SCIS 2012 The 29th Symposium on
Cryptography and Information Security
Kanazawa, Japan, Jan. 30 - Feb. 2, 2012
The Institute of Electronics,
Information and Communication Engineers

関数型暗号の応用: 鍵隔離暗号の構成
Application of Functional Encryption: Key-Insulated Encryption

星野 文学* 藤岡 淳*
Fumitaka Hoshino Atsushi Fujioka

あらまし 暗号システムを運用する際, 秘密鍵は常に漏洩の危機に晒されている. 特に個人用の端末は運用ミスやマルウェアの脅威により暗号学的に十分高い確率で潜在的侵入を受けていると考えて良い. 狭義の暗号では秘密鍵の漏洩事故はその事故が起こった事すら検出できない場合もあり, 暗号システムに対する重大な問題点となっている. Eurocrypt 2002 において, Dodis らは鍵隔離暗号 (Key-Insulated Encryption) の概念を提案し, この問題に答えた. 鍵隔離暗号では単一の公開鍵に対し多数の秘密鍵が存在する. 暗号文の送信者は公開鍵と現在時刻を用いて平文を暗号化し, 時刻 t に暗号化された暗号文は t 番目の秘密鍵を使用しないと復号できない. そして全ての秘密鍵はヘルパと呼ばれる特別な参加者が協力しないと得る事ができず, 仮に幾つかの秘密鍵が漏洩しても, 対応する時刻に暗号化された暗号文以外は攻撃されない. ここでヘルパは個人用端末からは隔離されており, より厳格に運用されるとする. 本稿では関数型暗号を用いた鍵隔離暗号の構成について考察する. 関数型暗号とは属性ベース暗号や述語暗号を含む ID ベース暗号の一般化概念で近年活発に研究されているものである. Crypto 2010 において, 岡本らが標準的な暗号学的仮定の下, 任意の述語に関して適応的 payload-hiding な関数型暗号を提案した事により, 属性ベース暗号などの関数型暗号を他の暗号の部品として用いる事に実装上の問題はほとんど無くなった. 本稿では岡本らの関数型暗号を用いた 2 つの鍵隔離暗号の構成方法を提案する. これらの構成では鍵更新の間隔を公開鍵とは独立に設定出来るため柔軟な運用が可能となる. さらに, 本稿ではこの 2 つの鍵隔離暗号について理論的な性能の比較を行なう. そして, 具体的な状況を想定したソフトウェア実装にて現実的な性能を確認したので結果を報告する.

キーワード Functional Encryption, Key-Insulated Encryption, Identity-Based Encryption, Attribute-Based Encryption, Predicate Encryption

1 はじめに

良く設計された暗号アプリケーションを攻撃するとき, 暗号解析によって暗号そのものを攻撃するより, システムを攻撃して秘密に直接アクセスする方がはるかに簡単である事が多い. 近年, OS やアプリケーションの脆弱性が修正プログラム提供前に攻撃される事態, 即ちゼロデイ攻撃が頻繁に報告されている [50]. ゼロデイ攻撃は, 一般的なネットワーク端末の利用状況下ではどんなに脆弱性対策を行っても完全に回避する事は困難である. そのような端末は暗号学的に十分高い確率で潜在的侵入を受けていると考えて良い. 秘密鍵を伴う暗号演算がこのような端末上で実行される事は本来望ましい事では無いが, ネットワークに接続されているからこそ暗号が必要なのであって, 現実にはそのような需要は極めて大きい. 即ち, ネットワーク端末で暗号システムを運用する時, 秘密鍵は常に漏洩の危機に晒されている. 秘密鍵の漏洩は暗号系にとって真に深刻な問題であり, 計算量的に安全な暗号におけるセキュリティパラメータをいくら大きく設定しても秘密鍵の漏洩に対しては何の意味も持たない. しかも, 秘密鍵漏洩事故の発生は実際に漏洩事故が起こっても気付かれない事が多い. 従って一般に暗号システムを安全に運用したい場合, 発行より一定の期間を過ぎた秘密鍵は破棄し, 新しい秘密鍵に更新する必要がある. 漏洩事故が単位時間当たり発生する確率を δ (定数) とするなら $1/\delta$ 程度の時間で漏洩事故が発生すると考えて良い. 正確にはシステムの運用開始から, ある時刻 t までに漏洩事故が発生している確率 p は

$$p = 1 - (1 - \delta)^t$$

* 日本電信電話株式会社 情報流通プラットフォーム研究所, 〒 180-8585 東京都武蔵野市緑町 3-9-11, NTT Information Sharing Platform Laboratories, NTT Corporation, 3-9-11 Midori-cho, Musashino-shi, Tokyo, 180-8585 Japan

であるから, $t > 1/\delta$ であるなら e を自然対数の底として

$$p > 1 - 1/e > 0.63$$

となり無視できない確率で漏洩が起こっている. 従って $t \ll 1/\delta$ の間に秘密鍵を更新する必要がある. 情報処理推進機構 (IPA) は 2009 年 4 月から 2010 年 1 月の 10ヶ月間に 15 件の緊急対策情報を発信し, その全てがゼロデイ攻撃に関するものであった事を発表している [50]. 利用者が緊急対策情報を理解し必要な対策を行うまでは端末は完全に無防備であり, 20 日につき 1 回はその状態が発生している計算になる. 仮に非常に真面目な攻撃者が秘密鍵を狙っていると仮定するなら, 一般的なネットワーク端末では通常のマルウェア対策を行っても 20 日に 1 回は潜在的漏洩事故が発生していると考えて良く, 従って, こうした端末では $t \ll 20$ 日の間に秘密鍵を更新する事が望ましい. 役所や省庁あるいはデータセンタなど秘密を厳重に運用管理出来る領域では, δ を限りなく 0 に近づけるよう努力する事により, 鍵更新までの期間を長く取る事が可能である. 一方, 個人用の端末など, 運用ミスやマルウェアの脅威に常に晒され続ける領域では, 頻繁に鍵更新を行う必要がある. ところで, 一般に公開鍵暗号では秘密鍵を更新すると公開鍵も更新する必要がある. 秘密鍵が毎日更新されるなら, 公開鍵も毎日更新する必要がある. しかし頻繁に公開鍵が更新されるような暗号系は公開鍵の配布コストが大きく現実的とは言えない. そのため公開鍵は固定したまま秘密鍵が更新出来るような暗号系が望ましい.

