社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE

Re-Encryption Schemes によるデータ格納方式
－楕円曲線上の等分点を利用した結託攻撃からの安全性についての検討－

扇 裕和†

†株式会社 メビウス 〒220-0073 横浜市西区岡野 1-14-1 大宗岡野町ビル
E-mail: †h_ougi@mebius.co.jp,

あらまし Re-Encryption Scheme によるデータの格納方式は、一つの鍵で作成された暗号文を、異なる他の鍵で復号できるように鍵を変換し、re-encryption key を与える方式であり、効率の良いファイル暗号方式として期待することができる。Ateniese *et al.*(2006)[1]に提案されている方式は、暗号文の作成者以外からの re-encryption key の生成を回避するため、proxy による演算処理を必要としている。本稿は、楕円曲線上の s-等分点を利用して、proxy を必要としないシステム構築方式について検討し、re-encryption key の結託攻撃からの安全性確保について検討したものである。

キーワード 暗号，公開鍵，結託攻撃，楕円曲線

Re-Encryption Schemes with Applications to Data Storage
－Investigation of achieving preventative from collusion operating over the torsion subgroup in an elliptic curve－

Hirokazu OUGI†

† Mebius Corporation 1-14-1 Okano, Nishi-ku, Yokohama-shi Kanagawa, 220-0073 Japan
E-mail: †h_ougi@mebius.cojp,

Abstract A method of managing encrypted file system using Re-encryption Scheme allows to transform a ciphertext computed under Alice's public key into one that can be opened by Bob's secret key. There are many useful applications of this primitive. This paper presents new Re-Encryption Scheme that a set of colluding delegatees provided with re-encryption key can not re-delegate decryption rights. This Scheme operates over the torsion subgroup in an elliptic curve.

Keyword Re-Encryption, Public-key, Collusion, Elliptic-curve

1. まえがき

Re-Encryption Scheme は、公開鍵暗号方式において、ひとつの秘密鍵で暗号化された暗号文を、他の秘密鍵で復号できるよう暗号文に対して変換を与える方式である[1][2][3]。もとの平文を操作することなく、暗号文を変換でき、また他の第３者による暗号文の変換ができないような方式であり、様々の有効な応用を期待することができる。

たとえば、作成された暗号文の秘密鍵の所有者が不在、もしくは秘密鍵を紛失した場合に、復号する権限を委譲された者が、当該暗号文を復号して利用することが可能である。

本稿は、公開鍵暗号方式による Re-Encryption Scheme のシステム構築方式について検討したものである。本提案方式の暗号処理手順では、まずデータを提供する送信元ユーザの持つ公開鍵で暗号文を作成する。次に、復号する権限を委譲する相手ユーザの公開鍵を取得し、自身の所有する秘密鍵を用い、当該公開鍵を変換して re-encryption key を生成し登録する。相手ユーザは、自身の秘密鍵と re-encryption key を用いて、当該暗号文を復号し利用する。

効率よく安全に利用するためには、

(1) re-encryption key は一方向性を所有すること。

即ち、相手ユーザは、送信元ユーザの暗号文を復号できるが、送信元ユーザは、登録された re-encryption key からは相手ユーザが作成する暗号文を復号することはできない。

(2) re-encryption key は結託攻撃に対して安全であること。

即ち、送信元ユーザ以外に、当該暗号を復号する re-encryption key を生成することはできない。

等の特性などが要求される。

従来、このようなセキュリティ上の安全性の要求を満足するには、proxy による演算処理を必要とするが、

本稿は、楕円曲線上の s-等分点を利用し、(trusted)proxy 処理を必要としない、結託攻撃に対して安全な re-encryption key の構成について検討したものである。

1.1. 背景

公開鍵による鍵共有の方式としては、あまりにも有名な Diffie-Hellman[4]の方式がある。暗号文の復号権限の委譲に関して、最初に満保、岡本[5]により提案された。1998 年、Blaze, Bleumer, Strauss[2] は、「atomic proxy cryptography」という概念を提案し、平文を処理することなく、暗号文を変換する atomic proxy function について検討した。この方式は、ElGamal 暗号の方式に基づいている。Diffie-Hellman Problem を基礎をとする暗号方式の Chosen-Ciphertext Attack 対する安全性確保については、藤崎、岡本[6]に続いて、Abdalla, Bellare, Rogaway による DHIES[7]の提案がなされた。一方、weil-paring を利用した暗号方式については、2001 年、光成、境、笠原[8]による提案にひき続いて、Boneh, Franklin[9] による IBE(Identity-based encryption)の提案がなされた。

Re-Encryption Scheme に関し、Dodis, Ivan[3]は、Elgamal, RSA, および IBE の scheme において、unidirectional proxy encryption の機能を実現した。次に、2006 年、Ateniese, Fu, Green, Hohenberger[1]は、ファイル暗号としての検討を実施するとともに、Tate-paring を用いて、Collusion-safe の機能、即ち、proxy と通信相手ユーザが結託しても、送信相手の秘密鍵が導出不可とする機能を可能とした。しかし、Non-transferable の機能、即ち、re-encryiption key に対して、送信元ユーザ以外のユーザが結託しても、送信元ユーザが登録した key 以外の re-encryiption key の生成が不可能となる機能については、提供していない。従って、Ateniese の提案する scheme を運用する場合、信頼できる第 3 者によるアクセス管理と、re-encryiption key に関する (trusted)proxy 演算処理が必要となる。

