情報処理学会論文誌 Vol.55 No.9 1964–1970 (Sep. 2014)

時限式IDベース暗号

押切 徹[1,a)] 齊藤 泰一[2]

受付日 2013年11月29日, 採録日 2014年6月17日

概要：時限式暗号（Timed-Release Encryption, TRE）とは，復号できる時刻を暗号化の際に指定できる暗号方式である．本稿ではIDベース暗号（Identity-Based Encryption, IBE）にTREの機能を持たせた時限式IDベース暗号（Timed-Release Identity-Based Encryption, TRIBE）を提案し，その安全性定義を行う．さらにIBEとワンタイム署名からなるTRIBEの一般的構成法（generic construction）を示し，定義した安全性を満たすことを証明をする．

キーワード：時限式暗号，IDベース暗号，ワンタイム署名

Timed-release Identity-based Encryption

Toru Oshikiri[1,a)] Taiichi Saito[2]

Received: November 29, 2013, Accepted: June 17, 2014

Abstract: Timed-Release Encryption (TRE) is an encryption mechanism that allows a receiver to decrypt a ciphertext only after the time that a sender designates. We propose a notion of identity-based encryption scheme with TRE encryption mechanism, timed-release identity-based encryption (TRIBE), and define its security models. Moreover we show a generic construction of TRIBE from IBE and one-time signature, and prove that the constructed scheme achieves the security we defined.

Keywords: timed-release encryption, identity-based encryption, one-time signature

1. はじめに

時限式暗号（Timed-Release Encryption, TRE）[8]は暗号文を復号できる時刻を暗号化の際に指定できる暗号方式である．

時限式公開鍵暗号（Timed-Release Public-Key Encryption, TRPKE）[2]は公開鍵暗号（Public-Key Encryption, PKE）にTREの機能を持たせた方式である．TRPKEは正当な秘密鍵を持っていても送信者が指定した時刻までは復号することができず，その時刻になると復号することができる．

TRPKEは，公開鍵と指定時刻を用いてメッセージを暗号化する送信者，時刻に対応する時刻鍵を生成する時刻サーバ，秘密鍵と時刻鍵を用いて暗号文を復号する受信者から構成される．

時限式IDベース暗号（Timed-Release Identity-Base Encryption, TRIBE）はIDベース暗号（Identity-Base Encryption, IBE）にTREの機能を持たせた方式である．TRIBEでも，正当な秘密鍵を持っていても送信者が指定した時刻までは復号することができず，その時刻になると復号することができる．

TRIBEは，受信者のIDと指定時刻を用いてメッセージを暗号化する送信者，IDに対応する秘密鍵を生成する鍵生成センタ（Key Generation Center, KGC），時刻に対応する時刻鍵を生成しブロードキャストする時刻サーバ（Time-Server, TS），秘密鍵と時刻鍵を用いて暗号文を復号する受信者から構成される．TRIBEは，公開鍵として受信者の識別子ID（メールアドレスなど）を用いるため，公開鍵と受信者の対応付け，暗号化の際の公開鍵の事前入

[1] 東京電機大学大学院工学研究科情報通信工学専攻
Graduate School of Engineering, Tokyo Denki University, Adachi, Tokyo 120–8551, Japan
[2] 東京電機大学工学部情報通信工学科
Department of Information and Communication Engineering, Tokyo Denki University, Adachi, Tokyo 120–8551, Japan
[a)] 13kmc06@ms.dendai.ac.jp

手が不要であるという利点がある．

これより，TRPKE では公開鍵とユーザの対応を PKI などの仕組みで保証する必要があるが，TRIBE ではそのような仕組みを必要としないため，導入コストを下げられる可能性がある．

2. 封印入札オークションへの応用

TRPKE を利用したアプリケーションとして封印入札オークション（Sealed-Bid Auction）があげられている [6]．封印入札オークションとは入札者が相互に入札額を知ることができない方式である．この封印入札オークションに TRPKE を利用する場合，入札者は出品者の公開鍵と入札終了時刻を用いて入札額を暗号化し出品者に送る．出品者は入札終了時刻に時刻サーバから発行される時刻鍵を用いて復号し落札者を決定することができる．

たとえば，出品者が何らかの理由で最高額入札者でない入札者を落札者に決定するという不正が発生したとする．この場合，最高額入札者は出品者にクレームを入れるが TRPKE を利用したオークションの場合，出品者の秘密鍵なしでは入札履歴を検証することができない．あるいは，入札者は，落札時まで，暗号化でも用いた乱数などの全データを記録しておく必要がる．

一方，TRIBE を利用したオークションの場合はこの不正を検出することができる．封印入札オークションに TRIBE を利用する場合，入札者は出品者の ID と入札終了時刻を用いて入札額を暗号化し出品者に送る．出品者は，KGC によって生成された秘密鍵と，入札終了時刻に時刻サーバから発行される時刻鍵を用いて復号し落札者を決定する．TRPKE を利用した場合と違い，TRIBE を利用したオークションでは上記のような不正が発生した場合，KGC はすべての出品者の秘密鍵を導けるため入札履歴を検証し，出品者の不正を指摘することが可能である．

このように出品者と入札者の間でトラブルが発生した場合，KGC がすべてのユーザの秘密鍵を導出できることから入札を復号することにより第三者としてオークションの正常運営をサポートすることができる．一方，KGC は，指定時刻以降には任意の暗号文を復号できてしまうため，信頼できる機関でなければならない．

