情報処理学会論文誌 Vol. 52 No. 9 2662–2673 (Sep. 2011)

検索クエリ中のワイルドカードを秘匿する隠れベクトル暗号システム

秋山浩岐[†1,*1] 満保雅浩[†2] 岡本栄司[†1]

検索可能暗号とは，復号権限を持つユーザが暗号文に対してキーワードなどによる検索を実現する暗号方式である．特に，キーワード検索のみを対象としていた従来の方式と比較して，より柔軟な検索を実現する方式として隠れベクトル暗号方式を用いた手法が提案されている．この隠れベクトル暗号方式を用いた手法により，大小比較検索や部分集合検索などが可能となった．しかし，これらの方式では検索クエリ情報からどの検索属性にワイルドカードが割り当てられているかという情報が漏洩し，その情報から検索内容に関する情報が漏洩する危険性があることが指摘されている．このことは暗号文書そのものの属性情報が漏洩してしまうことにもつながりうる．そこで本論文では Iovino らの方式に着目し，合成数位数の群上で定義される双線写像を用いることで検索クエリ中のワイルドカードを通信路上の盗聴者に対して秘匿できることを保証可能な方式を提案する．

A Hidden Vector Encryption Scheme Hiding Wild Cards of a Search Query

Hiroki Akiyama,[†1,*1] Masahiro Mambo[†2] and Eiji Okamoto[†1]

Searchable encryption allows one to search on encrypted data securely and is suitable for various systems including E-mail systems. Especially, hidden vector encryption scheme introduced by Boneh and Waters allows one to make flexible search on encrypted data, e.g. conjunctive keyword search, range search and subset search. Iovino and Persiano proposed a modified efficient scheme by using bilinear groups of prime order. But as pointed out by Boneh and Waters, both the scheme by Iovino and Persiano and that by Boneh and Waters have a problem such that a receiver's query exposes its attribute because anyone can distinguish whether receiver's query is wild card or not. As long as we know, this problem has not been solved. To solve this problem, we introduce a security notion, wild card hiding, meaning that there is no way for any PPT adversary (except the server) to distinguish whether receiver's query is wild card or not, and propose a scheme which is wild card hiding against anyone except the server by using bilinear groups of composite order.

1. はじめに

検索可能暗号とは，暗号文書に対して安全に検索を実現する暗号方式である．たとえばEメールの環境を想定する．送信者はメールサーバを含めた他者にメッセージを秘匿したければ，暗号化技術を用いることにより受信者へ安全に送信することができる．しかし，メールサーバは暗号化されたメッセージの内容を知ることができないため，メールサーバに大量に保管された暗号化メッセージの中で受信者が必要とするもののみを検索抽出し，受信者に送ることができないという問題がある．そこで，暗号化メッセージに対してメッセージの属性を示す情報を付加することで検索を実現する方式を検索可能暗号という．メッセージの暗号化自体に安全な暗号を用いるだけでなく，付加する属性情報の暗号化が必要である．

現在までに，キーワード検索を実現する方式が様々提案されている[1),4),6),7)]．一方で，キーワード以外による検索の実現も望まれている．すなわち，検索条件がキーワードとの一致として与えられるのではなく，たとえば大小比較として与えられるものである．これを実現する方式が Boneh らの方式[3)] や Iovino らの方式[5)] であり，隠れベクトル暗号方式と呼ばれる．隠れベクトル暗号方式では，メッセージの属性をベクトルで表し，これを暗号化する．そして受信者により与えられた検索クエリが適合したメッセージのみにおいて復号が成立する．しかし，これらの方式では，ある特定の検索において検索内容が分かってしまうという問題が指摘されており[3)]，統計的解析により暗号化メッセージ自体の性質が漏洩する危険性がある．著者の知る限りにおいて，この問題はこれまでに解決されていない問題である．

そこで，本論文では安全な検索可能暗号方式として，特に Iovino らの方式[5)] に着目し，通信路上の盗聴者に対してさえも解決されていなかった検索内容が漏洩するという問題を，通信路上の盗聴者に対して解決する方式を示す．

†1 筑波大学
University of Tsukuba
†2 金沢大学
Kanazawa University
*1 現在，日立ソリューションズ株式会社
Presently with Hitachi Solutions, Ltd.

2. 準　　備

2.1 双線形写像

$\mathbb{G}$，$\mathbb{G}_T$ を位数 n の異なる巡回群とする．このとき，双線形写像 $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ は，以下に示す 3 つの性質を満たす．

(1)　双線形性
　g，$h \in \mathbb{G}$ と a，$b \in \mathbb{Z}_r$ に対して，$e(g^a, h^b) = e(g, h)^{ab}$.

(2)　非縮退性
　$\mathbb{G}$ の生成元 g について，$e(g, g)$ は $\mathbb{G}_T$ の生成元となる．

(3)　計算可能性
　任意の g，$h \in \mathbb{G}$ について，$e(g, h)$ が効率的に計算できるアルゴリズムが存在する．

特に，異なる 2 つの素数 p，q に対して合成数 $n = pq$ を位数とする群上で定義される双線形写像について，位数 p，q の巡回群をそれぞれ $\mathbb{G}_p$，$\mathbb{G}_q$ として，以下の性質がある．

$e(h_p, h_q) = 1 \quad (h_p \in \mathbb{G}_p, h_q \in \mathbb{G}_q)$

2.2 隠れベクトル暗号方式

隠れベクトル暗号方式は Boneh らによって提案された検索可能暗号である[3)]．

彼らの方式では，セキュリティパラメータを 1^k，$m = poly(k)$ として暗号文属性ベクトルと呼ばれるベクトル $x \in \{0, 1\}^m$ を生成する．この暗号文属性ベクトルは暗号文の持つ属性を示すベクトルであり，ベクトルの各要素位置はそれぞれ 1 つの属性に対応する．各要素値が 1，0 であることは，それぞれその要素位置に対応する属性の有無を意味する．

一方で，受信者がいくつかの属性に注目した検索クエリを生成するには，検索属性ベクトル $y \in \{0, 1, *\}^m$ を生成する．ここで，*はワイルドカードを示す．

これらの 2 つのベクトルについて，関数 $P_x(y)$ は以下のように定義される．

$$P_x(y) = \begin{cases} 1, & x_i = y_i \ \vee \ y_i = * \ \text{ for } 1 \le i \le m; \\ 0, & \text{otherwise}; \end{cases}$$

以上を用いて，隠れベクトル暗号方式は以下のように定義される．提案方式は従来の隠れベクトル暗号方式に対してサーバ側で実行されるアルゴリズムである **ServerSetup** を加えて構成されるため，**ServerSetup** が存在する場合としない場合の両方に対応する定義となっている．なお，(SS) と $(\overline{SS})$ はそれぞれ **ServerSetup** が存在する場合と存在しない場合にのみ実行される処理を示し，また **ServerSetup** が存在しない場合の定義は文献 3), 5) で示されているものとなる．

Definition 1（隠れベクトル暗号方式）．隠れベクトル暗号方式は，以下に示すアルゴリズムから構成される．

Setup　入力としてセキュリティパラメータ 1^k と属性長 $m = poly(k)$ を得て，公開鍵 Pk とマスタ秘密鍵 Msk を生成する．

(SS) **ServerSetup**　入力として公開鍵 Pk を得て，サーバ公開鍵 Pk' とサーバ秘密鍵 Sk' を生成する．

KeyGenerateion　$(\overline{SS})$ 入力としてマスタ秘密鍵 Msk と検索属性ベクトル y を得て，検索クエリとして用いられる秘密鍵 K_y を生成する．
　(SS) 入力としてマスタ秘密鍵 Msk，サーバ公開鍵 Pk'，検索属性ベクトル y を得て，検索クエリとして用いられる秘密鍵 K_y を生成する．

Encryption　入力として公開鍵 Pk，暗号文属性ベクトル x，平文 M を得て，暗号文 Ct_x を生成する．

Decryption　$(\overline{SS})$ 入力として暗号文 Ct_x と秘密鍵 K_y を得て，平文 M を出力する．
　(SS) 入力として暗号文 Ct_x，秘密鍵 K_y，サーバ秘密鍵 Sk' を得て，平文 M を出力する．

$P_x(y) = 1$ を満たすすべての x，y について，隠れベクトル暗号方式は以下の式を満たす．

ServerSetup が存在しない場合．

$$\Pr[(Pk, Msk) \leftarrow \mathbf{SetUp}(1^k); K_y \leftarrow \mathbf{KeyGeneration}(Msk, y); \\ Ct_x \leftarrow \mathbf{Encryption}(Pk, x, M) : \mathbf{Decryption}(K_y, Ct_x) = M] = 1.$$

