SCIS 2013 The 30th Symposium on
Cryptography and Information Security
Kyoto, Japan, Jan. 22 - 25, 2013
The Institute of Electronics,
Information and Communication Engineers

On Decentralized Multi-Authority Functional Encryption
Decentralized Multi-Authority Functional Encryption に関して

星野 文学*
Fumitaka Hoshino

あらまし 分権複数局関数型暗号 (Decentralized Multi-Authority Functional Encryption,DMA-FE) では 1 人の受信者に対し複数の鍵生成局がそれぞれ秘密鍵を発行する. 各鍵生成局が全く予備連絡を行なわない場合, 秘密鍵と受信者を結びつける情報が無いと, 複数の悪意ある受信者が結託して, 本来解けない暗号文が解かれる事がある. このような結託攻撃を防ぐ為, 安全な DMA-FE では大域検証可能識別子 (globally verifiable identifier,GID) と呼ばれる受信者毎に固有の識別子が用いられ, 秘密鍵は必ず GID と結びつけられて発行される. 同じ GID を持つ秘密鍵は組み合わせて復号に使用する事が可能だが, 異なる GID を持つ秘密鍵は組み合わせて使用出来ないよう暗号文が構成される. ところで, 一般にある鍵生成局が単一の受信者に対して異なる属性を持つ複数の秘密鍵を発行する事がある. 例えば運転免許証は同じドライバに対して何度も更新される. そして無事故無違反などのドライバ属性に応じて免許証の種類は変化する. このような状況で, 一つの鍵生成局が 2 つ以上の属性を管理している場合を考えると単一の受信者が同じ GID の 2 つ以上の秘密鍵を持っているので結託攻撃と全く同じ攻撃が可能となる. このような鍵再発行攻撃を防ぐ為, 更新の度に異なる GID を用いると, 新たな秘密鍵は元々の GID を持つ他の鍵と組み合わせて使用する事が出来なくなってしまう. このような問題は鍵無効化 (key revocation) 機能付き DMA-FE の研究への強力な動機となるが, 本論文ではこの問題に対しより簡易なアプローチを考察する.

キーワード Decentralized Multi-Authority Functional Encryption, Globally Verifiable Identifier, Local Identifier, Synchronized Secret Share

1 はじめに

近年, 関数型暗号 (Functional Encryption, FE) [13, 7, 18, 15, 14] なる ID ベース暗号 [21, 5] の拡張が話題となっている. 関数型暗号と ID ベース暗号には構文の違いは無い [7]. 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 [21, 5], 内積述語暗号 [11], 属性ベース暗号 [20, 10, 19, 3] 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 [20, 10, 19, 3, 11, 8], 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 [7].

この関数型暗号 (暗号文ポリシー関数型暗号 (ciphertext policy functional encryption, CP-FE)[3]) の模型では, 単一の鍵生成局が全ての受信者に共通の属性変数を定義する. そして各受信者に紐付いた属性変数の値はその唯一の鍵生成局によって決定され, 秘密鍵という形で各受信者に配布される. 仮に世界中の受信者が利用可能な暗号システムを構築しようと考えると, 世界で唯一の鍵生成局が存在し, 世の中の全ての鍵を握りしめ, 政治, 経済, 軍事, といった国家の重要機密から, だれか個人の銀行口座や逢瀬の履歴といったありとあらゆる秘密に自由にアクセス可能な, 超権力を握ってしまう事になる. このような世界観に基づく暗号模型は現実世界を無理なく抽象してるとは言い難く, 超権力の成立を避けるために暗号の適用領域が自ずと狭まってしまう. 現実世界では, 政府当局, 外国政府, 軍当局, 通貨当局, 法務当局, 外務当局, 自治体, 公安, 郵便局, 電話局, 銀行, 大学, 信販会社, ネットワーク事業者, SNS 事業者, といった具合に複数の権力機関が勝手バラバラに成立し, それぞれが独立に秘密を管理している. 分権複数局関数型暗号 (Decentralized Multi-Authority Functional Encryption, DMA-FE) [12, 16, 13] はこうした世界観に基づき設計された関数型暗号の拡張であり, DMA-FE を

* NTT セキュア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 Secure Platform Laboratories, 3-9-11, Midori-cho Musashino-shi, Tokyo 180-8585 Japan

用いると暗号文の送信者は複数の鍵生成局がそれぞれ勝手に公開している属性変数を組み合わせて使用し受信者を指定する述語を構成することが出来る. そして送信者はその述語を公開鍵として平文を暗号化することが出来る. 各鍵生成局はその鍵生成局が管理する述語変数の値を受信者毎に割り当て, 値に対応する秘密鍵を受信者に渡す.

