修士論文

衝突困難ハッシュ関数を用いない電子署名に関する研究

A Study on Digital Signatures without Collision Resistant Hash Functions

指導教官　　松浦幹太　准教授

東京大学大学院 情報理工学系研究科 電子情報学専攻

66441　　松田 隆宏

平成20年2月4日提出

あらまし

電子署名の安全性とハッシュ関数の安全性とは非常に関係が深い。電子署名では、その用途にも依るが、様々なサイズのデータの署名を可能にするためには、データを一度ハッシュ関数に通し、一度固定長の空間に写すということが必要である。そして同時に、ハッシュ関数の安全性自体が電子署名の安全性証明の際の仮定の一つとしても用いられる。特によく用いられる安全性であるハッシュ関数の衝突困難性は、ハッシュ値が等しくなる任意の入力のペアを見つけることが困難であるという性質で、任意長の長さのメッセージをハッシュ関数をによって固定長に写し、そのハッシュ値に対して署名を付けるという “Hash-and-Sign” パラダイムを用いる方式などで用いられる。衝突困難性をある種の困難性の仮定の下に証明可能なハッシュ関数も存在するが、署名作成や検証の計算コストを考慮し、現実的には、実装の際は SHA-1 や MD5 などの実用的ハッシュ関数を用いることになる。

しかし近年、Wang らの SHA-1 への攻撃に代表されるこれら実用的ハッシュ関数に対して、従来考えられていたより遥かに少ない計算回数で衝突を見つける、衝突困難性を破る攻撃についての報告は、衝突困難性を持つ実用的ハッシュ関数を構成することは簡単ではないということを示している。

そこで本研究では、電子署名の中で使われるハッシュ関数は実用的ハッシュ関数に対する仮定までも考慮に入れ、衝突困難性を用いずに証明可能安全性を持ち、かつ効率のよい電子署名方式を目指した。そして、具体的に強偽造不可能性という電子署名における最強の安全性を持つ電子署名方式を 2 種類提案し、その安全性証明を示した。いずれの方式もスタンダードモデルでの CDH 仮定に基づいており、両方式は、スタンダードモデルで CDH 仮定に基づく強偽造不可能性を持つ電子署名方式で、最も効率が良い方式である BSW 署名と同程度に効率がよい。しかも安全性の証明の仮定としてハッシュ関数の衝突困難性を用いていないため、衝突発見攻撃によって内部で使用されているハッシュ関数の衝突困難性が破られても、電子署名としての安全性は揺るがない。

目次

Chapter 1　はじめに

1.1　電子署名とハッシュ関数

電子署名　我々は日常、文章の文責者を保証するために紙に判を押したり、署名を書いたりする。しかし、それらは紙媒体には有効な手段となるが、複製、改ざんが容易な電子データに対してはそうではない。そこで、電子的なデータにおいても、紙媒体と同要に文責者を保証するための仕組みとして、電子署名が用いられる (図 1.1)。

電子署名の主な役割は以下である。

- 署名作成者を特定することができる。
- 署名の対象であるデータが署名者以外により改竄された場合、署名を検証するための公開鍵 (検証鍵) を用いた検証アルゴリズムにより改竄の事実を検出できる。
- 署名者はいったん署名を生成すると、その署名を作成した事実を後で否認できない。

これらの機能を満足するために、以下の要件が求められる。

- 署名者が自分固有の秘密鍵 (署名鍵) を秘密に管理する限り、秘密鍵を持たない第三者による任意のデータに対する署名の偽造は困難である。
- 署名とそれに対応するデータが (複数) あって、それらを参考にしたとしても別のデータに対して正しい署名を計算すること (署名の偽造) は困難である。
- 同じ秘密鍵によって署名すると、同じ署名が生成されるような異なる複数のデータを見つけることは困難である。

このような安全性の要件は、“偽造不可能性” として定式化されている。

電子署名は、署名者の特定だけでなく、複製、改ざんが容易な電子データについて、署名が付けられてからのデータの改ざんを検知できるなどの紙媒体での署名よりも強力な機能を提供し、電子的データの完全性を保証する。特に、安全な通信を可能にする公開鍵暗号基盤 (Public Key Infrastructure, PKI) において非常に重要な役割を果たしている。

このような機能を持つ電子署名は、1976 年に Diffie と Hellman [29] により公開鍵暗号の概念と共にその概念が提唱されて以来、様々な方式が提案されている。似たような機能を持つものにメッセージ認証子 (Message Authentication Code, MAC) があるが、電子署名は、署名の検証は公開鍵により不特定多数の第三者でも行えるのに対し、メッセージ認証子では、検証は秘密鍵を持つ者しか行えないという違いがあり、使用の目的は異なっている。

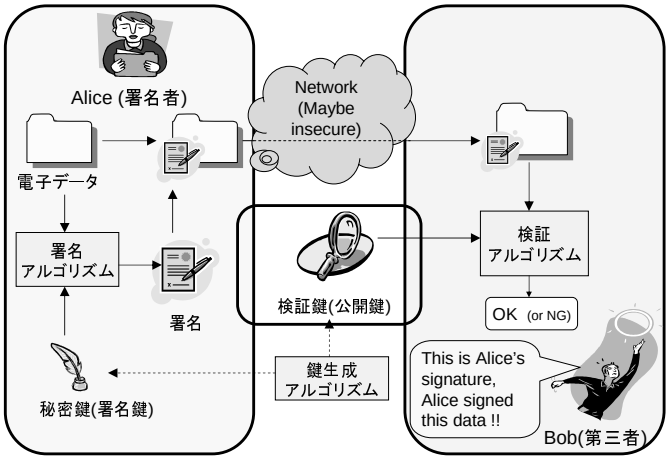

図 1.1: 電子署名の流れ

ハッシュ関数との関連性と問題　以降では電子的なデータのことを統一してメッセージと呼ぶことにする。電子署名では、その用途にも依るが、様々なサイズのデータの署名を可能にするためには、データを一度ハッシュ関数に通し、一度固定長の空間に写すということが必要である。従って、ハッシュ関数は、電子署名にほぼ必須の技術といえる。

この性質を持つハッシュ関数と、固定長 (例えば 160 ビット) のメッセージに対して偽造不可能な電子署名を付けることができる電子署名とを組み合わせれば、任意長 (実際はハッシュ関数の入力が許す限りだが、例えば SHA-1 では 2^{64} ビットなど、実用上は任意長と考えてよい) の長さのメッセージに対して偽造不可能な電子署名を付けることができる。現実的には、実装の際、SHA-1 や MD5 などを用いることになる。

電子署名の偽造不可能性は、ハッシュ関数の安全性と非常に関係が深い。特に電子署名で使われるハッシュ関数によく仮定される性質は衝突困難性 (Collision Resistance, CR) であり、ハッシュ値が同じになるような任意の入力ペアを見つけることが計算量的に難しいことを保証する性質である。電子署名方式において、任意長の長さのメッセージをハッシュ関数をによって固定長に写し、そのハッシュ値に対して署名を付けるという “Hash-and-Sign” パラダイム [28] を用いる方式では、ハッシュ関数の衝突困難性が破れると、もはや電子署名としての安全性である偽造不可能性を持っているとは言えなくなってしまう。理論的には、離散対数問題の困難性を仮定することで任意のサイズのメッセージを入力することができる衝突困難性を持つことが証明可能なハッシュ関数を構成することは可能である [23]。実際に、もし署名を付けたいメッセージが固定長でしかも小さいサイズであると予め分かっている方式では、そのような理論的構成に基づくハッシュ関数を用いた電子署名も実用に耐えうるかもしれない。しかし、もし署名されるメッセージのサイズが大きい、あるいは任意の長さをとりうる電子署名を構成する場合、署名や検証の計算の効率性を考慮に入れ、電子署名の中でメッセージを固定長に写すためのハッシュ関数は MD5 や SHA-1 などの実用的ハッシュ関数に置き換えられる。そしてそれら実用的ハッシュ関数は衝突困難性を満たすと仮定

して使用されることになる。

しかし近年、これら実用的ハッシュ関数 MD5 や SHA-1 などに対して、従来考えられていたより遥かに少ない計算回数で衝突を見つける、衝突困難性を破る攻撃についての大きな成果の報告が相次いでいる。特に、2005 年に報告された Wang ら [61] による SHA-1 への攻撃は、情報セキュリティ分野の専門家の間だけでなく、広く情報技術に携わる人々に対して大きな衝撃を与えた。このような近年の様々な実用的ハッシュ関数に対する衝突発見攻撃の研究の進展や成功報告は、衝突困難性を持っているような実用的ハッシュ関数を構成することは簡単ではないということを示している。よって、電子署名の中で使われるハッシュ関数は実用的ハッシュ関数に対する仮定までも考慮に入れて偽造不可能性を達成していると主張するために、電子署名の中で使われるハッシュ関数に必要とする仮定を衝突困難性よりも弱めることは取り組む価値があることである。

1.2　証明可能安全性

電子署名を含むあらゆる暗号技術は、現在知られている困難な問題に基づくなどして、安全性を証明可能であることが望ましい。証明可能安全性とは、暗号の安全性を形式的に定義し、数学的証明の正当性によって、定義の範囲内の安全の有無を判断できるようにするものである。安全性の証明がないことは、必ずしも安全ではないということを直接意味するわけではないが、より正確で、経験則の排除による客観的な安全性の議論を行うために、新たな方式考案の際など、安全性の証明をつけることは事実上当然のことになっている。

証明可能安全性を示すためには、示したい安全性目標のモデル、攻撃者の攻撃法のモデル、根拠とする困難な問題の形式的な定義を行う必要がある。根拠となる問題は、素因数分解問題や離散対数問題など、長くにわたって困難であると信じられている問題を使うことが多い。

安全性の定義は “現実に知られている難しい問題の困難性の仮定が成り立つならば安全性を無視できない確率で破るアルゴリズムが存在しない” というものになっている。証明の際には、その対偶を示すことで行われる。すなわち、安全性の証明をしたい方式を破る確率多項式時間アルゴリズムを入出力の決まったブラックボックスとして考え、その攻撃者を利用すると、安全性の根拠としたい困難な問題を解くことができるアルゴリズム[1]の存在を示すことができ、問題の困難性の仮定を破ることから、対偶により安全性を破る攻撃者は存在しないとするのである。確率多項式時間アルゴリズムは、現実に存在するアルゴリズムの能力を表している。この場合は方式の安全性を決定するセキュリティパラメータに対して多項式時間である。

[1]攻撃者にとっては、安全性の定義に用いられるゲーム (2.1、2.3 節参照) でのやりとりをしていることと変わりがなく、見分けが付かない様にできることを示さなければならない。攻撃者とチャレンジャーのやりとりをシミュレートすることから、このようなアルゴリズムをシミュレータ、あるいは、帰着アルゴリズム (Reduction Algorithm) という。本稿では前者の呼び名を用いる。

ランダムオラクルモデルとスタンダードモデル　証明可能安全性の枠組みにおいて、証明を考えるときにランダムオラクルモデル (Random Oracle Model) [31, 7] と呼ばれるモデルが使用されることがある。ランダムオラクルモデルとは、誰でもアクセスできる真にランダムな関数 (ランダムオラクル) が存在すると仮定するモデルのことである。これに対し、ランダムオラクルを使用しないモデルをスタンダードモデル (Standard Model) という。ランダムオラクルは、出力が真にランダムな値で、出力空間において一様分布であるとみなせるハッシュ関数であるとも考えることができる。実用の際には、ランダムオラクルの部分には SHA-1 や SHA-256 などの実用的なハッシュ関数を使用する。一般的に、ランダムオラクルを使う方式の方が、スタンダードモデルでの方式よりも計算コストや署名サイズなどの面で効率のよいものができる。実際に使用されている DSA [1]、RSA [2]、あるいは RSA-PSS [8] などといった多くの方式がランダムオラクルモデルでのものである (ただし DSA には厳密な証明はない)。

1999 年に Cramer らが初めてスタンダードモデルでの証明可能安全性を持つ効率的な方式 [25] を発表するまで、署名のサイズや計算コストの面においてランダムオラクルを使用する方式と張り合えるような方式は存在しなかった。

便利なツールであるランダムオラクルだが、あくまで理想的な存在であり、現実世界においては、ランダムオラクルのような真にランダムな出力を持つ関数は存在しない。また、ランダムオラクルモデルにおいて証明可能安全性を有する方式は、そのランダムオラクル以外のどのような関数に置き換えてもスタンダードモデルでは安全性を証明できなくなるものも多く存在することが分かっている [20, 49, 34, 6]。

このような理由により、最初にランダムオラクルモデルで設計して安全性を証明し、後に改良を加えてスタンダードモデルで証明できる方式を考えるということは行われることはあるものの、電子署名に限らず、新しく暗号学的な方式を考える際は、ランダムオラクルを使わなくてもよいならば、ランダムオラクルを用いない方式を目指す研究が増えている。

本研究でも、ランダムオラクルを用いない、スタンダードモデルでの電子署名方式を対称としている。

1.3　研究目的

1.1 節で述べた電子署名とハッシュ関数との関係、及び現状のハッシュ関数の衝突困難性の危殆化、そしてそして 1.2 節で述べた証明可能安全性の重要性を背景として、本研究では、ハッシュ関数に衝突困難性の仮定を必要としなくても、安全性をスタンダードモデルで証明可能であることを前提とした、効率の良い電子署名の構成を目指す。

ハッシュ関数の衝突困難性の代わりとして最もよい代替になると期待できるのは、Naor ら [48] によって導入されたターゲット衝突困難性と、Halevi ら [36] によって導入された強化ターゲット衝突困難性である。これらのハッシュ関数の安全性は、定義の上では第二原像計算困難性 (与えられた入力と、ハッシュ値が等しくなる異なる入力を見つけることが計算量的に困難であるという性質) とほぼ等価である。これらの安全性は、衝突困難性よりも原理的に破るのが難しいこともあり、有効な攻撃の報告は

少なく、例えば広く使われている SHA-1 はこれらの安全性を現実的な時間で破ることはまだ難しい信じられおり、前述の Wang らによる攻撃に代表されるような単に (意味の無い) 任意の衝突ペアを効率よく見つける手法が発見されたとしても、安全性に影響は受けない。

1.4　本研究の貢献

本研究では、電子署名として最強の安全性である、強偽造不可能性を持つと安全性の証明をできる電子署名の方式を提案した。提案方式は、スタンダードモデルの下、CDH 仮定に基づいて安全性を証明できる。提案方式ではハッシュ関数を用いて任意の長さのメッセージについて署名できるが、既に述べてきたハッシュ関数の衝突困難性の仮定は必要としない。これらの特徴を持つような電子署名は、既存の技術を用いて達成可能であるが、我々の方式は、強偽造不可能性を、スタンダードモデルで CDH 仮定に基づいて実現できるような方式の中では、署名作成、署名検証の計算コスト及び署名サイズの点において最も効率がよい方式となっている。

提案方式は 2 種類あり、1 つ目はターゲット衝突困難性 (Target Collision Resistance, TCR) をハッシュ関数に仮定した方式、2 つ目は、ターゲット衝突困難性という性質よりも強い (ただしそれでも衝突困難性よりは弱い) 性質である、強化ターゲット衝突困難性 (Enhanced TCR, eTCR) [36] を仮定した署名方式である。いずれの方式も、Boneh ら [16] による署名方式 (BSW 署名) 及び、Waters [62] による方式 (Waters 署名) を元にした構成法になっている。

1.5　本稿の構成

以下、2 章では、3 章以降の提案方式の説明のときに必要になる計算問題の仮定やハッシュ関数の安全性などの諸定義を概要と共に説明する。3 章では、提案する 2 つの電子署名方式と、その安全性証明を示す。4 章では、本研究の関連研究を紹介する。5 章では、提案方式と既存の強偽造不可能性を持つ方式間での比較を行う。6 章は本稿のまとめである。

Chapter 2　諸定義

本章では、3 章以降で必要となる電子署名とその安全性 (2.1 節)、計算的困難性の仮定 (2.2 節)、ハッシュ関数とその安全性 (2.3 節)、そして双線形写像 (2.4 節) について各々の概要と定義を振り返る。

本稿での記号の定義　本稿では、$x \leftarrow y$ と書くとき、y が集合ならばそこから一様ランダムに要素を取り出し x に代入、y が演算ならば結果を x に代入、y がアルゴリズムまたは関数ならば x を出力する操作を表す。また、$|z|$ は、z が集合ならば要素数を、z がある集合の要素ならばビット長を表す。"$a||b$" は a と b の連結を表す。

2.1　電子署名

電子署名 Σ は以下の 3 つのアルゴリズムからなる。

鍵生成 KeyGen: 確率的アルゴリズム。1^κ ($\kappa \in \mathcal{N}$ をセキュリティパラメータという) を入力とし、署名鍵 sk、検証鍵 vk を出力する。
この操作を $(sk, vk) \leftarrow \mathsf{KeyGen}(1^\kappa)$ と書く。

署名 Sign: 確率的または決定的アルゴリズム。署名鍵 sk、メッセージ $m \in \mathcal{M}$ を入力とし、メッセージ m に対する署名 $\sigma \in \mathcal{S}$ を出力する。
この操作を $\sigma \leftarrow \mathsf{Sign}(sk, m)$ と書く。

検証 Verify: 決定的アルゴリズム。検証鍵 vk、メッセージ m、及びそれに対する署名 σ を入力し、σ が m に対する正しい署名ならば `accept`、そうでないならば `reject` を出力する。
この操作を $\{\texttt{accept}, \texttt{reject}\} \leftarrow \mathsf{Verify}(vk, m, \sigma)$ と書く。

ただし、$\mathcal{M}$、$\mathcal{S}$ はそれぞれ電子署名 Σ のメッセージ空間、署名空間である。

正当性　全ての $\kappa \in \mathcal{N}$、KeyGen から出力された全ての鍵ペア (sk, vk)、全ての $m \in \mathcal{M}$ に対し、以下を満たさなければならない。

$$\mathsf{Verify}(vk, m, \mathsf{Sign}(sk, m)) = \texttt{accept}$$

2.1.1　EUF-CMA 安全性

電子署名の適応的選択文書攻撃に対する存在的偽造不可能性 (Existential Unforgeability against Adaptive Chosen Message Attacks, EUF-CMA 安全性) は、Goldwasser ら [35]

によって定式化された。この定義は電子署名の安全性の枠組みの中で、最も広く用いられる安全性の定義である。単に偽造不可能性とも呼ばれる[1]。本稿では通常 EUF-CMA 安全性と書くが、文脈によっては偽造不可能性と書くこともある。

EUF-CMA 安全性は、以下の攻撃者 $\mathcal{A}$ と SEUF-CMA チャレンジャー $\mathcal{C}$ 間の EUF-CMA ゲームを利用して定義される。

Setup. $\mathcal{C}$ は $\mathsf{KeyGen}(1^\kappa)$ を実行し、出力された検証鍵 vk を $\mathcal{A}$ に渡す。同時に出力される署名鍵 sk は保持しておく。

Queries. $\mathcal{A}$ は $\mathcal{C}$ に対し、最大 q 回の署名クエリ $m_1, m_2, \ldots, m_q$ を発行することができる。$\mathcal{C}$ はそれぞれの署名クエリ m_i に対し、m_i についての正しい署名 $\sigma_i \leftarrow \mathsf{Sign}(sk, m_i)$ を計算し、$\mathcal{A}$ に渡す。$\mathcal{A}$ は署名クエリ m_i を $\mathcal{A}$ 自身の過去に発行した署名クエリ $(m_1, \ldots, m_{i-1})$ 及びそれに対する $\mathcal{C}$ の応答 $(\sigma_1, \ldots, \sigma_{i-1})$ に依存して適応的に発行することができる。

Output. 最終的に、$\mathcal{A}$ は $\mathcal{C}$ に対し、偽造として $(\hat{m}, \hat{\sigma})$ のペアを出力する。$\mathsf{Verify}(vk, \hat{m}, \hat{\sigma}) = \mathtt{accept}$、かつ全ての $i \in \{1, \ldots, q\}$ について $\hat{m} \neq m_i$ を満たすならば、$\mathcal{A}$ の勝利となる。

ここで、ある電子署名方式 Σ に対する攻撃者 $\mathcal{A}$ の EUF-CMA アドバンテージ $\mathsf{Adv}^{\text{EUF-CMA}}_{\Sigma,\mathcal{A}}$ を $\mathcal{A}$ が EUF-CMA ゲームにおいて勝利する確率と定義する。

定義 2.1. 署名クエリを最大 q 回発行する動作時間 t 以下の全てのアルゴリズム $\mathcal{A}$ に対し $\mathsf{Adv}^{\text{EUF-CMA}}_{\Sigma,\mathcal{A}} \leq \epsilon$ を満たすとき、電子署名 Σ は (t, q, ϵ)-*EUF-CMA* 安全性を満たすという。また、ϵ が無視できる値のとき、単に Σ は *EUF-CMA* 安全性を満たす、あるいは偽造不可能性を持つという。

2.1.2　SEUF-CMA 安全性

EUF-CMA 安全性よりも強い安全性である、電子署名の適応的選択文書攻撃に対する強存在的偽造不可能性 (Strong Existential Unforgeability against Adaptive Chosen Message Attacks, SEUF-CMA 安全性) は、An ら [3] によって提案された。この定義は電子署名の安全性の枠組みの中で、現在最も強い安全性の定義となっている。単に強偽造不可能性とも呼ばれる。あらゆる方式について、SEUF-CMA 安全であるならば、必ず EUF-CMA 安全性も満たしている。また、署名アルゴリズムが決定的アルゴリズムである場合、EUF-CMA 安全性ならば必ず SEUF-CMA 安全となる。

強偽造不可能性を持つ署名方式は、電子署名としての通常の用途以外に、他の暗号学的な方式の構成要素として、利用されることがある (例えば、最強の安全性を持つ公開鍵暗号の構成 [21]、グループ署名の構成 [4, 11] など) これらの方式では、EUF-CMA 安全性のみしか達成していない場合、方式の安全性を示すのには不十分である。

