SCIS 2007 The 2007 Symposium on Cryptography and Information Security
Sasebo, Japan, Jan. 23-26, 2007
The Institute of Electronics, Information and Communication Engineers

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署名方式

A CDH-based Strongly Unforgeable Signature in the Standard Model without Collision Resistant Hash Function

松田 隆宏* Takahiro Matsuda
ナッタポン アッタラパドゥン * Nuttapong Attrapadung
花岡 悟一郎† Goichiro Hanaoka
松浦 幹太 * Kanta Matsuura
今井 秀樹 † Hideki Imai

あらまし 電子署名において、入力が任意長の衝突困難ハッシュ関数を用いてハッシュ値に対し署名を付ける Hash-and-Sign パラダイム [4] を用いる方式は、ハッシュ関数の衝突困難性が破れると安全ではなくなってしまう。また、実用の際は計算時間などを考慮し、SHA-1 などの標準ハッシュ関数を衝突困難であるとして利用することが多いが、近年それらの標準ハッシュ関数に対する衝突発見攻撃の報告が相次いでおり、実際には標準ハッシュ関数を使うとしても、ハッシュ関数に必要な仮定を弱めることは重要なことである。そこで衝突困難ハッシュ関数の代わりとして汎用一方向ハッシュ関数の利用を考える。Hash-and-Sign パラダイムに代わる一般的な方式 [6] は存在するが、署名長が長くなるという問題がある。また、任意長のメッセージの署名を可能にする用途以外には使えるかどうか分からない。本稿では、2006 年 Boneh らによって提案された署名方式 [3] に変形を加え、安全性の証明に必要とされる衝突困難ハッシュ関数を用いずに、それより弱い仮定で実現できる汎用一方向ハッシュ関数を利用し、署名長を伸ばすことなく、元の方式と同じ CDH 仮定で適応的選択文書攻撃に対し強存在的偽造不可能性を持つ方式を示した。

キーワード 電子署名、強存在的偽造不可能性、汎用一方向ハッシュ関数、スタンダードモデル

1 はじめに

電子署名においてハッシュ関数は、安全性の証明、及び任意長のメッセージに対する署名を可能にするという目的で使われることが多い。特に、任意長のメッセージに対する署名のために、入力が任意長の衝突困難ハッシュ関数 (Collision Resistant Hash Function, CRHF) を用いて、ハッシュ値に対し署名を付ける Hash-and-Sign パラダイム [4] を用いる方式は、ハッシュ関数の衝突困難性が破れると安全ではなくなってしまう。さらに、そのような署名方式の実用を考えると、計算時間等を考慮し、SHA-1 などの標準ハッシュ関数が衝突困難であるとして用いるのが普通であるが、近年それらの標準ハッシュ関数に対する衝突発見攻撃の報告が相次いでいる。従って、実際には同様に標準ハッシュ関数を使うにしても、方式設計の際に CRHF より弱い仮定のハッシュ関数を用いることは意義のあることである。

そこで、CRHF の代わりとして、ハッシュ関数を汎用一方向ハッシュ関数 (Universal One-Way Hash Function, UOWHF) として用いることを考える。UOWHF は Naor ら [7] により導入され、後に Bellare ら [2] により、ターゲット衝突困難ハッシュ関数 (Target Collision Resistant Hash Function, TCRHF) とも名付けられた鍵付きハッシュ関数である (以下本稿では汎用一方向ハッシュ関数を TCRHF と略す)。TCRHF の性質は、メッセージを決めてから、ハッシュ鍵が与えられ、その鍵の元、メッセージと衝突を起こす他のメッセージを見つけることが難しいというもので、CRHF より弱い仮定である [8]。

TCRHF は、その安全性の定義から、Hash-and-Sign パラダイムを利用する署名方式において単に CRHF と置き換えただけでは安全性の証明ができなくなる場合が多い。一般的な手法 [6] が存在するが、メッセージのハッシュ値とハッシュ鍵の連結に対して署名を作成し、さらに署名データにハッシュ鍵を追加しなければならず、署名のエレメント数が増えるという問題がある。また、この手法はハッシュ関数を Hash-and-Sign パラダイムの役割として使用しているときには利用できるが、それ以外の目的では一般的に使用できない。

Boneh らは [3] において、あるクラスにあてはまる署

* 東京大学 生産技術研究所 〒 153-8505 東京都目黒区駒場 4-6-1, Institute of Industrial Science, The University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan.
† 産業技術総合研究所 情報セキュリティセンター 〒 101-0021 東京都千代田区外神田 1-18-13, Research Center for Information Security, National Institute of Advanced Industrial Science and Technology, 1-18-13 Sotokanda, Chiyoda-ku, Tokyo 101-0021, Japan.

名方式の、適応的選択文書攻撃に対し強存在的偽造不可能性を持つ署名方式への変換と、具体的な CDH 仮定に基づく従来までより効率的な署名方式を示した。その方式では、CRHF をの入力空間を任意長とすると、結果的に任意長のメッセージに対する署名が可能となる。

本稿ではこの署名方式に対し、[6] のアイデアを参考に、CRHF を用いないでかつ署名のエレメント数が同じ方式を提案する。まず、あるクラスに当てはまる弱偽造不可能性を持つ電子署名に対する強偽造不可能性を持つ署名への変換方式を示す。そして CDH 仮定に基づく Waters による署名方式 [9] がそのクラスに当てはまることを示し、[3] と同様 CDH 仮定で強偽造不可能性を持つ署名方式を示す。

2 準備

本節では、本稿で使用する言葉や安全性の定義を行う。

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**:** セキュリティパラメータ 1^k を入力とし、公開鍵 pk と秘密鍵 sk の対を出力する。

署名 Sign**:** 秘密鍵 sk、メッセージ $m \in \mathcal{M}$ を入力とし、正しい署名 σ を出力する。

検証 Verify**:** 公開鍵 pk、メッセージ m、署名 σ を入力とし、σ がメッセージ 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。

2.2 強存在的偽造不可能性と (弱) 存在的偽造不可能性

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 under Adaptive Chosen Message Attack, sEUF-CMA) は、以下の sEUF-CMA Challenger と攻撃者 $\mathcal{A}$ 間の sEUF-CMA game を用いて定義する [1]。