一般に完全に独立な二つ以上の鍵生成局が, 全く何の連絡もなしに独立に秘密鍵を発行すると仮定すると, 各鍵生成局が生成した鍵が同じ受信者に対して発行されたのか否か区別する事は出来ない. 従ってそのように設計された DMA-FE では各鍵生成局が生成したどのような鍵の組み合わせも復号に利用できてしまう. 即ち複数の受信者が鍵を持ち寄ることによって, それぞれ単独では解く事の出来ない暗号文を復号出来る事がある. このような結託攻撃を防止する為, 安全な DMA-FE では同じ受信者に対して二つの鍵生成局から発行される秘密鍵は独立にはなり得ない [9]. Chase は大域検証可能識別子 (globally verifiable identifier, GID) と呼ばれる受信者固有の識別子を用いてこの結託攻撃を防止する複数局関数型暗号 (Multi-Authority Functional Encryption, MA-FE) を提案した [9]. Chase の方式は中央集権的当局 (central authority) を必要としていたので DMA-FE ではないが, 通常安全な DMA-FE ではこのアイディアを踏襲し, 異なる GID を用いて生成された秘密鍵は, 組み合わせて利用する事が出来ないよう方式が設計されている [12, 16, 13].

1.1 研究動機および貢献

本論文では, 単一の鍵生成局が 2 つ以上の属性変数を管理することが可能な DMA-FE を研究する. このような設定において, 鍵生成局が受信者の属性の時間的変化に伴い秘密鍵が再発行される状況を考える. 例えば運転免許証, パスポート (旅券), 健康保険被保険者証, 住民基本台帳カード (住基カード) のような身分証は, 同じ人物に対して何度も更新される. そしてこうした身分証は更新される度に当該人物に対する何らかの属性が変更される. 例えば日本の運転免許証の場合は, 第一種運転免許, 第二種運転免許, 仮運転免許 の 3 つの区分がある. 第一種運転免許には大型免許, 中型免許, 普通免許, 大型特殊免許, 大型二輪免許, 普通二輪免許, 小型特殊免許, 原付免許の種類があり, 第二種運転免許には大型第二種免許, 中型第二種免許, 普通第二種免許, 大型特殊第二種免許, 牽引第二種免許の種類がある. さらに自動車運転免許において, 運転に関する限定条件が付される事があり, そのような免許には AT 限定免許, 中型車 (8t) 限定免許, カタピラ車限定免許, 農耕車限定免許, 小型二輪限定免許, 小型トレーラ限定免許がある. そして運転免許にはこれらの属性の他に初回更新者, 違反運転者, 優良運転者などの運転者区分が存在する. 運転免許は複雑な属性構造を持ち, 同じ人物に対して何度も更新され, その度に属性が変化する. 他の身分証も多かれ少なかれ同様に複雑な属性構造を持ち, 同じ人物に対して何度も更新され, その度に属性が変化するであろう. そして, 各々の身分証発行局は各々の都合に従い全くバラバラのタイミングで独立に身分証を更新する. このような状況を DMA-FE の文脈で抽象化する事を考えたい.

今, 仮にある鍵生成局が 2 つの属性変数 x および y を公開しているとする. そしてある受信者の $(x, y) = (0, 1)$ なる値を持つ鍵を, $(x, y) = (1, 0)$ なる値に更新したとする. 即ち, 受信者は $(x, y) = (0, 1)$ および $(x, y) = (1, 0)$ の二つの鍵を所有する事となる. このような鍵更新を [12, 16, 13] のような従来の DMA-FE で, 同じ GID を使って単純に実行すると, 上記の結託攻撃と同様の状況が発生する. 受信者は, 更新前の鍵と更新後の鍵を組み合わせて, $(x, y) = (0, 0)$ や $(x, y) = (1, 1)$ なる値の鍵を生成する事が可能となる. このような鍵更新攻撃を防止するため更新後の鍵を更新前の鍵と異なる GID により生成すると, 新しい鍵は他の鍵生成局が生成した鍵と合わせて使う事が出来なくなってしまう. このような問題は DMA-FE で鍵無効化 (key revocation) あるいは鍵隔離暗号 (key insulated encryption) を実現する研究への強力な動機となるが, そうした機構を属性ベース暗号の機能を使って単純に実現しようとすると, やはり同じ問題に突き当たる. 本論文では局所識別子 (local identifier, LID) および同調秘密分散の概念を提案し, この問題を自然に解決する. そして [16] の DMA-FE をこの方法を使って拡張し, 結果としてランダムオラクルモデルにおいて DLIN 仮定のもと選択平文攻撃に対して適応的 payload 秘匿な方式を得る. この方式は [16] と同様に BK 変換 [6] を用いて選択暗号文攻撃に対して安全な方式に変換できる.

2 準備

2.1 鍵再発行 DMA-FE の論理体系

本論文のように鍵再発行を考えた DMA-FE(属性ベース暗号) で記述される論理が満たす体系には古典的な論理体系には無い非常に大きな困難が伴っている. 例えば, ある受信者がある鍵生成局から属性変数 x に関して $x = 0$ なる鍵を与えられたとする. この時この受信者は $x \neq 0$ なる述語で暗号化された暗号文を復号できるであろうか. 古典的な論理体系を考え, $x = 0$ であるなら $x \neq 0$ なる述語を満足する事が出来ないので, 復号出来ないとするのが典型的な属性ベース暗号の考え方である. しかし現実には受信者が $x = 0$ なる鍵を持っている事象

