数理解析研究所講究録 1420 巻 2005 年 117-127

双線型写像の公開鍵暗号への応用に関して

Application of bilinear map to public-key encryption

北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology

宮地充子
Atsuko Miyaji

概要

本報告書は，双線型写像により可能になる公開鍵暗号の様々な応用，さらに残る課題について報告する．

1 はじめに

ネットワークの普及により各種サービスがネットワークを介して実現できる基盤が確立しつつある．電子医療，電子政府などでは，物理的な文書に代わり電子的なデータがネットワーク上流通するようになる．こうして情報セキュリティの技術である公開鍵暗号系によるデータの秘匿及び完全性の技術が不可欠になった．公開鍵暗号系とは，公開鍵暗号とディジタル署名という二つの技術の総称で，ユーザは秘密鍵及び公開鍵と呼ばれる異なる鍵ペアを生成し，公開鍵は公開し，該当ユーザへのデータの暗号化や該当ユーザの作成したデータに対する署名の検証に用い，秘密鍵は秘密に管理し，自分宛の暗号文の復号や作成したデータに対する署名の生成に用いる．このような公開鍵暗号系を実現するのが，素因数分解問題，楕円曲線上の離散対数問題 (ECDLP)，有限体上の離散対数問題である．これらはすべて数論の分野で良く知られている事実を応用したものである．

近年，このような従来の問題では解決できない技術が脚光をあびるようになってきた．一つは，鍵進化公開鍵暗号系と呼ばれる新しい公開鍵暗号のパラダイムであり，もう一つはブロードキャスト暗号と呼ばれる 1 対 n の動的に受信者が変わる秘匿通信である．これらを解決する新しい数論の応用が双線型写像である．なお，双線型写像のうち実際に暗号への応用が期待されているのは，Weil 対 [7] である．

従来公開鍵暗号系では，ユーザがもつ秘密鍵は露呈しない，あるいは，秘密鍵は耐タンパ機器に格納され流出しないという前提のもと，その方式への理論攻撃に対する安全性を議論した．しかしながら，秘密鍵が露呈しないという前提は，明らかに非現実的である．実際，秘密鍵の露呈攻撃は，公開鍵暗号系の方式そのものに対する理論攻撃より容易であり，秘密鍵の露呈を考慮した安全性の議論は必須といえる．こうして鍵進化ディジタル署名及び鍵進化公開鍵暗号という鍵進化公開鍵暗号系の概念が提案された [?]．鍵進化公開鍵暗号系とは，公開鍵と秘密鍵という 2 つの鍵ペアのうち，公開鍵は不変で対応する秘密鍵のみ定期的に進化（変化）することで，秘密鍵の露呈攻撃に強化した概念である．鍵進化暗号は，その形態から大きく 2 つに分けられる．1 つはユーザのみの 1 エンティティで実現される方式であり，もう一つは 2 エンティティ，ユーザとベース[1] により実現される方式である．鍵進化公開鍵暗号系は，1 エンティティからなる Forward secure 暗号系，2 エンティティからなる Key-insulated 暗号系，Intrusion-resilient 暗号系の 3 つに分けられる．なお，最強の安全性を有するのが Intrusion-resilient 暗号系である．鍵進化ディジタル署名に関しては，2002 年に素因数分解に基づく効率的な Intrusion-resilient 署名が提案され，さらにそのモ

[1] ユーザは PDA などの携帯端末，ベースは自宅のパーソナルコンピュータに対応する

デル化の提案により，任意の一方向性関数を利用したIntrusion-resilient 署名が可能となった．一方，鍵進化公開鍵暗号は従来の素因数分解問題，ECDLP，DLP などでは実現ができず，双線型ディッフィヘルマン問題を利用してIntrusion-resilient 暗号が 2003 年に初めて実現された [4].

一方，ブロードキャスト暗号とは，デジタルコンテンツの著作権管理・保護を実現する電子配信の方法である [?, ?]. ブロードキャスト暗号は，コンテンツを得ることが許可されたユーザの集合に対し，公衆放送網を通しての安全で効率的な情報配信を目的とした暗号で，以下のフェーズからなる．(1) 秘密情報を各ユーザに秘密通信で配布，(2) 配信者は復号を許可するユーザの部分集合 S を決定，(3) コンテンツを暗号化し，暗号化の鍵 (session 鍵) を S に属するユーザのみが復号できる鍵 (subset 鍵) で暗号化，(4) 暗号化コンテンツ + { 暗号化 session 鍵 } を公衆放送網を通して全ユーザに送信，(5)S に含まれるユーザは，初期に配布された秘密情報から subset 鍵を生成し，session 鍵を復号し暗号化コンテンツを復号する．既存の方法 [6] は，コンテンツ配信者がユーザの秘密情報を管理することで実現する．つまり，ブロードキャスト暗号の安全性は，コンテンツ配信者自身の鍵管理に依存し，同じ配送網及びユーザ復号機を利用して他のコンテンツ配信者が利用できない．つまりフレキシブルなサービスの管理という観点では，ユーザの秘密情報を利用した鍵配信より，ユーザの公開鍵などの公開情報を用いた鍵配信が好ましいといえる．これが公開鍵に基づくブロードキャスト暗号である．しかしながら，公開鍵暗号を用いたブロードキャスト暗号の実現は難しく，従来の素因数分解問題，ECDLP，DLP などでは実現ができず，双線型ディッフィヘルマン問題を利用して公開鍵暗号を利用した現実的なブロードキャスト暗号は，初めて [2] により実現された．

