Computer Security Symposium 2019
21 - 24 October 2019

証明可能安全な鍵集約検索可能暗号の構成と実装評価

上村 真弘[1,a)]　矢内 直人[1]　岡村 真吾[2]　ジェイソン ポール クルーズ[1]

概要：Cui ら (IEEE Trans. on Comp. 2016) によって提案された鍵集約検索可能暗号 (KASE) は，マルチユーザ設定において，単一の集約鍵を通じた効率的なアクセス制御およびデータ検索が可能である．しかし，著者らの知る限り，Cui らの研究および後続の研究において，安全性の定式化とその証明は正しく議論されていない．本稿では，KASE の安全性を定式化し，その安全性を満たす方式を提案する．具体的には，まずサーバが一台の設定において，暗号文と集約鍵に関して BDHE 仮定と DHE 仮定の下で証明可能安全な方式を示す．次に，サーバが二台の設定において，検索内容のプライバシーについても XDH 仮定の下で安全性の証明可能な方式を示す．さらに，この二つの方式に対して実装評価を行ったところ，5000 個のファイルに対して，前者の方式では 3 秒程度，後者の方式では 6 秒程度で検索が可能であることを確認した．

キーワード：鍵集約検索可能暗号, 検索可能暗号, 安全性の定式化，安全性証明，マルチユーザ設定

A Note in Provable Security of Key-Aggregate Searchable Encryption

Masahiro Kamimura[1,a)]　Naoto Yanai[1]　Shingo Okamura[2]　Jason Paul Cruz[1]

Abstract: Key-aggregate searchable encryption (KASE) proposed by Cui et al. (IEEE Trans. on Comp. 2016) is able to perform access control efficiently in the multi-user setting by a single aggregate key. However, to the best of our knowledge, Cui et al. and their subsequent works have never discussed the security correctly. In this paper, we discuss the security of KASE formally and propose provably secure schemes. More specifically, we first proposed a provably secure scheme with respect to encrypted files and aggregate keys under the BDHE assumption and the DHE assumption in the single-server setting, respectively. Next, in two-server setting, we propose a provably secure scheme which can guarantee the privacy for search, as well as encrypted files and aggregate keys under the XDH assumption. Furthermore, we evaluate the performance of our schemes and we show that search can be performed within about three seconds in the former scheme and within about six seconds in the latter scheme for 5000 files.

Keywords: Key-Aggregate Searchable Encryption, Searchable Encryption, Provable Security and Multi-User Setting.

1. 序章

1.1 研究背景

近年，多くの利用者が写真やビデオといった大量のデータを保存・共有する際にクラウドストレージサービスを利用するケースが増えている．このとき，データ流出に対する対策として，データを暗号化してからクラウドに保存する形式が一般的である．ここで，暗号化状態でのデータ利用の利便性が低下するという問題を克服するために，検索可能暗号 [4, 19] が提案された．検索可能暗号は，暗号化した平文を復号することなく平文の検索が可能な暗号方式である．これらの方式に加え，アクセス制御を伴うデータ共有に対する需要が高まっているため，データを利用する人物が複数人いる状況（マルチユーザ設定）における検索

[1] 大阪大学
Osaka University
[2] 奈良工業高等専門学校
National Institute of Technology, Nara College
[a)] m-kamimr@ist.osaka-u.ac.jp

可能暗号の研究 [3,11,17,20] が盛んである．これらの研究では，本来のデータの持ち主であるデータ保有者がデータの共有相手である利用者に対して鍵を与えることで，利用者が一定範囲内のデータの検索可能な方式が提案されている．しかし，これらの方式 [3,11,17,20] では，データ数が増えるほど利用者に与える鍵の数が増えて，データ保有者の鍵管理が煩雑になる．それに加え，利用者が増えるほどクラウドに保存する暗号文の数が増える．

この問題を克服する暗号技術として，Cui らは鍵集約検索可能暗号 (KASE) [8] を提案した．KASE は，単一の鍵を利用者に与えることで検索可能な方式であり，鍵管理が簡単かつ暗号文の数が利用者の人数に依存しない．この方式を利用することで，マルチユーザ設定においてクラウドストレージサービスを効率よく運用できるようになることが期待できる．しかしながら，Cui らの KASE は安全性に関する定式化も安全性証明もされていない．後発の研究 [12,13,15] においても，安全性証明まで議論されている方式は存在しない．