と $x \neq 0$ なる鍵を持っていない事象には全く何の相関も無い. 即ち, $x = 0 \wedge x \neq 0$ なる述語は必ずしも偽とならず, 排中律は存在していない. 直観主義論理のような排中律の無い論理体系は非常に良く研究されているが, 古典的な論理体系ほど使いやすいものでは無い. 従来の典型的な属性ベース暗号では, 鍵生成局が $x = 0$ なる鍵発行と $x \neq 0$ なる鍵発行を同時に行なわないと言う事を暗黙の内に約束する事によって, 排中律のような性質を無理矢理担保していた. この場合も, 受信者が $x = 0$ なる鍵と $x \neq 0$ なる鍵を両方とも持たないという状況を禁止する事は出来ていないので, 厳密な意味では排中律は無いが, 暗号の自然なアプリケーションで問題が起こる事は無かった. しかし属性が時々刻々変化したり属性を幾つも持つといったアプリケーションでは排中律の欠如は問題を起こすであろう.

本論文では鍵生成局が複数の属性変数に相関を付けて鍵発行を行なう問題を取り扱っている. この事は属性ベース暗号で記述される論理体系にさらに重大な困難をもたらしている.

例えば述語変数 a と c は同じ鍵生成局が生成し, 述語変数 b は他の鍵生成局が生成したとし, $(a = 0) \wedge ((b = 0) \vee (c = 0))$ のような述語を考える. そして受信者は $(a, c) = (0, 1)$ および $(a, c) = (1, 0)$ および $b = 1$ の鍵を持っているとする. 括弧の一番内側の述語に関して 1 つの鍵を使用する事が出来ると考えると, 受信者は述語 $(a = 0)$ に $(a, c) = (0, 1)$ の鍵, 述語 $(c = 0)$ に $(a, c) = (1, 0)$ の鍵を用いる事によりこの述語を満足する事が出来る. このような述語の評価はアプリケーションにおいて変数 a と c に強い相関があるか否かによって自然であったり不自然であったりする. 現実的なアプリケーションでは例えば c が氏名や生年月日のような滅多に変更されないものであった場合, このような述語の評価は大きな問題を起こさないであろう. 残念ながらアプリケーションによって, そのような述語のとらえ方が極めて不自然となる場合がある. 本論文の提案方式でも, そのような分離した評価がどうしても必要となる場面が存在する. そのような評価は異なる鍵生成局の変数を使ったリテラル述語を葉に持つ選言節 (OR ゲート) により生成される. 上記の述語で (a, c) に強い相関がある場合, 送信者は (a, c) をなるべく一体として捉えるよう述語を記述する必要がある. 上記の述語の場合は $((a = 0) \wedge (b = 0)) \vee (a = 0 \wedge c = 0)$ と書き直せば, 後半の述語が一体として評価されるので前述の受信者を排除する事ができる. 即ち, 本論文の DMA-FE の論理体系では厳密には分配律は成立しない. 量子論理のような分配律の無い論理体系も非常に良く研究されているが, やはり古典的な論理体系ほど使いやすいものではない. 従って, 古典的な論理体系で設計された非常に複雑な述語を本論文の DMA-FE に用いる場合は述語の意味を正確に捉え直す必要がある. 必ずしも効率的になるとは限らないが, 一般には問題となる全ての選言節に関して加法標準形へ変換すればこの問題は解消できる. 本論文では必要ならば送信者がそのような変換を行なって適切な述語が暗号化関数に入力されると仮定する.

2.2 線形秘密分散 [1]

P_i を i 番目の share (holder) として, $\mathcal{P} = \{P_1, \ldots, P_n\}$ を全ての share の集合とする. 一般的な秘密分散において, どのような share の組み合わせによって, 秘密の復元が可能となるかを定義する集合族 $\Gamma \subseteq 2^{\mathcal{P}}$ をアクセス構造と呼ぶ. 即ちある集合 $A \subseteq \mathcal{P}$ が $A \in \Gamma$ であるなら A から秘密を復元出来ると定義する. Benaloh と Leichter はまず $A \in \Gamma$ であるか否かを識別する (単調) 回路を構築し, その回路から秘密分散法を構築する方法 (単調回路構成法, monotone circuit construction) を提案した [2]. 任意のアクセス構造は加法標準形を使って容易に単調回路に変換できるので, この方法は万能である (効率的であるとは限らない). 一般に AND, OR, NOT および原始述語のみからなる多項式長の論理式はリテラル述語を葉に持ち AND ゲート, OR ゲートのみからなる多項式サイズの樹状単調回路に容易に変換できるので, そのようなクラスのアクセス構造 (いわゆる non-monotone access structure) を単調回路構成法で線形秘密分散に変換し, 関数型暗号に応用する方法が良く研究されている [19].