1.2. 本稿の検討内容

本稿は、Ateniese[1]の提案する scheme に残された Non-transferable の機能を実現する方式について検討したものである。Ateniese は、Tate-paring という Bilinear Map を用いて Re-Encryption Scheme を構成したが、本稿では、この構成方式をもとに楕円曲線上の s-等分点を用い、この Scheme の改良を図った。楕円曲線上の離散対数問題において、素数位数 s の生成元を Go、整数 $a \in Z_s^*$ とし、Q = a・Go（・ドット；楕円曲線上の加法演算）とすると、{a、Go} から容易に Q を計算できるが、逆に Q から a を多項式時間で求めることは、困難とされている。このことは、楕円曲線上の s-等分点においても成立する。楕円曲線上の s-等分点の 2 組の素数位数 s の生成元を {Go, Ge}、整数 $a,b \in Z_s^*$ とし、Q = a・Go + b・Ge とすると、{Go, Ge, a, b }から Q を計算することは可能であるが、{Q, Go, Ge} から {a, b} を多項式時間内に求めることは、困難と考えることができる。

ここで、$a, u \in Z_s^*$, Q = (au)・Go = a・Po, (Po=u・Go) とし、{Q, a} を既知、Po を未知とした場合、a の Z_s^* の逆元を c とすると、

c・Q = c・(au・Go) = (cau)・Go = u・Go = Po

となり、{Q, a} から Po を求めることは可能である。

ところが、楕円曲線上の s-等分点を対象とした場合、Q = (au)・Go + (bv)・Ge= a・Po + b・Pe; $a, u, b, v \in Z_s^*$, (Po=u・Go, Pe=v・Ge)とし、{Q, a, b} を既知、{Po, Pe}を未知とした場合、{Q, a, b} から、多項式時間内に{Po, Pe}を求めることは、困難と考えられる。

本稿では、楕円曲線上の s-等分点上に、re-encryption key を構成し、この計算量的困難性を利用して、Non-transferable の機能の導入を検討した。

まず、ElGamal 暗号をベースに、送信側の公開鍵で、データを暗号化し、通信相手ユーザの秘密鍵で復号する簡単な方式について検討する。次にこれらの方式を整理して、Ateniese[1] の方式の改良を検討し、Re-Encryption Scheme に設定する機能について述べる。そしてこの機能を楕円曲線上の s-等分点上に実装する方式について述べる。また、暗号化および復号方式について、Ateniese の提案した方式を改良して得られた、re-encryption-key の Non-transferable 機能、即ち結託攻撃からの安全性について述べる。結託攻撃からの安全性が確保できれば、re-encryotion key に関して、(trusted)proxy 演算処理を不必要とすることが可能であり、本提案方式は、登録された、re-encryotion key 以外の key 生成を不可能とする方式となった。さらに、ユーザの結託による、新たな復号用の秘密鍵と re-encryption key との組合せの、生成が防止可能な方式については、付録に参考として検討内容を示した。

本提案方式は、ElGamal 暗号をベースに構築したものであり、Abdalla, Bellare, Rogaway [7]の方式に従えば、Chosen-Ciphertext Attack に対する安全性確保は可能と考えられる。また、re-encryption key を利用した署名方式の検討も必要と考えられるが、今後の課題とした。

2. 公開鍵で暗号化する方式

一般的な Re-Encryption Scheme の構成を図 1 に示す。

暗号化システムの運用手順は以下の通りである。

(i) Alice は自分自身の公開鍵（もしくは秘密鍵）で plaintext m を暗号化し ciphertext c を作成する。

(ii) Alice は Bob の公開鍵を取得し、次に自身の秘密鍵を用いて当該公開鍵を変換して、re-encryption-key を生成し登録する。

(iii) Bob は、re-encryption-key を取得し、自身の秘密鍵

により、ciphertext c を復号し plaintext m を取得する。

(vi) Access Control Server は、ユーザ認証等により、ciphertext, re-encrtption key の運用管理を実施する。

Re-Encryption Scheme に関しては、様々の方式が提案されている。まず、Elgamal 暗号をベースに簡単な 2 通りの方式について検討する。

ここで、位数 s の巡回群を G, 生成元を g とし、

Alice の key-pair を　$pka = x_a$ ($x_a = g^a$)

$ska = a \in Fs$

Bob の key-pair を　$pka = x_b$ ($x_b = g^b$)

$ska = b \in Fs$

とする。

2.1. ElGamal 乱数暗号化方式

(1) 暗号化（Enc）

$Enc(pka, m) = \{ g^r, mx_a^r\} = c$　(r; random number)

$Enc(pkb, r) = \{ g^k, rx_b^k\} = r_{A-B}$　(k; random number)

(2) 復号（Dec）

$Dec(skb, r_{A-B}) = rx_b^k/((g^k)^b) = r$

$Dec(ska, r, c) = mx_a^r/((g^r)^a) = m$

2.2. ElGamal-Re-Encryption 方式(typical)

(1) 暗号化(Enc)

$Enc(pka, m) = \{mg^r, x_a^r\} = c$ (r, random number)

・Re-encryption key の生成(R)

$R(ska, skb) = b/a = r_{A-B}$

(generated by trusted third party)

(2) 復号(Dec)

・proxy 演算処理(S)

$S(r_{A-B}, c) = (x_a^r)^{r_{A-B}} = x_b^r$

(calculated by trusted proxy)

$Dec(x_b^r, skb, c) = mg^r/((x_b^r)^{(1/b)}) = m$

2.3. 方式の比較検討

(1) ElGamal 乱数暗号化方式

proxy, third party での処理を必要とせず、Alice から Bob への暗号化通信を可能としている。また、unidirectional(一方向性)を有し、Bob は、復号能力を委譲されるが、Alice は、Bob の暗号文を復号することはできない。しかし、唯一の欠点として Elgamal 暗号の場合、乱数 r を常時固定とすることはできないため、暗号化通信を実施するごとに、r_{A-B} を生成することが必要となる。