1.1 関連研究

秘密の漏洩に対する対策は, 暗号学において重要な課題と考えられており, 歴史的には様々な技術の研究動機として何度も登場している. 代表的なものとして, Secret Sharing [45], Threshold Cryptography [19, 44], Proactive Cryptography [38], Exposure-Resilient Cryptography [15], Forward-Secure Cryptography [2, 4], Side-Channel Security [29, 30], Authenticated Key Exchange [6, 16, 31], Key-Insulated Cryptography [22, 5, 25], Intrusion-Resilient Cryptography [20, 21], Leakage Resilient Cryptography [1, 34] などを挙げることが出来, それぞれ密接な関係がある. これらの技術に関しては, 長い歴史の間に既に膨大な量の研究結果が発表されている. 上記の公開鍵系の鍵更新に関する問題について, Ross Anderson は 1997 年公開鍵暗号および署名における Forward-Security の概念を提案し様々な国際会議で講演を行ない, この講演の内容を 2002 年に文書化して公開した [2]. また Eurocrypt 2002 において Dodis らは鍵隔離暗号 (Key-Insulated Encryption) の概念を提案し, この問題に答えた [22].

1.1.1 鍵隔離暗号

鍵隔離暗号では単一の公開鍵に対し多数の秘密鍵が存在する. 暗号文の送信者は公開鍵と現在時刻を用いて平文を暗号化し, 時刻 t に暗号化された暗号文は t 番目の秘密鍵を使用しないと復号できない. そして全ての秘密鍵はヘルパと呼ばれる特別な参加者が協力しないと得る事ができず, 仮に幾つかの秘密鍵が漏洩しても, 対応する時刻に暗号化された暗号文以外は攻撃されない. ここでヘルパは個人用端末からは隔離されており, より厳格に運用されるとする. 公開鍵暗号を多項式回利用すれば, 容易に多項式的多数の時刻に対する鍵隔離暗号を構成する事が可能である. 従って, 通常は指数的多数の時刻に対して各種データ長や計算量が多項式のものを鍵隔離暗号と呼ぶ. 鍵隔離暗号の安全性は大雑把に捉えると

(1) ユーザの秘密が幾つか漏洩した際, 直接復号出来ない時間領域の暗号文の秘匿
(2) ヘルパの秘密が漏洩した際の全ての暗号文の秘匿
(3) ユーザとヘルパの両方の秘密が漏洩した際の過去の暗号文の秘匿

の 3 つが良く知られている. (1) を満たす鍵隔離暗号を弱い鍵隔離暗号と呼び, (1) と (2) を満たす鍵隔離暗号を強い鍵隔離暗号と呼び, (1) 〜 (3) を満たす鍵隔離暗号を Intrusion-Resilient Encryption と呼ぶ [22, 20, 21]. 本稿で扱う型の鍵隔離暗号 (ランダムアクセス鍵更新可能な鍵隔離暗号) では (3) を満たす事が出来ない [22] ので, 本稿では (3) を議論しない. また, 弱い鍵隔離暗号と公開鍵暗号が存在すれば, all-or-nothing 変換 [41, 14] を使用して比較的容易に強い鍵隔離暗号を構成する事が可能である [22]. 従って, 本稿では主に弱い鍵隔離暗号を考察する.

1.1.2 関数型暗号

Forward-Security あるいは鍵隔離暗号と ID ベース暗号 [46, 10] とは密接な関係があることが知られている [2, 22]. Bellare らは弱い鍵隔離暗号と ID ベース暗号の等価性を示している [5]. ところで近年, 関数型暗号 (Functional Encryption, FE) なる ID ベース暗号の拡張が話題となっている [12, 36, 35, 32]. 関数型暗号と ID ベース暗号には構文の違いは無い [12]. 即ち関数型暗号も ID ベース暗号と同様にセットアップ, 鍵生成, 暗号化, 復号の 4 つの確率的多項式時間アルゴリズムの組により構成される. 関数型暗号では正当性 (correctness) の定義が ID ベース暗号から拡張されており, 暗号文の受信者は復号によって, 鍵生成の入力文字列 x と暗号化の入力文字列 y に関して何らかの関数 $f(x, y)$ を評価する事が出来るようになっている. ID ベース暗号 [46, 10], 内積述語暗号 [27], 属性ベース暗号 [43, 24, 37, 7] などは全

て $f(\cdot,\cdot)$ のクラスを制限した関数型暗号と考える事ができる. 2005 年頃から, この型の高機能な暗号が様々な名称で導入されており [43, 24, 37, 7, 27, 13], それらを統一的に解釈する定式化として関数型暗号の概念が研究されている [12]. 関数型暗号と ID ベース暗号は構文上違わないのであるから, ID ベース暗号の代わりに関数型暗号を使用して鍵隔離暗号を構成するのは自然な発想である. しかし典型的な関数型暗号である属性ベース暗号は 2009 年まで選択的安全 (selectivly secure) なものしか存在していなかった [32, 35]. 選択的安全な属性ベース暗号を用いて適応的安全 (adaptivly secure) な鍵隔離暗号を構成するには, 攻撃される暗号化時刻をシミュレータが推定する必要がある. 指定可能な時刻の総量を $\mathcal{N}$ とすると, この推定が正解する確率は $O(1/\mathcal{N})$ であり, 帰着効率は $O(1/\mathcal{N})$ に落ちてしまうという問題があった. このように, 以前は汎用的な述語を記述できる関数型暗号は, 暗号アプリケーションを構成する為の部品としては使いにくいという側面があった. 2010 年に岡本らは標準的な暗号学的仮定の下, 選択暗号文攻撃に対し適応的 payload-hiding (CCA-PH) が証明できる汎用的な関数型暗号 (属性ベース暗号) を提案した [35]. 適応的安全な属性ベース暗号が使用可能になったことによって帰着効率はほとんど問題とならなくなった.