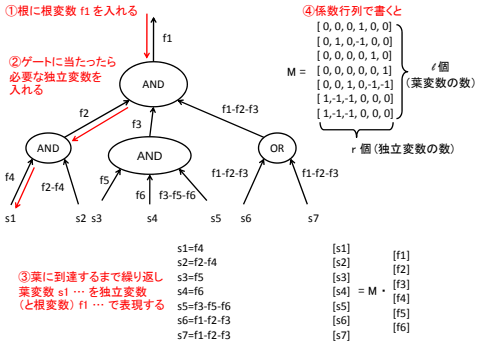

図 1: 単調回路構成法

2.3 同調秘密分散

典型的な DMA-FE では線形秘密分散を, アクセス構造を実現する機構として使用する他に, GID を実現する機構としても使用している [12, 16]. これらの方式では, 同じ GID の鍵を使用すると, 秘密分散の再構築過程においてノイズが自然にキャンセルされて, 正しい鍵を復元できるが, 異なる GID をもつ鍵が同時に利用されると高い確率でノイズがキャンセルされずに残ってしまう.

これらの方式ではアクセス構造を実現する秘密分散をそのようなノイズを計算するために再利用する (図 2). 本

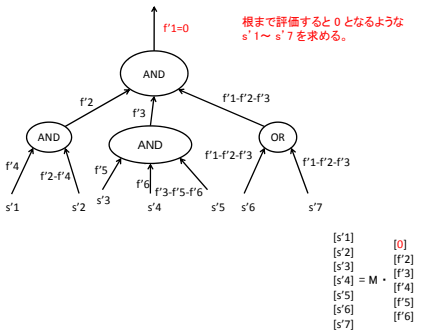

図 2: GID を実現する機構

論文ではこの概念を拡張して, アクセス構造を実現する秘密分散と同じ再構築過程 (reconstruction stage) を有する秘密分散を同調秘密分散と呼び, 局所識別子 (local identifier, LID) の概念を実現する為に同調秘密分散を利用する. LID は, その名の通り鍵生成局内部で利用する識別子 (または乱数) で, 必ずしも検証可能である必要は無く, 鍵発行の度にランダムに決定してよい. LID を導入する目的は GID と同様に, 鍵生成局が認めていない秘密鍵の組み合わせを防ぐ事であるが, GID との本質的な違いはこの拘束は鍵生成局内部で閉じている事である. 異なる鍵生成局が生成した鍵なら GID さえ一緒ならば LID が異なっていても組み合わせて使用する事ができる. 本研究では,LID を実現する為に, 回路を評価している途中で上手く消去されるような乱数を使用して暗号を構成する (図 3～5). 図 3 では同じ鍵生成局によって作成された述語変数で定義されたリテラル述語が接合される葉をグループ化している. その葉のみを含むような縮退木 (1 本とは限らない) を構成する方法を図 4 で示す. そして, それらの縮退木の根に相当する部分の値が 0 となるよう葉の乱数を決定する. この時アクセス構造を実現する秘密分散と同じ再構築過程で, 縮退木の根の値が 0 となる必要がある. 従って元の線形秘密分散が一般の線形秘密分散である場合は, その再構築過程に合あうような縮退木の秘密分散, 即ち同調秘密分散を構成する必要がある. 本論文では簡単のため, 元の回路が AND ゲートおよび OR ゲートのみから構成されており, 各ゲート毎に全ての枝が対称となるよう線形秘密分散が構成されているとする. 元の線形秘密分散をこのように構成しておくと, 同調秘密分散も同様の戦略で構成する事ができる. 即ち枝が対称となるような単調回路構成法を縮退木毎に実行すれば良い (図 4, 図 5). 求めた share generation matrix を用い乱数を決定する. 1 個の縮退木の中で LID が一致しない場合は回路を最後まで評価しても乱数が消去されずに暗号が復号出来ない. 一方, 違うグループでは異なる LID でも、乱数が消去され暗号が復号できる. 注目している鍵生成局について縮退木が複数本になる場合, 異なる縮退木には異なる LID が適用できる. このような状況は一見すると攻撃が成功してるように見えるが, 前述の通り述語の意味自体がそういう意味であると捉える.

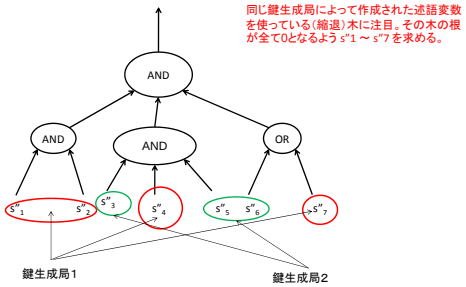

図 3: LID を実現する機構

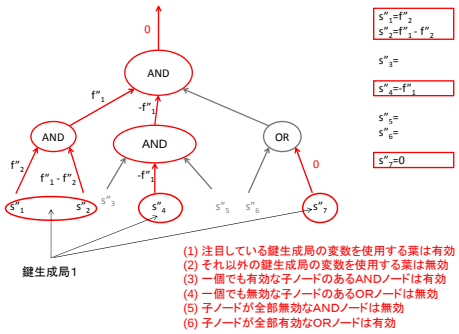

図 4: 縮退木の集合を構成する方法

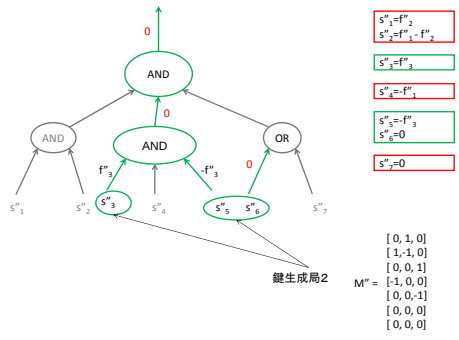

図 5: 全てのグループで繰り返す

2.4 DLIN 仮定

定義 1 (DLIN 仮定 [4]) $\beta \xleftarrow{\$} \{0,1\}$ に対して確率的チューリング機械 $\mathcal{G}_{\beta}^{DLIN}(1^{\lambda})$ を