上記の TRIBE を利用したオークションにおいては KGC が出品者の不正を指摘できたが，TRPKE を利用したオークションにおいては，信頼できる第三者がすべてのユーザの公開鍵と秘密鍵の生成を行うことにより出品者の不正を指摘できるようになる．しかし，この場合，公開鍵の正当性を保証する機関がさらに必要である．一方，TRIBE を利用したオークションにおいてはそのような機関を必要としないため，TRIBE の方が有用であると考えられる．

またオークションのシステムにおいては，出品者と落札者はそれぞれ ID によって情報が管理されている場合が多い．TRPKE を利用したオークションでは出品者の公開鍵を入手する必要があるが，TRIBE を利用した方式ではその ID を公開鍵として利用できるという利点がある．

3. 関連研究

3.1 TRPKE

TRPKE は秘密鍵と時刻鍵の 2 つの鍵が揃ったとき復号できる方式である．そのため，時刻鍵を持たない受信者や，時刻鍵のみを持つ TS は復号できてはならない．TRPKE の安全性については，Cheon ら [3], [4] は悪意のある受信者に対する安全性として IND-RTR-CCA 安全性を，悪意のある TS に対する安全性として IND-CCA-TS 安全性を定義し，それらの安全性を満たす一般的構成法を示した．その構成は Dodis ら [5] の多重暗号 “Parallel Encryption” に基づき，PKE，IBE および One-Time 署名を組み合わせたもので，その安全性はスタンダードモデルで証明されている．Cathalo ら [1] は IND-RTR-CCA 安全性よりも強い安全性である IND-CTCA 安全性を定義した．Fujioka ら [6] はこの安全性を満たす一般的構成法を示した．この構成法は “Sequential Multiple Encryption” に基づいており，PKE と IBE を組み合わせたもので，その安全性はランダムオラクルモデルで証明されている．

TRPKE を拡張した方式として Pre-Open 機能付き TRE（Timed-Release Encryption with Pre-Open Capability: TRE-PC）がある．これは Hwang ら [7] によって提案された方式で，送信者が Pre-Open Key と呼ばれる秘密情報を受信者に送ることで，指定時刻前であっても TS からの時刻鍵なしで復号することを可能としている．この TRE-PC について，Nakai ら [9] はスタンダードモデルで証明可能な安全性を持つ方式の一般的構成法を示している．

4. 貢献

本稿では TRIBE の安全性として，悪意のある TS に対する安全性である IND-ID-CCA$_{\mathsf{TS}}$ 安全性と，悪意のある受信者に対する安全性である IND-ID-CCA$_{\mathsf{CR}}$ 安全性を定義する．さらに，これらの安全性を満たす一般的構成法（generic construction）を示す．この構成法は Dodis ら [5] の Multiple Encryption の “Parallel Encryption” に基づいて IBE および One-Time 署名を組み合わせたものであり，その安全性はスタンダードモデルで証明可能である．構成要素である IBE が IND-ID-CCA 安全，One-Time 署名が OT-sEUF-CMA 安全ならば，構成された方式は IND-ID-CCA$_{\mathsf{CR}}$ 安全性および IND-ID-CCA$_{\mathsf{TS}}$ 安全性を満たす．

5. 準備

本章では提案方式の構成に必要な ID ベース暗号，One-Time 署名とその安全性について説明する．

5.1 ID ベース暗号

ID ベース暗号 Π は，以下のアルゴリズム（IBE.Setup，IBE.Ext，IBE.Enc，IBE.Dec）で構成される．

IBE.Setup(1^k)： セキュリティパラメータ 1^k を入力とし，公開パラメータ $params$ とマスタ秘密鍵 msk を出力する．

IBE.Ext($params, msk, id$)： 公開パラメータ $params$，マスタ秘密鍵 msk，ユーザ ID id を入力とし，id に対応するユーザ秘密鍵 d_{id} を出力する．

IBE.Enc($params$, id, m)： 公開パラメータ $params$，ユーザ ID id，メッセージ m を入力とし，暗号文 c を出力する．

IBE.Dec($params$, d_{id}, c)： 公開パラメータ $params$，ユーザ秘密鍵 d_{id}，暗号文 c を入力とし，メッセージ m もしくは $\perp$ を出力する．

ID ベース暗号では，任意の IBE.Ext 出力 d_{id} = IBE.Ext($params$, msk, id) に対して，任意の m に対して，c = IBE.Enc($params$, id, m) ならば m = IBE.Dec($params$, d_{id}, c) が成立する．

5.1.1 IND-ID-CCA 安全性

ID ベース暗号 Π の選択暗号文攻撃および適応的 ID 攻撃に対する識別不可能性（Indistinguishability against adaptive identity and chosen-ciphertext attacks, IND-ID-CCA）は，以下のチャレンジャ $\mathcal{C}$ と攻撃者 $\mathcal{A}$ のゲームを用いて定義される．

Setup $\mathcal{C}$ は ($params$, msk) $\leftarrow$ IBE.Setup(1^k) を実行し，$\mathcal{A}$ に $params$ を与え，msk を保持しておく．

Phase1 $\mathcal{A}$ は Extract クエリ id と Decrypt クエリ (id, c) を $\mathcal{C}$ に送ることができる．Extract クエリ id に対し，$\mathcal{C}$ はユーザ秘密鍵 d_{id} = IBE.Ext($params$, msk, id) を実行し，d_{id} を $\mathcal{A}$ に返す．Decrypt クエリ (id, c) に対し，$\mathcal{C}$ は d_{id} = IBE.Ext($params$, msk, id) を実行した後，IBE.Dec(d_{id}, c) を実行し，その出力の m または $\perp$ を $\mathcal{A}$ に返す．

Challenge $\mathcal{A}$ は同じ長さのメッセージ m_0，m_1 と Extract クエリとして送っていない id^* を $\mathcal{C}$ に送る．$\mathcal{C}$ は $b \in \{0,1\}$ をランダムに選び，c^* = IBE.Enc($params$, id^*, m_b) を実行し，c^* を $\mathcal{A}$ に返す．