ServerSetup が存在する場合．

$$\Pr[(Pk, Msk) \leftarrow \mathbf{SetUp}(1^k); K_y \leftarrow \mathbf{KeyGeneration}(Msk, Pk', y); \\ Ct_x \leftarrow \mathbf{Encryption}(Pk, x, M) : \mathbf{Decryption}(K_y, Ct_x, Sk') = M] = 1.$$

なお，**Definition 1** では，**Encryption** の入力にマスタ秘密鍵 Msk が含まれておらず，公開鍵 Pk が活用されるため，不特定多数の利用者が送信者となりうる E メールなどの状況に適用可能であり，送受信者間でマスタ秘密鍵 Msk を事前に共有するという鍵管理の問題が生じない．

2.3 関 連 研 究

これまでに，いくつかの検索可能暗号が提案されている．Goh は，秘密鍵暗号系の環境

において単一キーワード検索を実現する検索可能暗号方式を提案した[4]．また，Boneh らは公開鍵暗号系の環境において単一キーワード検索を実現する検索可能暗号方式を提案した[1]．これらの方式は単一キーワード検索を実現するという点で同一であるが，想定する暗号技術環境が異なる．

一方で，公開鍵暗号系における複数キーワード検索を実現する検索可能暗号方式も提案されている[6),7]．これらの方式により暗号文に対してマルチキーワード検索が可能になった．しかし，対象とする文書が構造化されている必要があることや，検索可能なキーワードにも制限があるという問題がある．

これらの方式と異なり，暗号文に対してより柔軟な検索を実現する方式が提案されている．Boneh らは連接キーワード検索や部分集合検索，大小検索や範囲検索を実現する方式を提案した[3]．彼らの方式では，合成数位数の群上で定義される双線形写像を用いている．一方で，Iovino らは素数位数の群上で定義される双線形写像を用いて同様の検索を実現する検索可能暗号方式を提案した[5]．

しかし，これらの方式には，ある種の検索を行う際に検索クエリから検索内容自体が漏洩する危険性があることが指摘されている[3]．このことは，検索クエリの内容とその結果を監視することで，間接的に暗号文自体の性質が部分的に漏洩することにつながりうる．

そこで，本論文では Iovino らの方式[5] を変形し，検索内容の漏洩を防ぐ検索可能暗号方式を提案する．特に，3.2 節に定義するワイルドカード秘匿を実現する方式は著者の知る限り存在しないため，通信路上という制限はあるものの，ワイルドカード秘匿を満たす方式を示す．

2.4　既存方式の問題点

既存方式[3),5)] では，復号演算が成功するために 2 つの条件が存在する．

(1)　検索属性ベクトルと暗号文属性ベクトルの属性値が同一．

(2)　検索属性ベクトルの属性値が*.

このうち，前者の条件に関しては，属性値ごとの双線形演算を用いて，属性値が同一ならば復号演算が成立するように暗号文と検索クエリを構成することで，演算の正当性を保証している．

後者の条件に関しては，暗号文属性ベクトルの属性値に依存せずに演算を成功させる必要がある．このため，既存方式では検索属性ベクトルの属性値が*の属性位置については復号演算に組み込まないことで演算の正当性を保証している．この性質のために，攻撃者は検索クエリを得ただけで，少なくとも検索属性値が 0 もしくは 1 であるのか，あるいは*であるのかを判別できてしまう．文献 3) で示される大小比較検索などの場合においては，検索属性ベクトル中の*の位置によって検索内容が直接漏洩してしまう場合がある．これにより，間接的に暗号文の性質が漏洩してしまう危険性がある．この問題を解決するために，ワイルドカードの位置を含めて検索クエリの属性値を秘匿する必要がある．

3.　安全性のモデル

3.1　計算困難性に関する仮定

本節では，以降で用いる 3 つの計算困難性に関する仮定を定義する．これらの仮定は一部の仮定において合成数位数に拡張されていることを除いて文献 2)，3)，5) において示されている仮定である．

composite Decision Bilinear Diffie Hellman：

cDBDH 仮定の記述に先立って，$\mathbf{cDBDHExp}^{\mathcal{A}}$ を定義する．

$\mathbf{cDBDHExp}^{\mathcal{A}}(1^k)$

セキュリティパラメータを 1^k として，$\mathcal{I} = [n = pq, \mathbb{G}, \mathbb{G}_T, g, e]$ を選択する；

$\mathbb{G}$ の部分群 $\mathbb{G}_p$，$\mathbb{G}_q$ について，それぞれ生成元 g_p，g_q を選択する；

a，b，$c \in \mathbb{Z}_n$ をランダムに選択する；

$\eta \in \{0, 1\}$ をランダムに選択する；

もし $\eta = 1$ ならば，$z \in \mathbb{Z}_n$ をランダムに選択する

　　そうでなければ，$z = abc$ とする；

$A = g_p^a$，$B = g_p^b$，$C = g_p^c$，$Z = e(g_p, g_p)^z$ とする；

$\eta' = \mathcal{A}(\mathcal{I}, g_p, g_q, A, B, C, Z)$ を攻撃者 $\mathcal{A}$ の出力とする；

もし $\eta = \eta'$ ならば 0 を，そうでなければ 1 を出力する；

Assumption 1（composite Decision Bilinear Diffie Hellman）．任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して，以下を満たす negligible な関数 $\nu(k)$ が存在する．

$$|\Pr[\mathbf{cDBDHExp}^{\mathcal{A}}(1^k) = 1] - 1/2| = \nu(k).$$

composite Decision Linear：

cDL 仮定の記述に先立って，以下のように $\mathbf{cDLExp}^{\mathcal{A}}$ を定義する．

$\mathbf{cDLExp}^{\mathcal{A}}(1^k)$

セキュリティパラメータを 1^k として，$\mathcal{I} = [n = pq, \mathbb{G}, \mathbb{G}_T, g, e]$ を選択する；

z_1，z_2，z_3，$u \in \mathbb{Z}_n$ をランダムに選択する；

$\eta \in \{0,1\}$ をランダムに選択する；
もし $\eta = 1$ ならば $z \in \mathbb{Z}_n$ をランダムに選択する
　そうでなければ $z = z_2(u - z_3)$ とする；
$Z_1 = g_p^{z_1}$，$Z_2 = g_p^{z_2}$，$Z_{13} = g_p^{z_1 z_3}$，$U = g_p^u$，$Z = e(g_p, g_p)^z$ とする；
$\eta' = \mathcal{A}(\mathcal{I}, g_p, g_q, Z_1, Z_2, Z_{13}, U, Z)$ を攻撃者 $\mathcal{A}$ の出力とする；
もし $\eta = \eta'$ なら 0 を，そうでなければ 1 を出力する；

Assumption 2（modified Decision Linear）．任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して，以下を満たす negligible な関数 $\nu(k)$ が存在する．

$$|\Pr[\mathbf{cDLExp}^{\mathcal{A}}(1^k) = 1] - 1/2| = \nu(k).$$

Subgroup Decision：
SD 仮定の記述に先立って，$\mathbf{SDExp}^{\mathcal{A}}$ を定義する．
$\mathbf{SDExp}^{\mathcal{A}}(1^k)$
セキュリティパラメータを 1^k として，$\mathcal{I} = [n = pq, \mathbb{G}, \mathbb{G}_T, g, e]$ を選択する；
$\eta \in \{0,1\}$ をランダムに選択する；
もし $\eta = 1$ ならば $x \in \mathbb{G}$ をランダムに選択する
　そうでなければ $x \in \mathbb{G}_p$ をランダムに選択する；
$\eta' = \mathcal{A}(\mathcal{I}, x)$ を攻撃者 $\mathcal{A}$ の出力とする；
もし $\eta = \eta'$ なら 0 を，そうでなければ 1 を出力する；

Assumption 3（Subgroup Decision）．任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して，以下を満たす negligible な関数 $\nu(k)$ が存在する．

$$|\Pr[\mathbf{SDExp}^{\mathcal{A}}(1^k) = 1] - 1/2| = \nu(k).$$

3.2　満たすべき安全性

3.2.1　ワイルドカード秘匿

ワイルドカード秘匿を本論文で新たに定義する．このワイルドカード秘匿は以下で定義される $\mathbf{WildCardHidingExp}_{\mathcal{A}}$ を用いて定義される．なお，サーバ側の鍵生成アルゴリズム **ServerSetup** が存在する提案方式では，攻撃者のモデルとしてサーバ以外の通信路上の盗聴者と，サーバを含めた攻撃者の 2 通りが考えられる．このうち，提案方式がワイルドカード秘匿を保証するのはサーバ以外の通信路上の盗聴者であるため，以下では攻撃者モデルを通信路上の盗聴者としてワイルドカード秘匿を定義する．

