格子問題に基づくSigncryption構成法の再考

佐藤 慎悟[1]　四方 順司[1,2]

概要：Signcryption は公開鍵暗号とディジタル署名の機能を同時に達成する方式であり, その安全性は多人数モデルにおいて内部攻撃者に対して強秘匿性と強偽造不可能性を満たすことが望ましい. SCIS2016 で著者らは, この安全性を満たす格子問題に基づく構成法を提案した. 本稿では, この構成法よりも暗号文長が短い格子ベースの構成法を提案する. さらに, 既知の一般的構成法に適切な格子ベースの暗号プリミティブを適用した構成法を効率性の観点から比較解析する.

キーワード：signcryption, 格子問題, 耐量子暗号.

Lattice-based Signcryption, Revisited

Shingo Sato[1]　Junji Shikata[1,2]

Abstract: Signcryption is a cryptographic scheme which achieves both functions of public-key encryption and digital signatures, and its security requires both strong confidentiality and strong unforgeability against insider adversaries in the multi-user setting. In SCIS2016, we proposed a lattice-based construction of signcryption which achieves the security. In this paper, we propose another lattice-based construction which has shorter size of ciphertexts than that of our previous construction. Furthermore, we analyse and compare efficiency of possible constructions of signcryption obtained from the known generic constructions with suitable lattice-based primitives.

Keywords: signcryption, lattice problem, post-quantum cryptography.

1. はじめに

Signcryption の概念は 1997 年に Zheng によって導入された [17]. Signcryption の安全性モデルとして 2 者モデルと多人数モデルがある. 2 者モデルは, 送信者と受信者が１人ずつでやり取りが行われるのに対して, 多人数モデルは複数の送信者と受信者でやり取りが行われるモデルである. 多人数モデルで方式を実現することが現実的だが, 2 者モデルで安全性が満たされても多人数モデルでも安全性が満たされるとは限らないことが知られている. さらに, 想定する攻撃者として外部攻撃者と内部攻撃者がある. 外部攻撃者は公開鍵しか知らないのに対し, 内部攻撃者は送信者と受信者のどちらか一方の秘密鍵を持っていて攻撃することができる. よって, 内部攻撃者に対して安全性を満たすことが望ましい. 強い安全性として, 多人数モデルで内部攻撃者に対して強秘匿性 (MU-IND-iCCA 安全性) と強偽造不可能性 (MU-sUF-iCMA 安全性) が定式化されている [10]. 本稿では, この安全性を満たす方式を考える.

耐量子暗号の一つとして, 格子暗号に関する研究が盛んに行われている. 量子コンピュータを用いても格子問題を解くことは困難と考えられており, また様々な高機能暗号を実現可能なことから, これまでに多くの格子問題に基づく暗号方式が提案されている.

本稿では, MU-IND-iCCA と MU-sUF-iCMA を満たす格子ベースの Signcryption 方式を提案する. SCIS2016 では, 著者らはタグベーストラップドア関数とワンタイム署名, ディジタル署名, カメレオンハッシュを用いて, これ

[1] 横浜国立大学 大学院環境情報学府/研究院. Graduate School of Environment and Information Sciences, Yokohama National University. E-mail: sato-shingo-cz@ynu.jp, shikata@ynu.ac.jp
[2] 横浜国立大学 先端科学高等研究院. Institute of Advanced Sciences, Yokohama National University.

らの安全性を満たす格子ベースの構成法を提案した．その構成法では, ワンタイム署名を利用して MU-IND-iCCA を達成しており, 暗号文の中にワンタイム署名の検証鍵が含まれる．そのため, すべてスタンダードな LWE/SIS で構成した場合に暗号文長が $\tilde{O}(n^2)$ になる．本稿では, これを $\tilde{O}(n)$ の長さに改善するためにワンタイム署名ではなく MAC と Encapuslation を使って構成する．ただし, IND-CCA を満たす公開鍵暗号と sUF-CMA を満たすディジタル署名を単純に組み合わせるだけでは MU-IND-iCCA あるいは MU-sUF-iCMA を満たせないことが知られている ([2], [12])．これを解決するため, タグベーストラップドア置換とそのハードコア述語を用いて一般的構成法を示し, これを基に格子ベースの構成法を示す．さらに, 既存の一般的構成法に格子ベースの暗号プリミティブを適用したものと比較する．

2. 準備

本節では, Signcryption の構成に用いる暗号プリミティブの定義, 格子にかかわる諸定義について述べる．また, 本稿では以下の記法を用いる：正の整数 n に対して $[n] := \{1, \ldots, n\}$ とする．ある定数 c に対して $f(n) = O(g(n) \cdot \log^c n), g(n) = \Omega(n)$ のとき, $f(n) = \tilde{O}(g(n))$ と記述する．本稿ではベクトルは基本的に列ベクトルとして, 行ベクトルはベクトル $\mathbf{x}$ に対して $\mathbf{x}^t$ のように表記する．ベクトルの長さ $|\cdot|$ はユークリッドノルムではかる．任意の定数 c に対して関数 $f(n) = o(n^{-c})$ が成り立つとき, $f(n)$ は negligible であるといい, $\mathrm{negl}(n)$ と記述する．また, 有限集合 Ω 上の 2 つの確率変数 X, Y の統計的距離を $\Delta(X,Y) = \frac{1}{2}\sum_{w\in\Omega} |\Pr[X = w] - \Pr[Y = w]|$ と定義する．

2.1 暗号プリミティブ

2.1.1 タグベーストラップドア関数

定義 1. タグベーストラップドア関数 TBTDF $= (\mathrm{Kg}, \mathrm{F}, \mathrm{F}^{-1})$ は次のように定義される: $(ek, TD) \leftarrow \mathrm{Kg}(1^k), y \leftarrow \mathrm{F}(ek, tag, x), x \leftarrow \mathrm{F}^{-1}(TD, tag, y)$.

Kg は, セキュリティパラメータ k を入力として *evaluation/trapdoor* 鍵のペア ek/TD を出力する．F は $\{0,1\}^k$ 上の関数 $f_{ek,tag}(\cdot)$ を計算するアルゴリズムであり, F^{-1} はその逆像 $f^{-1}_{ek,tag}(\cdot)$ を計算するアルゴリズムである．

TBTDF の安全性として tag-based adaptive one-wayness(TB-AOW) は次のように定義される [8].

定義 2 (TB-AOW)**.** $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ を, *TB-AOW* を破る多項式時間攻撃者として, 次のゲームを考える．

Step 1. $t \xleftarrow{U} \mathcal{A}_1(1^k)$,

Step 2. $(ek, TD) \xleftarrow{U} Kg(1^k), y \leftarrow \mathrm{F}(ek, tag, x)$,

Step 3. $x' \xleftarrow{U} \mathcal{A}_2^{\mathrm{F}^{-1}(TD,\cdot,\cdot)}(ek, tag, y)$,

すべての $\mathcal{A}$ に対して, アドバンテージ $Adv^{\text{TB-AOW}}_{\text{TBTDF},\mathcal{A}}(k) := \Pr[x = x'| \leq \mathrm{negl}(k)$ のとき, *TBTDF* は *TB-AOW* 安全であるという．

本稿ではタグベーストラップドア置換 (TBTDP) を用いる．

2.1.2 ディジタル署名

定義 3. ディジタル署名 DS $=$ (Setup, Kg, Sign, Vrfy) は次のように定義される: $prm \leftarrow \mathrm{Setup}(1^k), (vk, sk) \leftarrow \mathrm{Kg}(prm), S \leftarrow \mathrm{Sign}(sk, \mu), 1/0 \leftarrow \mathrm{Vrfy}(vk, S, \mu)$.

ここで, prm は公開パラメータ, vk を検証鍵, sk は署名鍵, μ はメッセージ, S は μ に対する署名, 1 は受理, 0 は拒否である．

ディジタル署名 DS の strong unforgeability against non-adaptive chosen message attack(sUF-naCMA) は次のように定義される．

定義 4 (sUF-naCMA)**.** $\mathcal{A}$ を *DS* の *sUF-naCMA* を破る多項式時間攻撃者として, 次のゲームを考える．

Step 1. $(\mu_1, \ldots, \mu_Q) \leftarrow \mathcal{A}(1^k)$.