Phase2 $\mathcal{A}$ は **Phase1** と同様に，Extract クエリと Decrypt クエリを送ることができる．ただし，Extract クエリとしてチャレンジ ID である id^* を，Decrypt クエリとして **Challenge** で現れた (id^*, c^*) を出すことはできない．

Guess $\mathcal{A}$ は推測値 $\tilde{b}$ を出力する．

もし，$b = \tilde{b}$ ならば $\mathcal{A}$ の勝ちとする．

$\mathcal{A}$ のアドバンテージを以下のように定義する．

$$Adv_{\Pi,\mathcal{A}}^{\text{IND-ID-CCA}}(1^k) = |\Pr[b = \tilde{b}] - \frac{1}{2}|$$

定義 1 ID ベース暗号 Π に対する多項式時間攻撃者 $\mathcal{A}$ のアドバンテージ $Adv_{\Pi,\mathcal{A}}^{\text{IND-ID-CCA}}(1^k)$ が negligible であるとき，IBE は IND-ID-CCA 安全であるという．

5.2 One-Time 署名

One-Time 署名 Σ は，以下のアルゴリズム（SigGen, Sign, Verify）で構成される．

SigGen(1^k)： セキュリティパラメータ 1^k を入力とし，検証鍵 vk と署名鍵 sk を出力する．

Sign(sk, m)： 署名鍵 sk，メッセージ m を入力とし，署名 σ を出力する．

Verify(vk, m, σ)： 検証鍵 vk，メッセージ m，署名 σ を入力とし，`accept`/`reject` を出力する．

One-Time 署名では，任意の SigGen 出力 (vk, sk) = SigGen(1^k) に対して，任意の m に対して，σ = Sign(sk, m) ならば `accept` = Verify(vk, m, σ) が成立する．

5.2.1 OT-sEUF-CMA 安全性

One-Time 署名 Σ の選択文書攻撃に対する強存在的偽造困難性（One-time strong existential unforgeability against chosen message attacks, OT-sEUF-CMA）は，以下のチャレンジャ $\mathcal{C}$ と偽造者 $\mathcal{F}$ のゲームを用いて定義される．

Setup $\mathcal{C}$ は (vk, sk) $\leftarrow$ SigGen(1^k) を実行し，$\mathcal{F}$ に vk を与え，sk を保持しておく．

Query $\mathcal{F}$ は $\mathcal{C}$ に対して 1 回だけメッセージ m に対する署名クエリを出すことができる．$\mathcal{C}$ は署名 σ = Sign(sk, m) を実行し，σ を $\mathcal{A}$ に返す．

Forge $\mathcal{F}$ はメッセージと署名の組 (m^*, σ^*) を出力する．$\mathcal{F}$ のアドバンテージを以下のように定義する．

$$\begin{aligned} &Adv_{\Sigma,\mathcal{F}}^{\text{OT-sEUF-CMA}}(1^k) \\ &\quad = \Pr[\text{Verify}(\ vk\ , m^*, \sigma^*) = \texttt{accept} \\ &\qquad\qquad \wedge (m, \sigma) \neq (m^*, \sigma^*)] \end{aligned}$$

定義 2 One-Time 署名 Σ に対する多項式時間攻撃者 $\mathcal{F}$ のアドバンテージ $Adv_{\Sigma,\mathcal{F}}^{\text{OT-sEUF-CMA}}(1^k)$ が negligible であるとき，One-Time 署名は OT-sEUF-CMA 安全であるという．

6. 提案

本章では時限式 ID ベース暗号とその安全性定義を記述する．

6.1 時限式 ID ベース暗号（TRIBE）

時限式 ID ベース暗号 Γ は，以下のアルゴリズム（TS.Setup, KGC.Setup, Release, Extract, Encrypt, Decrypt）で構成される．

TS.Setup(1^k)：セキュリティパラメータ 1^k を入力とし，

タイムサーバの公開鍵 tpk と秘密鍵 tsk を出力する．

KGC.Setup(1^k)：セキュリティパラメータ 1^k を入力とし，公開パラメータ $params$ とマスタ秘密鍵 msk を出力する．

Release(tpk, tsk, t)：タイムサーバの公開鍵 tpk と秘密鍵 tsk，時刻 t を入力とし，t に対応する時刻鍵 d_t を出力する．

Extract($params$, msk, id)：公開パラメータ $params$，マスタ秘密鍵 msk，ユーザ ID id を入力とし，id に対応するユーザ秘密鍵 d_{id} を出力する．

Encrypt(tpk, $params$, t, id, m)：タイムサーバの公開鍵 tpk，公開パラメータ $params$，指定時刻 t，ユーザ ID id，メッセージ m を入力とし，暗号文 c を出力する．

Decrypt(tpk, $params$, d_t, d_{id}, c)：タイムサーバの公開鍵 tpk，公開パラメータ $params$，時刻鍵 d_t，ユーザ秘密鍵 d_{id}，暗号文 c を入力とし，メッセージ m もしくは $\perp$ を出力する．

TRIBE では，任意の Release 出力 $d_t = \mathsf{Release}(tpk, tsk, t)$，任意の Extract 出力 $d_{id} = \mathsf{Extract}(params, msk, id)$ に対して，任意の m に対して，$c = \mathsf{Encrypt}(tpk, params, t, id, m)$ ならば $m = \mathsf{Decrypt}(tpk, params, d_t, d_{id}, c)$ が成立する．

