多人数環境を考慮したSigncryptionの簡潔な一般的構成法

松田 隆宏　　シュルツ ヤコブ　　松浦 幹太

東京大学生産技術研究所　153-8505 東京都目黒区駒場 4-6-1

{tmatsuda,schuldt,kanta}@iis.u-tokyo.ac.jp

あらまし 本稿では、多人数環境での安全性を考慮した Signcryption の、既存の要素技術の組み合わせによる簡潔な一般的構成法を複数提案する。本稿で提案する外部者に対する安全性を持つ方式では、共通鍵系の要素技術 (共通鍵暗号、MAC) が如何に有効かを示す。また、公開鍵暗号と署名方式の Sign-then-Encrypt 及び Encrypt-then-Sign による非常に有名で直感的な Signcryption の構成法において、”タグベース暗号”を用いることで、多人数環境における内部者安全を持つ Signcryption 方式を達成できることを示す。我々の提案手法は非常に直感的かつ簡潔であり、今後、個別の計算困難性に基づいて構成される Signcryption との比較のためのよい”ベンチマーク”となると考えられる。

Simple Generic Constructions for Signcryption Secure in the Multi-user Setting

Takahiro Matsuda　　Jacob Schuldt　　Kanta Matsuura

Instutite of Industrial Science, The University of Tokyo　4-6-1 Komaba, Meguro-ku, Tokyo 153-8505

Abstract In this paper, we propose several simple generic constructions of signcryption schemes secure in the multi-user setting from the existing primitives. Firstly, we show how symmetric primitives (symmetric-key encryption and MAC) can be used to efficiently achieve outsider security. Secondly, we show a simple optimization to the well known “sign-then-encrypt” and “encrypt-then-sign” approaches to the construction of signcryption schemes, by using tag-based encryption. Since all of our constructions are fairly simple and efficient, they provide a benchmark which can be used to evaluate future signcryption schemes.

1　はじめに

Signcryption は、Zheng [18] により導入された、公開鍵暗号 (PKE) と電子署名の機能、すなわち秘匿性と偽造不可能性を同時に達成するための要素技術である。元来の Signcryption の目的は、PKE による暗号化と署名を別々に行うよりも効率的に行うことに主眼を置かれていたこともあり、[18] では十分な安全性の議論がなされていなかったが、その後の研究 ([2, 3, 14, 9] など) において様々な安全性の定義がなされ、またそれぞれ異なった安全性定義の元、数え切れないほど多くの Signcryption の方式が提案されている ([18, 2, 14, 11, 17] など)。

最も簡潔で直感的な安全性のモデルは、一人の送信者、一人の受信者しか考えない “二人モデル” だが [2, 9]、Signcryption では二人モデルの安全性は多人数環境での安全性を一般的には意味しないため ([3] など参照)、多人数環境を考慮した安全性 ([14, 3] など) の議論が望ましい。

Signcryption では、攻撃者の種類として “外部者” 及び “内部者” という概念がある。内部者は、秘匿性を攻撃する場合は送信者の秘密鍵を、偽造不可能性を攻撃する場合は受信者の秘密鍵を知ることができる (外部者はいずれの場合も公開鍵のみが与えられる)。秘匿性と偽造不可能性は独立に考えられるため、“外部者に対する秘匿性と内部者に対する偽造不可能性” を持つ方式などがあり得る。秘匿性、偽造不可能性共に、内部者安全であれば外部者に対しても安全であるが、内部者安全性を達成するほうが一般に難しく、方式の効率が悪くなることが多い。Signcryption が使用される場面を考えれば、外部者安全性でも十分な場合も多いと考えられるため、本稿では外部者安全性と内部者安全性の組み合わさった方式も考える。

また、これまで、An ら [2] による PKE と署名の組み合わせ、及び Gorantla [11] らによる鍵交換プロトコルと Signcryption KEM の関係を議論した研究を除けば、既に十分研究されている既存要素技術を構成要素として Signcryption を構成しようとする研究は少ない。さらに、提案される方式が既存要素技術の組み合わせから構成した方式と効率の比較をされることもほとんど無い[1]。従って、そもそも効率の良い Signcryption 方式で、かつ安全性を具体的な数論的仮定に直接帰着するような構成を考える利点があるのかという点にも議論の余地があると言えよう。

[1] 例えば近年提案されたスタンダードモデルで、多人数環境で内部者安全な方式 [17] は、既存の PKE(KEM)[7] と署名方式 [5] に基づいて構成されているように見えるが、直感的なこれらの組み合わせ以上の計算が必要となっている。

本研究の貢献　本稿では、多人数環境での安全性を考慮した Signcryption の、既存の要素技術の組み合わせによる簡潔な一般的構成法を複数提案する。本稿で提案する外部者に対する安全性を持つ方式では、共通鍵系の要素技術 (共通鍵暗号、MAC) が如何に有効かを示す。これらの方式では、共通鍵系の要素技術を有効に用いるために、ある種の鍵交換方式を用いる。また、公開鍵暗号と署名方式の Sign-then-Encrypt 及び Encrypt-then-Sign による非常に有名で直感的な Signcryption の構成法において、"タグベース暗号"[15, 12] という要素技術を用いることで、従来よりも効率よく多人数環境における内部者安全を持つ Signcryption 方式を達成できることを示す。本研究で提案されたそれぞれの一般的構成法の構成要素として、既存の最も効率の良い方式を当てはめれば、同じ安全性を持つ方式間を比べると、多くの組み合わせにおいて既存の Signcryption の最も効率の良い方式と実質的に同等か、それ以上の効率を持つ方式が得られる。我々の提案手法は非常に直感的かつ簡潔であり、今後、個別の計算困難性に基づいて構成される Signcryption との比較のためのよい"ベンチマーク"となると考えられる。

