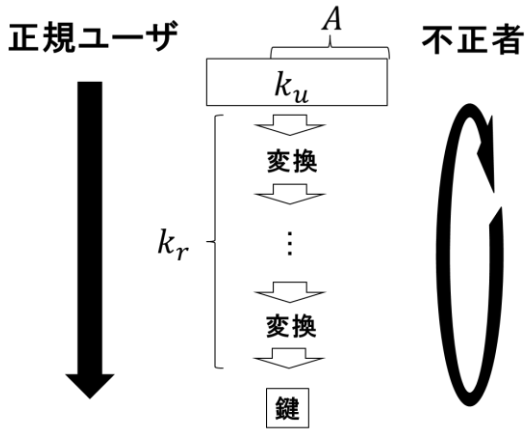

図 2 bcrypt および PBKDF2 のセキュリティパラメータ

4.3 計算機援用ユーザ認証

計算機援用ユーザ認証[4]は，「不正者と正規ユーザの両者に一定の計算コストを負担させる」というコンセプトに基づくユーザ認証方式である．認証情報の一部が正規ユーザにも未知となっていて，その情報を総当たり試行によって求めるという方法が採られている．

計算機援用ユーザ認証の認証手順を要約すると，

1. 登録フェーズにて，$\mathrm{H}(\mathrm{H}(P))$が認証サーバに登録されている

2. サーバはクライアントにH(H(P))を送信する
3. ユーザはクライアント端末にP_uを入力する
4. 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を総当たり試行によって探索する
5. クライアントはH($P_u|P_r$)をサーバに送信する
6. 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

となる. ここで, P_uはユーザが所持すべき認証情報, P_rは総当たり試行によって求める認証情報、$P = P_u|P_r$、H(·)はハッシュ関数である. 計算機援用ユーザ認証における認証情報P_u, P_rのビット長をそれぞれn_u, n_rとして相対型の定式化に当てはめると,

$k_u = n_u$, $k_r = n_r$,
計算機能力= $\mathrm{Poly}(n_u + n_r) \cdot 2^{n_r}$,
攻撃成功確率$\mathrm{Adv}(n_u) < \varepsilon(n_u)$

となる.

計算機援用ユーザ認証のセキュリティパラメータのイメージを図 3 に示す.

正規ユーザはk_u[bit]の秘密情報(認証情報P_u)を知っているため, P_rの探索を 1 回行えば認証に成功する. 正規ユーザのこの計算コストがリーズナブルになるように, 適正な大きさのセキュリティパラメータk_rを設定することが求められる. 例えば文献[4]では, 正規ユーザの計算コストが 1 秒以内となるように配慮されている.

k_u[bit]の秘密情報(パスワードP_u)に対し, 不正者が(P_rの探索を繰り返しながら)A[bit]分の情報を分析する能力を有している場合, 不正者の攻撃成功確率(不正者がk_u[bit]の秘密情報を言い当てる確率)は$1/2^{k_u-A}$である. 一方で, 不正者がk_u[bit]の秘密情報を完全に当て推量で言い当てる確率は$1/2^{k_u}$である. したがって, $\mathrm{Adv}(k) = 1/2^{k_u-A} - 1/2^{k_u}$であり, この値が negligible となる程度に十分大きなセキュリティパラメータk_uを設定することが求められる. 例えば文献[4]では, 不正者の計算コストが 1 年以上となるように配慮されている.

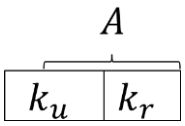

図 3 計算機援用ユーザ認証のセキュリティパラメータ

5 まとめと今後の課題

暗号や認証における計算量的安全性の定式化においては, 従来, 1 つのセキュリティパラメータを用いてセキュリティシステムの安全性を議論していた. これに対し, 本稿では, セキュリティパラメータを CPU の計算機能力の進化に関するパラメータ(エイジング要因)と不正者が攻撃に費やす計算コストに関するパラメータ(エフォート要因)に切り分けて扱う枠組みを提案した. 更に, エイジング要因については, その時代の計算機能力に応じた計算コストの負担を不正者にも正規ユーザにも一律に要求する方式に変更することによって, 計算量的安全性の定式化を相対型へと拡張した.

相対型の定式化によって既存のパスワードベース鍵生成関数アルゴリズムや計算機援用ユーザ認証システムの計算量的安全性を記述することができること, および、相対型の定式化が従来の計算量的安全性の定式化を包含することを示した. 相対型の定式化によれば, 将来計算機能力が向上しても, ユーザが所持すべき秘密情報の大きさを変更することなく, 安全性を保ちながらセキュリティシステムを運用することが可能である.

今後は, 更に実用的なセキュリティシステムに対して, 本定式化に基づいたより詳細な検討を行っていく. 特に, セキュリティパラメータの設定に関する具体的な方法を確立していきたい.

参考文献

[1] Moore, G: Cramming More Components onto Integrated Circuits, Proceedings of

the IEEE, VOL.86, NO.1, JANUARY (1998)
[2] Provos, N. and Mazieres, D.: A Future-Adaptable Password Scheme, USENIX Annual Technical Conference (1999)
[3] Turan M., Barker, E., Burr, W. and Chen, L.: NIST Special Publication 800-132 Recommendation for Password-Based Key Derivation
[4] 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, (2014.9) (採録決定)
[5] 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