[1] 強偽造不可能性との違いを明確にするために弱偽造不可能性と呼ばれることもある。

また、EUF-CMA 安全性しか示さないような署名方式を SEUF-CMA 安全性を持つ方式へと安全性を強める一般的な方式がいくつか提案されている。これについては関連研究において紹介している (4.2 節)。

SEUF-CMA 安全性は、以下の攻撃者 $\mathcal{A}$ と SEUF-CMA チャレンジャー $\mathcal{C}$ 間の SEUF-CMA ゲームを利用して定義される。

Setup と Queries. EUF-CMA ゲームと同様。

Output. 最終的に、$\mathcal{A}$ は $\mathcal{C}$ に対し、偽造として $(\hat{m}, \hat{\sigma})$ のペアを出力する。$\mathsf{Verify}(vk, \hat{m}, \hat{\sigma}) = \mathtt{accept}$、かつ全ての $i \in \{1, \ldots, q\}$ について $(\hat{m}, \hat{\sigma}) \neq (m_i, \sigma_i)$ を満たすならば、$\mathcal{A}$ の勝利となる。

ここで、ある電子署名方式 Σ に対する攻撃者 $\mathcal{A}$ の SEUF-CMA アドバンテージ $\mathsf{Adv}^{\text{SEUF-CMA}}_{\Sigma,\mathcal{A}}$ を $\mathcal{A}$ が SEUF-CMA ゲームにおいて勝利する確率と定義する。

定義 2.2. 署名クエリを最大 q 回発行する動作時間 t 以下の全てのアルゴリズム $\mathcal{A}$ に対し $\mathsf{Adv}^{\text{SEUF-CMA}}_{\Sigma,\mathcal{A}} \leq \epsilon$ を満たすとき、電子署名 Σ は (t, q, ϵ)-*SEUF-CMA* 安全性を満たすという。また、ϵ が無視できる値のとき、単に Σ は *SEUF-CMA* 安全性を満たす、あるいは強偽造不可能性を持つという。

2.2　困難性の仮定

本節では、3 節での各方式の安全性の証明の際に利用する困難性の仮定の定義を説明する。巡回群における CDH(Computational Diffie-Hellman) 仮定 (2.2.1 節) と DL(Discrete Logarithm, 離散対数) 仮定 (2.2.2 節) は、共に公開鍵暗号と電子署名の歴史と同程度以上の歴史を持つよく知られた困難性の仮定である。

特に CDH 問題は、その名前の通り、Diffie と Hellman [29] による事前の秘密通信による知識の共有一切無しの状態からの鍵共有方式の安全性の根拠として用いられた問題である。これらの仮定の強弱として、DL 仮定よりも CDH 仮定の方が強い仮定であるということが言える。逆を言えば、DL 問題を解けるアルゴリズムを用いて CDH 問題を解けるということになる。これらの仮定は証明可能安全性の枠組みで用いられる具体的な計算量的困難性の仮定としては基本的な仮定であり、実社会でも利用可能そうな効率を持つような暗号学的な方式の安全性をこれらの問題の困難性へと帰着できることは少ない。

実際、電子署名の場合は複数の方式の安全性が DL 仮定や CDH 問題の困難性へと帰着できることが示されているが、公開鍵暗号方式では、ElGamal 暗号 [30] や Cramer-Shoup 暗号 [24] などのよく知られた方式は CDH 仮定よりも真に強い仮定である DDH(Ddcisional DH, 決定 DH) 仮定に基づいた証明しか示されていない。

2.2.1　CDH 仮定

位数 p の巡回群 $\mathbb{G}$ における Computational Diffie-Hellman (CDH, 計算 DH) 問題は以下の様に定義される: "g、g^a、$g^b \in \mathbb{G}$ を与えられて、 $g^{ab} \in \mathbb{G}$ を求めよ。ただし g はランダムに選ばれた $\mathbb{G}$ の生成元であり、a と b はランダムに選ばれた $\mathbb{Z}_p$ の要素である。"

ここで、攻撃者 $\mathcal{A}$ の $\mathbb{G}$ における CDH 問題に対するアドバンテージ $\mathsf{Adv}^{\mathrm{CDH}}_{\mathbb{G},\mathcal{A}}$ を、$\mathcal{A}$ が CDH 問題を与えられて解ける確率と定義する。

定義 2.3. 動作時間 t 以下の全てのアルゴリズム $\mathcal{A}$ に対し、$\mathsf{Adv}^{\mathrm{CDH}}_{\mathbb{G},\mathcal{A}} \leq \epsilon$ を満たす場合、$\mathbb{G}$ において (t,ϵ)-*CDH* 仮定が成り立つという。また、ϵ が無視できる値のとき、単に $\mathbb{G}$ において *CDH* 仮定が成り立つという。

2.2.2　DL 仮定

位数 p の巡回群 $\mathbb{G}$ における離散対数 (Discrete Logarithm、DL) 問題は以下の様に定義される: "g、$g^a \in \mathbb{G}$ を与えられて、 $a \in \mathbb{Z}_p$ を求めよ。ただし g はランダムに選ばれた $\mathbb{G}$ の生成元であり、a はランダムに選ばれた $\mathbb{Z}_p$ の要素である。"

ここで、攻撃者 $\mathcal{A}$ の $\mathbb{G}$ における DL 問題に対するアドバンテージ $\mathsf{Adv}^{\mathrm{DL}}_{\mathbb{G},\mathcal{A}}$ を、$\mathcal{A}$ が DL 問題を与えられて解ける確率と定義する。

定義 2.4. 動作時間 t 以下の全てのアルゴリズム $\mathcal{A}$ に対し、$\mathsf{Adv}^{\mathrm{DL}}_{\mathbb{G},\mathcal{A}} \leq \epsilon$ を満たす場合、$\mathbb{G}$ において (t,ϵ)-*DL* 仮定が成り立つという。また、ϵ が無視できる値のとき、単に $\mathbb{G}$ において *DL* 仮定が成り立つという。

2.3　ハッシュ関数

$H : \mathcal{K} \times \mathcal{M}_{in} \to \mathcal{M}_{out}$ を鍵付きハッシュ関数とする。ただし、$\mathcal{K}$、$\mathcal{M}_{in}$、$\mathcal{M}_{out}$ はそれぞれ、H のハッシュ鍵空間、入力空間、出力空間である。

鍵付きハッシュ関数の考え方　証明可能安全性の枠組みでは、ハッシュ関数は、安全性を定義する場合は通常、ハッシュ鍵によって決定される鍵付きハッシュ関数、あるいは同義であるが、インデックスによって決定されるハッシュ関数族 (Hash Family) として扱われる。本稿では前者を用いている。

このような回りくどい定義を用いる理由は、鍵無しのハッシュ関数の定義の安全性、例えば衝突困難性を考えれば分かる。鍵無しハッシュ関数 $H : \mathcal{M}_{in} \to \mathcal{M}_{out}$ (簡単のため $|\mathcal{M}_{in}| > |\mathcal{M}_{out}|$ を考える) の衝突困難性は次の様になる: "ハッシュ関数においてハッシュ値が等しくなる任意の入力のペア (衝突) を無視できない値で出力するアルゴリズムが存在しない"(正式な定義は 2.3.1 節)。この定義を満たすようなハッシュ関数は存在しない。なぜならば、$|\mathcal{M}_{in}| > |\mathcal{M}_{out}|$ のため、鳩の巣原理により必ず衝突を起こすペアが存在し、人間がそのような衝突を起こすペアを発見できるかどうかに関わらず、衝突困難性を破るアルゴリズムとして、そのような衝突を出力するという

1 行の命令からなるアルゴリズムは、確率 1 で衝突困難性を破れるからである。同様の議論が衝突困難性以外の安全性においても考えられる。

このような自明な破綻を防ぐために鍵付きハッシュ関数を定義し、衝突困難性は与えられたハッシュ鍵の下での衝突を無視できない確率で見つけるアルゴリズムが存在しないという定義が用いられる。実際、衝突困難性を最初に定式化した Damgård [28] もハッシュ関数族を用いている。鍵無しのハッシュ関数の安全性を別の方法で定義して議論した研究 [54] も存在するが、本稿では鍵付きハッシュ関数を考えることにする。以下では明示的に書かない場合、ハッシュ関数は全て鍵付きハッシュ関数を考えているものとする。

2.3.1　衝突困難ハッシュ関数

衝突困難ハッシュ関数 (Collision Resistant Hash Function, CRHF) は Damgård [28] によって定式化された。一方向性とならんで、ハッシュ関数が満たすべき性質としてよく仮定される安全性である。

Simon [58] は、一方向置換 (One-Way Permutation) をどのように組み合わせても衝突困難ハッシュ関数は構成できないことを証明した。従って、一方向関数という、証明可能安全性の枠組みの中で用いられる仮定の中で最も弱い仮定である、一方向関数 (One-Way Function) からは構成できない。現在知られている衝突困難ハッシュ関数が存在する最も弱い仮定は、クローフリー置換族 (Craw Free Permutation Family) が存在することである [28]。

鍵付きハッシュ関数の衝突困難性は、以下の攻撃者 $\mathcal{A}$ と CR チャレンジャー $\mathcal{C}$ 間の CR ゲームを用いて定義される。

Step 1. $\mathcal{C}$ は $k \in \mathcal{K}$ を一様ランダムに選び、$\mathcal{A}$ に渡す。

Step 2. $\mathcal{A}$ は $m_1, m_2 \in \mathcal{M}_{in}$ のペアを出力する。$H_k(m_1) = H_k(m_2)$ かつ $m_1 \neq m_2$ ならば、$\mathcal{A}$ の勝ちとなる。

攻撃者 $\mathcal{A}$ の衝突困難ハッシュ関数 H に対するアドバンテージ $\mathsf{Adv}^{\mathrm{CR}}_{H,\mathcal{A}}$ を、$\mathcal{A}$ が CR ゲームにおいて勝利する確率と定義する。

定義 2.5. 動作時間 t 以下の全てのアルゴリズム $\mathcal{A}$ に対し、$\mathsf{Adv}^{\mathrm{CR}}_{H,\mathcal{A}} \leq \epsilon$ を満たすとき、H を (t, ϵ)-*CRHF* であるという。また、ϵ が無視できる値のとき、H を単に衝突困難ハッシュ関数であるという。

2.3.2　ターゲット衝突困難ハッシュ関数

ターゲット衝突困難ハッシュ関数 (Target CRHF, TCRHF) は、最初、Naor と Yung [48] により、汎用一方向ハッシュ関数 (Universal One-Way Hash Function, UOWHF) として導入され、後に、より性質を示す表現として、Bellare と Rogaway [9] は本節での名前を用いた。

ターゲット衝突困難性は、鍵無しの場合の第二原像計算困難性 (Second Preimage Resistance, SPR) と非常に近い安全性である。汎用一方向性との違いを考え、下記の定義よりやや弱い定義の Cramer と Shoup [26] による定義 (鍵付きハッシュ関数での SPR) を指してターゲット衝突困難性と呼ぶ論文もあるが、本稿では Bellare と Rogaway による定義を用いている。

Rompel による証明 [55, 38] によって、ターゲット衝突困難ハッシュ関数は、一方向関数だけから構成できることが示されている。

鍵付きハッシュ関数のターゲット衝突困難性は、以下の攻撃者 $\mathcal{A}$ と TCR チャレンジャー $\mathcal{C}$ 間の TCR ゲームを用いて定義される。

Step 1. $\mathcal{A}$ は任意の $m_1 \in \mathcal{M}_{in}$ を出力する。

Step 2. $\mathcal{C}$ は $k \in \mathcal{K}$ を一様ランダムに選び、$\mathcal{A}$ に渡す。

Step 3. $\mathcal{A}$ は $m_2 \in \mathcal{M}_{in}$ を出力する。$H_k(m_1) = H_k(m_2)$ かつ $m_2 \neq m_1$ ならば、$\mathcal{A}$ の勝ちとなる。

攻撃者 $\mathcal{A}$ のターゲット衝突困難ハッシュ関数 H に対するアドバンテージ $\mathsf{Adv}^{\mathrm{TCR}}_{H,\mathcal{A}}$ を、$\mathcal{A}$ が TCR ゲームにおいて勝利する確率と定義する。

定義 2.6. 動作時間 t 以下の全てのアルゴリズム $\mathcal{A}$ に対し、$\mathsf{Adv}^{\mathrm{TCR}}_{H,\mathcal{A}} \leq \epsilon$ を満たすとき、H を (t,ϵ)-*TCRHF* であるという。また、ϵ が無視できる値のとき、H を単にターゲット衝突困難ハッシュ関数であるという。

2.3.3　強化ターゲット衝突困難ハッシュ関数

強化ターゲット衝突困難ハッシュ関数 (Enhanced TCRHF、 eTCRHF) は、Halevi と Krawcyzk [36] によって導入された。名前の通り、この安全性はターゲット衝突困難性よりも強い安全性を表している。

鍵付きハッシュ関数のターゲット衝突困難性は、以下の攻撃者 $\mathcal{A}$ と eTCR チャレンジャー $\mathcal{C}$ 間の TCR ゲームを用いて定義される。

Step 1. $\mathcal{A}$ は任意の $m_1 \in \mathcal{M}_{in}$ を出力する。

Step 2. $\mathcal{C}$ は $k \in \mathcal{K}$ を一様ランダムに選び、$\mathcal{A}$ に渡す。

Step 3. $\mathcal{A}$ はハッシュ鍵 $k' \in \mathcal{K}$ 及び、$m_2 \in \mathcal{M}_{in}$ を出力する。$H_k(m_1) = H_{k'}(m_2)$ かつ $(k, m_1) \neq (k', m_2)$ ならば、$\mathcal{A}$ の勝ちとなる。

攻撃者 $\mathcal{A}$ の強化ターゲット衝突困難ハッシュ関数 H に対するアドバンテージ $\mathsf{Adv}^{\mathrm{eTCR}}_{H,\mathcal{A}}$ を、$\mathcal{A}$ が eTCR ゲームにおいて勝利する確率と定義する。

定義 2.7. 動作時間 t 以下の全てのアルゴリズム $\mathcal{A}$ に対し、$\mathsf{Adv}^{\mathrm{eTCR}}_{H,\mathcal{A}} \leq \epsilon$ を満たすとき、H を (t,ϵ)-*eTCRHF* であるという。また、ϵ が無視できる値のとき、H を単に強化ターゲット衝突困難ハッシュ関数であるという。

2.3.4　各安全性の関係

前節までに紹介したハッシュ関数の安全性の定義間には強弱の関係がある。

衝突困難性は、強化ターゲット衝突困難性よりも強い安全性であり、強化ターゲット衝突困難性はターゲット衝突困難性よりも強い安全性である。逆に考えれば、衝突困難性は強化ターゲット衝突困難性よりも破るのが容易であり、強化ターゲット衝突困難性はターゲット衝突困難性よりも破るのが容易である。従って前者ほど達成が難しく破るのは容易である。これらの違いは、証明の際に大きく現れる。

2.3.5　一般的な攻撃に対する安全性

ハッシュ関数は通常大きな空間から小さな空間への写像であるため、具体的なアルゴリズムや構成とは関係なく、一般的な攻撃法が存在する。それは、入力空間に対するランダムな探索をかけることである。出力空間が n ビットのときの、それぞれの安全性を持つハッシュ関数に対する一般的な攻撃に必要な計算量のオーダーを図 2.1 に示す。

強化ターゲット衝突困難性とターゲット衝突困難性は、攻撃者は衝突を見つけるべき入力ペアのうちの片方を固定されているため、2^n のオーダーのハッシュ関数の演算が必要になる。

しかし、衝突困難性を破るためには、衝突ペア探索の際両方を攻撃者が選ぶことができるため、誕生日のパラドックスによって、$2^{n/2}$ のオーダーで衝突が見つかってしまう。この事実を利用した攻撃は誕生日攻撃と呼ばれる。このため、n ビット安全性を得るためには最低でも $2n$ ビットの出力長が必要である。

上記は一般的な攻撃法であり、実際は実装の際のアルゴリズムの特徴をついた攻撃の研究により、計算に必要な回数はこれらより減らされてしまう。例えば、SHA-1 は出力が 160 ビットのハッシュ関数であり、近年まであまり有効な攻撃方法が存在しなかった。しかし、2005 年に Wang ら [61] によって、長く 2^{80} 程度と信じられていた計算回数を 2^{63} 程度に減らす攻撃を発見したと報告した。この報告により、事実上、SHA-1 の衝突困難性は破れたと考えられている。

表 2.1: n ビット出力を持つハッシュ関数を一般的な攻撃で破るための計算回数のオーダー

安全性	CRHF	TCRHF	eTCRHF
計算回数のオーダー	$2^{n/2}$	2^n	2^n

2.4　双線形写像

暗号学において双線形写像は、初め楕円曲線上の巡回群の離散対数系の仮定に基づいた方式を攻撃するためのものとして導入された (Menezes ら [44, 45] による MOV 還元

法、Frey ら [33] による FR 還元法など)。

しかし、2000 年以降、境、大岸、笠原 [56]、及び Boneh と Franklin [12, 13] による ID ベース暗号 [57] の効率的な構成法の提案の中で、方式実現のための構成要素として用いられたことを機に、現在まで様々な方式の構成に利用されている。

具体的な計算方法は、Weil ペアリングや Tate ペアリングと呼ばれる方法を用いるが、本稿では具体的なアルゴリズムとして何が用いられるかには立ち入らない。

$\mathbb{G}$ 及び $\mathbb{G}_T$ をそれぞれ位数が素数 p の巡回群とし、g を $\mathbb{G}$ における生成元とする。以下の性質を全て満たす写像 e: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ 及び群 $\mathbb{G}_T$ が存在する場合、$(\mathbb{G}, \mathbb{G}_T)$ を双線形群 (Bilinear Groups) であるという。また、写像 e を双線形写像 (Bilinear Map) という。

- 双線形性: 全ての $u, v \in \mathbb{G}$ 及び全ての $a, b \in \mathbb{Z}$ に対し、$e(u^a, v^b) = e(u, v)^{ab}$ が成り立つ。
- 非退化性: 全ての $g \in \mathbb{G}$ に対し、$e(g, g) \neq 1$ が成り立つ (ただしここでの 1 は $\mathbb{G}_T$ の単位元)。
- 効率的に計算可能: 全ての $u, v \in \mathbb{G}$ に対し、効率的に $e(u, v)$ を計算するアルゴリズムが存在する。

上記は定義域の二つの群として対称な群を考えているが、非対称な双線形写像 $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ (ただし、$|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T| = p$、$\mathbb{G}_1 \neq \mathbb{G}_2$) を考えることもできる。実際に何かの方式の中で利用される際、対称な双線形写像を非対称な双線形写像へと置き換えることができる方式が多い。

非対称な双線形写像を用いる最も大きな利点は、Barreto ら [5] による性質の良い楕円曲線を使える点にある。もしそれらが使用されれば、$\mathbb{G}_1$ の要素を表す際のビット長が 160 ビット程度と小さくできる。ただし、$\mathbb{G}_2$ の要素のビット長は $\mathbb{G}_1$ の要素の数倍に長くなってしまう。詳細は [51, 39] を参照されたい。

非対称な双線形写像を用いる場合には、対称なものを用いた場合と比べて安全性の根拠となる問題が若干変化する場合がある (例えば CDH 問題の困難性から、co-CDH 問題 [15] の困難性へなど)。

Chapter 3　衝突困難ハッシュ関数を用いない電子署名方式

本章では、本研究のでの提案手法である、衝突困難性ハッシュ関数を用いない電子署名の構成法を示す。提案する署名方式は2つあり、共にスタンダードモデルの下、CDH仮定に基づきSEUF-CMA安全性を満たすことが証明できる。

まず、3.1節で、提案手法の基となる方式として、Watersによる署名方式(Waters署名) [62]とBoneh、Shen、Watersによる署名方式(BSW署名) [16]、そしてその構成上の問題点について触れる。そして3.2節と3.3節で、TCRHFを用いた電子署名方式、eTCRHFを用いた電子署名方式についてそれぞれ述べる。

CDH仮定でSEUF-CMA安全性を証明可能で、しかも衝突困難ハッシュ関数を用いないでよい署名方式は、既に知られている方法だけを用いて達成することが可能である。例えば、スタンダードモデルの下で、CDH仮定に基づきEUF-CMA安全性を達成しているWaters署名 [62]に対し、署名可能なサイズを伸ばすために、ターゲット衝突困難性を使ったHash-and-Signのような構成 [48]と、一般的にEUF-CMA安全性を持つ署名方式を一般的にSEUF-CMA安全性を持つ方法へと変換する方法 [60, 59, 37]の組み合わせればよい。しかし、このような一般的な構成では署名サイズや署名作成、検証にかかる計算コストが大きくなる。しかし、本研究で提案した2つの方法は、現在CDH仮定でSEUF-CMA安全性を証明可能で最も効率がよいBSW署名 [16]と同等の効率を持つ。

3.1　提案方式の基礎となる2つの署名方式

3.1.1　Waters署名

2005年、Waters [62]はスタンダードモデルの下、CDH仮定に基づきEUF-CMA安全性を証明できる方式を示した。その構成法を図3.1に示す。

この方式以前のスタンダードモデルので証明可能安全性を持つ方式は、強RSA仮定やSDH仮定など安全性の証明にやや強い仮定が必要であるが、この方式は、スタンダードモデルにおいて、CDH問題の困難性という基礎的な問題に帰着できる方式で、実用に耐えうる様な効率性を持つ初めての方式である。ただし、SEUF-CMA安全性は満たさない。署名のサイズは、現在広く使用されているDSA [1]などと同様、最も効率よく方式を実装すれば、およそ320ビットとできる。

ただし、署名できるメッセージのサイズと証明の帰着効率にトレードオフがあり、署名できるメッセージのサイズに比例して公開鍵のパラメータが大きくなってしまうという性質がある。

構成要素：
$\mathbb{G}, \mathbb{G}_T$：位数が十分大きな素数 p の巡回群
$e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$：双線形写像

$\mathsf{KeyGen}(1^\kappa)$：	$\mathsf{Sign}(sk, m)$：
$g \leftarrow \mathbb{G}$; $\alpha \leftarrow \mathbb{Z}_p$; $g_1 = g^\alpha$	$r \leftarrow \mathbb{Z}_p$; $\sigma_2 \leftarrow g^r$
$g_2, h,, u', u_1, \ldots, u_n \leftarrow \mathbb{G}$	Parse m as $m_1 m_2 \ldots m_n$
$U \leftarrow (u_1, \ldots, u_n)$	(each of m_i is the i-th bit of m).
$sk \leftarrow g_2^\alpha, vk \leftarrow (g, g_1, g_2, h, u', U)$	$\sigma_1 \leftarrow g_2^\alpha \cdot (u' \prod_{i=1}^n u_i^{m_i})^r$
Output (sk, vk).	Output $\sigma \leftarrow (\sigma_1, \sigma_2)$.