1.1.3 その他の関連研究

関数型暗号では鍵生成と暗号化は, 構文上の双対性を持つ. 即ち, 関数型暗号では

- 鍵生成と暗号化
- マスタ鍵 K と公開パラメータ P
- 鍵生成入力文字列 x と暗号化入力文字列 y
- 秘密鍵と暗号文

をそれぞれ入れ替えても, 構文が入出力順序を除き不変となる. この双対性に基づき, 同じような構成の暗号を 2 通り考える事が出来る. 鍵ポリシ属性ベース暗号 (KP-ABE) [24] と暗号文ポリシ属性ベース暗号 (CP-ABE) [7] はこの双対性に基づく暗号概念である. 鍵隔離暗号は放送型 (broadcast type) の時限式暗号 (Timed-Release Encryption) [42, 9, 17, 49] (正確には時間特定暗号 (Time-Specific Encryption) [39]) の双対概念であり, 時限式暗号と同じような構成 [26] で鍵隔離暗号を得る事ができる. 鍵隔離暗号と時限式暗号の等価性が知られている [18]. 強い鍵隔離暗号は時限式公開鍵暗号 (Timed-Release Public-Key Encryption) の双対概念で, 弱い鍵隔離暗号は (受信者の特定が無い) 時限式暗号の双対概念と考えて良く, それぞれの構成もよく似ている. また, 鍵隔離暗号や時限式暗号は放送型暗号 (Broadcast Encryption) [23, 33] にも近い暗号概念であり, 放送型暗号と同じような技術を使って構成する事ができる.

1.2 貢献

本稿では, 汎用性が高く, 高度な情報処理機能を実現できる関数型暗号を鍵隔離暗号に応用する事を考察する. 汎用的な関数型暗号の使用を前提に, 鍵隔離暗号の定義および安全性の定義を秘密鍵の有効期間を任意に設定できるモデルに自然に拡張した. そして岡本らの関数型暗号 [35] を用い, の具体的な 2 つの構成 (KP-ABE による構成と CP-ABE による構成) を与え, それぞれの理論的な効率を比較し, 具体的な計算機環境の上で実装を行い比較した.

2 定義

2.1 鍵隔離暗号

鍵隔離暗号は以下のような機能を持つ 5 つの確率的多項式時間アルゴリズムの組 (KeyGen, HelperKeyUpdate, UserKeyUpdate, Enc, Dec) により構成される.

- $\mathtt{KeyGen}(1^\lambda) \xrightarrow{\$} (pk, usk_0, hsk)$: 鍵生成 - セキュリティパラメータ 1^λ を入力とし公開鍵 pk および初期ユーザ鍵 usk_0 およびマスタヘルパ鍵 hsk を出力する. 公開鍵 pk は公開され, ヘルパはマスタヘルパ鍵 hsk をメモリに記録する. ユーザは初期ユーザ鍵 usk_0 をユーザ鍵として記録する. 初期ユーザ鍵 usk_0 は初期鍵有効期間 $T_0 = [t_{0,0}, t_{0,1})$ の間だけ有効なユーザ鍵である.
- $\mathtt{HelperKeyUpdate}(hsk, T_i) \xrightarrow{\$} hsk_i$: ヘルパ鍵更新 - ユーザは i 番目の鍵有効期間 $T_i = [t_{i,0}, t_{i,1})$ を決定し, ヘルパに鍵有効期間 T_i のヘルパ鍵を要求すると, ヘルパはマスタヘルパ鍵 hsk および鍵有効期間 T_i を入力として HelperKeyUpdate アルゴリズムを起動しヘルパ鍵 hsk_i を得, hsk_i をユーザに返す.
- $\mathtt{UserKeyUpdate}(usk_{i-1}, hsk_i) \xrightarrow{\$} usk_i$: ユーザ鍵更新 - ユーザはヘルパからヘルパ鍵 hsk_i を受け取り, 現在のユーザ鍵 usk_{i-1} およびヘルパ鍵 hsk_i を入力として UserKeyUpdate アルゴリズムを起動し新しいユーザ鍵 usk_i を得る. ユーザは現在のユーザ鍵 usk_{i-1} およびヘルパ鍵 hsk_i (および使用した乱数) をメモリから削除し, 新しいユーザ鍵 usk_i を現在のユーザ鍵とする. ユーザ鍵 usk_i は鍵有効期間 $T_i = [t_{i,0}, t_{i,1})$ の間だけ有効なユーザ鍵である.
- $\mathtt{Enc}(pk, t, m) \xrightarrow{\$} c_t$: 暗号化 - 送信者は公開鍵 pk, 暗号化時刻 t, 平文 m を入力とし Enc アルゴリズムを起動し暗号文 c_t を得る. 送信者は c_t を受信者に送る.
- $\mathtt{Dec}(usk_\ell, c_t) \xrightarrow{\$} m'$: 復号 - 受信者は現在のユーザ鍵 usk_ℓ と暗号文 c_t を入力とし Dec アルゴリズ

ムを起動し平文 m' を得る.

$\ell = O(\mathtt{poly}(\lambda))$ として任意の (可能な) 鍵有効期間の列

$$T_i = [t_{i,0}, t_{i,1}),\ (i \in \{0, \ldots, \ell\})$$

および任意の $t \in T_\ell$ に対して,

$$\Pr\left[\, m = m' \;\middle|\; \begin{array}{l} (pk, usk_\emptyset, hsk) \xleftarrow{\$} \mathtt{KeyGen}(1^\lambda, 1^n); \\ \mathtt{for}\ \ i \in \{1, \ldots, \ell\} \\ \quad usk_i \xleftarrow{\$} \mathtt{UserKeyUpdate}(usk_{i-1}, \\ \qquad \mathtt{HelperKeyUpdate}(hsk, T_i)); \\ c_t \xleftarrow{\$} \mathtt{Enc}(pk, t, m); \\ m' \xleftarrow{\$} \mathtt{Dec}(usk_\ell, c_t); \end{array} \right]$$