1.2 貢献

本稿の貢献は，安全性の定式化とその下での証明可能安全な KASE の構成を示したことである．この安全性の定式化と証明は，我々が新しく取り組んだ問題である．詳細は 3.3 節で述べるが，一般的に暗号方式は機能性と安全性に関してトレードオフの関係があるため，定式化した安全性を満たす KASE を構成することは難しい．

安全性については，キーワード秘匿性，集約鍵偽造不可能性とトラップドア秘匿性について議論している．本稿では，利用するサーバ台数に応じて 2 つの構成 (構成 1 と構成 2) を提案する．構成 1 においては，l-BDHE 仮定の下でキーワード秘匿性，l-DHE 仮定の下で集約鍵偽造不可能性を満足することを証明する．構成 2 においては，キーワード秘匿性，集約鍵偽造不可能性に加えて XDH 仮定の下でトラップドア秘匿性を満足することを証明する．

それに加えて，本稿ではこの 2 つの構成の実装評価を行った．具体的には，5000 個のファイルに対して構成 1 では 3 秒程度，構成 2 では 6 秒程度で検索可能であることを確認した．暗号化処理については，どちらも 1 秒程度で完了することを確認した．

1.3 関連研究

著者らの知る限り，KASE は [12,13,15] で提案されている．しかしながら，[12,13] は Cui らの方式 [8] と同様に安全性が証明されていない．[15] は安全性の定式化と証明について言及しているが，その定式化は不十分である．それに加えて，証明の中で秘密鍵と公開鍵の構成が対応付いておらず，正しい証明となっていない．提案されている構成も，双線型写像を二重に入力として取る形式となっていて，正当性を満足しない．また，KASE の特殊な研究として，Zhou ら [22] は遠隔にあるセンサデバイスが暗号化するという状況下での構成を提案している．しかし，この方式はデータ保有者目線で見ると鍵の管理が増えている．他にも，Patranabis ら [16] は KASE に類似した方式を提案している．しかし，この方式は利用者に鍵を与えることで検索可能とする機能を保有していないので，本稿で扱う問題とは別の問題と言える．

また，KASE と類似した暗号方式として，属性ベース検索可能暗号 (SABE) [14,21] がある．これは，属性情報を埋め込んだ鍵を利用者に配布することで，データに埋め込まれた属性と対応するデータの検索が可能となる方式である．しかし，SABE は配布する鍵の数が属性数に依存するため，鍵管理が増えることになる．

2. 準備

双線型写像 双線型写像 $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$ を次のように定義する．

(1) $\mathbb{G}, \mathbb{H}, \mathbb{G}_T$ は素数位数 p の乗法群である．

(2) $g \in \mathbb{G}, h \in \mathbb{H}$ は $\mathbb{G}, \mathbb{H}$ の生成元である．

(3) 双線型写像 $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$ は次の性質を持つ写像である．

- (a) 任意の値 $u \in \mathbb{G}, v \in \mathbb{H}$ と $a, b \in \mathbb{Z}_p$ に対して，$e(u^a, v^b) = e(u, v)^{ab}$ が成り立つ．
- (b) $e(g, h) \neq 1$ が成り立つ．
- (c) 任意の値 $u \in \mathbb{G}, v \in \mathbb{H}$ に対して，$e(u, v)$ は効率よく計算することができる．

ここで，もし $\mathbb{G} = \mathbb{H}$ ならば，対称双線型写像となる．同様に，もし $\mathbb{G} \neq \mathbb{H}$ ならば，非対称双線型写像となる．