(2) ElGamal-Re-Encryption 方式(typical)

Re-encryption key r_{A-B} を 1 回生成し登録しておけば、異なる乱数 r を使用する暗号化通信に対して共通に利用することができる。ただし、proxy, third party での処理を必要とするなど、システム上の制約が生じる。

また、$r_{A-B}^{-1} = r_{B-A}$ であり、bidirectional(両方向性)となる。特に r_{A-B} が、Bob に漏洩すると、$(r_{A-B}^{-1})*b=a$ により、Alice の秘密鍵 ska が取得されてしまう。

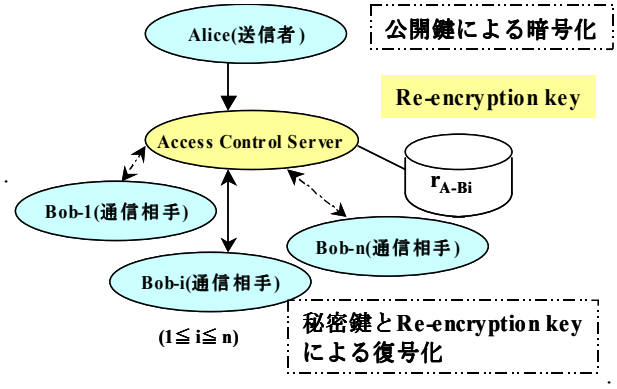

図 1　Re-Encryption Scheme の構成

3. Unidirectinal Proxy Re-Encryption

このように Re-Encryption システムでは、proxy, third party での処理など、システム上の制約、セキュリティ上の問題が生じるため、これらの制約を取り除くよう様々の提案がなされている。次に、Atenise[1]の Bilinear map（Tate-paing）を利用した改良方式について述べる。

3.1. Bilinear map

(1) G,H を群とする。

(i)　群 H,G は同じ素数位数 s の巡回群とする。

(ii) 群 G の生成元を g とする。

(iii) 群 G の任意の要素を Q とすると

$Q = a・g$ ($a \in Fs$; $Fs = Z/sZ$; Z 整数の集合)

と一意に表現される。(・は、群 G の演算)

(2) Bilinear map e_t G x G → H

は、以下の性質を有するものとする。

(i)　Bilinear　$e_t(a・g, b・g) = e_t(g,g)^{ab}$

(ii)　Non-degenerate

$e_t(g,g)$は、群 H を生成する。

3.2. Strong Bilinear Diffie-Hellman Problem

セキュリティについては、群 G,H に対して Strong Bilinear Diffie-Hellman Problem の計算量的困難性を仮定する[8]。

「計算量的困難性の仮定」

$a, b \in Fs$,$Q \in H$ を任意に与えた場合、

$\{g, a・g, b・g, Q\}$に対して、$Q= e_t(g,g)^{(a/b)}$

であるかどうかを、決定することは、計算量的に困難である。

3.3. Unidirectional Proxy Re-Encryption

(1) Key Generation (KG; 鍵の生成)

Alice key-pair　$pk_A = a・g$, $sk_A = a$

Bob　key-pair　$pk_B = b・g$, $sk_B = b$

(2) Re-Encryption Key Generation

$r_{A-B} = R(sk_A, pk_B) = (a^{-1}b)・g$

(3) Encryption(Enc)

(i) Encryption

$Enc(m, pk_A) = c1$　(または、$Enc(m, sk_A)$)

$= \{ mZ^{\kappa}, T_A \}$

ここで $Z = e_t(g,g)$, κ ; random number

$T_A = (a\kappa) \cdot g$

(ii) proxy Encryption(Pec)

信頼できる proxy において、re-encryption key に対する演算処理 Pec を行い、ciphertext を c1 から c2 に変換する。

$Pec(r_{A\text{-}B}, T_A) = e_t(a^{-1}b \cdot g, a\kappa \cdot g)$

$= Z^{\kappa b}$

ciphertext $c2 = \{ mZ^{\kappa}, Z^{\kappa b} \}$

(4) Decryption(Dec)

ciphertext c2 を、sk_B により復号する。

$$Dec(sk_B, c2) = \frac{mZ^{\kappa}}{(Z^{\kappa b})^{(\frac{1}{b})}} = m$$

3.4. Proxy に対する re-encryption key の安全性

Strong Bilinear Diffie-Hellman Problem の成立を仮定すると

Proxy と Bob が結託したとしても、$r_{A\text{-}B}=(a^{-1}b) \cdot g$ から

(i) $r_{B\text{-}A}$ を求めることはできない。(Unidirectional)

(ii) Alice の secret key $sk_A=a$ を求めることは出来ない。(Collusion-safe)

Proxy が取得しする re-encryption key に対して

(iii) $r_{A\text{-}B}$, $r_{B\text{-}C}$ から、 $r_{A\text{-}C}$ を生成することはできない。(Non-transitive)

等の性質を有していることがわかる。

3.5. re-encryption key に対する結託攻撃

再度 Proxy と Bob が結託した場合を考える。

re-encryption key $r_{A\text{-}B}=(a^{-1}b) \cdot g$ を Bob は取得できる。

Bob の secret key $sk_B=b$ より、

$b^{-1} \cdot r_{A\text{-}B}=(b^{-1}(a^{-1}b) \cdot g) =(a^{-1}) \cdot g$

そこで、Carroll の key-pair を $pk_C = c \cdot g$, $sk_C = c$ とすると Carroll の secret key により