本報告書では，このように新しい問題を解決する双線型 DH 問題について記述するとともに，鍵進化公開鍵暗号系，ブロードキャスト暗号について述べる．

2 準備

2.1 双線型 Diffie-Hellman 問題

$\mathbb{G}_1, \mathbb{G}_2$ を大きな素数 q を位数とする巡回群とし，$\mathbb{G}_1$ は加法的に，$\mathbb{G}_2$ は乗法的に表現する．また $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ を多項式時間で計算可能な非退化な双線型写像 $\hat{e}$ が定義されているとする．すなわち，$\hat{e}(aQ, bR) = \hat{e}(Q, R)^{ab}$ かつ $\hat{e}(G, G) \neq 1 \in \mathbb{Z}_l^*$ をみたす．この双線型写像を用いて，双線型 Diffie-Hellman 問題 (BDHP) が以下のように定義できる．

定義 1 $\mathbb{G}$ を素数位数 q の群 $\mathbb{G} =< G >$ とし，多項式時間で計算可能な非退化な双線型写像 $\hat{e}$ が定義されているとする．すなわち，$\hat{e}(aQ, bR) = \hat{e}(Q, R)^{ab}$ かつ $\hat{e}(G, G) \neq 1 \in \mathbb{Z}_l^*$ をみたす．この時，双線型 *Diffie-Hellman Problem(BDHP)* とは，$< G, aG, bG, cG >$ に対して $\hat{e}(G, G)^{abc}$ を求める問題である．

2.2 ID ベース暗号

ここでは，BDHP を用いた ID ベース暗号の概念について説明する [?]. ID ベース暗号は，ユーザの公開鍵にユーザの名前や住所，電話番号，メールアドレスという任意のビット列を利用する方式で，公開鍵暗号に必要となる公開鍵証明書が不要になる．なお具体的な方式は，前述の BDHP を用いて実現される．

定義 2 *ID* ベース暗号とは，以下の性質を満たす 4 つの確率的多項式時間アルゴリズム (IBE-KGen, IBE-KDer, IBE-Enc, IBE-Dec) の組である．

1. IBE-KGen は，入力のセキュリティパラメータ 1^k に対して，センタ公開鍵と秘密鍵の対，(PK, s) を出力する確率的多項式時間アルゴリズム，$(PK, s) \leftarrow \texttt{IBE-KGen}(1^k)$ である．
2. IBE-KDer は，センタの公開鍵 PK と秘密鍵 s，ユーザ $ID\,t$ に対して，ユーザ $ID\,t$ に対応する秘密鍵 s_t を出力する確率的多項式時間アルゴリズム $s_t \leftarrow \texttt{IBE-KDer}(PK, s, t)$ である．
3. IBE-Enc は，センタの公開鍵 PK，ユーザ $ID\,t$，平文 $m \in \{0,1\}^k$ に対して，暗号文 $c \in \{0,1\}^*$ を出力する確率的多項式時間アルゴリズム $c \leftarrow \texttt{IBE-Enc}(PK, t, m)$ である．
4. IBE-Dec は，暗号文 c，ユーザ $ID\,t$ の秘密鍵 s_t に対して，$m \in \{0,1\}^*$ を出力する確率的多項式時間アルゴリズム $m \leftarrow \texttt{IBE-Dec}(s_t, c)$ である．

2.2.1 階層的 ID ベース暗号

階層的 ID ベース暗号 ([3]) は，ユーザを木構造の各ノードに対応させた ID ベース暗号で，各ノードは子ノードの秘密鍵を生成し，ノードの ID はルートノードまでのノード列となる．ユーザ v を $v = v_0 v_1 \cdots v_t$ と表記し，木のルートを $v_0 = \varepsilon$，v の子ノードは $vv_{t+1} = v_0 \cdots v_t v_{t+1}$ と表記する．また，$v|_k = v_0 \cdots v_k$ である．以下の 4 つの確率的アルゴリズムで構成される．なお具体的な方式は，前述の BDHP を用いて実現される [3].