Step 2. $prm \leftarrow \mathrm{DS.Setup}(1^k), (vk, sk) \leftarrow \mathrm{DS.Kg}(prm)$, すべての $i \in [Q]$ に対して $S^{(i)} \leftarrow \mathrm{DS.Sign}(sk, \mu^{(i)})$.

Step 3. $(\mu^*, S^*) \leftarrow \mathcal{A}(vk, \sigma^{(1)}, \ldots, \sigma^{(Q)})$.

$1 = \mathrm{DS.Vrfy}(vk, S^*, \mu^*)$ かつ $(\mu^*, S^*) \neq (\mu^{(i)}, S^{(i)})$ のとき, $\mathcal{A}$ が勝つ事象 $[\mathcal{A}\ wins]$ とする．任意の $\mathcal{A}$ に対して, アドバンテージ $Adv^{\text{sUF-naCMA}}_{\text{DS},\mathcal{A}}(k) := \Pr[\mathcal{A}\ wins] \leq \mathrm{negl}(k)$ ならば, *DS* は *sUF-naCMA* 安全であるという．

2.1.3 カメレオンハッシュ

定義 5. カメレオンハッシュ $\mathrm{CH} = (\mathrm{Gen}, \mathrm{Hash}, \mathrm{Hash}^{-1})$ を次のように定義する: $(hk, td) \leftarrow \mathrm{Gen}(1^k), y \leftarrow \mathrm{Hash}(hk, \mu, r), r' \leftarrow \mathrm{Hash}^{-1}(td, \mu', y)$.

Gen はセキュリティパラメータ k を入力として *hash/trapdoor* 鍵のペア hk/td を出力する．$\mathcal{R}$ を乱数空間, $\mathcal{Y}$ をカメレオンハッシュ関数における値域とする．Hash はメッセージ μ と乱数 $r \in \mathcal{R}$ を入力としてハッシュ値 $y \in \mathcal{Y}$ を出力する．Hash^{-1} は $y = \mathrm{Hash}(hk, \mu', r')$ を満たす乱数 $r' \in \mathcal{R}$ を出力する．

カメレオンハッシュは安全性として一様性と衝突困難性を満たす ([9] を参照).

2.1.4 メッセージ認証子 (MAC)

定義 6. メッセージ認証子 MAC = (Sign, Vrfy) を次のように定義する: $\tau \leftarrow \mathrm{Sign}(K, \mu), 1/0 \leftarrow \mathrm{Vrfy}(K, \mu, \tau)$. K は共通鍵, μ はメッセージ, τ はタグ, 1 は受理, 0 は拒否である．

MAC の安全性として strong unforgeagility against one-time attack(sUF-OT) 安全性を満たす．本稿ではこれの具体的な定義は省略する ([3] などを参照).

2.1.5 Encapsulation

定義 7. *Encapsulation* E $=$ (Setup, Enc, Rec) は次のように定義される: $prm \leftarrow \mathrm{Setup}(1^k), (r, com, dec) \leftarrow \mathrm{Enc}(prm), r \leftarrow \mathrm{Rec}(prm, com, dec)$. ここで, k はセキュリティパラメータ, prm は公開パラメータ, $\mathcal{R}_{encap}$ は乱数空

間, $r \in \mathcal{R}_{encap}$ は乱数, com はコミットメント, dec はデコミットメントである.

Encapsulation E = (Setup, Enc, Rec) の安全性として Hiding と Binding は次のように定義される.

定義 8 (Hiding). 多項式時間攻撃者を $\mathcal{A}$ として, 次のゲームを考える.

Step 1. $prm \leftarrow \mathrm{E.Setup}(1^k)$,
$r_0 \xleftarrow{R} \mathcal{R}_{encap}, (r_1, com, dec) \leftarrow \mathrm{E.Enc}(prm)$,
$b \xleftarrow{U} \{0,1\}$,

Step 2. $b' \leftarrow \mathcal{A}(prm, com, r_b)$.

すべての $\mathcal{A}$ に対して, アドバンテージ $Adv^{hiding}_{\mathrm{Encap},\mathcal{A}}(k) := |\Pr[b = b'] - 1/2| \leq \mathrm{negl}(k)$ のとき, E は *Hiding* を満たすとする.

定義 9 (Binding). 多項式時間攻撃者を $\mathcal{A}$ として, 次のゲームを考える.

Step 1. $prm \leftarrow \mathrm{E.Setup}(1^k), (r, com, dec) \leftarrow \mathrm{Enc}(prm)$

Step 2. $dec^* \leftarrow \mathcal{A}(prm, com, dec)$.

$\mathrm{E.Rec}(prm, com, dec^*) \notin \{\perp, r\}$ のとき, $\mathcal{A}$ が勝つ事象 $[\mathcal{A}\ wins]$ とする. すべての $\mathcal{A}$ に対して, アドバンテージ $Adv^{binding}_{\mathrm{Encap},\mathcal{A}}(k) := \Pr[\mathcal{A}\ wins] \leq \mathrm{negl}(k)$ のとき, E は *Binding* を満たすとする.

2.2 格子

格子 Λ を n 個の線形独立なベクトル $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ から成る整数結合の集合として次のように定義する:$\Lambda = \mathcal{L}(\mathbf{B}) = \{\sum_{i\in[n]} c_i \mathbf{b}_i : c_i \in \mathbb{Z}\}$. 次に離散ガウス分布を定義する. 任意の実数 $s > 0$ に対して, $\mathbf{c} \in \mathbb{R}^n$ を平均とするガウス関数を $\rho_{s,\mathbf{c}}(\mathbf{x}) = \exp(-\pi |\mathbf{x} - \mathbf{c}|^2 / s^2)$ とする. $\rho_{s,\mathbf{c}}(\Lambda) = \sum_{\mathbf{x}\in\Lambda} \rho_{s,\mathbf{c}}(\mathbf{x})$ とする. n 次元格子 Λ の離散ガウス分布を次のように定義する. $\forall \mathbf{x} \in \Lambda, D_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{s,\mathbf{c}}(\mathbf{x})}{\rho_{s,\mathbf{c}}(\Lambda)}$. $\mathbf{c} = \mathbf{0}$ のときは $D_{\Lambda,s}$ のように記述する.

Small Integer Solution (SIS) 問題と Learning With Errors (LWE) 問題について述べる.

定義 10 ($\mathrm{SIS}_{q,\beta}$). 整数 q と実数 β に対して行列 $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ が与えられたとき, $\mathrm{SIS}_{q,\beta}$ とは, $|\boldsymbol{e}| \leq \beta$ かつ $\mathbf{Ae} = \mathbf{0} \bmod q$ を満たすベクトル $\mathbf{e} \in \mathbb{Z}^m \backslash \{\mathbf{0}\}$ を見つける問題である.

$\mathbb{T} = \mathbb{R}/\mathbb{Z}$ とする. ベクトル $\mathbf{s} \in \mathbb{Z}_q^n$ と $\mathbb{R}$ 上のガウス分布 D_α(α は標準偏差) に対して $A_{\mathbf{s},\alpha}$ を次のように定義する: $\mathbf{a} \xleftarrow{U} \mathbb{Z}_q^n$ とエラー $e \leftarrow D_\alpha$ を選び, $(\mathbf{a}, \mathbf{s}^t\mathbf{a}/q + e \bmod 1)$ を出力することで得られる $\mathbb{Z}_q^n \times \mathbb{T}$ 上の分布である.

定義 11 ($\mathrm{LWE}_{q,\alpha}$). $\mathbb{Z}_q^n \times \mathbb{T}$ から $m = \mathrm{poly}(n)$ 個のサンプルが与えられたとき, $\mathrm{LWE}_{q,\alpha}$ とは, $\mathbb{Z}_q^n \times \mathbb{T}$ 上の *2* つの分布 $A_{\mathbf{s},\alpha}(\mathbf{s} \in \mathbb{Z}_q^n)$ と一様分布を識別する問題である.