$\mathbf{WildCardHidingExp}_{\mathcal{A}}(1^k)$
Init.　攻撃者 $\mathcal{A}$ はチャレンジ用の 2 つの異なる検索クエリ属性ベクトル (y_0, y_1) をチャレンジャに渡す．
Setup.　チャレンジャは以下のように鍵を生成する．
- チャレンジャは **Setup** を実行して Msk，Pk を生成し，Pk を攻撃者 $\mathcal{A}$ へ渡す．
- チャレンジャは **ServerSetup** を実行して Sk'，Pk' を生成し，Pk' を攻撃者 $\mathcal{A}$ へ渡す．

Query Phase 1.　攻撃者 $\mathcal{A}$ は検索クエリ属性ベクトル y を選択してチャレンジャに渡す．チャレンジャは以下のように動作する．
- 検索クエリ $K_y \leftarrow \mathbf{KeyGeneration}(Msk, Pk', y)$ を計算し，これを $\mathcal{A}$ に返す．

Challenge.　チャレンジャはランダムに $\eta \in \{0,1\}$ を選択した後に．チャレンジ検索クエリ $K_{y_\eta} \leftarrow \mathbf{KeyGeneration}(Msk, Pk', y_\eta)$ を計算し，これを攻撃者 $\mathcal{A}$ に渡す．
Query Phase 2.　Query Phase 1 と同様．
Output.　攻撃者 $\mathcal{A}$ は 1 ビット $\eta' \in \{0,1\}$ をチャレンジャへ渡し，$\eta = \eta'$ ならば 0，そうでなければ 1 を出力する．

Definition 2（ワイルドカード秘匿）．任意の確率的多項式時間攻撃者 $\mathcal{A}$ について，以下を満たす negligible な関数 ν が存在するとき，隠れベクトル暗号方式はサーバ以外の通信路上の盗聴者に対してワイルドカード秘匿である．

$$|\Pr[\mathbf{WildCardHidingExp}_{\mathcal{A}}(1^k) = 1] - 1/2| = \nu(k).$$

3.2.2　意味論的安全性

以下では意味論的安全性の定義を示す．なお，以下において (SS) と $(\overline{SS})$ はそれぞれ **ServerSetup** が存在する場合，存在しない場合にのみ実行される処理を示し，また **ServerSetup** が存在しない場合の定義は文献 5) で示されているものと同様である．以下で定義される $\mathbf{SemanticExp}_{\mathcal{A}}$ を用いて意味論的安全性を定義する．
$\mathbf{SemanticExp}_{\mathcal{A}}(1^k)$
Init.　攻撃者 $\mathcal{A}$ はチャレンジ用の暗号文属性ベクトル x を指定し，チャレンジャに渡す．
Setup.　チャレンジャは以下のように鍵を生成する．
- チャレンジャは **Setup** を実行して Msk，Pk を生成し，Pk を攻撃者 $\mathcal{A}$ へ渡す．
- (SS) チャレンジャは **ServerSetup** を実行して Sk'，Pk' を生成し，Pk' を攻撃者 $\mathcal{A}$ へ渡す．

Query Phase 1.　攻撃者 $\mathcal{A}$ は $P_x(y)=0$ を満たす検索クエリ属性ベクトル y を選択し，検索クエリ K_y を要求するクエリを出す．チャレンジャは以下のように動作する．

- $(\overline{SS})$ 検索クエリ $K_y \leftarrow$ **KeyGeneration** (Msk, y) を計算し，これを攻撃者 $\mathcal{A}$ に返す．
- (SS) 検索クエリ $K_y \leftarrow$ **KeyGeneration** (Msk, Pk', y) を計算し，これを攻撃者 $\mathcal{A}$ に返す．

Challenge.　攻撃者 $\mathcal{A}$ は 2 つの平文 M_0，M_1 をランダムに選択し，チャレンジャに渡す．チャレンジャは $\eta \in \{0,1\}$ をランダムに選択し，暗号文 $Ct_x \leftarrow$ **Encryption**(Pk, x, M_η) を攻撃者 $\mathcal{A}$ に返す．

Query Phase 2.　Query Phase 1 と同一．

Output.　攻撃者 $\mathcal{A}$ は $\eta' \in \{0,1\}$ を出力する．$\eta = \eta'$ なら 0 を，そうでなければ 1 を出力する．

Definition 3（意味論的安全性）．任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して以下の式を満たす negligible な関数 ν が存在するとき，隠れベクトル暗号方式は意味論的に安全である．

$$|\Pr[\mathbf{SemanticExp}_{\mathcal{A}}(1^k)=1] - 1/2| = \nu(k).$$

3.2.3　属 性 秘 匿

以下では属性秘匿の定義を示す．なお，以下において (SS) と $(\overline{SS})$ はそれぞれ **ServerSetup** が存在する場合，存在しない場合にのみ実行される処理を示し，また **ServerSetup** が存在しない場合の定義は文献 5) で示されているものと同様である．以下で定義される $\mathbf{AttributeHidingExp}_{\mathcal{A}}$ を用いて属性秘匿を定義する．

$\mathbf{AttributeHidingExp}_{\mathcal{A}}(1^k)$

Init.　攻撃者 $\mathcal{A}$ はチャレンジ用に 2 つの異なる暗号文属性ベクトル (x_0, x_1) を生成し，チャレンジャに渡す．

Setup.　チャレンジャは以下のように鍵を生成する．

- チャレンジャは **Setup** を実行して Msk，Pk を生成し，Pk を攻撃者 $\mathcal{A}$ へ渡す．
- (SS) チャレンジャは **ServerSetup** を実行して Sk'，Pk' を生成し，Pk' を攻撃者 $\mathcal{A}$ へ渡す．

Query Phase 1.　攻撃者 $\mathcal{A}$ は $P_{x_0}(y) = P_{x_1}(y) = 0$ を満たす検索クエリ属性ベクトル y を選択し，検索クエリ K_y を要求するクエリを出す．チャレンジャは以下のように動作する．

- $(\overline{SS})$ 検索クエリ $K_y \leftarrow$ **KeyGeneration** (Msk, y) を計算し，これを攻撃者 $\mathcal{A}$ に返す．
- (SS) 検索クエリ $K_y \leftarrow$ **KeyGeneration** (Msk, Pk', y) を計算し，これを攻撃者 $\mathcal{A}$ に返す．

Challenge.　攻撃者 $\mathcal{A}$ は 2 つの平文 M_0，M_1 をランダムに選択する．チャレンジャは $\eta \in \{0,1\}$ をランダムに選択し，暗号文 $Ct_x \leftarrow$ **Encryption**(Pk, x_η, M_η) を攻撃者 $\mathcal{A}$ に返す．

Query Phase 2.　Query Phase 1 と同一．

Output.　攻撃者 $\mathcal{A}$ は $\eta' \in \{0,1\}$ を出力する．$\eta = \eta'$ なら 0 を，そうでなければ 1 を出力する．

Definition 4（属性秘匿）．任意の確率的多項式時間攻撃者 $\mathcal{A}$ に対して以下の式を満たす negligible な関数 ν が存在するとき，隠れベクトル暗号方式は属性秘匿である．

$$|\Pr[\mathbf{AttributeHidingExp}_{\mathcal{A}}(1^k)=1] - 1/2| = \nu(k).$$

4. 提 案 方 式

以下に提案方式の構成を示す．

Setup.　入力としてセキュリティパラメータ 1^k を得て，以下のように公開鍵 Pk とマスタ秘密鍵 Msk を生成する．

(1)　双線形写像パラメータ $\mathcal{I} = [n, \mathbb{G}, \mathbb{G}_T, g, e]$ を選択する．ここで，$e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ は 2 つの異なる素数 p，q を用いた合成数 $n = pq$ を位数とする群上で定義される双線形写像である．

(2)　$\gamma \in \mathbb{Z}_n$ をランダムに選択し，$\mathbb{G}_p$ の生成元 g_p をランダムに選択し，$\Gamma = e(g_p, g_p)^\gamma$ を計算する．ここで，$\mathbb{G}_p$ は $\mathbb{G}$ の部分群のうち位数が p となる群を示す．

(3)　$1 \le i \le m$ $(m = poly(k))$ について，t_i，v_i，r_i，$m_i \in \mathbb{Z}_n$ をランダムに選択し，$T_i = g_p^{t_i}$，$V_i = g_p^{v_i}$，$R_i = g_p^{r_i}$，$M_i = g_p^{m_i}$ とする．

(4)　$\beta \in \mathbb{Z}_n$ をランダムに選択し，g^β を計算する．

(5)　$Pk = [g_p, g^\beta, \mathcal{I}, \Gamma, (T_i, V_i, R_i, M_i)_{i=1}^m]$，$Msk = [g_p, \beta, \mathcal{I}, \gamma, (t_i, v_i, r_i, m_i)_{i=1}^m]$ として，これらを出力する．