$\mathcal{G}_{\beta}^{DLIN}(1^{\lambda})$:
$\quad \boldsymbol{param}_{\mathbb{G}} := (q, \mathbb{G}_1, \mathbb{G}_T, G, e) \xleftarrow{\$} \mathcal{G}_{\mathtt{bpg}}(1^{\lambda})$,
$\quad (\kappa, \delta, \xi, \sigma) \xleftarrow{\$} \mathbb{F}_q^4$,
$\quad Y_0 \leftarrow (\delta + \sigma)G, Y_1 \xleftarrow{\$} \mathbb{G}_1$,
$\quad \boldsymbol{return}\ \ (\boldsymbol{param}_{\mathbb{G}}, G, \xi G, \kappa G, \delta\xi G, \sigma\kappa G, Y_{\beta})$,

として, $(\boldsymbol{param}_{\mathbb{G}}, G, \xi G, \kappa G, \delta\xi G, \sigma\kappa G, Y_{\beta}) \xleftarrow{\$} \mathcal{G}_{\beta}^{DLIN}(1^{\lambda})$ としたとき, $\beta \in \{0,1\}$ を言い当てることを *decisional linear (DLIN)* 問題と呼ぶ. 確率的チューリング機械 $\mathcal{E}$ に対し, *DLIN* 問題に対する $\mathcal{E}$ の利得 $\boldsymbol{Adv}_{\mathcal{E}}^{DLIN}(\lambda)$ を

$$\boldsymbol{Adv}_{\mathcal{E}}^{DLIN}(\lambda) := \left|\Pr\left[\mathcal{E}(1^{\lambda}, \varrho) \to 1 \,\middle|\, \varrho \leftarrow \mathcal{G}_0^{DLIN}(1^{\lambda})\right] - \Pr\left[\mathcal{E}(1^{\lambda}, \varrho) \to 1 \,\middle|\, \varrho \leftarrow \mathcal{G}_1^{DLIN}(1^{\lambda})\right]\right|.$$

とする. どのような確率的多項式時間攻撃者 $\mathcal{E}$ に対しても, 利得 $\boldsymbol{Adv}_{\mathcal{E}}^{DLIN}(\lambda)$ が λ について無視可能であるという仮定を *DLIN* 仮定と呼ぶ.

2.5 DMA-FE

DMA-FEは次の 5 つのアルゴリズム (GSetup, ASetup, AttrGen, Enc, Dec) から構成される.

- $\mathtt{GSetup}(1^{\lambda}) \xrightarrow{\$} \mathtt{gparam}$: 大域セットアップ - セキュリティパラメタ 1^k を入力とし共通参照文字列 gparam を出力する確率的多項式時間アルゴリズム. 標準化機関などが適当なセキュリティパラメタを決めて誰でも参照可能な形で gparam を公開する事を想定している. 共通参照文字列 gparam に何らかの落し戸が仕掛けられる事を避ける為, gparam はその生成過程が公開検証可能である事が望ましい [16].
- $\mathtt{ASetup}(\mathtt{gparam}) \xrightarrow{\$} (\mathtt{ask}, \mathtt{apk})$: 局セットアップ - 共通参照文字列 gparam を入力とし局公開パラメタ apk とマスター鍵 ask を出力する確率的多項式時間アルゴリズム. 誰でも参照可能な gparam を利用して, 誰でも自由に鍵生成局を立ち上げる事が出来る. 各鍵生成局はそれぞれマスター鍵 ask を秘密として保持し, 局公開パラメタ apk は誰でも参照可能な形で公開する.
- $\mathtt{AttrGen}(\mathtt{ask}, \mathtt{gid}, x) \xrightarrow{\$} \mathtt{sk}_x$: 属性生成 - マスター鍵 ask, GID gid および鍵識別子 (述語変数への値の割り当て) x を入力とし, 対応する秘密鍵 $\mathtt{sk}_x$ を出力する確率的多項式時間アルゴリズム. 鍵生成局は gid なる GID を持つ受信者に対し述語変数の具体的な値 (の組) x を割り当てて, マスター鍵 ask を利用して x に対応する秘密鍵 $\mathtt{sk}_x$ を計算し, 受信者に $\mathtt{sk}_x$ を配布する.
- $\mathtt{Enc}(\mathbb{S}, m) \xrightarrow{\$} c_{\mathbb{S}}$: 暗号化 - 各鍵生成局の局公開パラメタ $\mathtt{apk}_1, \ldots, \mathtt{apk}_n$ を使用して記述した受信者識別子 (述語) $\mathbb{S}$ および平文 m を入力とし, 暗号文 $c_{\mathbb{S}}$ を出力する確率的多項式時間アルゴリズム.
- $\mathtt{Dec}(\{\mathtt{sk}_{x_1}, \ldots, \mathtt{sk}_{x_{n'}}\}, c_{\mathbb{S}}) \xrightarrow{\$} m'$: 復号 - 秘密鍵 $\mathtt{sk}_{x_1}, \ldots, \mathtt{sk}_{x_{n'}}$ および暗号文 $c_{\mathbb{S}}$ を入力とし, 平文 m' を出力する確率的多項式時間アルゴリズム.