なお、紙面の都合上定義の一部、具体的な Signcryption の構成の例、及び全ての安全性証明を省略した。詳細は本稿のフルバージョン [16] において言及する。

2　諸定義

本稿では、"$x||y$" は x と y の連結を表す。"PPTA" は確率多項式時間アルゴリズムを指す。特に断りが無い限り k はセキュリティパラメータを表す。$\mathsf{neg}(k)$ をあらゆる多項式 $p(k)$ と十分大きな k について $\mathsf{neg}(k) < 1/p(k)$ が成り立つ関数とする (negligible 関数)。

タグベース暗号　タグベース暗号 (TBE)[15, 12] は、直感的には暗号化と復号の際に任意の文字列 tag を入力するする様な PKE であり、以下の 4 つのアルゴリズムからなる: $\mathtt{Setup}$ は 1^k を入力として受け取り、公開パラメータ par を出力する; $\mathtt{KG}$ は par を受け取り、公開鍵/秘密鍵対 (pk, sk) を出力する; $\mathtt{TEnc}$ は par、pk、タグ tag、及び平文 m を受け取り、暗号文 c を出力する; $\mathtt{TDec}$ は par、sk、tag、及び c を受け取り、m または $\perp$ を出力する。全ての $par \leftarrow \mathtt{Setup}(1^k)$、全ての $(pk, sk) \leftarrow \mathtt{KG}(par)$、全てのタグ tag、及び全ての平文 m について、$c \leftarrow \mathtt{TEnc}(par, pk, \mathsf{tag}, m)$ ならば $m = \mathtt{TDec}(par, sk, \mathsf{tag}, c)$ が要求される。

全ての PPTA $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ に対し、

$$\Pr[par \leftarrow \mathtt{Setup}(1^k);\ \ (pk, sk) \leftarrow \mathtt{KG}(par);$$
$$(m_0, m_1, \mathsf{tag}^*, st) \leftarrow \mathcal{A}_1^{\mathtt{TDec}(par, sk_R, \cdot, \cdot)}(par, pk);$$
$$b \leftarrow \{0,1\};\ \ c^* \leftarrow \mathtt{TEnc}(par, pk, \mathsf{tag}^*, m_b);$$
$$b' \leftarrow \mathcal{A}_2^{\mathtt{TDec}(par, sk_R, \cdot, \cdot)}(st, c^*) : b' = b] \leq \tfrac{1}{2} + \mathsf{neg}(k)$$

ならば、TBE 方式 TE は $\mathtt{IND\text{-}tag\text{-}CCA}$ 安全であるという。ただし、$\mathcal{A}_2$ が (tag^*, c^*) を問い合わせることは許されない。

(タグベース) 非対話型鍵交換　非対話型鍵交換 (NIKE) は、以下の 3 つのアルゴリズムからなる (本稿では [8] のモデルの簡易版を用いる): $\mathtt{Setup}$ は 1^k を受け取り、公開パラメータ par を出力する (共有鍵空間 $\mathcal{K}$ の記述も含まれているとする); $\mathtt{KG}$ は par を受け取り、公開鍵/秘密鍵対 (pk, sk) を出力する; $\mathtt{Share}$ は par、一つのエンティティの公開鍵 pk_1、及びもう一つのエンティティの秘密鍵 sk_2 を受け取り、共有鍵 $K \in \mathcal{K}$ を出力する。全ての $par \leftarrow \mathtt{Setup}(1^k)$、及び $\mathtt{KG}(par)$ から出力される全ての鍵対 (pk_1, sk_1) 及び (pk_2, sk_2) について、$\mathtt{Share}(par, pk_1, sk_2) = \mathtt{Share}(par, pk_2, sk_1)$ が成り立つことが要求される。

全ての PPTA $\mathcal{A}$ に対し

$$\Pr[b \leftarrow \{0,1\}; par \leftarrow \mathtt{Setup}(1^k);$$
$$b' \leftarrow \mathcal{A}^{\mathcal{O}}(par) : b = b'] \leq \tfrac{1}{2} + \mathsf{neg}(k)$$

ならば、NIKE 方式 N は能動的攻撃者に対して安全であるという。ただし、ここでオラクル $\mathcal{O}$ は 3 種のクエリ $\{\mathtt{Add\text{-}Honest\text{-}User}, \mathtt{Honest\text{-}Key}, \mathtt{Corrupt\text{-}Key}\}$ を受け付ける。それぞれのクエリは以下の様に応答される:

- $\mathtt{Add\text{-}Honest\text{-}User}$: 問い合わせを受けると、$(pk, sk) \leftarrow \mathtt{KG}(par)$ を実行し、(pk, sk) を "誠実なユーザ鍵リスト" L_H(試行開始時は空) に追加し、pk を返す。
- $\mathtt{Honest\text{-}Key}$: (pk_1, pk_2) を受け取ると、$pk_1 \in L_H$ かつ $pk_2 \in L_H$ のときに限り以下を実行する: $b = 0$ のとき、$\mathcal{O}$ は L_H から pk_2 に対応する秘密鍵 sk_2 を見つけ $\mathtt{Share}(par, pk_1, sk_2)$ を返す。$b = 1$ の場合は共有鍵空間 $\mathcal{K}$ から一様ランダムに K を選び、それを返す。$b = 1$ の場合は自明な攻撃を防ぐため、同じペア (pk_1, pk_2) には常に同じ値 K を返すことにする。すなわち、(pk_1, pk_2) と (pk_2, pk_1)、及びそれらを 2 回以上問い合わせても同じ値が返される。
- $\mathtt{Corrupt\text{-}Key}$: (pk_1, pk_2) を受け取ると、$pk_1 \notin L_H$ かつ $pk_2 \in L_H$ のときに限り、$\mathcal{O}$ は pk_2 に対応する秘密鍵 sk_2 を L_H から見つけ、$\mathtt{Share}(par, pk_1, sk_2)$ を返す。