Setup. Challenger はセキュリティパラメータ 1^k を入力として KeyGen を実行する。生成した pk を $\mathcal{A}$ に渡し、sk を保持しておく。

Queries. $\mathcal{A}$ は Challenger に対し、最大で q 回のクエリ $m_1, m_2, \ldots, m_q$ を発行する。Challenger はそれぞれのクエリ m_i に対し、正しい署名 σ_i を返す。クエリ m_i は過去のクエリ $(m_1, \ldots, m_{i-1})$ に対する Challenger の返答 $(\sigma_1, \ldots, \sigma_{i-1})$ に依存して適応的に発行される。

Output. Queries が終了すると、$\mathcal{A}$ は $(\hat{m}, \hat{\sigma})$ のペアを出力する。$\mathsf{Verify}(pk, \hat{m}, \hat{\sigma}) = \mathtt{accept}$ かつ $1 \leq \forall i \leq q$, $(\hat{m}, \hat{\sigma}) \neq (m_i, \sigma_i)$ ならば $\mathcal{A}$ の勝利となる。

定義 1 ϵ より大きな確率で sEUF-CMA game に対し勝利できる動作時間 t のアルゴリズムが存在しない場合、電子署名 Σ は (t, q, ϵ)-sEUF-CMA 安全性を持つという。

同様に、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 under Adaptive Chosen Message Attack,EUF-CMA) は (強偽造不可能性との違いを明確にするため、弱偽造不可能性とも呼ばれる)、以下の EUF-CMA Challenger と攻撃者 $\mathcal{A}$ 間の EUF-CMA game を用いて定義する [5]。

Setup. と Queries. sEUF-CMA game と同内容。

Output. Queries が終了すると、$\mathcal{A}$ は $(\hat{m}, \hat{\sigma})$ のペアを出力する。$\mathsf{Verify}(pk, \hat{m}, \hat{\sigma}) = \mathtt{accept}$ かつ $1 \leq \forall i \leq q$, $\hat{m} \neq m_i$ ならば $\mathcal{A}$ の勝利となる。

定義 2 ϵ より大きな確率で EUF-CMA game に対し勝利できる動作時間 t のアルゴリズムが存在しない場合、電子署名 Σ は (t, q, ϵ)-EUF-CMA 安全性を持つという。

2.3 Computational Diffie-Hellman 仮定と離散対数仮定

位数 p の有限巡回群 $\mathbb{G}$ における Computational Diffie-Hellman (CDH) 問題を解くとは、$a, b \in \mathbb{Z}_p, g \in \mathbb{G}$ について、g, g^a, g^b を与えられて g^{ab} を出力することである。

定義 3 ϵ より大きな確率で $\mathbb{G}$ における CDH 問題を解ける動作時間 t のアルゴリズムが存在しない場合、$\mathbb{G}$ において (t, ϵ)-CDH 仮定が成り立つという。

同様に、位数 p の有限巡回群 $\mathbb{G}$ における離散対数 (Discrete Logarithm, DL) 問題を解くとは、$a \in \mathbb{Z}_p, g \in \mathbb{G}$ について、g, g^a を与えられて a を出力することである。

定義 4 ϵ より大きな確率で $\mathbb{G}$ における DL 問題を解ける動作時間 t のアルゴリズムが存在しない場合、$\mathbb{G}$ において (t, ϵ)-DL 仮定が成り立つという。

2.4 衝突困難ハッシュ関数と汎用一方向性ハッシュ関数

$H : \mathcal{K} \times \mathcal{M}_{in} \to \mathcal{M}_{out}$ を鍵付きハッシュ関数、$k \in \mathcal{K}$ を H のハッシュ鍵とする。ただし、$\mathcal{M}_{in}$:入力空間、$\mathcal{M}_{out}$:出力空間、$\mathcal{K}$:ハッシュ鍵空間とする。

CRHF は、CR Challenger からランダムに選ばれた $k \in \mathcal{K}$ を与えられた攻撃者 $\mathcal{A}$ が、$H_k(m_1) = H_k(m_2)$ かつ、$m_1 \neq m_2 \in \mathcal{M}_{in}$ となる任意のメッセージの組 (m_1, m_2) を見つけ出すと勝利となる CR game を用いて以下の様に定義する。

定義 5 ϵ より大きな確率で CR game に対し勝利できる動作時間 t のアルゴリズムが存在しない場合、H は (t, ϵ)-CRHF であるという。

TCRHF の定義は、以下の様な TCR Challenger と攻撃者 $\mathcal{A}$ 間の TCR game を用いて記述される。

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

Step 2. $\mathcal{A}$ は、Challenger からランダムに選ばれた $k \in \mathcal{K}$ を受け取る。

Step 3. $H_k(m_1) = H_k(m_2)$ かつ、$m_2 \neq m_1$ となるメッセージ $m_2 \in \mathcal{M}_{in}$ を出力すると $\mathcal{A}$ の勝利となる。

定義 6 ϵ より大きな確率で TCR game に対し勝利できる動作時間 t のアルゴリズムが存在しない場合、H は (t, ϵ)-TCRHF であるという。

2.5 Bilinear map

本稿での Biliniar map とは以下の性質を持つものとする。$\mathbb{G}$ と $\mathbb{G}_1$ を位数が素数 p の有限巡回群とし、e は

$\mathbb{G}\times\mathbb{G}\to\mathbb{G}_1$ となる写像であり、1.bilinear: 全ての $g\in\mathbb{G}$ および全ての $a,b\in\mathbb{Z}$ に対し、$e(g^a,g^b)=e(g,g)^{ab}$、2.non-degenerate: $e(g,g)\neq 1$、3.computable: 全ての $g\in\mathbb{G}$ に対して e を効率よく計算できる。