安全性仮定 本稿で用いる安全性仮定は以下の通りである．

定義 1 ((ϵ, l)-BDHE 仮定) l-BDHE 問題 [6] は，対称双線型写像 $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ 上における問題である．この問題は，一様ランダムな $(g, h) \in \mathbb{G}$ と $\alpha \in \mathbb{Z}_p$ に対して $(g, h, g^{\alpha}, g^{\alpha^2}, ..., g^{\alpha^l}, g^{\alpha^{l+2}}, ..., g^{\alpha^{2l}}, Z)$ を入力として与え，$Z \in \mathbb{G}_T$ が $e(g^{\alpha^{l+1}}, h)$ もしくはランダムな値 R かどうかを決定する問題である．もし $|Pr[\mathcal{A}(g, h, \boldsymbol{y}_{g,\alpha,l}, e(g^{\alpha^{l+1}}, h)) = 0] - Pr[\mathcal{A}(g, h, \boldsymbol{y}_{g,\alpha,l}, R) = 0]| \geq \epsilon$ が成立するならば，多項式時間アルゴリズム $\mathcal{A}$ は l-BDHE 問題を ϵ のアドバンテージで解くと定義する．ただし，$\boldsymbol{y}_{g,\alpha,l} = (g^{\alpha}, g^{\alpha^2}, ..., g^{\alpha^l}, g^{\alpha^{l+2}}, ..., g^{\alpha^{2l}})$ である．もし l-BDHE 問題を ϵ で解く多項式時間アルゴリズムが存在しなければ，(ϵ, l)-BDHE 仮定は成立するとする．

定義 2 ((ϵ, l)-DHE 仮定) l-DHE 問題 [10] は，一様ランダムな $g \in \mathbb{G}$ と $\alpha \in \mathbb{Z}_p$ に対して $(g, g^{\alpha}, g^{\alpha^2}, ..., g^{\alpha^l}, g^{\alpha^{l+2}}, ..., g^{\alpha^{2l}})$ を入力として与え，$g^{\alpha^{l+1}}$ を出力する問題である．もし $Pr[\mathcal{A}(g, \boldsymbol{y}_{g,\alpha,l}, g^{\alpha^{l+1}})] \geq \epsilon$ が成立するならば，多項式時間アルゴリズム $\mathcal{A}$ は l-DHE

問題を ϵ のアドバンテージで解くと定義する．ただし，$\boldsymbol{y}_{g,\alpha,l} = (g^{\alpha}, g^{\alpha^2}, ..., g^{\alpha^l}, g^{\alpha^{l+2}}, ..., g^{\alpha^{2l}})$ である．もし l-DHE 問題を ϵ で解く多項式時間アルゴリズムが存在しなければ，(ϵ, l)-DHE 仮定は成立するとする．

定義 3 (XDH 仮定) XDH 問題 [2] は，一様ランダムな $g \in \mathbb{G}, h \in \mathbb{H}$ と $a, b \in \mathbb{Z}_p$ に対して (g, h, g^a, g^b, Z) を入力として与え，$Z \in \mathbb{G}$ が g^{ab} もしくはランダムな値 R かどうかを決定する問題である．もし $|Pr[\mathcal{A}(g, h, g^a, g^b, g^{ab}) = 0] - Pr[\mathcal{A}(g, h, g^a, g^b, R) = 0]| \geq \epsilon$ が成立するならば，多項式時間アルゴリズム $\mathcal{A}$ は XDH 問題を ϵ のアドバンテージで解くと定義する．もし XDH 問題を ϵ で解く多項式時間アルゴリズムが存在しなければ，XDH 仮定は成立するとする．

3. 鍵集約検索可能暗号 (KASE)

本節では，KASE [8] の性質について説明する．まず，本稿では単一のデータ保有者，複数人のデータ利用者，クラウドからなる 3 者モデルを想定する．このうち，データ保有者は単一の秘密鍵を所有する．データ保有者は暗号化データをクラウドに保存する際，そのデータを表すキーワードを暗号化して暗号化キーワードを生成してクラウドに保存する．次に，データ保有者は指定した範囲のデータ検索が可能な単一の集約鍵を生成してデータ利用者に送信する．そして，データ利用者はデータに対してキーワード検索を行うために単一のトラップドアを生成してクラウドに送信する．クラウド側では，受信したトラップドアを通して検索処理を行い，検索結果をデータ利用者に返す．

3.1 アルゴリズム

KASE のアルゴリズムを次のように定義する．

$params \leftarrow Setup(1^{\lambda}, n)$: このアルゴリズムはクラウドサービス提供者によって実行される．セキュリティパラメーター 1^{λ} とデータの最大数 n を入力として，公開パラメータ $params$ を出力する．

$sk \leftarrow KeyGen(params)$: このアルゴリズムはデータ保有者によって実行され，秘密鍵 sk を出力する．

$c_{i,l} \leftarrow Encrypt(params, sk, i, w_l)$: このアルゴリズムはデータ保有者によって実行され，i 番目のデータに紐づくキーワードの暗号化を行う．データ保有者の秘密鍵 sk，データ番号 i，キーワード $w_l \in \mathcal{KS}$ を入力として，暗号化キーワード $c_{i,l}$ を出力する．なお，$\mathcal{KS}$ はキーワード空間である．