ServerSetup.　入力として公開鍵 Pk を得て，以下のようにサーバ公開鍵 Pk' とサーバ秘密鍵 Sk' を生成する．

（1） $\alpha \in \mathbb{Z}_n$ をランダムに選択する．

（2） $Pk' = g^{\alpha}$，$Sk' = \alpha$ として，これらを出力する．

Encryption. 入力として平文 $M \in \mathbb{G}_T$，暗号文属性ベクトル x，公開鍵 Pk を得て，以下のように暗号文を生成する．ただし，ここでの平文 M とは検索対象となる文書そのものを意味するものではなく，事前に決められた検索成功を意味する符号である．

（1） $s \in \mathbb{Z}_n$ をランダムに選択する．

（2） $1 \le i \le m$ について $s_i \in \mathbb{Z}_n$ を選択し，以下の暗号文 Ct_x を出力する．

$$\mathbf{Encryption}(Pk, x, M) = Ct_x = [\Omega, (X_i, W_i)_{i=1}^m].$$

ここで

$$\Omega = M \cdot \Gamma^{-s}, X_i = \begin{cases} T_i^{s-s_i}, & \text{if } x_i = 1 \\ R_i^{s-s_i}, & \text{if } x_i = 0, \end{cases} \quad W_i = \begin{cases} V_i^{s_i}, & \text{if } x_i = 1 \\ M_i^{s_i}, & \text{if } x_i = 0. \end{cases}$$

KeyGeneration. 入力としてマスタ秘密鍵 Msk，サーバ公開鍵 Pk'，検索クエリ属性ベクトル $y \in \{0, 1, *\}^m$ を得て，以下のように検索クエリ K_y を生成する．

（1） $S_y = S_y^1 \cup S_y^0$ を $y_i \neq *$ を満たすすべての i の集合とする．ここで，S_y^i は検索クエリ属性ベクトル y の要素位置のうち，値が i となる要素位置の集合を示す．

（2） $1 \le i \le m$ について $a_i \in \mathbb{Z}_n$ をランダムに選択する．ここで，$\sum_{i \in S_y} a_i = \gamma$ を満たす．

（3） $1 \le i \le m$ について $rnd_{i,0}$，$rnd_{i,1} \in \mathbb{Z}_n$ をランダムに選択する．

（4） 以下のような検索クエリ $K_y = [(g^{rnd_{i,0}}, g^{rnd_{i,1}}, Y_i, L_i)_{i=1}^m]$ を出力する．ここで，

$$Y_i = \begin{cases} g_p^{a_i/t_i} g^{\alpha \cdot rnd_{i,0}}, & \text{if } y_i = 1 \\ g_p^{a_i/r_i} g^{\alpha \cdot rnd_{i,0}}, & \text{if } y_i = 0 \\ g^{\alpha \cdot \beta \cdot rnd_{i,0}}, & \text{if } y_i = *, \end{cases} \quad L_i = \begin{cases} g_p^{a_i/v_i} g^{\alpha \cdot rnd_{i,1}}, & \text{if } y_i = 1 \\ g_p^{a_i/m_i} g^{\alpha \cdot rnd_{i,1}}, & \text{if } y_i = 0 \\ g^{\alpha \cdot \beta \cdot rnd_{i,1}}, & \text{if } y_i = *. \end{cases}$$

Decryption. 入力としてサーバ秘密鍵 Sk'，暗号文 Ct_x，検索クエリ K_y を得て，以下のように計算する．

（1） サーバ秘密鍵 $Sk' = \alpha$ を用いて $1 \le i \le m$ について以下のように動作する．

- $e(g^{\alpha \cdot \beta}, g^{rnd_{i,0}})$ と $e(g, Y_i)$ を計算し，$e(g^{\alpha \cdot \beta}, g^{rnd_{i,0}}) = e(g, Y_i)$ ならば $Y_i' = L_i' = 1$ とする．
- そうでなければ，$Y_i' = Y_i / g^{\alpha \cdot rnd_{i,0}}, L_i' = L_i / g^{\alpha \cdot rnd_{i,1}}$ とする．

（2） 以下のように復号演算を行う．

$$\mathbf{Decryption}(Sk', Ct_x, K_y) = \Omega \prod_{1 \le i \le m} e(X_i, Y_i') e(W_i, L_i').$$

この復号演算の正当性を示す．復号演算は，検索属性が 1 のとき $e(X_i, Y_i')e(W_i, L_i') = e(T_i^{s-s_i}, g_p^{a_i/t_i}) e(V_i^{s_i}, g_p^{a_i/v_i}) = e(g_p, g_p)^{a_i(s-s_i)} e(g_p, g_p)^{a_i s_i} = e(g_p, g_p)^{a_i s}$ となり，検索属性が 0 のとき $e(X_i, Y_i')e(W_i, L_i') = e(g_p, g_p)^{a_i s}$ となり，検索属性が*のとき $e(X_i, Y_i')e(W_i, L_i') = 1 \cdot 1 = 1$ となる．したがって，検索抽出したい暗号文において，以下のように復号が成功する．

$$\mathbf{Decryption}(Sk', Ct_x, K_y) = \Omega \prod_{1 \le i \le m} e(X_i, Y_i') e(W_i, L_i') = \Omega \Big(\prod_{i \in S_y} e(g_p, g_p)^{a_i s} \Big)$$
$$= M \cdot e(g_p, g_p)^{-\gamma s} e(g_p, g_p)^{\gamma s} = M.$$

なお，式変形に $\Omega = M \cdot \Gamma^{-s} = M \cdot e(g_p, g_p)^{-s\gamma}$，$\Sigma_{i \in S_y} a_i = \gamma$ が成り立つことを用いた．

この提案方式では，サーバ公開鍵 Pk' が g 以外の Pk 中の値に依存して生成されないため，サーバはユーザごとに異なる Pk' を準備する必要がない．**Definition 1** の直後に記載した性質に加えて，この性質を有する提案方式は E メールなどの状況に適しているといえる．つまり，送信者側で受信者の Pk を用いて作成された検索機能付き暗号文は，ネットワーク上を転送された後に受信者が使用している（メール）サーバに届き，受信者は，自己のマスタ秘密鍵 Msk とサーバ公開鍵 Pk' などを用いて検索クエリを作成して検索を行うことにより，検索条件を満たす自己宛に暗号化されたメールのみを取り出すことができる．

5. 安全性証明

本章では，提案方式の安全性を証明する．提案方式が満たす安全性はワイルドカード秘匿，意味論的安全性，属性秘匿の 3 つである．このうち，意味論的安全性と属性秘匿は既存方式[3),5)] において満たされている性質である．そこで，本章では特にワイルドカード秘匿について示し，意味論的安全性，属性秘匿に関しては付録で詳細な安全性の議論を行う．

Theorem 1. 双線形写像が定義される巡回群 $\mathbb{G}$ について部分群判定問題が計算困難であるならば，提案方式はサーバ以外の通信路上の盗聴者に対してワイルドカード秘匿を満たす．

Proof. 今，部分群判定問題に対する攻撃者 $\mathcal{B}$ を考える．攻撃者 $\mathcal{B}$ は **WildCardHiding** $\mathbf{Exp}_{\mathcal{A}}(1^k)$ において *non-negligible* なアドバンテージ $\epsilon(k)$ を持つ攻撃者 $\mathcal{A}$ を利用して，以下のように動作する．

Input. 攻撃者 $\mathcal{B}$ は部分群判定問題のチャレンジとして，$[n, \mathbb{G}, \mathbb{G}_T, e, x']$ を得る．ただし，

$x \leftarrow \mathbb{G}$ に対して $x' = x$ or x^q である．すなわち x は $\mathbb{G}$ のランダムな要素か，あるいは $\mathbb{G}_p$ のランダムな要素である．

Init. 攻撃者 $\mathcal{A}$ はチャレンジ用の 2 つの異なる検索クエリ属性ベクトル (y_0, y_1) を選択し，これを攻撃者 $\mathcal{B}$ に渡す．

Setup. 攻撃者 $\mathcal{B}$ は以下のように公開鍵 Pk，サーバ公開鍵 Pk' を生成し，Pk を攻撃者 $\mathcal{A}$ に渡して動作させる．

(1) $\gamma \in \mathbb{Z}_n$ をランダムに選択し，$\Gamma = e(x', x')^\gamma$ を計算する．

(2) $1 \le i \le m$（$m = poly(k)$）について，t_i，v_i，r_i，$m_i \in \mathbb{Z}_n$ をランダムに選択し，$T_i = x'^{t_i}$，$V_i = x'^{v_i}$，$R_i = x'^{r_i}$，$M_i = x'^{m_i}$ とする．

(3) α，$\beta \in \mathbb{Z}_n$ をランダムに選択する．

(4) $\mathbb{G}$ の生成元 g をランダムに選択する．

(5) $Pk = [x', g^\beta, \mathcal{I}, \Gamma, (T_i, V_i, R_i, M_i)_{i=1}^m]$，$Pk' = g^\alpha$ とする．

