情報処理学会論文誌 Vol.59 No.9 1557–1569 (Sep. 2018)

計算機援用セキュリティスキームの一般的構成法

神農 泰圭[1,a)] 土屋 貴史[1] 大木 哲史[1] 高橋 健太[2] 尾形 わかは[3] 西垣 正勝[1]

受付日 2017年12月11日, 採録日 2018年6月8日

概要：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ーを増加させる必要がある．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．実際に，CPU の計算機能力を使用して正規ユーザの秘密情報の安全性を高めるセキュリティスキームが提案されている．本稿では，そのような「CPU の計算機能力を使用して正規ユーザの秘密情報の安全性を高めるセキュリティスキーム」を「計算機援用セキュリティスキーム CASS」と呼び，それらをメカニズムの違いにより 2 つのタイプに分類する．そして，一方のタイプに対して，一般的構成法を提案し，その安全性証明を行う．そのタイプの CASS の実現可能性について議論する．

キーワード：暗号，計算機援用セキュリティ，一般的構成法，共通鍵暗号，メッセージ認証コード，公開鍵暗号，デジタル署名，ユーザ認証

General Constructions of Computer-aided Security Schemes

Yasuyoshi Jinno[1,a)] Takashi Tsuchiya[1] Tetsushi Ohki[1] Kenta Takahashi[2]
Wakaha Ogata[3] Masakatsu Nishigaki[1]

Received: December 11, 2017, Accepted: June 8, 2018

Abstract: CPU performance has been increasing continually. To resist attacks in such a technological environment, we must increase the entropy of secrets. However, the progress of CPU performance benefits not only attackers but also legitimate users. Security schemes specifically designed to enhance legitimate users' security by exploiting CPU performance have been proposed. In this paper, we call such schemes "computer-aided security schemes", or CASS, and classify them into two types in terms of the aiding mechanism. We propose general constructions and provide security proofs of CASS for one type. We discuss the feasibility of CASS.

Keywords: cryptology, computer-aided security, general construction, symmetric key encryption, message authentication code, public key encryption, digital signature, user authentication

1. はじめに

情報セキュリティ技術として共通鍵暗号，メッセージ認証コード，公開鍵暗号，デジタル署名，ユーザ認証等の暗号プリミティブが広く使用されている．これらの多くは計算量的安全性を持つように設計されており，秘密鍵やパスワードといった秘密情報のエントロピーを増加させると，攻撃に必要な計算量が爆発的に大きくなることを安全性の根拠としている．一方で，ムーアの法則で知られるように CPU の計算能力もまた時代とともに指数関数的に向上を続けており，このため秘密情報のエントロピーもまた時代とともに増加させていく必要がある．しかし，以下に示すように，秘密情報のエントロピーを増加させることが困難であるセキュリティスキームが存在する．ユーザがパスワードを記憶して，そのパスワードを秘密情報として扱う「パスワードベーススキーム」においては，ユーザの記憶能力に限界があるため，パスワードのエントロピーを増加させることが困難である．同じ機能を持つ回路でも，製造過

[1] 静岡大学
Shizuoka University, Hamamatsu, Shizuoka 432–8011, Japan
[2] 株式会社日立製作所
Hitachi Ltd., Yokohama, Kanagawa 244–0817, Japan
[3] 東京工業大学
Tokyo Institute of Technology, Meguro, Tokyo 152–8552, Japan
[a)] yasu05182002@gmail.com

程における微細な差異を秘密情報のエントロピーとして利用する PUF [1] の出力を秘密情報として用いて暗号スキームを構築することができる．また，生体情報を鍵として用いて，暗号スキームを構築しようとする試み（バイオメトリック暗号）が存在する [2]．このように，PUF の出力や，生体情報を秘密情報として用いる場合でも，情報源に限りがあるため，秘密情報のエントロピーを増加させることが困難である．

これに対し，「正規ユーザの計算機能力を用いて秘密情報のセキュリティを強化する」というアプローチの研究もされている．これらの研究では，CPU の計算能力の進化は攻撃者のみを有利にするものではなく，正規ユーザにも恩恵を与えることに注目し，システムの安全性を高めるために正規ユーザがその時代の計算能力を活用する方法を提案している．著者らは，この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ぶ．計算機援用セキュリティスキーム CASS として，パスワードベース鍵生成関数 bcrypt [3]，PBKDF2 [4] や，計算機援用ユーザ認証 [7] が提案されている．

パスワードベース鍵生成関数 f は，ハッシュ関数や鍵スケジューリング関数の反復演算によって，入力された短い秘密情報 s_{short} から，長い秘密情報 s_{long} を生成する．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させることができ，これによって入力値のエントロピー不足を補っている．本稿では，$f(s_{short})$ を秘密情報として使うスキームを秘密情報拡張型 CASS と呼ぶ．f における反復演算回数を L 回とする．文献 [5], [6] では f に関する安全性が解析されており，文献 [6] では，$f(s_{short})$ を同じ長さの一様乱数と識別するためには，s_{short} を同じ長さの一様乱数と識別する場合に比べて，L 倍の計算量が必要なことが証明されている．この結果を用いれば，秘密情報拡張型 CASS の一般的な安全性を解析することが可能であり，$\log_2 L$ ビット分の安全性が上乗せされることが証明できる．しかし，文献 [6] では，証明においてランダムオラクルモデルを仮定しているため，このような CASS を現実のシステムとして使用することを考えたときには問題が発生する可能性がある．

この問題に対し著者らは文献 [7] で，短い秘密情報を拡張するのではなく，十分なエントロピーを持つ情報の一部のみを記憶すればよい CASS を提案した．このユーザ認証スキームでは，s_{short} に，ランダムビット列 r がパディングされる．すなわち，$s_{short}\|r = s_{long}$ が秘密情報として登録され，ユーザは s_{short} のみを秘密に記憶，所持する．ここで，$s_{short}\|r$ は s_{short} と r のビット列の連結を表す．認証フェーズでは，サーバは r のヒント（$s_{short}\|r$ の二重ハッシュ値 $h(h(s_{short}\|r))$）をユーザに送り，ユーザは s_{short} とヒントを用いた総当たり探索により，r を復元し，認証される．本稿では，このようなコンセプトに基づく CASS を，秘密情報補完型 CASS と呼ぶ．

文献 [7] では，秘密情報補完型 CASS の具体的なインスタンスとして，ユーザ認証について詳述した．しかし，そのユーザ認証スキームの安全性証明はなされていない．また，ユーザ認証以外の暗号システムへの適用については，検討されていない．

そこで本稿では，ユーザ認証を含む様々なプリミティブに対しての秘密情報補完型 CASS の一般的構成法を提案し，その安全性証明を行う．具体的には，以下のとおりである．

初めに，文献 [7] のユーザ認証スキームを一般化した秘密情報補完型 CASS のモデルを定義する．

次に，様々なプリミティブに対して，秘密情報補完型 CASS の一般的構成法を提案する．より正確には，任意の共通鍵暗号，メッセージ認証コード，公開鍵暗号，デジタル署名，チャレンジ&レスポンス（C&R）型ユーザ認証スキームを，秘密情報補完型 CASS へ変形する構成方法を提案する．秘密情報拡張型の CASS の安全性は，ランダムオラクルモデルと，用いるプリミティブの安全性をもととしているが，秘密情報補完型 CASS の安全性は，興味深いことに，用いるプリミティブの安全性のみをもととしている．

最後に，秘密情報補完型 CASS の実現可能性について議論する．

2. 計算機援用セキュリティスキーム：CASS

暗号スキームにおいては，正規ユーザは秘密鍵や秘密情報 s_{long} を記憶，もしくは所持する．典型的には，そのスキームのセキュリティパラメータ k は s_{long} のビット長である．多くは $k = 128$ である [8]．

一方で，CASS においては，正規ユーザが「秘密に」記憶，もしくは所持する情報としては，k_u（$< k$）ビットの s_{short} のみでよい．すなわち，CASS とは，s_{short} から s_{long} を生成する，もしくは復元するという計算機援用メカニズムを用いて，s_{short} のセキュリティを強化することが可能なセキュリティスキームである．

秘密情報補完型 CASS においては，s_{long} は $s_{long} = s_{short}\|r$ という 2 つの部分からなり，ユーザは s_{short} のみを秘密に記憶，もしくは所持する．s_{short} と r は独立であるため，秘密情報拡張型とは異なり，s_{short} のみから s_{long} を求めることはできない．そのため，ユーザは追加の情報として，s_{short} から r を復元するためのヒントが必要となる．そのヒントは公開情報であり，攻撃者も知ることができるため，設計には気をつけなければならない．秘密情報補完型 CASS の典型的な例は，文献 [7] のユーザ認証スキームである．

r はシステムが生成した一様乱数である．一方，s_{short} は，ユーザが選んだビット列とすることもできるし，システムが生成した一様乱数とすることもできる．

3. CASSの一般的構成法

本章では，共通鍵暗号，メッセージ認証コード，公開鍵暗号，デジタル署名，C&R型ユーザ認証の各プリミティブに対する秘密情報補完型CASSの一般的な構成法を説明し，その安全性証明を行う．

3.1 計算機援用共通鍵暗号

3.1.1 モデル

通常の共通鍵暗号 Π_{SKE} は3つのアルゴリズム (Gen, Enc, Dec) からなる．鍵生成アルゴリズム Gen は，1^k を入力とし，一様ランダムに k ビット列 s_{long} を生成し，それを秘密鍵として出力する．暗号化アルゴリズム Enc は s_{long} と平文 $m \in \{0,1\}^{l(k)}$ を入力とし，暗号文 c を出力する．ここで，$l(k)$ は k に関する多項式であり，$\{0,1\}^{l(k)}$ は $l(k)$ ビット長のビット列の集合を表す．Enc は確率的アルゴリズムとなりうる．復号アルゴリズム Dec は s_{long} と c を入力とし，m を出力する．Dec は決定的アルゴリズムである．すべての k，$s_{long} \leftarrow Gen(1^k)$，$m$ に対して，$Dec(s_{long}, Enc(s_{long}, m)) = m$ であること（正当性）が要求される．