$\mathsf{Verify}(vk, m, \sigma)$：
Parse σ as (σ_1, σ_2).
Parse m as $m_1 m_2 \ldots m_n$ (each of m_i is the i-th bit of m).
Check $e(\sigma_1, g) \stackrel{?}{=} e(\sigma_2, u' \prod_{i=1}^n u_i^{m_i}) \cdot e(g_1, g_2)$.
Output `accept` if this holds. Otherwise output `reject`.

図 3.1: Waters 署名

Waters 署名は、同論文内で提案された初めてのスタンダードモデルで高い安全性を証明可能な ID ベース暗号 [57, 12] に対し、Noar による “任意の ID ベース暗号から電子署名へを構成することができる” という考察 ([12, 27] でそのことについて言及されている) を適用して得られた署名方式である。次節で取り上げる Boneh-Shen-Waters 署名 [16] の構成の基礎となっている他、他の様々な機能付き署名の構成の基礎として用いられている [43, 42, 53]。

以下は Waters 署名の安全性についての記述である。署名可能なメッセージのビット長 n は、ハッシュ関数を用いるなどして大きくできるが、ここではその場合を考えていない。

定理 3.1. $\mathbb{G}$ において $(t, \epsilon/8(n+1)q)$-*CDH* 仮定が成り立つとき、*Waters* 署名 (図 *3.1*) は (t, q, ϵ)-*EUF-CMA* 安全性を持つ。ただし n は署名可能なメッセージのビット長である。

3.1.2　BSW 署名

2006 年、Boneh、Shen、Waters [16] は、スタンダードモデルの下、CDH 仮定に基づき SEUF-CMA 安全性を証明できる方式 (以下 BSW 署名) を示した。

著者らが BSW 署名の構成、及び安全性の証明を与える手順は以下のような流れになっている。

1. 署名方式の署名アルゴリズムの構成に特徴を持つ “*partitioned*” というクラスを導入する。

2. EUF-CMA 安全性を持つ *partitioned* 署名を、SEUF-CMA 安全性を持つ署名方式へと変換する一般的構成法 (BSW 変換) の構成を示す。BSW 変換の安全性証明には、変換前の署名方式の EUF-CMA 安全性、DL 仮定、ハッシュ関数の衝突困難性が必要である。

3. 既に EUF-CMA 安全性が示されている Waters 署名 [62] が *partitioned* クラスに当てはまることを証明し、Waters 署名に BSW 変換を適用する。

[16] で導入された *partitioned* 署名の定義は以下である。

定義 3.1. 電子署名方式 Σ が以下の *2* つの性質を持つ場合、Σ を *"partitioned"* 署名であるという。

- **性質 1:** 署名アルゴリズム Sign が以下の *2* つの決定的アルゴリズムが、S_1 と S_2 に分割可能であり、メッセージ m についての署名は、署名鍵 sk を用いて以下の様に記述される。

  1. $r \in \mathcal{R}$ を一様ランダムに選ぶ。
  2. $\sigma_1 \leftarrow \mathsf{S}_1(sk, m, r)$ と $\sigma_2 \leftarrow \mathsf{S}_2(sk, r)$ を計算する。
  3. $\sigma = (\sigma_1, \sigma_2) \quad \in \mathcal{S}_1 \times \mathcal{S}_2$ を出力する。

- **性質 2:** $\sigma_2 \in \mathcal{S}_2$ 及び $m \in \mathcal{M}$ を与えられると、(σ_1, σ_2) が検証鍵 vk の下に正しい m の署名となるような $\sigma_1 \in \mathcal{S}_1$ は高々 *1* つしか存在しない。

ただし、$\mathcal{R}$ は、アルゴリズム S_1 と S_2 のための乱数空間、$\mathcal{M}$ は Σ のメッセージ空間、そして $\mathcal{S}_1$、$\mathcal{S}_2$ はそれぞれアルゴリズム S_1、S_2 の出力空間である。

BSW 変換の構成法を図 3.2 に示す。BSW 変換は、EUF-CMA 安全性を SEUF-CMA 安全性へと強めることができるという意味で、非常に便利である。著者らの研究の後、BSW 変換では必要である *partitioned* という制約を取り払い、任意の EUF-CMA 安全性を持つ署名方式を SEUF-CMA 安全性を持つ方式を得ることのできる変換の研究が 3 つ続いている [60, 59, 37]。これらについては関連研究で紹介している (4.2 節)。安全性についての記述は以下である。

定理 3.2. 以下の条件が成り立つとき、*BSW* 変換 *(図 3.2)* を適用した後の方式 Σ_{new} は (t, q, ϵ)-*SEUF-CMA* 安全性を満たす電子署名方式である。

- 構成要素の電子署名方式 Σ は $(t, q, \epsilon/3)$-*EUF-CMA* 安全性を満たす *partitioned* 署名である。

- $\mathbb{G}$ において $(t, \epsilon/3)$-*DL* 仮定が成り立つ。

- H は $(t, \epsilon/3)$-*CRHF* である。

<table>
<tr><td colspan="2">構成要素：
$\mathsf{S}_1, \mathsf{S}_2$：変換前の署名方式 Σ の partitioned の性質によって
分割された 2 つの署名アルゴリズム
$\mathbb{G}$：位数が十分大きな素数 p の巡回群
$H : \mathcal{K} \times \{0,1\}^* \to \mathbb{Z}_p$：CRHF</td></tr>
<tr><td>$\mathsf{KeyGen}_{new}(1^\kappa)$：
$(sk, vk) \leftarrow \mathsf{KeyGen}(1^\kappa)$
$g, h \leftarrow \mathbb{G}$; $k \leftarrow \mathcal{K}$
$SK \leftarrow (sk)$, $VK \leftarrow (vk, g, h, k)$
Output (SK, VK).</td><td>$\mathsf{Sign}_{new}(SK, M)$：
$s \leftarrow \mathbb{Z}_p$; $r \leftarrow \mathcal{R}$; $\sigma_2 \leftarrow \mathsf{S}_2(sk, r)$
$t \leftarrow H_k(\sigma_2 || M)$; $m \leftarrow g^t h^s$
$\sigma_1 \leftarrow \mathsf{S}_1(sk, m, r)$
Output $\sigma \leftarrow (\sigma_1, \sigma_2, s)$.</td></tr>
<tr><td colspan="2">$\mathsf{Verify}_{new}(VK, M, \sigma)$：
Parse σ as (σ_1, σ_2, s).
$t \leftarrow H_k(\sigma_2 || M)$; $m \leftarrow g^t h^s$
Output <code>accept</code> if $\mathsf{Verify}(vk, m, (\sigma_1, \sigma_2)) =$ <code>accept</code>. Otherwise output <code>reject</code>.</td></tr>
</table>

図 3.2: BSW 変換の構成

BSW 署名の構成を図 3.3 に示す。BSW 署名は、スタンダードモデルにおいて、SEUF-CMA 安全性を CDH 問題の困難性という基礎的な問題に帰着できる方式で、実用に耐えうる様な効率性を持つ初めての方式である。署名のサイズは、最も効率よく方式を実装すれば、およそ 480 ビットとできる。

BSW 署名の安全性の記述は以下である。

系 3.1. 以下の条件が成り立つとき、*BSW* 署名 *(*図 *3.3* は (t, q, ϵ)*-SEUF-CMA* 安全性を持つ。ただし n は $\mathbb{G}$ の要素をビットで表したときのビット長である。

- $\mathbb{G}$ において $(t, \epsilon/24(n+1)q)$*-CDH* 仮定が成り立つ。
- H は $(t, \epsilon/3)$*-CRHF* である。

3.1.3　2 つの方式の問題点

Waters 署名は SEUF-CMA 安全性を満たさないことについては既にふれた。実際、$\mathsf{Verify}(vk, m, \sigma)$ となる m と署名

$$\sigma = (\sigma_1, \sigma_2) = ((sk) \cdot (u' \prod_{i=1}^{n} u_i^{m_i})^r, g^r)$$

を見た後、別の乱数 r' を用いて $R_1 = (u' \prod_{i=1}^{n} u_i^{m_i})^{r'}$ と $R_2 = g^{r'}$ を計算し、

$$\sigma' = (\sigma_1 \cdot R_1, \sigma_2 \cdot R_2) = ((sk) \cdot (u' \prod_{i=1}^{n} u_i^{m_i})^{r+r'}, g^{r+r'})$$

構成要素：
$\mathbb{G}$, $\mathbb{G}_T$：位数が十分大きな素数 p の巡回群
$e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$：双線形写像
$H : \mathcal{K} \times \{0,1\}^* \to \mathbb{Z}_p$：CRHF

$\mathsf{KeyGen}(1^\kappa)$：	$\mathsf{Sign}(SK, M)$：
$g \leftarrow \mathbb{G}$; $\alpha \leftarrow \mathbb{Z}_p$; $g_1 \leftarrow g^\alpha$	$s, r \leftarrow \mathbb{Z}_p$; $\sigma_2 \leftarrow g^r$
g_2, h,, u', $u_1, \ldots, u_n \leftarrow \mathbb{G}$	$t \leftarrow H_k(\sigma_2 \| M)$; $m \leftarrow g^t h^s$
$U \leftarrow (u_1, \ldots, u_n)$	Parse m as $m_1 m_2 \ldots m_n$
$k \leftarrow \mathcal{K}$	(each of m_i is the i-th bit of m).
$SK \leftarrow g_2^\alpha, VK \leftarrow (g, g_1, g_2, h, u', U, k)$	$\sigma_1 \leftarrow g_2^\alpha \cdot (u' \prod_{i=1}^n u_i^{m_i})^r$
Output (SK, VK).	Output $\sigma \leftarrow (\sigma_1, \sigma_2, s)$.

$\mathsf{Verify}(VK, M, \sigma)$：
Parse σ as (σ_1, σ_2, s).
$t \leftarrow H_k(\sigma_2 \| M)$; $m \leftarrow g^t h^s$
Parse m as $m_1 m_2 \ldots m_n$ (each of m_i is the i-th bit of m).
Check $e(\sigma_1, g) \stackrel{?}{=} e(\sigma_2, u' \prod_{i=1}^n u_i^{m_i}) \cdot e(g_1, g_2)$.
Output `accept` if this holds. Otherwise output `reject`.

図 3.3: BSW 署名

を計算できてしまう。この署名は乱数 $r + r'$ を用いた整合性のとれた署名になっているため、$\mathsf{Verify}(vk, m, \sigma') = \mathtt{accept}$ となる。

また、BSW 署名は Waters 署名の問題を解消したが、安全性証明のために構成要素として衝突困難ハッシュ関数を用いる必要があった。しかし、これによりハッシュ関数の入力空間、実際には任意長のメッセージに対して安全に署名することができる構成になっている。

次節以降、この衝突困難ハッシュ関数を取り除けるような方式を提案する。

3.2　ターゲット衝突困難ハッシュ関数を用いた方式

3.1.3 節で見たように、BSW 署名はスタンダードモデルの下で CDH 仮定という基礎的な仮定に基づいて SEUF-CMA 安全性を示せ、かつ署名サイズや署名生成、検証の計算コストも効率がよい方式であるが、構成には衝突困難ハッシュ関数が必要である。

本節では、この BSW 署名を改良し、署名サイズについては同等、署名生成、署名検証についてもほぼ同等で、衝突困難ハッシュ関数は用いないという署名方式を示す。衝突困難ハッシュ関数の代わりとして用いるハッシュ関数は、ターゲット衝突困難ハッシュ関数である。

本節での提案方式を得るまでとそれに対する安全性の証明は、BSW 方式とほぼ同様である。その流れは以下のようになる。

1. BSW 署名の際に用いられた “*partitioned*” 署名 (定義 3.1) のさらに特別な場合である “*simulatable-partitioned*” 署名を定義する (定義 3.2)。

2. EUF-CMA 安全性を持つ *simulatable-partitioned* 署名を、SEUF-CMA 安全性を持つ署名方式へと変換する一般的構成法の構成を示す (3.2.1 節)。この変換には、BSW 変換の際に必要な衝突困難ハッシュ関数は必要としない。代わりにターゲット衝突困難ハッシュ関数を用いる。安全性証明には、変換前の署名方式の EUF-CMA 安全性、DL 仮定、ハッシュ関数のターゲット衝突困難性が必要である。

3. Waters 署名 [62] が *simulatable-partitioned* クラスに当てはまることを証明し、Waters 署名に提案する変換を適用する (3.2.2 節)。

partitioned 署名のさらに特別な場合である *simulatable-partitioned* 署名の定義は以下である。

定義 3.2. 電子署名方式 Σ が以下の *4* つの性質を持つ場合、Σ は *simulatable-partitioned* 署名であるという。

- **性質 1 と性質 2:** *partitioned* の定義と同様 *(*定義 *3.1)*。
- **性質 3:** 以下の *2* つのアルゴリズムが存在する。
  - KeyGen' *:* 確率的アルゴリズム。1^κ *(*セキュリティパラメータ $\kappa \in \mathcal{N}$*)* を入力とし、KeyGen の出力 (sk, vk) と分布が等しい署名鍵、検証鍵の対 *(*sk'*,* vk'*)* と共に、以下の S'_1 のためのトラップドア TD を出力する。この操作を $(sk', vk', TD) \leftarrow \mathsf{KeyGen}'(1^\kappa)$ と書く。
  - S'_1 *:* 決定的アルゴリズム。署名鍵 sk、署名の一部 σ_2、メッセージ m、上記 KeyGen' から出力された TD を入力とし、$\mathsf{Verify}(vk, m, (\sigma'_1, \sigma_2)) = \mathtt{accept}$ となるような $\sigma'_1 \in \mathcal{S}_1$ を出力する。この操作を $\sigma'_1 \leftarrow \mathsf{S}'_1(sk, m, \sigma_2, TD)$ と書く。
- **性質 4:** 全ての sk に対し、$\mathsf{S}_2(sk, r)$ は $\mathcal{R}$ から $\mathcal{S}_2$ への全単写となっている。

このような定義を用いる必要がある理由は、次節の証明のアイデアで述べる。

3.2.1　強偽造不可能性を持つ方式への変換

本節では、任意の EUF-CMA 安全性を持つ *simulatable-partitioned* 署名を衝突困難ハッシュ関数を用いず、代わりにターゲット衝突困難ハッシュ関数を用いて SEUF-CMA 安全性を持つ署名方式へと一般的に変換する方法を示す。

$\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ を EUF-CMA 安全性を持ち、*simulatable-partitioned* 署名の性質を全て満たす署名方式とする。すなわち、性質 1 により、署名アルゴリズム Sign が二つの決定的アルゴリズム S_1 と S_2 に分割できる。p を十分大きな素数、$\mathbb{G}$ を位数が p の巡回群とする。*simulatable-partitioned* 署名の性質 2 が成り立つように、$\mathbb{G}$

の要素は一意に特定できるビットでの表し方があるとする。$H : \mathcal{S}_2 \times \{0,1\}^* \to \mathbb{Z}_p$、$G : \mathcal{K} \times \mathcal{S}_2 \to \mathbb{Z}_p$ 及び $F : \mathcal{K} \times \mathbb{G} \to \mathcal{M}$ を全てターゲット衝突困難ハッシュ関数とする。ただし、全てのハッシュ関数で、定義域のうち最初の集合はハッシュ鍵空間、そして後の集合はハッシュ関数の入力空間とする。また、H のハッシュ鍵空間は、$\mathcal{S}_2$(S_2 の出力空間) であるとする。

提案するターゲット衝突困難ハッシュ関数を用いた変換は図 3.4 である ($\Sigma_{new} = (\mathsf{KeyGen}_{new}, \mathsf{Sign}_{new}, \mathsf{Verify}_{new})$ は変換後に得られる署名方式)。

<table>
<tr><td colspan="2">構成要素：
$\mathsf{S}_1, \mathsf{S}_2$：変換前の署名方式 Σ の simulatable-partitioned の性質によって
分割された 2 つの署名アルゴリズム
$\mathbb{G}$：位数が十分大きな素数 p の巡回群
$H : \mathcal{S}_2 \times \{0,1\}^* \to \mathbb{Z}_p, G : \mathcal{K} \times \mathbb{G} \to \mathbb{Z}_p, F : \mathcal{K} \times \mathbb{G} \to \mathcal{M}$：全て TCRHF</td></tr>
<tr><td>$\mathsf{KeyGen}_{new}(1^\kappa)$:
$(sk, vk) \leftarrow \mathsf{KeyGen}(1^\kappa)$
$g, h_1, h_2 \leftarrow \mathbb{G}$; $k \leftarrow \mathcal{K}$
$SK \leftarrow (sk)$, $VK \leftarrow (vk, g, h_1, h_2, k)$
Output (SK, VK).</td><td>$\mathsf{Sign}_{new}(SK, M)$:
$s \leftarrow \mathbb{Z}_p$; $r \leftarrow \mathcal{R}$; $\sigma_2 \leftarrow \mathsf{S}_2(sk, r)$
$t \leftarrow H_{\sigma_2}(M)$; $t' \leftarrow G_k(\sigma_2)$; $m \leftarrow g^t h_1^s h_2^{t'}$
$m' \leftarrow F_k(m)$; $\sigma_1 \leftarrow \mathsf{S}_1(sk, m', r)$
Output $\sigma \leftarrow (\sigma_1, \sigma_2, s)$.</td></tr>
<tr><td colspan="2">$\mathsf{Verify}_{new}(VK, M, \sigma)$:
Parse σ as (σ_1, σ_2, s).
$t \leftarrow H_{\sigma_2}(M)$; $t' \leftarrow G_k(\sigma_2)$; $m \leftarrow g^t h_1^s h_2^{t'}$; $m' \leftarrow F_k(m)$
Output <code>accept</code> if $\mathsf{Verify}(vk, m', (\sigma_1, \sigma_2)) =$ <code>accept</code>. Otherwise output <code>reject</code>.</td></tr>
</table>

図 3.4: TCRHF を用いた変換の構成

安全性　以下は、安全性についての記述である。

定理 3.3. 以下の条件が成り立つとき、*TCRHF* を用いた変換 *(図 3.4)* を適用した後の方式 Σ_{new} は (t, q, ϵ)-*SEUF-CMA* 安全性を満たす電子署名方式である。

- 構成要素の電子署名方式 Σ は $(t, q, \epsilon/6)$-*EUF-CMA* 安全性を満たす *simulatable-partitioned* 署名である。
- $\mathbb{G}$ において $(t, \epsilon/3)$-*DL* 仮定が成り立つ。
- H、G、及び F はそれぞれ、$(t, \epsilon/6q)$、$(t, \epsilon/6q)$、$(t, \epsilon/6q)$-*TCRHF* である。

変換の構成及び証明のアイデア　証明は、提案する方式を破る攻撃者がいる場合、その攻撃者に対し SEUF-CMA ゲームをシミュレートして利用しながら、仮定となる変換適用前の方式、DL 仮定、またはターゲット衝突困難ハッシュ関数いずれかの安全性を破れるシミュレータが構成できることを示す。

変換の構成方法は、[16] を基にしている。我々の方式では、3 つのターゲット衝突困難ハッシュ関数を使用する。H の役割は、署名するメッセージをハッシュし、固定長のデータに圧縮することである。G の役割は、ターゲット衝突困難性により、異なる σ_2 は、異なる $\mathbb{Z}_p$ の要素に写されることを保障することである。よって、$\mathbb{G}$ から $\mathbb{Z}_p$ への単射が使用できるならば、ハッシュ関数である必要は無い (その場合、安全性証明での場合分けを少なくすることができる)。F の役割は、G と同様、異なる $\mathbb{G}$ の要素が異なる $\mathcal{M}$(変換前の署名方式のメッセージ空間) へと写されることを保障することである。従って、もし $\mathbb{G}$ の要素が常に $\mathcal{M}$ 上の要素に単射されると分かっているならば、F がハッシュ関数である必要は無い。

署名の要素のうち、メッセージに依存しないランダムな部分 σ_2 をハッシュ関数 H のハッシュ鍵として再利用することは、[47] で用いられたアイデアである。この σ_2 は、内部の Sign が実行されるたびに作られるメッセージに依存しないランダムな値のため、ターゲット衝突困難ハッシュ関数を用いた Hash-and-Sign [48] におけるハッシュ鍵の様な役割を果たすことができる。σ_2 が $\mathcal{S}_2$ において一様ランダムな値であることは、σ_2 を作成する際にその内部の乱数 $r \in \mathcal{R}$ を一様ランダムに選んでいることによって、定義 3.2 の性質 4 により保障されている。

BSW 変換 [16] の場合と同様に、署名のメッセージに依存する要素 σ_1 を作成するということは、署名者は、Sign_{new} に入力されたメッセージ M と同時に、署名のうちメッセージに依存しないランダムな要素 σ_2 にも署名するということを意味する。よって、メッセージ M を変更したり、別の乱数 r' が用いられるような署名にしようとすると、署名 (σ_1, σ_2) は無効なものになってしまう。この性質が署名の SEUF-CMA 安全性につながる。[16] の著者達は、衝突困難ハッシュ関数を用いて、(M, σ_2) のペアのハッシュ値の衝突が起こらないことを保障した。提案方式では、離散対数仮定に基づくカメレオンハッシュ [41] とターゲット衝突困難ハッシュ関数 H と G の組み合わせにより [16] の場合と同様の機能を達成している。そして、カメレオンハッシュ関数は、安全性証明で、シミュレータ構成の際の循環の矛盾、すなわち内部の署名アルゴリズム Sign への入力メッセージ m' を作るためには、シミュレータは Sign から出力される署名のランダムな要素 σ_2 の内部の乱数 r を前もって知っておく必要がある、ということ解消するという役割を果たしている。