HIBE-KGen
: セキュリティ・パラメータ 1^k の入力に対し，システムの公開鍵 PK とルートの秘密鍵 SK_ε を生成する確率的多項式時間アルゴリズム，$(PK, SK_\varepsilon) \leftarrow \texttt{HIBE-KGen}(1^k)$ である．

HIBE-KDer
: システムの公開鍵 PK とノード $v = v_1 \cdots v_t$ とその秘密鍵 SK_v の入力に対し，v の子 vv_{t+1} の秘密鍵 $SK_{vv_{t+1}}$ を生成する確率的多項式時間アルゴリズム，$SK_{vv_{t+1}} \leftarrow \texttt{HIBE-KGen}(PK, v, v_{t+1}, SK_v)$ である．

HIBE-Enc
: システムの公開鍵 PK，送信先ノード $v = v_0 \cdots v_t$ とメッセージ m の入力に対し，暗号文 c を出力する確率的多項式時間アルゴリズム $c \leftarrow \texttt{HIBE-Enc}(PK, v, m)$ である．

HIBE-Dec
: システムの公開鍵 PK，ノード v の秘密鍵 SK_v と暗号文 c の入力に対し，メッセージ m を出力する確率的多項式時間アルゴリズム，$m \leftarrow \texttt{HIBE-Dec}(SK_v, c)$ である．

3 鍵進化公開鍵暗号系

本章では，鍵進化暗号の基本的な概念及び安全性のレベルについて述べる．

3.1 鍵露呈に対する安全性

鍵進化暗号では，ユーザの秘密鍵 sk を一ヵ月などの一定の期間毎に更新し，ある期間の秘密鍵の露呈が全期間の秘密鍵の露呈を意味しないように公開鍵暗号を構築する．ここで，対応する公開鍵は変化せずに

そのままであることに注意したい．通常の公開鍵暗号における暗号化が，平文と公開鍵の入力でなされるのに対し，鍵進化暗号の暗号化は，平文と公開鍵，期間 t の3つの入力でなされる．鍵進化暗号において考慮する安全性は次の二つになる．

Forward security: ある期間 t_0 の秘密鍵 sk_{t0} が露呈しても，その期間より前の秘密鍵 sk_t ($t < t_0$) は露呈しない．

Future security: ある期間 t_0 の秘密鍵 sk_{t0} が露呈しても，その期間より後の秘密鍵 sk_t ($t > t_0$) は露呈しない．

鍵進化暗号は，その形態から大きく2つに分けられる．1つはユーザのみの1エンティティで実現される方式であり，もう一つは2エンティティ，ユーザとベースにより実現される方式である．ユーザの秘密鍵を2つに分離しない限り，future security は実現できないことに注意したい．耐鍵侵入暗号は，その機能により，forward-security を実現する forward secure 暗号，ユーザ秘密鍵を2つにわけ，future-security も考慮した Key-insulated 暗号，Intrusion-resilient 暗号の3つに分けられる．以降，Forward secure 暗号と Intrusion-resilient 暗号のそれぞれの原理及び安全性の定義を述べる．

3.2 Forward-secure 暗号

Forward-secure 暗号は，1エンティティで実現される鍵進化暗号であり，Forward security を実現する．つまり，ある期間 t_0 の秘密鍵 sk_{t0} の露呈は，その期間より前の $t < t_0$ の秘密鍵の sk_t の安全性に影響しない．

定義 3 ([1]) *Forward-secure* 暗号とは，以下の性質を満たす *4* つの確率的多項式時間アルゴリズム (FSKG, FSUpd, FSEnc, FSDec) の組である．

1. FSKG は，入力のセキュリティパラメータ 1^k に対して，ユーザ公開鍵，ユーザの秘密鍵の初期値 (P, sk_0) を出力する確率的多項式時間アルゴリズム $(P, sk_0) \leftarrow \mathtt{FSKG}(1^k)$ である．
2. FSUpd は，時間 t のユーザの秘密鍵 sk_t に対して時間 $t+1$ のユーザ秘密鍵 sk_{t+1} を出力する確率的多項式時間アルゴリズム $sk_{t+1} \leftarrow \mathtt{FSUpd}(sk_t, t)$ である．
3. FSEnc は，ユーザ公開鍵 P, 期間 t, 平文 $m \in \{0,1\}^k$ に対して，暗号文 $c \in \{0,1\}^*$ を出力する確率的多項式時間アルゴリズム $c \leftarrow \mathtt{FSEnc}(P, t, m)$ である．
4. FSDec は，暗号文 c, 期間 t のユーザの秘密鍵 sk_t に対して，$m \in \{0,1\}^*$ を出力する確率的多項式時間アルゴリズム $m \leftarrow \mathtt{FSDec}(sk_t, c)$ である．

次に安全性の定義を与える．Forward-secure 暗号の安全性の定義は公開鍵暗号の定義とほぼ同様に与えられる．公開鍵暗号との違いは，Forward-secure 暗号では，各期間 t に対応する秘密鍵も攻撃者が入手（鍵オラクル）できることである．つまり，攻撃者には復号オラクル ($O_{\mathtt{FSDec}}$) と鍵オラクル ($O_{\mathtt{FSsec}}$) が与えられ，適応的に対応する平文や鍵が入手できる．以下で，復号オラクルも与えられるより強い安全性に関する定義のみを与える．

定義 4 (FS-IND-CCA) *Forward-secure* 暗号 (FSKG, FSUpd, FSEnc, FSDec) に対して，アルゴリズム A は，*3* つのオラクルが与えられる．

- 復号オラクル $O_{\mathtt{fsDec}}$: 任意の暗号文と期間 t の入力に対し，期間 t でのユーザ秘密鍵の復号結果を出力

する.
- 鍵オラクル $O_{\mathtt{fsSec}}$: 任意の期間 t に対し，期間 t のユーザ秘密鍵を出力する.
- チャレンジオラクル $O_{\mathtt{LR}}$: 長さの等しい2つの平文 m_0 と m_1 の入力に対しそのどちらかの暗号文 $c^* = \mathtt{FSEnc}(P, t, m_b)$ $(b = 0, 1)$ を出力する.