Π_{SKE} の安全性モデルは，様々な定義が存在する．その中で最も強い安全性は選択平文暗号文攻撃に対する識別不可能性（IND-P2-C2）[9] である．IND-P2-C2の攻撃ゲームでは，初めに，$Gen(1^k)$ により，s_{long} が生成される．攻撃者は，暗号化オラクル，復号オラクル，チャレンジオラクルに対して，適応的にクエリを送ることができる．攻撃者が m_i，もしくは c_j を，それぞれ暗号化オラクル，復号オラクルに送ると，そのオラクルは $c_i = Enc(s_{long}, m_i)$，$m_j = Dec(s_{long}, c_j)$ を返す．攻撃者が，(m_0^*, m_1^*) をチャレンジオラクルに送ると，そのオラクルはランダムにチャレンジビット b を選び，$c_b^* = Enc(s_{long}, m_b^*)$ を計算し，それを返す．最後に，攻撃者は推測ビット b' を出力する．$b = b'$ であれば攻撃者の勝ち．攻撃者のアドバンテージは $|\Pr[b = b'] - 1/2|$ で定義される．

計算機援用共通鍵暗号 $\widehat{\Pi}_{SKE}$ も，Π_{SKE} と同様に，3つのアルゴリズム $(\widehat{Gen}, \widehat{Enc}, \widehat{Dec})$ からなる．鍵&ヒント生成アルゴリズム $\widehat{Gen}$ は，1^k と $str \in \{0,1\}^{k_u}$ を入力とし，$s_{short} \in \{0,1\}^{k_u}$ とヒント $hint$ を出力する．$\widehat{Gen}$ は，str をそのまま s_{short} として使うことや，その長さ k_u のみを使うこともできる．前者は，ユーザ自身が s_{short} を選ぶことができることを意味し，後者は，s_{short} がランダム k_u ビット列となることを意味する．他のプリミティブのCASSにおける鍵（もしくは，秘密情報）&ヒント生成アルゴリズムにおいても str を入力とするが，その使用方法は上記と同様である．暗号化アルゴリズム $\widehat{Enc}$ は，秘密鍵 s_{long} の代わりに，s_{short}，平文 m，$hint$ を入力とし，暗号文 $c = \widehat{Enc}(s_{short}, m, hint)$ を出力する．復号アルゴリズム $\widehat{Dec}$ は s_{short}，c，$hint$ を入力とし，$m = \widehat{Dec}(s_{short}, c, hint)$ を出力する．すべての k，str，$(s_{short}, hint) \leftarrow \widehat{Gen}(1^k, str)$，$m$ に対して，$\widehat{Dec}(s_{short}, \widehat{Enc}(s_{short}, m, hint), hint) = m$ が成り立つ確率は1に限りなく近いこと（正当性）が要求される．

$\widehat{\Pi}_{SKE}$ に対する攻撃ゲームは，Π_{SKE} に対する攻撃ゲームとほぼ同様に定義できる．異なる部分は，オラクルが (Gen, Enc, Dec) ではなく $(\widehat{Gen}, \widehat{Enc}, \widehat{Dec})$ を実行し，$\widehat{Gen}$ において s_{short} と $hint$ が生成され，$hint$ を攻撃者が知ることができるという点である．たとえば，IND-P2-C2の攻撃ゲームにおいては，初めに，$\widehat{Gen}(1^k, str)$ により $(s_{short}, hint)$ が生成される．攻撃者は，$hint$ を与えられて，適応的に暗号化，復号，チャレンジオラクルにクエリを送る．攻撃者が m_i，もしくは c_j を，それぞれ暗号化オラクル，復号オラクルに送ると，そのオラクルは $c_i = \widehat{Enc}(s_{short}, m_i, hint)$，$m_j = \widehat{Dec}(s_{short}, c_j, hint)$ を返す．攻撃者が，(m_0^*, m_1^*) をチャレンジオラクルに送ると，そのオラクルはランダムにチャレンジビット b を選び，$c_b^* = \widehat{Enc}(s_{short}, m_b^*, hint)$ を計算し，それを返す．最後に，攻撃者は推測ビット b' を出力する．$b = b'$ であれば攻撃者の勝ち．

3.1.2 一般的構成法

$\Pi_{SKE} = (Gen, Enc, Dec)$ を用いて，$\widehat{\Pi}_{SKE} = (\widehat{Gen}, \widehat{Enc}, \widehat{Dec})$ を次のように構成できる．我々の構成では，すべてのプリミティブにおいて，s_{short} は $\widehat{Gen}$ において一様ランダムに生成される．

鍵&ヒント生成 $\widehat{Gen}(1^k, str)$：1^k を入力とし，秘密鍵 $s_{long} = Gen(1^k)$ を生成する．次に，一様ランダムに N 個の平文 $m_0, \ldots, m_{N-1}$ を選び，それに対応する暗号文 $c_0 = Enc(s_{long}, m_0), \ldots, c_{N-1} = Enc(s_{long}, m_{N-1})$ を生成する．N の決め方については後述する．$(s_{short}, hint)$ を出力する．ここで，$s_{long} = s_{short} \| r$，$|s_{short}| = |str| = k_u$，$hint = ((m_0, c_0), \ldots, (m_{N-1}, c_{N-1}))$ である．

暗号化 $\widehat{Enc}(s_{short}, m, hint)$：初めに，$Dec((s_{short} \| r), c_0) = m_0, \ldots, Dec((s_{short} \| r), c_{N-1}) = m_{N-1}$ をすべて満たす r を総当たりで探索し，$s_{long} = s_{short} \| r$ を復元する．次に，s_{long} を用いて m を暗号化し，暗号文 $c = Enc(s_{long}, m)$ を出力する．

復号 $\widehat{Dec}(s_{short}, c, hint)$：$\widehat{Enc}$ と同様の方法で，s_{short} と $hint$ を用いて s_{long} を復元する．次に，c を復号し，平文 $m = Dec(s_{long}, c)$ を出力する．

N の値は，次のように決定する．ランダムに選ばれた $s_{long} = (s_{short} \| r), m_0, \ldots, m_{N-1}$ に対して，

$$Dec((s_{short} \| r'), Enc((s_{short} \| r), m_0)) = m_0$$
$$\vdots$$
$$Dec((s_{short} \| r'), Enc((s_{short} \| r), m_{N-1})) = m_{N-1}$$

が成り立つ r' $(\neq r)$ が存在する確率を $\delta(N)$ とする．$\widehat{\Pi}_{SKE}$ においては，確率 $(1 - \delta(N))$ で，ユーザはヒントから正

しい r を復元できる．そこで，$\delta(N)$ が無視できるほど小さくなる N を選ぶ．

このように N を選択することによって，圧倒的確率 $(1-\delta(N))$ でヒントから正しい r が復元できるため，正当性が成り立つ．

具体的な N の値については方式 Π_{SKE} によるが，Π_{SKE} が理想的なランダム性を持つと仮定すると，$\delta = 2^{-(Nl(k)-k+k_u)}$ が十分小さくなるように N を選べばよい．このとき，$r' \neq r$ ならば任意の m に対して $\Pr[Dec((s_{short}\|r'), Enc((s_{short}\|r), m)) = m] = 1/2^{l(k)}$ が成り立つので，$\delta(N)$ は，

$$
\begin{aligned}
\delta(N) &= \Pr[\exists r'(\neq r)\forall i \in \{0,\ldots,N-1\} \\
&\qquad : Dec((s_{short}\|r'), Enc((s_{short}\|r), m_i)) \\
&\qquad = m_i] \\
&= 1 - \Pr[\forall r'(\neq r)\exists i \in \{0,\ldots,N-1\} \\
&\qquad : Dec((s_{short}\|r'), Enc((s_{short}\|r), m_i)) \\
&\qquad \neq m_i] \\
&= 1 - \left(1 - \left(\frac{1}{2^{l(k)}}\right)^N\right)^{2^{k-k_u}-1} \\
&\approx 1 - \left(1 - \frac{2^{k-k_u}}{2^{Nl(k)}}\right) \quad \left(\frac{2^{k-k_u}}{2^{Nl(k)}} \ll 1 \text{ より}\right) \\
&= 2^{-Nl(k)+k-k_u} = \delta
\end{aligned}
$$

となる．たとえば，$l(k) = k = 128$，$k_u = 102$ であるとき，$\delta < 2^{-128}$ としたいならば $N = 2$ とすればよい．

実際に利用される Π_{SKE} は理想的なランダム性を持たないため，さらに大きな N を使う必要がある．なお，平文空間が非常に小さい場合には，$\delta(N)$ を無視できるほど小さくすることが困難であるため，我々の構成によって計算機援用共通鍵暗号は構成できない．

3.1.3 安全性証明

我々の構成では，s_{short} は $\widehat{Gen}$ において一様ランダムに生成されるため，s_{long} は一様乱数となる．それにより，ランダムオラクル仮定を必要とせず，次のことが証明できる．他のプリミティブに対しても同様である．

上記の構成法を用いて Π_{SKE} から構成された $\widehat{\Pi}_{SKE}$ は，Π_{SKE} とほとんど同じ安全性を持つことを証明する．ここでは，もし Π_{SKE} が最も強い安全性を持っていたら，$\widehat{\Pi}_{SKE}$ も最も強い安全性を持つことを証明する．

Theorem 1. Π_{SKE} が IND-P2-C2 安全ならば，$\widehat{\Pi}_{SKE}$ もまた IND-P2-C2 安全である．より正確には，$\widehat{\Pi}_{SKE}$ に対して，暗号化，復号クエリを合計 q 回行い，無視できないアドバンテージで IND-P2-C2 ゲームに勝利する攻撃者 A が存在するならば，Π_{SKE} に対して，暗号化，復号クエリを合計 $q+N$ 回行い，無視できないアドバンテージで IND-P2-C2 ゲームに勝利する攻撃者 B が存在する．