本稿ではさらに "タグベース" 非対話型鍵交換 (TNIKE) という NIKE を単純に拡張した要素技術を導入する。TNIKE と NIKE の違いは TBE と PKE の違いの様にタグを考えるかどうかである。すなわち TNIKE では、NIKE での $\mathtt{Share}$ アルゴリズムに任意の文字列タグ tag を入力できるようにする (このアルゴリズムを $\mathtt{TShare}$ と書く)。TNIKE の正当性として、全ての $par \leftarrow \mathtt{Setup}(1^k)$、$\mathtt{KG}(par)$ から出力された全ての鍵対 (pk_1, sk_1) と (pk_2, sk_2)、及び全ての tag について、$\mathtt{TShare}(par, pk_1, sk_2, \mathsf{tag}) = \mathtt{TShare}(par, pk_2, sk_1, \mathsf{tag})$ を要求する。

TNIKE の能動的攻撃者に対する安全性は、$\mathcal{O}$ が Honest-Key クエリと Corrupt-Key クエリで (pk_1, pk_2) の他に tag を受け取るという変更以外は NIKE の場合と同様に定義する。

署名　署名方式は以下の 4 つのアルゴリズムからなる: Setup は 1^k を受け取り、公開パラメータ par を出力する; KG は par を受け取り、公開鍵/秘密鍵対 (pk, sk) を出力する; Sign は par、sk、及びメッセージ m を受け取り、署名 σ を出力する; SVer は par、pk、m、及び σ を受け取り、σ が m の pk の下の正しい署名である場合 $\top$ を、そうでなければ $\bot$ を出力する。

署名方式の強偽造不可能性 (sUF-CMA 安全性)[2] 及び (弱) 偽造不可能性 (wUF-CMA 安全性)[10] の定義は省略する。

par, pk, m が与えられたとき、$\mathtt{SVer}(par, pk, m, \sigma) = \top$ を満たす σ が一つしか存在しない場合、署名方式 S は "1 対 1" であるという (例えば BLS 署名 [6] がこの性質を持つ)。

共通鍵暗号　共通鍵暗号 (SKE) は以下の 2 つのアルゴリズムからなる: SEnc は共通鍵 K(鍵空間 $\mathcal{K}$) と平文 m を受け取り、暗号文 c を出力する; SDec は $K \in \mathcal{K}$ と c を受け取り、m または $\bot$ を返す。

SKE の IND-CCA 安全性、IND-CPA 安全性、及び INT-CTXT 安全性 [4] の定義は省略する。

メッセージ認証コード　メッセージ認証コード (MAC) は以下の 2 つのアルゴリズムからなる: Mac は共通鍵 K(鍵空間 $\mathcal{K}$) 及びメッセージ m を受け取り、MAC タグ σ を出力する。MVer は $K \in \mathcal{K}$、m、及び σ を受け取り、σ が鍵 K の下 m の正しい MAC タグならば $\top$ を、そうでなければ $\bot$ を出力する。

MAC の sUF-CMA 安全性の定義は省略する。

K と m が与えられたとき、$\mathtt{MVer}(K, m, \sigma) = \top$ となる σ が一つしか存在しない場合、MAC M は "1 対 1" であるという (CMAC など既存の MAC の多くがこの性質を持つ)。

3　Signcryption の定義

Signcryption は以下の 5 つのアルゴリズムからなる: Setup は 1^k を受け取り、公開パラメータ par を出力する。$\mathtt{KG}_S$ は par を受け取り、送信者用の公開鍵/秘密鍵対 (pk_S, sk_S) を出力する。$\mathtt{KG}_R$ は par を受け取り、受信者用の公開鍵/秘密鍵対 (pk_R, sk_R) を出力する。SC は par、sk_S、pk_R、及び平文 m を受け取り、暗号文 c を出力する。USC は par、pk_S、sk_R、及び c を受け取り、m または $\bot$ を出力する.

全ての $par \leftarrow \mathtt{Setup}(1^k)$、全ての $(pk_S, sk_S) \leftarrow \mathtt{KG}_S(par)$、全ての $(pk_R, sk_R) \leftarrow \mathtt{KG}_R(par)$、及び全ての平文 m について、$c \leftarrow \mathtt{SC}(par, sk_S, pk_R, m)$ ならば $m = \mathtt{USC}(par, pk_S, sk_R, c)$ が成り立つことが要求される。

秘匿性　本研究では内部攻撃者 $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ の多人数環境での秘匿性を図 1 左上の試行を用いて定義する ([14] などで使用されたモデル)。ただし、$\mathcal{A}_2$ が USC オラクルに (pk_S^*, c^*) の組を問い合わせることは許されない。この定義では攻撃者は、あらゆる送信者用鍵を自由に作成し、それを用いて USC オラクルにアクセス、あるいは攻撃対象送信者鍵として指定できる。本稿ではこのモデルを動的多人数モデル (dyhamic multi-user model) と呼び、このモデルでの安全性を内部者の選択暗号文攻撃に対する識別不可能性 (dM-IND-iCCA 安全性) と呼ぶことにする[2]。