A は3つのオラクルが与えられ，次の過程を実行する多項式時間アルゴリズムである

初期設定： 鍵生成アルゴリズム FSKG を実行させて，(P, sk_0) を出力させ，A に P を入力する.

攻撃： A は，復号オラクル $O_{\mathtt{fsDec}}$，鍵オラクル $O_{\mathtt{fsSec}}$，チャレンジオラクル $O_{\mathtt{LR}}$ に適応的に質問を繰り返しその回答を入手する．但し，以下の制限事項をまもる.
- $O_{\mathtt{LR}}$ には一度しか質問しない.
- $O_{\mathtt{LR}}$ の出力 c^* を $O_{\mathtt{fsDec}}$ には質問しない.
- $O_{\mathtt{LR}}$ の入力 t^* に対して $t^* \geq t$ なる t を $O_{\mathtt{fsDec}}$ には質問しない.

出力： A は，$b' = \{0, 1\}$ を出力し，$b = b'$ であれば A は攻撃に成功したとする.

Forward-secure 暗号が適応的選択暗号文攻撃に対して識別不可能性を満たす（*IND-FS-CCA*）とは，上記で定義された任意の多項式時間アルゴリズム A の成功確率がセキュリティパラメータに対して無視できるぐらい小さいことをいう.

3.3 Intrusion-resilient 暗号

Intrusion-resilient 暗号は，Key-isulated 暗号の安全性を強化した概念であり，ユーザとベースの両方を攻撃されても，同じ期間が攻撃されない限り，ユーザの秘密鍵の安全性は保たれる．つまり，ある期間 t_0 の秘密鍵 sk_0 の露呈は，その期間より前 $t < t_0$ (forward-security), そしてその期間より後 $t > t_0$ (future security) の秘密鍵の安全性に影響しない．さらに，ユーザとベースの両方が同じ期間 t_0 において攻撃されても，その期間より前 $t < t_0$ (forward-security) の秘密鍵 sk_t の安全性に影響しない.

Intrusion-resielient 暗号では，ベースとユーザの同時期の鍵露呈攻撃に対する安全性強化のために リフレッシュ r というパラメータが追加される．リフレッシュ r はベースとユーザの鍵を更新するためのパラメータであるが，1つの期間 t 内を分割するパラメータとなり，暗号化には関与しない．つまり，データの暗号化には，期間情報 t は必要だがリフレッシュ情報 r は不要である．例えば，期間 t が1日に対応し リフレッシュ r が1時間に対応するとすると，攻撃者が同じ月日かつ同じ時間のユーザとベースの鍵を入手しない限り，future security も forward security が満たされることになる．一方，暗号化は，$\mathtt{IREnc}(P, 1900/12/31, m)$ というように日単位までの情報だけで実現できる．端的にいえば，簡単な負荷で安全性を強化するパラメータといえる.

ここでは，Intrusion-resilient 暗号の厳密な定義を与える.

定義 5 ([5]) *Intrusion-resilient* 暗号とは，以下の性質を満たす 7 つの確率的多項式時間アルゴリズム (IRKG, IRUserUpd, IRBaseUpd, IRUserRef, IRBaseRef, IREnc, IRDec) の組である.

1. IRKG は，入力のセキュリティパラメータ 1^k に対して，公開鍵，ベース鍵，ユーザの秘密鍵の初期値 $(P, skb_{0.0}, sk_{0.0})$ を出力する確率的多項式時間アルゴリズム $(P, skb_{0.0}, sk_{0.0}) \leftarrow \mathtt{IRKG}(1^k)$ である.

2. IRBaseUpd は，時間 t，リフレッシュ r のベース秘密鍵 $skb_{t.r}$ に対して，時間 $t+1$ のベース鍵 $skb_{t+1.0}$，ユーザ秘密鍵の更新データ sku_t を出力する確率的多項式時間アルゴリズム $(skb_{t+1.0}, sku_t) \leftarrow \mathtt{IRBaseUpd}(skb_{t.r})$ である．
3. IRUserUpd は，時間 t，リフレッシュ r のユーザの秘密鍵 $sk_{t.r}$ とベースからの更新データ sku_t に対して時間 $t+1$ のユーザ秘密鍵 $sk_{t+1.0}$ を出力する確率的多項式時間アルゴリズム $sk_{t+1.0} \leftarrow \mathtt{IRUserUpd}(sk_{t.r}, sku_t)$ である．
4. IRBaseRef は，時間 t，リフレッシュ r のベース鍵 $skb_{t.r}$ に対して，時間 t，リフレッシュ $r+1$ のベース鍵 $skb_{t.r+1}$，鍵リフレッシュデータ $skr_{t.r}$ を出力する確率的多項式時間アルゴリズム $(skb_{t.r+1}, skr_{t.r}) \leftarrow \mathtt{IRBaseUpd}(skb_{t.r})$ である．
5. IRUserRef は，時間 t，リフレッシュ r のユーザの秘密鍵 $sk_{t.r}$ とベースからの鍵リフレッシュデータ $skr_{t.r}$ に対して，時間 t，リフレッシュ $r+1$ のユーザ秘密鍵 $sk_{t+1.0}$ を出力する確率的多項式時間アルゴリズム $sk_{t.r+1} \leftarrow \mathtt{KIUserUpd}(sk_{t.r}, skr_{t.r})$ である．
6. IREnc は，ユーザの公開鍵 P，期間 t，平文 $m \in \{0,1\}^k$ に対して，暗号文 $c \in \{0,1\}^*$ を出力する確率的多項式時間アルゴリズム $c \leftarrow \mathtt{IREnc}(P,t,m)$ である．
7. IRDec は，暗号文 c，期間 t，リフレッシュ r のユーザの秘密鍵 $sk_{t.r}$ に対して，$m \in \{0,1\}^*$ を出力する確率的多項式時間アルゴリズム $m \leftarrow \mathtt{IRDec}(sk_{t.r}, c)$ である．