6.2 TRIBE の安全性

TRIBE は時刻鍵とユーザ秘密鍵という 2 つの鍵が揃ったときにのみ復号できる暗号方式である．そのため，以下の 3 つのエンティティに対する安全性を考慮する必要がある．

- TS に対する安全性
  任意の時刻鍵を利用できても，ユーザ秘密鍵なしでは暗号文からメッセージの情報を得ることができない．
- 受信者に対する安全性
  ユーザ秘密鍵を持っていても，時刻鍵なしでは暗号文からメッセージの情報を得ることができない．
- 外部者に対する安全性
  TS からブロードキャストされる時刻鍵だけでは暗号文からメッセージの情報を得ることができない．

外部者が持つ情報はすべて TS も得ることができるため，TS に対する安全性のみ考慮する．また受信者に対する安全性については，攻撃者にマスタ秘密鍵を与えるという強い安全性を定義する．

6.2.1 IND-ID-CCA$_{\mathsf{TS}}$ 安全性

時限式 ID ベース暗号 Γ の悪意のある TS に対する IND-ID-CCA$_{\mathsf{TS}}$ 安全性は，以下のチャレンジャ $\mathcal{C}$ と攻撃者 $\mathcal{A}$ のゲームを用いて定義される．

Setup $\mathcal{C}$ は $(tpk, tsk) \leftarrow \mathsf{TS.Setup}(1^k)$，$(params, msk) \leftarrow \mathsf{KGC.Setup}(1^k)$ を実行し，$\mathcal{A}$ に tpk，tsk，$params$ を与え，msk を保持しておく．

Phase1 $\mathcal{A}$ は Extract クエリ id と Decrypt クエリ (t, id, c) を $\mathcal{C}$ に送ることができる．$\mathcal{A}$ の Extract クエリ id に対し，$\mathcal{C}$ は $d_{id} = \mathsf{Extract}(params, msk, id)$ を実行し，d_{id} を $\mathcal{A}$ に返す．$\mathcal{A}$ の Decrypt クエリ (t, id, c) に対し，$\mathcal{C}$ は $d_t = \mathsf{Release}(tpk, tsk, t)$ と $d_{id} = \mathsf{Extract}(params, msk, id)$ を実行した後，$\mathsf{Decrypt}(tpk, params, d_t, d_{id}, c)$ を実行し，その出力 m または $\perp$ を $\mathcal{A}$ に返す．

Challenge $\mathcal{A}$ は同じ長さの任意のメッセージ m_0，m_1 と指定時刻 t^*，Extract クエリを送っていない id^* を $\mathcal{C}$ に送る．$\mathcal{C}$ は $b \in \{0,1\}$ をランダム選び，$c^* = \mathsf{Encrypt}(tpk, params, t^*, id^*, m_b)$ を実行し，c^* を $\mathcal{A}$ に返す．

Phase2 $\mathcal{A}$ は **Phase1** と同様に，Extract クエリと Decrypt クエリを送ることができる．ただし，Extract クエリとしてチャレンジ ID である id^* を，Decrypt クエリとして **Challenge** で現れた (t^*, id^*, c^*) を送ることはできない．

Guess $\mathcal{A}$ は推測値 $\tilde{b}$ を出力する．

もし，$b = \tilde{b}$ ならば $\mathcal{A}$ の勝ちとなる．$\mathcal{A}$ のアドバンテージを以下のように定義する．

$$Adv^{\text{IND-ID-CCA}_{\mathsf{TS}}}_{\Gamma,\mathcal{A}}(1^k) = |\Pr[b = \tilde{b}] - \frac{1}{2}|$$

定義 3 時限式 ID ベース暗号 Γ に対する多項式時間攻撃者 $\mathcal{A}$ のアドバンテージ $Adv^{\text{IND-ID-CCA}_{\mathsf{TS}}}_{\Gamma,\mathcal{A}}(1^k)$ が negligible であるとき，TRIBE Γ は IND-ID-CCA$_{\mathsf{TS}}$ 安全であるという．

6.2.2 IND-ID-CCA$_{\mathsf{CR}}$ 安全性

TRIBE Γ の悪意のある受信者（curious receiver）に対する IND-ID-CCA$_{\mathsf{CR}}$ 安全性は，以下のチャレンジャ $\mathcal{C}$ と攻撃者 $\mathcal{A}$ のゲームを用いて定義される．

Setup $\mathcal{C}$ は $(tpk, tsk) \leftarrow \mathsf{TS.Setup}(1^k)$，$(params, msk) \leftarrow \mathsf{KGC.Setup}(1^k)$ を実行し，$\mathcal{A}$ に tpk，$params$，msk を与え，tpk を保持しておく．

Phase1 $\mathcal{A}$ は Release クエリ t と Decrypt クエリ (t, id, c) を $\mathcal{C}$ に送ることができる．$\mathcal{A}$ の Release クエリに対し，$\mathcal{C}$ は $d_t = \mathsf{Release}(tpk, tsk, t)$ を実行し，d_t を $\mathcal{A}$ に返す．$\mathcal{A}$ の Decrypt クエリに対し，$\mathcal{C}$ は $d_t = \mathsf{Release}(tpk, tsk, t)$ と $d_{id} = \mathsf{Extract}(params, msk, id)$ を実行した後，$\mathsf{Decrypt}(tpk, params, d_t, d_{id}, c)$ を実行し，その出力 m または $\perp$ を $\mathcal{A}$ に返す．

Challenge $\mathcal{A}$ は同じ長さの任意メッセージ m_0，m_1 とユーザ ID id^*，Release クエリを送っていない t^* を $\mathcal{C}$ に送る．$\mathcal{C}$ は $b \in \{0,1\}$ をランダム選び，$c^* = \mathsf{Encrypt}(tpk, params, t^*, id^*, m_b)$ を実行し，c^* を $\mathcal{A}$ に