また、外部攻撃者 $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ に対する多人数環境での秘匿性を図 1 右上の試行を用いて定義する ([3] などで使用されたモデル)。攻撃対象の送信者の秘密鍵 sk_S^* を自由に選べない代わり、SC オラクルへの問い合わせが出来るようになる。攻撃対象 (の鍵) は攻撃者に選択権限が無いため、本稿ではこのモデルを攻撃対象鍵固定多人数モデル (fixed challenge key multi-user model) と呼び、このモデルでの安全性を外部者の選択暗号文攻撃に対する識別不能性 (fM-IND-oCCA 安全性) と呼ぶことにする[3]。

定義 1　全ての *PPTA* $\mathcal{A}$ に対し、$\Pr[\mathrm{Exp}^{\mathtt{X\text{-}IND\text{-}Y}}_{SC,\mathcal{A}}(k) = 1] \leq \frac{1}{2} + \mathtt{neg}(k)$ ならば、*Signcryption* 方式 SC は X-IND-Y 安全であるという。ただし、$(\mathtt{X}, \mathtt{Y}) \in \{(\mathtt{dM}, \mathtt{iCCA}), (\mathtt{fM}, \mathtt{oCCA})\}$ である。

偽造不可能性　秘匿性の場合とほぼ同様に、動的多人数モデルにおける内部攻撃者 $\mathcal{A}$ の偽造不可能性 (dM-wUF-iCMA 安全性)、及び攻撃対象鍵固定多人数モデルにおける外部攻撃者 $\mathcal{A}$ の偽造不可能性 (fM-wUF-iCMA 安全性) を図 1 左下及び図 1 右下の試行を用いてそれぞれ定義する。ただし、偽造不可能性の試行において、SC オラクルに問い合わせたクエリ (pk_R, m) は全てリスト L に記録され、$\mathcal{A}$ がそれぞれの試行で勝利するためには、$\mathcal{A}$ が最終的に出力した暗号文 c^* を復号した結果 m^* が、$(pk_R^*, m^*) \notin L$ を満たさねばならない。

定義 2　全ての *PPTA* $\mathcal{A}$ に対し、$\Pr[\mathrm{Exp}^{\mathtt{X\text{-}wUF\text{-}Y}}_{SC,\mathcal{A}}(k) = 1] \leq \mathtt{neg}(k)$ ならば、*Signcryption* 方式 SC は X-wUF-Y 安全であるという。ただし、$(\mathtt{X}, \mathtt{Y}) \in \{(\mathtt{dM}, \mathtt{iCMA}), (\mathtt{fM}, \mathtt{oCMA})\}$ である。

また、強偽造不可能性 (sUF) を考える場合には、$\mathcal{A}$ に対し全く同様の試行を行うが、リスト L には SC オラクルへの問い合わせと返答の組 (pk_R, m, c) が記録され、$\mathcal{A}$ が強偽造不可能性を破るためには、$\mathcal{A}$ が最終的に出力した暗号文 c^* とそれを復号した結果 m^* が、$(pk_R^*, m^*, c^*) \notin L$ を満たさねばならない。dM-sUF-iCMA 安全性、fM-sUF-oCMA 安全性も定義 2 と同様に定義する。

[2] 一つの受信者鍵 pk_R^* ではなく複数の受信者の鍵 $\{(pk_{Ri}^*)\}_{1 \leq i \leq \mathsf{poly}(k)}$ を考え、$\mathcal{A}_1$ に攻撃対象受信者を指定させるような定義も可能だが、一人の受信者の定義からのほぼ自明な帰着が可能であるため本研究では考えない。

[3] 攻撃対象鍵固定多人数モデルでも内部者安全性 (fM-IND-iCCA) を考えることができる ([2] など)。本稿のフルバージョンで、この様な安全性を持つ方式についても議論する。複数の受信者鍵や複数の送信者鍵を考えない理由は動的モデルの場合と同様。

$\mathbf{Exp}^{\texttt{dM-IND-iCCA}}_{SC,\mathcal{A}}(k)$: $par \leftarrow \texttt{Setup}(1^k)$
$(pk_R^*, sk_R^*) \leftarrow \texttt{KG}_R(par)$
$(pk_S^*, sk_S^*, m_0, m_1, st) \leftarrow \mathcal{A}_1^{\texttt{USC}(par,\cdot,sk_R^*,\cdot)}(par, pk_R^*)$
$b \leftarrow \{0,1\}$; $c^* \leftarrow \texttt{SC}(par, sk_S^*, pk_R^*, m_b)$
$b' \leftarrow \mathcal{A}_2^{\texttt{USC}(par,\cdot,sk_R^*,\cdot)}(st, c^*)$
If $b = b'$ then return 1 Else return 0

$\mathbf{Exp}^{\texttt{fM-IND-oCCA}}_{SC,\mathcal{A}}(k)$: $par \leftarrow \texttt{Setup}(1^k)$
$(pk_S^*, sk_S^*) \leftarrow \texttt{KG}_S(par)$; $(pk_R^*, sk_R^*) \leftarrow \texttt{KG}_R(par)$
$(m_0, m_1, st) \leftarrow \mathcal{A}_1^{\texttt{SC}(par,sk_S^*,\cdot,\cdot),\texttt{USC}(par,\cdot,sk_R^*,\cdot)}(par, pk_S^*, pk_R^*,)$
$b \leftarrow \{0,1\}$; $c^* \leftarrow \texttt{SC}(par, sk_S^*, pk_R^*, m_b)$
$b' \leftarrow \mathcal{A}_2^{\texttt{SC}(par,sk_S^*,\cdot,\cdot),\texttt{USC}(par,\cdot,sk_R^*,\cdot)}(st, c^*)$
If $b = b'$ then return 1 Else return 0