Proof. B が，暗号化，復号，チャレンジオラクルと，$\widehat{\Pi}_{SKE}$ に対する IND-P2-C2 ゲームに無視できないアドバンテージ ε で勝利する A を使って（A の勝利確率は $1/2 \pm \varepsilon$），Π_{SKE} に対する IND-P2-C2 ゲームに無視できないアドバンテージで勝利しようとしている状況を考える．

ここで，Π_{SKE} に対する IND-P2-C2 ゲームと，$\widehat{\Pi}_{SKE}$ に対する IND-P2-C2 ゲームは，攻撃の際に *hint* を入手できるか（$\widehat{\Pi}_{SKE}$ に対する IND-P2-C2 ゲーム）否か（Π_{SKE} に対する IND-P2-C2 ゲーム）が異なるのみである．そのため，B はオラクルとのやりとりを通して，A に入力するための *hint* を生成し，A に入力する．それ以降は，B は A からのクエリをそのままオラクルに送り，オラクルからの返答をそのまま A に返してやるだけでよい．B は，A に入力するための *hint* を生成するために，クエリ回数が A よりも N 回多くなる．

具体的には，次のような手順となる．初めに，$Gen(1^k)$ より，秘密鍵 s_{long} が生成される．B は，A に *hint* である平文と暗号文の N ペアを入力するために，一様ランダムに平文 $m_0,\ldots,m_{N-1}$ を生成し，$m_0,\ldots,m_{N-1}$ を暗号化オラクルへと送る．暗号化オラクルは $c_0 = Enc(s_{long}, m_0),\ldots,c_{N-1} = Enc(s_{long}, m_{N-1})$ を B に返す．B は $((m_0,c_0),\ldots,(m_{N-1},c_{N-1}))$ を *hint* として A に入力する．この時点で B は A を起動する．A は，B に m_i もしくは c_j を適応的に送ってくる．B は，m_i が送られてきたら，暗号化オラクルに m_i を送り，$c_i = Enc(s_{long}, m_i)$ を得て，A に c_i を返す．同様にして，B は，c_j が送られてきたら，$m_j = Dec(s_{long}, c_j)$ を得て，A に返す．A が，(m_0^*, m_1^*) を B に送ってきたら，B はチャレンジオラクルに (m_0^*, m_1^*) を送る．チャレンジオラクルはランダムにチャレンジビット b を選び，$c_b^* = Enc(s_{long}, m_b^*)$ を計算し，B にそれを返す．B は，A に c_b^* を返す．最後に，A は推測ビット b' を出力するので，B は b' を出力する．

$\widehat{\Pi}_{SKE}$ では確率 $(1-\delta(N))$ で正しい r が復元され，この確率で，上記の A の環境は，$\widehat{\Pi}_{SKE}$ に対する IND-P2-C2 ゲームと同じとなり，A は確率 $1/2 \pm \varepsilon$ で $b' = b$ となる b' を出力する．一方，確率 $\delta(N)$ で，上記の A の環境は，$\widehat{\Pi}_{SKE}$ に対する IND-P2-C2 ゲームとは異なる環境となり，A が $b' = b$ となる b' を出力する確率は不明である．したがって，B が Π_{SKE} の IND-P2-C2 ゲームに勝利する確率は，

$$
\begin{aligned}
&\Pr[r \text{ を復元可能}]\Pr[b' = b \mid r \text{ を復元可能}] \\
&+ \Pr[r \text{ を復元不可能}]\Pr[b' = b \mid r \text{ を復元不可能}] \\
&= (1-\delta(N))\Pr[b = b' \mid r \text{ を復元可能}] \\
&\qquad + \delta(N)\Pr[b' = b \mid r \text{ を復元不可能}]
\end{aligned}
$$

となる．A が確率 $1/2+\varepsilon$（$0 < \varepsilon < 1/2$）で $b' = b$ となる b' を出力する場合は，

$$\begin{aligned}&\Pr[B\text{ が }\Pi_{SKE}\text{ の IND-P2-C2 ゲームに勝利する}]\\&\quad\geq(1-\delta(N))\Pr[b=b'\mid r\text{ を復元可能}]\\&\quad\quad+\delta(N)\times 0=(1-\delta(N))(1/2+\varepsilon)\\&=1/2+\varepsilon-\delta(N)(1/2+\varepsilon)\\&\geq 1/2+\varepsilon-\delta(N)\quad(1/2+\varepsilon\leq 1\text{ より})\end{aligned}$$

となり，B のアドバンテージは $\varepsilon-\delta(N)$ 以上となる．一方，A が確率 $1/2-\varepsilon$（$0<\varepsilon<1/2$）で $b'=b$ となる b' を出力する場合は，同様にして，B のアドバンテージは $\varepsilon-\delta(N)$ 以上となる．すなわち，B のアドバンテージは $\varepsilon-\delta(N)$ となる．

Q.E.D.

同様にして，Π_{SKE} が「適応的選択平文，もしくは，非適応的選択平文攻撃，既知平文攻撃」に対する「識別不可能性，もしくは，一方向性，鍵回復攻撃耐性」を持つならば，$\widehat{\Pi}_{SKE}$ も同じ安全性を持つことが証明できる．

3.2 計算機援用メッセージ認証コード

3.2.1 モデル

通常のメッセージ認証コード Π_{Mac} は 3 つのアルゴリズム (Gen, Mac, Ver) からなる．鍵生成アルゴリズム Gen は，1^k を入力とし，一様ランダムに k ビット列 s_{long} を生成し，それを秘密鍵として出力する．認証子生成アルゴリズム Mac は s_{long} とメッセージ $m\in\{0,1\}^*$ を入力とし，認証子 $t\in\{0,1\}^l$ を出力する．ここで，$\{0,1\}^*$ は有限長のビット列の集合を表す．Mac は確率的アルゴリズムとなりうる．検証アルゴリズム Ver は s_{long} と m と t を入力とし，検証結果 b を出力し，$b=1$ なら正当な認証子として受理し，$b=0$ なら不正な認証子として拒否する．Ver は決定的アルゴリズムである[*1]．すべての k，$s_{long}\leftarrow Gen(1^k)$，$m$ に対して，$Ver(s_{long}, m, Mac(s_{long}, m))=1$ であること（正当性）が要求される．

Π_{Mac} の安全性モデルは，様々な定義が存在する．その中で最も強い安全性は適応的選択メッセージ攻撃に対するメッセージ認証子の存在的偽造不可能性（EUF-CMA）である．EUF-CMA の攻撃ゲームでは，初めに，$Gen(1^k)$ より，s_{long} が生成される．攻撃者は，認証子生成オラクルに対して，適応的にクエリを送ることができる．攻撃者が m_i を認証子生成オラクルに送ったら，そのオラクルは，$t_i=Mac(s_{long}, m_i)$ を返す．最後に攻撃者は，クエリしたメッセージ以外のメッセージ m^* と認証子 t^* を出力する．$Ver(s_{long}, m^*, t^*)=1$ であれば攻撃者の勝ち．

計算機援用メッセージ認証コード $\widehat{\Pi}_{Mac}$ も，Π_{Mac} と同様に，3 つのアルゴリズム $(\widehat{Gen}, \widehat{Mac}, \widehat{Ver})$ からなる．鍵&ヒント生成アルゴリズム $\widehat{Gen}$ は，1^k と $str\in\{0,1\}^{k_u}$ を入力とし，$s_{short}\in\{0,1\}^{k_u}$ と $hint$ を出力する．認証子生成アルゴリズム $\widehat{Mac}$ は，秘密鍵 s_{long} の代わりに，s_{short}，メッセージ m，$hint$ を入力とし，認証子 $t=\widehat{Mac}(s_{short}, m, hint)$ を出力する．検証アルゴリズム $\widehat{Ver}$ は s_{short}，m，t，$hint$ を入力とし，検証結果 b を出力し，$b=1$ なら正当な認証子として受理し，$b=0$ なら不正な認証子として拒否する．すべての k，str，$(s_{short}, hint)\leftarrow\widehat{Gen}(1^k, str)$，$m$ に対して，$\widehat{Ver}(s_{short}, m, \widehat{Mac}(s_{short}, m, hint), hint)=1$ が成り立つ確率は 1 に限りなく近いこと（正当性）が要求される．

$\widehat{\Pi}_{Mac}$ に対する攻撃ゲームは，Π_{Mac} に対する攻撃ゲームとほぼ同様に定義できる．異なる部分は，$\widehat{\Pi}_{SKE}$ に対する攻撃ゲームと同様に，オラクルが (Gen, Mac, Ver) ではなく，$(\widehat{Gen}, \widehat{Mac}, \widehat{Ver})$ を実行し，攻撃者が $hint$ を得ることができるという点である．

3.2.2 一般的構成法

$\Pi_{MAC}=(Gen, Mac, Ver)$ を用いて，$\widehat{\Pi}_{MAC}=(\widehat{Gen}, \widehat{Mac}, \widehat{Ver})$ を次のように構成できる．

鍵&ヒント生成 $\widehat{Gen}(1^k, str)$：1^k を入力とし，秘密鍵 $s_{long}=Gen(1^k)$ を生成する．次に，一様ランダムに N 個のメッセージ $m_0,\ldots,m_{N-1}$ を選び，それに対応する認証子 $t_0=Mac(s_{long}, m_0),\ldots,t_{N-1}=Mac(s_{long}, m_{N-1})$ を生成する．$(s_{short}, hint)$ を出力する．ここで，$s_{long}=s_{short}\|r$，$|s_{short}|=|str|=k_u$，$hint=((m_0,t_0),\ldots,(m_{N-1},t_{N-1}))$ である．

認証子生成 $\widehat{Mac}(s_{short}, m, hint)$：初めに，$Ver((s_{short}\|r), m_0, t_0)=1,\ldots,Ver((s_{short}\|r), m_{N-1}, t_{N-1})=1$ をすべて満たす r を総当たりで探索し，$s_{long}=s_{short}\|r$ を復元する．次に，s_{long} を用いて m から認証子 $t=Mac(s_{long}, m)$ を生成し，出力する．