返す．

Phase2 $\mathcal{A}$ は **Phase1** と同様に，Release クエリと Decrypt クエリを出すことができる．ただし，Release クエリとしてチャレンジ時刻である t^* を，Decrypt クエリとして **Challenge** で現れた (t^*, id^*, c^*) を送ることはできない．

Guess $\mathcal{A}$ は推測値 $\tilde{b}$ を出力する．

もし，$b = \tilde{b}$ ならば $\mathcal{A}$ の勝ちとなる．$\mathcal{A}$ のアドバンテージを以下のように定義する．

$$Adv^{\mathsf{IND\text{-}ID\text{-}CCA_{CR}}}_{\Gamma,\mathcal{A}}(1^k) = |\Pr[b = \tilde{b}] - \frac{1}{2}|$$

定義 4 時限式 ID ベース暗号 Γ に対する多項式時間攻撃者 $\mathcal{A}$ のアドバンテージ $Adv^{\mathsf{IND\text{-}ID\text{-}CCA_{CR}}}_{\Gamma,\mathcal{A}}(1^k)$ が negligible であるとき，TRIBE Γ は $\mathsf{IND\text{-}ID\text{-}CCA_{CR}}$ 安全であるという．

7. 一般的構成法

本章では 2 つの ID ベース暗号 $\Pi = (\mathsf{IBE.Setup}, \mathsf{IBE.Ext}, \mathsf{IBE.Enc}, \mathsf{IBE.Dec})$，$\Pi' = (\mathsf{IBE'.Setup}, \mathsf{IBE'.Ext}, \mathsf{IBE'.Enc}, \mathsf{IBE'.Dec})$ と One-Time 署名 $\Sigma = (\mathsf{SigGen}, \mathsf{Sign}, \mathsf{Verify})$ を用いた時限式 ID ベース暗号 Γ の構成法を示す．

7.1 構成

$\mathsf{TS.Setup}(1^k)$:

Step 1: Run $\mathsf{IBE.Setup}(1^k)$ to generate $(params, msk)$.

Step 2: Set $tpk = params$ and $tsk = msk$.

Step 3: Return (tpk, tsk).

$\mathsf{KGC.Setup}(1^k)$:

Step 1: Run $\mathsf{IBE'.Setup}(1^k)$ to generate $(params, msk)$.

Step 2: Return $(params, msk)$.

$\mathsf{Release}(tpk, tsk, t)$:

Step 1: Run $\mathsf{IBE.Ext}(tpk, tsk, t)$ to obtain d_t.

Step 2: Return d_t.

$\mathsf{Extract}(params, msk, id)$:

Step 1: Run $\mathsf{IBE'.Ext}(params, msk, id)$ to obtain d_{id}.

Step 2: Return d_{id}.

$\mathsf{Encrypt}(tpk, params, t, id, m)$:

Step 1: Run $\mathsf{SigGen}(1^k)$ to generate (sk, vk).

Step 2: Randomly choose $s_1 \in \{0,1\}^{|m|}$.

Step 3: Compute $s_2 = m \oplus s_1$.

Step 4: Compute $c_1 = \mathsf{IBE.Enc}(tpk, s_1||vk, t)$.

Step 5: Compute $c_2 = \mathsf{IBE'.Enc}(params, s_2||vk, id)$.

Step 6: Compute $\sigma = \mathsf{Sign}(sk, c_1||c_2||t||id)$.

Step 7: Set $c = (c_1, c_2, t, id, vk, \sigma)$.

Step 8: Return c.

$\mathsf{Decrypt}(tpk, params, d_t, d_{id}, c)$:

Step 1: Parse c as $c = (c_1, c_2, t, id, vk, \sigma)$.

Step 2: If $\mathsf{Verify}(vk, c_1||c_2||t||id, \sigma) = \mathtt{reject}$ then return $\perp$ and stop.

Step 3: Compute $s_1||vk' = \mathsf{IBE.Dec}(tpk, d_t, c_1)$.

Step 4: Compute $s_2||vk'' = \mathsf{IBE.Dec}(params, d_{id}, c_2)$.

Step 5: If $vk = vk' = vk''$ then return $m = s_1 \oplus s_2$ else return $\perp$.

7.2 構成された TRIBE の安全性

IND-ID-CCA 安全な ID ベース暗号と OT-sEUF-CMA 安全な One-Time 署名で構成された TRIBE は，悪意のある TS に対する $\mathsf{IND\text{-}ID\text{-}CCA_{TS}}$ 安全性および悪意のある受信者に対する $\mathsf{IND\text{-}ID\text{-}CCA_{CR}}$ 安全性を満たす．

7.2.1 $\mathsf{IND\text{-}ID\text{-}CCA_{TS}}$ 安全性

定理 1 ID ベース暗号 Π' が IND-ID-CCA 安全，One-Time 署名 Σ が OT-sEUF-CMA 安全ならば，提案方式 Γ は $\mathsf{IND\text{-}ID\text{-}CCA_{TS}}$ 安全性を満たす．

証明 $\mathcal{A}$ を Γ の $\mathsf{IND\text{-}ID\text{-}CCA_{TS}}$ 安全性に対する攻撃者とする．そしてこの $\mathcal{A}$ を利用して，Π' の IND-ID-CCA 安全性を破る攻撃者 $\mathcal{B}$ を構成する．また暗号文 $c = (c_1, c_2, t, id, vk, \sigma)$ が $\mathsf{Verify}(vk, c_1||c_2||t||id, \sigma) = \mathtt{accept}$ を満たす場合，この暗号文を正当な暗号文と呼ぶこととし，$c^* = (c_1^*, c_2^*, t^*, id^*, vk^*, \sigma^*)$ でチャレンジ暗号文を表すとする．ここで 2 つのイベント $\mathtt{Forge}$，$\mathtt{Succ}$ を定義する．