$\mathbf{Exp}^{\texttt{dM-wUF-iCMA}}_{SC,\mathcal{A}}(k)$: $L \leftarrow \emptyset$; $par \leftarrow \texttt{Setup}(1^k)$
$(pk_S^*, sk_S^*) \leftarrow \texttt{KG}_S(par)$
$(pk_R^*, sk_R^*, c^*) \leftarrow \mathcal{A}^{\texttt{SC}(par,sk_S^*,\cdot,\cdot)}(par, pk_S^*)$
$m^* \leftarrow \texttt{USC}(par, pk_S^*, sk_R^*, c^*)$
If $m^* \neq \perp \wedge (pk_R^*, m^*) \notin L$ then return 1 Else return 0

$\mathbf{Exp}^{\texttt{fM-wUF-oCMA}}_{SC,\mathcal{A}}(k)$: $L \leftarrow \emptyset$; $par \leftarrow \texttt{Setup}(1^k)$
$(pk_S^*, sk_S^*) \leftarrow \texttt{KG}_S(par)$; $(pk_R^*, sk_R^*) \leftarrow \texttt{KG}_R(par)$
$c^* \leftarrow \mathcal{A}^{\texttt{SC}(par,sk_S^*,\cdot,\cdot),\texttt{USC}(par,\cdot,sk_R^*,\cdot)}(par, pk_S^*, pk_R^*)$
$m^* \leftarrow \texttt{USC}(par, pk_S^*, sk_R^*, c^*)$
If $m^* \neq \perp \wedge (pk_R^*, m^*) \notin L$ then return 1 Else return 0

図 1: 内部者、外部者それぞれに対する秘匿性と偽造不可能性を定義するための試行

4 共通鍵系の要素技術を用いた一般的構成

本節では、秘匿性か偽造不可能性のどちらか (あるいは両方) について外部者安全性のみが求められる場合、共通鍵系の要素技術である SKE 及び MAC を用いれば効率良くしかも一般的に安全な Signcryption を達成できることを示す。(T)NIKE を用いることで、共通鍵系の要素技術のための鍵を共有するために何らかの通信を必要としないため、暗号文の無駄なサイズ増加を抑えることができる。

$N = (\texttt{Setup}_n, \texttt{KG}_n, \texttt{Share})$ を NIKE 方式、$TN = (\texttt{Setup}_{tn}, \texttt{KG}_{tn}, \texttt{TShare})$ を TNIKE 方式、$TE = (\texttt{Setup}_{te}, \texttt{KG}_{te}, \texttt{TEnc}, \texttt{TDec})$ を TBE 方式、$S = (\texttt{Setup}_{sig}, \texttt{KG}_{sig}, \texttt{Sign}, \texttt{SVer})$ を署名方式、$SE = (\texttt{SEnc}, \texttt{SDec})$ を SKE、そして $M = (\texttt{Mac}, \texttt{MVer})$ を MAC とする。このとき、Signcryption 方式 TEtK&M[4] を図 2 左、Signcryption 方式 TK&SEtS[5] を図 2 中央、Signcryption 方式 K&SE[6] を図 2 右の様に構成する。

TEtK&M 方式では偽造不可能性が MAC と NIKE の組み合わせにより達成される。送信者の公開鍵を TBE のタグとして用いることで、送信者の鍵と暗号文を強く結びつけている。この方式の秘匿性達成のためには MAC M が 1 対 1 でなければならない点に注意されたい。

定理 3 *TBE 方式 TE が* IND-tag-CCA *安全かつ MAC M が 1 対 1 ならば、*TEtK&M *は* dM-IND-iCCA *安全である。また、NIKE 方式 N が能動的攻撃者に対し安全かつ MAC M が* sUF-CMA *安全ならば* TEtK&M *は* fM-sUF-oCMA *安全である。*

TK&SEtS 方式では秘匿性を SKE と TNIKE(及び署名) の組み合わせにより達成する。“T”NIKE を使う必要があるのは、暗号文と送信者の鍵のうち署名の公開鍵を強く結びつけるためである。この方式の秘匿性達成のためには、SKE SE の安全性として IND-CPA 安全性しか必要としない。

定理 4 署名方式 S が sUF-CMA 安全、*TNIKE 方式 TN が能動的攻撃者に対し安全、かつ SKE SE が* IND-CPA *安全ならば、*TK&SEtS *は* fM-IND-oCCA *安全である。また、署名方式 S が* sUF-CMA *安全ならば、*TK&SEtS *は* dM-sUF-iCMA *安全である。*

K&SE 方式では、NIKE と SKE を直感的に組み合わせるだけの非常に簡潔な構成であり、暗号文は SKE の暗号文そのものとなる。それでも SKE が認証付き共通鍵暗号 (Authenticated Encryption) [4] の安全性を満たせば、秘匿性、偽造不可能性ともに外部者安全性を達成できる。

定理 5 *NIKE 方式 N が能動的攻撃者に対し安全かつ SKE SE が* IND-CCA *安全ならば、*K&SE *は* fM-IND-oCCA *安全である。また、NIKE 方式 N が能動的攻撃者に対し安全かつ SKE SE が* INT-CTXT *安全ならば、*K&SE *は* fM-sUF-oCMA *安全である。*

具体的な NIKE と TNIKE の構成 素数位数 p の群を $\mathbb{G}$、ハッシュ関数を H として、各エンティティ i が $(pk_i, sk_i) = (g^{x_i}, x_i) \in (\mathbb{G}, \mathbb{Z}_p)$ を持ち、エンティティ i とエンティティ j が鍵交換をする場合に $K = H(g^{x_i}, g^{x_j}, g^{x_i x_j})$(ただし g^{x_j} の方が g^{x_i} よりも辞書順で早い場合は順を入れ替える) を計算するような NIKE(“Hashed Diffie-Hellman 鍵交換” を略し、本稿では HDH と呼ぶ) は、Gap DH(GDH) 仮定の下ランダムオラクルモデルでの能動的攻撃者に対する安全性を証明可能である。また、[8] ではこの方式よりも少し計算効率が悪いが、CDH 仮定から同様の安全性を証明できる方式が示されている。さらに、上記方式のランダムオラクル H に tag を入力すれば、ランダムオラクルモデルで能動的攻撃者に対して安全な TNIKE となる (tHDH と書く)。