本節の変換を適用することにより得られる方式の署名は、ターゲット衝突困難ハッシュ関数 H のハッシュ鍵自体を含んでいる。よって、安全性証明において安全性を H のターゲット衝突困難性に帰着する場合、攻撃者により出力された偽造署名は $\mathcal{S}_2$ から一様ランダムには選ばれていない署名のメッセージに依存しない要素 $\hat{\sigma_2}$ を含んでいる可能性がある (ここでは、ハット付き文字は SEUF-CMA ゲームにおいて攻撃者が出力したものを指すことにする)。しかし、本節のシミュレータの構成法では、シミュレータはその場合でも、攻撃者の最後に出力する偽造署名 $(\hat{\sigma} = (\hat{\sigma_1}, \hat{\sigma_2}))$ の要素 $\hat{\sigma_2}$ が、TCR ゲームの定義に沿ったシミュレータに与えられる $\mathcal{S}_2$ から一様ランダムに選ばれたハッシュ鍵 $\bar{\sigma_2}$ と等しい場合に、H に関する TCR ゲームに勝利することができる (定理 3.3 の証明の中での Type 6 攻撃者の偽造署名)。*simulatable-partitioned* 署名の性質 3 を満たすアルゴリズム KeyGen' と S'_1、そしてトラップドア TD は、このタイプの攻

撃者に対するシミュレータ構成の際に必要になる。H のハッシュ鍵に σ_2 を使うことの代償として、シミュレータがハッシュ鍵 $\bar{\sigma_2}$ を与えられたとき、シミュレータは、署名の内部での乱数 r が、σ_1 と $\bar{\sigma_2}$ の中で等しくなっているようなもう片方の署名の要素 σ_1 を計算できなければならない。r は常に σ_2 から直接計算できるわけではないため (例えば、Waters 署名 [62] では $\sigma_2 = g^r$ であり、g^r から r を計算できることは、離散対数問題を解けることになってしまう)、シミュレータが乱数 r に矛盾の無い署名 (σ_1, σ_2) を r を直接知らずに計算するため、前もって知っておくことのできるトラップドア TD が必要になる。

定理 3.3 の証明　$\mathcal{A}$ を、提案方式 Σ_{new} の (t, q, ϵ)-SEUF-CMA 安全性を破る攻撃者とする。SEUF-CMA ゲームの *Setup* の段階において、$\mathcal{A}$ は公開鍵 $VK = (vk, g, h_1, h_2, k)$ を与えられる。また、q を $\mathcal{A}$ が発行する署名クエリの総数とし、*Queries* の段階において、$\mathcal{A}$ は i 回目 $(i \in \{1, \ldots, q\})$ の署名クエリとして M_i を発行し、それに対応する署名 $\sigma_i = (\sigma_{i,1}, \sigma_{i,2}, s_i)$ を与えられるとする。一般性を失わずに、$q > 0$ とする。*Output* の段階において、$\mathcal{A}$ はメッセージと偽造署名の対 $(\hat{M}, \hat{\sigma} = (\hat{\sigma_1}, \hat{\sigma_2}, \hat{s}))$ を出力するものとする[1]。$\hat{t} = H_{\hat{\sigma_2}}(\hat{M})$、$\hat{t'} = G_k(\hat{\sigma_2})$、$\hat{m} = g^{\hat{t}} h_1^{\hat{s}} h_2^{\hat{t'}}$、及び、$\hat{m'} = F_k(\hat{m})$ であると定義する。$\mathcal{A}$ の出力する偽造署名とメッセージと偽造署名 $(\hat{M}, \hat{\sigma})$ のタイプによって $\mathcal{A}$ を以下の 6 つに分類する。

Type 1. $\forall i \in \{1, \ldots, q\} : \hat{m'} \neq m'_i$.

Type 2. $\exists i \in \{1, \ldots, q\} : \hat{m'} = m'_i \wedge \hat{m} \neq m_i$.

Type 3. $\exists i \in \{1, \ldots, q\} : \hat{m'} = m'_i \wedge \hat{m} = m_i \wedge \hat{t'} \neq t'_i$.

Type 4. $\exists i \in \{1, \ldots, q\} : \hat{m'} = m'_i \wedge \hat{m} = m_i \wedge \hat{t'} = t'_i \wedge \hat{\sigma_2} \neq \sigma_{i,2}$.

Type 5. $\exists i \in \{1, \ldots, q\} : \hat{m'} = m'_i \wedge \hat{m} = m_i \wedge \hat{t'} = t'_i \wedge \hat{\sigma_2} = \sigma_{i,2} \wedge \hat{t} \neq t_i$.

Type 6. $\exists i \in \{1, \ldots, q\} : \hat{m'} = m'_i \wedge \hat{m} = m_i \wedge \hat{t'} = t'_i \wedge \hat{\sigma_2} = \sigma_{i,2} \wedge \hat{t} = t_i$.

$\mathcal{A}$ が偽造に成功する場合、$\mathcal{A}$ により出力される $(\hat{M}, \hat{\sigma})$ は、必ず上記 6 タイプのどれかにあてはまる。

以下の議論では、それぞれのタイプの SEUF-CMA を破る攻撃者 $\mathcal{A}$ に対し、SEUF-CMA ゲームをシミュレーションを行いながら Σ_{new} の構成要素の安全性を破ることができるシミュレータ $\mathcal{B}_i (i \in \{1, \ldots, 6\})$ を構成できることを示す。$\mathcal{A}$ がタイプ 1 の攻撃者の場合、$\mathcal{B}_1$ は構成要素である署名方式 Σ の EUF-CMA 安全性を、タイプ 3 及びタイプ 5 の攻撃者の場合、$\mathcal{B}_3$ と $\mathcal{B}_5$ は $\mathbb{G}$ における離散対数問題を、タイプ 2、タイプ 4、タイプ 6 の攻撃者の場合、$\mathcal{B}_2$、$\mathcal{B}_4$、$\mathcal{B}_6$ は TCRHF の安全性を破ることができる。シミュレータは $\mathcal{A}$ に対し SEUF-CMA ゲームのシミュレーションを始める前にコインを振り、$\mathcal{A}$ がどのタイプの攻撃者かを推測し、$\mathcal{B}_1, \ldots, \mathcal{B}_6$ のどのシミュレータとして動くかを決めることができる。以下、それぞれのタイプに対するシミュレータの具体的構成法を示す。

[1] ハット付き文字は、$\mathcal{A}$ が作る偽造署名を作成する過程で使われる文字のみに使用する。

Type 1:
$\mathcal{A}$ を Σ_{new} の (t, q, ϵ)-SEUF-CMA 安全性を破るタイプ 1 の攻撃者であると仮定する。この $\mathcal{A}$ を用いて構成要素である電子署名方式 Σ の (t, q, ϵ)-EUF-CMA 安全性を破ることのできる $\mathcal{B}_1$ を構成する。$\mathcal{B}_1$ は最初に vk を与えられ、EUF-CMA ゲームの勝利条件を満たすような $(\hat{m}', \hat{\sigma} = (\hat{\sigma_1}, \hat{\sigma_2}))$ を出力する。$\mathcal{B}_1$ は以下の様に $\mathcal{A}$ に対し SEUF-CMA ゲームのシミュレートを行いつつ利用して自身の EUF-CMA チャレンジャー $\mathcal{C}$ との間で EUF-CMA ゲームを行う。

Setup. $\mathcal{B}_1$ は以下の様にして VK を作成し、$\mathcal{A}$ に渡す。

1. $g \in \mathbb{G}$、及び $k \in \mathcal{K}$ を一様ランダムに選ぶ。
2. $a, b \in \mathbb{Z}_p^*$ を一様ランダムに選び、$h_1 \leftarrow g^a, h_2 \leftarrow g^b$ を計算する。
3. $VK \leftarrow (vk, g, h_1, h_2, k)$ とし、$\mathcal{A}$ に渡す。

Queries. $\mathcal{B}_1$ は $\mathcal{A}$ の発行する署名クエリ $M_{i \in \{1,\ldots,q\}}$ に対し σ_i を作成し応答する。

1. $w_i \in \mathbb{Z}_q$ を一様ランダムに選び、$m_i \leftarrow g^{w_i}$ を計算する。
2. $m_i' \leftarrow F_k(m_i)$ を計算する。
3. m_i' を自身の i 回目の署名クエリとして $\mathcal{C}$ に問い合わせ、m_i' の署名 $(\sigma_{i,1}, \sigma_{i,2})$ を受け取る。
4. $t_i \leftarrow H_{\sigma_{i,2}}(M_i)$、及び $t_i' \leftarrow G_k(\sigma_{i,2})$ を計算する。
5. $s_i \leftarrow (w_i - t_i - bt_i')/a$ を計算する。
6. $\sigma_i \leftarrow (\sigma_{i,1}, \sigma_{i,2}, s_i)$ とし、$\mathcal{A}$ に渡す。

Output. $\mathcal{A}$ がタイプ 1 のメッセージと偽造署名の対 $(\hat{M}, (\hat{\sigma_1}, \hat{\sigma_2}, \hat{s}))$ を出力する。$\mathcal{B}_1$ は Verify_{new} のアルゴリズムに沿って正しく $\hat{m}'$ を計算する。$\mathcal{C}$ に対し、$(\hat{m}', (\hat{\sigma_1}, \hat{\sigma_2}))$ を出力する。

$\hat{m}'$ が Verify_{new} のアルゴリズムに沿って正しく計算された場合、Σ_{new} の構成により、$\mathsf{Verify}_{new}(VK, \hat{M}, (\hat{\sigma_1}, \hat{\sigma_2}, \hat{s})) = \mathtt{accept}$ が成り立つならば、$\mathsf{Verify}(vk, \hat{m}', (\hat{\sigma_1}, \hat{\sigma_2})) = \mathtt{accept}$ が成り立つことになる。$\mathcal{A}$ はタイプ 1 の攻撃者であるため、全ての $i \in \{1, \ldots q\}$ について、$\hat{m}' \neq m_i'$ が成り立つ。従って、$\mathcal{B}_1$ は EUF-CMA ゲームに対して正しく動作する攻撃者になっている。故に、$\mathcal{A}$ がタイプ 1 の偽造に成功する場合はいつでも、$\mathcal{B}_1$ は構成要素である電子署名方式 Σ における新しいメッセージ $\hat{m}'$ の署名を出力でき、EUF-CMA ゲームに勝利することができる。

Type 2:
$\mathcal{A}$ を Σ_{new} の (t, q, ϵ)-SEUF-CMA 安全性を破るタイプ 2 の攻撃者であると仮定する。この $\mathcal{A}$ を用いて構成要素である $(t, \epsilon/q)$-TCRHF F の安全性を破ることのできる $\mathcal{B}_2$ を構成する。$\mathcal{B}_2$ は以下の様に $\mathcal{A}$ に対し SEUF-CMA ゲームのシミュレートを行いつつ利用して自身の TCR チャレンジャー $\mathcal{C}$ との間で F に関する TCR ゲームを行う。

Setup. $\mathcal{B}_2$ は以下の様にして VK を作成し、$\mathcal{A}$ に渡す。

1. インデックス $j \in \{1, \ldots, q\}$ を一様ランダムに選ぶ。
2. KeyGen を実行し、sk, vk を得る。
3. $g \in \mathbb{G}$ を一様ランダムに選ぶ。
4. $a, b \in \mathbb{Z}_p^*$ を一様ランダムに選び、$h_1 \leftarrow g^a, h_2 \leftarrow g^b$ を計算する。
5. $\bar{w} \in \mathbb{Z}_p$ を一様ランダムに選び、$\bar{m} \leftarrow g^{\bar{w}}$ を計算する。
6. $\bar{m}$ を TCR ゲームの最初の入力として $\mathcal{C}$ に出力し、$k \in \mathcal{K}$ を受け取る。
7. $\bar{m}' \leftarrow F_k(\bar{m})$ を計算する。
8. $VK \leftarrow (vk, g, h_1, h_2, k)$ とし、$\mathcal{A}$ に渡す。$SK = sk$ は保持しておく。

Queries. $\mathcal{B}_2$ は $\mathcal{A}$ の発行する署名クエリ $M_{i \in \{1,\ldots,q\}}$ に対し i の状態により以下の様に σ_i を作成し応答する。

$i \neq j$ のとき : $\sigma_i \leftarrow \mathsf{Sign}_{new}(SK, M_i)$ を計算し、$\mathcal{A}$ に渡す。

それ以外のとき :

1. $m_j \leftarrow \bar{m}$、$m'_j \leftarrow \bar{m}'$ とする。
2. $\mathsf{Sign}(sk, m'_j)$ を実行し、$(\sigma_{j,1}, \sigma_{j,2})$ を得る。
3. $t_j \leftarrow H_{\sigma_{j,2}}(M_j)$、$t'_j \leftarrow G_k(\sigma_{j,2})$ を計算する。
4. $s_j \leftarrow (\bar{w} - t_j - bt'_j)/a$ を計算する。
5. $\sigma_j \leftarrow (\sigma_{j,1}, \sigma_{j,2}, s_j)$ とし、$\mathcal{A}$ に渡す。

Output. $\mathcal{A}$ がタイプ 2 のメッセージと偽造署名の対 $(\hat{M}, (\hat{\sigma_1}, \hat{\sigma_2}, \hat{s}))$ を出力する。$\mathcal{B}_2$ は Verify_{new} のアルゴリズムに沿って正しく $\hat{m}$ を計算する。$F_k(\hat{m}) = F_k(m_j) \wedge \hat{m} \neq m_j$ が成り立つならば、$\mathcal{B}_2$ は $\mathcal{C}$ に対し、$\hat{m}$ を出力する。成り立たない場合、$\mathcal{B}_2$ は TCR ゲームに勝利することを諦め停止する。

$\mathcal{A}$ はタイプ 2 の攻撃者であり、$\hat{m}' = m'_i$ は $F_k(\hat{m}) = F_k(m_i)$ を意味するため、$F_k(\hat{m}) = F_k(m_i) \wedge \hat{m} \neq m_i$ が成り立つ様なインデックス $i \in \{1, \ldots, q\}$ が少なくとも一つ存在する。従って、$\hat{m}$ は F について、ハッシュ鍵 k の下、m_i と同一のハッシュ値を持つ。$\mathcal{B}_2$ がその様なインデックス i を Setup の段階でインデックス j として選ぶ確率は少なくとも $1/q$ である。故に、$\mathcal{A}$ がタイプ 2 の偽造に成功する場合、$\mathcal{B}_2$ は少なくとも確率 $1/q$ で F についての TCR ゲームに勝利することができる。

Type 3:

$\mathcal{A}$ を Σ_{new} の (t, q, ϵ)-SEUF-CMA 安全性を破るタイプ 3 の攻撃者であると仮定する。この $\mathcal{A}$ を用いて $\mathbb{G}$ における (t, ϵ)-DL 仮定を破るシミュレータ $\mathcal{B}_3$ を構成する。$\mathcal{B}_3$ は最初に離散対数問題 (g, X) を与えられ、以下の様に $\mathcal{A}$ に対し SEUF-CMA ゲームのシミュレートを行いつつ利用して離散対数問題の解 $\log_g X$ を出力する。

Setup. $\mathcal{B}_3$ は以下の様にして VK を作成し、$\mathcal{A}$ に渡す。

1. g を VK の要素とし、$h_2 \leftarrow X$ とする。
2. KeyGen を実行し、sk, vk を得る。
3. $a \in \mathbb{Z}_p^*$ を一様ランダムに選び、$h_1 \leftarrow g^a$ を計算する。
4. $k \in \mathcal{K}$ を一様ランダムに選ぶ。
5. $VK \leftarrow (vk, g, h_1, h_2, k)$ とし、$\mathcal{A}$ に渡す。$SK \leftarrow sk$ は保持しておく。

Queries. $\mathcal{B}_3$ は $\mathcal{A}$ の発行する署名クエリ $M_{i\in\{1,\ldots,q\}}$ に対し、$\sigma_i \leftarrow \mathsf{Sign}_{new}(SK, M_i)$ を計算し、$\mathcal{A}$ に渡す。

Output. $\mathcal{A}$ はタイプ 3 のメッセージと偽造署名の対 $(\hat{M}, (\hat{\sigma_1}, \hat{\sigma_2}, \hat{s}))$ を出力する。$\mathcal{B}_3$ は Verify_{new} のアルゴリズムに沿って正しく $\hat{t}$ 及び $\hat{t}'$ を計算する。$\log_g X \leftarrow (\hat{t} + a\hat{s} - t_i - as_i)/(t_i' - \hat{t}')$ を計算し、DL 問題の解として出力する。

$\mathcal{A}$ はタイプ 3 の攻撃者であるため、$\hat{m} = m_i \wedge \hat{t}' \neq t_i'$ が成り立つ様なインデックス $i \in \{1, \ldots, q\}$ が少なくとも一つ存在し、$\mathcal{B}_3$ は Output の段階でその様な i を知ることができる。$\hat{m} = m_i$ は $g^{\hat{t}} h_1^{\hat{s}} h_2^{\hat{t}'} = g^{t_i} h_1^{s_i} h_2^{t_i'}$ を意味するが、この式は $g^{\hat{t}+a\hat{s}} X^{\hat{t}'} = g^{t_i+as_i} X^{t_i'}$ と書くことができる。この等式から、底を g とする対数を考えることで、X の底を g とする対数、すなわち与えられた離散対数問題 (g, X) の解を得ることができる。$\hat{t}' \neq t_i'$ によって、$t_i' - \hat{t}' \neq 0$ であることは保障されている。故に、$\mathcal{A}$ がタイプ 3 の偽造に成功する場合はいつでも、$\mathcal{B}_3$ は与えられた $\mathbb{G}$ における離散対数問題を解くことができる。

Type 4:

$\mathcal{A}$ を Σ_{new} の (t, q, ϵ)-SEUF-CMA 安全性を破るタイプ 4 の攻撃者であると仮定する。この $\mathcal{A}$ を用いて構成要素である $(t, \epsilon/q)$-TCRHF G の安全性を破ることのできる $\mathcal{B}_4$ を構成する。$\mathcal{B}_4$ は以下の様に $\mathcal{A}$ に対し SEUF-CMA ゲームのシミュレートを行いつつ利用して自身の TCR チャレンジャー $\mathcal{C}$ との間で G に関する TCR ゲームを行う。

Setup. $\mathcal{B}_4$ は以下の様にして VK を作成し、$\mathcal{A}$ に渡す。

1. インデックス $j \in \{1, \ldots, q\}$ を一様ランダムに選ぶ。
2. KeyGen を実行し、sk, vk を得る。
3. $g, h_1, h_2 \in \mathbb{G}$ を一様ランダムに選ぶ。
4. $\bar{r} \in \mathcal{R}$ を一様ランダムに選び、$\bar{\sigma_2} \leftarrow \mathsf{S}_2(sk, \bar{r})$ を計算する。
5. $\bar{\sigma_2}$ を TCR ゲームの最初の入力として $\mathcal{C}$ に出力し、$k \in \mathcal{K}$ を得る。
6. $VK \leftarrow (vk, g, h_1, h_2, k)$ とし、$\mathcal{A}$ に渡す。$SK \leftarrow sk$ は保持しておく。

Queries. $\mathcal{B}_4$ は $\mathcal{A}$ の発行する署名クエリ $M_{i\in\{1,\ldots,q\}}$ に対し i の状態により以下の様に σ_i を作成し応答する。

$i \neq j$ **のとき** ：$\sigma_i \leftarrow \mathsf{Sign}_{new}(SK, M_i)$ を計算し、$\mathcal{A}$ に渡す。

それ以外のとき ：

1. $r_j \leftarrow \bar{r}$、及び $\sigma_{2,j} \leftarrow \bar{\sigma_2}$ とする。
2. $t_j \leftarrow H_{\sigma_{j,2}}(M_j)$、及び $t'_j \leftarrow G_k(\sigma_{j,2})$ を計算する。
3. $s_j \in \mathbb{Z}_p^*$ を一様ランダムに選ぶ。
4. $m_j \leftarrow g^{t_j} h_1^{s_j} h_2^{t'_j}$、及び $m'_j \leftarrow F_k(m_j)$ を計算する。
5. $\sigma_{j,1} \leftarrow \mathsf{S}_1(sk, m', r_j)$ を計算する。
6. $\sigma_j \leftarrow (\sigma_{j,1}, \sigma_{j,2}, s_j)$ とし、$\mathcal{A}$ に渡す。

Output. $\mathcal{A}$ がタイプ 4 のメッセージと偽造署名の対 $(\hat{M}, (\hat{\sigma_1}, \hat{\sigma_2}, \hat{s}))$ を出力する。$G_k(\hat{\sigma_2}) = G_k(\sigma_{j,2}) \wedge \hat{\sigma_2} \neq \sigma_{j,2}$ が成り立つならば、$\mathcal{B}_4$ は $\mathcal{C}$ に対し、$\hat{\sigma_2}$ を出力する。成り立たない場合、$\mathcal{B}_4$ は TCR ゲームに勝利することを諦め停止する。

$\mathcal{A}$ はタイプ 4 の攻撃者であり、$\hat{t'} = t'_j$ は $G_k(\hat{\sigma_2}) = G_k(\sigma_{i,2})$ を意味するため、$G_k(\hat{\sigma_2}) = G_k(\sigma_{i,2}) \wedge \hat{\sigma_2} \neq \sigma_{i,2}$ が成り立つ様なインデックス $i \in \{1, \ldots, q\}$ が少なくとも一つ存在する。従って、$\hat{\sigma_2}$ は G について、ハッシュ鍵 k の下、$\sigma_{i,2}$ と同一のハッシュ値を持つ。$\mathcal{B}_4$ がその様なインデックス i を Setup の段階でインデックス j として選ぶ確率は少なくとも $1/q$ である。故に、$\mathcal{A}$ がタイプ 4 の偽造に成功する場合、$\mathcal{B}_4$ は少なくとも確率 $1/q$ で G についての TCR ゲームに勝利することができる。

Type 5:

$\mathcal{A}$ を Σ_{new} の (t, q, ϵ)-SEUF-CMA 安全性を破るタイプ 5 の攻撃者であると仮定する。この $\mathcal{A}$ を用いて $\mathbb{G}$ における (t, ϵ)-DL 仮定を破るシミュレータ $\mathcal{B}_5$ を構成する。$\mathcal{B}_5$ は最初に離散対数問題 (g, X) を与えられ、以下の様に $\mathcal{A}$ に対し SEUF-CMA ゲームのシミュレートを行いつつ利用して離散対数問題の解 $\log_g X$ を出力する。

Setup. $\mathcal{B}_5$ は以下の様に VK を作成し、$\mathcal{A}$ に渡す。

1. g を VK の要素とし、$h_1 \leftarrow X$ とする。
2. KeyGen を実行し、sk, vk を得る。
3. $h_2 \in \mathbb{G}$、$k \in \mathcal{K}$ をそれぞれ一様ランダムに選ぶ。
4. $VK \leftarrow (vk, g, h_1, h_2, k)$ とし、$\mathcal{A}$ に渡す。$SK \leftarrow sk$ は保持しておく。

Queries. $\mathcal{B}_5$ は $\mathcal{A}$ の発行する署名クエリ $M_{i \in \{1,\ldots,q\}}$ に対し、$\sigma_i \leftarrow \mathsf{Sign}_{new}(SK, M_i)$ を計算し、$\mathcal{A}$ に渡す。

Output. $\mathcal{A}$ はタイプ 5 のメッセージと偽造署名の対 $(\hat{M}, (\hat{\sigma_1}, \hat{\sigma_2}, \hat{s}))$ を出力する。$\mathcal{B}_5$ は $\hat{t} \leftarrow H_{\hat{\sigma_2}}(\hat{M})$ を計算する。$\log_g X \leftarrow (\hat{t} - t_i)/(s_i - \hat{s})$ を計算し、DL 問題の解として出力する。

$\mathcal{A}$ はタイプ 5 の攻撃者であるため、$\hat{m} = m_i \wedge \hat{t} \neq t_i$ が成り立つ様なインデックス $i \in \{1, \ldots, q\}$ が少なくとも一つ存在し、$\mathcal{B}_5$ は Output の段階でその様な i を知ることができる。$\hat{m} = m_i$ は $g^{\hat{t}} X^{\hat{s}} = g^{t_i} X^{s_i}$ を意味する。この等式において、$\hat{s} = s_i \wedge g^{\hat{t}} X^{\hat{s}} = g^{t_i} X^{s_i}$ は $\hat{t} = t_i$ を意味するため、$s_i - \hat{s} = 0$ は起こらない。よって、この等式から、底 g とする対数を考えることで、X の g を底とする対数、すなわち与えられた離散対数問題 (g, X) の解を得ることができる。故に、$\mathcal{A}$ がタイプ 5 の偽造に成功する場合はいつでも、$\mathcal{B}_5$ は与えられた $\mathbb{G}$ における離散対数問題を解くことができる。

Type 6:

$\mathcal{A}$ を Σ_{new} の (t, q, ϵ)-SEUF-CMA 安全性を破るタイプ 6 の攻撃者であると仮定する。この $\mathcal{A}$ を用いて構成要素である $(t, \epsilon/q)$-TCRHF H の安全性を破ることのできる $\mathcal{B}_6$ を構成する。$\mathcal{B}_6$ は以下の様に $\mathcal{A}$ に対し SEUF-CMA ゲームのシミュレートを行いつつ利用して自身の TCR チャレンジャー $\mathcal{C}$ との間で H に関する TCR ゲームを行う。

Setup. $\mathcal{B}_6$ は以下の様にして VK を作成し、$\mathcal{A}$ に渡す。

1. インデックス $j \in \{1 \ldots q\}$ を一様ランダムに選ぶ。
2. KeyGen' を実行し、$sk,\ vk,\ TD$ を得る。
3. $g, h_1, h_2 \in \mathbb{G}$、及び、$k \in \mathcal{K}$ をそれぞれ一様ランダムに選ぶ。
4. $VK \leftarrow (vk, g, h_1, h_2, k)$ を $\mathcal{A}$ に渡す。$SK \leftarrow sk$ は保持しておく。

Queries. $\mathcal{B}_6$ は $\mathcal{A}$ の発行する署名クエリ $M_{i \in \{1,\ldots,q\}}$ に対し i の状態により以下の様に σ_i を作成し応答する。

$i \neq j$ のとき：$\sigma_i \leftarrow \mathsf{Sign}_{new}(SK, M_i)$ を計算し、$\mathcal{A}$ に渡す。

それ以外のとき：

1. M_j を TCR ゲームの最初の入力として、$\mathcal{C}$ に出力し、$\bar{\sigma_2} \in \mathcal{S}_2$ を受け取る。
2. $\sigma_{j,2} \leftarrow \bar{\sigma_2}$ とする。
3. $t_j \leftarrow H_{\sigma_{j,2}}(M_j)$、及び $t'_j \leftarrow G_k(\sigma_{j,2})$ を計算する。
4. $s_j \in \mathbb{Z}_p$ を一様ランダムに選ぶ。
5. $m_j \leftarrow g^{t_j} h_1^{s_j} h_2^{t'_j}$、及び $m'_j \leftarrow F_k(m_j)$ を計算する。
6. $\sigma_{j,1} \leftarrow \mathsf{S}'_1(sk, m'_j, \bar{\sigma_2}, TD)$ を計算する。
7. $\sigma_j \leftarrow (\sigma_{j,1}, \sigma_{j,2}, s_j)$ とし、$\mathcal{A}$ に渡す。

Output. $\mathcal{A}$ がタイプ 6 のメッセージと偽造署名の対 $(\hat{M}, (\hat{\sigma_1}, \hat{\sigma_2}, \hat{s}))$ を出力する。$H_{\hat{\sigma_2}}(\hat{M}) = H_{\sigma_{j,2}}(M_j) \wedge \hat{\sigma_2} = \sigma_{j,2} \wedge \hat{M} \neq M_j$ が成り立つならば、$\mathcal{B}_6$ は $\mathcal{C}$ に対し $\hat{M}$ を出力する。成り立たない場合、$\mathcal{B}_6$ は TCR ゲームに勝利することを諦め停止する。

$\sigma_{j,2} = \bar{\sigma_2}$ を作る際に使われた $r_j = \bar{r}$ を知らないで内部の r_j が $\sigma_{j,2}$ と整合性がとれている正しい $\sigma_{j,1}$ を計算するために、KeyGen' と S'_1 が使われていることに注意されたい。

$\mathcal{A}$ が SEUF-CMA ゲームに対し勝利する場合、SEUF-CMA ゲームの定義より、全てのインデックス $i \in \{1 \dots q\}$ について、$(\hat{M}, \hat{\sigma_1}, \hat{\sigma_2}, \hat{s}) \neq (M_i, \sigma_{i,1}, \sigma_{i,2}, s_i)$ が成り立つ。また、$\mathcal{A}$ はタイプ 6 の攻撃者であり、$\hat{t} = t_i$ は $H_{\hat{\sigma_2}}(\hat{M}) = H_{\sigma_{i,2}}(M_i)$ を意味するので、$\hat{t} = t_i \wedge \hat{\sigma_2} = \hat{\sigma_{i,2}} \wedge \hat{s} = s_i$ が成り立つ様な $i \in \{1, \dots, q\}$ が少なくとも一つ存在する。*simulatable-partitioned* 署名の定義より、$\hat{m}' = m'_i \wedge \hat{\sigma_2} = \sigma_{i,2}$ ならば、$\hat{\sigma_1} = \sigma_{i,1}$ が成り立つ。以上をまとめると、$\hat{M} \neq M_i \wedge \hat{\sigma_2} = \sigma_{i,2} \wedge H_{\hat{\sigma_2}}(\hat{M}) = H_{\sigma_{i,2}}(M_i)$ が成り立つ様なインデックス $i \in \{1, \dots, q\}$ が存在する。すなわち、$\hat{M}$ が H について、ハッシュ鍵 $\hat{\sigma_2} = \sigma_{i,2}$ の下、M_i と同一のハッシュ値を持つという条件を満たす $i \in \{1, \dots, q\}$ が存在する。$\mathcal{B}_6$ がその様なインデックス i を Setup の段階でインデックス j として選ぶ確率は少なくとも $1/q$ である。故に、$\mathcal{A}$ がタイプ 6 の偽造に成功する場合、$\mathcal{B}_6$ は少なくとも確率 $1/q$ で H についての TCR ゲームに勝利することができる。

以上の議論で、6 タイプ全ての攻撃者に対してのシミュレータの構成法を示した。全てのタイプの攻撃者に対するシミュレーションは完全である。タイプ 1 の攻撃者は構成要素である Σ の EUF-CMA 安全性を破ることに、タイプ 3 とタイプ 5 の攻撃者は DL 仮定を破ることに、そしてタイプ 2、タイプ 4、タイプ 6 の攻撃者は TCRHF の安全性を破ることに使えることを示した。以上により、定理 3.3 は証明された。　□

3.2.2　CDH 仮定に基づく具体的な署名方式

本節では、CDH 仮定に基づく具体的な電子署名の構成方法を示す。Waters 署名 [62] に、前節で提案した変換法を適用するというのが、SEUF-CMA 安全性を持つ具体的な電子署名方式を得るための基本的アイデアである。

具体的な構成法の記述は次の通りである。p を十分に大きな素数とし、$\mathbb{G}$ を位数 p の双線形写像を持つような巡回群とする。$H : \mathbb{G} \times \{0,1\}^* \to \mathbb{Z}_p$、$G : \mathcal{K} \times \mathbb{G} \to \mathbb{Z}_p$、及び $F : \mathcal{K} \times \mathbb{G} \to \{0,1\}^n$ を全てターゲット衝突困難ハッシュ関数とする。$e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ を双線形写像とする。$\mathbb{G}$ の各要素は、それぞれ一意のビット列にエンコードできると仮定する。Waters 署名の場合、$\mathcal{S}_2 = \mathbb{G}$ となっている。そこで、ハッシュ関数 H の鍵空間が $\mathbb{G}$ とできることも仮定する (このようなハッシュ関数を仮定することは少ないが、ハッシュ関数 H を、 $H : \mathcal{K} \times \{0,1\}^* \to \mathbb{Z}_p$ の様にハッシュ関数 G と F と同様の鍵空間を持つとし、$\mathbb{G}$ と $\mathcal{K}$ の間に、両方向の計算を効率的にできる全単射が存在すると仮定することができる。また、例えば p を素数として、$\mathcal{K} = \mathbb{Z}_p$ の様に限定するならば、楕円曲線上の点からなる位数 p の群 $\mathbb{G}$ と $\mathbb{Z}_p$ の間の全単射 [17] の具体的な構成法も存在する。)。

提案する方式 $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ は 図 3.5 である。Verify の中の双線形写像の計算 $e(g_1, g_2)$ は、KeyGen の段階で前もって行っておき、検証鍵 VK の一部に入れておくことができる。ただし、ここでは [16] での記述法に習い、このようにそのまま記す。

構成要素：
$\mathbb{G}, \mathbb{G}_T$：位数が十分大きな素数 p の巡回群
$e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$：双線形写像
$H : \mathbb{G} \times \{0,1\}^* \to \mathbb{Z}_p, G : \mathcal{K} \times \mathbb{G} \to \mathbb{Z}_p, F : \mathcal{K} \times \mathbb{G} \to \{0,1\}^n$：全て TCRHF

$\mathsf{KeyGen}(1^\kappa)$：	$\mathsf{Sign}(SK, M)$：
$g \leftarrow \mathbb{G}$; $\alpha \leftarrow \mathbb{Z}_p$; $g_1 \leftarrow g^\alpha$	$s, r \leftarrow \mathbb{Z}_p$; $\sigma_2 \leftarrow g^r$
$g_2, h_1, h_2, u', u_1, \ldots, u_n \leftarrow \mathbb{G}$	$t \leftarrow H_{\sigma_2}(M)$; $t' \leftarrow G_k(\sigma_2)$
$U \leftarrow (u_1, \ldots, u_n)$	$m \leftarrow g^t h_1^s h_2^{t'}$; $m' \leftarrow F_k(m)$
$k \leftarrow \mathcal{K}$	Parse m' as $m'_1 m'_2 \ldots m'_n$
$SK \leftarrow g_2^\alpha, VK \leftarrow (g, g_1, g_2, h_1, h_2, u', U, k)$	(each of m'_i is the i-th bit of m').
Output (SK, VK).	$\sigma_1 \leftarrow g_2^\alpha \cdot (u' \prod_{i=1}^n u_i^{m'_i})^r$
	Output $\sigma \leftarrow (\sigma_1, \sigma_2, s)$.

$\mathsf{Verify}(VK, M, \sigma)$：
Parse σ as (σ_1, σ_2, s).
$t \leftarrow H_{\sigma_2}(M)$; $t' \leftarrow G_k(\sigma_2)$; $m \leftarrow g^t h_1^s h_2^{t'}$; $m' \leftarrow F_k(m)$
Parse m' as $m'_1 m'_2 \ldots m'_n$ (each of m'_i is the i-th bit of m').
Check $e(\sigma_1, g) \stackrel{?}{=} e(\sigma_2, u' \prod_{i=1}^n u_i^{m'_i}) \cdot e(g_1, g_2)$.
Output `accept` if this holds. Otherwise output `reject`.

図 3.5: TCRHF を用いる CDH 仮定に基づいた具体的な署名方式

安全性　以下は、ターゲット衝突困難ハッシュ関数を用いた署名方式 (図 3.5) の安全性についての記述である。

系 3.2. 以下の条件が成り立つとき、図 *3.5* に示した電子署名方式は (t, q, ϵ)-*SEUF-CMA* 安全性を持つ。

- $\mathbb{G}$ において $(t, \epsilon/48(n+1)q)$-*CDH* 仮定が成り立つ。
- H、G、及び F はそれぞれ、$(t, \epsilon/6q)$、$(t, \epsilon/6q)$、$(t, \epsilon/6q)$-*TCRHF* である。

系 3.2 の証明　既に Waters により [62] において、Water 署名は $\mathbb{G}$ において $(t, \epsilon/8(n+1)q)$-CDH 仮定が成り立つならば (t, q, ϵ)-EUF-CMA 安全性を持つことが示されている (ただし、n は Waters 署名のメッセージ長 (n-bit) であり、本節で提案する方式では、TCRHF F の出力長に対応する。)(定理 3.1)。本証明ではこの結果を利用する。すなわち、Waters 署名は $\mathbb{G}$ において $(t, \epsilon/48(n+1)q)$ が成り立つ場合 $(t, q, \epsilon/6)$-EUF-CMA 安全性を持つ。また、このとき明らかに $\mathbb{G}$ において $(t, \epsilon/3)$-DL は成り立っている。

よって、あとは Waters 署名が *simulatable-partitioned* の定義を満たすことを示せばよい。以下は Waters 署名の鍵生成アルゴリズム KeyGen である。

KeyGen：　1. $g, g_2 \in \mathbb{G}$ をそれぞれ一様ランダムに選ぶ。

2. $\alpha \in \mathbb{Z}_p$ を一様ランダムに選び、$g_1 \leftarrow g^\alpha$ を計算する。
3. $u', u_1, u_2, \ldots, u_n \ \in \mathbb{G}$ をそれぞれ一様ランダムに選ぶ。
4. $sk \leftarrow g_2^\alpha$、$vk \leftarrow (g, g_1, g_2, u', u_1, u_2, \ldots, u_n)$ とする。
5. (sk, vk) を出力する。

ここで、この KeyGen の 3 行以降を次の様に書き換え、KeyGen′ として定義する (1 行目と 2 行目は同一である。)。

KeyGen′ :　1. $g, g_2 \in \mathbb{G}$ をそれぞれ一様ランダムに選ぶ。
2. $\alpha \in \mathbb{Z}_p$ を一様ランダムに選び、$g_1 \leftarrow g^\alpha$ を計算する。
3. $\beta', \beta_1, \beta_2, \ldots, \beta_n \ \in \mathbb{Z}_p$ をそれぞれ一様ランダムに選び、$u' \leftarrow g^{\beta'}, u_1 \leftarrow g^{\beta_1}, u_2 \leftarrow g^{\beta_2}, \ldots, u_n \leftarrow g^{\beta_n}$ を計算する。
4. $sk' \leftarrow g_2^\alpha$、$vk' \leftarrow (g, g_1, g_2, u', u_1, u_2, \ldots, u_n)$、$TD \leftarrow (\beta', \beta_1, \beta_2, \ldots, \beta_n)$ とする。
5. (sk', vk', TD) を出力する。

上記の様に定義された KeyGen′ から出力された (sk', vk') の対が KeyGen から出力された署名鍵と検証鍵の対 (sk, vk) と同一の分布を持つことは明らかである。

Waters の署名方式が *partitioned* 署名であることは [16] で示されている。すなわち、Waters の署名方式は *simulatable-partitioned* 署名の性質 1 と性質 2 を満たす。Waters の署名方式の署名アルゴリズム S_1 及び S_2 は以下の様に記述できる。

$$\begin{aligned} \sigma_1 &\leftarrow \mathsf{S}_1(sk, m, r) = (sk) \cdot (u' \textstyle\prod_{i=1}^n u_i^{m_i})^r \quad \in \mathbb{G} \\ \sigma_2 &\leftarrow \mathsf{S}_2(sk, r) = g^r \quad \in \mathbb{G} \end{aligned}$$

ただし、$m_i \in \{0,1\}$ は $m \in \{0,1\}^n$ の i 番目のビットを表す。また、$u', u_1, \ldots, u_n \in \mathbb{G}$、$r \in \mathbb{Z}_p$、そして $(sk) \in \mathbb{G}$ である。ここで、トラップドア $TD = (\beta', \beta_1, \ldots, \beta_n)$、署名の一部 σ_2 を用いて、σ_1 を以下の様に表すことができる。

$$\begin{aligned} \sigma_1 &= (sk) \cdot (u' \textstyle\prod_{i=1}^n u_i^{m_i})^r = (sk) \cdot (g^{r\beta'} \prod_{i=1}^n (g^{r\beta_i m_i})) \\ &= (sk) \cdot (g^r)^{\beta' + \sum_{i=1}^n \beta_i m_i} = (sk) \cdot (\sigma_2)^{\beta' + \sum_{i=1}^n \beta_i m_i} \end{aligned}$$

この等式の最右辺は、内部の乱数 r の整合性が取れている正しい σ_1 は、r そのものを与えられなくても sk, m, σ_2, TD のみから計算できることを示している。よって、この等式で決定的アルゴリズム $\mathsf{S}_1'(sk, m, \sigma_2, TD)$ を定義できる。よって、S_1'、KeyGen′、及び TD が *simulatable-partitioned* 署名の性質 3 を満たしていることを示した。また、$\mathsf{S}_2(sk, r) = g^r$ は $\mathbb{Z}_p$ から $\mathbb{G}$ への全単射であり、S_2 の計算に sk が不要なことから、Waters の署名方式が *simulatable-partitioned* 署名の性質 4 を満たすことは明らかである。故に、Waters の署名方式が *simulatable-partitioned* 署名の全ての性質を持っているということができる。よって、Waters の署名方式は *simulatable-partitioned* 署名である。

以上の議論で、系 3.2 の条件が全て成り立つ場合、定理 3.3 の条件が満たされることを示した。よって、図 3.5 の署名方式は SEUF-CMA 安全性を持つことが示された。 □

非対称な双線形写像の利用　非対称な双線形写像 (2.4 節) を使い、署名の構成要素 σ_1 と σ_2 を両方とも $\mathbb{G}_1$ の要素とすることで、検証鍵の大部分の要素が $\mathbb{G}_2$ の要素になり、サイズが大きくなってしまうことを犠牲にして署名のサイズを小さくできることができる。また、もう一つの可能性として、検証鍵の要素の大部分 (特に検証鍵の U の要素) を $\mathbb{G}_1$ の要素とし、署名の要素が $\mathbb{G}_2$ の要素となり署名サイズを犠牲にして検証鍵のサイズを小さくするという方法も考えられる。

3.3　強化ターゲット衝突困難ハッシュ関数を用いた方式

本節では、3.2 節で示した方式とは別の衝突困難ハッシュ関数を用いないでスタンダードモデルで CDH 仮定に基づいて SEUF-CMA 安全な電子署名を得る方式を示す。本節の方式では、ターゲット衝突困難ハッシュ関数よりも強い安全性を持つ強化ターゲット衝突困難ハッシュ関数を用いて目的を達成する。興味深いことに、本節で最終的に示す署名方式は、BSW 署名とハッシュ関数の安全性が違うだけでそれ以外のパラメータ、署名サイズや計算コストは全く同一となる。すなわち、BSW 署名から衝突困難ハッシュ関数を置き換えただけの構成になっている。従って、本節の結果からいえることは、BSW 署名では本質的には衝突困難ハッシュ関数は必要ないということである。

提案方式を得るまでの基本的な流れは、BSW 署名やターゲット衝突困難ハッシュ関数を用いた署名方式 (3.2) の場合と全く同様である。すなわち、*simulatable-partitioned* 署名の性質を全て満たす EUF-CMA 安全な署名方式を SEUF-CMA 安全性を持つ方式へと変換し (3.3.1 節)、Waters 署名に対しその変換を適用する (3.3.2 節)。

3.3.1　強偽造不可能性を持つ方式への変換

本節では強化ターゲット衝突困難ハッシュ関数を用いた任意の EUF-CMA 安全性を持つ *simulatable-partitioned* 署名を SEUF-CMA 安全性を持つ署名方式へと一般的に変換する方法を示す。

$\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ を EUF-CMA 安全性を持ち、*simulatable-partitioned* 署名の性質を全て満たす署名方式とする。すなわち、性質 1 により、署名アルゴリズム Sign が二つの決定的アルゴリズム S_1 と S_2 に分割できる。p を十分大きな素数、$\mathbb{G}$ を位数が p の巡回群とする。*simulatable-partitioned* 署名の性質 2 が成り立つように、$\mathbb{G}$ の要素は一意に特定できるビットでの表し方があるとする。$H : \mathcal{S}_2 \times \{0,1\}^* \to \mathbb{Z}_p$ を強化ターゲット衝突困難ハッシュ関数、$F : \mathcal{K} \times \mathbb{G} \to \mathcal{M}$ をターゲット衝突困難ハッシュ関数とする。ただし、全てのハッシュ関数で、定義域のうち最初の集合はハッシュ鍵空間、そして後の集合はハッシュ関数の入力空間とする。また、H のハッシュ鍵空間は、$\mathcal{S}_2$(S_2 の出力空間) であるとする。