$k_{agg} \leftarrow Extract(params, sk, S)$: このアルゴリズムはデータ保有者によって実行され，集約鍵を生成する．データ保有者の秘密鍵 sk とデータ番号集合 S を入力として，集約鍵 k_{agg} を出力する．

$Tr \leftarrow Trapdoor(params, k_{agg}, w_l)$: このアルゴリズムは，キーワード検索を行うデータ利用者によって実行される．集約鍵 k_{agg} とキーワード w_l を入力として，単一のトラップドア Tr を出力する．

$Tr_i \leftarrow Adjust(params, i, S, Tr, \{f_{1,j}\}_{j\in[1,m_1]})$: このアルゴリズムは，データ利用者からトラップドアを受信したクラウドサーバによって実行される．集合 S，検索対象のデータ番号 i，トラップドア Tr と補助関数 $\{f_j\}_{j\in[1,m_1]}(m_1 \in \mathbb{N})$ を入力として，S に含まれる i 番目のデータに対するトラップドア Tr_i を出力する．

$b \leftarrow Test(params, Tr_i, S, c_{i,l}, \{f_{2,j}\}_{j\in[1,m_2]})$: このアルゴリズムは，キーワード検索を行うクラウドサーバによって実行される．トラップドア Tr_i，データ番号 i と補助関数 $\{f_j\}_{j\in[1,m_2]}(m_2 \in \mathbb{N})$ を入力として，検索結果 b を出力する．もし i 番目のデータがキーワード w_l を含んでいれば $b = 1$，そうでなければ $b = 0$ となる．

なお，このアルゴリズムは複数のサーバ設定を含んだ定義となっている．具体的には，Adjust アルゴリズムと Test アルゴリズムの入力における補助関数 f_j は，複数のサーバ間での相互通信データを表す．この f_i を任意に設定することで，複数のサーバ設定を表現できる．

ここで，データ番号 $i \in S$ と紐づく任意のキーワード w_l に対して，次の記述が成立するならば KASE は正当性を満足すると定義する．任意の $1^{\lambda}, n \in \mathbb{N}, i \in [1, n], w_l \in \mathcal{KS}$ において，公開パラメータ $params \leftarrow Setup(1^{\lambda}, n)$，秘密鍵 $sk \leftarrow KeyGen(params)$ と暗号化キーワード $c_{i,l} \leftarrow Encrypt(params, sk, i, w_l)$ が利用される場合，もし $Tr \leftarrow Trapdoor(params, k_{agg}, w_l)$ かつ $Tr_i \leftarrow Adjust(params, i, S, Tr, \{f_{1,j}\}_{j\in[1,m_1]})$ ならば，$Test(params, Tr_i, S, c_{i,l}, \{f_{2,j}\}_{j\in[1,m_2]}) = true$.

それに加えて，KASE の性質を満たすために，KASE は compactness を満足する必要がある．

定義 4 (compactness) 任意のデータ数 n とデータ利用者数 m に対して，もし集約鍵とトラップドアのサイズが $\mathcal{O}(1)$，かつ暗号化キーワードのサイズが $\mathcal{O}(n)$ ならば，KASE は compactness を満たすと定義する．

3.2 安全性定義

本節では，KASE に関する 3 つの安全性要件を定義する．これらの安全性要件は，攻撃者 $\mathcal{A}$ と挑戦者 $\mathcal{C}$ の間における次のゲームによって定義される．それぞれのゲームにおいて，$\mathcal{C}$ と $\mathcal{A}$ は入力として $(1^{\lambda}, n)$ を与えられ，$\mathcal{A}$ は query フェーズにおいて集約鍵，暗号化キーワードとトラップドアをそれぞれ集約鍵オラクル $\mathcal{O}_{Extract}$，暗号化オラクル $\mathcal{O}_{Encrypt}$ とトラップドアオラクル $\mathcal{O}_{Trapdoor}$ に問い合わせることが可能である．

$\mathcal{O}_{Extract}$: $S \subseteq [1, n]$ を入力として，$k_{agg} \leftarrow Extract(params, sk, S)$. を返す．