具体的な TBE の構成 あらゆる IND-CCA 安全な PKE から IND-tag-CCA 安全な TBE への変換法が知られている [12] が、計算コストは増加しないものの暗号文サイズが tag の分増加してしまう。

しかし、IND-CCA 安全性を証明可能な PKE(または KEM) のうち、ランダムオラクルを用いる方式や、スタンダードモデルの方式のうち構成要素として (ターゲット) 衝突困難ハッシュ関数を用いる様なものは、ハッシュに tag を入力することで暗号文サイズ、計算コスト増大実質無しで IND-tag-CCA 安全な TBE へと変換できる場合がある。この方法が適用できるかどうかは方式依存だが、既存の多くの方式が

[4] **T**ag-based-**E**ncrypt **t**hen **K**ey-exchange **and M**AC の略。
[5] **T**ag-based-**K**ey-exchange **and S**ymmetric-key-**E**ncrypt **t**hen **S**ign の略。
[6] **K**ey-exchange **and S**ymmetric-key-**E**ncrypt の略。

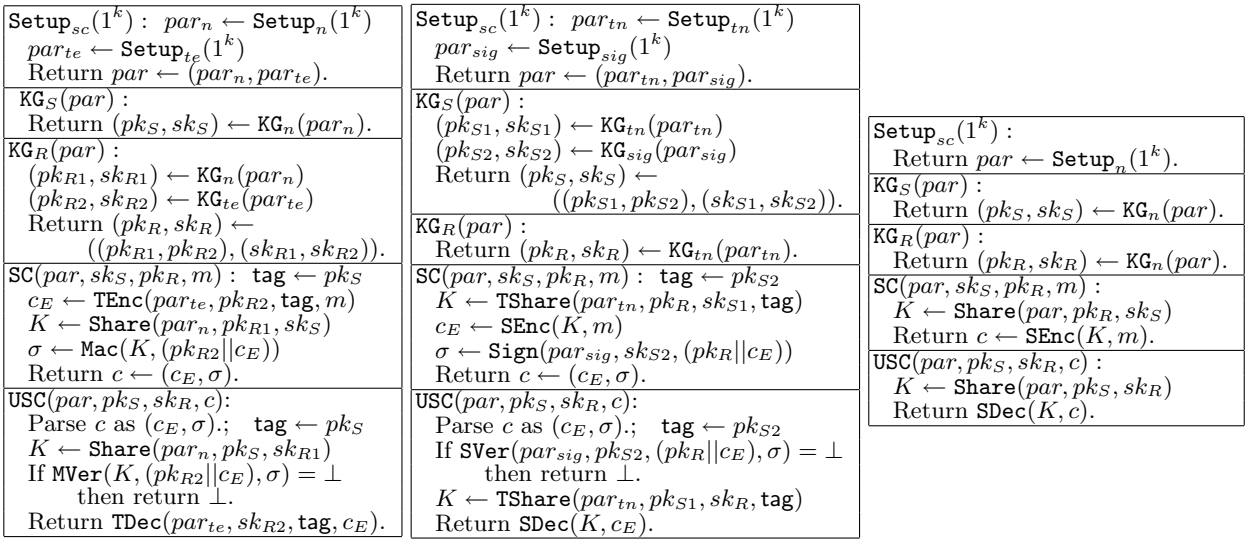

図 2: 共通鍵系要素技術を用いた Signcryption の一般的構成: TEtK&M (左)、TK&SEtS (中央)、K&SE (右).

この様な構成になっている。例としては DHIES [1], BMW [7] などがある。(これらから得られる TBE をそれぞれ tDHIES、tBMW と書くことにする。)

5 TBE と署名による内部者安全な構成

An ら [2] は PKE と署名からなる Signcryption の安全性を議論した。彼らは直感的で最も簡潔な組み合わせ法である Sign-then-Encrypt と Encrypt-then-Sign の構成について、*generalized IND-CCA* 安全性となることしか示していない。また、これらの構成法はそのままでは多人数環境で安全とは言えない。An らは送信者と受信者の鍵[7]を PKE の平文として暗号化し、さらに送信者と受信者の鍵を署名アルゴリズムに入力することで、多人数環境で安全になることを主張したが、この方法では PKE の平文として入力した鍵の分暗号文サイズが増加してしまう。

本節では、Sign-then-Encrypt 及び Encrypt-then-Sign の構成で、PKE でなく TBE を用いることで暗号文サイズ増加を回避でき、かつ多人数環境での安全性も達成できることを示す。$TE = (\mathtt{Setup}_{te}, \mathtt{KG}_{te}, \mathtt{TEnc}, \mathtt{TDec})$ を TBE 方式、及び $S = (\mathtt{Setup}_{sig}, \mathtt{KG}_{sig}, \mathtt{Sign}, \mathtt{SVer})$ を署名方式とする。このとき、*"Sign-then-Tag-based-Encrypt"* 方式 (StTE 方式) と *"Tag-based-Encrypt-then-Sign"* 方式 (TEtS 方式) をそれぞれ図 3 の様に構成する。

構成のアイデアは An らの方法とほぼ全く同様であり、送信者鍵 pk_S を平文として共に暗号化するのではなく TBE の tag として用いることで、暗号文と送信者鍵の結びつきを強固にする。また、署名アルゴリズムには受信者の鍵のみを入力すればよい。