提案するターゲット衝突困難ハッシュ関数を用いた変換は図 3.4 である ($\Sigma_{new} = (\mathsf{KeyGen}_{new}, \mathsf{Sign}_{new}, \mathsf{Verify}_{new})$ は変換後に得られる署名方式)。

安全性　以下は、安全性についての記述である。

<table>
<tr><td colspan="2">構成要素：
$\mathsf{S}_1, \mathsf{S}_2$：変換前の署名方式 Σ の simulatable-partitioned の性質によって
分割された 2 つの署名アルゴリズム
$\mathbb{G}$：位数が十分大きな素数 p の巡回群
$H : \mathcal{S}_2 \times \{0,1\}^* \to \mathbb{Z}_p$：eTCRHF
$F : \mathcal{K} \times \mathbb{G} \to \mathcal{M}$：TCRHF</td></tr>
<tr><td>$\mathsf{KeyGen}_{new}(1^\kappa)$:
$(sk, vk) \leftarrow \mathsf{KeyGen}(1^\kappa)$
$g, h \leftarrow \mathbb{G}$; $k \leftarrow \mathcal{K}$
$SK \leftarrow (sk)$, $VK \leftarrow (vk, g, h, k)$
Output (SK, VK).</td><td>$\mathsf{Sign}_{new}(SK, M)$:
$s \leftarrow \mathbb{Z}_p$; $r \leftarrow \mathcal{R}$; $\sigma_2 \leftarrow \mathsf{S}_2(sk, r)$
$t \leftarrow H_{\sigma_2}(M)$; $m \leftarrow g^t h^s$
$m' \leftarrow F_k(m)$; $\sigma_1 \leftarrow \mathsf{S}_1(sk, m', r)$
Output $\sigma \leftarrow (\sigma_1, \sigma_2, s)$.</td></tr>
<tr><td colspan="2">$\mathsf{Verify}_{new}(VK, M, \sigma)$:
Parse σ as (σ_1, σ_2, s).
$t \leftarrow H_{\sigma_2}(M)$; $m \leftarrow g^t h^s$; $m' \leftarrow F_k(m)$
Output <code>accept</code> if $\mathsf{Verify}(vk, m', (\sigma_1, \sigma_2)) =$ <code>accept</code>. Otherwise output <code>reject</code>.</td></tr>
</table>

図 3.6: eTCRHF を用いた変換の構成

定理 3.4. 以下の条件が成り立つとき、*eTCRHF* を用いた変換 *(*図 *3.6)* を適用した後の方式 Σ_{new} は (t, q, ϵ)-*SEUF-CMA* 安全性を満たす電子署名方式である。

- 構成要素の電子署名方式 Σ は $(t, q, \epsilon/4)$-*EUF-CMA* 安全性を満たす *simulatable-partitioned* 署名である。
- $\mathbb{G}$ において $(t, \epsilon/4)$-*DL* 仮定が成り立つ。
- H は $(t, \epsilon/4q)$-*eTCRHF* である。
- F は $(t, \epsilon/4q)$-*TCRHF* である。

構成及び証明のアイデア　ターゲット衝突困難ハッシュ関数 F は、実質的には $\mathbb{G}$ の要素のエンコードの方法と考えることができるため、本節での変換法は、変換の前提条件と、ハッシュ関数の安全性が違う以外は、本質的には BSW 変換と全く同一である。したがって、以下に示す証明の流れは、F に関する部分を除き、BSW 変換 [16] の安全性証明の流れとよく似ている。ただし、我々の変換はやはり *simulatable-partitioned* 署名の性質 3 の KeyGen'、S_1'、及び TD を用いなければならない場合 (定理 3.4 証明の際の Type 4 攻撃者) がある。しかしそれに対するシミュレータの動作は、定理 3.3 の証明の際の Type 6 攻撃者に対するシミュレータの動作とほぼ同様である。

定理 3.4 の証明　$\mathcal{A}$ を、提案方式 Σ_{new} の (t, q, ϵ)-SEUF-CMA 安全性を破る攻撃者とする。SEUF-CMA ゲームの *Setup* の段階において、$\mathcal{A}$ は公開鍵 $VK = (vk, g, h, k)$ を

与えられる。また、q を $\mathcal{A}$ が発行する署名クエリの総数とし、*Queries* の段階において、$\mathcal{A}$ は i 回目 $(i \in \{1, \ldots, q\})$ の署名クエリとして M_i を発行し、それに対応する署名 $\sigma_i = (\sigma_{i,1}, \sigma_{i,2}, s_i)$ を与えられるとする。一般性を失わずに、$q > 0$ とする。*Output* の段階において、$\mathcal{A}$ はメッセージと偽造署名の対 $(\hat{M}, \hat{\sigma} = (\hat{\sigma_1}, \hat{\sigma_2}, \hat{s}))$ を出力するものとする。$\hat{t} = H_{\hat{\sigma_2}}(\hat{M})$、$\hat{m} = g^{\hat{t}} h^{\hat{s}}$、及び、$\hat{m}' = F_k(\hat{m})$ であると定義する。$\mathcal{A}$ の出力する偽造署名とメッセージと偽造署名 $(\hat{M}, \hat{\sigma})$ のタイプによって $\mathcal{A}$ を以下の 4 つに分類する。

Type 1. $\forall i \in \{1, \ldots, q\} : \hat{m}' \neq m'_i$.

Type 2. $\exists i \in \{1, \ldots, q\} : \hat{m}' = m'_i \wedge \hat{m} \neq m_i$.

Type 3. $\exists i \in \{1, \ldots, q\} : \hat{m}' = m'_i \wedge \hat{m} = m_i \wedge \hat{t} \neq t_i$.

Type 4. $\exists i \in \{1, \ldots, q\} : \hat{m}' = m'_i \wedge \hat{m} = m_i \wedge \hat{t} = t_i$.

$\mathcal{A}$ が偽造に成功する場合、$\mathcal{A}$ により出力される $(\hat{M}, \hat{\sigma})$ は、必ず上記 4 タイプのどれかにあてはまる。

以下の議論では、それぞれのタイプの SEUF-CMA を破る攻撃者 $\mathcal{A}$ に対し、SEUF-CMA ゲームをシミュレーションを行いながら Σ_{new} の構成要素の安全性を破ることができるシミュレータ $\mathcal{B}_i (i \in \{1, \ldots, 4\})$ を構成できることを示す。$\mathcal{A}$ がタイプ 1 の攻撃者の場合、$\mathcal{B}_1$ は構成要素である署名方式 Σ の EUF-CMA 安全性を、タイプ 2 の攻撃者の場合、$\mathcal{B}_2$ は TCRHF の安全性を、タイプ 3 の攻撃者の場合、$\mathcal{B}_3$ は $\mathbb{G}$ における離散対数問題を、そしてタイプ 4 の攻撃者の場合、$\mathcal{B}_4$、$\mathcal{B}_6$ は eTCRHF の安全性を破ることができる。シミュレータは $\mathcal{A}$ に対し SEUF-CMA ゲームのシミュレーションを始める前にコインを振り、$\mathcal{A}$ がどのタイプの攻撃者かを推測し、$\mathcal{B}_1, \ldots, \mathcal{B}_4$ のどのシミュレータとして動くかを決めることができる。以下、それぞれのタイプに対するシミュレータの具体的構成法を示す。

Type 1:

$\mathcal{A}$ を Σ_{new} の (t, q, ϵ)-SEUF-CMA 安全性を破るタイプ 1 の攻撃者であると仮定する。この $\mathcal{A}$ を用いて構成要素である電子署名方式 Σ の (t, q, ϵ)-EUF-CMA 安全性を破ることのできる $\mathcal{B}_1$ を構成する。$\mathcal{B}_1$ は最初に vk を与えられ、EUF-CMA ゲームの勝利条件を満たすような $(\hat{m}', \hat{\sigma} = (\hat{\sigma_1}, \hat{\sigma_2}))$ を出力する。$\mathcal{B}_1$ は以下の様に $\mathcal{A}$ に対し SEUF-CMA ゲームのシミュレートを行いつつ利用して自身の EUF-CMA チャレンジャー $\mathcal{C}$ との間で EUF-CMA ゲームを行う。

Setup. $\mathcal{B}_1$ は以下の様にして VK を作成し、$\mathcal{A}$ に渡す。

1. $g \in \mathbb{G}$、及び $k \in \mathcal{K}$ を一様ランダムに選ぶ。
2. $a \in \mathbb{Z}_p^*$ を一様ランダムに選び、$h \leftarrow g^a$ を計算する。
3. $VK \leftarrow (vk, g, h, k)$ とし、$\mathcal{A}$ に渡す。

Queries. $\mathcal{B}_1$ は $\mathcal{A}$ の発行する署名クエリ $M_{i\in\{1,\ldots,q\}}$ に対し σ_i を作成し応答する。

1. $w_i \in \mathbb{Z}_q$ を一様ランダムに選び、$m_i \leftarrow g^{w_i}$ を計算する。
2. $m'_i \leftarrow F_k(m_i)$ を計算する。
3. m'_i を自身の i 回目の署名クエリとして $\mathcal{C}$ に問い合わせ、m'_i の署名 $(\sigma_{i,1}, \sigma_{i,2})$ を受け取る。
4. $t_i \leftarrow H_{\sigma_{i,2}}(M_i)$ を計算する。
5. $s_i \leftarrow (w_i - t_i)/a$ を計算する。
6. $\sigma_i \leftarrow (\sigma_{i,1}, \sigma_{i,2}, s_i)$ とし、$\mathcal{A}$ に渡す。

Output. $\mathcal{A}$ がタイプ 1 のメッセージと偽造署名の対 $(\hat{M}, (\hat{\sigma_1}, \hat{\sigma_2}, \hat{s}))$ を出力する。$\mathcal{B}_1$ は Verify_{new} のアルゴリズムに沿って正しく $\hat{m}'$ を計算する。$\mathcal{C}$ に対し、$(\hat{m}', (\hat{\sigma_1}, \hat{\sigma_2}))$ を出力する。

$\hat{m}'$ が Verify_{new} のアルゴリズムに沿って正しく計算された場合、Σ_{new} の構成により、$\mathsf{Verify}_{new}(VK, \hat{M}, (\hat{\sigma_1}, \hat{\sigma_2}, \hat{s})) = \mathtt{accept}$ が成り立つならば、$\mathsf{Verify}(vk, \hat{m}', (\hat{\sigma_1}, \hat{\sigma_2})) = \mathtt{accept}$ が成り立つことになる。$\mathcal{A}$ はタイプ 1 の攻撃者であるため、全ての $i \in \{1, \ldots q\}$ について、$\hat{m}' \neq m'_i$ が成り立つ。従って、$\mathcal{B}_1$ は EUF-CMA ゲームに対して正しく動作する攻撃者になっている。故に、$\mathcal{A}$ がタイプ 1 の偽造に成功する場合はいつでも、$\mathcal{B}_1$ は構成要素である電子署名方式 Σ における新しいメッセージ $\hat{m}'$ の署名を出力でき、EUF-CMA ゲームに勝利することができる。

Type 2:

$\mathcal{A}$ を Σ_{new} の (t, q, ϵ)-SEUF-CMA 安全性を破るタイプ 2 の攻撃者であると仮定する。この $\mathcal{A}$ を用いて構成要素である $(t, \epsilon/q)$-TCRHF F の安全性を破ることのできる $\mathcal{B}_2$ を構成する。$\mathcal{B}_2$ は以下の様に $\mathcal{A}$ に対し SEUF-CMA ゲームのシミュレートを行いつつ利用して自身の TCR チャレンジャー $\mathcal{C}$ との間で F に関する TCR ゲームを行う。

Setup. $\mathcal{B}_2$ は以下の様にして VK を作成し、$\mathcal{A}$ に渡す。

1. インデックス $j \in \{1, \ldots, q\}$ を一様ランダムに選ぶ。
2. KeyGen を実行し、sk, vk を得る。
3. $g \in \mathbb{G}$ を一様ランダムに選ぶ。
4. $a \in \mathbb{Z}_p^*$ を一様ランダムに選び、$h \leftarrow g^a$ を計算する。
5. $\bar{w} \in \mathbb{Z}_p$ を一様ランダムに選び、$\bar{m} \leftarrow g^{\bar{w}}$ を計算する。
6. $\bar{m}$ を TCR ゲームの最初の入力として $\mathcal{C}$ に出力し、$k \in \mathcal{K}$ を受け取る。
7. $\bar{m}' \leftarrow F_k(\bar{m})$ を計算する。
8. $VK \leftarrow (vk, g, h, k)$ とし、$\mathcal{A}$ に渡す。$SK \leftarrow sk$ は保持しておく。

Queries. $\mathcal{B}_2$ は $\mathcal{A}$ の発行する署名クエリ $M_{i\in\{1,\ldots,q\}}$ に対し i の状態により以下の様に σ_i を作成し応答する。

$i \neq j$ **のとき** ：$\sigma_i \leftarrow \mathsf{Sign}_{new}(SK, M_i)$ を計算し、$\mathcal{A}$ に渡す。

それ以外のとき ：

1. $m_j \leftarrow \bar{m}$、$m'_j \leftarrow \bar{m}'$ とする。
2. $\mathsf{Sign}(sk, m'_j)$ を実行し、$(\sigma_{j,1}, \sigma_{j,2})$ を得る。
3. $t_j \leftarrow H_{\sigma_{j,2}}(M_j)$ を計算する。
4. $s_j \leftarrow (\bar{w} - t_j)/a$ を計算する。
5. $\sigma_j \leftarrow (\sigma_{j,1}, \sigma_{j,2}, s_j)$ とし、$\mathcal{A}$ に渡す。

Output. $\mathcal{A}$ がタイプ 2 のメッセージと偽造署名の対 $(\hat{M}, (\hat{\sigma}_1, \hat{\sigma}_2, \hat{s}))$ を出力する。$\mathcal{B}_2$ は Verify_{new} のアルゴリズムに沿って正しく $\hat{m}$ を計算する。$F_k(\hat{m}) = F_k(m_j) \wedge \hat{m} \neq m_j$ が成り立つならば、$\mathcal{B}_2$ は $\mathcal{C}$ に対し、$\hat{m}$ を出力する。成り立たない場合、$\mathcal{B}_2$ は TCR ゲームに勝利することを諦め停止する。

$\mathcal{A}$ はタイプ 2 の攻撃者であり、$\hat{m}' = m'_i$ は $F_k(\hat{m}) = F_k(m_i)$ を意味するため、$F_k(\hat{m}) = F_k(m_i) \wedge \hat{m} \neq m_i$ が成り立つ様なインデックス $i \in \{1, \ldots, q\}$ が少なくとも一つ存在する。従って、$\hat{m}$ は F について、ハッシュ鍵 k の下、m_i と同一のハッシュ値を持つ。$\mathcal{B}_2$ がその様なインデックス i を Setup の段階でインデックス j として選ぶ確率は少なくとも $1/q$ である。故に、$\mathcal{A}$ がタイプ 2 の偽造に成功する場合、$\mathcal{B}_2$ は少なくとも確率 $1/q$ で F についての TCR ゲームに勝利することができる。

Type 3:

$\mathcal{A}$ を Σ_{new} の (t, q, ϵ)-SEUF-CMA 安全性を破るタイプ 3 の攻撃者であると仮定する。この $\mathcal{A}$ を用いて $\mathbb{G}$ における (t, ϵ)-DL 仮定を破るシミュレータ $\mathcal{B}_3$ を構成する。$\mathcal{B}_3$ は最初に離散対数問題 (g, X) を与えられ、以下の様に $\mathcal{A}$ に対し SEUF-CMA ゲームのシミュレートを行いつつ利用して離散対数問題の解 $\log_g X$ を出力する。

Setup. $\mathcal{B}_3$ は以下の様にして VK を作成し、$\mathcal{A}$ に渡す。

1. g を VK の要素とし、$h \leftarrow X$ とする。
2. KeyGen を実行し、sk, vk を得る。
3. $k \in \mathcal{K}$ を一様ランダムに選ぶ。
4. $VK \leftarrow (vk, g, h, k)$ とし、$\mathcal{A}$ に渡す。$SK \leftarrow sk$ は保持しておく。

Queries. $\mathcal{B}_3$ は $\mathcal{A}$ の発行する署名クエリ $M_{i\in\{1,\ldots,q\}}$ に対し、$\sigma_i \leftarrow \mathsf{Sign}_{new}(SK, M_i)$ を計算し、$\mathcal{A}$ に渡す。

Output. $\mathcal{A}$ はタイプ 3 のメッセージと偽造署名の対 $(\hat{M}, (\hat{\sigma}_1, \hat{\sigma}_2, \hat{s}))$ を出力する。$\mathcal{B}_3$ は Verify_{new} のアルゴリズムに沿って正しく $\hat{t}$ を計算する。$\log_g X \leftarrow (\hat{t} - t_i)/(s_i - \hat{s})$ を計算し、DL 問題の解として出力する。

$\mathcal{A}$ はタイプ 3 の攻撃者であるため、$\hat{m} = m_i \wedge \hat{t} \neq t_i$ が成り立つ様なインデックス $i \in \{1, \ldots, q\}$ が少なくとも一つ存在し、$\mathcal{B}_3$ は Output の段階でその様な i を知ることができる。$\hat{m} = m_i$ は $g^{\hat{t}} h^{\hat{s}} = g^{t_i} h^{s_i} = g^{\hat{t}} X^{\hat{s}} = g^{t_i} X^{s_i}$ を意味するため、この等式から、底を g とする対数を考えることで、X の底を g とする対数、すなわち与えられた離散対数問題 (g, X) の解を得ることができる。$\hat{t} \neq t_i$ によって、$s_i - \hat{s} \neq 0$ であることは保障されている。故に、$\mathcal{A}$ がタイプ 3 の偽造に成功する場合はいつでも、$\mathcal{B}_3$ は与えられた $\mathbb{G}$ における離散対数問題を解くことができる。

Type 4:

$\mathcal{A}$ を Σ_{new} の (t, q, ϵ)-SEUF-CMA 安全性を破るタイプ 4 の攻撃者であると仮定する。この $\mathcal{A}$ を用いて、構成要素である $(t, \epsilon/q)$-eTCRHF H の安全性を破ることのできる $\mathcal{B}_4$ を構成する。$\mathcal{B}_4$ は以下の様に $\mathcal{A}$ に対し SEUF-CMA ゲームのシミュレートを行いつつ利用して自身の eTCR チャレンジャー $\mathcal{C}$ との間で H に関する eTCR ゲームを行う。

Setup. $\mathcal{B}_4$ は以下の様にして VK を作成し、$\mathcal{A}$ に渡す。

1. インデックス $j \in \{1 \ldots q\}$ を一様ランダムに選ぶ。
2. KeyGen' を実行し、$sk,\ vk,\ TD$ を得る。
3. $g, h, \in \mathbb{G}$、及び、$k \in \mathcal{K}$ をそれぞれ一様ランダムに選ぶ。
4. $VK \leftarrow (vk, g, h, k)$ を $\mathcal{A}$ に渡す。$SK \leftarrow sk$ は保持しておく。

Queries. $\mathcal{B}_4$ は $\mathcal{A}$ の発行する署名クエリ $M_{i \in \{1,\ldots,q\}}$ に対し i の状態により以下の様に σ_i を作成し応答する。

$i \neq j$ のとき ：$\sigma_i \leftarrow \mathsf{Sign}_{new}(SK, M_i)$ を計算し、$\mathcal{A}$ に渡す。

それ以外のとき ：

1. M_j を eTCR ゲームの最初の入力として $\mathcal{C}$ に出力し、$\bar{\sigma_2} \in \mathcal{S}_2$ を受け取る。
2. $\sigma_{j,2} \leftarrow \bar{\sigma_2}$ とする。
3. $t_j \leftarrow H_{\sigma_{j,2}}(M_j)$ を計算する。
4. $s_j \in \mathbb{Z}_p$ を一様ランダムに選ぶ。
5. $m_j \leftarrow g^{t_j} h^{s_j}$ を計算する。
6. $\sigma_{j,1} \leftarrow \mathsf{S}'_1(sk, m'_j, \bar{\sigma_2}, TD)$ を計算する。
7. $\sigma_j \leftarrow (\sigma_{j,1}, \sigma_{j,2}, s_j)$ とし、$\mathcal{A}$ に渡す。

Output. $\mathcal{A}$ がタイプ 4 のメッセージと偽造署名の対 $(\hat{M}, (\hat{\sigma_1}, \hat{\sigma_2}, \hat{s}))$ を出力する。$H_{\hat{\sigma_2}}(\hat{M}) = H_{\sigma_{j,2}}(M_j) \wedge (\hat{M}, \sigma_2) \neq (M_j, \sigma_{2,j})$ が成り立つならば、$\mathcal{B}_4$ は $\mathcal{C}$ に対し $(\hat{M}, \hat{\sigma_2})$ のペアを出力する。成り立たない場合、$\mathcal{B}_4$ は eTCR ゲームに勝利することを諦め停止する。

$\sigma_{j,2} = \bar{\sigma_2}$ を作る際に使われた $r_j = \bar{r}$ を知らないで内部の r_j が $\sigma_{j,2}$ と整合性がとれている正しい $\sigma_{j,1}$ を計算するために、KeyGen' と S'_1 が使われていることに注意されたい。