$\mathcal{O}_{Encrypt}$: $i \in [1, n], w_l \in \mathcal{KS}$ を入力として，$c_{i,l} \leftarrow Encrypt(params, sk, i, w_l)$ を返す．

$\mathcal{O}_{Trapdoor}$: $S \subseteq [1,n], w_l \in \mathcal{KS}$ を入力として，$Tr \leftarrow Trapdoor(params, Extract(params, sk, S), w_l)$ を返す．

定義 5 ((ϵ, n)-キーワード秘匿性) このゲームでは，$\mathcal{A}$ はチャレンジ暗号化キーワードがチャレンジキーワードもしくはランダムキーワードのどちらかを判別する．

Init: $\mathcal{A}$ はチャレンジデータ番号 $i^* \in [1,n]$ を宣言して $\mathcal{C}$ に送る．

Setup: $\mathcal{C}$ は $params \leftarrow Setup(1^\lambda, n)$ と $sk \leftarrow KeyGen(params)$ を生成し，$params$ を $\mathcal{A}$ に送る．

Query: $\mathcal{A}$ は最大 $n-1$ 回 $\mathcal{O}_{Extract}$ に，任意回数 $\mathcal{O}_{Encrypt}$ に問い合わせを行うことができる．ただし，$\mathcal{O}_{Extract}$ においては，$S \subseteq [1,n] \setminus \{i^*\}$ という制約を課す．

Guess: $\mathcal{A}$ はチャレンジキーワード w_{l^*} を宣言して $\mathcal{C}$ に送る．$\mathcal{C}$ はランダムに $\theta \in \{0,1\}$ を選ぶ．もし $\theta = 0$ ならば，$w_\theta = w_{l^*}$ とする．そうでなければ，w_θ はランダムキーワードとする．ただし，$|w_0| = |w_1|$ とする．$\mathcal{C}$ は $c_{i^*,\theta} \leftarrow Encrypt(params, sk, i^*, w_\theta)$ を $\mathcal{A}$ に送る．$\mathcal{A}$ は $\theta' \in \{0,1\}$ を選ぶ．

確率的多項式時間アルゴリズムを持つ $\mathcal{A}$ と十分大きなサイズの 1^λ に対して，$|Pr[\theta = \theta'] - 1/2| < negl(1^\lambda)$ が成り立つならば KASE は (ϵ, n)-キーワード秘匿性を満足すると定義する．

定義 6 ((ϵ, n)-集約鍵偽造不可能性) このゲームでは，$\mathcal{A}$ はデータの検索可能な集約鍵の偽造を行う．

Setup: $\mathcal{C}$ はランダムに $i^* \in [1,n]$ を選ぶ．$\mathcal{C}$ は $params \leftarrow Setup(1^\lambda, n)$ と $sk \leftarrow KeyGen(params)$ を生成し，$params$ を $\mathcal{A}$ に送る．

Query: $\mathcal{A}$ は最大 $n-1$ 回 $\mathcal{O}_{Extract}$ に，任意回数 $\mathcal{O}_{Encrypt}$ に問い合わせを行うことができる．ただし，$\mathcal{O}_{Extract}$ においては，$i^* \notin S$ という制約を課す．

Forge: $\mathcal{A}$ は $S^* \subseteq [1,n], k^*_{agg}$ を出力する．ただし，$i^* \in S^*$ を満足するとする．

確率的多項式時間アルゴリズムを持つ $\mathcal{A}$，任意のキーワード w_l と十分大きなサイズの 1^λ に対して，次の関係が成り立つならば KASE は (ϵ, n)-集約鍵偽造不可能性を満足すると定義する．

$$Pr[Test(params, Adjust(params, i^*, S^*, Trapdoor(params, k^*_{agg}, w_l))) = Test(params, Adjust(params, i^*, S^*, Trapdoor(params, Extract(params, sk, S^*), w_l)))] < negl(1^\lambda)$$

定義 7 ((ϵ, n)-トラップドア秘匿性) このゲームでは，$\mathcal{A}$ はチャレンジトラップドアに埋め込まれているキーワードがチャレンジキーワードもしくはランダムキーワードのどちらかを判別する．

Init: $\mathcal{A}$ はチャレンジデータ番号集合 $S^* \subseteq [1,n]$ とチャレンジキーワード w_{l^*} を宣言して $\mathcal{C}$ に送る．