Intrusion-resilient 暗号の安全性の定義は，他の鍵進化暗号の定義とほぼ同様に与えられる．つまり，攻撃者には復号オラクル ($O_{\mathtt{IRDec}}$) と鍵オラクル ($O_{\mathtt{IRsec}}$) が与えられ，適応的に対応する平文や鍵が入手できる．鍵オラクルでは，ユーザ秘密鍵，ベース鍵，更新データ，リフレッシュデータのすべてにアクセスできるので，期間 t，リフレッシュ r に加えてオラクル情報 (ユーザ秘密鍵 sk，ベース鍵 bk，更新データ u，リフレッシュデータ r) も記述する．鍵オラクルへの質問は，Intrusion-resilient 暗号での鍵保管条件と照らし合わせて以下を仮定する．

鍵オラクルの質問順序

◇ (sk, $t'.r'$) を質問した後に (sk, $t.r$) ($t' > t$ あるいは $t' = t$ かつ $r' > r$) を質問しない (ユーザ秘密鍵の更新（リフレッシュ）後，更新（リフレッシュ）前のユーザ秘密鍵は消去)．

◇ (bk, $t'.r'$) を質問した後に (bk, $t.r$) ($t' > t$ あるいは $t' = t$ かつ $r' > r$) を質問しない (ベース鍵更新（リフレッシュ）後，更新（リフレッシュ）前のベース鍵は消去)．

◇ (r, $t'.r'$) を質問した後に (bk, $t.r$) ($t' > t$ あるいは $t' = t$ かつ $r' \geq r$) を質問しない（リフレッシュデータ送信後，リフレッシュ前のベース鍵は消去)．

◇ (u, t') を質問した後に (bk, $t.r$) ($t' \geq t$) を質問しない（更新データ送信後，更新前のベース鍵は消去)．

また鍵オラクルへの質問により，明らかにユーザ秘密鍵 $sk_{t.r}$ が露呈する場合がある．例えば，ユーザ秘密鍵 $sk_{t.r-1}$ とリフレッシュデータ $skr_{t.r-1}$ を入手すれば明らかに $sk_{t.r}$ は計算できる．このような質問により鍵が露呈することが自明な場合を以下に記述する．

鍵露呈質問

Que を $O_{\mathtt{IRSec}}$ への質問の集合とする．このとき $sk_{t.r}$ が露呈する自明な質問は以下で定義される．

- (sk, $t.r$) $\in Que$.
- $r > 1$ のとき (r, $t.(r-1)$) $\in Que$ かつ $sk_{t.(r-1)}$ は鍵露呈.
- $r = 1$ のとき (u, $t-1$) $\in Que$ かつ $sk_{(t-1).R}$ が鍵露呈.

- $r < R$ のとき $(\mathrm{r}, t.r) \in Que$, かつ $sk_{t.r+1}$ は鍵露呈.

定義 6 (IR-IND-CCA) *Intrusion-resilient* 暗号 (IRKG, IRUserUpd, IRBaseUpd, IRUserRef, IRBaseRef, IREnc, IRDec) に対して，アルゴリズム A は 3 つのオラクルが与えられる.
○ 復号オラクル $O_{\mathtt{IRDec}}$：任意の暗号文と期間 t の入力に対し，期間 t，リフレッシュ r のユーザ秘密鍵での復号結果を出力する.
○ 鍵オラクル $O_{\mathtt{IRSec}}$: 任意の期間 t，リフレッシュ r，オラクル情報 (ユーザ秘密鍵 sk，ベース鍵 bk，更新データ u，リフレッシュデータ r) の入力に対し，それぞれ対応するユーザ秘密鍵，ベース鍵，更新データ，リフレッシュデータを出力する.
○ チャレンジオラクル $O_{\mathtt{LR}}$：長さの等しい 2 つの平文 m_0 と m_1 の入力に対し，そのどちらかの暗号文 $c^* = \mathtt{IREnc}(P, t, m_b)$ $(b = 0, 1)$ を出力する.
初期設定： 鍵生成アルゴリズム IRKG を実行させて，(P, skb_0, sk_0) を出力させ，A に P を入力する.