StTE の安全性については以下が得られる。

定理 6 *TBE* 方式 TE が IND-tag-CCA 安全ならば、StTE は dM-IND-iCCA 安全である。また、署名方式 S が wUF-CMA 安全ならば、StTE は dM-wUF-iCMA 安全である。

StTE 方式では、署名方式 S として sUF-CMA 安全な物を用いても一般的には強偽造不可能性 (dM-sUF-iCMA 安全性) を達成できない。

TEtS については、An らが Encrypt-then-Sign について示した様に一般的には dM-IND-iCCA 安全性を達成できないが、署名方式が 1 対 1 という性質を満たせば可能である。また、署名方式 S が sUF-CMA 安全性を満たせば、TEtS は dM-sUF-iCMA 安全性を満たす。

定理 7 *TBE* 方式 TE が IND-tag-CCA 安全かつ署名方式 S が *1* 対 *1* ならば、TEtS は dM-IND-iCCA 安全である。また、署名方式 S が sUF-CMA 安全ならば、TEtS は dM-sUF-iCMA 安全である。

1 対 1 という性質を持つ署名方式は、現在のところ BLS 署名 [6] のみしか効率的な方式が知られていないため、TEtS については結果はそれほど一般的であるとは言い難い。しかし BLS 署名は最も効率的な署名方式の一つであり、前節で述べた tDHIES と組み合わせれば、現在知られている dM-IND-iCCA 安全かつ dM-sUF-iCMA 安全な Signcryption 方式 [14] と同程度の効率を持つ方式が得られる。

6 比較とまとめ

既存方式の代表的なものと、本研究で得られた構成法において構成要素に既存方式のうち効率の良いものを当てはめて得られる方式の比較を図 4 にまとめる。"外部者に対する秘匿性と外部者に対する偽造不可能性" の様に、同種の攻撃者に対する安全性を持つ方式と効率を比べると、(一部強い仮定が必要な場合があるが) 既存方式よりも暗号文サイズの面で効率が良いか、より強いモデルでの安全性を達成している。図 4 に示した組み合わせは本の一例で

[7] 厳密には彼らは鍵の持ち主の ID を入力するとしているが、鍵の持ち主と ID の対応は一意に特定できるモデルを考えているため、本稿ではこの様に記述することにした。

共通部分
$\mathtt{Setup}_{sc}(1^k)$: $par_{te} \leftarrow \mathtt{Setup}_{te}(1^k)$ $par_{sig} \leftarrow \mathtt{Setup}_{sig}(1^k)$ Return $par \leftarrow (par_{te}, par_{sig})$.
$\mathtt{KG}_S(par)$: Return $(pk_S, sk_S) \leftarrow \mathtt{KG}_{sig}(par_{sig})$.
$\mathtt{KG}_R(par)$: Return $(pk_R, sk_R) \leftarrow \mathtt{KG}_{te}(par_{te})$.

Sign-then-Tag-based-Encrypt StTE
$\mathtt{SC}(par, sk_S, pk_R, m)$: $\mathtt{tag} \leftarrow pk_S$ $\sigma \leftarrow \mathtt{Sign}(par_{sig}, sk_S, (pk_R\|\|m))$ $c \leftarrow \mathtt{TEnc}(par_{te}, pk_R, \mathtt{tag}, (m\|\|\sigma))$ Return c.
$\mathtt{USC}(par, pk_S, sk_R, c)$: $\mathtt{tag} \leftarrow pk_S$ $(m\|\|\sigma)/\perp \leftarrow \mathtt{TDec}(par_{te}, sk_R, \mathtt{tag}, c)$ If $\mathtt{SVer}(par_{sig}, pk_S, (pk_R\|\|m), \sigma) = \perp$ then return $\perp$. Return m.

Tag-based-Encrypt-then-Sign TEtS
$\mathtt{SC}(par, sk_S, pk_R, m)$: $\mathtt{tag} \leftarrow pk_S$ $c_E \leftarrow \mathtt{TEnc}(par_{te}, pk_R, \mathtt{tag}, m)$ $\sigma \leftarrow \mathtt{Sign}(par_{sig}, sk_S, (pk_R\|\|c_E))$ Return $c \leftarrow (c_E, \sigma)$.
$\mathtt{USC}(par, pk_S, sk_R, c)$: $\mathtt{tag} \leftarrow pk_S$ Parse c as (c_E, σ). If $\mathtt{SVer}(par_{sig}, pk_S, (pk_R\|\|c_E), \sigma) = \perp$ then return $\perp$. Return $\mathtt{TDec}(par_{te}, sk_R, \mathtt{tag}, c_E)$.

図 3: TBE と署名の組み合わせによる Signcryption (StTE と TEtS の Setup、$\mathtt{KG}_S$、及び $\mathtt{KG}_R$ は共通).