$\mathtt{Forge}$：$\mathcal{A}$ が **Phase2** において復号オラクルに正当な暗号文 $(c_1, c_2, t, id, vk^*, \sigma)$ を問い合わせる．

$\mathtt{Succ}$：$\mathcal{B}$ が IND-ID-CCA ゲームに勝利する．

このとき，以下の 2 つの補題が成立する．

補題 1 $\Pr[\mathtt{Forge}]$ は negligible である．

補題 2 $\Pr[\mathtt{Succ}|\overline{\mathtt{Forge}}] = Adv^{\mathsf{IND\text{-}ID\text{-}CCA_{TS}}}_{\Gamma,\mathcal{A}} + \frac{1}{2}$ である．

証明［補題 1］

イベント $\mathtt{Forge}$ が起こると仮定すると，$\mathsf{IND\text{-}ID\text{-}CCA_{TS}}$ 攻撃者 $\mathcal{A}$ を使って Σ の OT-sEUF-CMA の安全性を破る署名偽造者 $\mathcal{F}$ を構成できることを示す．

Setup チャレンジャ $\mathcal{C}$ は $(vk^*, sk^*) \leftarrow \mathsf{SigGen}(1^k)$ を実行し，$\mathcal{F}$ に vk^* を渡す．$\mathcal{F}$ は $\mathcal{A}$ に対して $\mathsf{IND\text{-}ID\text{-}CCA_{TS}}$ ゲームをシミュレートするため $(tpk, tsk) \leftarrow \mathsf{TS.Setup}(1^k)$，$(params, msk) \leftarrow \mathsf{KGC.Setup}(1^k)$ を実行し，$\mathcal{A}$ に $(tpk, tsk, params)$ を与える．

Query $\mathcal{A}$ の Extract クエリと Decrypt クエリに対して，$\mathcal{F}$ は tsk と msk を持っているためすべて答えることができる．

Challenge $\mathcal{A}$ がチャレンジとして (m_0, m_1, t^*, id^*) を出力した場合，$\mathcal{F}$ はランダムに，メッセージと同じ長さの値 $s_1 \in \{0,1\}^{|m|}$ と $b \in \{0,1\}$ を選ぶ．$s_2 = m_b \oplus s_1$ を実行し，$c_1^* = \mathsf{IBE.Enc}(tpk, t^*, s_1||vk^*)$，$c_2^* = \mathsf{IBE'.Enc}(params, id^*, s_2||vk^*)$ を実行する．$m^* =$

$(c_1||c_2||t^*||id^*)$ を $\mathcal{F}$ の OT-sEUF-CMA ゲームの署名クエリとして出力し，m^* に対する署名 σ^* を得る．最後に $\mathcal{F}$ はチャレンジ暗号文として $c^* = (c_1^*, c_2^*, t^*, id^*, vk^*, \sigma^*)$ を $\mathcal{A}$ に返す．

Forge イベント Forge が起こるという仮定より $\mathcal{A}$ は **Phase2** において正当な暗号文 c を Decrypt オラクルに問い合わせる．このとき，IND-ID-CCA$_{\mathsf{TS}}$ ゲームの Decrypt オラクルの入力制限 $c \neq c^*$ より，$(c_1, c_2, t, id, \sigma) \neq (c_1^*, c_2^*, t^*, id^*, \sigma^*)$ が成立していなければならないため，$\mathcal{F}$ は偽造署名として $(c_1||c_2||t||id, \sigma)$ を出力する．

上記より Forge が起こると仮定すると，OT-sEUF-CMA 安全性に反する．よって，Pr[Forge] は negligible である． □

証明［補題 2］

$\mathcal{A}$ を IND-ID-CCA$_{\mathsf{TS}}$ 安全性に対する攻撃者と仮定すると，この $\mathcal{A}$ を利用して Π' の IND-ID-CCA 安全性を破る攻撃者 $\mathcal{B}$ を構成できることを示す．

Setup チャレンジャ $\mathcal{C}$ は $(params, msk) \leftarrow \mathsf{IBE.Setup}(1^k)$ を実行し，$\mathcal{B}$ に $params$ を渡す．$\mathcal{B}$ は $\mathcal{A}$ に対して IND-ID-CCA$_{\mathsf{TS}}$ ゲームをシミュレートするため $(tpk, tsk) \leftarrow \mathsf{TS.Setup}(1^k)$ を実行し，$\mathcal{A}$ に $(tpk, tsk, params)$ を与える．

Phase1 $\mathcal{A}$ の Extract クエリ id に対して，$\mathcal{B}$ は IND-ID-CCA ゲームの Extract クエリとして出力し，受け取った d_{id} を $\mathcal{A}$ に返す．$\mathcal{A}$ の Decrypt クエリ $c = (c_1, c_2, t, id, vk, \sigma)$ に対しては，$\mathsf{Verify}(vk, c_1||c_2||t||id, \sigma) =$ reject ならば $\perp$ を返す．そうでなければ，$s_1||vk' \leftarrow \mathsf{IBE.Dec}(tpk, d_t, c_1)$ を実行する．$\mathcal{B}$ は Decrypt クエリ (id, c_2) を出力し，$(s_2||vk'')$ を受け取る．もし，$vk = vk' = vk''$ ならば $m = s_1 \oplus s_2$ を実行し，m を $\mathcal{A}$ に返す．そうでなければ $\perp$ を返す．