3 提案方式

本節では、Boneh らによる “Partitioned” という概念 [3] を少し強めた概念である “Simulatable Partitioned” という概念を定義する。そして、[3] において示された変換法 (BSW 変換) に変更を加え、EUF-CMA 安全性を持つ Simulatable Partitioned な署名方式に対して本節で述べる変換 (TCR-BSW 変換と呼ぶことにする) を適応すると、BSW 変換では必要だった入力空間が任意長の CRHF を用いず、代わりに TCRHF を用いて sEUF-CMA 安全性を持つ署名方式への一般的な変換ができることを示す。

定義 7 署名方式 Σ が以下の 3 つの性質を満たすとき、Σ は Simulatable Partitioned であるという。

- 性質 1: メッセージ入力 $m\in\mathcal{M}$ に対し、署名アルゴリズム Sign が以下の二つの決定的アルゴリズム S_1 と S_2 に分けることができ、sk を用いて以下の様に記述される。
  1. ランダムに $r\in\mathcal{R}$
  2. $\sigma_1=\mathsf{S}_1(sk;m,r)\ \in\mathcal{S}_1$,
     $\sigma_2=\mathsf{S}_2(sk;r)\ \in\mathcal{S}_2$ を計算する。
  3. $\sigma=(\sigma_1,\sigma_2)$ を出力する。

  ただし、$\mathcal{S}_1$ は S_1 の、$\mathcal{S}_2$ は S_2 の出力空間、$\mathcal{R}$ は S_1 と S_2 に使われる乱数の空間である。
- 性質 2: σ_2 と m が与えられると、$\mathsf{Verify}(pk,m,(\sigma_1,\sigma_2))=\mathtt{accept}$ となるような σ_1 は高々1 つしか存在しない。
- 性質 3: 以下の 2 つのアルゴリズムが存在する。
  - KeyGen' : セキュリティパラメータ 1^k が与えられると、KeyGen の出力 (sk,pk) と分布が等しく、Σ に対する正しい鍵ペアである (sk',pk') と共に、S'_1 のためのトラップドア TD を出力する。すなわち、
    $(sk',pk',TD)\leftarrow\mathsf{KeyGen}'(1^k)$
  - S'_1 : 決定的アルゴリズムであり、sk, σ_2, m と、KeyGen' から出力された TD が与えられると、r を用いずに $\mathsf{Verify}(pk,m,(\sigma'_1,\sigma_2))=\mathtt{accept}$ となるような正しい σ'_1 を出力する。すなわち、
    $\sigma'_1=\mathsf{S}'_1(sk,m,\sigma_2,TD)\ \in\mathcal{S}_1$

定義 8 署名方式 Σ が定義 7 のうち性質 1 と性質 2 を満たすとき、Σ は Partitioned であるという [3]。

“Simulatable Partitioned” という名前は、安全性証明のときにシミュレータの構成の際に必要となる性質であることから名付けた。

3.1 TCR-BSW 変換

次に、本稿で提案する変換方式 TCR-BSW 変換について記述する。$\Sigma=(\mathsf{KeyGen},\mathsf{Sign},\mathsf{Verify})$ を EUF-CMA 安全性を持つ Simulatable Partitioned な署名方式とする。すなわち、Sign は $\mathsf{S}_1,\mathsf{S}_2$ に分割することができ、定義 7 の性質 3 を満たすような $\mathsf{KeyGen}',\mathsf{S}'_1$ が存在する。p を十分大きな素数とし、$\mathbb{G}$ を位数 p の有限巡回群とする。$H:\mathcal{S}_2\times\{0,1\}^*\to\mathbb{Z}_p, G:\mathcal{K}\times\mathcal{S}_2\to\mathbb{Z}_p, F:\mathcal{K}\times\mathbb{G}\to\mathcal{M}$ をそれぞれ TCRHF とする。ただし、$\mathcal{M},\mathcal{R},\mathcal{S}_1,\mathcal{S}_2,\mathcal{K}$ はそれぞれ本稿の 2 章、3 章の各定義で述べたものが当てはまるとする。提案する変換により得られる新しい署名方式 Σ_{new} の記述は以下である。

$\mathsf{KeyGen}_{new}(1^k)$: ランダムに $g,h_1,h_2\in\mathbb{G}$ と $k'\in\mathcal{K}$ を選ぶ。$\mathsf{KeyGen}(1^k)$ を実行し、(sk,pk) を得る。KeyGen_{new} に対する秘密鍵 sk', 公開鍵 pk' はそれぞれ、
$sk'=(sk),\ pk'=(pk,g,h_1,h_2,k')$

$\mathsf{Sign}_{new}(sk',M):M\in\{0,1\}^*$ に対する署名 σ は以下の様に生成する。

1. ランダムに $s\in\mathbb{Z}_p$ と $r\in\mathcal{R}$ を選ぶ。
2. $\sigma_2=\mathsf{S}_2(sk,r)\ \in\mathcal{S}_2$ を計算する。
3. $m=g^{H_{\sigma_2}(M)}h_1^s h_2^{G_{k'}(\sigma_2)}\ \in\mathbb{G}$ を計算する。
4. $\sigma_1=\mathsf{S}_1(sk,F_{k'}(m),r)\ \in\mathcal{S}_1$ を計算する。
5. $\sigma=(\sigma_1,\sigma_2,s)$ を出力する。

$\mathsf{Verify}_{new}(pk',M,\sigma)$: M に対する署名 $\sigma=(\sigma_1,\sigma_2,s)$ は以下の様に検証する。

1. $m=g^{H_{\sigma_2}(M)}h_1^s h_2^{G_{k'}(\sigma_2)}$ を計算する。
2. $\mathsf{Verify}(pk,F_{k'}(m),(\sigma_1,\sigma_2))$ を出力する。

3.2 提案方式の証明

$\Sigma_{new}=(\mathsf{KeyGen}_{new},\mathsf{Sign}_{new},\mathsf{Verify}_{new})$ を 3.1 節の様に構成したとする。このとき、Σ_{new} は、sEUF-CMA 安全性を持つことを証明する。

定理 1 以下の条件が成り立つとき、Σ_{new} は (t,q,ϵ)-sEUF-CMA 安全性を持つ。

- 基となっている Σ は $(t,q,\epsilon/6)$-EUF-CMA 安全性を持つ Simulatable Partitioned な署名方式である
- $\mathbb{G}$ において $(t,\epsilon/3)$-DL 仮定が成り立つ
- H,G,F はそれぞれ、$(t,\epsilon/6q)$, $(t,\epsilon/6q)$, $(t,\epsilon/6q)$-TCRHF