なる確率が λ に関して圧倒的であるとき, 鍵隔離暗号 KIE は正当 (correct) であると云う. 鍵有効期間の列 T_i に特に制限が無い鍵隔離暗号をランダムアクセス鍵更新可能な鍵隔離暗号と呼ぶ [22]. 暗号化時に指定可能な暗号化時刻の総量を $\mathcal{N}$ とする. 鍵漏洩により直接復号が可能となる暗号化時刻の総量がある閾値 $\mathcal{T}$ 以下の場合に, それ以外の暗号化時刻の暗号文の秘匿が守られる鍵隔離暗号を $(\mathcal{T}, \mathcal{N})$-鍵隔離暗号と呼び, $(\mathcal{N}-1, \mathcal{N})$-鍵隔離暗号のことを最良閾値鍵隔離暗号と呼ぶ [22]. 本稿では専らランダムアクセス鍵更新可能な最良閾値鍵隔離暗号を考察するので, 以降特に断らない限り鍵隔離暗号と記述したらランダムアクセス鍵更新可能な最良閾値鍵隔離暗号の事を意味する.

2.1.1 安全性の定義

D, E をそれぞれオラクル照会履歴保存用テーブルとして, $T = [t_0, t_1)$ に対して, 復号オラクル $\mathcal{O}_d$, ヘルパ鍵漏洩オラクル $\mathcal{O}_e$, チャレンジオラクル $\mathcal{O}_c^{(b)}$, をそれぞれ

$\mathcal{O}_d(T, c) :=$
  $D[c] \xleftarrow{\cup} T$;
  return $\mathtt{Dec}(\mathtt{UserKeyUpdate}($
    $usk, \mathtt{HelperKeyUpdate}(hsk, T)), c)$;

$\mathcal{O}_e(T) :=$
  $E \xleftarrow{\cup} T$;
  return $\mathtt{HelperKeyUpdate}(hsk, T)$;

$\mathcal{O}_c^{(b)}(t, m_0, m_1) :=$
  if $(c^* = \perp) \wedge (|m_0| = |m_1|)$ then
    $t^* \leftarrow t$; $c^* \xleftarrow{\$} \mathtt{Enc}(pk, t, m_b)$;
    return c^*;

と定義し, $\mathcal{A}$ を攻撃者として実験 $\mathrm{Exp}_{\mathcal{A}}(\lambda)$ を

$\mathrm{Exp}_{\mathcal{A}}(\lambda) :=$
  clear D, E; $t^* \leftarrow \perp$; $c^* \leftarrow \perp$;
  $(pk, usk, hsk) \xleftarrow{\$} \mathtt{KeyGen}(1^\lambda)$;
  $b \xleftarrow{\$} \{0,1\}$; $b' \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_d, \mathcal{O}_e, \mathcal{O}_c^{(b)}}(pk, usk)$;
  if $(t^* \in D[c^*] \cup E)$ then $b' \xleftarrow{\$} \{0,1\}$;
  return $b \stackrel{?}{=} b'$;

と定義し, 標本空間を $\mathrm{Exp}_{\mathcal{A}}(\lambda)$ に与えるランダムテープの空間として攻撃者の利得 $\mathrm{Adv}_{\mathcal{A}}(\lambda)$ を

$$\mathrm{Adv}_{\mathcal{A}}(\lambda) = 2\Pr\left[\mathrm{Exp}_{\mathcal{A}}(\lambda) = 1\right] - 1$$

とする. 如何なる確率的多項式時間アルゴリズム $\mathcal{A}$ に対しても $\mathrm{Adv}_{\mathcal{A}}(\lambda)$ が λ に関する無視可能関数である時, その (KeyGen, HelperKeyUpdate, UserKeyUpdate, Enc, Dec) は弱い鍵隔離暗号と呼ぶ. さらに $\mathrm{Exp}'_{\mathcal{A}}(\lambda)$ を

$\mathrm{Exp}'_{\mathcal{A}}(\lambda) :=$
  clear D, E; $t^* \leftarrow \perp$; $c^* \leftarrow \perp$;
  $(pk, usk, hsk) \xleftarrow{\$} \mathtt{KeyGen}(1^\lambda)$;
  $b \xleftarrow{\$} \{0,1\}$; $b' \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_d, \mathcal{O}_c^{(b)}}(pk, hsk)$;
  if $(t^* \in D[c^*] \cup E)$ then $b' \xleftarrow{\$} \{0,1\}$;
  return $b \stackrel{?}{=} b'$;

と定義し, 標本空間を $\mathrm{Exp}'_{\mathcal{A}}(\lambda)$ に与えるランダムテープの空間として攻撃者の利得 $\mathrm{Adv}'_{\mathcal{A}}(\lambda)$ を

$$\mathrm{Adv}'_{\mathcal{A}}(\lambda) = 2\Pr\left[\mathrm{Exp}'_{\mathcal{A}}(\lambda) = 1\right] - 1$$

とする. 如何なる確率的多項式時間アルゴリズム $\mathcal{A}$ に対しても $\mathrm{Adv}'_{\mathcal{A}}(\lambda)$ が λ に関する無視可能関数である時, その弱い鍵隔離暗号を強い鍵隔離暗号と呼ぶ.

2.2 関数型暗号

関数型暗号は以下のような機能を持つ 4 つの確率的多項式時間アルゴリズムの組 (Setup, KeyGen, Enc, Dec) により構成される.

- $\mathtt{Setup}(1^\lambda) \xrightarrow{\$} (P, K)$: セットアップ— セキュリティパラメータ 1^λ を入力とし公開パラメータ P とマスタ鍵 K を出力する確率的多項式時間アルゴリズム.
- $\mathtt{KeyGen}(K, x) \xrightarrow{\$} k_x$: 鍵生成 — マスタ鍵 K と鍵識別子 x を入力とし, x に対応する秘密鍵 k_x を出力する確率的多項式時間アルゴリズム.
- $\mathtt{Enc}(P, y) \xrightarrow{\$} c_y$: 暗号化 — 公開パラメータ P と暗号文識別子 y を入力とし, 暗号文 c_y を出力する確率的多項式時間アルゴリズム.
- $\mathtt{Dec}(k_x, c_y) \xrightarrow{\$} m$: 復号 — 秘密鍵 k_x と暗号文 c_y を入力とし, 文字列 m を出力する確率的多項式時間アルゴリズム.