Challenge $\mathcal{A}$ がチャレンジとして (m_0, m_1, t^*, id^*) を出力する．$\mathcal{B}$ は $(sk^*, vk^*) \leftarrow \mathsf{SigGen}(1^k)$ を実行する．メッセージと同じ長さの値 $r \in \{0,1\}^{|m|}$ をランダムに選び，$c_1^* = \mathsf{IBE.Enc}(tpk, t^*, r||vk^*)$ を実行する．$M_0 = [(m_0 \oplus r)||vk^*]$，$M_1 = [(m_1 \oplus r)||vk^*]$ を計算し，(M_0, M_1, id^*) を $\mathcal{B}$ のチャレンジとして出力し，暗号文 c_2^* を受け取る．$\mathcal{B}$ は $\sigma^* = \mathsf{Sign}(sk^*, c_1^*||c_2^*||t^*||id^*)$ を実行し，$c^* = (c_1^*, c_2^*, t^*, id^*, vk^*, \sigma^*)$ をチャレンジ暗号文として $\mathcal{A}$ に返す．

Phase2 $\mathcal{A}$ の Extract クエリに対しては **Phase1** と同様に動作する．Decrypt クエリ $c = (c_1, c_2, t, id, vk, \sigma)$ に対しては，以下の Step1〜4 を順に実行する．

Step1. $\mathsf{Verify}(vk, c_1||c_2||t||id, \sigma) =$ reject ならば $\perp$ を返す．

Step2. $vk = vk^*$ ならばシミュレートを停止しランダムビットを出力する．

Step3. $(c_2, id) = (c_2^*, id^*)$ ならば $\perp$ を返す．

Step4. **Phase1** と同様に動作する．

Guess $\mathcal{B}$ は $\mathcal{A}$ が出力する推測値 $\tilde{b}$ をそのまま出力する．

上記の IND-ID-CCA 攻撃者 $\mathcal{B}$ の構成において，**Phase2** での IND-ID-CCA$_{\mathsf{TS}}$ 攻撃者 $\mathcal{A}$ の Decrypt クエリに対するシミュレートについて説明する．

Step1 の場合：本構成の Decrypt アルゴリズムの Step1 の判定条件を満たさない．よって正しくシミュレートできている．

Step2 の場合：イベント Forge である．

Step3 の場合：$c_2 (= c_2^*)$ の復号結果は M_0 あるいは M_1 であるが，$vk \neq vk^*$ であるため，本構成の Decrypt アルゴリズムの Step4 の判定条件を満たさない．よって正しくシミュレートできている．

Step4 の場合：$(c_2, id) \neq (c_2^*, id^*)$ が成立しているため，$\mathcal{B}$ の IND-ID-CCA ゲームの Decrypt クエリとして (c_2, id) を出すことができる．よって正しくシミュレートできている．

よって Forge が起こらない限り，完全なシミュレーションになっている．以上より，

$$\Pr[\mathtt{Succ}|\overline{\mathtt{Forge}}] = Adv_{\Gamma,\mathcal{A}}^{\text{IND-ID-CCA}_{\mathsf{TS}}} + \frac{1}{2}$$

が成立する． □

また，

$$\begin{aligned}
\Pr[\mathtt{Succ}] &\geq \Pr[\mathtt{Succ} \wedge \overline{\mathtt{Forge}}] \\
&= \Pr[\mathtt{Succ}|\overline{\mathtt{Forge}}] \cdot \Pr[\overline{\mathtt{Forge}}] \\
&= \Pr[\mathtt{Succ}|\overline{\mathtt{Forge}}] \cdot (1 - \Pr[\mathtt{Forge}]) \\
&= \Pr[\mathtt{Succ}|\overline{\mathtt{Forge}}] - \Pr[\mathtt{Succ}|\overline{\mathtt{Forge}}] \\
&\quad \cdot \Pr[\mathtt{Forge}] \\
&\geq \Pr[\mathtt{Succ}|\overline{\mathtt{Forge}}] - \Pr[\mathtt{Forge}]
\end{aligned}$$

となることから，**補題 2** より

$$\Pr[\mathtt{Succ}] \geq Adv_{\Gamma,\mathcal{A}}^{\text{IND-ID-CCA}_{\mathsf{TS}}} + \frac{1}{2} - \Pr[\mathtt{Forge}]$$

が成立する．よって $Adv_{\Gamma,\mathcal{A}}^{\text{IND-ID-CCA}_{\mathsf{TS}}}$ が無視できないとすると，**補題 1** より $Adv_{\Pi,\mathcal{B}}^{\text{IND-ID-CCA}} = |\Pr[\mathtt{Succ}] - \frac{1}{2}|$ も無視できなくなるため**定理 1** は成立する． □

7.2.2 IND-ID-CCA$_{\mathsf{CR}}$ 安全性

定理 2 ID ベース暗号 Π が IND-ID-CCA 安全，One-Time 署名 Σ が OT-sEUF-CMA 安全ならば，提案方式 Γ は IND-ID-CCA$_{\mathsf{CR}}$ 安全性を満たす．

$\mathcal{A}$ を Γ の IND-ID-CCA$_{\mathsf{CR}}$ 安全性に対する攻撃者とする．また暗号文 $c = (c_1, c_2, t, id, vk, \sigma)$ が $\mathsf{Verify}(vk, c_1||c_2||t||id, \sigma) =$ accept を満たす場合，この暗号文を正当な暗号文と呼ぶこととし，$c^* = (c_1^*, c_2^*, t^*, id^*, vk^*, \sigma^*)$ でチャレンジ