検証 $\widehat{Ver}(s_{short}, m, t, hint)$：$\widehat{Mac}$ と同様の方法で，s_{short} と $hint$ を用いて s_{long} を復元する．次に，s_{long} を用いて m と t に対して検証を行い，検証結果 $b=Ver(s_{long}, m, t)$ を出力する．

N の決定については，共通鍵暗号の場合と同様に，$Ver((s_{short}\|r'), m_i, t_i)=1$ がすべての $i\in\{0,\ldots,N-1\}$ について成り立つような r'（$\neq r$）が存在する確率 $\delta(N)$ が無視できるほど小さくなるように選ぶ．このとき，r は圧倒的確率で正しく復元されるため，正当性が成り立つ．

具体的な N の値は Π_{Mac} に依存する．一般に，MAC には CBC-MAC 等のブロック暗号を利用した方式と，HMAC 等のハッシュ関数を利用した方式がある．一般的に Ver では，Mac で認証子を再計算して，受け取った認証子と等しいかを確認することに注意されたい．

Π_{MAC} がブロック暗号を使用したアルゴリズムである場合，そのブロック暗号が理想的なランダム性を持つと仮定すると，

$$\Pr[Mac((s_{short}\|r'), m_i)=Mac(s_{long}, m_i)]=1/2^l$$

[*1] 上述したとおり，Mac は確率的になりうるが，一般的には決定的であるため，Ver では，Mac で認証子を再計算して，受け取った認証子と等しいかを確認する．

が成り立つため，共通鍵暗号の場合と同様に $\delta(N) = 2^{-l\cdot N+(k-k_u)}$ が求まる．すなわち，$l\cdot N$ が $k-k_u$ より十分大きくなるように，N の値を設定すればよい．たとえば，$l=256$，$k=128$ の CBC-MAC で，$k_u=102$ である場合，$\delta<2^{-k}$ としたいならば，$N=1$ に設定すればよい．実際に利用されるブロック暗号は理想的なランダム性を持たないため，さらに大きな N を使う必要がある．なお，メッセージ空間が非常に小さい場合には，$\delta(N)$ を無視できるほど小さくすることが困難であるため，我々の構成によって計算機援用メッセージ認証コードは構成できない．

一方，Π_{MAC} がハッシュ関数を使用したアルゴリズムである場合，$N=1$ で十分である．なぜならば，$\delta(1)$ が有意に高ければ，それは用いているハッシュ関数の衝突困難性を破っていることになる．すなわち，用いているハッシュ関数が衝突困難性を有していれば，$\delta(1)$ は無視できるほど小さい値となる．

3.2.3 安全性証明

上記の構成法を用いて Π_{MAC} から構成された $\widehat{\Pi}_{MAC}$ は，Π_{MAC} とほとんど同じ安全性を持つことを証明する．ここでは，もし Π_{MAC} が最も強い安全性を持っていたら，$\widehat{\Pi}_{MAC}$ も最も強い安全性を持つことを証明する．

Theorem 2. Π_{MAC} が EUF-CMA 安全ならば，$\widehat{\Pi}_{MAC}$ もまた EUF-CMA 安全である．より正確には，$\widehat{\Pi}_{MAC}$ に対して，認証子生成クエリを合計 q 回行い，無視できない確率で EUF-CMA ゲームに勝利する攻撃者 A が存在するならば，Π_{MAC} に対して，認証子生成クエリを合計 $q+N$ 回行い，無視できない確率で EUF-CMA ゲームに勝利する攻撃者 B が存在する．

Proof. Theorem 1 の証明とほぼ同様である．異なる部分は，Theorem 1 の証明における IND-P2-C2 が，EUF-CMA に置き換わる点である．ここで，A は，$\widehat{\Pi}_{MAC}$ に対する EUF-CMA ゲームに無視できない確率 ε で勝利する．

確率 $1-\delta(N)$ で，$r'\neq r$ となるすべての r' に対して，$Ver((s_{short}\|r), m_0, t_0)\neq 1, \ldots, Ver((s_{short}\|r), m_{N-1}, t_{N-1})\neq 1$ の少なくとも 1 つが成り立ち，このとき，B によってシミュレートされた A の環境は，$\widehat{\Pi}_{MAC}$ に対する EUF-CMA ゲームと同じとなり，B は Π_{MAC} に対する EUF-CMA ゲームに確率 ε で勝利することができる．したがって，B は確率 $(1-\delta(N))\varepsilon > \varepsilon-\delta(N)$ で Π_{MAC} の EUF-CMA ゲームに勝利する．

Q.E.D.

同様にして，Π_{MAC} が「非適応的選択メッセージ攻撃，もしくは，既知メッセージ攻撃」に対する「存在的偽造不可能性，もしくは，選択的偽造不可能性，一般的偽造不可能性」を持つならば，$\widehat{\Pi}_{MAC}$ も同じ安全性を持つことが証明できる．

3.3 計算機援用公開鍵暗号

3.3.1 モデル

通常の公開鍵暗号 Π_{PKE} は 3 つのアルゴリズム (Gen, Enc, Dec) からなる．鍵生成アルゴリズム Gen は 1^k（k はセキュリティパラメータ）を入力とし，秘密鍵 sk，公開鍵 pk を出力する確率的アルゴリズムである．ここで，Gen が用いる乱数 r を $k_l(k)$ ビットとし，乱数を明示する場合は，$Gen(1^k; r)$ と書くことにする．暗号化アルゴリズム Enc は pk と平文 m を入力とし，暗号文 c を出力する．復号アルゴリズム Dec は sk と c を入力とし，m を出力する．すべての k，$(sk, pk)\leftarrow Gen(1^k; r)$，$m$ に対して，$Dec(sk, Enc(pk, m)) = m$ であること（正当性）が要求される．

Π_{PKE} の安全性モデルは，様々な定義が存在する．その中で最も強い安全性は，適応的選択暗号文攻撃に対する識別不可能性（IND-CCA2）[10] である．IND-CCA2 の攻撃ゲームは，IND-P2-C2 の攻撃ゲームとほぼ同様に定義できる．異なる部分は，初めに，$Gen(1^k; r)$ より，(sk, pk) が生成され，pk が攻撃者に与えられ，攻撃者は任意の平文に対する暗号文を自分で計算することができるという点である．

素因数分解を基とした公開鍵暗号の鍵生成アルゴリズムに対しては，安全のために，非常に長いビット長の一様乱数を入力することが望ましい．しかし，そのような一様乱数は生成が困難であるため，実際には，その代わりに，擬似乱数が用いられる．たとえば，RSA 暗号 [11] の sk は，ランダムに選ばれた素数 p，q と，$(p-1)(q-1)$ と互いに素な数の中からランダムに選ばれた e によって生成される．そして，実際の運用において，p，q として，一様乱数の代わりに擬似乱数生成器 PRG [12] より生成された擬似乱数が用いられる．したがって，実際に Π_{PKE} を運用する場合には，一般に PRG を用いた以下のような公開鍵暗号 Π_{PKE}^{PRG} が利用されている．Π_{PKE}^{PRG} は 3 つのアルゴリズム (Gen_{PRG}, Enc, Dec) からなる．鍵生成アルゴリズム Gen_{PRG} は，1^k を入力とし，$(sk, pk) = Gen(1^k; PRG(seed))$ を出力する．ただし，$seed$ は Gen_{PRG} が用いる k ビットの一様乱数である．Π_{PKE} と同様に，正当性が要求される．Π_{PKE}^{PRG} に対する攻撃ゲームは，初めに Gen_{PRG} より (sk, pk) が生成されること以外は，Π_{PKE} に対する攻撃ゲームとまったく同様である．

上述したように，素因数分解を基とした公開鍵暗号においては，安全のために，非常に長いビット長の一様乱数を鍵生成アルゴリズムに入力することが望ましい．そして，長い一様乱数を用いることにより，秘密鍵もまた，非常に長いビット列となる．たとえば，RSA 暗号の秘密鍵は，現時点で 2,048 ビット [8] である．そのため，秘密鍵を直接復元するという計算機援用方法は現実的ではない．もし，$s_{long} = s_{short}\|r$ を秘密鍵として用いた場合，s_{short} が記憶，もしくは所持できないほど長いビット長となるか，もしく

は r が総当たりで探索できないほど長いビット長となる．そこで，計算機援用公開鍵暗号 $\widehat{\Pi}_{PKE}$ においても，Π_{PKE}^{PRG} と同様に，PRG を用いる．すなわち，s_{long} を PRG に入力するシードとし，生成された擬似乱数を鍵生成アルゴリズムに入力するという方法をとる．

計算機援用公開鍵暗号 $\widehat{\Pi}_{PKE}$ も，Π_{PKE}^{PRG} と同様に 3 つのアルゴリズム $(\widehat{Gen}, \widehat{Enc}, \widehat{Dec})$ からなる．鍵&ヒント生成アルゴリズム $\widehat{Gen}$ は，1^k と $str \in \{0,1\}^{k_u}$ を入力とし，$s_{short} \in \{0,1\}^{k_u}$，$hint$，$pk$ を出力する．暗号化アルゴリズム $\widehat{Enc}$ は Enc とまったく同様のアルゴリズムである．復号アルゴリズム $\widehat{Dec}$ は s_{short}, c, $hint$ を入力として，m を出力する．すべての k，str，$(s_{short}, hint, pk) \leftarrow \widehat{Gen}(1^k, str)$，$m$ に対して，$\widehat{Dec}(s_{short}, \widehat{Enc}(pk, m), hint) = m$ が成り立つ確率は，1 に限りなく近いこと（正当性）が要求される．

$\widehat{\Pi}_{PKE}$ に対する攻撃ゲームは，Π_{PKE}^{PRG} に対する攻撃ゲームとほぼ同様に定義できる．異なる部分は，これまでの CASS に対する攻撃ゲームと同様に，オラクルが (Gen_{PRG}, Enc, Dec) ではなく，$(\widehat{Gen}, \widehat{Enc}, \widehat{Dec})$ を実行し，攻撃者が $hint$ を得ることができるという点である．