$c \cdot ((a^{-1}) \cdot g) = (a^{-1}c) \cdot g$

$= r_{A\text{-}C}$

このように、Carroll は、Alice から委譲される re-encryption key を取得することが可能となる。

従って、本方式は、trusted proxy による re-encryption key に関しての演算処理を必要とする。

4. Unidirectinal Re-Encryption Sheme

この Atenise[1]の Proxy Re-Encryption Scheme に基づいて検討した、proxy 演算処理を必要としない、結託攻撃に対して安全な方式の機能構成について述べる。

4.1. システムの機能構成

図 2 を参照して、本提案システムの機能構成について述べる。暗号化通信は、送信者を A(Aice)、通信相手を Bi(Bobi, 1≦i≦n)とする。

(1) Key Generation(KG) $\{(pkx , skx, akx) \leftarrow KG(1^m) \}$

(i) ユーザ x に対して、key-pair(pkx, skx)を生成する。

(ii)ユーザ x が作成した暗号化データを復号する権限を委譲するために使用する受託秘密鍵(secret accept key)akx を生成する。

(2) First-Level Encryption(E1,D1)

通常の公開鍵による暗号処理機能を有する。

(i) 暗号化(E1)

E1(pkx, m) = c1 { m; plaintext; c1; ciphertext }

(ii) 復号(D1)

D1(skx, c1) = m

(3) Second-Level Encryption(E2)

key-pair pk_A と accept key ak_A を使用して、複数のユーザに、暗号化データを提供する暗号処理機能を有する。

$E2(pk_A, ak_A, m) = \{c2, T_A \}$

{ m, plaintext; c2, ciphertext; T_A, random dara }

(4) Re-Encryption Key Generation(R)

A(Aice)が Bi(Bob-i, 1≦i≦n)に、暗号化データの復号権限を委譲するため、accept key ak_A を利用して re-encryption key $r_{A\text{-}Bi}$ を生成する。

$r_{A\text{-}Bi} = R(pk_{Bi}, sk_A, ak_A)$

{ input Bi(Bob-i)の pk_{Bi} ; A(Alice)の sk_A,ak_A}

(5) Sub-Decryption data Generation(S)

A(Aice)が提供する暗号化データ c2 を Bi(Bob-i)が復号するために使用する Sub-decryption data $d_{A\text{-}Bi}$ を生成する。

$d_{A\text{-}Bi} = S(sk_{Bi}, T_A)$

{ input Bi(Bob-i)の sk_{Bi} ; A(Aice)が作成した c2 の T_A}

(6) Second-Level decryption(D2)

A(Aice)が E2 により提供する暗号化データ c2 を、$r_{A\text{-}Bi}$, $d_{A\text{-}Bi}$ により演算処理し、復号する。

$D2(r_{A\text{-}Bi} , d_{A\text{-}Bi}, c2) = m$

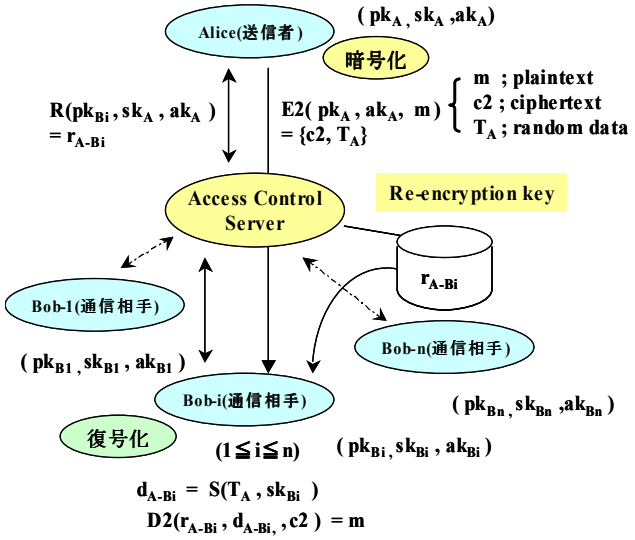

図 2 Re-Encryption Scheme の方式

4.2. 本提案方式の特性

次に、本提案方式により提供される機能の特性を示す。Atenise[1]の方式と異なる機能については、★を記載した。特に、本提案方式では、Atenise の方式では、得られなかった結託攻撃に関する Non-tansferrable の機能を提供することが可能となった。なお、結託攻撃に関しては、さらに厳しい non-key leakage の機能を有する方式についても検討した。この機能も達成可能と考えられるが、まだ、数学的な確証を得ていないので、検討内容については付録に示した。なお、この機能を有することができれば、ユーザに対する認証をせずに、暗号文を送付することも可能と考えられる。

(1) Unidirectional

Alice と Bob の間で使用する re-encryption key の $r_{A\text{-}B}$ と $r_{B\text{-}A}$ とは、$r_{A\text{-}B} \neq r_{B\text{-}A}$ であり、さらに一方からもう片方の key を導出することはできない。従って、$r_{A\text{-}B}$ だけが登録された場合、Bob は Alice の暗号化データを復号することができるが、逆に Alice は Bob の暗号化データを復号することはできない。

(2) Non-interactive

re-encryption key は、$r_{A\text{-}B} = R(pk_B, sk_A, ak_A)$ となっており、Alice は Bob の public key pk_B を取得し、次に Alice 所有の secret key sk_A と accept key ak_A との演算処理より $r_{A\text{-}B}$ を生成する。従って、re-encryption key の生成は、trusted third party の介在を必要としない。

(3) Proxy-less★

ciphertext c2 の復号処理 D2 に必要とするデータ re-encryption key $r_{A\text{-}B}$ および、sub-decryption data $d_{A\text{-}B}$ は、Alice と Bob の演算処理のみで作成され、両者の結託から安全のため、proxy による演算処理を必要としない。ただし、c2, $r_{A\text{-}B}$ の取得には、Access-Conntorl Server による Bob の public key の認証は必要である。