DMA-FE では暗号文の受信者は鍵識別子 (述語変数の値) $x_1, \ldots, x_{n'}$ を持つ秘密鍵の集合 $\{\mathtt{sk}_{x_1}, \ldots, \mathtt{sk}_{x_{n'}}\}$ を所有しており, 述語変数の値 $x_1, \ldots, x_{n'}$ が述語 $\mathbb{S}$ を満足するなら圧倒的確率で $m = m'$ となる.

3 提案法

本論文では [16] に基づいて, CPA-PH な (GSetup, ASetup, AttrGen, Enc, Dec) を構成する. CCA-PH な方法への変換は [16] と同じである. ρ_i を i 番目のリテラル述語として, $\hat{\rho}(i)$ を ρ_i で使用されている述語変数とする. 簡単のため $\hat{\rho}(i)$ を単射と仮定する. この制約を緩和する方法は [16] を参照. このような制約を完全に無くす方法が [17] で提案されている.

3.1 GSetup

1^{λ} :セキュリティパラメタ

$\mathtt{GSetup}(1^{\lambda})\{$
$\quad \mathtt{param}_{\mathbb{G}} := (q, \mathbb{G}_1, \mathbb{G}_T, G, e) \xleftarrow{\$} \mathcal{G}_{\mathtt{bpg}}(1^{\lambda})$;
$\quad H \xleftarrow{\$} \{\mathtt{RandomOracle} : \{0,1\}^* \to \mathbb{G}_1\}$;
$\quad G_0 \leftarrow H(0^{\lambda})$;
$\quad G_1 \leftarrow H(0^{\lambda-1}1)$;
$\quad g_T \leftarrow e(G_0, G_1)$;
$\quad \mathtt{gparam} \leftarrow (\mathtt{param}_{\mathbb{G}}, H, G_0, G_1, g_T)$;
$\quad \mathbf{return}\ \mathtt{gparam}$;
$\}$

3.2 ASetup

$\mathtt{ASetup}(\mathtt{gparam})\{$
$\quad //\ d$: number of variables
$\quad //\ n_i$: dimension of i-th variable
$\quad \mathbf{for}\ \ i \in \{1, \ldots, d\}\ \ \mathbf{do}\ \ \{$
$\quad\quad (\mathtt{vsk}_i, \mathtt{var}_i) \leftarrow \mathtt{GenerateVar}(\mathtt{gparam}, n_i)$;
$\quad \}$
$\quad \mathtt{ask} \leftarrow (\mathtt{vsk}_1, \ldots, \mathtt{vsk}_d)$;
$\quad \mathtt{apk} \leftarrow (\mathtt{var}_1, \ldots, \mathtt{var}_d)$;
$\quad \mathbf{return}\ \ (\mathtt{ask}, \mathtt{apk})$;
$\}$

$//\ n$: dimension of the variable
$\mathtt{GenerateVar}(\mathtt{gparam}, n)\{$
$\quad (\mathtt{param}_{\mathbb{G}}, H, G_0, G_1, g_T) \leftarrow \mathtt{gparam}$;
$\quad N \leftarrow 7n + 1$;
$\quad X \xleftarrow{\$} GL(N, \mathbb{F}_q)$;
$\quad \mathbb{B} \leftarrow X \cdot G_0$;
$\quad \widehat{\mathbb{B}} \leftarrow (\mathbb{B}_1, \cdots, \mathbb{B}_{3n}, \mathbb{B}_N)$;
$\quad Y \leftarrow (X^{-1})^T$;
$\quad \mathtt{var} \leftarrow (\widehat{\mathbb{B}}, n, \mathtt{gparam})$;
$\quad \mathtt{vsk} \leftarrow (Y, \mathtt{var})$;
$\quad \mathbf{return}\ \ (\mathtt{vsk}, \mathtt{var})$;
$\}$

3.2.1 AttrGen

// gid : GID
// $x := ((t_1, \vec{x}_1), \ldots, (t_a, \vec{x}_a))$:
// 述語変数への値の割り当てリスト
// $t_i \in \{1, \ldots, d\}$: 代入されるべき述語変数の添え字
// $\vec{x}_i$: 値
$\mathtt{AttrGen}(\mathtt{ask}, \mathtt{gid}, x)\{$
  $(\mathtt{vsk}_1, \ldots, \mathtt{vsk}_d) \leftarrow \mathtt{ask}$;
  $((t_1, \vec{x}_1), \ldots, (t_a, \vec{x}_a)) \leftarrow x$;
  $G_{\mathtt{gid}} \leftarrow H(\mathtt{gid})$;
  $G_{\mathtt{lid}} \xleftarrow{\$} \mathbb{G}_1$;
  for $i \in \{1, \ldots, a\}$ **do** {
    $j \leftarrow t_i$;
    $k_i \xleftarrow{\$} \mathtt{AssignVar}(\mathtt{vsk}_j, G_{\mathtt{gid}}, G_{\mathtt{lid}}, \vec{x}_i)$;
  }
  $\mathtt{sk}_x \leftarrow (k_1, \ldots, k_a)$;
  return $\mathtt{sk}_x$;
}