攻撃： A は，復号オラクル $O_{\mathtt{IRDec}}$ と，鍵オラクル $O_{\mathtt{IRSec}}$ チャレンジオラクル $O_{\mathtt{LR}}$ に適応的に質問を繰り返しその回答を入手する. 但し，以下の制限事項をまもる.

- ○ $O_{\mathtt{LR}}$ には一度しか質問しない.
- ○ $O_{\mathtt{LR}}$ の出力 c^* を $O_{\mathtt{IRDec}}$ には質問しない.
- ○ $O_{\mathtt{LR}}$ の入力 t^* と任意のリフレッシュ r に対して，$sk_{t*.r}$ の鍵露呈質問をしない.
- ○ $O_{\mathtt{IRSec}}$ に対する質問は順序を遵守.

出力： A は，$b' = \{0, 1\}$ を出力し，$b = b'$ であれば A は攻撃に成功したとする.

Intrusion-resilient 暗号が適応的選択暗号文攻撃に対して識別不可能性を満たす (*IND-IR-CCA*) とは，上記で定義された任意の多項式時間アルゴリズム A の成功確率がセキュリティパラメータに対して無視できるぐらい小さいことをいう.

3.4 残される課題

本章では，Intrusion-resilient 暗号において，現在残される課題について述べる. 鍵進化暗号系の効率は，鍵更新期間 T のオーダで計る. 表は，既存の Forward secure 暗号と Intrusion resilient 暗号の効率を表す.

表 1: 既存の鍵進化公開鍵暗号系の効率

	Forward secure 暗号 [1]	Intrusion-resilient 暗号 [4]
鍵生成時間	$O(1)$	$O(1)$
暗号・復号時間	$O(\log T)$	$O(\log T)$
鍵更新時間	$O(1)$	$O(\log T)$
暗号文長	$O(\log T)$	$O(\log T)$
秘密鍵長	$O(\log T)$	$O(\log T^2)$
公開鍵長	$O(1)$	$O(1)$

表より，Forward secure 暗号には，多項式時間オーダで可能な効率の良い方法があるのに対し，Intrusion-resitlient 暗号には存在しない．このため，効率的かつ安全な Intrusion-resitlient 暗号の構築は残る課題といえる．

4 ブロードキャスト暗号

本章では，ブロードキャスト暗号の概念と公開鍵ブロードキャスト暗号について述べる．

4.1 ブロードキャスト暗号の枠組み

ブロードキャスト暗号の基本的な仕組みについて述べる．システム内の全ユーザの集合を $\mathcal{N}$，復号を許可しないユーザの集合を $\mathcal{R}$，それぞれのユーザ数を $|\mathcal{N}| = n$, $|\mathcal{R}| = r$ とする．センターは，メッセージ m を $\mathcal{N} \setminus \mathcal{R}$ に属するユーザ (privileged receiver) が正しく復号でき，$\mathcal{R}$ に属するユーザ (revoked receiver) が結託しても復号できないように配信するために，以下の手順を行う．

- Initialization センターは各ユーザ u に復号に必要な秘密鍵 SK_u を配布する．
- Covering algorithm センターは privileged receivers からなる集合を独立な部分集合，$\mathcal{N} \setminus \mathcal{R} = \cup S_{i_j}$ に分割し，各集合 S_{i_j} に対応する秘密鍵 $K_{S_{i_j}}$ を用いて，セッション鍵 K を暗号化し，

$$\langle \{S_{i_j}\}, \{E_1(K_{S_{i_j}}, K)\}, E_2(K, M) \rangle$$

  を全ユーザに送信する．ここで，$E_i(K, M)$ $(i = 1, 2)$ は鍵 K による M の暗号化を意味し，$\{E(K_{S_{i_j}}, K)\}$ のサイズが transmission rate を決定する．
- Decryption algorithm $\mathcal{N} \setminus \mathcal{R}$ に含まれる任意のユーザ u は，SK_u より自分のカバー集合 S_{i_j} の秘密鍵 $K_{S_{i_j}}$ を導出し，M を復号する．

なお，カバー集合全体を $\mathcal{S} = \{Si\}$ と表す．(秘密鍵に基づく) ブロードキャスト暗号のスキームは，transmission rate，ユーザの秘密鍵サイズ，ユーザの subset 鍵の導出にかかる計算量の観点で評価する．なお，公開鍵に基づく場合，これに加えて公開鍵のサイズでも評価する．

4.2 既存方法

既存のブロードキャスト暗号は，ユーザを完全二分木の葉に配置し，完全木の各ノードに秘密鍵を設定し，各ユーザが自分を含む全ての部分集合に対応する秘密鍵が導出できるようにノードの秘密鍵を配布する．言うまでもなく，ユーザに配布する鍵の方法は，正規ユーザのカバーリングに依存する．これまで最もよく利用される方法は，Complete Subtree 法 (CS 法) と Subset Difference 法 (SD 法) [6] である．CS 法は，正規ユーザのみを葉にもつ最大部分完全 2 分木により正規ユーザをカバーする．この結果，正規ユーザは自分をカバーする任意の部分完全 2 分木の頂点に対応する秘密鍵を保持すれば，どのようなユーザが削除されても任意の暗号文を復号できる．つまり，各ユーザは自分が配置された葉から根までの経路上の節点に対応する鍵を秘密鍵として保持すれば復号可能となる．