証明 $\mathcal{A}$ を Σ_{new} の (t,q,ϵ)-sEUF-CMA 安全性を破る攻撃者とする。$\mathcal{A}$ は最初に公開鍵 $pk'=(pk,g,h_1,h_2,k')$ を与えられる。

攻撃者 $\mathcal{A}$ は sEUF-CMA game の Queries. の段階において、$1\le\forall i\le q$ に対し、i 回目のクエリに M_i を発行し、$\sigma_i=(\sigma_{i,1},\sigma_{i,2},s_i)$ を受け取るとする。$m_i=g^{H_{\sigma_{i,2}}(M_i)}h_1^{s_i}h_2^{G_{k'}(\sigma_{i,2})}$ であると定義する。また、攻撃者 $\mathcal{A}$ は sEUF-CMA game の Output. の段階において、$(\hat{M},\hat{\sigma}=(\hat{\sigma_1},\hat{\sigma_2},\hat{s}))$ を出力するとし、$\hat{m}=g^{H_{\hat{\sigma_2}}(\hat{M})}h_1^{\hat{s}}h_2^{G_{k'}(\hat{\sigma_2})}$ であると定義する。

以下の 6 タイプを考える。攻撃者 $\mathcal{A}$ が sEUF-CMA game に勝利する場合、必ず以下のタイプ 1 からタイプ 6 のどれかにあてはまる。

タイプ 1: $1\le\forall i\le q, F_{k'}(\hat{m})\neq F_{k'}(m_i)$

タイプ 2: $1 \leq \exists i \leq q, F_{k'}(\hat{m}) = F_{k'}(m_i) \wedge \hat{m} \neq m_i$

タイプ 3: $1 \leq \exists i \leq q, F_{k'}(\hat{m}) = F_{k'}(m_i) \wedge \hat{m} = m_i \wedge G_{k'}(\hat{\sigma_2}) \neq G_{k'}(\sigma_{i,2})$

タイプ 4: $1 \leq \exists i \leq q, F_{k'}(\hat{m}) = F_{k'}(m_i) \wedge \hat{m} = m_i \wedge G_{k'}(\hat{\sigma_2}) = G_{k'}(\sigma_{i,2}) \wedge \hat{\sigma_2} \neq \sigma_{i,2}$

タイプ 5: $1 \leq \exists i \leq q, F_{k'}(\hat{m}) = F_{k'}(m_i) \wedge \hat{m} = m_i \wedge G_{k'}(\hat{\sigma_2}) = G_{k'}(\sigma_{i,2}) \wedge \hat{\sigma_2} = \sigma_{i,2} \wedge H_{\hat{\sigma_2}}(\hat{M}) \neq H_{\sigma_{i,2}}(M_i)$

タイプ 6: $1 \leq \exists i \leq q, F_{k'}(\hat{m}) = F_{k'}(m_i) \wedge \hat{m} = m_i \wedge G_{k'}(\hat{\sigma_2}) = G_{k'}(\sigma_{i,2}) \wedge \hat{\sigma_2} = \sigma_{i,2} \wedge H_{\hat{\sigma_2}}(\hat{M}) = H_{\sigma_{i,2}}(M_i)$

以下、それぞれのタイプに対する、シミュレータの構成を示す。最初にシミュレータ B はランダムに、どのタイプの偽造が来るのかを予想することとする。

タイプ 1: $\mathcal{A}_1$ がタイプ 1 攻撃者で、Σ_{new} の (t, q, ϵ)-sEUF-CMA 安全性を破れると仮定する。このとき、Σ の (t, q, ϵ)-EUF-CMA 安全性を破れるシミュレータ $\mathcal{B}_1$ を構成する。$\mathcal{B}_1$ は公開鍵 pk を与えられ、$\mathcal{B}_1$ 自身の EUF-CMA game に勝利することが目的である。シミュレートは以下である。

Setup. $\mathcal{B}_1$ は公開鍵 pk' を以下の様に作る。

1. ランダムに $g \in \mathbb{G}, k' \in \mathcal{K}$ を選ぶ。
2. ランダムに $a, b \in \mathbb{Z}_p^*$ を選び、$h_1 = g^a, h_2 = g^b$ とする。
3. $pk' = (pk, g, h_1, h_2, k')$ を出力する。

Queries. $\mathcal{B}_1$ は $\mathcal{A}_1$ のクエリ M_i に対し、以下の様に答える。

1. ランダムに $w_i \in \mathbb{Z}_q$ を選び、$m_i = g^{w_i}$とする。
2. $\mathcal{B}_1$ 自身の Challenger に対し、$F_{k'}(m_i)$ をクエリとして尋ね、$(\sigma_{i,1}, \sigma_{i,2})$ を得る。
3. $s_i = (w_i - H_{\sigma_{i,2}}(M_i) - bG_{k'}(\sigma_{i,2}))/a$ を計算する。
4. $\sigma_i = (\sigma_{i,1}, \sigma_{i,2}, s_i)$ を出力する。

Output. Queries. が終了すると、$\mathcal{A}_1$ は偽造 $(\hat{M}, (\hat{\sigma_1}, \hat{\sigma_2}, \hat{s}))$ を出力する。$\mathcal{B}_1$ は以下の様にして自身の Challenger に対し、偽造を出力する。

1. $\hat{m} = g^{H_{\hat{\sigma_2}}(\hat{M})} h_1^{\hat{s}} h_2^{G_{k'}(\hat{\sigma_2})}$ を計算する。
2. $(F_{k'}(\hat{m}), \hat{\sigma_1}, \hat{\sigma_2})$ を出力する。