(4) Original-access

Alice は、E2 で暗号化した ciphertext c2 を、自身の所有する secret key sk_A で復号することができるため、ciphertext c2 の保守・管理が可能である。

(5) Key optimal

復号処理 D2 に必要な、re-encription key $r_{A\text{-}B}$ および、sub-decryption data $d_{A\text{-}B}$ は、システムがサポートするユーザの利用数に影響されず、サイズは一定である。

(6) Collusion-safe

re-encryption key $r_{A\text{-}B}$ ($r_{B\text{-}A}$)には Alice(Bob)の secret key $sk_A(sk_B)$と accept key $ak_A(ak_B)$の情報が含まれているが、Bob(Alice)は、re-encryption key から、$\{sk_A, ak_A\}$ ($\{sk_B, ak_B\}$) を算出して取得することはできない。

(7) Non-transitive

re-encryption key $r_{A\text{-}B}$, $r_{B\text{-}C}$ から $r_{A\text{-}C}$ を生成することはできない。

(8) Non-transferable★

Alice が提供する re-encryption key $r_{A\text{-}Bi}$ のユーザ Bob-i($1 \leqq i \leqq n$)が結託しても、Alice の accept key ak_A を生成することはできないため、Alice が許可したユーザ以外のユーザに対する、re-encryption key を生成することはできない。

(9) Non-key leakage★

Alice が Bob1,Bob2 に対して re-encryption key $r_{A\text{-}B1}$, $r_{A\text{-}B2}$ を委譲した場合、Bob1 と Bob2 の結託により、Alice の ciphertext c2 を復号可能な re-encryption key と key-pair との組みを生成することは可能である。

しかし、生成した key-pair に対する電子署名を偽造することは不可能である。このため、Access-Contorl Server による認証を実施すれば、不都合は生じない。

(10) Secret sharing★

First-Lebel Encryrption E1,D1 および re-encryption key $r_{A\text{-}B}$、sub-decryption data $d_{A\text{-}B}$ の生成には、secret key をによる操作が必要となる。この secret key を(2,2)で秘密分散し、さらに乱数を掛けて処理するため、secret key に対して秘密かつ分散した処理が実施可能である。

5. 本提案の実装方式

次に、本提案の scheme を Bilenear map を利用した実装内容について示す。Atenise[1]は、Bilenear map として Tate-paring を利用したが、本提案方式では、weil-paring[8][9][10][11]を利用した。

5.1. Bilinear map

(1) G, H を群とする。

(i) 群 H は、素数位数 s の群とする。

(ii) 群 G は、アーベル群であり素数位数 s の異なる生成元 Go, Ge から生成されるものとする。

従って,$G \equiv Fs \times Fs$ ($Fs = (Z/sZ)$, Z; 整数の集合)

(iii) 群 G の要素を Q とすると、

$Q = a \cdot Go + b \cdot Ge$ ($a, b \in Fs$)

と一意に表現される。

(2) Bilinea map e_w $G \times G \rightarrow H$ は、

以下の性質を有するものとする[11]。

(i) Bilinear $e_w(S1+S2, T) = e_w(S1, T)\, e_w(S2, T)$

$e_w(S, T1+T2) = e_w(S, T1)\, e_w(S, T2)$

(ii) Alternating $e_w(T, T) = 1$

$e_w(S, T) = e_w(T, S)^{-1}$

(iii) Non-degenerate $e_w(T, T) = 1$

ならば、$T = 0 \in G$

(iv) よって $e_w(Go,Ge) = Z$ とすると

$e_w(a \cdot Go + b \cdot Ge, c \cdot Go + d \cdot Ge) = Z^{(ad-bc)}$

$= e_w(Go + Ge, bc \cdot Go + ad \cdot Ge)$

5.2. Key-structure

(1) Trapdoorfunction

$x \in Fs$ に対して関数 TRo,TRe を

TRo(x)=x・Go, TRe(x)=x・Ge
とすると、関数 TRo,TRe は、trapdoorfunction
となり離散対数問題が成立する。
(2) Key-pair
G≡Fs×Fs は、有限体 Fs 上のベクトル空間と考えることができる。従って、(a,b) (a,b ∈ Fs)に対して
Q1= a・Go+b・Ge、もしくは、Q2= b・Go+a・Ge とは、1 対 1 に対応する。
要素 a,b の順序付けを考えなければ、
秘密鍵 sk=(a,b) に対して
公開鍵 pk ={(a・Go , b・Ge) , (a・Ge , b・Go)}
と考えることができる。

5.3. Strong Bilinear Diffie-Hellman Problem

Go, Ge が生成する巡回群 $\hat{Go}=<Go>,\hat{Ge}=<Ge>$ と群 H に対して、Strong Bilenear Deffie-Hellman Problem の計算量的困難性を仮定する。
「計算量的困難性の仮定」
a,b∈ Fs, Q∈ H を任意に与えた場合、
$\{Go, Ge, a\cdot Go, b\cdot Ge, Q\}$ に対して、$Q=e_w(Go,Ge)^{(a/b)}$ であるかどうかを決定することは、計算量的に困難である。