Setup: $\mathcal{C}$ は $params \leftarrow Setup(1^\lambda, n)$ と $sk \leftarrow KeyGen(params)$ を生成し，$params$ を $\mathcal{A}$ に送る．

Query: $\mathcal{A}$ は最大 $n - |S^*|$ 回 $\mathcal{O}_{Trapdoor}$ に，任意回数 $\mathcal{O}_{Encrypt}$ に問い合わせを行うことができる．ただし，$\mathcal{O}_{Encrypt}$ においては，$w_l \neq w_{l^*} \wedge i \notin S^*$ という制約を課す．

Guess: $\mathcal{C}$ はランダムに $\theta \in \{0,1\}$ を選ぶ．もし $\theta = 0$ ならば，$w_\theta = w_{l^*}$ とする．そうでなければ，w_θ はランダムキーワードとする．ただし，$|w_0| = |w_1|$ とする．$\mathcal{C}$ は $Tr^* \leftarrow Trapdoor(params, Extract(params, sk, S^*), w_\theta)$ を $\mathcal{A}$ に送る．$\mathcal{A}$ は $\theta' \in \{0,1\}$ を選ぶ．

確率的多項式時間アルゴリズムを持つ $\mathcal{A}$ と十分大きなサイズの 1^λ に対して，$|Pr[\theta = \theta'] - 1/2| < negl(1^\lambda)$ が成り立つならば KASE は (ϵ, n)-トラップドア秘匿性を満足すると定義する．

ここで，(ϵ, n)-トラップドア秘匿性はわずかな数の検索可能暗号 [1,7,11] でのみ議論されているオプションの安全性であるが，この (ϵ, n)-トラップドア秘匿性を満足することは可能な限り望ましい．

3.3 技術的困難性

KASE の構成の難しさは，安全性と機能面のトレードオフの関係に起因する．すなわち，compactness を満足することのみを考えると安全性を満足しない可能性がある．具体的には，文書と利用者の数に依存せず集約鍵とトラップドアのサイズが $\mathcal{O}(1)$ である必要があるので，そのための代数構造が限定される．従って，正当性を満足するための潜在的構成が限定されることになる．事実として，紙面上の都合省略するが，Cui らの方式 [8] はキーワード秘匿性を満たさない．さらに，集約鍵は具体的代数構造を持つ必要があるため，集約鍵を埋め込んだトラップドアは，$\mathcal{O}(1)$ となるようなより限定的な代数構造を持つ必要がある．このため，トラップドア秘匿性を達成可能な KASE を構成するのが一層難しくなっている．次の 4 節では，KASE の機能を持ちつつ安全性を高くするためにどのように KASE を構成するかということについて説明する．

4. 構成

本節では，3.2 節で述べた安全性要件の観点から，2 つの構成 (構成 1, 構成 2) を提案する．構成 1 はキーワード秘匿性と集約鍵偽造不可能性を満足する．構成 2 はキーワード秘匿性と集約鍵偽造不可能性に加えて，トラップドア秘匿性を満足する．

4.1 アイデア

KASE 構成の主なアイデアは，Boneh らの放送暗号 [6] と Boneh らの集約署名 [5] を組み合わせることにある．放送暗号は，多数の利用者が存在する中で，送信者が選択した利用者のみに対し，ブロードキャストチャネルを通して

安全かつ効率的にデータを送信する方式である．集約署名は，複数人の署名を集約して定数サイズの署名を生成する方式である．

Boneh らの放送暗号 [6] は，利用者の人数に依存せず暗号文が定数サイズという性質を持つことから，その暗号化アルゴリズムを踏襲して KASE の Encrypt アルゴリズムを構成できる．KASE の Extract アルゴリズムでは，集約署名の署名集約アルゴリズムを踏襲することで，定数サイズの集約鍵が生成可能である．従って，この放送暗号と集約署名を踏襲することで compactness が達成可能である．

KASE の安全性については，暗号文に関して Boneh らの放送暗号 [6] の安全性が証明されていることからキーワード秘匿性を達成する．集約鍵偽造不可能性についても，Boneh らの集約署名 [5] が署名の偽造不可能性を満足することから達成可能である．