ここで、Σ_{new} の定義より、$\mathsf{Verify}_{new}(\hat{M}, (\hat{\sigma_1}, \hat{\sigma_2}, \hat{s})) = \mathsf{accept}$ ならば、$\mathsf{Verify}((F_{k'}(\hat{m}), (\hat{\sigma_1}, \hat{\sigma_2})) = \mathsf{accept}$ であり、タイプ 1 の条件より、$1 \leq \forall i \leq q, F_{k'}(\hat{m}) \neq F_{k'}(m_i)$ なので、$\mathcal{B}_1$ は、$\mathcal{A}_1$ がタイプ 1 の偽造に成功するときはいつも EUF-CMA game に勝利することができ、Σ に対する正しい署名を偽造できる。

タイプ 2: $\mathcal{A}_2$ がタイプ 2 攻撃者で、Σ_{new} の (t, q, ϵ)-sEUF-CMA 安全性を破れると仮定する。このとき、$(t, \epsilon/q)$-TCRHF である F の TCR game に勝利できるシミュレータ $\mathcal{B}_2$ を構成する。$\mathcal{B}_2$ は $\mathcal{A}_2$ を以下の様に走らせる。

Setup. $\mathcal{B}_2$ は公開鍵 pk' を以下の様に作る。

1. ランダムに $j \in \{1 \ldots q\}$ を選ぶ。
2. KeyGen を実行し、sk, pk を得る。
3. ランダムに $g \in \mathbb{G}$ を選ぶ。
4. ランダムに $a, b \in \mathbb{Z}_p^*$ を選び、$h_1 = g^a, h_2 = g^b$ とする。
5. ランダムに $\bar{w}$ を選び、$\bar{m} = g^{\bar{w}}$ とする。
6. $\mathcal{B}$ 自身の Challenger に $\bar{m}$ を出力し、$k' \in \mathcal{K}$ を得る。
7. $pk' = (pk, g, h_1, h_2, k')$ を出力する。$sk' = sk$ は保持しておく。

Queries. $\mathcal{B}_2$ は $\mathcal{A}_2$ のクエリ M_i に対し、
$i \neq j$ のときは、$\sigma_i = \mathsf{Sign}_{new}(sk', M_i)$ を出力する。
$i = j$ のときは、以下を実行する。

1. $(\sigma_{j,1}, \sigma_{j,2}) = \mathsf{Sign}(sk, F_{k'}(\bar{m}))$ を計算する。
2. $s_j = (\bar{w} - H_{\sigma_{i,2}}(M_j) - bG_{k'}(\sigma_{j,2}))/a$ を計算する。
3. $\sigma_j = (\sigma_{j,1}, \sigma_{j,2}, s_j)$ を出力する。

Output. Queries. が終了すると、$\mathcal{A}_2$ は偽造 $(\hat{M}, (\hat{\sigma_1}, \hat{\sigma_2}, \hat{s}))$ を出力する。$\mathcal{B}_2$ は自身の Challenger に対し、
$\hat{m} = g^{H_{\hat{\sigma_2}}(\hat{M})} h_1^{\hat{s}} h_2^{G_{k'}(\hat{\sigma_2})}$ を出力する。

タイプ 2 の条件より、$1 \leq \exists i \leq q, F_{k'}(\hat{m}) = F_{k'}(m_i) \wedge \hat{m} \neq m_i$ であり、$\mathcal{B}_2$ が Setup. の段階にそのような i を j として選ぶ確率は、少なくとも $1/q$ である。従って、$\mathcal{B}_2$ は、$\mathcal{A}_2$ がタイプ 2 の偽造に成功するときは少なくとも $1/q$ の確率で F についての TCR game に勝利することができる。

タイプ 3: $\mathcal{A}_3$ がタイプ 3 攻撃者で、Σ_{new} の (t, q, ϵ)-sEUF-CMA 安全性を破れると仮定する。このとき、$\mathbb{G}$ における (t, ϵ)-DL 仮定を破れるシミュレータ $\mathcal{B}_3$ を構成する。$\mathcal{B}_3$ は DL 問題 (g, X) を与えられ、$\log_g X$ を出力することが目的である。$\mathcal{B}_3$ は $\mathcal{A}_3$ を以下の様に走らせる。

Setup. $\mathcal{B}_3$ は公開鍵 pk' を以下の様に作る。

1. KeyGen を実行し、sk, pk を得る。
2. ランダムに $a \in \mathbb{Z}_p^*$ を選び、$h_1 = g^a, h_2 = X$ とする。
3. ランダムに $k' \in \mathcal{K}$ を選ぶ。
4. $pk' = (pk, g, h_1, h_2, k')$ を出力する。$sk' = sk$ は保持しておく。

Queries. $\mathcal{B}_3$ は $\mathcal{A}_3$ のクエリ M_i に対し、
$\sigma_i = \mathsf{Sign}_{new}(sk', M_i)$ を出力する。

Output. Queries. が終了すると、$\mathcal{A}_3$ は偽造 $(\hat{M}, (\hat{\sigma_1}, \hat{\sigma_2}, \hat{s}))$ を出力する。$\mathcal{B}_3$ は自身の Challenger に対し、$\log_g X = \{(H_{\hat{\sigma_2}}(\hat{M}) + a\hat{s}) - (H_{\sigma_{i,2}}(M_i) + as_i)\}/(G_{k'}(\sigma_{i,2}) - G_{k'}(\hat{\sigma_2}))$ を出力する。

これは、$\hat{m} = m_i$ 、すなわち、$g^{H_{\hat{\sigma_2}}(\hat{M}) + a\hat{s}} X^{G_{k'}(\hat{\sigma_2})} = g^{H_{\sigma_{i,2}}(M_i) + as_i} X^{G_{k'}(\sigma_{i,2})}$ という等式の両辺の $\log_g$ をとって整理して得られた式である。$G_{k'}(\hat{\sigma_2}) \neq G_{k'}(\sigma_{i,2})$ より、分母は 0 ではない。従って、$\mathcal{B}_3$ は、$\mathcal{A}_3$ がタイプ 3 の偽造に成功するときはいつも DL 問題を解ける。

タイプ 4: $\mathcal{A}_4$ がタイプ 4 攻撃者で、Σ_{new} の (t, q, ϵ)-sEUF-CMA 安全性を破れると仮定する。このとき、$(t, \epsilon/q)$-TCRHF である G の TCR game に勝利できるシミュレータ $\mathcal{B}_4$ を構成する。$\mathcal{B}_4$ は $\mathcal{A}_4$ を以下の様に走らせる。