5.4. 2 次元における計算量的困難性の仮定

Go,Ge で生成される、2 次元の群 G に対して、さらに次の計算量的困難性を仮定する。
Theorem
Q, R, S∈ H, a, b ∈ Fs を任意に与えた場合、
{Q, R, S, a, b}に対して、Q=a・R+b・S であるかどうかを決定することは、計算量的に困難である。
Proof
u1,u2,v1,v2∈ Fs を定め、
R = u1・Go+v1・Ge, S = u2・Go+v2・Ge とする。
Q = (au1+bu2)・Go + (av1+bv2)・Ge
これより、
$e_w(Q,Ge) = Z^{(au1+bu2)}$, $e_w(Go,Q) = Z^{(av1+bv2)}$
{u1, u2, v1, v2}は既知であるが、{au1+bu2, av1+bv2}は、Z のべき乗の離散対数問題であり、求めることは出来ないため、a,b を求めることは計算量的に困難である。
(注) なお、Q=a・R の場合、$R=a^{-1}$・Q と逆算することができる。従って、ここで示した計算量的困難性は、2 次元にて成立する特有の性質と考えられる。

5.5. 実装方式

(1) Key Generation (KG; 鍵の生成)
まず、Alice,Bob, Carroll の key-pair および、(secret) accept key を、下記のように定めておく。

Alice key-pair pk_A = (a・Go , b・Ge
b・Go , a・Ge)
sk_A = (a, b)
accept key ak_A = (u_A , v_A)

Bob key-pair pk_B = (α・Go , β・Ge
β・Go , α・Ge)
sk_B = (α , β)
accept key ak_B = (u_B , v_B)

Carroll key-pair sk_C = (ξ・Go , η・Ge
η・Go , ξ・Ge)
sk_C = (ξ , η)
accept key ak_C = (u_C , v_C)

(2) Re-Encryption Key Genration (R)
Alice は Bob に、Alice の暗号化データを復号する権限を委譲するため、Bob の公開情報をもとに re-encryption key r_{A-B} を作成する。
r_{A-B} = R(pk_B , sk_A , ak_A)
= (α au_A)・Go + (β bv_A)・Ge
(3) First-Level Encryption (E1,D1)
plaintext m ∈ Fs
Z = e_w(Go, Ge)、 κ , τ ∈ Fs (random number)とする。
E1(pk_A, m, κ , τ) = c1
= {$mZ^{(\kappa a-\tau b)}$, Z^{κ} , Z^{τ} }
$$D1(sk_A,c1)=\frac{mZ^{(\kappa a1-\tau b)}}{(Z^{\kappa})^{a}(Z^{\tau})^{-b}}=m$$
ここで、e_w(a・Go, Ge) = Z^{a} e_w (Go, b・Ge) = Z^{b}
このように、E1, D1 は通常の Elgamal 暗号の暗号処理方式を踏襲している。
(4) Second-Level Encryption(E2)
E2(pk_A, ak_A, m, κ , τ) = { c2 , T_A }
$$c2=mZ^{(\kappa au_A-\tau bv_A)}$$ (κ , τ ∈ Fs; random number)
T_A ={ τ・Go, κ・Ge }
E2 には、accept key ak_A が含まれる。ciphertext c2 は、{au_A ,bv_A }{(bv_A)Go , (au_A)・Go} {(au_A)・Go ,(bv_A)Ge } いずれかの組の値が求まれば、復号することができる。
(5) Sub-decryption data Generation(S)
d_{A-B} = S(sk_B , T_A)
= β^{-1}・(τ・Go) + α^{-1}・(κ・Ge)
(6) Decryption(D2)
$$D2(r_{A-B},d_{A-B},c2)=\frac{mZ^{(\kappa au_A-\tau bv_A)}}{e_w(r_{A-B},d_{A-B})}=m$$
ここで
$$e_w(r_{A-B},d_{A-B})=Z^{(\kappa au_A-\tau bv_A)}$$
$$=e_w(\alpha au_A\cdot Go+\beta bv_A\cdot Ge,\beta^{-1}\tau\cdot Go+\alpha^{-1}\kappa\cdot Ge)$$

6. Re-Encryption key の安全性について

6.1. Re-Encryption key の安全性

次の re-encryption key r_{X-Y} を考える。
r_{A-B} = (α au_A)・Go + (β bv_A)・Ge
r_{B-A} = (a α u_B)・Go + (b β v_B)・Ge
r_{B-C} = (ξ α u_B)・Go + (η β v_B)・Ge
r_{A-C} = (ξ au_A)・Go + (η bv_A)・Ge
5.4 項で示した、群 G の計算量的困難性より、以下の re-encryption key の安全性を定めることが出きる。

(1) Unidirectional

Bob は、$r_{A\text{-}B}$ から、{au_A・Go, bv_A・Ge} を求めることはできない。

(2) Collusion-safe

Alice Bob は、$r_{A\text{-}B}$,$r_{B\text{-}A}$ から、{ au_A, bv_A }
{au_A・Go, bv_A・Ge} を求めることはできない。

(3) Non-transitive

$r_{A\text{-}B}$, $r_{B\text{-}C}$ から $r_{A\text{-}C}$ を求める場合、
$\xi\, au_A = (\alpha\, au_A)(\xi / \alpha)$, $\eta\, bv_A = (\beta\, bv_A)(\eta / \beta)$
必ずしも、$\xi / \alpha = \eta / \beta$ が成立するとは限らない。

(4) Non-transferable

複数のユーザの結託攻撃によっても、
{au_A, bv_A} を求めることはできない。

6.2. Non-key leakage

$r_{A\text{-}(B+C)} = r_{A\text{-}B} + r_{A\text{-}C}$
$= ((\alpha + \xi)au_A)$・Go + $((\beta + \eta)bv_A)$・Ge
よって
$sk_{(B+C)} = (\alpha + \xi,\ \beta + \eta)$とすると
$d_{A\text{-}(B+C)} = S(sk_{(B+C)}, T_A)$

$$= (\frac{\tau}{\beta+\eta})\cdot Go + (\frac{\kappa}{\alpha+\xi})\cdot Ge$$

$$e_w(r_{A-(B+C)}, d_{A-(B+C)})$$
$$= Z^{(\kappa au_A - \tau bv_A)}$$

従って、Alice の ciphertext c2 を復号する key-pair $pk_{(B+C)}$, $sk_{(B+C)}$ および、re-encryption key $r_{A\text{-}(B+C)}$を Bob と Carroll の結託攻撃によって生成することができる。このため、Non-key leakage の機能は満足できない。
なお、$pk_{(B+C)}$の電子署名を偽造するのは不可能である。