暗号文を表すとする．ここで 2 つのイベント Forge，Succ を定義する．

Forge：$\mathcal{A}$ が **Phase2** において復号オラクルに正当な暗号文 $(c_1, c_2, t, id, vk^*, \sigma)$ を問い合わせる．

Succ：$\mathcal{B}$ が IND-ID-CCA ゲームに勝利する．

このとき，以下の 2 つの補題が成立する．

補題 3 $\Pr[\mathtt{Forge}]$ は negligible である．

補題 4 $\Pr[\mathtt{Succ}|\overline{\mathtt{Forge}}] = Adv^{\mathsf{IND\text{-}ID\text{-}CCA_{CR}}}_{\Gamma,\mathcal{A}} + \frac{1}{2}$ である．

補題 3 は**補題 1** と，**補題 4** は**補題 2** と同様に証明できる．

また，

$$\begin{aligned}\Pr[\mathtt{Succ}] &\geq \Pr[\mathtt{Succ} \wedge \overline{\mathtt{Forge}}] \\ &\geq \Pr[\mathtt{Succ}|\overline{\mathtt{Forge}}] - \Pr[\mathtt{Forge}]\end{aligned}$$

となることから，**補題 4** より

$$\Pr[\mathtt{Succ}] \geq Adv^{\mathsf{IND\text{-}ID\text{-}CCA_{CR}}}_{\Gamma,\mathcal{A}} + \frac{1}{2} - \Pr[\mathtt{Forge}]$$

が成立する．よって $Adv^{\mathsf{IND\text{-}ID\text{-}CCA_{CR}}}_{\Gamma,\mathcal{A}}$ が無視できないとすると，**補題 3** より $Adv^{\mathsf{IND\text{-}ID\text{-}CCA}}_{\Pi,\mathcal{B}} = |\Pr[\mathtt{Succ}] - \frac{1}{2}|$ も無視できなくなるため**定理 2** は成立する． □

8. おわりに

ID ベース暗号に時限式暗号の機能を持たせた時限式 ID ベース暗号を提案した．さらに，その安全性定義と構成方法を示した．提案した構成法は ID ベース暗号と One-Time 署名からなる generic construction である．構成要素である ID ベース暗号が IND-ID-CCA 安全，One-Time 署名が OT-sEUF-CMA 安全ならば，構成された TRIBE 方式は $\mathsf{IND\text{-}ID\text{-}CCA_{CR}}$ 安全性および $\mathsf{IND\text{-}ID\text{-}CCA_{TS}}$ 安全性を満たすことをスタンダードモデルで証明した．

参考文献

[1] Cathalo, J., Libert, B. and Quisquater, J.-J.: Efficient and Non-interactive Timed-Release Encryption, *ICICS 2005*, Qing, S., Mao, W., Lopez, J. and Wang, G. (Eds.), Lecture Notes in Computer Science, Vol.3783, pp.291–303, Springer-Verlag (2005).

[2] Chan, A.C.-F. and Blake, I.F.: Scalable, Server-Passive, User-Anonymous Timed Release Cryptography, *ICDCS 2005*, pp.504–513, IEEE Computer Society (2005).

[3] Cheon, J.H., Hopper, N., Kim, Y. and Osipkov, I.: Timed-Release and Key-Insulated Public Key Encryption, *FC 2006*, Di Crescenzo, G. and Rubin, A. (Eds.), Lecture Notes in Computer Science, Vol.4107, pp.191–205, Springer-Verlag (2006).

[4] Cheon, J.H., Hopper, N., Kim, Y. and Osipkov, I.: Provably Secure Timed-Release Public Key Encryption, *ACM Trans. Information and System Security* (*TISSEC*), Vol.11, No.2, Article 4 (2008).

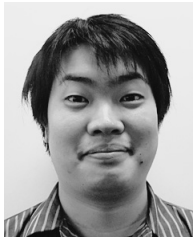
[5] Dodis, Y. and Katz, J.: Chosen-Ciphertext Security of Multiple Encryption, *TCC 2005*, Kilian, J. (Ed.), Lecture Notes in Computer Science, Vol.3378, pp.188–209, Springer-Verlag (2005).

[6] Fujioka, A., Okamoto, Y. and Saito, T.: Generic Construction of Strongly Secure Timed-Release Public-Key Encryption, *IEICE Trans.*, Vol.96-A, No.1, pp.76–91 (2013).

[7] Hwang, Y.H., Yum, D.H. and Lee, P.J.: Timed-Release Encryption with Pre-open Capability and Its Application to Certified E-mail System, *ISC 2005*, Zhou, J., Lopez, J., Deng, R.H. and Bao, F. (Eds.), Lecture Notes in Computer Science, Vol.3650, pp.344–358, Springer-Verlag (2005).

[8] May, T.: Timed-Release Crypto, Manuscript (1993).

[9] Nakai, Y., Matsuda, T., Kitada, W. and Matsuura, K.: Efficient Generic Constructions of Timed-Release Encryption with Pre-open Capability, *IWSEC 2009*, Takagi, T. and Mambo, M. (Eds.), Lecture Notes in Computer Science, Vol.5824, pp.53–70, Springer-Verlag (2009).



押切 徹

平成 25 年東京電機大学工学部第二部情報通信工学科卒業．現在，東京電機大学大学院工学研究科情報通信工学専攻在学中．

齊藤 泰一 （正会員）

平成元年早稲田大学理工学部数学科卒業．平成 3 年早稲田大学大学院理工学研究科修士課程数学専攻修了．同年日本電信電話株式会社へ入社．平成 13 年中央大学理工学研究科情報工学専攻博士後期課程修了．平成 20 年より東京電機大学教授．暗号理論，情報セキュリティの研究に従事．博士（工学）．電子情報通信学会会員．