表 2: 既存方法の性能

	CS [6]	SD [6]	公開鍵 SD [2]	公開鍵 CS [2]
transmission rate	$r\log\frac{N}{r}$	$2r-1$	$(2r-1)\lvert C_{\mathtt{HIBE}}\rvert^{\dagger}$	$r\log\frac{N}{r}$
ユーザの秘密鍵量	$O(\log N)$	$O(\frac{1}{2}\log^2 N)$	$O(\lvert SK_{\mathtt{HIBE}}\rvert\log^N)^{\dagger}$	$O(\log N)$
$\lvert S\rvert$	$2N-1$	$N\log N$	$N\log N$	$2N-1$
ユーザ当たりのカバー集合	$\log N$	$O(2N)$	$O(2N)$	$\log N$
公開鍵サイズ	—	—	$O(1)$	$O(1)$

†: $|C_{\mathtt{HIBE}}|$, $|SK_{\mathtt{HIBE}}|$ は階層的 ID ベース暗号の暗号文のサイズとノード鍵のサイズである.

一方, SD 法は直感的には, CS 法の正規ユーザの隣接するカバリングを一つに統合し, ヘッダ数を抑える方法である. つまり, 正規ユーザは二つの部分完全 2 分木の差集合で覆われることになる. つまり, 二つの節点を v^a と v^b (v^a は v^b の祖先) とするとき, v_a を根とする部分完全 2 分木の葉から v_b を根とする部分完全 2 分木の葉を除いたものを一つの部分集合 S_{v^a,v^b} とし, 正規ユーザはこの差集合 $S_{v^a}-S_{v^b}$ によってカバーされる. なお二つの節点のうち祖先である v^a を primary root, 子孫である v^b を secondary node と呼ぶ. この結果, 正規ユーザ u は二つの頂点情報 v^a と v^b から一意的に導ける秘密情報 $K_{v^a.v^b}$ をもつ必要がある. なお u から root までの path を ρ_u, ρ_u にぶら下がる兄弟 node の集合を $\{v^{\rho_u}\}$ と記述するとき,

$$S_{v^a,v^b} \ni u \quad \Leftrightarrow \quad \rho_u \ni v^a$$
$$\{v^{\rho_u}\} \text{ のある元が } v^b \text{ の祖先}$$

になる. そこで SD 法では primary root v^a に対して秘密鍵 SK_{v^a,v^a} を設定し, 鍵生成関数 BE-KDer を用いて v_a を primary root とする全ての差集合 S_{v^a,v^b} に対応する秘密鍵 $K_{v^a.v^b}$ を生成する.

- BE-KDer$(v^a, v^b, SK_{v^a,v^b}, PK) = (SK_{v^a.v^b0}, SK_{v^a.v^b1}, K_{v^a.v^b})$
  primary node v^a, v^a の子孫のノード v^b ($b=a$ の場合も含む), v^b の秘密鍵 SK_{v^a,v^b}, 及び公開鍵 PK の入力に対して, v^b の子供のノード v^b0 と v^b1 の鍵 SK_{v^a,v^b0} と $SK_{v^a.v^b1}$ 及び差集合 S_{v^a,v^b} の鍵 K_{v^a,v^b} を出力する.

この結果, $S_{v^a,v^b} \ni u$ が自分をカバーするすべての部分集合 S_{v^a,v^b} の鍵を復元するには, u から root までの path ρ_u 上の任意の node を primary node とする差集合の鍵が生成できるとよい. つまり, ρ_u の任意の node v^a と u との path ρ_{u,v^a} にぶら下がる兄弟 node の集合を $\{v^{\rho_u,v^a}\}$ と記述するとき, $\{v^{\rho_u,v^a}\} \ni \forall v^b$ を secondary node とする node 鍵 $\{K_{v^a.v^b}\}$ を秘密鍵として保持すればよい. このとき, 各ユーザが保管する鍵の大きさは, $O(\log^2(N))$ となる. なお実際にユーザがカバーされる可能性のある集合は $O(N)$ より, この鍵生成関数の存在がデータの削減に大きい影響を与える.

各方法の性能を表 2 に示す.

4.3 公開鍵ブロードキャスト暗号

ブロードキャスト暗号に公開鍵を導入する際の問題点は，公開鍵の大きさである．表 2 により，CS 法，SD 法を単純に公開鍵暗号で構成する場合，S に含まれる集合毎に公開鍵を設定するので，$2N-1$，$N\log N$ の公開鍵が必要になる．一方，秘密鍵に関しては，CS 法の場合，ユーザの保管する秘密鍵量とユーザあたりのカバー集合の個数が一致するので，公開鍵に対応する秘密鍵をユーザが保管するとよい．しかし，SD 法の場合，ユーザあたりのカバー集合の個数が $O(2N)$ であるのに対し，秘密鍵量を $\log^2 N$ と抑えているため，秘密鍵導出関数も構成する必要がある．公開鍵 CS 法に関しては，ID ベース暗号を用いることで，ユーザを配置する完全 2 分木の各ノードに対してノード名を公開鍵とする秘密鍵を対応させることで，公開鍵サイズを $2N-1$ から $O(1)$ に減らすことができる．一方，SD 法に関しては，鍵導出関数を構成する必要があり，ID ベース暗号を単純に適用することはできない．