$\mathcal{A}$ が SEUF-CMA ゲームに対し勝利する場合、SEUF-CMA ゲームの定義より、全てのインデックス $i \in \{1 \dots q\}$ について、$(\hat{M}, \hat{\sigma_1}, \hat{\sigma_2}, \hat{s}) \neq (M_i, \sigma_{i,1}, \sigma_{i,2}, s_i)$ が成り立つ。$\mathcal{A}$ はタイプ 4 の攻撃者なので、$\hat{t} = t_i$、すなわち、$H_{\hat{\sigma_2}}(\hat{M}) = H_{\sigma_{i,2}}(M_i)$ が成り立つインデックス $i \in \{1, \dots, q\}$ が必ず存在する。また、同じインデックス i の下では、必ず $(\hat{M}, \hat{\sigma_2}) \neq (M_i, \sigma_{i,2})$ が成り立っている。なぜならば、$(\hat{M}, \hat{\sigma_2}) = (M_i, \sigma_{i,2})$ が成り立つと仮定すると、*simulatable-partitioned* 署名 (定義 3.2) の性質 2 により、$\hat{\sigma_1} = \sigma_{i,1}$ が得られる。また、$\hat{m} = m_i \wedge \hat{t} = t_i$ より $\hat{s} = s_i$ も成り立ってしまうため、このインデックス i の下では $(\hat{M}, \hat{\sigma_1}, \hat{\sigma_2}, \hat{s}) = (M_i, \sigma_{i,1}, \sigma_{i,2}, s_i)$ が成立していることになり、$\mathcal{A}$ が SEUF-CMA 攻撃者であることに矛盾するからである。

以上をまとめると、$(\hat{M}, \hat{\sigma_2}) \neq (M_i, \sigma_{i,2}) \wedge H_{\hat{\sigma_2}}(\hat{M}) = H_{\sigma_{i,2}}(M_i)$ が成り立つ様なインデックス $i \in \{1, \dots, q\}$ が存在する。すなわち、この様なインデックス i の下では、最初に $\mathcal{B}_4$ が選んだ M_j と $\mathcal{C}$ により選ばれたハッシュ鍵 $\sigma_{2,j} = \bar{\sigma_2}$ のペア $(M_j, \sigma_{j,2})$ と、$\mathcal{B}_4$ が Output の段階で出力したメッセージとハッシュ鍵のペア $(\hat{M}, \hat{\sigma_2})$ が eTCR ゲームを破る条件を満たしている。$\mathcal{B}_4$ がその様なインデックス i を Setup の段階でインデックス j として選ぶ確率は少なくとも $1/q$ である。故に、$\mathcal{A}$ がタイプ 4 の偽造に成功する場合、$\mathcal{B}_4$ は少なくとも確率 $1/q$ で H についての eTCR ゲームに勝利することができる。

以上の議論で、4 タイプ全ての攻撃者に対してのシミュレータの構成法を示した。全てのタイプの攻撃者に対するシミュレーションは完全である。それぞれ、タイプ 1 の攻撃者は構成要素である Σ の EUF-CMA 安全性を、タイプ 2 は TCRHF の安全性を、タイプ 3 は DL 仮定を、タイプ 4 は eTCRHF の安全性を破ることに使えることを示した。以上により、定理 3.4 は証明された。　□

3.3.2　CDH 仮定に基づく具体的な署名方式

本節では、Waters 署名に対し、前節の強化ターゲット衝突困難ハッシュ関数を用いた変換を適用して得られた署名方式を示す。

具体的な構成法の記述は次の通りである。p を十分に大きな素数とし、$\mathbb{G}$ を位数 p の双線形写像を持つような巡回群とする。$e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ を双線形写像とする。$H : \mathbb{G} \times \{0,1\}^* \to \mathbb{Z}_p$ を強化ターゲット衝突困難ハッシュ関数、$F : \mathcal{K} \times \mathbb{G} \to \{0,1\}^n$ をターゲット衝突困難ハッシュ関数とする。Waters の署名方式の場合、$\mathcal{S}_2 = \mathbb{G}$ となっている。そこで、ハッシュ関数 H の鍵空間が $\mathbb{G}$ とできることも仮定する。$\mathbb{G}$ の各要素は、それぞれ一意のビット列にエンコードできると仮定する。

提案する方式 $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ は 図 3.7 である。Verify の中の双線形写像の計算 $e(g_1, g_2)$ は、KeyGen の段階で前もって行っておき、検証鍵 VK の一部に入れておくことができる。

前節の変換法がハッシュ関数の安全性と変換の適用条件が違うだけで BSW 変換と

本質的には同様の変換法であったが、変換結果である本節での署名方式は BSW 署名とハッシュ関数 H の安全性が違うだけであり、その他のパラメータや計算は本質的には全て同じとなっている。

構成要素：
$\mathbb{G}$, $\mathbb{G}_T$：位数が十分大きな素数 p の巡回群
$e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$：双線形写像
$H : \mathbb{G} \times \{0,1\}^* \to \mathbb{Z}_p$：eTCRHF $F : \mathcal{K} \times \mathbb{G} \to \{0,1\}^n$：全て TCRHF

$\mathsf{KeyGen}(1^\kappa)$：	$\mathsf{Sign}(SK, M)$：
$g \leftarrow \mathbb{G}$; $\alpha \leftarrow \mathbb{Z}_p$; $g_1 = g^\alpha$	$s, r \leftarrow \mathbb{Z}_p$; $\sigma_2 \leftarrow g^r$
g_2, h, u', $u_1, \ldots, u_n \leftarrow \mathbb{G}$	$t \leftarrow H_{\sigma_2}(M)$; $m \leftarrow g^t h^s$; $m' \leftarrow F_k(m)$
$U \leftarrow (u_1, \ldots, u_n)$	Parse m' as $m'_1 m'_2 \ldots m'_n$
$k \leftarrow \mathcal{K}$	(each of m'_i is the i-th bit of m').
$SK \leftarrow g_2^\alpha, VK \leftarrow (g, g_1, g_2, h, u', U, k)$	$\sigma_1 \leftarrow g_2^\alpha \cdot (u' \prod_{i=1}^n u_i^{m'_i})^r$
Output (SK, VK).	Output $\sigma \leftarrow (\sigma_1, \sigma_2, s)$.

$\mathsf{Verify}(VK, M, \sigma)$：
Parse σ as (σ_1, σ_2, s).
$t \leftarrow H_{\sigma_2}(M)$; $m \leftarrow g^t h^s$; $m' \leftarrow F_k(m)$
Parse m' as $m'_1 m'_2 \ldots m'_n$ (each of m'_i is the i-th bit of m').
Check $e(\sigma_1, g) \stackrel{?}{=} e(\sigma_2, u' \prod_{i=1}^n u_i^{m'_i}) \cdot e(g_1, g_2)$.
Output `accept` if this holds. Otherwise output `reject`.

図 3.7: eTCRHF を用いる CDH 仮定に基づいた具体的な署名方式

安全性　以下は、強化ターゲット衝突困難ハッシュ関数を用いた署名方式 (図 3.5) の安全性についての記述である。証明の流れは、系 3.2 と全く同じとなるため、証明は省略する。

系 3.3. 以下の条件が成り立つとき、図 *3.7* に示した電子署名方式は (t, q, ϵ)-*SEUF-CMA* 安全性を持つ。

- $\mathbb{G}$ において $(t, \epsilon/32(n+1)q)$-*CDH* 仮定が成り立つ。
- H は $(t, \epsilon/4q)$-*eTCRHF* である。
- F は $(t, \epsilon/4q)$-*TCRHF* である。

Chapter 4　関連研究

本章では本研究の関連研究を大きく 3 つの話題、すなわち、スタンダードモデルで証明可能安全性を持ち、かつ実用に耐えうるような効率性を持つような署名の代表的なもの (4.1 節)、提案方式を得るための途中の段階として考えたような、EUF-CMA 安全性を持つ署名方式を SEUF-CMA 安全性を満たすものへと変換する方法 (4.2 節)、そして、本研究での目的である電子署名の中でのハッシュ関数の衝突困難性を取り除こうとする研究 (4.3 節) に分けて紹介する。

4.1　スタンダードモデルでの電子署名方式

Cramer-Shoup　1999 年、Cramer と Shoup [25] は、スタンダードモデルの下、強 RSA 仮定に基づく EUF-CMA 安全性を示すことのできる方式を示した。これ以前の署名方式は、スタンダードモデルの下で証明可能安全性を有していても、鍵や署名サイズなどのパラメータが大きい、署名作成や検証にかかる計算コストが実用とは程遠いなど、実用に耐えうるような効率を持つ方式は存在していなかったが、Cramer-Shoup 署名は、実用化されている RSA [2] や RSA-PSS [8] などと比べ、署名サイズは 2 倍程度、署名計算速度も数倍程度という方式であった。実際は SEUF-CMA 安全性も満たすことが知られている。

Fischlin　2003 年、Fischlin [32] は Cramer-Shoup 署名 [25] を改良し、署名サイズが約半分程度にできる方式を提案した。スタンダードモデルの下、強 RSA 仮定に基づいて、EUF-CMA 安全性を証明できる。

Boneh-Boyen　2004 年、Boneh と Boyen [10] はスタンダードモデルの下、SDH 仮定に基づき SEUF-CMA 安全性を証明できる方式を示した。この方式は、同程度の安全性を仮定する場合、DSA とほぼ同等の署名長を実現できるという特徴を持っている。他の様々な機能付き署名の構成の基礎として用いられることが多い。

SDH 仮定という著者らが導入した困難性の仮定は、DL 仮定や、CDH 仮定よりも強い仮定であるとして知られている。ただし、この仮定は、パラメータの選び方などにより簡単に破れてしまう場合もあることが示されている [22, 40]。

Waters　2005 年、Waters [62] はスタンダードモデルの下、CDH 仮定に基づき EUF-CMA 安全性を証明できる方式を示した。3.1.1 節において詳しく説明している。

Boneh-Shen-Waters　2006 年、Boneh、Shen、Waters [16] は Waters 署名 [62] に変換を加え、スタンダードモデルの下、CDH 仮定に基づき SEUF-CMA 安全性を証明で

きる方式を示した。3.1.2 節において詳しく説明している。

その他の方式　この他、Camenisch と Lysyanskaya による強 RSA 仮定に基づく方式 [19] や、Okamoto による SDH 仮定に基づく方式 [50] などがある。これらの方式は必ずしも SEUF-CMA 安全性をスタンダードモデルで示せる方式であることを目的として提案されたわけではないため、本節で紹介した方式よりも署名サイズや、計算コストの面で効率が悪い。

4.2　EUF-CMA 安全性から SEUF-CMA 安全性への変換

Boneh-Shen-Waters　Boneh、Shen、Waters [16] は、*partitioned* 署名という性質を持つ特殊なクラスにあてはまる EUF-CMA 安全性を持つ署名方式を SEUF-CMA 安全性を持つ電子署名へと変換する方法 (BSW 変換) を示した。これについては 3.1.2 節において詳しく説明している。

Teranishi-Oyama-Ogata　Teranishi、Oyama、Ogata [60] は、BSW 変換で安全性の他に仮定された *partitioned* 署名の制約を取り除き、一般的な EUF-CMA 安全性を持つ署名方式を SEUF-CMA 安全性へと変換する方法を 2 つ示した。変換の構成法は BSW とよく似ている。

1 つの変換の構成はランダムオラクルモデルでのもので、変換前の署名の EUF-CMA 安全性と DL 仮定に基づいて変換後の署名方式の SEUF-CMA 安全性を証明できる。署名サイズ、変換の効率は BSW 変換の場合と同様となる。

もう 1 つの変換はスタンダードモデルのもので、変換前の署名の EUF-CMA 安全性と DL 仮定、そしてハッシュ関数の衝突困難性を利用し、変換後の署名方式の SEUF-CMA 安全性を証明できる。変換の計算コストのオーバーヘッドは BSW より若干良くできるものの、変換による署名サイズのオーバーヘッドは BSW よりも長くなる。

Steinfeld-Pieprzyk-Wang　Steinfeld、Pierprzyk、Wang [59] は、Teranishi らによる変換 [60] と同様、一般的な EUF-CMA 安全性を持つ署名方式を SEUF-CMA 安全性へと変換する方法を示した。変換の構成では BSW 変換の構成に変形を加えさらに少し一般化し、Randomized Trapdoor Hash Function (RTHF) という特殊なハッシュ関数を考えている。この RTHF は、衝突困難ハッシュ関数と DL 仮定や RSA 仮定に基づいて作ることができることが同論文内に示されている。

DL 仮定を利用して構成した RTHF を使うと、Teranishi らの方法と比べ、変換による署名サイズのオーバーヘッドは同等にできるものの、計算コストのオーバーヘッドは大きくなってしまう。

しかし、変換自体を RSA 仮定という素因数分解系の問題の困難さを用いて構成するという選択肢を与えることができる点がこの研究の貢献であるといえる。

Huang-Wong-Zhao　Huang、Wong、Zhao [37] は、Teranishi らによる方法と同様、一般的な EUF-CMA 安全性を持つ署名方式を SEUF-CMA 安全性へと変換する方法を示した。

この方法では、構成要素として、Strong One-Time 署名 (使い捨て署名) を用いる。Strong One-Time 署名とは、定義 2.2 を満たす署名のうち、署名クエリの上限回数 $q = 1$ の場合である。One-Time 署名は一方向性関数の仮定だけから構成できることが知られている。一方向性関数の仮定は、証明可能安全性の枠組みで仮定される最も弱い基本的な仮定であり、Teranishi らによる変換 [60] や Steinfeld らによる変換 [59] 変換の安全性を DL 仮定や RSA 仮定などの数論を基にした具体的な仮定を用いなくともこのような弱い安全性だけからいえることがこの研究での貢献であるといえる。

ただし、実際に一方向性だけの仮定から実際に One-Time 署名を構成しようとすると、署名のサイズや鍵長が大きくなってしまう。

4.3　電子署名において衝突困難ハッシュ関数を用いないようにする研究

Mironov　Mironov [47] は、ターゲット衝突困難ハッシュ関数を用いて署名可能なメッセージのサイズを伸ばすパラダイム [48] を用いると一般的にハッシュ鍵の分署名のサイズが大きくなってしまうという問題を、DSA [1]、RSA-PSS [8]、Cramer-Shoup 署名 [25] という具体的な署名方式について解決した。ただしこれらの成果は全て各方式の具体的な方式に依存するものであり、一般的な署名方式について同様の結果を得られるかということについては何もふれられていない。

DSA と RSA-PSS についてはメッセージを固定長の値にするためのハッシュ関数を署名可能メッセージ長を伸ばすためのハッシュ関数としてターゲット衝突困難ハッシュ関数で置き換え、ランダムオラクルとして用いる必要のある関数は従来より入力を小さくできる。また、元の方式を少し変形し、署名作成の際に作られる乱数をターゲット衝突困難ハッシュ関数のハッシュ鍵として用いるような方式になっており、署名検証の際は署名からそのハッシュ鍵を復元できるため、署名としてハッシュ鍵をつける必要が無く、一般的な方法と違い署名長が伸びない。

Cramer-Shoup 署名はもともと安全性の証明のためにターゲット衝突困難ハッシュ関数を使うようになっており、その入力にメッセージを入力するような構成になっている。そして署名にハッシュ鍵をつけておく必要があった。これを方式を少し変形することで、DSA や RSA-PSS の変形と同じように、ハッシュ鍵を署名だけから復元できるようにした。これにより、ハッシュ鍵を送らなくてよくなる分元の方式より署名サイズを小さくできる。

この研究での署名アルゴリズムで作られる乱数を署名作成とメッセージをハッシュするために 2 通りの意味で用いるという方法は提案手法の変換でも使われている。

Halevi-Krawcyzk　一般的に、広く用いられているアルゴリズムなどに問題が発生し、変更の必要があった場合、たとえ問題を解決策があったとしてもシステムを大きく書き換えてしまうような変更であれば、その適用は大きなコストを伴ってしまう。

現状主に実用されている DSA [1]、RSA [2]、RSA-PSS [8] などでは電子署名の安全性のために衝突困難性が必要である。しかし、Wang ら [61] による SHA-1 に対する攻撃に代表される攻撃によって、実用ハッシュ関数で衝突困難性を達成するのは難しそうに考えられる。

そこで、Halevi と Krawcyzk [36] は強化ターゲット衝突困難性 (2.3.3 節参照) という衝突困難性よりも真に弱い安全性である新しいハッシュ関数の安全性の概念を提案し、署名システムにほんのわずかな変更を加えるだけで証明可能安全性の枠組みで、この衝突困難性よりも弱い安全性だけを用いて (すなわち衝突困難性を用いないで) 署名方式の安全性の証明を与えることができるような方法について議論した。ここでのほんのわずかな変更とは、本質的な署名方式の構造は変更しないで (署名方式をブラックボックスとして用いることができるなど)、署名したいメッセージとして署名アルゴリズムに入力する前に乱数でマスクするなどの非常に小さな変更である。根本的な考え方は、鍵無しで決定的なアルゴリズムとして用いることになっているハッシュ関数にランダムなハッシュ鍵を導入し、ハッシュの演算を確率的なアルゴリズムとして考えることである。

同論文では、強化ターゲット衝突困難性を達成するハッシュ関数の構成やターゲット衝突困難性との関係性などについても論じている。

Panisi-Vaudenay　Panishi と Vaudenay [52] は、Mironov [47] が行った具体的な方式に依存するものの、衝突困難性を用いないで任意長に対して署名可能にし、かつターゲット衝突困難ハッシュ関数を用いた固定長署名可能な署名可能方式を任意長署名可能な署名可能方式への変換 [48] を効率化するという研究の方向性を継承し、主に強化衝突困難ハッシュ関数を用いて、より広いクラスの署名方式に適用できるように一般化する方法について議論した。具体的には固定長のメッセージに対して安全に署名できる署名方式を、下記の 2 つの方法で任意長のメッセージに対して安全に署名できる変換を示し、その安全性証明を行っている。両変換とも現在用いられている多くの方式があてはまる、署名アルゴリズムがメッセージが与えられる前に事前計算できる部分と、与えられた後に事前計算の結果を用いて署名を作成するに分けられるようなクラスの署名方式を変換前の方式として仮定する。両変換とも、強化ターゲット衝突困難ハッシュ関数をメッセージに対して最初にかけるハッシュ関数として使用する。また、変換に対する計算コストのオーバーヘッドは、ハッシュ関数の演算から生じるものだけなので、本質的な意味では計算コストのオーバーヘッドは無い。

変換法の 1 つは、[48] とは異なった方法で任意長メッセージに署名可能な方式へ変換できる。スタンダードモデルの下で、証明可能な安全性証明が可能なものの、署名サイズはやはりハッシュ鍵の分長くなってしまう。もう 1 つは、ランダムオラクルの下で証明可能安全性を証明できる変換方法である。この場合、署名サイズは変換前の方法の署名サイズと同様にできる。

同論文では、強化ターゲット衝突困難性ハッシュ関数をメッセージのハッシュに用い、EUF-CMA 安全性より弱い既知文書攻撃に対する安全性しか満たさない固定長に対するメッセージに対して署名可能な方式から、任意長に対して署名可能な EUF-CMA 安全性を示せる変換方法も提案している。ただし証明はランダムオラクルモデルの下でのものとなっている。

Chapter 5　方式間の比較

本章では、3.2 節と 3.3 節で提案した 2 つの署名方式と、既存のスタンダードモデルで証明可能安全な署名方式とを比較した。これをまとめたものが表 5.1 である。比較対称は、Cramer-Shoup (CS) 方式 [25]、Boneh-Boyen (BB) 方式 [10]、Boneh-Shen-Waters (BSW) 方式 [16]、Waters 方式 [62] に Teranishi-Oyama-Ogata (TOO) [60] 変換のうちスタンダードモデルの変換法を適用した方式 (Waters + TOO 方式と呼ぶことにする) [1]である。Waters + TOO 方式以外は全て 4.1 節であげた方式である。

表 5.1 の見方　"HF" は各署名方式の中でどのような安全性を仮定されたハッシュ関数がメッセージをハッシュするのに使用されているか (メッセージとともに他の要素が入力されている場合も含む) を表している。この列において、"CR"、"eTCR"、"TCR"、"–" はそれぞれ、衝突困難性、強化ターゲット衝突困難性、ターゲット衝突困難性、ハッシュ関数は使用されていないことを意味する。"$\mathcal{M}$" の列は各署名方式で署名することができるメッセージ空間の種類を表している。従って、このメッセージ空間よりも大きなメッセージを署名したい場合、衝突困難ハッシュ関数による Hash-and-Sign パラダイム [28] や、ターゲット衝突困難ハッシュ関数を用いた Hash-and-Sign パラダイムに似た構成法 [48] など、何らかの形でハッシュ関数によるメッセージを固定長のデータへと写す演算が必要になる。また、表には他方式との比較のため、SEUF-CMA 安全性は持たない Waters 署名も入れている。p を巡回群 $\mathbb{G}$ と $\mathbb{G}_1$ の位数とする。

全般的な特徴　表 5.1 の基準で比較を行うと、本研究での 2 つの提案手法はハッシュ関数の提案手法はハッシュ関数の安全性の強弱以外は違いが現れないため、以下の各々の方式との比較はターゲット衝突困難ハッシュ関数を用いる提案手法との比較について議論する。ただし、必要に応じて強化衝突困難ハッシュ関数を用いた方式との違いにもふれる。まず個々の方式とを比較する前に、我々の提案した方式の署名としての潜在的に不利な点として、内部で使用しているハッシュ関数 F の出力長 n によって、$O(n)$ のオーダーで検証鍵の大きさが左右されてしまう。しかし、3.1.1 節でもふれたが、検証鍵は一般的に秘匿する必要がないために、ある程度の大きさであっても、問題にならない場合もある。表 5.1 においてこの特徴を持つのは、Waters 署名を基にす

[1]Steinfeld ら [59] と Huang ら [37] による安全性を強化する変換を Waters 署名に適用すれば、さらにスタンダードモデルでの SEUF-CMA 安全性を証明可能な署名方式が得られる。しかし、Steinfeld らの変換は、4.2 節でもふれたように、Randomized Trapdoor Hash Function を DL 仮定に基づいて構成すると TOO 変換を適用したときよりも計算コストの効率が悪くなり、RSA 仮定に基づいて構成すると DL 系の仮定に基づく提案方式の方式との比較が容易ではない。また、Huang らの変換方法は証明可能安全な Strong One-Time 署名を用いなければならない。提案方式の安全性の仮定である CDH 仮定より弱い仮定だけから構成しようとすれば、署名サイズ、は提案手法より大きくなると考えられる。以上の理由により、本章での比較に両変換を用いた構成は考えないことにした。