2.3 トラップドアを用いたアルゴリズム

本節では, 格子ベースのタグベーストラップドア関数 (置換) について述べる. 2012 年に Micciancio と Peikert が提案したトラップドアの概念とトラップドアを用いたアルゴリズム [13] について次が成り立つ.

命題 1 ([13]). 任意の整数 $n \geq 1, q \geq 2$, 十分大きい $m = O(n \log n)$ が与えられるとき, 行列 $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ とトラップドア $\mathbf{R}$ を出力する効率的な確率的アルゴリズム $\mathrm{GenTrap}(1^n, 1^m, q)$ が存在する. さらに, 圧倒的確率で次の処理を行う多項式時間アルゴリズム Invert と SampleD が存在する.

- 任意の $\mathbf{s} \in \mathbb{Z}_q^n$ と $|\mathbf{e}| < q/O(\sqrt{n \log q})$ あるいは $1/\alpha \geq \sqrt{n \log q} \cdot \omega(\sqrt{\log n})$ に対して $\mathbf{e} \leftarrow D_{\mathbb{Z},\alpha q}$ である $\mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t$ に対して, 決定性アルゴリズム $\mathrm{Invert}(\mathbf{R}, \mathbf{A}, \mathbf{b})$ は $\mathbf{s}$ と $\mathbf{e}$ を出力する.
- 任意の $\mathbf{u} \in \mathbb{Z}_q^n$ と十分大きな $s = O(\sqrt{n \log q})$ に対して, 確率的アルゴリズム $\mathrm{SampleD}(\mathbf{R}, \mathbf{A}, \mathbf{u}, s)$ は $D_{\Lambda^\perp_{\mathbf{u}}(\mathbf{A}), s\cdot\omega(\sqrt{\log n})}$ との統計的距離が無視できるほど小さい分布からサンプリングする.

命題 1 を満たす primitive matrix $\mathbf{G}$ を用いて, 上記を満たす行列 $\mathbf{A}$ とトラップドア $\mathbf{R}$ が生成される. 本稿では [13] に従って $\mathbf{G}$ は次のとおりである. 整数 $k = \lceil \log q \rceil$ に対して $\mathbf{g}^t = [1, 2, 2^2, \ldots, 2^{k-1}] \in \mathbb{Z}_q^{1\times k}$ として $\mathbf{G} := \begin{bmatrix} \mathbf{g}^t & & & 0 \\ & \mathbf{g}^t & & \\ & & \ddots & \\ 0 & & & \mathbf{g}^t \end{bmatrix} \in \mathbb{Z}_q^{n\times nk}$

GenTrap アルゴリズムは, 行列 $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times m}$ と正則行列 $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$ を入力として, $\mathbb{Z}^{\bar{m}\times w}$ の分布 D から $\mathbf{R}$ を選び, 行列 $\mathbf{A} = \left[\bar{\mathbf{A}} | \mathbf{HG} - \bar{\mathbf{A}}\mathbf{R}\right]$ を計算する. そして, 行列 $\mathbf{A}$ とそのトラップドア $\mathbf{R}$ を出力する. $\mathbf{H}$ はトラップドアのタグとして用いられる.

3. Signcryption

本節では, Signcryption のモデルと安全性を定義し, 一般的構成法と格子ベースの構成法を提案する.

3.1 Signcryption の定義

定義 12. *Signcryption* 方式 *(*SCS*)* は次の *5* つのアルゴリズムで構成される.

- $prm \leftarrow \mathrm{Setup}(1^n)$
- $(pk_R, sk_R) \leftarrow \mathrm{KeyGen}_R(prm)$
- $(pk_S, sk_S) \leftarrow \mathrm{KeyGen}_S(prm)$
- $\sigma \leftarrow \mathrm{SC}(prm, pk_R, sk_S, \mu)$
- $\mu / \perp \leftarrow \mathrm{USC}(prm, pk_S, sk_R, \sigma)$

ここで n はセキュリティパラメータ, prm は公開パラメータ, (pk_R, sk_R) はそれぞれ受信者の公開鍵と秘密鍵, (pk_S, sk_S) はそれぞれ送信者の公開鍵と秘密鍵, μ はメッセージ, σ は署名付き暗号文である.

Signcryption の安全性としてメッセージの秘匿性と偽造不可能性がある. 本稿では多人数モデルで内部攻撃

者に対する強い安全性を考える．秘匿性において multi-user indistinguishability against insider chosen ciphertext attack(MU-IND-iCCA) と偽造不可能性において multi-user strong unforgeability against insider chosen message attack(MU-sUF-iCMA) を考える．これらの安全性は次のように定義される．

定義 13 (MU-IND-iCCA)**.** *Signcryption* 方式 $\mathrm{SCS} = (\mathrm{Setup}, \mathrm{KeyGen}_R, \mathrm{KeyGen}_S, \mathrm{SC}, \mathrm{USC})$ に対して，*MU-IND-iCCA* は次のように定義される．$\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2\}$ を SCS に対する攻撃者 *(*多項式時間アルゴリズム*)* として，次のゲームを考える．

Step 1. $prm \leftarrow \mathrm{Setup}(1^n)$,
$(pk_R, sk_R) \leftarrow \mathrm{KeyGen}_R(prm)$

Step 2. $(\mu_0, \mu_1, pk_S^*, sk_S^*, st) \leftarrow \mathcal{A}_1^{\mathcal{O}(\cdot)}(prm, pk_R)$

Step 3. $b \overset{U}{\leftarrow} \{0,1\}, \sigma^* \leftarrow \mathrm{SC}(prm, pk_R, sk_S^*, \mu_b)$

Step 4. $b' \leftarrow \mathcal{A}_2^{\mathcal{O}(\cdot)}(st, \sigma^*)$

ここで $\mu_0 \neq \mu_1$ かつ $|\mu_0| = |\mu_1|$ であり st は状態情報とする．*unsigncrypt* オラクル $\mathcal{O}(\cdot)$ はクエリ (pk_S, σ) を入力として受け取ったら，$\mu/\perp \leftarrow \mathrm{USC}(prm, pk_S, sk_R, \sigma)$ を返し，$\mathcal{A}$ はいつでもオラクルにクエリを聞くことができる．ただし，$\mathcal{A}_2$ はオラクルにクエリ (pk_S^*, σ^*) を聞くことはできない．すべての多項式時間アルゴリズム $\mathcal{A}$ に対して，$\mathcal{A}$ のアドバンテージ $Adv^{\text{MU-IND-iCCA}}_{\mathrm{SCS},\mathcal{A}}(n) := |\Pr[b' = b] - 1/2|$ が無視できるほど小さいとき，SCS は *MU-IND-iCCA* 安全であるという．

定義 14 (MU-sUF-iCMA)**.** Signcryption 方式 $\mathrm{SCS} = (\mathrm{Setup}, \mathrm{KeyGen}_R, \mathrm{KeyGen}_S, \mathrm{SC}, \mathrm{USC})$ に対して，*MU-sUF-iCMA* は次のように定義される．$\mathcal{A}$ を SCS に対する攻撃者 *(*多項式時間アルゴリズム*)* として，次のゲームを考える．

Step 1. $prm \leftarrow \mathrm{Setup}(1^n)$,
$(pk_S, sk_S) \leftarrow \mathrm{KeyGen}_S(prm)$

Step 2. $(pk_R^*, sk_R^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}(\cdot)}(prm, pk_S)$

$\mathcal{A}$ は高々 Q 回，*signcrypt* オラクル $\mathcal{O}(\cdot)$ に問い合わせることができ，(pk_R, μ) を入力として $\mathrm{SC}(prm, pk_R, sk_S, \mu)$ を返す．$\{(pk_R^{(1)}, \mu^{(1)}, \sigma^{(1)}), \ldots, (pk_R^{(Q)}, \mu^{(Q)}, \sigma^{(Q)})\}$ をオラクルに対するクエリとその応答の集合とする．$\mathcal{A}$ が勝つ事象 [$\mathcal{A}$ wins] を次のように定義する*:* $\mathrm{USC}(prm, pk_s, sk_R^*, \sigma^*) = \mu^* \wedge\ i = 1, \ldots, Q$ に対して $(pk_R^*, \mu^*, \sigma^*) \neq (pk_R^{(i)}, \mu^{(i)}, \sigma^{(i)})$