3.3.2 一般的構成法

$\Pi_{PKE}^{PRG} = (Gen_{PRG}, Enc, Dec)$ を用いて，$\widehat{\Pi}_{PKE} = (\widehat{Gen}, \widehat{Enc}, \widehat{Dec})$ を次のように構成できる．

鍵&ヒント生成 $\widehat{Gen}(1^k, str)$：k ビットの一様乱数 s_{long} を生成する．次に，鍵ペア $(pk, sk) = Gen_{PRG}(1^k; s_{long})$ を生成する．$(s_{short}, hint, pk)$ を出力する．ここで，$s_{long} = s_{short} \| r$，$|s_{short}| = |str| = k_u$，$hint = pk$ であるため，実際に出力しているのは (s_{short}, pk) である．

暗号化 $\widehat{Enc}(pk, m)$：暗号文 $c = Enc(pk, m)$ を出力する．

復号 $\widehat{Dec}(s_{short}, c, hint)$：初めに，$Gen_{PRG}(1^k; s_{short} \| r) = pk$ を満たす r を総当たりで探索し，$s_{long} = s_{short} \| r$ を復元する．復元されると同時に sk も復元される．次に，c を復号し，平文 $m = Dec(sk, c)$ を出力する．

$r \neq r'$ に対して，$Gen_{PRG}(1^k; s_{short} \| r) = (pk, sk)$，$Gen_{PRG}(1^k; s_{short} \| r) = (pk, sk')$ であった場合，鍵&ヒント生成において用いられた r とは異なる r' が復元され，$sk' \neq sk$ である可能性がある．しかし，Π_{PKE} の正当性から，sk' は sk とまったく同じように秘密鍵として利用できるため，$\widehat{\Pi}_{PKE}$ も正当性を持つ．

なお，RSA-OAEP のようにランダムオラクルを仮定している場合は，$hint$ を s_{long} のハッシュ値 $h(s_{long})$ とすることで，上述の構成法よりも正規ユーザの計算コストの負担を減らすことができる．この場合，復号では，$h(s_{long}) = h(s_{short} \| r)$ を満たす r を総当たりで探索し，r を復元する．あるいは，秘密情報拡張型の CASS を用いることもできる．

3.3.3 安全性証明

ここでは，$\widehat{\Pi}_{PKE}$ が Π_{PKE} とほぼ同じ安全性を持つことを証明する．安全性証明は，次の 2 段階で行う．1 段階目として，Π_{PKE} より構成された Π_{PKE}^{PRG} は，安全な PRG を用いれば，Π_{PKE} とほとんど同じ安全性を持つことを証明する．2 段階目として，Π_{PKE}^{PRG} より構成された $\widehat{\Pi}_{PKE}$ は，Π_{PKE}^{PRG} と同じ安全性を持つことを証明する．これらの証明により，Π_{PKE} が安全であり，かつ安全な PRG を用いれば，Π_{PKE}^{PRG} は安全であり，そしてその Π_{PKE}^{PRG} より構成された $\widehat{\Pi}_{PKE}$ もまた安全であることが証明できる．ここでは，もし Π_{PKE} が最も強い安全性を持っていたら，Π_{PKE}^{PRG}，$\widehat{\Pi}_{PKE}$ も最も強い安全性を持つことを証明する．

まず，Π_{PKE} より構成された Π_{PKE}^{PRG} は，安全な PRG を用いれば，Π_{PKE} とほとんど同じ安全性を持つことを証明する．PRG の安全性を示す擬似ランダム性は，次の攻撃ゲームによって定義される．初めに，チャレンジオラクルが，ランダムにチャレンジビット b を選ぶ．$b = 0$ の場合は，真性乱数 r を生成する．$b = 1$ の場合は，シード $seed$ を生成し，$r = PRG(seed)$ を計算する．次に，攻撃者は r を与えられる．最後に，攻撃者は推測ビット b' を出力する．$b = b'$ であれば攻撃者の勝ち．攻撃者のアドバンテージは $|\Pr[b = b'] - 1/2|$ で定義される．

Theorem 3. Π_{PKE} が IND-CCA2 安全であり，かつ用いる PRG が安全ならば，Π_{PKE}^{PRG} もまた IND-CCA2 安全である．より正確には，Π_{PKE} が IND-CCA2 安全であり，Π_{PKE}^{PRG} に対して，無視できないアドバンテージで IND-CCA2 ゲームに勝利する攻撃者 A が存在するならば，PRG の攻撃ゲームに無視できないアドバンテージで勝利する攻撃者 B が構成できる．

Proof. 仮定より，A を含むあらゆる攻撃者は，Π_{PKE} の IND-CCA2 ゲームに無視できないアドバンテージで勝利することはできない．一方，A は Π_{PKE}^{PRG} の IND-CCA2 ゲームに無視できないアドバンテージで勝利することができる．すなわち，A が Π_{PKE} の IND-CCA2 ゲームに勝利する確率は無視できるほど小さい γ によって $(1/2 \pm \gamma)$ で表され，Π_{PKE}^{PRG} の IND-CCA2 ゲームに勝利する確率は無視できないほど大きい ζ によって $(1/2 \pm \zeta)$ で表される．そして，Π_{PKE} の IND-CCA2 ゲームと Π_{PKE}^{PRG} の IND-CCA2 ゲームの違いは，鍵生成に一様乱数を用いているか，PRG の出力する擬似乱数を用いているかのみである．そこで，B は，A の 2 つのゲームに対する勝ち負けの結果を利用することにより，PRG の攻撃ゲームにアドバンテージ $(\zeta - \gamma)/2$ で勝利することができる．

具体的には次のような手順となる．初めに，PRG に対する攻撃ゲームのチャレンジオラクルが，ランダムにチャレンジビット b を選ぶ．$b = 0$ の場合は，一様乱数 r を生成する．$b = 1$ の場合は，シード $seed$ を生成し，$r = PRG(seed)$ を計算する．次に，B は r を与えられる．B は与えられた r を用いて，Gen を実行し，(sk, pk) を得る．次に，B は (sk, pk) を用いて，A との間で IND-CCA2 ゲームをシミュ

レートする．A がその IND-CCA2 ゲームに敗北した場合は，B は r が一様乱数だと判断し，推測ビット $b'=0$ を出力する．一方，A がその IND-CCA2 ゲームに勝利した場合は，B は r が擬似乱数だと判断し，$b'=1$ を出力する．

$b=0$ の場合に，B が $b'=0$ を出力する確率は，A が Π_{PKE} の IND-CCA2 ゲームに敗北する確率 $1-(1/2\pm\gamma)$ である．$b=1$ の場合に，B が $b'=1$ を出力する確率は，A が Π_{PKE}^{PRG} の IND-CCA2 ゲームに勝利する確率 $1/2\pm\zeta$ である．よって，B は確率 $(1\pm\gamma\pm\zeta)/2$ で PRG の攻撃ゲームに勝利するため，アドバンテージは $(\zeta-\gamma)/2$ となる．

Q.E.D.

同様にして，Π_{PKE} が「非適応的選択暗号文攻撃，もしくは，選択平文攻撃」に対する「識別不可能性，もしくは，一方向性」を持ち，かつ PRG が安全ならば，Π_{PKE}^{PRG} も同じ安全性を持つことが証明できる．

次に，Π_{PKE}^{PRG} より構成された $\widehat{\Pi}_{PKE}$ は，Π_{PKE}^{PRG} と同じ安全性を持つことを証明する．

Theorem 4. Π_{PKE}^{PRG} が IND-CCA2 安全ならば，$\widehat{\Pi}_{PKE}$ もまた IND-CCA2 安全である．より正確には，$\widehat{\Pi}_{PKE}$ に対して，暗号化クエリを合計 q 回行い，無視できないアドバンテージで IND-CCA2 ゲームに勝利する攻撃者 A が存在するならば，Π_{PKE}^{PRG} に対して，暗号化クエリを合計 q 回行い，無視できないアドバンテージで IND-CCA2 ゲームに勝利する攻撃者 B が存在する．

Proof. 我々の $\widehat{\Pi}_{PKE}$ の構成では，$hint=pk$ であるため，Theorem 1，2 の証明とは異なり，Π_{PKE}^{PRG} に対する IND-CCA2 ゲームと，$\widehat{\Pi}_{PKE}$ に対する IND-CCA2 ゲームは，攻撃の際に入手できる情報はまったく同じである．そのため，初めに，B は公開鍵かつ $hint$ である pk を与えられたら，それを A に入力し，あとは B は A からのクエリをそのままオラクルに送り，オラクルからの返答をそのまま A に返してやるだけでよい．ここで，A は，$\widehat{\Pi}_{PKE}$ に対する IND-CCA2 ゲームに無視できないアドバンテージ ε で勝利する（A の勝利確率は $1/2\pm\varepsilon$）．

具体的には次のような手順となる．初めに，一様乱数 $seed$ が生成され，$seed$ を用いて鍵ペア $(pk,sk)\leftarrow Gen_{PRG}(1^k;seed)$ が生成される．次に，B に pk が入力される．B は，pk を A に入力する．この時点で B は A を起動する．A は，B に c_i を適応的に送ってくる．B は，c_i が送られてきたら，復号オラクルに c_i を送り，$m_i=Dec(sk,c_i)$ を得て，A に m_i を返す．A が，(m_0^*,m_1^*) を B に送ってきたら，B はチャレンジオラクルに (m_0^*,m_1^*) を送る．チャレンジオラクルはランダムにチャレンジビット b を選び，$c_b^*=Enc(pk,m_b^*)$ を計算し，B にそれを返す．B は，A に c_b^* を返す．最後に，A は推測ビット b' を出力するので，B は b' を出力する．

上記の A の環境は，$\widehat{\Pi}_{PKE}$ に対する IND-CCA2 ゲームと同じである．したがって，A の出力は少なくとも確率 $1/2\pm\varepsilon$ で $b'=b$ を満たす．すなわち，B は確率 $1/2\pm\varepsilon$ で Π_{PKE}^{PRG} に対する IND-CCA2 ゲームに勝利するため，アドバンテージは ε となる．

Q.E.D.