ある関数 $f(\cdot,\cdot)$ が存在し $\forall x, \forall y \in \{0,1\}^{\mathrm{poly}(\lambda)}$ に対して

$$\Pr\left[m = f(x,y) \;\middle|\; \begin{array}{l} (P,K) \xleftarrow{\$} \mathtt{Setup}(1^\lambda); \\ k_x \xleftarrow{\$} \mathtt{KeyGen}(K,x); \\ c_y \xleftarrow{\$} \mathtt{Enc}(P,y); \\ m \xleftarrow{\$} \mathtt{Dec}(k_x,c_y); \end{array} \right]$$

なる確率が λ に関して圧倒的であるとき, 関数型暗号 $(\mathtt{Setup},\mathtt{KeyGen},\mathtt{Enc},\mathtt{Dec})$ は正当であると云う [12, 36]. 特に, ある関係 $R(\cdot,\cdot)$ が存在し,

$$f(i,j\|m) = \begin{cases} m & (R(i,j) = \text{True の時}) \\ \perp & (R(i,j) = \text{False の時}) \end{cases}$$

なる型の f を持つ関数型暗号類は様々な暗号を包含している [12, 36, 35, 32]. 例えば

$$f(i,j\|m) = \begin{cases} m & (i = j \text{ の時}) \\ \perp & (i \neq j \text{ の時}) \end{cases}$$

を利用して ID ベース暗号を関数型暗号の一種として再定義する事ができる. より高度な R を持つ関数型暗号類が研究されており, その中でも汎用性の高いものとして属性ベース暗号 (Attribute-Based Encryption, ABE) あるいは述語暗号 (Predicate Encryption, PE) 等がよく研究されている. i を述語, j を述語変数の具体的な値として,

$$f(i,j\|m) = \begin{cases} m & (j \text{ が述語 } i \text{ を充足する時}) \\ \perp & (j \text{ が述語 } i \text{ を充足しない時}) \end{cases}$$

なる f を持つ関数型暗号は KP-ABE (Key-Policy Attribute-Based Encryption) と呼ばれる. j を述語, i を述語変数の具体的な値として,

$$f(i,j\|m) = \begin{cases} m & (i \text{ が述語 } j \text{ を充足する時}) \\ \perp & (i \text{ が述語 } j \text{ を充足しない時}) \end{cases}$$

なる f を持つ関数型暗号は CP-ABE (Ciphertext-Policy Attribute-Based Encryption) と呼ばれる. $\mathtt{Enc}$ の入力 $j\|m$ に関して, 平文 m の秘匿を Payload-Hiding, 受信者識別子 j の秘匿を Attribute-Hiding と呼ぶ.

2.2.1 安全性の定義

D, E をそれぞれオラクル照会履歴保存用テーブルとして, 復号オラクル $\mathcal{O}_d$, 鍵暴露オラクル $\mathcal{O}_e$, チャレンジオラクル $\mathcal{O}_c^{(b)}$ をそれぞれ

$\mathcal{O}_d(i,c) :=$
  $D[c] \xleftarrow{\cup} \{i\}$;
  return $\mathtt{Dec}(\mathtt{KeyGen}(K,i),c)$;

$\mathcal{O}_e(i) :=$
  $E \xleftarrow{\cup} \{i\}$;
  return $\mathtt{KeyGen}(K,i)$;

$\mathcal{O}_c^{(b)}(j,m_0,m_1) :=$
  if $(c^* = \perp) \wedge (|m_0| = |m_1|)$ then
    $j^* \leftarrow j$; $c^* \xleftarrow{\$} \mathtt{Enc}(P, j\|m_b)$;
    return c^*;

と定義し, $\mathcal{B}$ を攻撃者として実験 $\mathrm{Exp}_{\mathcal{B}}^{\mathtt{CCA\text{-}PH}}(\lambda)$ を

$\mathrm{Exp}_{\mathcal{B}}^{\mathtt{CCA\text{-}PH}}(\lambda) :=$
  clear D, E; $j^* \leftarrow \perp$; $c^* \leftarrow \perp$;
  $(P,K) \xleftarrow{\$} \mathtt{Setup}(1^\lambda)$;
  $b \xleftarrow{\$} \{0,1\}$; $b' \xleftarrow{\$} \mathcal{B}^{\mathcal{O}_d,\mathcal{O}_e,\mathcal{O}_c^{(b)}}(P)$;
  $I^* \leftarrow D[c^*] \cup E$;
  if $(\bigvee_{i^* \in I^*} R(i^*, j^*) = \text{True})$ then $b' \xleftarrow{\$} \{0,1\}$;
  return $b \stackrel{?}{=} b'$;

と定義し, 標本空間を $\mathrm{Exp}_{\mathcal{B}}^{\mathtt{CCA\text{-}PH}}(\lambda)$ に与えるランダムテープの空間として攻撃者の利得 $\mathrm{Adv}_{\mathcal{B}}^{\mathtt{CCA\text{-}PH}}(\lambda)$ を

$$\mathrm{Adv}_{\mathcal{B}}^{\mathtt{CCA\text{-}PH}}(\lambda) = 2\Pr\left[\mathrm{Exp}_{\mathcal{B}}^{\mathtt{CCA\text{-}PH}}(\lambda) = 1\right] - 1$$

とする. 如何なる確率的多項式時間アルゴリズム $\mathcal{B}$ に対しても $\mathrm{Adv}_{\mathcal{B}}^{\mathtt{CCA\text{-}PH}}(\lambda)$ が λ に関する無視可能関数である時, その $(\mathtt{Setup}, \mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec})$ は CCA-PH を満たすという [35].