すべての多項式時間アルゴリズム $\mathcal{A}$ に対して，$\mathcal{A}$ のアドバンテージ $Adv^{\text{MU-sUF-iCMA}}_{\mathrm{SCS},\mathcal{A}}(n) := \Pr[\mathcal{A} \text{ wins}]$ が無視できるほど小さいとき，SCS は *MU-sUF-iCMA* 安全であるという．

3.2 提案方式：一般的構成法

本節では，タグベーストラップドア置換と MAC, Encapsulation, (sUF-naCMA 安全な) ディジタル署名，カメレオンハッシュを用いた一般的構成法を示す．基本的アイデアは，まずタグベーストラップドア置換とそのハードコア述語を用いて IND-sTag-CCA のタグベース暗号を構成する．次に，これと MAC と Encapsulation を用いて MU-IND-iCCA を満たすように構成する．また，カメレオンハッシュは sUF-naCMA 安全なディジタル署名から sUF-CMA 安全なディジタル署名を構成するために利用する．

提案方式は Sign-then-Enc で構成されるが，IND-CCA 安全な公開鍵暗号と sUF-CMA 安全なディジタル署名を単純に組み合わせて構成しても MU-sUF-iCMA 安全性を満たさないことが知られている ([2], [12])．この事は，MU-sUF-iCMA ゲームにおいて受信者の秘密鍵を持っている内部攻撃者は，オラクルからの応答である暗号文からメッセージと署名を復号することができ，それらに対して再び暗号化すれば検証を通る偽造を生成できることに起因する．これを解決するために，タグベーストラップドア置換とそのハードコア述語を用いて構成する．署名を生成する際に，メッセージ μ と受信者公開鍵 pk_R だけでなくタグベーストラップドア置換の出力 $c_0 \leftarrow \mathrm{F}_{tag}(ek, tag, x)$ に対しても署名 S を生成する．そして，入力値 x とハードコア述語 hc を使って μ, S に対する暗号文 $c_1 \leftarrow hc(x) \oplus (\mu || S)$ を生成する．これによって，上記の攻撃方法で MU-sUF-iCMA を破るには x を偽造する必要がある．しかし，x と c_0 は 1 対 1 に対応しているので，ディジタル署名の sUF-CMA を破らなければ，そのような偽造を生成することはできない，というのが構成のアイデアである．

以下では，セキュリティパラメータを k として，タグベーストラップドア置換 TBTDP = (Kg, F, F^{-1}), TBTDP のハードコア述語 $hc : \{0,1\}^k \to \{0,1\}$, ディジタル署名 DS = (Setup, Kg, Sign, Vrfy), カメレオンハッシュ CH = (Gen, Hash, Hash^{-1}), MAC = (Sign, Vrfy), Encapsulation E = (Setup, Enc, Rec) をプリミティブとして利用する．メッセージ空間を $\{0,1\}^\ell$, ディジタル署名の署名長を s ビット, Encapsulation の dec サイズを d ビット，カメレオンハッシュの値域を $\{0,1\}^\lambda$ とする．本稿で提案する $\mathrm{SCS} = (\mathrm{Setup}, \mathrm{KeyGen}_R, \mathrm{KeyGen}_S, \mathrm{SC}, \mathrm{USC})$ は次のように構成される．

- $\mathrm{Setup}(1^k)$:
  $prm_{DS} \leftarrow \mathrm{DS.Setup}(1^k)$,
  $prm_E \leftarrow \mathrm{E.Setup}(1^k)$,
  Output $prm = (prm_{DS}, prm_E)$.
- $\mathrm{KeyGen}_R(prm)$:
  $(ek_R, TD_R) \leftarrow \mathrm{TBTDP.Kg}(1^k)$,
  Output $pk_R = ek_R$, $sk_R = TD_R$.
- $\mathrm{KeyGen}_S(prm)$:
  $(vk_{DS}, sk_{DS}) \leftarrow \mathrm{DS.Kg}(prm_{DS})$,
  $(hk_S, td_S) \leftarrow \mathrm{CH.Gen}(1^k)$,
  Output $pk_S = (vk_{DS}, hk_S), sk_S = (sk_{DS}, td_S)$.

- SC($prm, pk_R, sk_S, \mu \in \{0,1\}^\ell$):
  - $(r, com, dec) \leftarrow$ E.Encap(prm_E),
  - $tag \leftarrow com$,
  - For $i \in [\ell + s + d]$, $x_i \xleftarrow{R} \{0,1\}^k$, $c_{0,i} \leftarrow$ TBTDP.F(ek_R, tag, x_i),
  - $c_0 \leftarrow (c_{0,1}, \ldots, c_{0,\ell+s+d})$ とする.
  - μ, pk_R, c_0 に対する署名を生成する.
    - $r_S \xleftarrow{R} \mathcal{R}$,
    - $h_S \leftarrow$ CH.Hash($hk_S, \mu||pk_R||c_0||com, r_S$),
    - $S_{DS} \leftarrow$ DS.Sign(sk_{DS}, h_S),
    - 署名 $S \leftarrow (S_{DS}, r_S)$.
  - μ, S, dec に対する暗号文を生成する.
    - $\mu' \leftarrow \mu||S||dec$,
    - For $i \in [\ell + s + d]$, $c_{1,i} \leftarrow hc(x_i) \oplus \mu'_i$,
    - $c_1 \leftarrow (c_{1,1}, \ldots, c_{1,\ell+s+d})$,
    - 暗号文 $C \leftarrow (c_0, c_1)$ とする.
  - $\tau \leftarrow$ MAC.Sign($r, C||pk_S$)
  - $\sigma = (com, C, \tau)$ を出力する.
- USC(prm, pk_S, sk_R, σ):
  - $tag \leftarrow com$,
  - For $i \in [\ell + s]$, $x_i \xleftarrow{R} \{0,1\}^k$, $\mu'_i \leftarrow c_{1,i} \oplus hc($TBTDP.F$^{-1}(TD_R, tag, c_{0,i}))$,
  - μ' を (μ, S_{DS}, r_S, dec) に分解する,
  - $r \leftarrow$ E.Rec(prm_E, com, dec),
  - $1 \leftarrow$ MAC.Vrfy($r, C||pk_S, \tau$) ならば, $h_S \leftarrow$ CH.Hash($hk_S, \mu||pk_R||c_0||com, r_S$) を計算する,
  - $1 \leftarrow$ DS.Vrfy(vk_{DS}, S_{DS}, h_S) ならば, μ を出力する. そうでなければ, $\perp$ を出力する.

上記の SCS の安全性に関して次が成り立つ.

定理 1. *タグベーストラップドア置換が TB-AOW 安全, MAC が sUF-OT 安全, Encapsulation が安全ならば, 上記の SCS は MU-IND-iCCA 安全である.*

定理 2. *ディジタル署名が sUF-naCMA 安全, カメレオンハッシュが衝突困難性をみたし, Encapsulation が Binding をみたすならば, 上記の SCS は MU-sUF-iCMA 安全である.*

3.3 安全性証明

3.3.1 定理 1 の証明

基本的に BK 変換 [3] の方法に従う. MU-IND-iCCA 安全性を破る攻撃者を $\mathcal{A}$ とする. $Game_0$ を通常の MU-IND-iCCA ゲームとして, このゲームから少しずつ変換した次のような $Game_1, Game_2, Game_3$ を考える.

$Game_1$: $\mathcal{A}$ がクエリ $(pk_S, \sigma = (com^*, C, \tau))$ を出力したとき, オラクルは $\perp$ を返す. それ以外は $Game_0$ と同じである.

$Game_2$: $\mathcal{A}$ が $(\mu_0, \mu_1, pk_S, sk_S)$ を出力したとき, 次のような暗号文 σ^* を返す: Challenge フェーズにおいて $hc(x)$ のかわりに $hc^* \xleftarrow{U} \{0,1\}^{\ell+s+d}$ を用いて $\mu' = 0^\ell||S^*||0^d$ に対して C^* を計算し, $\tau^* \leftarrow$ MAC.Sign($r^*, C^*||pk^*_S$) とする. それ以外は $Game_1$ と同じである.