る BSW 署名、本研究での提案方式 2 つ、Waters + TOO 方式である。

各方式との比較　以下、個別に各方式との比較を行う。BB 方式を見ると、BB 方式はこの表の中で署名サイズ及び署名計算コストにおいて最も効率が良い方式になっている。しかし、本研究での提案方式は CDH 問題の困難性に SEUF-CMA 安全性を帰着できるのに対し、CDH 仮定は BB 方式の安全性の根拠である q-SDH 仮定 [10] よりも弱い仮定であることが知られている。4.1 節でもふれたが、q-SDH 仮定は、方式実現の際の楕円曲線のパラメータの選び方によっては安全でなくなる場合があることが示されている [22, 40]。そして、BB 方式のメッセージ空間は、$\mathbb{Z}_p$ として定義されている。すなわち、任意長のメッセージに対して署名するためには、署名したいメッセージに対する何らかのハッシュ関数の演算が必要である。もしスタンダードモデルで、BB 方式に対して任意長のメッセージ空間にすることを提案方式と同様に衝突困難性より弱い仮定だけから実現しようとすれば、4.3 節で紹介した強化ターゲット衝突困難ハッシュ関数 [52] の変換法か、ターゲット衝突困難ハッシュ関数を用いた Hash-and-Sign のような一般的な変換法 [48] しか知られていないが、どちらの方法をとるにせよ、ハッシュ鍵の分署名サイズが長くなってしまう。さらに、後者を用いる場合は、土台となる巡回群の位数 p を、$k||H_k(M)$ が $\mathbb{Z}_p$ の要素とみなせるように十分な大きさをとらなければならないため、ハッシュ鍵以外の署名のサイズも大きくなってしまう。

BSW 方式との違いは、3 節での各提案方式の説明の際に述べてきたが、提案方式は衝突困難ハッシュ関数を用いなくてもよいことが最も大きな違いである。署名サイズについては 2 つの提案方式も BSW と全く同等にできる。計算のコストについては、強化ターゲット衝突困難ハッシュ関数を用いた方の署名方式は完全に同一になる。ターゲット衝突困難ハッシュ関数を用いた方の署名方式は、署名及び検証の際、入力されたメッセージ M から、途中の値 m を求める際に、提案方式では $g^t h_1^s h_2^{t'}$ という 3 つの底の累乗計算の形をしているのに対し、BSW 署名では、$g^t h^s$ と、2 つの底の累乗計算となっている。しかしながらこのような計算の差は、simultaneous exponentiation technique [46] という複数の底の累乗計算の効率化方法を用いることで実質的にはほぼ同等とみなすことができる。

Waters + TOO 方式と比べると、我々の方式は署名コストが若干悪くなるものの、署名サイズは、一つの整数 ($|p|$ ビット) 分署名長を短くできる。この署名サイズの差は、提案方式を得るために用いた変換法 (3.2.1 節) と TOO 変換の構成の違いからくるものである。我々の変換では変換前の署名方式の構造にあまり一般的とはいえない性質 "*simulatable-partitioned*" を必要とする分変換による署名サイズへのオーバーヘッドは一つの整数だけでよかったが、TOO 変換は本研究での変換法で必要な署名である必要が無く、より広く一般的な署名方式に適用可能である分署名サイズへのオーバーヘッドが 2 つの整数分生じてしまう。しかしながら、BSW 変換との構成の違い同様、TOO 変換でも衝突困難ハッシュ関数を用いる必要があったため、Waters + TOO 方式でもハッシュ関数に同安全性が必要であるが、提案方式ではハッシュ関数に衝突困難性よりも弱い安全性の仮定だけでよい点が利点として挙げられる。

CS 方式、及び Fischlin 方式とは、安全性の仮定となる問題の性質が異なるため、

署名サイズや計算コストの厳密な比較は難しい。しかし、現在よく用いられる RSA モジュラス 1024 ビットを考慮すれば、本研究での提案方式の方が署名サイズについては短くできることがいえる。安全性の根拠となる仮定についても、強 RSA 仮定より、CDH 仮定の方が長い歴史を持っているため、CDH 仮定の方がより妥当な仮定ということができると考えられる。

表 5.1: スタンダードモデルでの強偽造不可能な電子署名間での比較

	仮定	$\mathcal{M}$	HF	署名サイズ	署名コスト ‡†	検証コスト ‡†
CS [25]([47])†	強 RSA	$\{0,1\}^*$	TCR	$161 + 2\|n'\|$	–	–
Fischlin [32]†	強 RSA	$\{0,1\}^*$	CR	$321 + \|n'\|$	–	–
BB [10]‡	q-SDH	$\mathbb{Z}_p$	–	$\|\mathbb{G}_1\| + \|p\|$	$1\text{exp}_{\mathbb{G}_1}$	$1P + 1\text{m-exp}_{\mathbb{G}_1}$
BSW [16]	CDH	$\{0,1\}^*$	CR	$2\|\mathbb{G}\| + \|p\|$	$1\text{m-exp}_{\mathbb{G}} + 3\text{exp}_{\mathbb{G}}$	$2P + 1\text{m-exp}_{\mathbb{G}} + 1\text{exp}_{\mathbb{G}}$
Waters [62] + TOO [60]	CDH	$\{0,1\}^*$	CR	$2\|\mathbb{G}\| + 2\|p\|$	$4\text{exp}_{\mathbb{G}}$	$2P + 1\text{m-exp}_{\mathbb{G}} + 1\text{exp}_{\mathbb{G}}$
TCRHF に基づく方式 (3.2 節)	CDH	$\{0,1\}^*$	TCR	$2\|\mathbb{G}\| + \|p\|$	$1\text{m-exp}_{\mathbb{G}} + 3\text{exp}_{\mathbb{G}}$	$2P + 1\text{m-exp}_{\mathbb{G}} + 1\text{exp}_{\mathbb{G}}$
eTCRHF に基づく方式 (3.3 節)	CDH	$\{0,1\}^*$	eTCR	$2\|\mathbb{G}\| + \|p\|$	$1\text{m-exp}_{\mathbb{G}} + 3\text{exp}_{\mathbb{G}}$	$2P + 1\text{m-exp}_{\mathbb{G}} + 1\text{exp}_{\mathbb{G}}$
Waters [62]	CDH	$\{0,1\}^n$	–	$2\|\mathbb{G}\|$	$3\text{exp}_{\mathbb{G}}$	$2P + 1\text{exp}_{\mathbb{G}}$

† CS 方式は、Mironov [47] によるターゲット衝突困難ハッシュ関数を用いた改善方式が使われているとする。n' は RSA モジュラス (二つの大きな素数の積) とする。

‡ BB 方式では、非対称な双線形写像 $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ が使われている。

‡† これらの列では、P、$\text{exp}_{\mathbb{G}}$、そして $\text{m-exp}_{\mathbb{G}}$ はそれぞれ、署名や検証に必要な双線形写像の回数、底が一つの $\mathbb{G}$ における累乗計算 (g^r のような計算) の回数、底が複数の累乗計算 ($g^t \cdot h^s$ のような計算) の回数を意味している。それ以外の演算 (ハッシュ関数など) は無視できるとする。Waters 署名 (そしてそれに基づいた方式) において用いられている ($u' \prod_{i=1}^{n} u_i^{m'_i}$)(“Waters hash” と呼ばれることもある) を一回の底が一つの指数計算と数えている。また、Waters 署名 (及びそれに基づく方式) の検証の中で出てくるの双線形写像 $e(g_1, g_2)$ の計算については KeyGen の段階で事前に行っておくことができるため、この表では数えていない。

Chapter 6　まとめ

本研究では、電子署名とハッシュ関数の安全性の関係について着目し、近年問題になっている実用ハッシュ関数に対する数々の衝突発見攻撃の成功報告に示唆されるように達成が難しいと考えられるハッシュ関数の衝突困難性を用いずに、安全性を証明可能でかつ効率の良い電子署名方式を目指した。

そして、衝突困難ハッシュ関数を用いずに強偽造不可能性を持つ電子署名方式を 2 種類提案し、その安全性証明を示した。提案方式の一つは、ターゲット衝突困難ハッシュ関数を使用する方式である。もう一つの提案方式はは強化ターゲット衝突困難ハッシュ関数を使用する方式で、偶然にも得られた方式は、基にした BSW 署名 [16] とはハッシュ関数の使い方が違うだけで他の計算や署名サイズなどが全く同じになる方式となる。いずれの方式もスタンダードモデルでの CDH 仮定に基づいており、両方式は、スタンダードモデルで CDH 仮定に基づく強偽造不可能性を持つ電子署名方式である BSW 署名と同程度に効率がよい。しかし提案方式の構成においては、BSW 署名では用いられている衝突困難ハッシュ関数を使用していないため、内部で使用されているハッシュ関数から、衝突発見攻撃により衝突困難性が破られても、電子署名としての安全性は揺るがない。両提案方式は、あるクラスに当てはまる偽造不可能性を持つ電子署名を強偽造不可能性へと変換する変換方法を提案し安全性証明を行い、Waters 署名 [62] に対して、その変換を適用することにより提案方式を得るという方法で示した。

また、既存のスタンダードモデルでの強偽造不可能性を持つ電子署名と比較を行い、署名サイズと署名生成、署名検証の計算コストについては、強い仮定を用いている分高い効率性を持つ BB 署名 [10] 以外とならば、同程度以上に効率が良いことを示した。

謝辞

本研究にあたり、日頃から常にご指導を頂きました東京大学生産技術研究所 松浦幹太准教授に心から感謝致します。松浦先生には、研究の内容及び研究の進め方や考え方のみならず、研究に対する姿勢や着眼点など基礎的なところから教えて頂き、また学会参加など多数の各種活動の機会を与えて頂いたことで、修士 2 年間を非常に有意義に過ごすことができました。

また、研究に関して大いに助言、議論をしていただいた、産業技術総合研究所 情報セキュリティ研究センターの IBE 勉強会メンバーの花岡悟一郎さん、北川隆さん、張鋭さん、崔洋さん、Nuttapong Attrapadung(ナッツ) さんには深く感謝いたします。特に花岡さん、ナッツさんには論文執筆、発表資料の作成など細部に渡るまで助言を頂きました。

そして、私たちの研究活動が円滑に進むように日頃から尽力してくださっている教授室秘書の仲野さん、研究室秘書だった鶴山さんにも改めて感謝致します。

また、松浦研のメンバーである楊鵬さん、Jacob Schuldt さん、Vadim Zendejas さん、Phan Thi Lan Anh さん、北田亘君、渡邉悠君にも、日頃から研究室内での議論や、松浦研定期ミーティングにおいて、活発に議論をしたり、適切な助言をいただきました。改めて感謝致します。

最後になりますが、松浦研定期ミーティングの参加者の皆様、常日頃から私を支えてくれた家族、そして修士 2 年間の研究を通してお世話になりました全ての方に深く感謝致します。

参考文献

[1] Digital Signature Standard (DSS). FIPS 186, 1994.

[2] PKCS #1 v2.1: RSA Cryptography Standard, RSA Laboratories, June 14, 2002.

[3] J.H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *Proc. of EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 83–107. Springer, 2002.

[4] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Proc. of CRYPTO 2000*, volume 1880 of *LNCS*, pages 255–270. Springer, 2000.

[5] P. Barreto and M. Naehrig. Pairing-friendly elliptic curves of prime order. In *Proc. of SAC 2005*, volume 3897 of *LNCS*, pages 319–331. Springer, 2006.

[6] M. Bellare, A. Boldyreva, and A. Palacio. An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In *Proc. of ASIACRYPT 2004*, volume 3027 of *LNCS*, pages 171–188. Springer, 2004.

[7] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proc. of CCS 1993*, pages 62–73. ACM, 1993.

[8] M. Bellare and P. Rogaway. The exact security of digital signatures — how to sign with RSA and Rabin. In *Proc. of EUROCRYPT 1996*, volume 1070 of *LNCS*, pages 399–416. Springer, 1996.

[9] M. Bellare and P. Rogaway. Collision-resistant hashing: Towards making UOWHFs practical. In *Proc. of CRYPTO 1997*, volume 1294 of *LNCS*, pages 320–335. Springer, 1997.

[10] D. Boneh and X. Boyen. Short signatures without random oracles. In *Proc. of EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73. Springer, 2004.

[11] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Proc. of CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, 2004.

[12] D. Boneh and M. Franklin. Identity-based encryption from theWeil pairing. In *Proc. of CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.

[13] D. Boneh and M. Franklin. Identity-based encryption from theWeil pairing. *SIAM J. Computing*, 32(3):585–615, 2003. Full version of [12].

[14] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In *Proc. of ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 514–532. Springer, 2001.

[15] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. *J. of Cryptology*, 17(4):297–319, 2004. Full version of [14].

[16] D. Boneh, E. Shen, and B. Waters. Strongly unforgeable signatures based on computational diffie-hellman. In *Proc. of PKC 2006*, volume 3958 of *LNCS*, pages 229–240. Springer, 2006.

[17] X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques, 2005. Updated version of [18].

[18] X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques. In *Proc. of CCS 2005*, pages 320–329. ACM, 2005.

[19] J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In *Proc. of SCN 2002*, volume 2576 of *LNCS*, pages 268–289. Springer, 2002.

[20] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, rivisited. In *Proc. of STOC 1998*, pages 209–218. ACM, 1998.

[21] R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *Proc. of EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer, 2004.

[22] J. Cheon. Security analysis of the strong Diffie-Hellman problem. In *Proc. of EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 1–11. Springer, 2006.

[23] S. Contini, A.K. Lenstra, and R. Steinfeld. VSH, an efficient ant provable collision-resistant hash function. In *Proc. of EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 165–182. Springer, 2006.

[24] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Proc. of CRYPTO 1998*, volume 1462 of *LNCS*, pages 13–25. Springer, 1998.

[25] R. Cramer and V. Shoup. Signature schemes based on the strong rsa assumption. *ACM TISSEC*, 3(3):161–185, 2000. Extended abstract in Proc. of CCS 1999, ACM, 1999.

[26] R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM J. Computing*, 33(1):167–226, 2003.

[27] Y. Cui, E. Fujisaki, G. Hanaoka, H. Imai, and R. Zhang. Formal security treatments for IBE to signature transformation: Relations among security notions. In *Proc. of ProvSec 2007*, volume 4784 of *LNCS*, pages 218–227. Springer, 2007.

[28] I. Damgård. Collision free hash functions and public key signature schemes. In *Advances in Cryptology — EUROCRYPT 1987*, volume 304 of *LNCS*, pages 203–216. Springer, 1988.

[29] W. Diffie and M.E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[30] T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Theory*, IT-31:469–472, 1985.

[31] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Proc. of CRYPTO 1986*, volume 263 of *LNCS*, pages 186–194. Springer, 1987.

[32] M. Fischlin. The Cramer-Shoup strong-RSA signature scheme revisited. In *Proc. of PKC 2003*, volume 2567 of *LNCS*, pages 116–129. Springer, 2003.

[33] G. Frey and H. Rück. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Mathematics of Computation*, 62(206):865–874, 1994.

[34] S. Goldwasser and Y.T. Kalai. On the (in)security of the Fiat-Shamir paradigm. In *Proc. of FOCS 2003*, pages 102–113. IEEE Computer Society Press, 2003.

[35] S. Goldwasser, S. Micali, and R. Rivest. A digital signature schemes secure against adaptive chosen-message attacks. *SIAM J. Computing*, 17(2):281–308, 1988.

[36] S. Halevi and H. Krawczyk. Strengthening digital signatures via randomized hashing. In *Proc. of CRYPTO 2006*, volume 4117 of *LNCS*, pages 41–59. Springer, 2006.

[37] Q. Huang, D.S. Wong, and Y. Zhao. Generic transformation to strongly unforgeable signatures. In *Proc. of ACNS 2007*, volume 4521 of *LNCS*, pages 1–17. Springer, 2007.

[38] J. Katz and C.J. Koo. On constructing universal one-way hash functions from arbitrary one-way functions, 2005. Available at eprint.iacr.org/2005/328.

[39] N. Koblitz and A. Menezes. Pairing-based cryptography at high security levels. In *Proc. of Cryptography and Coding 2005*, volume 3796 of *LNCS*, pages 13–36. Springer, 2005.

[40] S. Kozaki, T. Kutsuma, and K. Matsuo. Remarks on Cheon's algorithms for pairing-related problems. In *Proc. of Pairing 2007*, volume 4575 of *LNCS*, pages 302–316. Springer, 2007.

[41] H. Krawczyk and T. Rabin. Chameleon hashing and signatures. In *Proc. of NDSS 2000*, pages 143–154. Internet Society, 2000.

[42] F. Laguillaumie, B. Libert, and J. Quisquater. Universal designated verifier signatures without random oracles or non-black box assumptions. In *Proc. of SCN*, pages 63–77. Springer, 2006.

[43] S. Lu, R. Ostrovsky, A. Sahai, H. Shacham, and B. Waters. Sequential aggregate signatures and multisignatures without random oracles. In *Proc. of EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 465–485. Springer, 2006.

[44] A.J. Menezes, T. Okamoto, and S.A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. In *Proc. of STOC 1991*, pages 80–89. ACM, 1991.

[45] A.J. Menezes, T. Okamoto, and S.A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, IT-39(5):1639–1646, 1993. Journal version of [44].

[46] A.J. Menezes, P.C. Oorschot, and S.A. Vanstone. *HANDBOOK of APPLIED CRYPTOGRAPHY*. CRC Press, 1996.

[47] I. Mironov. Collision resistant no more: Hash-and-sign paradigm revisited. In *Proc. of PKC 2006*, volume 3958 of *LNCS*, pages 140–156. Springer, 2006.

[48] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *Proc. of STOC 1989*, pages 33–43. ACM, 1989.

[49] J.B. Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In *Proc. of CRYPTO 2002*, LNCS, pages 191–214. Springer, 2002.

[50] T. Okamoto. Efficient blind and partially blind signatures without random oracles. In *Proc. of TCC 2006*, volume 3876 of *LNCS*, pages 80–99. Springer, 2006.

[51] D. Page, N.P. Smart, and F. Vercauteren. A comparison of MNT curves and supersingular curves. *Applicable Algebra in Engineerings, Communication and Computing (AAECC)*, 17(5):379–392, 2006.

[52] S. Pasini and S. Vaudenay. Hash-and-sign with weak hashing made secure. In *Proc. of ACISP 2007*, volume 4586 of *LNCS*, pages 338–354. Springer, 2007.

[53] K.G. Paterson and J.C.N. Schuldt. Efficient identity-based signatures secure in the standard model. In *Proc. of ACISP 2006*, volume 4058 of *LNCS*, pages 207–222. Springer, 2006.

[54] P. Rogaway. Formalizing human ignorance: Collision-resistanct hashing without the keys. In *Proc. of VIETCRYPT 2006*, volume 4341 of *LNCS*, pages 221–228. Springer, 2006.

[55] J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proc. of STOC 1990*, pages 387–394, 1990.

[56] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing. In *Proc. of SCIS 2000 (Japan)*, 2000.

[57] A. Shamir. Identity-based cryptosystems and signature schemes. In *Proc. of CRYPTO 1984*, LNCS, pages 47–53. Springer, 1985.

[58] D.R. Simon. Finding collision on one-way street: Can secure hash functions be based on general assumptions? In *Proc. of EUROCRYPT 1998*, volume 1403 of *LNCS*, pages 334–345. Springer, 1998.

[59] R. Steinfeld, J. Pieprzyk, and H. Wang. How to strengthen any weakly unforgeable signature into a strongly unforgeable signature. In *Proc. of CT-RSA 2007*, volume 4377 of *LNCS*, pages 357–371. Springer, 2007.

[60] I. Teranishi, T. Oyama, and W. Ogata. General conversion for obtaining strongly existentially unforgeable signatures. In *Proc. of INDOCRYPT 2006*, volume 4329 of *LNCS*, pages 191–205. Springer, 2006.

[61] X. Wang, Y.L. Yin, and H. Yu. Finding collisions in the full SHA-1. In *Proc. of CRYPTO 2005*, volume 3621 of *LNCS*, pages 12–36. Springer, 2005.

[62] B. Waters. Efficient identity-based encryption without random oracles. In *Proc. of EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer, 2005.

発表文献

英文論文誌

1. Takahiro Matsuda, Nuttapong Attrapadung, Goichiro Hanaoka, Kanta Matsuura, Hideki Imai. “A Strongly Unforgeable Signature under the CDH Assumption without Collision Resistant Hash Functions”, IEICE Transactions on Information and Systems, (採録通知済み).

査読付き国際会議投稿論文

ii Takahiro Matsuda, Nuttapong Attrapadung, Goichiro Hanaoka, Kanta Matsuura, Hideki Imai. “A CDH-Based Strongly Unforgeable Signature without Collision Resistant Hash Function”, Proc. of First International Conference on Provable Security (ProvSec 2007), Wollongong, Australia. Nov. 2007. LNCS 4784, pp. 83-107, Springer 2007.

iii Takahiro Matsuda, Goichiro Hanaoka, Kanta Matsuura, Hideki Imai. “A Practical Provider Authentication System for Bidirectional Broadcast Service”, Proc. of 11th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES 2007), Vietri sul Mare, Italy. Sep. 2007. LNAI 4694, pp. 967-974, Springer, 2007.

査読無し国内会議投稿論文

iv 松田隆宏, 花岡悟一郎, 松浦幹太, 今井秀樹. “任意の頑強な ID ベース暗号に基づく CCA 安全な公開鍵暗号の効率的構成方法”, 2008 年 暗号と情報セキュリティシンポジウム (SCIS 2008) 予稿集 CDROM, 2F1-1. 宮崎, 1 月・2008 年.

v 松田隆宏, アッタラパドゥン ナッタポン, 花岡悟一郎, 松浦幹太, 今井秀樹. “スタンダードモデルでの CDH 仮定に基づく衝突困難ハッシュ関数を用いない強偽造不可能性を持つ電子署名”, 2007 年 暗号と情報セキュリティシンポジウム (SCIS 2007) 予稿集 CDROM, 3C4-4. 長崎, 1 月・2007 年.