3 関数型暗号に基づく鍵隔離暗号

ID ベース暗号を使って弱い鍵隔離暗号を構成出来る事が知られている [2, 22]. この構成では鍵有効期間の長さは一定と考え, 異なる鍵有効期間が重なりを持つ事は無い. このとき鍵有効期間の集合は $\mathbb{N}$ (の部分集合) と同一視してよい. 本稿では汎用的な関数型暗号の使用を前提としているので, このモデルを幾らか拡張し, 鍵の有効期間を $\mathbb{N}^2$ (の部分集合) によって定義した. 即ちユーザ鍵 usk_i は鍵有効期間 $T_i = [t_0, t_1)$ の時間領域に対して有効となる. この事を実現するためには t を暗号化時刻, t_0 を鍵有効化時刻, t_1 を鍵無効化時刻として, 次のような関数型暗号を考えれば良い.

$$f(t_0\|t_1, t\|m) = \begin{cases} m & (t_0 \leq t < t_1 \text{の時}) \\ \perp & (\text{それ以外の時}) \end{cases}$$

暗号化時刻 t が鍵の有効期間 $T_i = [t_0, t_1)$ に含まれるならば平文 m が出力され, 含まれないなら $\perp$ が出力される. こうした関数型暗号を属性ベース暗号に基づいて構成する方式は 2 通り存在する. 一つは KP-ABE を用いた方式である. この場合 X を暗号化時刻の述語変数としてセットアップを行う. ヘルパは $(t_0 \leq X) \wedge (X < t_1)$ なる述語に関して鍵生成を行い, その出力をヘルパ鍵 hsk_i

として出力する. 暗号文の送信者は暗号化時刻 $X = t$ を属性に持つ暗号文 c_t を生成する. もう一つは CP-ABE を用た方式である. この場合 X_0 を鍵有効化時刻の述語変数, X_1 を鍵無効化時刻の述語変数としてセットアップを行う. ヘルパは $X_0 = t_0$ および $X_1 = t_1$ なる属性に関して鍵生成を行い, その出力をヘルパ鍵 hsk_i として出力する. 暗号文の送信者は述語 $(X_0 \leq t) \wedge (t < X_1)$ に関する暗号文 c_t を生成する. 属性ベース暗号を用いた大小比較 [7] を岡本らの関数型暗号 [35] に適用すれば, 上記どちらの方式でも適応的安全な $f(t_0 \| t_1, t \| m)$ に関する関数型暗号 FE を容易に実現できる. ID ベース暗号から弱い鍵隔離暗号を構成する方法 [2, 22] を自然に拡張して, この関数型暗号 FE から本稿の定義に従う弱い鍵隔離暗号 wKIE を次のように構成できる.

- $\mathtt{KeyGen}(1^\lambda) \xrightarrow{\$} (pk, usk_0, hsk)$:

  parse T_0 as $[t_0, t_1)$;
  $(pk, sk) \xleftarrow{\$} \mathtt{FE.Setup}(1^\lambda)$;
  return $(pk, \mathtt{FE.KeyGen}(sk, t_0 \| t_1), sk)$;

- $\mathtt{HelperKeyUpdate}(hsk, T_i) \xrightarrow{\$} hsk_i$:

  parse T_i as $[t_0, t_1)$;
  return $\mathtt{FE.KeyGen}(hsk, t_0 \| t_1)$;

- $\mathtt{UserKeyUpdate}(usk_{i-1}, hsk_i) \xrightarrow{\$} usk_i$:

  erase usk_{i-1}; return hsk_i;

- $\mathtt{Enc}(pk, t, m) \xrightarrow{\$} c_t$: return $\mathtt{FE.Enc}(pk, t \| m)$;
- $\mathtt{Dec}(usk_\ell, c_t) \xrightarrow{\$} m'$: return $\mathtt{FE.Dec}(usk_\ell, c_t)$;

弱い鍵隔離暗号 wKIE と公開鍵暗号 $\mathtt{PKE} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec})$ が存在すれば all-or-nothing 変換 [41, 14] を使用して次のような強い鍵隔離暗号 KIE を構成する事が出来る [22].

- $\mathtt{KeyGen}(1^\lambda) \xrightarrow{\$} (pk, usk_0, hsk)$:

  $(pk_0, sk_0) \xleftarrow{\$} \mathtt{PKE.KeyGen}(1^\lambda)$;
  $(pk_1, usk'_0, hsk) \xleftarrow{\$} \mathtt{wKIE.Setup}(1^\lambda)$;
  $pk \leftarrow (pk_0, pk_1)$; $usk_0 \xleftarrow{\$} (sk_0, usk'_0)$;
  return (pk, usk_0, hsk);

- $\mathtt{HelperKeyUpdate}(hsk, T_i) \xrightarrow{\$} hsk_i$:

  return $\mathtt{wKIE.HelperKeyUpdate}(hsk, T_i)$;

- $\mathtt{UserKeyUpdate}(usk_{i-1}, hsk_i) \xrightarrow{\$} usk_i$:

  parse usk_{i-1} as (sk_0, usk'_{i-1});
  $usk'_i \xleftarrow{\$} \mathtt{wKIE.UserKeyUpdate}(usk'_{i-1}, hsk_i)$;
  return $usk_i \leftarrow (sk_0, usk'_i)$;

- $\mathtt{Enc}(pk, t, m) \xrightarrow{\$} c_t$:

  parse pk as (pk_0, pk_1);
  $r \xleftarrow{\$} \mathcal{M}$;
  $c_0 \xleftarrow{\$} \mathtt{PKE.Enc}(pk_0, r \oplus m)$;
  $c_1 \xleftarrow{\$} \mathtt{wKIE.Enc}(pk_1, t, r)$;
  return $c_t \leftarrow (c_0, c_1)$;

- $\mathtt{Dec}(usk_\ell, c_t) \xrightarrow{\$} m'$:

  parse usk_ℓ as (sk_0, usk'_ℓ);
  parse c_t as (c_0, c_1);
  $r' \xleftarrow{\$} \mathtt{wKIE.Dec}(usk'_\ell, c_1)$;
  return $r' \oplus \mathtt{PKE.Dec}(sk_0, c_0)$;

4 安全性

定理 1 上記 *FE* が *CCA-PH* を満たすなら, 上記 *wKIE* は弱い鍵隔離暗号となる. また, 上記 *FE* が *CCA-PH* を満たし, かつ上記 *PKE* が *IND-CCA[40, 3]* を満たすなら, 上記 *KIE* は強い鍵隔離暗号となる.