同様にして，Π_{PKE}^{PRG} が「非適応的選択暗号文攻撃，もしくは，選択平文攻撃」に対する「識別不可能性，もしくは，一方向性」を持つならば，$\widehat{\Pi}_{PKE}$ も同じ安全性を持つことが証明できる．

3.4 計算機援用デジタル署名

3.4.1 モデル

通常のデジタル署名 Π_{DS} は 3 つのアルゴリズム (Gen,Sig,Ver) からなる．鍵生成アルゴリズム Gen は 1^k（k はセキュリティパラメータ）を入力とし，秘密鍵 sk，公開鍵 pk を出力する確率的アルゴリズムである．公開鍵暗号と同様に，Gen が用いる乱数 r を $k_l(k)$ ビットとし，乱数を明示する場合は，$Gen(1^k;r)$ と書くことにする．署名アルゴリズム Sig は sk とメッセージ m を入力とし，署名 σ を出力する．検証アルゴリズム Ver は pk，m，σ を入力とし，検証結果 b を出力し，$b=1$ なら正当な署名として受理し，$b=0$ なら不正な署名として拒否する．すべての k，$(sk,pk)\leftarrow Gen(1^k;r)$，$m$ に対して，$Ver(pk,m,Sig(sk,m))=1$ であること（正当性）が要求される．

Π_{DS} の安全性モデルは，様々な定義が存在する．その中で最も強い安全性は，適応的選択メッセージ攻撃に対する存在的偽造不可能性（EUF-CMA）[13] である．EUF-CMA の攻撃ゲームは，MAC における EUF-CMA ゲームとほぼ同様に定義できる．異なる部分は，初めに，$Gen(1^k;r)$ より，(sk,pk) が生成され，pk が攻撃者に与えられるという点である．

公開鍵暗号と同様にして，実際に Π_{DS} を運用する場合には，一般に PRG を用いた以下のようなデジタル署名 Π_{DS}^{PRG} が利用されている．Π_{DS}^{PRG} は 3 つのアルゴリズム (Gen_{PRG},Sig,Ver) からなる．鍵生成アルゴリズム Gen_{PRG} は，1^k を入力とし，$(sk,pk)=Gen(1^k;PRG(seed))$ を出力する．ただし，$seed$ は Gen_{PRG} が用いる k ビットの一様乱数である．Π_{DS} と同様に正当性が要求される．Π_{DS}^{PRG} に対する攻撃ゲームは，初めに Gen_{PRG} より (sk,pk) が生成されること以外は，Π_{DS} に対する攻撃ゲームとまったく同様である．

公開鍵暗号と同様にして，計算機援用デジタル署名においても，PRG を用いる．計算機援用デジタル署名 $\widehat{\Pi}_{DS}$ も，Π_{DS}^{PRG} と同様に 3 つのアルゴリズム $(\widehat{Gen},\widehat{Sig},\widehat{Ver})$ からなる．鍵&ヒント生成アルゴリズム $\widehat{Gen}$ は，1^k と $str\in\{0,1\}^{k_u}$ を入力とし，$s_{short}\in\{0,1\}^{k_u}$，$hint$，$pk$ を出力する．署名アルゴリズム $\widehat{Sig}$ は s_{short}，m，$hint$ を入力として，σ を出力する．検証アルゴリズム $\widehat{Ver}$ は

Ver とまったく同様のアルゴリズムである．すべての k，str，$(s_{short}, hint, pk) \leftarrow \widehat{Gen}(1^k, str)$，$m$ に対して，$\widehat{Ver}(pk, m, \widehat{Sig}(s_{short}, m, hint)) = 1$ が成り立つ確率は 1 に限りなく近いこと（正当性）が要求される．

$\widehat{\Pi}_{DS}$ に対する攻撃ゲームは，Π_{DS}^{PRG} に対する攻撃ゲームとほぼ同様に定義できる．異なる部分は，これまでの CASS に対する攻撃ゲームと同様に，オラクルが (Gen_{PRG}, Sig, Ver) ではなく，$(\widehat{Gen}, \widehat{Sig}, \widehat{Ver})$ を実行し，攻撃者が $hint$ を得ることができるという点である．

3.4.2 一般的構成法

$\Pi_{DS}^{PRG} = (Gen_{PRG}, Sig, Ver)$ を用いて，$\widehat{\Pi}_{DS} = (\widehat{Gen}, \widehat{Sig}, \widehat{Ver})$ を次のように構成できる．

鍵&ヒント生成 $\widehat{Gen}(1^k, str)$：k ビットの一様乱数 s_{long} を生成する．次に，鍵ペア $(pk, sk) = Gen_{PRG}(1^k; s_{long})$ を生成する．$(s_{short}, hint, pk)$ を出力する．ここで，$s_{long} = s_{short} \| r$，$|s_{short}| = |str| = k_u$，$hint = pk$ であるため，実際に出力しているのは (s_{short}, pk) である．

署名 $\widehat{Sig}(s_{short}, m, hint)$：初めに，$Gen_{PRG}(1^k; s_{short} \| r) = pk$ を満たす r を総当たりで探索し，$s_{long} = s_{short} \| r$ を復元する．復元されると同時に sk も復元される．次に，m の署名を生成し，署名 $\sigma = Sig(sk, m)$ を出力する．

検証 $\widehat{Ver}(pk, m, \sigma)$：$pk$ を用いて m と σ に対して検証を行い，検証結果 $b = Ver(pk, m, \sigma)$ を出力する．

$r \neq r'$ に対して，$Gen_{PRG}(1^k; s_{short} \| r) = (pk, sk)$，$Gen_{PRG}(1^k; s_{short} \| r') = (pk, sk')$ であった場合，鍵&ヒント生成において用いられた r とは異なる r' が復元され，$sk' \neq sk$ である可能性がある．しかし，Π_{DS} の正当性から，sk' は sk とまったく同じように秘密鍵として利用できるため，$\widehat{\Pi}_{DS}$ も正当性を持つ．

なお，Π_{DS}^{PRG} はランダムオラクルを仮定している方式が多いため，通常は $hint$ を $h(s_{long})$ とすればよい．しかし，ランダムオラクルを仮定していない場合は，上述の構成法をとればよい．

3.4.3 安全性証明

$\widehat{\Pi}_{PKE}$ と同様に，$\widehat{\Pi}_{DS}$ の安全性証明においても 2 段階の証明を行う．ここでは，もし Π_{DS} が最も強い安全性を持っていたら，Π_{DS}^{PRG}，$\widehat{\Pi}_{DS}$ もまた最も強い安全性を持つことを証明する．

Theorem 5. Π_{DS} が EUF-CMA 安全であり，かつ用いる PRG が安全ならば，Π_{DS}^{PRG} もまた EUF-CMA 安全である．より正確には，Π_{DS} が EUF-CMA 安全であり，Π_{DS}^{PRG} に対して，無視できない確率で EUF-CMA ゲームに勝利する攻撃者 A が存在するならば，PRG の攻撃ゲームに無視できない確率で勝利する攻撃者 B が構成できる．

Proof. Theorem 3 の証明における IND-CCA2 を EUF-CMA に変えれば証明できるため，省略する．

Q.E.D.

同様にして，Π_{DS} が「非適応的選択メッセージ攻撃，もしくは，既知メッセージ攻撃，鍵単独攻撃」に対する「存在的偽造不可能性，もしくは，選択的偽造不可能性，一般的偽造不可能性」を持ち，かつ PRG が安全ならば，Π_{DS}^{PRG} も同じ安全性を持つことが証明できる．

Theorem 6. Π_{DS}^{PRG} が EUF-CMA 安全ならば，$\widehat{\Pi}_{DS}$ もまた EUF-CMA 安全である．より正確には，$\widehat{\Pi}_{DS}$ に対して，署名生成クエリを合計 q 回行い，無視できない確率で EUF-CMA ゲームに勝利する攻撃者 A が存在するならば，Π_{DS}^{PRG} に対して，署名生成クエリを合計 q 回行い，無視できない確率で EUF-CMA ゲームに勝利する攻撃者 B が存在する．

Proof. 証明のスケッチは Theorem 4 の証明と同様である．異なる部分は，Thereom 4 の証明における IND-CCA2 が，EUF-CMA に置き換わる点である．ここで，A は，$\widehat{\Pi}_{DS}$ に対する EUF-CMA ゲームに無視できない確率 ε で勝利する．B によってシミュレートされた A の環境は，$\widehat{\Pi}_{DS}$ に対する EUF-CMA ゲームと同じである．すなわち，B は確率 ε で Π_{DS}^{PRG} の EUF-CMA ゲームに勝利する．

Q.E.D.