Query Phase 1. 攻撃者 $\mathcal{A}$ による，検索属性ベクトル y の秘密鍵要求クエリに対して，$\mathcal{B}$ は以下の手順に従って適切に K_y を生成し，これを $\mathcal{A}$ に返す．

(1) $1 \le i \le m$ について $rnd_{i,0}$，$rnd_{i,1} \in \mathbb{Z}_n$ をランダムに選択する．

(2) $i \in S_y$ について，$a_i \in \mathbb{Z}_n$ をランダムに選択する．ただし，$\sum_{i \in S_y} a_i = \gamma$ を満たす．

(3) $K_y = [(g^{rnd_{i,0}}, g^{rnd_{i,1}}, Y_i, L_i)_{i=1}^m]$ を計算する．ただし，$(Y_i, L_i)_{i=1}^m$ は以下のとおり定める．

$$Y_i = \begin{cases} x'^{a_i/t_i} g^{\alpha \cdot rnd_{i,0}}, & \text{if } y_i = 1 \\ x'^{a_i/r_i} g^{\alpha \cdot rnd_{i,0}}, & \text{if } y_i = 0 \\ g^{\alpha \cdot \beta \cdot rnd_{i,0}}, & \text{if } y_i = *, \end{cases}$$

$$L_i = \begin{cases} x'^{a_i/v_i} g^{\alpha \cdot rnd_{i,1}}, & \text{if } y_i = 1 \\ x'^{a_i/m_i} g^{\alpha \cdot rnd_{i,1}}, & \text{if } y_i = 0 \\ g^{\alpha \cdot \beta \cdot rnd_{i,1}}, & \text{if } y_i = *. \end{cases}$$

Challenge. 攻撃者 $\mathcal{A}$ は 2 つの検索属性ベクトル y_0，y_1 を選択し，これを $\mathcal{B}$ へ渡す．攻撃者 $\mathcal{B}$ はランダムに 1 ビット $\eta \in \{0, 1\}$ を選択し，Step (3) の手順に従って K_{y_η} を生成し，これを $\mathcal{A}$ に返す．

Query Phase 2. $\mathcal{A}$ は Query Phase 1 と同様に動作する．

Output. $\mathcal{A}$ は 1 ビット $\eta' \in \{0, 1\}$ を選択し，これを攻撃者 $\mathcal{B}$ へ返す．攻撃者 $\mathcal{B}$ は $\eta = \eta'$ ならば 0 を，そうでなければ 1 を出力する．

ここで，$x' \in \mathbb{G}$ のとき，チャレンジ K_{y_η} はビット η に独立して $\mathbb{G}$ 上に一様ランダムに分布する．したがって，$\Pr[\eta = \eta'] = 1/2$ である．一方，$x' \in \mathbb{G}_p$ のとき，公開鍵 Pk とチャレンジ K_{y_η} は $\mathbf{WildCardHidingExp}_{\mathcal{A}}(1^k)$ におけるものと同一である．したがって，$\Pr[\eta = \eta'] > 1/2 + \epsilon(k)$ で与えられる．すなわち，$\mathcal{B}$ の部分群判定問題に対するアドバンテージ $SDAdv_{\mathcal{B}} = \epsilon(k)$ で表される．今，部分群判定問題の困難性を仮定するので $SDAdv_{\mathcal{B}} = \epsilon(k) \le negl$ である．

したがって，部分群判定問題が困難であるならば，$\epsilon \le \nu(k)$ となるので，提案方式はワイルドカード秘匿である．□

以上のように，提案方式はサーバ以外の攻撃者に対するワイルドカード秘匿を保証でき，従来方式[3),5)] に存在した検索内容漏洩問題を部分的に解決できる．すなわち，通信路上においてサーバを除くあらゆる確率的多項式時間攻撃者に対してワイルドカードを秘匿できる．しかし，サーバに対しては依然として検索内容の漏洩問題が存在する．サーバはサーバ秘密鍵 $Sk' = \alpha$，公開鍵 Pk に含まれる g^β，検索クエリ K_y に含まれる $g^{rand_{i,0}}$，$g^{rand_{i,1}}$ と Y_i，Li より，$e(Y_i, g) = e(g^\beta, (g^{rand_{i,0}})^\alpha)$，$e(L_i, g) = e(g^\beta, (g^{rand_{i,1}})^\alpha)$ が成立するか否かを確認することにより，y_i がワイルドカードであるか否かを特定することができる．サーバを含めた他者に対して検索内容を秘匿しながら検索を行える方が安全性が高いため，サーバに対してもワイルドカードを秘匿することが今後の課題となる．

提案方式の意味論的安全性，属性秘匿に関しては，付録で詳細な議論を行う．本章では，それぞれの定理と，安全性証明の方針について簡潔に示す．なお，提案方式は Iovino らの方式[5)] を基本として合成数位数の群上で定義される双線形写像を用いた構成となっているため，意味論的安全性，属性秘匿の証明の方針も同様のものとなる．

Theorem 2. **cDBDH** 問題が計算困難であるならば，提案方式は意味論的に安全である．

（証明概要）**SemanticExp** において *non-negligible* なアドバンテージを持つ攻撃者 $\mathcal{A}$ と，$\mathcal{A}$ を利用する **cDBDHExp** における攻撃者 $\mathcal{B}$ を仮定する．$\mathcal{B}$ は $\mathcal{A}$ によって最初に選ばれた暗号文属性ベクトル x と Query Phase で $\mathcal{A}$ が選択する検索属性ベクトル y の属性が同一の場合にのみ **cDBDH** の入力 A，および B が埋め込まれた値を返答する．さらに，$\mathcal{B}$ は $\mathcal{A}$ へのチャレンジの中に C，および Z を埋め込む．このとき，チャレンジは $Z = e(g_p, g_p)^{abc}$ のとき，**Encryption** の出力と同一の分布を持つように構成する．これにより，$\mathcal{A}$ のアドバンテージを持って **cDBDH** を解く攻撃者 $\mathcal{B}$ を構成できる．

Theorem 3. **cDL** 問題が困難であるならば，提案方式は属性秘匿である．

（証明概要）提案方式の属性秘匿を証明するために，以下の分布 **Dist** を定義する．

$\mathbf{Dist}_j(x, M)$

(1)　$\mathcal{I} = [n, \mathbb{G}, \mathbb{G}_T, g, e]$ を選択する．

(2)　$\mathbf{Setup}(1^k, n)$ を実行し，(Msk, Pk) を得る．

(3)　$R_0 \in \mathbb{G}_T$，$s \in \mathbb{Z}_n$ をランダムに選択する．また，$C_0 = g_p^s$ とする．

(4)　$i = 1, \ldots, j$ について，X_i，$W_i \in \mathbb{G}_p$ をランダムに選択する．

(5)　$i = j+1, \ldots, m$ について，r，$s_i \in \mathbb{Z}_n$ をランダムに選択し，X_i，W_i を以下のとおりに計算する．

$$X_i = \begin{cases} T_i^{s-s_i}, & \text{if } x_i = 1 \\ R_i^{s-s_i}, & \text{if } x_i = 0, \end{cases} \quad W_i = \begin{cases} V_i^{s_i}, & \text{if } x_i = 1 \\ M_i^{s_i}, & \text{if } x_i = 0 \end{cases}$$

(6)　$(R_0, C_0, (X_i, W_i)_{i=1}^m)$ を出力する．

これを用いて，以下の **Lemma** を示す．

Lemma 3.1. **cDL** 仮定が成り立つならば，$\ell = 1, 2, \ldots, m$ とすべての $x \in \{0,1\}^m$ について，**KeyGeneration** にアクセス可能な攻撃者にとって $\mathbf{Dist}_\ell$ と $\mathbf{Dist}_{\ell-1}$ の分布は計算量的に識別不可能である．

提案方式の意味論的安全性より，**cDBDH** 仮定の下で $\mathbf{Dist}_0(x, M)$ と $\mathbf{Encryption}(M, x, Pk)$ は計算量的に識別不可能であることがいえる．これに対して，$\mathbf{Dist}_m$ はすべての $(X_i, W_i)_{i=1}^m$ が $\mathbb{G}_p$ 上に一様ランダムに分布する．すなわち，(x, M) とは独立した $x' \neq x$，$M' \neq M$ を満たす新たな暗号文属性ベクトルと平文の組 (x', M') に対するチャレンジの分布と見ることができる．したがって，前述の意味論的安全性と上記の **Lemma** を満たすことは提案方式が属性秘匿を満たすことを意味する．

6. おわりに

本論文では，既存方式[3),5)] に存在した検索クエリ中のワイルドカードが漏洩するという問題について，公開鍵暗号系の環境における Iovino らの方式[5)] を参考に，通信路上の盗聴者に対する検索クエリ中のワイルドカードの守秘性を保証する公開鍵暗号系の環境における方式を提案した．新たにワイルドカード秘匿を定義し，提案方式がこのワイルドカード秘匿を満たすこと，および既存方式で保証されている意味論的安全性や属性秘匿も満たすことを証明した．今後の課題として，サーバに対する検索クエリ中のワイルドカードの守秘性を保証することがあげられる．