$Game_3$: com^* と C^* は $Game_2$ と同じだが, 一様ランダムな鍵 r^* を使って $\tau^* \leftarrow$ MAC.Sign($r^*, C^*||pk^*_S$) を計算する.

さらに, 次のような事象を定義する.

$Succ_i$: $Game_i$ において, $\mathcal{A}$ が勝つ事象

$Valid_i$: $Game_i$ において, $\mathcal{A}$ が暗号文 $\sigma = (com^*, C, \tau)$ を出力する事象

$NoBind_i$: $Game_i$ において, $\mathcal{A}$ が次を満たす暗号文 $\sigma = (com^*, C, \tau)$ を出力する事象: $x \leftarrow$ TBTDP.F$^{-1}(TD_R, tag, c_1), \mu||S||dec \leftarrow c_2 \oplus hc(x)$ かつ E.Rec(prm_E, com^*, dec) $= r \notin \{r^*, \perp\}$.

$Forge_i$: $Game_i$ において, $\mathcal{A}$ が次を満たす暗号文 $\sigma = (com^*, C, \tau)$ を出力する事象: MAC.Vrfy(r^*, C, τ) $= 1$.

MU-IND-iCCA を破る多項式時間アルゴリズム $\mathcal{A}$ のアドバンテージは

$$
\begin{aligned}
&Adv^{\text{MU-IND-iCCA}}_{\text{SCS},\mathcal{A}}(k) = \left|\Pr[\mathsf{Succ}_0] - \frac{1}{2}\right| \\
&\leq |\Pr[\mathsf{Succ}_0] - \Pr[\mathsf{Succ}_1]| + \left|\Pr[\mathsf{Succ}_1] - \frac{1}{2}\right| \\
&\leq \Pr[\mathsf{NoBind}_1] + \Pr[\mathsf{Forge}_1] \\
&\quad + |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]| + \left|\Pr[\mathsf{Succ}_2] - \frac{1}{2}\right| \\
&\leq \Pr[\mathsf{NoBind}_1] + \Pr[\mathsf{Forge}_3] \\
&\quad + |\Pr[\mathsf{Forge}_2] - \Pr[\mathsf{Forge}_3]| \\
&\quad + |\Pr[\mathsf{Forge}_1] - \Pr[\mathsf{Forge}_2]| \\
&\quad + |\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]| + \left|\Pr[\mathsf{Succ}_2] - \frac{1}{2}\right|.
\end{aligned}
$$

である. $\Pr[\mathsf{NoBind}_1]$ は Encapsulation の Binding から, $\Pr[\mathsf{Forge}_3]$ は MAC の sUF-OT 安全性から, $|\Pr[\mathsf{Forge}_2] - \Pr[\mathsf{Forge}_3]|$ は Encapsulation の Hiding から, $|\Pr[\mathsf{Forge}_1] - \Pr[\mathsf{Forge}_2]|$ と $|\Pr[\mathsf{Succ}_1] - \Pr[\mathsf{Succ}_2]|$ はタグベーストラップドア置換の TB-AOW とハードコア述語の安全性から無視できる値となる. よって, $Adv^{\text{MU-IND-iCCA}}_{\text{SCS},\mathcal{A}}(k) \leq \text{negl}(k)$ であり, SCS は MU-IND-iCCA 安全であることが示された.

3.3.2 定理 2 の証明

$\mathcal{A}$ を MU-sUF-iCMA を破る多項式時間アルゴリズムとして, $\mathcal{A}$ を利用してカメレオンハッシュの衝突困難性, Encapsulation の Binding, あるいはディジタル署名の sUF-naCMA を破る多項式時間アルゴリズム $\mathcal{S}$ を構成する.

$\mathcal{A}$ が出力する高々 Q 個のクエリを $(\mu^{(1)}, pk_R^{(1)}), \ldots, (\mu^{(Q)}, pk_R^{(Q)})$ とする. また, $i \in [Q]$ に対して, $M^{(i)} := \mu^{(i)}||pk_R^{(i)}||c_0^{(i)}||com^{(i)}$ とする. 攻撃者 $\mathcal{A}$ は次のタイプに分類される.

Type-1. カメレオンハッシュ CH の衝突を見つけることで偽造を生成する.

Type-2. Encapsulation の Binding を破って偽造を生成する.

Type-3. CH の衝突困難性と Encapsulation の Binding を破らずに偽造を生成する.

(a) クエリで発行されたメッセージと公開鍵のペア (μ, pk_R) を使わずに偽造を生成する.

(b) クエリで発行されたメッセージと公開鍵のペア (μ, pk_R) を使って新しい偽造を生成する.

$\mathcal{A}$ が Type-1 の場合を考える. $\mathcal{A}$ を使って $\mathcal{S}$ を次のように構成する. $\mathcal{S}$ は入力としてカメレオンハッシュ CH のハッシュ鍵 hk_S を受け取る. $prm \leftarrow \mathrm{SCS.Setup}(1^k), (vk_S, sk_S) \leftarrow \mathrm{DS.Kg}(prm_{DS})$ を生成し, $pk_S \leftarrow (vk_S, hk_S)$ とする. クエリ $(\mu^{(i)}, pk_R^{(i)}), i \in [Q]$ に対して SCS.SC アルゴリズムに従って $\sigma^{(i)}$ を計算して返す. $\mathcal{A}$ が偽造 $(pk_R^*, sk_R^*, \sigma^* = (com^*, C^*, \tau^*))$ を出力したら, SCS.USC アルゴリズムに従って μ^* を計算する. クエリの中から $\mathrm{CH.Hash}(hk_S, M^{(i)}, r_S^{(i)}) = \mathrm{CH.Hash}(hk_S, M^*, r_S^*)$ を満たす $(M^{(i)}, r_S^{(i)})$ を見つけ, CH の衝突 $(M^*, r_S^*), (M^{(i)}, r_S^{(i)})$ を出力する.

$\mathcal{A}$ が Type-2 の場合, Encapsulation の Binding を破る $\mathcal{S}$ を次のように構成する. $\mathcal{S}$ は入力として (prm, com^*, dec^*) を受け取る. $(pk_S, sk_S) \leftarrow \mathrm{SCS.KeyGen}_S(prm)$ を生成する. $j \xleftarrow{R} \{1, \ldots, Q\}$ を選んで, j 番目のクエリに対して (r^*, com^*, dec^*) を用いて $\sigma^{(j)}$ を計算して $\mathcal{A}$ に返す. $\mathcal{A}$ が $(pk_R^*, sk_R^*, \sigma^* = (com^*, C^*, \tau^*))$ を出力したら, $\mathcal{S}$ は sk_R^* を使って dec' を復号する. $\mathrm{E.Rec}(prm, com^*, dec') \notin \{r^*, \perp\}$ であれば, $\mathcal{S}$ は dec' を出力する.

$\mathcal{A}$ が Type-3a, または Type-3b の場合, (s)UF-naCMA を破る $\mathcal{S}$ を次のように構成する. $\mathcal{S}$ は最初に $h_S^{(i)} \xleftarrow{R} \{0,1\}^\lambda, i \in [Q]$ を選んで出力する. $(prm_{DS}, vk_S, S_{DS}^{(1)}, \ldots, S_{DS}^{(Q)})$ を受け取ったら, $(hk_S, td_S) \leftarrow \mathrm{CH.Gen}(1^k)$ を生成して $pk_S \leftarrow (vk_S, hk_S)$ とする. クエリ $(\mu^{(i)}, pk_R^{(i)})$ を受け取ったら, USC.SC アルゴリズムに従って $M^{(i)} = \mu^{(i)} || pk_R^{(i)} || c_0^{(i)} || com^{(i)}$ として $r_S^{(i)} \leftarrow \mathrm{CH.Hash}^{-1}(td_S, M^{(i)}, h_S^{(i)})$ を計算する. これらを用いて $\sigma^{(i)}$ を計算して返す. $\mathcal{A}$ が $(pk_R^*, sk_R^*, \sigma^* = (com^*, C^*, \tau^*))$ を出力したら, $\mu^* \leftarrow \mathrm{SCS.USC}(prm, pk_S^*, sk_R^*, \sigma^*)$ を計算する. $\mathcal{S}$ は $h_S^* \leftarrow \mathrm{CH.Hash}(hk_S, \mu^* || pk_R^* || c_0^* || com^*, r_S^*)$ と, これに対する署名 S_{DS}^* を出力する.