同様にして，Π_{DS}^{PRG} が「非適応的選択メッセージ攻撃，もしくは，既知メッセージ攻撃，鍵単独攻撃」に対する「存在的偽造不可能性，もしくは，選択的偽造不可能性，一般的偽造不可能性」を持つならば，$\widehat{\Pi}_{DS}$ も同じ安全性を持つことが証明できる．

3.5 計算機援用 C & R 型ユーザ認証

3.5.1 モデル

通常の C&R 型ユーザ認証 Π_{CRUA} は 5 つのアルゴリズム $(Gen, Enr, CG, C2R, Ver)$ からなる．秘密情報生成アルゴリズム Gen は 1^k（k はセキュリティパラメータ）を入力とし，一様ランダムに k ビット列 s_{long} を生成し，それを秘密情報として出力する．登録アルゴリズム Enr は s_{long} を入力とし，検証情報 VI を出力する．チャレンジ生成アルゴリズム CG は 1^k を入力とし，チャレンジ C を出力する．レスポンス生成アルゴリズム $C2R$ は s_{long} と C を入力とし，レスポンス R を出力する．検証アルゴリズム Ver は C，R，VI を入力とし，検証結果 b を出力し，$b = 1$ なら認証に成功したものとし，$b = 0$ なら失敗したものとする．すべての k，$s_{long} \leftarrow Gen(1^k)$，$C \leftarrow CG(1^k)$ に対して，$Ver(C, C2R(s_{long}, C), Enr(s_{long})) = 1$ であること（正当性）が要求される．

Π_{CRUA} の安全性モデルは，様々な定義が存在する．その中で最も強い安全性は，concurrent man-in-the-middle attack によるなりすまし（IMP-CMIM）[16], [20] に対する安全性である．その攻撃ゲームでは，初めに，$Gen(1^k)$ より，s_{long} が生成される．次に，$Enr(s_{long})$ より VI が生成され，VI が攻撃者に送られる．攻撃者は，サーバオラ

クル，複数の正規ユーザオラクル P_j に対して，適応的にクエリを送ることができる[*2]．複数の正規ユーザオラクル P_j は，同じ s_{long} を持っているが，初めにセットされるランダムテープが異なる．攻撃者が ${C'}_j$ を正規ユーザオラクル P_j に送ったら，P_j は，${R'}_j = C2R(s_{long}, {C'}_j)$ を返し，内部状態を「認証終了」に更新し，以後はクエリを受け付けない．サーバオラクルは，C^* を攻撃者に送る．なお，攻撃者が C^* をクエリとして，P_j に送ることは禁止される．最後に，攻撃者は，C^* に対応する R^* を出力する．$Ver(C^*, R^*, VI) = 1$ であれば攻撃者の勝ち．

計算機援用 C&R 型ユーザ認証 $\widehat{\Pi}_{CRUA}$ も，Π_{CRUA} と同様に，5 つのアルゴリズム $(\widehat{Gen}, \widehat{Enr}, \widehat{CG}, \widehat{C2R}, \widehat{Ver})$ からなる．秘密情報&ヒント生成アルゴリズム $\widehat{Gen}$ は 1^k と $str \in \{0,1\}^{k_u}$ を入力とし，$s_{long} \in \{0,1\}^k$, $s_{short} \in \{0,1\}^{k_u}$, $hint$ を出力する．登録アルゴリズム $\widehat{Enr}$ は Enr と，チャレンジ生成アルゴリズム $\widehat{CG}$ は CG とまったく同様のアルゴリズムである．レスポンス生成アルゴリズム $\widehat{C2R}$ は s_{short}, C, $hint$ を入力とし，R を出力する．検証アルゴリズム $\widehat{Ver}$ は Ver とまったく同様のアルゴリズムである．すべての k, str, $(s_{long}, s_{short}, hint) \leftarrow \widehat{Gen}(1^k, str)$, $C \leftarrow \widehat{CG}(1^k)$ に対して，$\widehat{Ver}(C, \widehat{C2R}(s_{short}, C, hint), \widehat{Enr}(s_{long})) = 1$ が成り立つ確率は 1 に限りなく近いこと（正当性）が要求される．

$\widehat{\Pi}_{CRUA}$ に対する攻撃ゲームは，Π_{CRUA} に対する攻撃ゲームとほぼ同様に定義できる．異なる部分は，これまでの CASS に対する攻撃ゲームと同様に，オラクルが $(Gen, Enr, CG, C2R, Ver)$ ではなく，$(\widehat{Gen}, \widehat{Enr}, \widehat{CG}, \widehat{C2R}, \widehat{Ver})$ を実行し，攻撃者が $hint$ を得ることができるという点である．

3.5.2 一般的構成法

$\Pi_{CRUA} = (Gen, Enr, CG, C2R, Ver)$ を用いて，$\widehat{\Pi}_{CRUA} = (\widehat{Gen}, \widehat{Enr}, \widehat{CG}, \widehat{C2R}, \widehat{Ver})$ を次のように構成できる．

秘密情報&ヒント生成 $\widehat{Gen}(1^k, str)$：1^k を入力とし，秘密情報 $s_{long} = Gen(1^k)$ を生成する．次に，CG に 1^k を入力し，N 個のチャレンジ $C_0, \ldots, C_{N-1}$ を生成し，それに対応するレスポンス $R_0 = C2R(s_{long}, C_0), \ldots, R_{N-1} = C2R(s_{long}, C_{N-1})$ を生成する．$(s_{long}, s_{short}, hint)$ を出力する．ここで，$s_{long} = s_{short} \| r$，$|s_{short}| = |str| = k_u$，$hint = ((C_0, R_0), \ldots, (C_{N-1}, R_{N-1}))$ である．

登録 $\widehat{Enr}(s_{long})$：Enr に s_{long} を入力し，検証情報 VI を生成し，出力する．

チャレンジ生成 $\widehat{CG}(1^k)$：CG に 1^k を入力し，チャレンジ C を生成し，出力する．

レスポンス生成 $\widehat{C2R}(s_{short}, C, hint)$：初めに，$Ver(C_0, R_0, Enr(s_{short} \| r)) = 1, \ldots, Ver(C_{N-1}, R_{N-1}, Enr(s_{short} \| r)) = 1$ をすべて満たす r を総当たりで探索し，$s_{long} = s_{short} \| r$ を復元する．次に，s_{long} と C を用いてレスポンス $R = C2R(s_{long}, C)$ を生成し，出力する．

検証 $\widehat{Ver}(\mathbf{C}, \mathbf{R}, \boldsymbol{VI})$：$Ver$ に C，R，VI を入力し，検証結果 b を生成し，出力する．

N の決定については，共通鍵暗号，MAC の場合と同様に，$Ver(C_i, R_i, Enr(s_{short} \| r')) = 1$ がすべての $i \in \{0, \ldots, N-1\}$ について成り立つような r'（$\neq r$）が存在する確率 $\delta(N)$ が無視できるほど小さくなるように選ぶ．このとき，r は圧倒的確率で正しく復元されるため，正当性が成り立つ．

具体的な N の値は Π_{CRUA} に依存する．典型的な Π_{CRUA} のアルゴリズムとして，ハッシュ関数 h を用いたアルゴリズムがあげられる．具体的には，$VI = h(s_{long})$, $R = h(h(s_{long}), C)$ であり，Ver ではレスポンスを再計算して，受け取ったレスポンスと等しいかを確認するというアルゴリズムである．このアルゴリズムの場合は $N = 1$ で十分である．なぜならば，$\delta(1)$ が有意に高ければ，それは用いているハッシュ関数の衝突困難性を破っていることになる．すなわち，用いているハッシュ関数が衝突困難性を有していれば，$\delta(1)$ は無視できるほど小さい値となる．なお，$\delta(N)$ を無視できるほど小さくすることが困難である場合は，我々の構成によって計算機援用 C&R 型ユーザ認証は構成できない．

3.5.3 安全性証明

$\widehat{\Pi}_{CRUA}$ は，上記の構成法を用いて Π_{CRUA} から構成されたと仮定する．そのような $\widehat{\Pi}_{CRUA}$ は，Π_{CRUA} とほとんど同じ安全性を持つことを証明する．ここでは，もし Π_{CRUA} が最も強い安全性を持っていたら，$\widehat{\Pi}_{CRUA}$ も最も強い安全性を持つことを証明する．

Theorem 7. Π_{CRUA} が IMP-CMIM に対して安全ならば，$\widehat{\Pi}_{CRUA}$ もまた IMP-CMIM に対して安全である．より正確には，$\widehat{\Pi}_{CRUA}$ に対して，正規ユーザオラクルへのクエリを合計 q 回行い，無視できない確率で IMP-CMIM ゲームに勝利する攻撃者 A が存在するならば，Π_{CRUA} に対して，正規ユーザオラクルへのクエリを合計 $q + N$ 回行い，無視できない確率で IMP-CMIM ゲームに勝利する攻撃者 B が存在する．

Proof. 証明のスケッチは，Theorem 1，2 の証明と同様である．しかし，Theorem1，2 の IND-P2-C2，EUF-CMA をそのまま IMP-CMIM に置き換えればよいというわけではない．ここで，A は，$\widehat{\Pi}_{CRUA}$ に対する IMP-CMIM ゲームに無視できない確率 ε で勝利するものとする．

具体的には次のような手順となる．以下では，A は正規ユーザオラクルとして $P_0, \ldots, P_{q-1}$ を持ち，B は $P_0, \ldots, P_{q-1+N}$ を持つものとする．まず B は，$Enr(Gen(1^k))$ により生成された検証情報 VI を受け取る．B は，A に $hint$ であるチャレンジとレスポンス

[*2] ここでは，ユーザ側のみを concurrent とする left-concurrency を考える．

の N ペアを入力するために，$CG(1^k)$よりチャレンジ $C_0,\ldots,C_{N-1}$ を生成し，$C_0,\ldots,C_{N-1}$ を正規ユーザオラクル $P_q,\ldots,P_{q-1+N}$ に送る．$P_q,\ldots,P_{q-1+N}$ は，$R_0 = C2R(s_{long},C_0),\ldots,R_{N-1} = C2R(s_{long},C_{N-1})$ を B に返すので，B は $((C_0,R_0),\ldots,(C_{N-1},R_{N-1}))$ を $hint$ として，VI とともに，A に入力する．この時点で B は A を起動する．A は，B に C_i を正規ユーザオラクル P_j に対するクエリとして，適応的に送ってくる．B は自身のオラクル P_j にそのままクエリを送り，オラクルからの返答を A に返す．サーバオラクルが $C^* = CG(1^k)$ を，B に送ってきたら，B は，A に C^* を送る．最後に，A は R^* を出力するので，B は R^* を出力する．

確率 $1-\delta(N)$ で，$r' \neq r$ となるすべての r' に対して，$Ver(C_0,R_0,Enr(S_{short}\|r')) \neq 1,\ldots,Ver(C_{N-1},R_{N-1},Enr(s_{short}\|r')) \neq 1$ の少なくとも 1 つが成り立ち，このとき，B によってシミュレートされた A の環境は，$\widehat{\Pi}_{CRUA}$ に対する IMP-CMIM ゲームと同じとなり，B は Π_{CRUA} に対する IMP-CMIM ゲームに確率 ε で勝利することができる．したがって，B は確率 $(1-\delta(N))\varepsilon > \varepsilon - \delta(N)$ で Π_{CRUA} の IMP-CMIM ゲームに勝利する．