Setup. $\mathcal{B}_4$ は公開鍵 pk' を以下の様に作る。

1. ランダムに $j \in \{1 \dots q\}$ を選ぶ。
2. KeyGen を実行し、sk, pk を得る。
3. ランダムに $g, h_1, h_2 \in \mathbb{G}$ を選ぶ。
4. ランダムに $\bar{r} \in \mathcal{R}$ を選び、$\bar{\sigma_2} = \mathsf{S}_2(sk, \bar{r})$ とする。
5. $\mathcal{B}$ 自身の Challenger に $\bar{\sigma_2}$ を出力し、$k' \in \mathcal{K}$ を得る。
6. $pk' = (pk, g, h_1, h_2, k')$ を出力する。$sk' = sk$ は保持しておく。

Queries. $\mathcal{B}_4$ は $\mathcal{A}_4$ のクエリ M_i に対し、
$i \neq j$ のときは、 $\sigma_i = \mathsf{Sign}_{new}(sk', M_i)$ を出力する。
$i = j$ のときは、以下を実行する。
1. ランダムに $s_j \in \mathbb{Z}_p$ を選ぶ。
2. $m_j = g^{H_{\bar{\sigma_2}}(M_j)} h_1^{s_j} h_2^{G_{k'}(\bar{\sigma_2})}$ を計算する。
3. $\sigma_{j,1} = S_1(sk, F_{k'}(m_j), \bar{r})$ を計算する。
4. $\sigma_{j,2} = \bar{\sigma_2}$ とする。
5. $\sigma_j = (\sigma_{j,1}, \bar{\sigma_2}, s_j)$ を出力する。

Output. Queries. が終了すると、$\mathcal{A}_4$ は偽造 $(\hat{M}, (\hat{\sigma_1}, \hat{\sigma_2}, \hat{s}))$ を出力する。$\mathcal{B}_4$ は自身の Challenger に対し、$\hat{\sigma_2}$ を出力する。

タイプ 4 の条件より、$1 \leq \exists i \leq q, G_{k'}(\hat{\sigma_2}) = G_{k'}(\sigma_{i,2}) \wedge \hat{\sigma_2} \neq \sigma_{i,2}$ であり、$\mathcal{B}_4$ が Setup. の段階にそのような i を j として選ぶ確率は、少なくとも $1/q$ である。従って、$\mathcal{B}_4$ は、$\mathcal{A}_4$ がタイプ 4 の偽造に成功するときは少なくとも $1/q$ の確率で G についての TCR game に勝利できる。

タイプ 5: $\mathcal{A}_5$ がタイプ 5 攻撃者で、Σ_{new} の (t, q, ϵ)-sEUF-CMA 安全性を破れると仮定する。このとき、$\mathbb{G}$ における (t, ϵ)-DL 仮定を破れるシミュレータ $\mathcal{B}_5$ を構成する。$\mathcal{B}_5$ は DL 問題 (g, X) を与えられ、$\log_g X$ を出力することが目的である。$\mathcal{B}_5$ は $\mathcal{A}_5$ を以下の様に走らせる。

Setup. $\mathcal{B}_5$ は公開鍵 pk' を以下の様に作る。
1. KeyGen を実行し、sk, pk を得る。
2. $h_1 = X$ とする。
3. ランダムに $h_2 \in \mathbb{G}, k' \in \mathcal{K}$ を選ぶ。
4. $pk' = (pk, g, h_1, h_2, k')$ を出力する。$sk' = sk$ は保持しておく。

Queries. $\mathcal{B}_5$ は $\mathcal{A}_5$ のクエリ M_i に対し、
$\sigma_i = \mathsf{Sign}_{new}(sk', M_i)$ を出力する。

Output. Queries. が終了すると、$\mathcal{A}_5$ は偽造 $(\hat{M}, (\hat{\sigma_1}, \hat{\sigma_2}, \hat{s}))$ を出力する。$\mathcal{B}_5$ は自身の Challenger に対し、
$\log_g X = (H_{\hat{\sigma_2}}(\hat{M}) - H_{\sigma_{i,2}}(M_i))/(s_i - \hat{s})$ を出力する。

ここで、$\hat{m} = m_i \wedge \hat{\sigma_2} = \sigma_{i,2}$ より、
$g^{H_{\hat{\sigma_2}}(\hat{M})} X^{\hat{s}} = g^{H_{\sigma_{i,2}}(M_i)} X^{s_i}$ である。また、タイプ 5 の条件 $H_{\hat{\sigma_2}}(\hat{M}) \neq H_{\sigma_{i,2}}(M_i)$ より、必ず $\hat{s} \neq s_i$ である。よって、等式の両辺の $\log_g$ を取って整理すると、Output. での $\mathcal{B}_5$ の出力する値が得られる。従って、$\mathcal{B}_5$ は、$\mathcal{A}_5$ がタイプ 5 の偽造に成功するときはいつも DL 問題を解くことができる。

タイプ 6: $\mathcal{A}_6$ がタイプ 6 攻撃者で、Σ_{new} の (t, q, ϵ)-sEUF-CMA 安全性を破れると仮定する。このとき、$(t, \epsilon/q)$-TCRHF である H の TCR game に勝利できるシミュレータ $\mathcal{B}_6$ を構成する。$\mathcal{B}_6$ は $\mathcal{A}_6$ を以下の様に走らせる。

Setup. $\mathcal{B}_6$ は公開鍵 pk' を以下の様に作る。
1. ランダムに $j \in \{1 \dots q\}$ を選ぶ。
2. KeyGen$'$ を実行し、$sk,\ pk,\ TD$ を得る。
3. ランダムに $g, h_1, h_2 \in \mathbb{G}, k' \in \mathcal{K}$ を選ぶ。
4. $pk' = (pk, g, h_1, h_2, k')$ を出力する。$sk' = sk$ は保持しておく。