$\mathcal{A}$ が Type-3a の場合, $\mathcal{S}$ が出力する値はすべての $i \in [Q]$ に対して $h_S^* \neq h_S^{(i)}$ かつ $\mathrm{DS.Vrfy}(vk_S, S_{DS}^*, h_S^*) = 1$ を満たす (h_S^*, S_{DS}^*) だから, UF-naCMA を破っている. また, $\mathcal{A}$ が Type-3b の場合, $\mathcal{S}$ が出力する値は, ある $i \in [Q]$ に対して $h_S^* = h_S^{(i)}$ かつ $\mathrm{DS.Vrfy}(vk_S, S_{DS}^*, h_S^*) = 1$ を満たす (h_S^*, S_{DS}^*) だから, sUF-naCMA を破っている.

以上より, SCS は MU-sUF-iCMA 安全であることが証明された.

3.4 提案方式：格子ベースの構成

3.2 節の構成法をそのまま適用すると, メッセージと署名を 1 ビット暗号化するたびにタグベーストラップドア置換の出力値が増えていくため, 暗号文長が $\tilde{O}(n^2)$ になってしまう. そのため, 本節では格子ベースの構成法を改めて考える. 基本的に一般的構成法と同じだが, この構成ではメッセージと署名に対して暗号化する際にハードコア述語ではなく LWE の値を加算して暗号化する. 格子ベースの構成法を以下に示す.

- $prm \leftarrow \mathrm{Setup}(1^n)$:
  セキュリティパラメータ n を十分大きい正の整数として, 素数 $q = \mathrm{poly}(n)$, $k = \lceil \log q \rceil = O(\log n)$, $\bar{m} = O(nk)$, $m = \bar{m} + nk$, $\alpha^{-1} = O(nk)^2 \cdot \omega(\sqrt{\log n})$, $\delta = \sqrt{\lambda nk} \cdot \omega(\sqrt{\log n})$ とする. p は δ より十分大きく $\alpha q < q/(2p)$ を満たす整数とする. ℓ はメッセージ長, λ はカメレオンハッシュ関数のハッシュ値の長さとする.
  - $\mathbf{G} = \begin{bmatrix} \mathbf{g}^t & & 0 \\ & \ddots & \\ 0 & & \mathbf{g}^t \end{bmatrix} \in \mathbb{Z}_q^{n \times nk}$
    $(\mathbf{g}^t = \left[1, 2, 2^2, \ldots, 2^{k-1}\right] \in \mathbb{Z}_q^{1 \times k})$,
  - FRD encoding $H([1])$: $\mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$,
  - $prm_E \leftarrow \mathrm{E.Setup}(1^k)$,

  $prm = (q, \bar{m}, m, \alpha, \delta, p, \mathbf{G}, H, prm_E)$ を出力する.
- $\mathrm{KeyGen}_R(prm)$: 公開パラメータ prm を入力として, 次の処理を行う.
  - $\bar{\mathbf{A}}_R \xleftarrow{U} \mathbb{Z}_q^{n \times \bar{m}}, \mathbf{R}_R \leftarrow D_{\mathbb{Z}, \omega(\sqrt{\log n})}^{\bar{m} \times nk}$ を選ぶ.
  - $\mathbf{A}_R \leftarrow \left[\bar{\mathbf{A}}_R | -\bar{\mathbf{A}}_R \mathbf{R}_R\right]$ とする.
  - $\mathbf{U}_{R,1} \xleftarrow{U} \mathbb{Z}_q^{n \times \ell}, \mathbf{U}_{R,2} \xleftarrow{U} \mathbb{Z}_q^{n \times (2m+nk)}, \mathbf{U}_{R,3} \xleftarrow{U} \mathbb{Z}_q^{n \times m}$ を選ぶ.
  - 受信者の公開鍵 $pk_R = (\mathbf{A}_R, \mathbf{U}_{R,1}, \mathbf{U}_{R,2}, \mathbf{U}_{R,3})$, 秘密鍵 $sk_R = \mathbf{R}_R$ を出力する.
- $\mathrm{KeyGen}_S(prm)$: 公開パラメータ prm を入力として, 次の処理を行う.
  - $\bar{\mathbf{A}}_S \xleftarrow{U} \mathbb{Z}_q^{n \times \bar{m}}, \mathbf{R}_S \leftarrow D_{\mathbb{Z}, \omega(\sqrt{\log n})}^{\bar{m} \times nk}$ を選ぶ.
  - $\mathbf{A}_S \leftarrow \left[\bar{\mathbf{A}}_S | \mathbf{G} - \bar{\mathbf{A}}_S \mathbf{R}_S\right]$ とする.
  - $\mathbf{u}_S \xleftarrow{U} \mathbb{Z}_q^n, \mathbf{A}_0, \ldots, \mathbf{A}_\lambda \xleftarrow{U} \mathbb{Z}_q^{n \times nk}$ を選ぶ.
  - $(hk_S, td_S) \leftarrow \mathrm{CH.Gen}(1^k)$ を生成する.
  - 送信者の公開鍵 $pk_S = (\mathbf{A}_S, \mathbf{u}_S, \mathbf{A}_0, \ldots, \mathbf{A}_\lambda, hk_S)$, 秘密鍵 $sk_S = \mathbf{R}_S$ を出力する.
- $\mathrm{SC}(prm, pk_R, sk_S, \mu \in \{0,1\}^\ell)$:
  - $(r, com, dec) \leftarrow \mathrm{E.Enc}(prm_E)$, $tag \leftarrow com$,
  - $\mathbf{A}_{R,tag} \leftarrow \left[\bar{\mathbf{A}}_R | H(tag)\mathbf{G} - \bar{\mathbf{A}}_R \mathbf{R}_R\right] \in \mathbb{Z}_q^{n \times m}$,
  - $\mathbf{s} \xleftarrow{U} \mathbb{Z}_q^n, \mathbf{x}_0 \leftarrow D_{\alpha q}^m, \mathbf{x}_1 \leftarrow D_{\alpha q}^\ell, \mathbf{x}_2 \leftarrow D_{\alpha q}^{2m+nk}, \mathbf{x}_3 \leftarrow D_{\alpha q}^m$,

- $\mathbf{c}_0 \leftarrow \mathbf{s}^t\mathbf{A}_{R,tag} + \mathbf{x}_0^t \in \mathbb{Z}_q^m$,
- $\bar{\mathbf{c}}_1 \leftarrow \mathbf{s}^t\mathbf{U}_{R,1} + \mathbf{x}_1^t \in \mathbb{Z}_q^\ell$, $\bar{\mathbf{c}}_2 \leftarrow \mathbf{s}^t\mathbf{U}_{R,2} + \mathbf{x}_2^t \in \mathbb{Z}_q^{2m+nk}$, $\bar{\mathbf{c}}_3 \leftarrow \mathbf{s}^t\mathbf{U}_{R,3} + \mathbf{x}_3^t \in \mathbb{Z}_q^m$,
- $\bar{C} \leftarrow (\mathbf{c}_0, \bar{\mathbf{c}}_1, \bar{\mathbf{c}}_2, \bar{\mathbf{c}}_3)$ とする.
- $\mu, pk_R, \bar{C}$ に対する署名を生成する.
  - $r_S \xleftarrow{R} D_\delta^m$,
  - $h_S \leftarrow \mathrm{CH.Hash}(hk_S, \mu||pk_R||\bar{C}||com, r_S)$,
  - $\mathbf{A}_{S,h_S} \leftarrow \left[\mathbf{A}_S|\mathbf{A}_0 + \sum_i^\lambda h_{S,i}\mathbf{A}_i\right]$,
  - $\mathbf{e} \leftarrow \mathrm{SampleD}(\mathbf{R}_S, \mathbf{A}_{S,h_S}, \mathbf{u}_S, \delta)$,
  - 署名 $S \leftarrow (\mathbf{e}, r_S) \in \mathbb{Z}^{m+nk} \times \mathbb{Z}^m$ とする.
- $\mu||S||dec$ を暗号化する.
  - $\mathbf{c}_1 \leftarrow \bar{\mathbf{c}}_1 + \mu\lfloor q/2 \rfloor \in \mathbb{Z}_q^\ell$,
  - $\mathbf{c}_2 \leftarrow \bar{\mathbf{c}}_2 + S\lfloor q/p \rfloor \in \mathbb{Z}_q^{2m+nk}$,
  - $\mathbf{c}_3 \leftarrow \bar{\mathbf{c}}_3 + dec\lfloor q/p \rfloor \in \mathbb{Z}_q^m$,
  - $C \leftarrow (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$ とする.
- $\tau \leftarrow \mathrm{MAC.Vrfy}(r, C||pk_S)$,
- $\sigma = (com, C, \tau)$ を出力する.