同様にして，π_{CRUA} が「man-in-the-middle attack，concurrent attack [14]，もしくは，能動的攻撃，受動的攻撃」に対する「なりすまし耐性，パスワード推測耐性」を持つならば，$\widehat{\Pi}_{CRUA}$ も同じ安全性を持つことが証明できる．

4. CASS の実現可能性

秘密情報補完型 CASS においては，その時代で安全となる長さに s_{long} を設定し，それに合わせて正規ユーザは s_{short} と r の長さを調節する．将来的に CPU の計算機能力が向上し，安全となる s_{long} の長さが大きくなった場合でも，正規ユーザも計算機能力を使用するため，r の長さを調節することにより，s_{short} の長さを変えずに，r の復元に要する時間を増加させずに，s_{long} の長さを大きくすることが可能である．本章では，s_{short} がパスワード，または生体情報である場合を例に，秘密情報補完型 CASS の実現可能性について議論する．

初めに，s_{short} がパスワードの場合について議論する．現在は 128 ビット鍵が主流であり，現在のコンピュータでは短時間（たとえば 1 秒）で 2^{26} 程度の処理が可能なため r を 26 ビットとすると，s_{short} は 102 ビットとなる．文献 [15] で，人間は 102 ビットのパスワードを記憶できることが示されている．将来，計算機能力が 2^{64} 倍に増えたときには，192 ビット鍵へと移行する必要ある．s_{short} の長さは人間の記憶可能な 102 ビットのまま $s_{long} = s_{short}\|r$ の長さを 128 ビットから 192 ビットにするためには，r の長さを 26 ビットから 90 ビットへと変更する必要があるが，計算機の能力が 2^{64} 倍となっているため，r の復元に要する時間は 1 秒程度の短時間に保たれる．したがって，時代が変わろうとも，パスワードを用いた秘密情報補完型 CASS は実現可能であるといえる．

次に，s_{short} が生体情報の場合について議論する．生体情報のエントロピーについては，その評価尺度の定義や評価方法に関して，様々な提案がなされている．たとえば，文献 [17] では，ある生体認証システム S を通じて観測できる生体情報のエントロピーを，S における本人同士の照合スコアの確率分布 $f_G(x)$ と，他人同士の照合スコアの確率分布 $f_I(x)$ の間の Kullback-Leibler 距離 [18] $D(f_G\|f_I)$ によって評価することを提案している．S の出力が OK（一致）/NG（不一致）の 2 値であった場合は，本人なのに不一致となる確率 β と，他人なのに一致となる確率 α を用いて，$D(f_G\|f_I) = (1-\beta)\log_2\frac{1-\beta}{\alpha} + \beta\log_2\frac{\beta}{1-\alpha}$ と表現でき，$\beta \ll 1$ の場合は近似的に $D(f_G\|f_I) = -\log_2\alpha$ と表すことができる．そして，文献 [19] で提案されている指静脈を用いたバイオメトリック暗号においては，ユーザが生体情報を入力するたびにそれまで得られたスコアを融合してユーザを判定する「逐次融合判定」を組み込むことにより，$\alpha = 2^{-64}$，2^{-80}，2^{-128} とすることができ，エントロピーとしては，$D(f_G\|f_I) = 64$，もしくは，80，128 ビットまで実現できることを実験的に示している．ただし，α の値を小さくするためには，指の入力回数（使用する指の本数）の平均値を増やす必要がある．

以上より，生体情報を 102 ビットの一様乱数である s_{short} として利用することは可能である．また，生体情報においても，将来的に CPU の計算機能力が向上し，安全となる s_{long} の長さが大きくなった場合でも，r の長さを大きくすることにより，指の入力回数の平均値を増やすことなく（s_{short} を大きくすることなく），r の復元に要する時間が 1 秒のまま，s_{long} の長さを大きくすることが可能となる．よって，時代が変わろうとも生体情報を用いた秘密情報補完型 CASS は実現可能であるといえる．

本章のこれまでの議論は，正規ユーザが，攻撃者と同じ性能を持つ計算機を利用できると仮定している．しかし，現実的には，正規ユーザはスマートフォン等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，攻撃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．そのような場合は，単純に「時代が変わろうとも，秘密情報補完型 CASS は実現可能である」ということにはならない．しかし，パスワードや，生体認証，PUF 等の秘密情報のエントロピーを増やすことが困難であるあらゆる情報に対して，計算機能力を使って安全性を高めることが可能であるということはいえる．

参考文献

[1] Gassend, B., Clarke, D., Dijk, M. and Devadas, S.: Silicon physical random functions, *9th ACM Conference*

on Computer and Communications Security, pp.148–160 (2002).
[2] Juels, A. and Wattenberg, M.: A fuzzy commitment scheme, *6th ACM Conference on Computer and Communications Security*, pp.28–36 (1999).
[3] Proves, N. and Mazieres, D.: A Future-Adaptable Password Scheme, *USENIX Annual Technical Conference* (1999).
[4] Turan, M.S., Barker, E., Burr, W. and Chen, L.: Recommendation for Password-Based Key Derivation Part 1: Storage Applications, *NIST Special Publication*, 800-132 (2010).
[5] Kelsey, J., Schneier, B., Hall, C. and Wagner, D.: Secure Applications of Low-Entropy Keys, *1997 Information Security Workshop*, pp.121–134 (1997).
[6] Yao, F.F. and Yin, Y.L.: Design and Analysis of Password-Based Key Derivation Functions, *IEEE Trans. Information Theory*, Vol.51, No.9, pp.3292–3297 (2005).
[7] 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).
[8] Barker, E. and Roginsky, A.: Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths, *NIST Special Publication*, 800-131A Revision 1 (2015).
[9] Katz, J. and Yung, M.: Characterization of Security Notions for Probabilistic Private-Key Encryption, *Journal of Cryptology*, Vol.19, No.1, pp.67–95 (2006).
[10] Rackoff, C. and Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack, *Proc. Advances in Cryptology – CRYPTO 91*, Vol.576, pp.433–444 (1991).
[11] Rivest, R.L., Shamir, A. and Adleman, L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *Comm. ACM*, Vol.21, No.2, pp.120–126 (1978).
[12] Yao, A.C.: Theory and Applications of Trapdoor Functions, *23rd Annual Symposium on Foundations of Computer Science*, pp.80–91 (1982).
[13] Goldwasser, S., Micali, S. and Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks, *SIAM Journal on Computing*, Vol.17, No.2, pp.281–308 (1988).
[14] Bellare, M. and Palacio, A.: GQ and Schnorr Identification Schemes, Proofs of Security against Impersonation under Active and Concurrent Attacks, *Advances in Cryptology – CRYPTO 2002*, pp.162–177 (2002).
[15] Fujita, M., Yamada, M., Arimura, S., Ikeya, Y. and Nishigaki, M.: An Attempt to Memorize Strong Passwords while Playing Games, *Network-Based Information Systems*, pp.264–268 (2015).
[16] Gennaro, R.: Multi-trapdoor Commitments and Their Applications to Proofs of Knowledge Secure Under Concurrent Man-in-the-Middle Attacks, *Advances in Cryptology – CRYPTO 2004*, pp.220–236 (2004).
[17] Takahashi, K. and Murakami, T.: A Measure of Information Gained through Biometric Systems, *Elsevier Image and Vision Computing*, Vol.32, No.12, pp.1194–1203 (2014).
[18] Kullback, S. and Leibler, R.A.: On Information and Sufficiency, *The Annuals of Mathematical Statistics*, Vol.22, No.1, pp.79–86 (1951).
[19] Murakami, T., Ohki, T. and Takahashi, K.: Optimal sequential fusion for multibiometric cryptosystems, Vol.32, pp.93–108 (2016).
[20] Anada, H. and Arita, S.: Identification Schemes of Proofs of Ability Secure against Concurrent Man-in-the-Middle Attacks, *International Conference on Provable Security, ProveSec 2010*, pp.18–34 (2010).

神農 泰圭

2016 年静岡大学情報学部情報科学科卒業．2018 年同大学院修士課程修了．在学中，情報セキュリティに関する研究に従事．

土屋 貴史

2015 年静岡大学情報学部情報科学科卒業．2017 年同大学院修士課程修了．在学中，情報セキュリティに関する研究に従事．

大木 哲史 （正会員）

2010 年早稲田大学理工学総合研究所次席研究員，2013 年産業技術総合研究所特別研究員を経て，2017 年より静岡大学大学院総合科学技術研究科講師．個人認証を中心としたネットワークセキュリティに関する研究に従事．

高橋 健太 （正会員）

1998 年東京大学理学部情報科学科卒業．2000 年同大学院理学系研究科情報科学専攻修士課程修了．同年（株）日立製作所入社．2012 年東京大学大学院情報理工学系研究科博士後期課程修了．博士（情報理工学）．2015 年より東京大学大学院客員准教授．現在，（株）日立製作所研究開発グループユニットリーダ主任研究員．生体認証，暗号技術および情報セキュリティの研究開発に従事．2008 年情報処理学会論文賞，2011 年 SISAP Best paper，2012 年 IEEE BTAS Best reviewed paper，2015 年情報処理学会長尾真記念特別賞，2016 年ドコモ・モバイル・サイエンス賞先端技術部門優秀賞，関東地方発明表彰発明奨励賞等受賞．電子情報通信学会会員．

尾形 わかは

1989年東京工業大学理学部物理学科卒業，1991年同大学院総合理工学研究科物理情報工学専攻修士課程修了，1994年同大学院理工学研究科電気・電子工学専攻博士後期課程修了．1995年兵庫県立姫路工業大学工学部情報工学科助手．2000年東京工業大学理財工学研究センター助教授，2013年同大学大学院イノベーションマネジメント研究科技術経営専攻教授．2015年より同大学工学院情報通信系教授．博士（工学）．公開鍵システム，マルチパーティプロトコルをはじめとする暗号理論全般の研究に従事．

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．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．2013～014年情報処理学会コンピュータセキュリティ研究会主査．2015～2016年電子情報通信学会バイオメトリクス研究専門委員会委員長．2016年より日本セキュリティマネジメント学会常任理事．本会フェロー．