Queries. $\mathcal{B}_6$ は $\mathcal{A}_6$ のクエリ M_i に対し、
$i \neq j$ のときは、 $\sigma_i = \mathsf{Sign}_{new}(sk', M_i)$ を出力する。
$i = j$ のときは、以下を実行する。
1. $\mathcal{B}_6$ 自身の Challenger に M_j を出力し、$\bar{\sigma_2} \in \mathcal{S}_2$ を得る。$\sigma_{j,2} = \bar{\sigma_2}$ とする。
2. ランダムに $s_j \in \mathbb{Z}_p$ を選ぶ。
3. $m_j = g^{H_{\bar{\sigma_2}}(M_j)} h_1^{s_j} h_2^{G_{k'}(\bar{\sigma_2})}$ を計算する。
4. $\sigma_{j,1} = \mathsf{S}'_1(sk, F_{k'}(m_j), \bar{\sigma_2}, TD)$ を計算する。
5. $\sigma_j = (\sigma_{j,1}, \sigma_{j,2}, s_j)$ を出力する。

Output. Queries. が終了すると、$\mathcal{A}_6$ は偽造 $(\hat{M}, (\hat{\sigma_1}, \hat{\sigma_2}, \hat{s}))$ を出力する。$\mathcal{B}_6$ は自身の Challenger に対し、$\hat{M}$ を出力する。

$\mathcal{A}_6$ が sEUF-CMA game に勝利する条件により、$1 \leq \forall i \leq q, (\hat{M}, \hat{\sigma_1}, \hat{\sigma_2}, \hat{s}) \neq (M_i, \sigma_{i,1}, \sigma_{i,2}, s_i)$ であり、タイプ 5 の条件と同様にタイプ 6 でも、$\hat{\sigma_2} = \sigma_{i,2} \wedge \hat{s} = s_i$。また、Simulatable Partitioned の性質 2 より、$\hat{\sigma_2} = \sigma_{i,2}$ ならば、$\hat{\sigma_1} = \sigma_{i,1}$。これらとタイプ 6 の条件を合わせて、$1 \leq \exists i \leq q, \hat{M} \neq M_i \wedge \hat{\sigma_2} = \sigma_{i,2} \wedge H_{\hat{\sigma_2}}(\hat{M}) = H_{\sigma_{i,2}}(M_i)$ となる。$\mathcal{B}_6$ が Setup. の段階にそのような i を j として選ぶ確率は、少なくとも $1/q$ である。従って、$\mathcal{B}_6$ は、$\mathcal{A}_6$ がタイプ 6 の偽造に成功するときは少なくとも $1/q$ の確率で H についての TCR game に勝利することができる。

以上、全てのタイプに対するシミュレータの構成法を示した。全ての $\mathcal{A}$ に対するシミュレートは完全である。タイプ 1 は基となる署名 Σ の EUF-CMA 安全性を、タイプ 3,5 は離散対数仮定を、タイプ 2,4,6 は TCRHF の安全性を、それぞれ破るのに使用できることを示した。以上により、定理 1 は証明された。 □

3.3 備考

離散対数系の仮定に基づく方式の場合、鍵生成の際に群 $\mathbb{G}$(位数 p) の元を複数作成する必要があるときは、先に一つの元をランダムに生成し、残りの元は離散対数を $\mathbb{Z}_p$ から選んでから作ることができる。定義 7 の Simulatable Partitioned の性質 3 の TD としては、例えばそのような値があてはまる。この場合、性質 1 と性質 2 を満たすならば、性質 3 を満たす S'_1 が存在するような方式も少なくないと考えられる。

4 CDH 仮定に基づく具体的な署名方式

Waters 署名 [9] は、$\mathbb{G}$ において $(t, \epsilon/8(n+1)q)$-CDH 仮定が成り立つとすると、スタンダードモデルで (t, q, ϵ)-EUF-CMA 安全性を持つ署名方式である [3]。以下に KeyGen アルゴリズムを示す。

KeyGen :
1. ランダムに $g, g_2 \in \mathbb{G}$ を選ぶ。

2. ランダムに $\alpha \in \mathbb{Z}_p$ を選び、$g_1 = g^\alpha$を計算する。
3. ランダムに $u', u_1, u_2, \ldots, u_n \in \mathbb{G}$ を選ぶ。
4. $sk = g_2^\alpha, pk = (g, g_1, g_2, u', u_1, u_2, \ldots, u_n)$ を出力する。

このとき、3. 以降を

3. ランダムに $\beta', \beta_1, \beta_2, \ldots, \beta_n \in \mathbb{Z}_p$ を選び、$u' = g^{\beta'}, u_1 = g^{\beta_1}, u_2 = g^{\beta_2}, \ldots, u_n = g^{\beta_n}$ を計算する。
4. $sk = g_2^\alpha, pk = (g, g_1, g_2, u', u_1, u_2, \ldots, u_n), TD = (\beta', \beta_1, \beta_2, \ldots, \beta_n)$ を出力する。

としたアルゴリズムを KeyGen' とすると、KeyGen' から出力された sk, pk の対は KeyGen から出力された sk, pk の対と同一視することができるのは明らかである。

Waters 署名が Partitioned な署名であることは [3] で示されており、その際の S_1 と S_2 は以下である。

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

ただし、$m_i \in \{0,1\}$ は $m \in \{0,1\}^n$ の i 番目のビット、$u', u_1, \ldots, u_n \in \mathbb{G}, r \in \mathbb{Z}_p, sk \in \mathbb{G}$ である。ここで、先ほどの TD と σ_2 を用いて σ_1 を変形すると、

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

と書くことができ、r を使わずに sk, m, σ_2, TD の関数として σ_1 を表せ、正しい σ_1 を作ることができる。この等式の最後の式で $\mathsf{S}'_1(sk, m, \sigma_2, TD)$ を定義すれば、S'_1、KeyGen'、そして TD は定義 7 の性質 3 を満たすので、Waters 署名が Simulatable Partitioned であることを示すことができた。