Dodis ら [2] は，階層的 ID ベース暗号の概念を利用し，ユーザの鍵保管量は $O(\log^2 N)$，公開鍵量は $O(1)$ の公開鍵暗号に基づく SD 法を実現した．方法は非常にシンプルである．SD 法を公開鍵暗号で実現する最大のポイントは，BE-KDer をどのように設定するかということである．そこで階層的 ID ベース暗号を適用する多分木を SD 法の primary node v^a 毎の差集合，すなわち $\{ S_{v^a,v^b} \}$ で構成する．ここで，S_{v^a,v^a} を便宜上，$\{ S_{v^a,v^b} \}$ のルートに設定し，そのノード鍵 K_{v^a,v^a} を設定する．つまり v^a を primary node とする差集合 S_{v^a,v^b} の鍵を生成するには

- ノード鍵 K_{v^a,v^b} の生成
- 差集合の鍵 SK_{v^a,v^b} の生成

の二つのステップが必要である．これを以下のように階層的 ID ベース暗号の鍵導出関数を再帰的に利用して生成する．v^a を v^b の親ノードとし $v^b = v^a v_{t+1}$ とする．図 はユーザを配置した二分木を表し，図 は差集合の鍵を階層的 ID ベース暗号で導出するための多分木である．

$$SK_{v^a,v^b} \leftarrow \texttt{HIBE-KGen}(PK, v^a, v_{t+1}, SK_{v^a,v^a})$$
$$K_{v^a,v^b} \leftarrow \texttt{HIBE-KGen}(PK, v^b, 2, SK_{v^a,v^b})$$

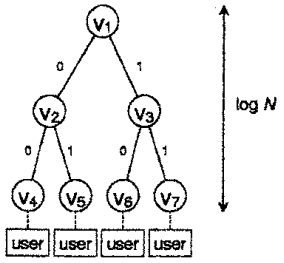

図 1: User Tree

DF 法は階層的 ID ベース暗号の概念のみを適用するため，効率的な階層的 ID ベース暗号が存在すれば，SD 法と同じ効率で実現できる．しかし，現存の階層的 ID ベース暗号 [3] は，HIBE-KDer，HIBE-Enc，HIBE-Dec，暗号文，ユーザ（ノード鍵）の大きさが，ユーザ数 N の場合，それぞれ $O(1)$, $O(\log N)$, $O(\log N)$, $O(\log N)$, $O(\log N)$ となり，CS 法よりも効率が落ちる．

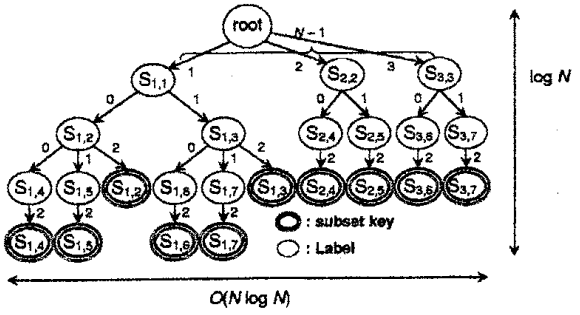

図 2: Key Derivation Tree

参考文献

[1] R. Canetti, S. Halevi, and J. Katz. "A forward-secure public-key encryption scheme." *Advances in Cryptology — Eurocrypt 2003*, Lecture Notes in Computer Science, **2656**, Springer-Verlag, 2003.

[2] Y. Dodis and N. Fazio: "Public Key Broadcast Encryption for Stateless Receivers", *proceeding of ACM DRM '02*, LNCS **2696**, Springer-Verlag, pp.61–80, 2002.

[3] C. Gentry and A. Silverberg: "Hierarchical ID-Based Cryptography" *Advances in Cryptology-Asiacrypt 2002*, LNCS **2501**(2002), Springer-Verlag, 548–566.

[4] Y. Dodis, M. Franklin, J. Katz, A. Miyaji and M. Yung, "Intrusion-Resilient Public-Key Encryption", *RSA-CT 2003*, Lecture Notes in Computer Science, **2612**(2003), Springer-Verlag, 19-32.

[5] Y. Dodis, M. Franklin, J. Katz, A. Miyaji and M. Yung, "Generic construction for Intrusion-resilient public-key encryption." *RSA — Cryptographers' Track 2004*, LNCS **2964**(2004), Springer-Verlag, 81-98.

[6] D. Naor, M. Naor and J. Lotspiech: "Revocation and Tracing Scheme for Stateless Receivers", *Advances in Cryptology-CRYPTO 2001*, LNCS **2139**, Springer-Verlag, pp.41–62, 2001.

[7] J. H. Silverman, *The Arithmetic of Elliptic Curves*, GTM106, Springer-Verlag, New York, 1986.