方式	要ランダムオラクル?	秘匿性/仮定	偽造不可能性/仮定	計算コスト SC / USC	暗号文サイズ $\|c\| - \|m\|$ =?	ビット
Dent [9]	Yes	IND-oCCA/ CDH	sUF-oCMA/ CDH	[2,0;0]/ [1,0;0]	$\|\mathbb{G}_e\| + \|\mathrm{MAC}\|$	240
Zheng [18, 3]	Yes	fM-IND-oCCA/ GDH	dM-sUF-iCMA/ GDL	[1,0;0]/ [1,1;0]	$2\|\mathbb{Z}_p\|$	320
GBN [11] + HMQV [13]	Yes	sM-IND-iCCA/ CDH	sM-sUF-oCMA/ CDH	[2,0;0]/ [0,1;0]	$\|\mathbb{G}_e\| + \|\mathrm{MAC}\|$	240
LQ [14]	Yes	dM-IND-iCCA/ co-CDH	dM-sUF-iCMA/ co-CDH	[3,0;0]/ [1,0;2]	$2\|\mathbb{G}_p\|$	342
Tan [17]	No	dM-IND-iCCA/ DBDH	dM-sUF-iCMA(†)/ q-CDH	[3,2;0]/ [2,1;4]	$3\|\mathbb{G}_p\| + 2\|\mathbb{Z}_p\|$	833
K&SE (HDH)	Yes	fM-IND-oCCA/ GDH	fM-sUF-oCMA/ GDH	[1,0;0]/ [1,0;0]	$\|IV\| + \|\mathrm{MAC}\|$	160
TK&SEtS (tHDH + BLS)	Yes	fM-IND-oCCA/ GDH & co-CDH	dM-sUF-iCMA/ co-CDH	[2,0;0]/ [1,0;2]	$\|IV\| + \|\mathbb{G}_p\|$	251
TEtK&M (HDH + tDHIES)	Yes	dM-IND-iCCA/ GDH	fM-sUF-oCMA/ GDH	[3,0;0]/ [2,0;0]	$\|\mathbb{G}_e\| + \|\mathrm{MAC}\|$	240
TEtS (tDHIES + BLS)	Yes	dM-IND-iCCA/ GDH	dM-sUF-iCMA/ co-CDH	[3,0;0]/ [1,0;2]	$\|\mathbb{G}_e\| + \|\mathbb{G}_p\|$	331
StTE (tBMW + BB [5])	No	dM-IND-iCCA/ DBDH	dM-wUF-iCMA/ q-SDH q-SDH	[4,0;0]/ [1,1;2]	$3\|\mathbb{G}_p\| + \|\mathbb{Z}_p\|$	673

図 4: 既存方式との比較。sM-の安全性は本稿で定義した dM-や fM-の定義よりも真に弱い多人数モデルでの安全性 (詳細はフルバージョンを参照)。計算コスト [a,b;c] は、それぞれ a:べき乗計算, b:多重べき乗計算 ($g^\alpha h^\beta$ など), c:ペアリングの回数. $\mathbb{G}_p$、$\mathbb{G}_e$、$\mathbb{Z}_p$, MAC, IV はそれぞれ、ペアリングを持つ双線形群、ペアリング無しの群、それら群の位数 p を法とする整数、MAC タグ、SKE の初期ベクタを表す。暗号文サイズは 80 ビット安全性を達成するための理論的値であり $|\mathbb{G}_p| = 171$、$|\mathbb{G}_e| = |\mathbb{Z}_p| = 160$、$|MAC| = |IV| = 80$ とした。† この論文で定義した dM-sUF-iCMA 安全性よりもやや弱い定義であり、dM-wUF-iCMA 安全性とも直接強弱を比べられない。

あり、構成要素の組み合わせ方により様々な具体的な Signcryption の構成の可能性があり得る。

参考文献

[1] M. Abdalla, M. Bellare, and P. Rogaway. The oracle diffie-hellman assumptions and an analysis of dhies. CT-RSA 2001, LNCS 2020, pp. 143–158, 2001.

[2] J.H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. EUROCRYPT 2002, LNCS 2332, pp. 83–107, 2002.

[3] J. Baek, R. Steinfeld, and Y. Zheng. Formal proofs for the security of signcryption. *J. Cryptology*, 20(2):203–235, 2007.

[4] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. ASIACRYPT 2000, LNCS 1976, pp. 531–545, 2000.

[5] D. Boneh and X. Boyen. Short signatures without random oracles. EUROCRYPT 2004, LNCS 3027, pp. 56–73, 2004.

[6] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. *J. Cryptology*, 17(4):297–319, 2004.

[7] X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques. ACMCCS 2005, pp. 320–329, 2005.

[8] D. Cash, E. Kiltz, and V. Shoup. The twin diffie-hellman problem and applications. EUROCRYPT 2008, LNCS 4965, pp. 127–145, 2008.

[9] A.W. Dent. Hybrid signcryption schemes with outsider security. ISC 2005, LNCS 3650, pp. 203–217, 2005

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

[11] M.C. Gorantla, C. Boyd, and J.M.G. Nieto. On the connection between signcryption and one-pass key establishment. Cryptography and Coding, LNCS 4887, pp. 277–301, 2007.

[12] E. Kiltz. Chosen-ciphertext security from tag-based encryption. TCC 2006,, LNCS 3876, pp. 581–600, 2006.

[13] H. Krawczyk. Hmqv: A high-performance secure diffie-hellman protocol. CRYPTO 2005, LNCS 3621, pp. 546–566, 2005.

[14] B. Libert and J.-J. Quisquater. Efficient signcryption with key privacy from gap diffie-hellman groups. Updated version of the paper by the same authors in PKC 2004, LNCS 2947, pp. 187–200, 2004. Available at http://www.dice.ucl.ac.be/~libert/.

[15] P.D. MacKenzie, M.K. Reiter, and K. Yang. Alternatives to non-malleability: Definitions, constructions, and applications (extended abstract). TCC 2004, LNCS 2951, pp. 171–190, 2004.

[16] T. Matsuda, K. Matsuura, and J.C.N. Schuldt. Efficient constructions of signcryption schemes and signcryption composability. Unpublished manuscript. Available from authors.

[17] C.H. Tan. Signcryption scheme in multi-user setting without random oracles. IWSEC 2008, LNCS 5312, pp. 64–82, 2008.

[18] Y. Zheng. Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption). CRYPTO 1997, LNCS 1294, pp. 165–179, 1997.