$\mathtt{AssignVar}(\mathtt{vsk}, G_{\mathtt{gid}}, G_{\mathtt{lid}}, \vec{x})\{$
  $(Y, \mathtt{var}) \leftarrow \mathtt{vsk}$;
  $(\widehat{\mathbb{B}}, n, \mathtt{gparam}) \leftarrow \mathtt{var}$;
  $(\mathtt{param}_{\mathbb{G}}, H, G_0, G_1, g_T) \leftarrow \mathtt{gparam}$;
  $\vec{\phi} \xleftarrow{\$} \mathbb{F}_q^n$;
  $k^* \leftarrow Y \cdot (\vec{x} \cdot G_1, \vec{x} \cdot G_{\mathtt{gid}}, \vec{x} \cdot G_{\mathtt{lid}}, \mathcal{O}^{3n}, \vec{\phi} \cdot G_1, \mathcal{O})$;
  $k \leftarrow (k^*, \vec{x}, \mathtt{var})$;
  return k ;
}

3.2.2 Enc

// l : # of leaves (literal predicates)
// r : # of independent variables for M
// r'' : # of independent variables for M''
// f, f', f'' : independent variables
// s, s', s'' : leaf variables
// $s_0 = (1, 0, \ldots, 0) \cdot f$: root variable
// M : $l \times r$-$\mathbb{F}_q$ 行列 (share generator)
// M'' : $l \times r''$-$\mathbb{F}_q$ 行列 (synchronized share generator)
// ρ : list of literal predicates
// ρ_i : i-th literal predicate $(\mathtt{Neg}, \mathtt{var}, \vec{v})$;
// $\mathbb{S} := (M, M'', \rho)$;
$\mathtt{Enc}(\mathbb{S}, m)\{$
  $(M, M'', \rho) \leftarrow \mathbb{S}$;
  $f \xleftarrow{\$} \mathbb{F}_q^r$; $f' \xleftarrow{\$} \{0\} \times \mathbb{F}_q^{r-1}$; $f'' \xleftarrow{\$} \mathbb{F}_q^{r''}$;
  $s \leftarrow M \cdot f$; $s' \leftarrow M \cdot f'$; $s'' \leftarrow M'' \cdot f''$;
  $s_0 \leftarrow f_1$;
  for $i \in \{1, \ldots, \ell\}$ **do** {
    $(\mathtt{Neg}, \mathtt{var}, \vec{v}) = \rho_i$;
    $(\widehat{\mathbb{B}}, n, \mathtt{gparam}) \leftarrow \mathtt{var}$;
    $N \leftarrow 7n + 1$;
    $\eta \xleftarrow{\$} \mathbb{F}_q$;
    if $\mathtt{Neg} = 0$ **then** {
      $(\theta, \theta', \theta'') \xleftarrow{\$} \mathbb{F}_q^3$;
      $c_i \leftarrow (s_i \vec{e_1} + \theta \vec{v}, s'_i \vec{e_1} + \theta' \vec{v}, s''_i \vec{e_1} + \theta'' \vec{v}, \eta)_{\widehat{\mathbb{B}}}$;
    } **else** {
      $c_i \leftarrow (s_i \vec{v}, s'_i \vec{v}, s''_i \vec{v}, \eta)_{\widehat{\mathbb{B}}}$;
    }
  } ;
  $c_{\ell+1} \leftarrow m \, g_T^{s_0}$;
  $c_{\mathbb{S}} \leftarrow (\mathbb{S}, c_1, \ldots, c_\ell, c_{\ell+1})$;
  return $c_{\mathbb{S}}$;
}

3.2.3 Dec

// Γ : 属性鍵 $(t, \vec{x}, k^*, \mathtt{apk})$ のリスト
// a : Γ のエントリー数
$\mathtt{Dec}(\{\mathtt{sk}_{x_1}, \ldots, \mathtt{sk}_{x_{n'}}\}, c_{\mathbb{S}})\{$
  $\{k_1, \ldots, k_a\} \leftarrow \bigcup_i \mathtt{sk}_{x_i}$;
  $(\mathbb{S}, c_1, \ldots, c_\ell, c_{\ell+1}) \leftarrow c_{\mathbb{S}}$;
  $(\mathtt{acc}, \beta, J) \leftarrow \mathtt{Accept}(\mathbb{S}, \{k_1, \ldots, k_a\})$;
  // acc : accept or reject
  // β : modification factor
  // J : map from literal index to key index
  // Accept : see appendix
  if $\mathtt{acc} = \mathrm{True}$ **then** {
    $K = 1$;
    for $i \in \{1, \ldots, \ell\}$ **do** {
      if $\beta_i \neq 0$ **then** {
        $j \leftarrow J_i$;
        $(k^*, \vec{x}, \mathtt{var}) \leftarrow k_j$;
        $K \xleftarrow{\times} e(c_i, k^*)^{\beta_i}$;
      }
    }
    $m' \leftarrow c_{\ell+1} / K$;
    return m' ;
  } **else** {
    return $\perp$;
  }
}

4 安全性

定理 1 提案 *DMA-FE* はランダムオラクルモデルにおいて *DLIN* 仮定のもと選択平文攻撃に対して適応的 *payload* 秘匿である.