前半および後半はそれぞれ FE の CCA-PH および PKE の IND-CCA に容易に帰着される.

5 実装

KP-ABE を用いた方式と CP-ABE を用いた方式の両方の弱い鍵隔離暗号をソフトウェア実装した. ベースとなる KP-ABE および CP-ABE はどちらも岡本らの関数型暗号 [35] より導出した. KP-ABE を用た方式と CP-ABE を用た方式とでは使用する属性変数の数が異なっている. さらに岡本らの関数型暗号では述語の中で同じ変数を何度も参照する場合に使用する双対ペアリングベクトル空間の次元を増やす必要がある [35]. これらの違いにより KP-ABE を用た方式と CP-ABE を用た方式とで実装の効率は微妙に異なる. 表 1 に各アルゴリズムの実行時間の概算をまとめた. n は時間を表現するのに必要な bit 数である. n に具体的な数値 $n = 64$ を当てはめた場合の値を表 2 にまとめた. HelperKeyUpdate, Enc, Dec については述語や属性の具体的な値によって実行時間が変化するので, これらの表には最悪の場合を記述している. また UserKeyUpdate についてはメモリ操作のみのアルゴリズムである為, 表 2 以降は省略してある. 表 3 の環境にて, これらのアルゴリズムを実装し, 各アルゴリズムの実行時間を実測した結果を表 4 に記載する. CCA 変換は BK 変換 [11] を使用した. 実測には, 1970 年 01 月 01 日 00 時 00 分 00 秒 000000 μ 秒からの経過 μ 秒を 64 bit 整数として表現した表 3 の時刻を用いた. 50 万年程度の時刻が表現できるので, この暗号系を 50 万年程度しか使用しないのであれば 64 bit で十分であろう. さらに μ 秒単位で鍵更新する必要がない場合は, より短い bit 数でも実用になると思われる. 鍵隔離暗号では, 秘密鍵に条件を付けるのであるから KP-ABE を用いるのが自然な方法のように思えるが, 本方式では

CP-ABE を用いる方が, 概ね効率が良い事が表 1〜表 4 の結果より読み取れる.

表 3: 実測環境・使用パラメータなど

CPU	`Intel Core i7-2600 3.4GHz`
コア (スレッド)/使用数	`4(8)/1(1)`
OS	`Linux 2.6.32-71.29.1.el6.x86_64 (CentOS Linux release 6.0)`
コンパイラ	`gcc version 4.4.4 20100726`
ペアリングパラメータ	254 bit 素体 BN 曲線 [8]
要素演算の速度 (表 1 参照)	$e_1 = 0.267$, $e_2 = 0.646$, $e_T = 1.66$, $p = 0.856$ (msec)
鍵有効化時刻	t_0 = 1323763344837607 (2011/12/13 17:02:24 837607u)
暗号化時刻	t = 1323763510385534 (2011/12/13 17:05:10 385534u)
鍵無効化時刻	t_1 = 1323849744837607 (2011/12/14 17:02:24 837607u)

表 4: $n = 64$ の実測値 (sec)

	KP-ABE	CP-ABE
`KeyGen`	10.48	6.07
`HelperKeyUpdate`	7.38	1.80
`Enc`	0.46	1.14
`Dec`	0.24	0.17

参考文献

[1] A. Akavia, S. Goldwasser, and V. Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In O. Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 474–495. Springer, 2009.

[2] R. Anderson. Two remarks on public key cryptology. Technical Report UCAM-CL-TR-549, University of Cambridge, Computer Laboratory, Dec. 2002.

[3] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk, editor, *CRYPTO*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45. Springer, 1998.

[4] M. Bellare and S. K. Miner. A forward-secure digital signature scheme. In Wiener [48], pages 431–448.

[5] M. Bellare and A. Palacio. Protecting against key-exposure: strongly key-insulated encryption with optimal threshold. *Appl. Algebra Eng. Commun. Comput.*, 16(6):379–396, 2006.

[6] M. Bellare and P. Rogaway. Entity authentication and key distribution. In Stinson [47], pages 232–249.

[7] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pages 321–334. IEEE Computer Society, 2007.

[8] J.-L. Beuchat, J. E. González-Díaz, S. Mitsunari, E. Okamoto, F. Rodríguez-Henríquez, and T. Teruya. High-speed software implementation of the optimal ate pairing over barreto-naehrig curves. In M. Joye, A. Miyaji, and A. Otsuka, editors, *Pairing*, volume 6487 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2010.

[9] I. F. Blake and A. C.-F. Chan. Scalable, server-passive, user-anonymous timed release public key encryption from bilinear pairing. Cryptology ePrint Archive, Report 2004/211, 2004. `http://eprint.iacr.org/`.

[10] D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. In Kilian [28], pages 213–229.

[11] D. Boneh and J. Katz. Improved efficiency for cca-secure cryptosystems built using identity-based encryption. In A. Menezes, editor, *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 87–103. Springer, 2005.

[12] D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In Y. Ishai, editor, *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 253–273. Springer, 2011.

[13] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In S. P. Vadhan, editor, *TCC*, volume 4392 of *Lecture Notes in Computer Science*, pages 535–554. Springer, 2007.

[14] V. Boyko. On the security properties of oaep as an all-or-nothing transform. In Wiener [48], pages 503–518.

[15] R. Canetti, Y. Dodis, S. Halevi, E. Kushilevitz, and A. Sahai. Exposure-resilient functions and all-or-nothing transforms. In B. Preneel, editor, *EUROCRYPT*, volume 1807 of *Lecture Notes in Computer Science*, pages 453–469. Springer, 2000.

[16] R. Canetti and H. Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In B. Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 453–474. Springer, 2001.

[17] J. H. Cheon, N. Hopper, Y. Kim, and I. Osipkov. Timed-release and key-insulated public key encryption. Cryptology ePrint Archive, Report 2004/231, 2004. `http://eprint.iacr.org/`.