謝辞　貴重なコメントをいただいた査読者ならびに編集担当者に，謹んで感謝の意を表する．

参考文献

1) Boneh, D., Di Crescenzo, G., Ostrovsky, R. and Persiano, G.: Public Key Encryption with Keyword Search, *EUROCRYPT 2004*, LNCS Vol.3027, pp.506–522, Springer, Heidelberg (2004).
2) Boneh, D., Goh, E.-J. and Nissim, K.: Evaluating 2-DNF Formulas on Ciphertexts, *TCC 2005*, LNCS 3378, pp.325–341, Springer, Heidelberg (2005).
3) Boneh, D. and Waters, B.: Conjunctive, subset and range queries on encrypted data, *TCC 2007*, LNCS Vol.4392, pp.535–554, Springer, Heidelberg (2007).
4) Goh, E.-J.: Secure Indexes, Cryptology ePrint Archive, Report 2003/216 (2003).
5) Iovino, V. and Persiano, G.: Hidden-Vector Encryption with Groups of Prime Order, *Pairing 2008*, LNCS Vol.5209, pp.75–88, Springer, Heidelberg (2008).
6) Park, D.J., Kim, K. and Lee, P.J.: Public Key Encryption with Conjunctive Field Keyword Search, *WISA 2004*, LNCS Vol.3325, pp.73–86, Springer, Heidelberg (2004).
7) Hwang, Y.H. and Lee, P.J.: Public Key Encryption with Conjunctive Keyword Search and Its Extension to a Multi-user System, *Pairing 2007*, LNCS Vol.4575, pp.2–22, Springer-Verlag, Berlin, Heidelberg (2007).

付　録

A.1　意味論的安全性（Theorem 2 の証明）

Proof. 今，**cDBDH** に対する攻撃者 $\mathcal{B}$ を考える．攻撃者 $\mathcal{B}$ は $\mathbf{SemanticExp}_{\mathcal{A}}(1^k)$ において *non-negligible* なアドバンテージ $\epsilon(k)$ を持つ攻撃者 $\mathcal{A}$ を利用して，以下のように動作する．

Input.　攻撃者 $\mathcal{B}$ は **cDBDH** のチャレンジとして，$[g_p, g_q, \mathcal{I} = (n, \mathbb{G}, \mathbb{G}_T, e), A = g_p^a, B = g_p^b, C = g_p^c, Z]$ を得る．ただし，Z は $e(g_p, g_p)^{abc}$ もしくは $\mathbb{G}_T$ のランダムな要素である．また，g_p，g_q はそれぞれ $\mathbb{G}$ の部分群 $\mathbb{G}_p$，$\mathbb{G}_q$ の生成元である．

Init.　攻撃者 $\mathcal{B}$ は攻撃者 $\mathcal{A}$ を動作させ，チャレンジに用いる暗号文属性ベクトル x を得る．

Setup.　攻撃者 $\mathcal{B}$ は以下のように公開鍵 Pk，サーバ公開鍵 Pk' を生成し，Pk を攻撃者 $\mathcal{A}$ に渡して動作させる．

(1)　$\Gamma = e(A, B)$ とする．すなわち，$\gamma = ab$ である．

（2）　$1 \le i \le m$ について t'_i，v'_i，r'_i，$m'_i \in \mathbb{Z}_n$ をランダムに選択し，T_i，V_i，R_i，M_i を以下のように計算する．

$$T_i = \begin{cases} g_p^{t'_i}, & \text{if } x_i = 1 \\ B^{t'_i}, & \text{if } x_i = 0, \end{cases} \quad V_i = \begin{cases} g_p^{v'_i}, & \text{if } x_i = 1 \\ B^{v'_i}, & \text{if } x_i = 0, \end{cases}$$

$$R_i = \begin{cases} B^{r'_i}, & \text{if } x_i = 1 \\ g_p^{r'_i}, & \text{if } x_i = 0, \end{cases} \quad M_i = \begin{cases} B^{m'_i}, & \text{if } x_i = 1 \\ g_p^{m'_i}, & \text{if } x_i = 0. \end{cases}$$

ここで，$x_i = 1$ のときは $t_i = t'_i$，$v_i = v'_i$，$r_i = br'_i$，$m_i = bm'_i$ であり，$x_i = 0$ のときは $t_i = bt'_i$，$v_i = bv'_i$，$r_i = r'_i$，$m_i = m'_i$ である．

（3）　$\mathbb{G}$ の生成元 g をランダムに選択する．

（4）　α，$\beta \in \mathbb{Z}_n$ をランダムに選択する．

（5）　$Pk = [g_p, g^\beta, \mathcal{I}, \Gamma, (T_i, V_i, R_i, M_i)_{i=1}^m]$，$Pk' = g^\alpha$ とする．

Query Phase 1.　攻撃者 $\mathcal{A}$ による，$P_x(y) = 0$ を満たす検索属性ベクトル y の秘密鍵要求クエリに対して，$\mathcal{B}$ は以下の手順に従って適切に K_y を生成し，これを $\mathcal{A}$ に返す．

（1）　$x_j \ne y_j$ かつ $y_j \ne *$ を満たす j を選択する．

（2）　すべての $i \ne j$ かつ $y_i \ne *$ について $a'_i \in \mathbb{Z}_n$ をランダムに選択し，$a' = \Sigma a'_i$ とする．

（3）　$1 \le i \le m$ について $rnd_{i,0}$，$rnd_{i,1} \in \mathbb{Z}_n$ をランダムに選択する．

（4）　Y_j，L_j を以下のように計算する．

$$Y_j = \begin{cases} A^{1/t'_j} g_p^{-a'/t'_j} g^{\alpha \cdot rnd_{i,0}}, & \text{if } y_j = 1 \\ A^{1/r'_j} g_p^{-a'/r'_j} g^{\alpha \cdot rnd_{i,0}}, & \text{if } y_j = 0, \end{cases}$$

$$L_j = \begin{cases} A^{1/v'_j} g_p^{-a'/v'_j} g^{\alpha \cdot rnd_{i,1}}, & \text{if } y_j = 1 \\ A^{1/m'_j} g_p^{-a'/m'_j} g^{\alpha \cdot rnd_{i,1}}, & \text{if } y_j = 0. \end{cases}$$

（5）　$i \ne j$ を満たす Y_i，L_i を以下のように計算する．

$$Y_i = \begin{cases} B^{a'_i/t'_i} g^{\alpha \cdot rnd_{i,0}}, & \text{if } x_i = y_i = 1 \\ B^{a'_i/r'_i} g^{\alpha \cdot rnd_{i,0}}, & \text{if } x_i = y_i = 0 \\ g_p^{a'_i/r'_i} g^{\alpha \cdot rnd_{i,0}}, & \text{if } x_i = 1, y_i = 0 \\ g_p^{a'_i/t'_i} g^{\alpha \cdot rnd_{i,0}}, & \text{if } x_i = 0, y_i = 1 \\ g^{\alpha \cdot \beta \cdot rnd_{i,0}}, & \text{if } y_i = *, \end{cases}$$

$$L_i = \begin{cases} B^{a'_i/v'_i} g^{\alpha \cdot rnd_{i,1}}, & \text{if } x_i = y_i = 1 \\ B^{a'_i/m'_i} g^{\alpha \cdot rnd_{i,1}}, & \text{if } x_i = y_i = 0 \\ g_p^{a'_i/m'_i} g^{\alpha \cdot rnd_{i,1}}, & \text{if } x_i = 1, y_i = 0 \\ g_p^{a'_i/v'_i} g^{\alpha \cdot rnd_{i,1}}, & \text{if } x_i = 0, y_i = 1 \\ g^{\alpha \cdot \beta \cdot rnd_{i,1}}, & \text{if } y_i = *. \end{cases}$$

ここで，K_y の分布は **KeyGeneration** の出力の分布と同一である．すなわち，$a_i = ba'_i$，$a_j = b(a - a')$ となるので，$\Sigma_{i \in S_y} a_i = ab = \gamma$ の形式をとる．