従って、定理 1 をそのまま Waters の署名に当てはめれば、目標である CDH 仮定に基づき、衝突困難ハッシュ関数は用いずに、sEUF-CMA 安全性を持つ署名方式が得られる。構成は以下の通りである。

p を十分大きな素数とし、$\mathbb{G}$ を位数 p の有限巡回群とする。$H : \mathbb{G} \times \{0,1\}^* \to \mathbb{Z}_p, G : \mathcal{K} \times \mathbb{G} \to \mathbb{Z}_p, F : \mathcal{K} \times \mathbb{G} \to \{0,1\}^n$ をそれぞれ TCRHF とする。$e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ を Bilinear map とする。

KeyGen_{new} : ランダムに $g \in \mathbb{G}$ を選ぶ。ランダムに $\alpha \in \mathbb{Z}_p$ を選び、$g_1 = g^\alpha$を計算する。ランダムに $g_2, h_1, h_2, u', u_1, \ldots, u_n \in \mathbb{G}$ と $k' \in \mathcal{K}$ を選ぶ。$U = (u_1, \ldots, u_n)$ とおく。以下を出力する。

$sk = g_2^\alpha$, $pk = (g, g_1, g_2, h_1, h_2, u', U, k')$

Sign_{new} : $M \in \{0,1\}^*$ に対する署名 σ は以下の様に生成する。

1. ランダムに $s, r \in \mathbb{Z}_p$ を選ぶ。
2. $\sigma_2 = g^r \quad \in \mathbb{G}$ を計算する。
3. $m = g^{H_{\sigma_2}(M)} h_1^s h_2^{G_{k'}(\sigma_2)} \quad \in \mathbb{G}$ を計算する。
4. $m' = F_{k'}(m)$ を計算する。
   m'_i を m' の i 番目のビット $(i \in \{1, \ldots, n\})$ とする。
5. $\sigma_1 = g_2^\alpha \cdot (u' \prod_{i=1}^n u_i^{m'_i})^r \quad \in \mathbb{G}$ を計算する。
6. $\sigma = (\sigma_1, \sigma_2, s)$ を出力する。

Verify_{new} : M に対する署名 $\sigma = (\sigma_1, \sigma_2, s)$ は以下の様に検証する。

1. $m = g^{H_{\sigma_2}(M)} h_1^s h_2^{G_{k'}(\sigma_2)} \in \mathbb{G}$ を計算する。
2. $m' = F_{k'}(m)$ を計算する。
   m'_i を m' の i 番目のビット $(i \in \{1, \ldots, n\})$ とする。
3. $e(\sigma_1, g) \stackrel{?}{=} e(\sigma_2, u' \prod_{i=1}^n u_i^{m'_i}) \cdot e(g_1, g_2)$
   が成りたつならば `accept`, そうでないならば `reject` を出力する。

系 1 以下の条件が成り立つとき、上記署名方式は (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, F はそれぞれ、$(t, \epsilon/6q)$, $(t, \epsilon/6q)$, $(t, \epsilon/6q)$-TCRHF

証明 Waters の署名が $\mathbb{G}$ における $(t, \epsilon/8(n+1)q)$-CDH 仮定が成り立つと、スタンダードモデルで (t, q, ϵ)-EUF-CMA 安全性を持つ署名方式であり、かつ Simulatable Partitioned であることは既に示した。このとき、$\mathbb{G}$ において明らかに $(t, \epsilon/3)$-DL 仮定は成り立っている。従って、定理 1 の条件を満たしたので、上記署名方式が sEUF-CMA 安全性を持つことが示された。 □

5 まとめ

本稿では、まず署名方式に対する Simulatable Partitioned の概念を定義した。そして、その概念にあてはまる EUF-CMA 安全性を持つ署名方式に対する sEUF-CMA 安全性を持つ署名方式へのスタンダードモデルでの一般的な変換法とその安全性証明を示した。我々の方式は、任意長入力の CRHF を必要としない。さらに、CDH 仮定に基づく Waters の署名 [9] が Simulatable Partitioned にあてはまることを示し、我々の提案方式を適用することでスタンダードモデルで CDH 仮定に基づき、任意長入力の CRHF を用いずに強存在的偽造不可能性を持つ署名方式を示した。

参考文献

[1] J.H.An, Y.Dodis, T.Rabin, "On the Security of Joint Signature and Encryption," In L.R.Knudsen, editor, *Proceedings of EUROCRYPT 2002*, LNCS 2332, pp. 83-107, Springer-Verlag, 2002.

[2] M.Bellare, P.Rogaway, "Collision-Resistant Hashing: Towards Making UOWHFs Practical," In Buston S.Kaliski Jr., editor, *Advances in Cryptology — CRYPTO 1997*, LNCS 1294, pp. 320-335,1997.

[3] D.Boneh, E.Shen, B.Waters, "Strongly Unforgeable Signatures Based on Computational Diffie-Hellman," In *Proceedings of PKC 2006*, LNCS 3958, pp. 229-240, Springer-Verlag, 2006.

[4] I.Damgard, "Collision Free Hash Functions and Public Key Signature Schemes," In *Proceedings of EUROCRYPT'87*, LNCS 304, pp. 203-216, Springer-Verlag, 1988.

[5] S.Goldwasser, S.Micali, R.Rivest, "A Digital Signature Schemes Secure Against Adaptive Chosen-Message Attacks," SIAM J. Computing, 17(2):281-308, 1988.

[6] I.Mironov, "Collision Resistant No More: Hash-and-Sign Paradigm Revisited," In *Proceedings of PKC 2006*, LNCS 3958, pp. 140-156, Springer-Verlag, 2006.

[7] M.Naor, M.Yung, "Universal One-Way Hash Functions and their Cryptographic Applications," In *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pp. 33-43, 15-17 May 1989.

[8] D.R.Simon, "Finding Collision on One-Way Street: Can Secure Hash Functions Be Based on General Assumptions?, " In K.Nyberg, editor, *Advances in Cryptology — EUROCRYPT 1998*, LNCS 1403, pp. 334-345, Springer, 1998.

[9] B.Waters, "Efficient Identity-Based Encryption without Random Oracles," In R. Cramer, editor, *Advances in EUROCRYPT 2005*, LNCS 3494, pp. 114-127, Springer-Verlag, 2005.