暗号の構成は基本的に [16] とほとんど同じなので, 安全性の証明はほぼ [16] を踏襲する. CCA への変換も [16] を踏襲する.

参考文献

[1] A. Beimel. *Secure Schemes for Secret Sharing and Key Distribution*. PhD thesis, Israel Institute of Technology, Haifa, Israel, 1996.

[2] J. C. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In S. Goldwasser, editor, *CRYPTO*, volume 403 of *Lecture Notes in Computer Science*, pages 27–35. Springer, 1988.

[3] 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.

[4] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. K. Franklin, editor, *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer, 2004.

[5] D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. In J. Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.

[6] 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.

[7] 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.

[8] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In Vadhan [22], pages 535–554.

[9] M. Chase. Multi-authority attribute based encryption. In Vadhan [22], pages 515–534.

[10] 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.

[11] 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.

[12] A. Lewko and B. Waters. Decentralizing attribute-based encryption. In *Proceedings of the 30th Annual international conference on Theory and applications of cryptographic techniques: advances in cryptology*, EUROCRYPT'11, pages 568–588, Berlin, Heidelberg, 2011. Springer-Verlag.

[13] A. B. Lewko. *Functional Encryption: New Proof Techniques and Advancing Capabilities*. PhD thesis, The University of Texas at Austin, 2012.

[14] 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.

[15] 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.

[16] T. Okamoto and K. Takashima. Decentralized attribute-based signatures. *IACR Cryptology ePrint Archive*, 2011:701, 2011.

[17] T. Okamoto and K. Takashima. Fully secure unbounded inner-product and attribute-based encryption. *IACR Cryptology ePrint Archive*, 2012:671, 2012.

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

[19] 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.

[20] 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.

[21] 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.

[22] S. P. Vadhan, editor. *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, volume 4392 of *Lecture Notes in Computer Science*. Springer, 2007.

A grocery

```
Accept(𝕊, k) {
  (M, M'', ρ) ← 𝕊 ;
  for i ∈ {1, ..., ℓ} do {
    (Neg, var, v⃗) ← ρ_i ;
    if (∃j s.t. k_j = (k*, var, x⃗) s.t. (
          ((Neg = 0) ∧ (x⃗ · v⃗ = 0)) ∨
          ((Neg ≠ 0) ∧ (x⃗ · v⃗ ≠ 0)))) then {
      J_i ← j ;
      γ_i ← x⃗ · v⃗ ;
    } else {
      J_i ← ⊥ ;
      M_i ← 0⃗ ;
    }
  }
  (M', V, w, ℓ') ← GaussianElimination(M) ;
  // M を階段行列 M' に変換する
  // M' : 階段行列化された M
  // V : M を階段行列化する為の ℓ×ℓ 行列 (M' = V·M)
  // w : w_i が M'_i の先頭非ゼロ成分要素のインデックス
  // ℓ' : M' の非ゼロ行の個数
  σ ← (1, 0, ..., 0) ;
  α ← 0⃗ ; // (α_1, ..., α_ℓ) ← (0, ..., 0) 初期化
  for i ∈ {1, ..., ℓ'} do {
    t ← w_i ;
    α +← σ_t · V_i ; σ -← σ_t · M'_i ;
  }
  acc ← (σ ?= 0⃗) ;
  // M_i の線形結合で σ をゼロに出来るなら True
  β ← α ;
  for i ∈ {1, ..., ℓ} do {
    (Neg, var, v⃗) ← ρ_i ;
    if (β_i ≠ 0) ∧ (Neg ≠ 0) ∧ (γ_i ≠ 0) then β_i ÷← γ_i ;
  }
  return (acc, β, J) ;
}
GaussianElimination(M) {
  V ← I_ℓ ; // ℓ×ℓ 単位行列
  t ← 0 ;
  ℓ' ← 0 ;
  for i ∈ {1, ..., ℓ} do {
    // 部分ピボット選択開始
    t++ ; if t > r then goto last ;
    j ← i ;
    while M_{j,t} = 0 do {
      j++ ;
      if j > ℓ then {
        t++ ; if t > r then goto last ;
        j ← i ;
      }
    }
    swap(M_i, M_j) ; swap(V_i, V_j) ; // 行の入れ替え
    // 部分ピボット選択終了
    ℓ' ← i ;
    w_i ← t ;
    V_i ÷← M_{i,t} ; M_i ÷← M_{i,t} ;
    for j ∈ {i+1, ..., ℓ} do {
      if M_{j,t} ≠ 0 then {
        V_j -← M_{j,t} · V_i ; M_j -← M_{j,t} · M_i ;
      }
    }
  }
last : return (M, V, w, ℓ') ;
}
```