Challenge.　$\mathcal{A}$ はランダムに M_0，$M_1 \in \mathbb{G}_T$ を選択し，これを $\mathcal{B}$ へ渡す．$\mathcal{B}$ はランダムに $\eta \in \{0, 1\}$，$s_i \in \mathbb{Z}_n$（$1 \le i \le m$），$r \in \mathbb{Z}_n$ を選択する．そして，$\mathcal{B}$ は以下のように $Ct_x = [\Omega, (X_i, W_i)_{i=1}^m]$ を計算し，これを $\mathcal{A}$ に渡す．

$$\Omega = M_\eta Z^{-1},\ X_i = \begin{cases} C^{t'_i} g_p^{-t'_i s_i}, & \text{if } x_i = 1 \\ C^{r'_i} g_p^{-r'_i s_i}, & \text{if } x_i = 0, \end{cases} \quad W_i = \begin{cases} g_p^{v'_i s_i}, & \text{if } x_i = 1 \\ g_p^{m'_i s_i}, & \text{if } x_i = 0. \end{cases}$$

ここで，$s = c$ である．したがって，もし $Z = e(g_p, g_p)^{abc}$ ならば $\Omega = M_\eta \Gamma^{-s}$ となる．また，Ct_x は $s = c$ と M_η についての正しい暗号文の形式となっている．

Query Phase 2.　Query Phase 1 と同一．

Output.　$\mathcal{A}$ は $\eta' \in \{0, 1\}$ を出力し，$\mathcal{B}$ は $\eta = \eta'$ なら 0 を出力する．

もし $Z = e(g_p, g_p)^{abc}$ ならば，$\mathcal{B}$ が 0 を出力する確率は $\mathcal{A}$ の non-negligible なアドバンテージ $1/poly(k)$ を用いて $1/2 + 1/poly(k)$ で表される．一方で，もし Z が $\mathbb{G}_T$ のランダムな要素であるならば，$\mathcal{B}$ が 0 を出力する確率は 1/2 となる．これは **cDBDH** 仮定に矛盾する．したがって，**cDBDH** が困難であるならば，提案方式は意味論的に安全である．□

A.2　属性秘匿（Lemma 3.1 の証明）

Proof.　確率的多項式時間攻撃者 $\mathcal{A}$ が $\mathbf{Dist}_l(x)$ と $\mathbf{Dist}_{l-1}(x)$ が識別できると仮定する．ここで，攻撃者 $\mathcal{B}$ を **mDLExp** への攻撃者とし，$\mathcal{A}$ を利用して以降のように動作する．

Input. $\mathcal{B}$ は $[\mathcal{I}, g, g_p, Z_1 = g_p^{z_1}, Z_2 = g_p^{z_2}, Z_{13} = g_p^{z_1 z_3}, U = g_p^u, Z]$ をチャレンジ入力として得る．ここで，Z は $g_p^{z_2(u-z_3)}$ もしくは $\mathbb{G}_T$ 上のランダムな要素であり，$\mathcal{I} = [n, \mathbb{G}, \mathbb{G}_T, e]$ である．

Init. $\mathcal{B}$ は $\mathcal{A}$ を動作させ，チャレンジ用の暗号文属性ベクトル x を得る．

Setup. 攻撃者 $\mathcal{B}$ は以下のように公開鍵 Pk，サーバ公開鍵 Pk' を生成し，Pk を攻撃者 $\mathcal{A}$ に渡して動作させる．

(1)　$\Gamma = e(Z_1, Z_2)$ とする．すなわち，$\gamma = z_1 z_2$ である．

(2)　$1 \le i \le m$ について t'_i，v'_i，r'_i，$m'_i \in \mathbb{Z}_n$ をランダムに選択し，T_i，V_i，R_i，M_i を以下のように計算する．

$$T_l = \begin{cases} Z_2^{t'_l}, & \text{if } x_l = 1 \\ Z_1^{t'_l}, & \text{if } x_l = 0, \end{cases} \quad V_l = \begin{cases} Z_1^{v'_l}, & \text{if } x_l = 1 \\ Z_1^{v'_l}, & \text{if } x_l = 0, \end{cases}$$

$$R_l = \begin{cases} Z_1^{r'_l}, & \text{if } x_l = 1 \\ Z_2^{r'_l}, & \text{if } x_l = 0, \end{cases} \quad M_l = \begin{cases} Z_1^{m'_l}, & \text{if } x_l = 1 \\ Z_1^{m'_l}, & \text{if } x_l = 0. \end{cases}$$

さらに，$i \neq l$ について，

$$T_i = \begin{cases} g_p^{t'_i}, & \text{if } x_i = 1 \\ Z_1^{t'_i}, & \text{if } x_i = 0, \end{cases} \quad V_i = \begin{cases} g_p^{v'_i}, & \text{if } x_i = 1 \\ Z_1^{v'_i}, & \text{if } x_i = 0, \end{cases}$$

$$R_i = \begin{cases} Z_1^{r'_i}, & \text{if } x_i = 1 \\ g_p^{r'_i}, & \text{if } x_i = 0, \end{cases} \quad M_i = \begin{cases} Z_1^{m'_i}, & \text{if } x_i = 1 \\ g_p^{m'_i}, & \text{if } x_i = 0. \end{cases}$$

ここで，$\gamma = z_1 z_2$ であり，$i \neq l$ かつ $x_i = 1$ のときは $t_i = t'_i$，$v_i = v'_i$，$r_i = z_1 r'_i$，$m_i = z_1 m'_i$，$i \neq l$ かつ $x_i = 0$ のときは $x_i = 1$，$t_i = z_1 t'_i$，$v_i = z_1 v'_i$，$r_i = r'_i$，$m_i = m'_i$ である．さらに，$x_l = 1$ のときは $t_i = z_2 t'_i$，$v_i = z_1 v'_i$，$r_i = z_2 r'_i$，$m_i = z_1 m'_i$，$x_l = 0$ のときは $t_i = z_1 t'_i$，$v_i = z_1 v'_i$，$r_i = z_1 r'_i$，$m_i = z_1 m'_i$ となる．

(3)　$\mathbb{G}$ の生成元 g をランダムに選択する．

(4)　α，$\beta \in \mathbb{Z}_n$ をランダムに選択する．

(5)　$Pk = [g_p, g^\beta, \mathcal{I}, \Gamma, (T_i, V_i, R_i, M_i)_{i=1}^m]$，$Pk' = g^\alpha$ とする．

Query Phase 1. $\mathcal{B}$ は $\mathcal{A}$ による $P_x(y) = 0$ を満たす検索属性ベクトル y についての秘密鍵要求クエリに対して，以下のように返答する．

Case1: $x_l = y_l$ or $y_l = *$ のとき． この場合，$x_j \neq y_j$ かつ $y_j \neq *$ となる $j \neq l$ が存在する．

(1)　$i \neq j$ について $a'_i \in \mathbb{Z}_n$ をランダムに選択する．また，このとき，$a' = \Sigma_{i \neq j, l} a'_i$ とする．

(2)　$1 \le i \le m$ について $rnd_{i,0}$，$rnd_{i,1} \in \mathbb{Z}_n$ をランダムに選択する．

(3)　$i \neq j$ かつ $i \neq l$ を満たす i について，以下を計算する．

$$Y_i = \begin{cases} Z_1^{a'_i/t'_i} g_p^{\alpha \cdot rnd_{i,0}}, & \text{if } x_i = y_i = 1 \\ Z_1^{a'_i/r'_i} g_p^{\alpha \cdot rnd_{i,0}}, & \text{if } x_i = y_i = 0 \\ g_p^{a'_i/r'_i} g_p^{\alpha \cdot rnd_{i,0}}, & \text{if } x_i = 1, y_i = 0 \\ g_p^{a'_i/t'_i} g_p^{\alpha \cdot rnd_{i,0}}, & \text{if } x_i = 0, y_i = 1 \\ g_p^{\alpha \cdot \beta \cdot rnd_{i,0}}, & \text{if } y_i = *, \end{cases}$$

$$L_i = \begin{cases} Z_1^{a'_i/v'_i} g_p^{\alpha \cdot rnd_{i,1}}, & \text{if } x_i = y_i = 1 \\ Z_1^{a'_i/m'_i} g_p^{\alpha \cdot rnd_{i,1}}, & \text{if } x_i = y_i = 0 \\ g_p^{a'_i/m'_i} g_p^{\alpha \cdot rnd_{i,1}}, & \text{if } x_i = 1, y_i = 0 \\ g_p^{a'_i/v'_i} g_p^{\alpha \cdot rnd_{i,1}}, & \text{if } x_i = 0, y_i = 1 \\ g_p^{\alpha \cdot \beta \cdot rnd_{i,1}}, & \text{if } y_i = *. \end{cases}$$

(4)　$i = l$ について，以下を計算する．

$$Y_l = \begin{cases} Z_1^{a'_l/t'_l} g_p^{\alpha \cdot rnd_{i,0}}, & \text{if } y_l = 1 \\ Z_1^{a'_l/r'_l} g_p^{\alpha \cdot rnd_{i,0}}, & \text{if } y_l = 0 \\ g_p^{\alpha \cdot \beta \cdot rnd_{i,0}}, & \text{if } y_l = *, \end{cases}$$

$$L_l = \begin{cases} Z_2^{a'_l/v'_l} g_p^{\alpha \cdot rnd_{i,1}}, & \text{if } y_l = 1 \\ Z_2^{a'_l/m'_l} g_p^{\alpha \cdot rnd_{i,1}}, & \text{if } y_l = 0 \\ g_p^{\alpha \cdot \beta \cdot rnd_{i,1}}, & \text{if } y_l = *. \end{cases}$$