[18] J. H. Cheon, N. Hopper, Y. Kim, and I. Osipkov. Timed-release and key-insulated public key encryption. In G. D. Crescenzo and A. D. Rubin, editors, *Financial Cryptography*, volume 4107 of *Lecture Notes in Computer Science*, pages 191–205. Springer, 2006.

[19] Y. Desmedt and Y. Frankel. Threshold cryptosystems. In G. Brassard, editor, *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 307–315. Springer, 1989.

[20] Y. Dodis, M. K. Franklin, J. Katz, A. Miyaji, and M. Yung. Intrusion-resilient public-key encryption. In M. Joye, editor, *CT-RSA*, volume 2612 of *Lecture Notes in Computer Science*, pages 19–32. Springer, 2003.

[21] Y. Dodis, M. K. Franklin, J. Katz, A. Miyaji, and M. Yung. A generic construction for intrusion-resilient public-key encryption. In T. Okamoto, editor, *CT-RSA*, volume 2964 of *Lecture Notes in Computer Science*, pages 81–98. Springer, 2004.

[22] Y. Dodis, J. Katz, S. Xu, and M. Yung. Key-insulated public key cryptosystems. In L. R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 65–82. Springer, 2002.

[23] A. Fiat and M. Naor. Broadcast encryption. In Stinson [47], pages 480–491.

[24] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In A. Juels, R. N. Wright, and S. D. C. di Vimercati, editors, *ACM Conference on Computer and Communications Security*, pages 89–98. ACM, 2006.

[25] Y. Hanaoka, G. Hanaoka, J. Shikata, and H. Imai. Identity-based hierarchical strongly key-insulated encryption and its application. In B. K. Roy, editor, *ASIACRYPT*, volume 3788 of *Lecture Notes in Computer Science*, pages 495–514. Springer, 2005.

[26] F. Hoshino and A. Fujioka. An Application of Functional Encryption: Veriable Timed-Release Encryption. SCIS 2011 The 2011 Symposium on Cryptography and Information Security Kokura, Japan, Jan. 25-28, 2A4-1, 2011.

[27] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In N. P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 146–162. Springer, 2008.

表 1: 各種パラメータと実行時間の概算

	KP-ABE	CP-ABE
述語変数の数	n	$2n$
最大変数参照	2	1
DPVS 次元	13	7
`KeyGen`	$(169n+74)(e_1+e_2)$	$(98n+74)(e_1+e_2)$
`HelperKeyUpdate`$*$	$(156n+52)e_2$	$(56n+52)e_2$
`UserKeyUpdate`	0	0
`Enc`$*$	$(39n+36)e_1+e_T$	$(42n+36)e_1+e_T$
`Dec`$*$	$(26n+12)p+2ne_T+7e_2$	$(14n+12)p+2ne_T+7e_2$

e_1, e_2, e_T はそれぞれ $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ 上の冪乗実行時間, p はペアリング実行時間, n は時間を表現するのに必要な bit 数, *付きのアルゴリズムは t, t_0, t_1 の値によって変動するため上界を記載

表 2: $n = 64$ の場合 (*は上界)

	KP-ABE	CP-ABE
`KeyGen`	$10890(e_1+e_2)$	$6346(e_1+e_2)$
`HelperKeyUpdate`$*$	$10036e_2$	$3636e_2$
`Enc`$*$	$2532e_1+e_T$	$2724e_1+e_T$
`Dec`$*$	$1676p+7e_2+128e_T$	$908p+7e_2+128e_T$

[28] J. Kilian, editor. *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*. Springer, 2001.

[29] P. C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In N. Koblitz, editor, *CRYPTO*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.

[30] P. C. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In Wiener [48], pages 388–397.

[31] B. A. LaMacchia, K. Lauter, and A. Mityagin. Stronger security of authenticated key exchange. In W. Susilo, J. K. Liu, and Y. Mu, editors, *ProvSec*, volume 4784 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2007.

[32] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In H. Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 62–91. Springer, 2010.

[33] D. Naor, M. Naor, and J. Lotspiech. Revocation and tracing schemes for stateless receivers. In Kilian [28], pages 41–62.

[34] M. Naor and G. Segev. Public-key cryptosystems resilient to key leakage. In S. Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 18–35. Springer, 2009.

[35] T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In T. Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 191–208. Springer, 2010.

[36] A. O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010. `http://eprint.iacr.org/`.

[37] R. Ostrovsky, A. Sahai, and B. Waters. Attribute-based encryption with non-monotonic access structures. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *ACM Conference on Computer and Communications Security*, pages 195–203. ACM, 2007.

[38] R. Ostrovsky and M. Yung. How to withstand mobile virus attacks (extended abstract). In *PODC*, pages 51–59, 1991.

[39] K. G. Paterson and E. A. Quaglia. Time-specific encryption. In J. A. Garay and R. D. Prisco, editors, *SCN*, volume 6280 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2010.

[40] C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In J. Feigenbaum, editor, *CRYPTO*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444. Springer, 1991.

[41] R. L. Rivest. All-or-nothing encryption and the package transform. In E. Biham, editor, *FSE*, volume 1267 of *Lecture Notes in Computer Science*, pages 210–218. Springer, 1997.

[42] R. L. Rivest, A. Shamir, and D. A. Wagner. Time-lock puzzles and timed-release crypto. Technical report, Cambridge, MA, USA, 1996.

[43] A. Sahai and B. Waters. Fuzzy identity-based encryption. In R. Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer, 2005.

[44] A. D. Santis, Y. Desmedt, Y. Frankel, and M. Yung. How to share a function securely. In *STOC*, pages 522–533, 1994.

[45] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[46] A. Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and D. Chaum, editors, *CRYPTO*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984.

[47] D. R. Stinson, editor. *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*. Springer, 1994.

[48] M. J. Wiener, editor. *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*. Springer, 1999.

[49] M. Yoshida, S. Mitsunari, and T. Fujiwara. Time-Capsule Encryption. *IEICE Technical Report*, ISEC2004-98:1–5, 2004.

[50] 独立行政法人 情報処理推進機構 セキュリティセンター (IPA/ISEC). 修正プログラム提供前の脆弱性を悪用したゼロデイ攻撃について, 2010. `http://www.ipa.go.jp/security/virus/zda.html`.