7. まとめ

Atenise *et al.*(2006)[1]によって提案されている Proxy Re-Encryption Scheme に対して、結託を受けない、即ち、演算機能を proxy に持たせる必要のない方式の検討を行った。本方式を、2 次元の Bilinear map を有する群上に、2 次元特有の計算量的困難性を仮定して構成した。具体的には、有限体上の楕円曲線上の s 等分点と、weil-paring の演算により、実装することが可能である。

また、秘密鍵を分散させ、鍵に乱数を掛けて処理するため、秘密分散させた秘密鍵は、分散させたまま合成することなく処理することが可能である。

今後の課題としては、(i) Elgamal 暗号方式に共通鍵暗号方式を組み合わせて、IND-CCA2 を満足する方式とすること[6][7]。(ii) IBE 方式のように、よりユーザに使いやすい方式とすること[9]。(iii) re-encryption key と組み合わせた署名方式を検討することなどが考えられる。

本方式は、2 次元上の鍵構成であるため、鍵の管理が複雑となる可能性があるが、複数のユーザが暗号化データを共有できる従来にない機能を有しており、ユーザに使いやすい方式とすることができれば、広く一般に利用していただけるものと考える。

8. 謝辞

本報告に関し、公的機関の研究者の方、本学会の論文査読者の方より、貴重なご意見をいただきました。公表に耐える内容になったのではないかと考え報告する次第です。本報告を執筆するにあたり、お世話になりました様々の専門家の方々および、小さな会社にもかかわらず、執筆の機会を与えていただいた弊社社長に、心より感謝申し上げます。

9. 付録

Non-key leakage 対策方式の検討

4.2 項で述べた、設定した re-encryption key 以外の key 生成を不可能とする方式についての検討内容を以下に示す。

9.1. ElGamal 暗号の weil-paring による分解

$W(x) = x + \frac{1}{x}$とする。

$r_{A-B}, s_{A-B}, t_A, t_{\alpha\beta}, q_{A-B}$を$Alice, Bob$の$key-pair$より
下記のように定める。

$$r_{A-B} = W(\alpha a)\cdot Go + W(\beta b)\cdot Ge$$
$$s_{A-B} = W(\frac{a}{\alpha})(\frac{1}{W(\alpha)})\cdot Go + W(\frac{b}{\beta})(\frac{1}{W(\beta)})\cdot Ge$$
$$t_A = \tau\cdot Go + \kappa\cdot Ge$$
$$t_{\alpha\beta} = (\frac{\tau}{W(\beta)})\cdot Go + (\frac{\kappa}{W(\alpha)})\cdot Ge$$
$$q_{A-B} = W(\frac{a}{\alpha})\cdot Go + W(\frac{b}{\beta})\cdot Ge$$

すると

$$W(x)W(y) = W(xy) + W(\frac{x}{y})$$

より

$$r_{A-B} + q_{A-B} = W(\alpha)W(a)\cdot Go + W(\beta)W(b)\cdot Ge$$

よって

$$e_w(W(\alpha)W(a)\cdot Go + W(\beta)W(b)Ge\ ,\ \frac{\tau}{W(\beta)}\cdot Go + \frac{\kappa}{W(\alpha)}\cdot Ge)$$
$$= e_w(W(a)\cdot Go + W(b)\cdot Ge\ ,\ \tau\cdot Go + \kappa\cdot Ge)$$
$$= Z^{W(a)\kappa - W(b)\tau}$$
$$= e_w(r_{A-B} + q_{A-B}, t_{\alpha\beta}) = e_w(r_{A-B}, t_{\alpha\beta})\cdot e_w(q_{A-B}, t_{\alpha\beta})$$

ここで

$$e_w(q_{A-B}, t_{\alpha\beta})$$
$$= e_w(W(\frac{a}{\alpha})\cdot Go + W(\frac{b}{\beta})\cdot Ge\ ,\ \frac{\tau}{W(\beta)}\cdot Go + \frac{\kappa}{W(\alpha)}\cdot Ge)$$
$$= e_w(W(\frac{a}{\alpha})\frac{1}{W(\alpha)}\cdot Go + W(\frac{b}{\beta})\frac{1}{W(\beta)}\cdot Ge\ ,\ \tau\cdot Go + \kappa\cdot Ge)$$
$$= e_w(s_{A-B}, t_A)$$

以上より次の関係式が定まる。

・ElGamal 暗号の weil-paring による分解

$$Z^{(a+\frac{1}{a})\kappa-(b+\frac{1}{b})\tau} = Z^{W(a)\kappa-W(b)\tau}$$
$$= e_w(r_{A-B}, t_{\alpha\beta}) \cdot e_w(s_{A-B}, t_A)$$

9.2. 暗号方式

(1) Encryption(Enc)

$$Enc(pk_A, m) = c = \{mZ^{W(a)\kappa-W(b)\tau}, T_A\}$$
$$T_A = \{\tau \cdot Go, \kappa \cdot Ge\}(\tau, \kappa\ random)$$

(2) Re-Encryption Key Generation(Rr,Rs)