- $\mathrm{USC}(prm, pk_S, sk_R, \sigma)$:
- $tag \leftarrow com$, $\mathbf{A}_{R,tag} \leftarrow \left[\bar{\mathbf{A}}_R|H(tag)\mathbf{G} - \bar{\mathbf{A}}_R\mathbf{R}_R\right]$,
- $(\mathbf{s}, \mathbf{x}_0) \leftarrow \mathrm{Invert}(\mathbf{R}_R, \mathbf{A}_{tag}, \mathbf{c}_0)$,
- $\mu\lfloor q/2 \rfloor + \mathbf{x}_1 \leftarrow \mathbf{c}_1 - \mathbf{s}^t\mathbf{U}_{R,1}$, $S\lfloor q/p \rfloor + \mathbf{x}_2 \leftarrow \mathbf{c}_2 - \mathbf{s}^t\mathbf{U}_{R,2}$, $dec\lfloor q/p \rfloor + \mathbf{x}_3 \leftarrow \mathbf{c}_3 - \mathbf{s}^t\mathbf{U}_{R,3}$,
- $r \leftarrow \mathrm{E.Rec}(prm, com, dec)$,
- $0 = \mathrm{MAC.Vrfy}(r, C||pk_S, \tau)$ ならば, $\perp$ を出力する.
- $\bar{\mathbf{c}}_1 \leftarrow \mathbf{c}_1 - \mu\lfloor q/2 \rfloor \bmod q, \bar{\mathbf{c}}_2 \leftarrow \mathbf{c}_2 - S\lfloor q/p \rfloor \bmod q, \bar{\mathbf{c}}_3 \leftarrow \mathbf{c}_3 - dec\lfloor q/2 \rfloor \bmod q$,
- $\bar{C} \leftarrow (\mathbf{c}_0, \bar{\mathbf{c}}_1, \bar{\mathbf{c}}_2, \bar{\mathbf{c}}_3)$ とする,
- S を $(\mathbf{e}, r_S)$ に分割する,
- $h_S \leftarrow \mathrm{CH.Hash}(hk_S, \mu||pk_R||\bar{C}||com, r_S)$,
- $\mathbf{A}_{S,h_S} \cdot \mathbf{e} = \mathbf{u}_S \bmod q, |\mathbf{e}| \le \delta\sqrt{m+nk}$ ならば, μ を出力する. そうでなければ, $\perp$ を出力する.

定理 1, 2 から, 上記の構成に関しても次が成り立つ.

定理 3. $\alpha' = \alpha/2 \ge 2\sqrt{n}/q$ に対して $\mathrm{LWE}_{q,\alpha'}$ 仮定が成り立ち, *MAC* が *sUF-OT* 安全, *Encapsulation* が安全ならば, 上記の *SCS* は *MU-IND-iCCA* 安全である.

定理 4. $\beta = O((nk)^{5/2}) \cdot \omega(\sqrt{\log n})^2$ に対して $\mathrm{SIS}_{q,\beta}$ 仮定が成り立ち, カメレオンハッシュが衝突困難, *Encapsulation* が *Binding* を満たすならば, 上記の *SCS* は *MU-sUF-iCMA* 安全である.

4. 比較

本節では, 既存の構成法との比較を行う. 格子ベースの Signcryption として, 著者らは SCIS2016 にて具体的な構成法 [16] を提案した. また, MU-IND-iCCA と MU-sUF-iCMA を満たす Signcryption の一般的構成法として, タグベース KEM を利用する $\mathcal{SC}_{TK}$[6] と CHK 変換をベースにした $\mathcal{SC}_{CHK}$[14] がある. これらの構成法に必要な各プリミティブには以下の格子ベースの構成法を適用する.

- $\mathcal{SC}_{TK}$[6]: IND-Tag-CCA 安全なタグベース KEM(付録と [5]), sUF-CMA 安全なディジタル署名 ([13] と [5]), IND-CCA 安全な DEM.
- $\mathcal{SC}_{CHK}$[14]: IND-sID-CPA 安全な ID ベース暗号 ([1]), sUF-OT 安全なワンタイム署名 ([11]), UF-CMA 安全なディジタル署名 ([4]).

ただし, $\mathcal{SC}_{TK}$[6] でのプリミティブに関して, IND-Tag-CCA 安全なタグベース KEM の格子ベース構成は提案されていないが, タグベーストラップドア関数 [13] を使って IND-sTag-CCA 安全なタグベース KEM を構成することができ (付録参照), さらにカメレオンハッシュ [5] を用いて IND-Tag-CCA を満たすことができる [7]. また, sUF-CMA 安全なディジタル署名は, sUF-naCMA 安全なディジタル署名 [13] とカメレオンハッシュ [5] を使って構成する.

公開鍵長と暗号文長について既存の構成法と比較した結果が表 1 である. SCIS2016 で提案した構成法 [16] を提案方式 1 とし, 本稿で提案した構成法を提案方式 2 とする. 受信者秘密鍵と送信者秘密鍵の鍵長はすべての構成法において $\tilde{O}(n^2)$ である.

表 1 より, すべてスタンダードな LWE/SIS で構成した場合に $|vk| = \tilde{O}(n^2)$ になるので, 暗号文長において提案方式 1 よりも提案方式 2 の方が短い. 例えば, 具体的な値として, $n = 256, q = 2^{24}, m = 3n\log q$ を適用して比較した場合, $\mathcal{SC}_{TK}$ が最も暗号文長が短い. 受信者の公開鍵長において, 提案方式 2 の公開鍵 $\mathbf{U}_{R,1}, \mathbf{U}_{R,2}, \mathbf{U}_{R,3}$ を公開パラメータとみなすことができる. 他の構成法も同様に冗長な部分を公開パラメータとみなして係数で比較した場合, $|CH_R| = \tilde{O}(n^2)$ の分だけ提案方式 2 が最も短く, この点において利点がある.

方式	公開鍵 (受信者)	公開鍵 (送信者)	暗号文長
$\mathcal{SC}_{TK}$[6]	$O(n^2\log^2 n + Kn\log n) + \lvert CH_R\rvert$	$O(n^3\log^3 n) + \lvert CH_S\rvert$	$O(n\log^2 n + K\log n + \ell)$
$\mathcal{SC}_{CHK}$[14]	$O(n^2\log^2 n + \ell n\log n)$	$O(n^4\log^4 n + \ell n^3\log^3 n)$	$O(n\log^2 n + \ell\log n) + \lvert vk\rvert$
提案方式 1[16]	$O(n^2\log^2 n + \ell n\log n)$	$O(n^3\log^3 n) + \lvert CH_S\rvert$	$O(n\log^2 n + \ell\log n) + \lvert vk\rvert$
提案方式 2(本論文)	$O(n^2\log^2 n + \ell n\log n)$	$O(n^3\log^3 n) + \lvert CH_S\rvert$	$O(n\log^2 n + \ell\log n) + \lvert MAC\rvert$

表 1 鍵長と暗号文長の比較: n はセキュリティパラメータ, ℓ はメッセージ長, $|vk|$ はワンタイム署名の検証鍵長, $|CH|$ はカメレオンハッシュ鍵の鍵長, $|MAC|$ は MAC のタグ長, K は DEM の鍵長

なお, [6] では, CCA 安全な KEM を用いた別の構成法が提案されているが, 格子ベースの CCA 安全な KEM[15] を

用いた場合, 鍵長・暗号文長において $\mathcal{SC}_{TK}, \mathcal{SC}_{CHK}$ よりも非効率的であるため, 本稿では比較を省略する.