(5)　$i = j$ について，以下を計算する．

$$Y_j = \begin{cases} Z_2^{(1-a'_l)/t'_j} g^{-a'/t'_j} g_p^{\alpha \cdot rnd_{i,0}}, & \text{if } y_j = 1 \\ Z_1^{(1-a'_l)/r'_j} g^{-a'/r'_j} g_p^{\alpha \cdot rnd_{i,0}}, & \text{if } y_j = 0 \\ g_p^{\alpha \cdot \beta \cdot rnd_{i,0}}, & \text{if } y_l = *, \end{cases}$$

$$L_j = \begin{cases} Z_2^{(1-a'_l)/v'_j} g^{-a'/v'_j} g_p^{\alpha \cdot rnd_{i,1}}, & \text{if } y_j = 1 \\ Z_1^{(1-a'_l)/m'_j} g^{-a'/m'_j} g_p^{\alpha \cdot rnd_{i,1}}, & \text{if } y_j = 0 \\ g_p^{\alpha \cdot \beta \cdot rnd_{i,1}}, & \text{if } y_l = *. \end{cases}$$

このとき，$i \neq j$ かつ $i \neq l$ について $a_i = z_1 a'_i$，$a_l = z_1 z_2 a'_l$，$a_j = z_1 z_2 - z_1 z_2 a'_l - z_1 a'$ となる．したがって，a_i は **KeyGeneration** において $z_1 z_2 = \gamma$ とした場合と同一の分布を持つ．

Case 2: $x_l \neq y_l$ **かつ** $y_l \neq *$ **のとき．** 以下のように動作する．

(1)　$i \neq l$ について $a'_i \in \mathbb{Z}_n$ をランダムに選択する．このとき，$a' = \Sigma_{i \neq l} a'_i$ とする．

(2)　$1 \leq i \leq m$ について $rnd_{i,0}$，$rnd_{i,1} \in \mathbb{Z}_n$ をランダムに選択する．

(3)　$i \neq j$ かつ $i \neq l$ について，以下を計算する．

$$Y_i = \begin{cases} Z_1^{a'_i/t'_i} g_p^{\alpha \cdot rnd_{i,0}}, & \text{if } x_i = y_i = 1 \\ Z_1^{a'_i/r'_i} g_p^{\alpha \cdot rnd_{i,0}}, & \text{if } x_i = y_i = 0 \\ g_p^{a'_i/r'_i} g_p^{\alpha \cdot rnd_{i,0}}, & \text{if } x_i = 1, y_i = 0 \\ g_p^{a'_i/t'_i} g_p^{\alpha \cdot rnd_{i,0}}, & \text{if } x_i = 0, y_i = 1 \\ g_p^{\alpha \cdot \beta \cdot rnd_{i,0}}, & \text{if } y_i = *, \end{cases}$$

$$L_i = \begin{cases} Z_1^{a'_i/v'_i} g_p^{\alpha \cdot rnd_{i,1}}, & \text{if } x_i = y_i = 1 \\ Z_1^{a'_i/m'_i} g_p^{\alpha \cdot rnd_{i,1}}, & \text{if } x_i = y_i = 0 \\ g_p^{a'_i/m'_i} g_p^{\alpha \cdot rnd_{i,1}}, & \text{if } x_i = 1, y_i = 0 \\ g_p^{a'_i/v'_i} g_p^{\alpha \cdot rnd_{i,1}}, & \text{if } x_i = 0, y_i = 1 \\ g_p^{\alpha \cdot \beta \cdot rnd_{i,1}}, & \text{if } y_i = *. \end{cases}$$

(4)　$i = l$ について，以下を計算する．

$$Y_l = \begin{cases} Z_2^{1/t'_l} g^{-a'/t'_l} g_p^{\alpha \cdot rnd_{i,0}}, & \text{if } y_l = 1 \\ Z_2^{1/r'_l} g^{-a'/r'_l} g_p^{\alpha \cdot rnd_{i,0}}, & \text{if } y_l = 0, \end{cases}$$

$$L_l = \begin{cases} Z_2^{1/v'_l} g^{-a'/v'_l} g_p^{\alpha \cdot rnd_{i,1}}, & \text{if } y_l = 1 \\ Z_2^{1/m'_l} g^{-a'/m'_l} g_p^{\alpha \cdot rnd_{i,1}}, & \text{if } y_l = 0. \end{cases}$$

ここで，$a_i = z_1 a'_i$，$a_l = z_1 z_2 - z_1 a'$ である．したがって，a_i は **KeyGeneration** において $z_1 z_2 = \gamma$ とした場合と同一の分布を持つ．

Challenge.　$\mathcal{B}$ は $R_0 \in \mathbb{G}_T$ をランダムに選択し，また $l \leq i \leq m$ について r，$s'_i \in \mathbb{Z}_n$ をランダムに選択する．このとき，$\mathcal{B}$ は以下のタプルを計算する．

$D^* = (R_0, (X_i, W_i)_{i=1}^m)$.

ここで，$i < l$ について X_i，$W_i \in \mathbb{G}$ をランダムに選択する．一方，$i \leq l$ について $\mathcal{B}$ は以下の計算をする．

$$X_i = \begin{cases} Z^{t'_l}, & \text{if } i = l, x_i = 1 \\ Z^{r'_l}, & \text{if } i = l, x_i = 0 \\ U^{t'_l} g^{-t'_i s'_i}, & \text{if } i > l, x_i = 1 \\ U^{r'_l} g^{-r'_i s'_i}, & \text{if } i > l, x_i = 0, \end{cases} \quad W_i = \begin{cases} Z_{13}^{v'_l}, & \text{if } i = l, x_i = 1 \\ Z_{13}^{m'_l}, & \text{if } i = l, x_i = 0 \\ g_p^{v'_i s'_i}, & \text{if } i > l, x_i = 1 \\ g_p^{m'_i s'_i}, & \text{if } i > l, x_i = 0. \end{cases}$$

ここで，もし $Z = g^{z_2(u-z_3)}$ ならば D^* は $s = u$，$s_l = z_3$，$s_i = s'_i$（$i > l$）とした $\mathbf{Dist}_{l-1}(x)$ の分布に従う．一方で，もし Z がランダムな要素ならば，D^* は $s = u$，$s_l = z_3$，$s_i = s'_i$（$i > l$）とした $\mathbf{Dist}_l(x)$ の分布に従う．

Quary Phase 2.　Query Phase 1 と同一．

Output.　最終的に，$\mathcal{A}$ は $\eta \in \{0, 1\}$ を出力する．ここで，$\eta = 0$ は D_{l-1} を示し，$\eta = 1$ は D_l を示す．$\mathcal{B}$ は η' を出力する．

もし $Z = g^{z_2(u-z_3)}$ ならば，$\mathcal{A}$ のビューは $\mathbf{Dist}_{l-1}(x)$ におけるビューと同一．また，もし Z がランダムな要素ならば，$\mathcal{A}$ のビューは $\mathbf{Dist}_l(x)$ におけるビューと同一．したがって，もし $\mathcal{A}$ が $\mathbf{Dist}_l(x)$ と $\mathbf{Dist}_{l-1}(x)$ を識別できるならば，$\mathcal{B}$ は **DL** 問題を解く．これは **DL** 仮定に矛盾する．

□

(平成 22 年 12 月 2 日受付)
(平成 23 年 6 月 3 日採録)

秋山　浩岐

昭和 61 年生．平成 21 年筑波大学第三学群情報学類卒業．平成 23 年筑波大学大学院システム情報工学研究科博士前期課程修了．暗号学の研究に従事．同年日立ソリューションズ（株）入社．

満保　雅浩（正会員）

1988 年金沢大学工学部電気・情報工学科卒業．1993 年東京工業大学大学院理工学研究科博士後期課程修了．博士（工学）．同年北陸先端科学技術大学院大学助手．その後，東北大学助教授，筑波大学助教授・准教授を経て，2011 年より金沢大学教授．情報セキュリティの教育・研究に従事．

岡本　栄司（正会員）

1973 年東京工業大学工学部電子工学科卒業．1978 年同大学大学院博士課程修了．工学博士．同年日本電気中央研究所入社．その後，北陸先端科学技術大学院大学，東邦大学をへて，2002 年より筑波大学教授．情報セキュリティの教育・研究に従事．1990 年電子情報通信学会論文賞，1993 年本会ベストオーサ賞受賞，2008 年本会論文賞．著書『暗号理論入門』（共立出版），『電子マネー』（岩波書店）等．