Alice Bob 間の interaction により

$$r_{A-B} = R_r(sk_A, P_{k1B}(sk_B))$$
$$s_{A-B} = R_s(sk_A, p_{k2B}(sk_B))$$

ここで、$p_{k1B}(sk_B), p_{k2B}(sk_B)$は、$Bob$が$sk_B$より生成し公開可能な情報を示す。

(3) Decryption(Dec)

・Sub-Decryption data Generation (S_A,$S_{\alpha\beta}$)

$$S_A(sk_B, T_A) = t_A$$
$$S_{\varepsilon\beta}(sk_B, T_A) = t_{\alpha\beta}$$

・Decryption

$$Dec(c, t_A, t_{\alpha\beta}) = \frac{mZ^{W(a)\kappa-W(b)\tau}}{e_w(r_{A-B}, t_{\alpha\beta}) \cdot e_w(s_{A-B}, t_A)} = m$$

9.3. 結託攻撃からの安全性についての検討

5.5 項に示す構成を key-pair とすると、Alice Bob の accept key は{u_A,v_A}{u_B,v_B}であることから、Enc を Second-Level Encryption (Enc2)として記載すると、

$$Enc2(pk_A, m) = c2 = \{mZ^{W(au_A)\kappa-W(bv_A)\tau}, T_A\}$$

となる。re-encryption key として、

$$r_{A-B} = W(\alpha au_A) \cdot Go + W(\beta bv_A) \cdot Ge$$
$$s_{A-B} = W(\frac{au_A}{\alpha})(\frac{1}{W(\alpha)}) \cdot Go + W(\frac{bv_A}{\beta})(\frac{1}{W(\beta)}) \cdot Ge$$
$$r_{B-A} = W(a\alpha u_B) \cdot Go + W(b\beta v_B) \cdot Ge$$
$$s_{B-A} = W(\frac{\alpha u_B}{a})(\frac{1}{W(a)}) \cdot Go + W(\frac{b}{\beta})(\frac{1}{W(b)}) \cdot Ge$$
$$r_{A-C} = W(\xi au_A) \cdot Go + W(\eta bv_A) \cdot Ge$$
$$s_{A-C} = W(\frac{au_A}{\xi})(\frac{1}{W(\xi)}) \cdot Go + W(\frac{bv_A}{\eta})(\frac{1}{W(\eta)}) \cdot Ge$$

等が、考えられる。これらの re-encryption key をもとに、Collusion-safe, Non-transitive, Non-transferable 等の機能の成立を確認することができる。ここでは、Non-key leakage 機能の成立について検討する。

Alice が Bob、Carroll に委譲する re-encryption key の和 $r_{A-B} + r_{A-C}$ を求める。

この和を与える secret key を sk=(x,y)とすると、

$$r_{A-B} + r_{A-C}$$
$$= (W(\alpha au_A) + W(\xi au_A)) \cdot Go + (W(\beta bv_A) + W(\eta bv_A)) \cdot Ge$$
$$= W(x) \cdot Go + W(y) \cdot Ge$$
$$(\alpha au_A + \frac{1}{\alpha au_A}) + (\xi au_A + \frac{1}{\xi au_A}) = x + \frac{1}{x}$$
$$(\beta bv_A + \frac{1}{\beta bv_A}) + (\eta bv_A + \frac{1}{\eta bv_A}) = y + \frac{1}{y}$$

au_A, bv_A は、未知のため、α , β , ξ , η の値だけでは、この等式を満足する x, y を求めることはできない。同様の性質が、s_{A-B}, s_{A-C} に対しても成立する。このように、re-encryption key の線型結合では、対応する secret key を求めることはできないことがわかる。

おそらく、au_A, bv_A を求めずに結託攻撃を成功させることは、極めて困難であると想定される。本方式の安全性に関する数学的な証明や、さらに同報暗号への応用は今後の課題としたい。

10. 文　献

[1] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger "**I**mproved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage (eprint)"ACM Transactions on Information and System Security (TISSEC), February 2006

[2] Matt Blaze, G. Bleumer, and M. Strauss. "Divertible protocols and atomic proxy cryptography." In Proceedings of Eurocrypt '98,volume 1403, pages 127-144, 1998

[3] Yevgenity Dodis and Anca Ivan. "Proxy cryptography revised." In Proceegings of the Tenth Network and Distributed System Security Symposium, February 2003

[4] Whitfield Diffie, Martin E. Hellman "New Directions in Cryptography" IEEE Transactions on Information Theory(1976)

[5] Masahiro Mambo and Eiji Okamoto. "Proxy cryptosystems: Delegation of the power to decrypt cipertext." IEICE Trans. Fund Electronics Communications and Computer Science, E80-A/1:54-63,1997

[6] Eiichiro Fujisaki and Tatsuaki Okamoto. "Secure integration of asymmetric encryption schemes. " In Proceedings of Crypto '99,volume 1666 of LNCS,pages 537-554, 1999

[7] M. Abdalla, M. Bellare and P. Rogaway."DHIES: An encryption scheme based on the Diffie-Hellman Problem" CT-RSA 01, Lecture Notes in Computer Science Vol. 2020, D. Naccache ed, Springer-Verlag, 2001.

[8] S. Mitsunari, R. Sakai, and M. Kasahara. "A new traitor tracing." IEICE Trans. Fundamentals, E85A (2):481--484, 2002.

[9] Dan Boneh and Matt Franklin. "Identity-based encryption from the weil Paring. " SIAM Journal of Compuring, 32(3):586-615, 2003

[10] Michael Scott. MIRACL library,2005,Indigo Software. http://indigo.ie/~mscott/#Elliptic

[11] J. Silverman,"The arithmetic of elliptic curve" , Springer-Verlag,1986