5. まとめ

Signcrypion の一般的構成法として, タグベーストラップドア置換と sUF-naCMA 安全なディジタル署名, カメレオンハッシュ, MAC, Encapsulation を利用する一般的構成法を提案し, これをベースとして格子問題に基づく構成を示した. また, この構成法と既存の構成法について鍵長と暗号文長の比較を行った結果, 次のことが示された. 受信者公開鍵長と送信者公開鍵長において本論文で提案した構成法が最も効率が良く, 暗号文長においては [14] と [16] の構成より優れているが, [6] よりも劣っている.

参考文献

[1] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In EUROCRYPT, volume 6110 of LNCS, pages 553-572. Springer, 2010.

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

[3] D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. SIAM J. Conput., 36(5):1301-1328, 2007.

[4] X. Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In PKC, volume 6056 of LNCS, pages 499-517. Springer, 2010.

[5] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In EUROCRYPT, volume 6110 of LNCS, pages 523-552. Springer, 2010.

[6] D. Chiba, T. Matsuda, J. C. N. Schuldt, Efficient Generic Constructions of Signcryption with Insider Security in the Multi-user Setting. In ACNS, volume 6715 of LNCS, pages 220-237, Springer, 2011

[7] 千葉大輝, 松田隆宏, 松浦幹太. "タグベース KEM の選択的タグ安全性から適応的タグ安全性へのカメレオンハッシュを用いた強化手法と Signcryption への応用," SCIS2010, 3A2-1, 2010.

[8] E. Kiltz, P. Mohassel, and A. O ' Neill. Adaptive trapdoor functions and chosen-ciphertext security. In EUROCRYPT, volume 6110 of LNCS, pages 673-692. Springer, 2010.

[9] H. Krawczyk and T. Rabin. Chameleon hashing and signatures. In NDSS, pages 143-154. The Internet Society, 2000.

[10] B. Libert and J. Quisquater. Improved signcryption with key privacy from gap diffie-hellman groups. In PKC, volume 2947 of LNCS, pages 187-200. Springer, 2004.

[11] V. Lyubashevsky and D. Micciancio. Asymptotically efficient lattice-based digital signatures. In TCC, volume 4948 of LNCS, pages 37-54. Springer, 2008.

[12] T. Matsuda, K. Matsuda, and J. C. N. Schuldt, Efficient constructions of signcryption schemes and signcryption composability. In INDOCRYPT, volume 5922 of LNCS, pages 321-342, Springer, 2009.

[13] D. Micciancio, and C.Peikert, Trapdoors for lattices: Simpler, tighter, faster, smaller. In EUROCRYPT, volume 7237 of LNCS, pages 700-718. Springer, 2012.

[14] R. Nakano and J. Shikata. Constructions of signcryption from identity-based encryption. In IMACC, volume 8308 of LNCS, pages 324-343. Springer, 2013.

[15] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In STOC, pages 333-342. ACM, 2009.

[16] 佐藤慎悟, 四方順司. "格子問題に基づく Signcryption のスタンダードモデルでの構成," SCIS2016, 1D1-1, 2016.

[17] Y. Zheng. Digital signcryption or how to achieve cost(signature & encryption) $\ll$ cost(signature) + cost(encryption). In CRYPTO, volume 1294 of LNCS, pages 165-179. Springer, 1997.

付録：タグベース KEM の構成

格子ベースのタグベース KEM の具体的な構成は提案されていないが, タグベーストラップドア関数 [13] から IND-sTag-CCA 安全な格子ベースのタグベース KEM を構成できることは自明なため, その具体的構成法を記述する.

- $prm \leftarrow \mathrm{Setup}(1^n)$:
  セキュリティパラメータ n を入力とし, パラメータの値は次の通りに設定する: 素数 $q = \mathrm{poly}(n)$, $k = \lceil \log q \rceil = O(\log n)$, $\bar{m} = O(nk)$, $m = \bar{m} + nk$, $\alpha^{-1} = O(nk)^2 \cdot \omega(\sqrt{\log n})$ とする.
  $$\mathbf{G} = \begin{bmatrix} \mathbf{g}^t & & 0 \\ & \ddots & \\ 0 & & \mathbf{g}^t \end{bmatrix} \in \mathbb{Z}_q^{n \times nk}$$
  $(\mathbf{g}^t = \left[1, 2, 2^2, \ldots, 2^{k-1}\right] \in \mathbb{Z}_q^{1 \times k})$
  $prm = (n, q, k, \bar{m}, m, \alpha, \mathbf{G})$ を出力する.
- $(pk, sk) \leftarrow \mathrm{Kg}(prm)$:
  - $\bar{\mathbf{A}} \xleftarrow{U} \mathbb{Z}_q^{n \times \bar{m}}, \mathbf{R} \leftarrow D_{\omega(\log n)}^{\bar{m} \times nk}$,
  - $\mathbf{A} \leftarrow \left[\bar{\mathbf{A}} | -\bar{\mathbf{A}}\mathbf{R}\right] \in \mathbb{Z}_q^{n \times m}$,
  - $\mathbf{U} \xleftarrow{U} \mathbb{Z}_q^{n \times \ell}$,
  - $pk = (\mathbf{A}, \mathbf{U}), sk = \mathbf{R}$ を出力する.
- $(C, K) \leftarrow \mathrm{Encap}(pk, tag)$:
  - $K \xleftarrow{R} \{0,1\}^\ell$,
  - $\mathbf{s} \xleftarrow{U} \mathbb{Z}_q^n, \mathbf{x}_0 \leftarrow D_{\alpha q}^m, \mathbf{x}_1 \leftarrow D_{\alpha q}^\ell$,
  - $\mathbf{A}_{tag} \leftarrow \left[\bar{\mathbf{A}} | H(tag)\mathbf{G} - \bar{\mathbf{A}}\mathbf{R}\right] \in \mathbb{Z}_q^{n \times m}$,
  - $\mathbf{c}_0 \leftarrow \mathbf{s}^t\mathbf{A}_{tag} + \mathbf{x}_0^t \in \mathbb{Z}_q^m$,
  - $\mathbf{c}_1 \leftarrow \mathbf{s}^t\mathbf{U} + \mathbf{x}_1^t + K\lfloor q/2 \rfloor \in \mathbb{Z}_q^\ell$,
  - 暗号文 $C = (\mathbf{c}_0, \mathbf{c}_1)$ と鍵 K を出力する.
- $K \leftarrow \mathrm{Decap}(sk, tag, C)$:
  - $\mathbf{A}_{tag} \leftarrow \left[\bar{\mathbf{A}} | H(tag)\mathbf{G} - \bar{\mathbf{A}}\mathbf{R}\right]$,
  - $(\mathbf{s}, \mathbf{x}_0) \leftarrow \mathrm{Invert}(\mathbf{R}, \mathbf{A}_{tag}, \mathbf{c}_0)$,
  - $\mathbf{d} \leftarrow \mathbf{c}_1 - \mathbf{s}^t\mathbf{U} \in \mathbb{Z}_q^\ell$,
  - $\mathbf{d} = (d_1, \ldots, d_\ell)$ として, 各 d_i の値が $\lfloor q/2 \rfloor$ よりも 0 に近いとき $k_i = 0$ として, それ以外のとき $k_i = 1$ とする.
  - 鍵 $K \leftarrow (k_1, \ldots, k_\ell) \in \{0,1\}^\ell$ を出力する.