構成 1 は，上述した既存暗号方式を単純に合成した方式に近い．構成 1 におけるトラップドアは既存暗号方式を取り入れた構成となっていないため，トラップドア秘匿性を満足しない．構成 2 では，トラップドア秘匿性を満足した構成を提案する．構成 1 では，トラップドアは確定的な構成となっている．そこで，トラップドア秘匿性を達成するために，本稿ではトラップドアに乱数を埋め込み確率的な構成とすることを考えた．そのためには，同じ乱数の値をクラウドに知られないようにしつつ暗号化キーワードにもその乱数を埋め込み，正当性を満足する必要がある．そこで，構成 2 では秘密分散 [18] と n-out-of-n しきい値復号 [9] のアイデアを取り入れ，互いに結託しない 2 台のサーバ環境を用意することを考えた．

構成 2 の構成方法は次の通りである．KASE の Trapdoor アルゴリズムにおいてトラップドアに埋め込んだ乱数において，秘密分散の要領でその乱数を分割して 2 つのシェアとし，それぞれのサーバに送る．これらのシェアは，しきい値復号における秘密鍵とみなす．次に，KASE の Adjust アルゴリズムと Test アルゴリズムにおいて，n-out-of-n しきい値復号における結合アルゴリズムの要領で暗号化キーワードへの乱数埋め込みを行う．この一連の処理を行うことで，クラウド側で乱数自体の値を知ることなく暗号化キーワードに対して乱数を埋め込むことができる．したがって，トラップドア秘匿性を満足しつつ正当性を満足することが可能となる．

4.2 構成 1

構成 1 は次の通りである．

$params \leftarrow Setup(1^\lambda, n)$: 双線型写像で利用するパラメータ $\mathbb{B} = (p, \mathbb{G}, \mathbb{G}_T, e(\cdot,\cdot))$ を生成する．ここで，p は $2^\lambda < p < 2^{\lambda+1}$ を満足する $\mathbb{G}$ の素数位数である．n をデータの最大数とする．ランダムな生成元 $g \in \mathbb{G}$ とランダムな $\alpha \in \mathbb{Z}_p$ を用いて $g_i = g^{(\alpha^i)} \in \mathbb{G}(i \in \{1,2,...,n,n+2,...,2n\})$ を計算する．一方向性ハッシュ関数 $H : \{0,1\}^* \rightarrow \mathbb{G}$ を選ぶ．そして，公開パラメータ $params = (\mathbb{B}, PubK, H)$ を出力する．ここで，$PubK = (g, g_1, ..., g_n, g_{n+2}, ..., g_{2n}) \in \mathbb{G}^{2n}$ とする．

$sk \leftarrow KeyGen(params)$: ランダムな $\beta \in \mathbb{Z}_p$ を選び，秘密鍵 $sk = \beta$ を出力する．

$c_{i,l} \leftarrow Encrypt(params, sk, i, w_l)$: ランダムな $t_{i,l} \in \mathbb{Z}_p$ を選び，次の暗号化キーワード $c_{i,l} = (c_{1,i,l}, c_{2,i,l}, c_{3,i,l})$ を出力する．ただし，$c_{1,i,l} = g^{t_{i,l}}, c_{2,i,l} = (g^\beta \cdot g_i)^{t_{i,l}}, c_{3,i,l} = e(H(w_l), g)^{t_{i,l}} / e(g_1, g_n)^{t_{i,l}}$ である．

$k_{agg} \leftarrow Extract(params, sk, S)$: データ番号集合 $S \subseteq [1,n]$ に対して，集約鍵 $k_{agg} = \Pi_{j \in S} g_{n+1-j}^\beta$ を出力する．

$Tr \leftarrow Trapdoor(params, k_{agg}, w_l)$: 集約鍵 k_{agg} に対して，キーワード w_l に対するトラップドア $Tr = k_{agg} \cdot H(w_l)$ を出力する．

$Tr_i \leftarrow Adjust(params, i, S, Tr)$: 集合 S に対して，トラップドア $Tr_i = Tr \cdot \Pi_{j \in S, j \neq i} g_{n+1-j+i}$ を出力する．

$b \leftarrow Test(params, Tr_i, S, c_{i,l})$: トラップドア Tr_i に対して，$e(Tr_i, c_{1,i,l}) / e(c_{2,i,l}, pub) =^? c_{3,i,l}$ が成り立つかどうかを判定して $true$ もしくは $false$ を出力する．ここで，$pub = \Pi_{j \in S} g_{n+1-j}$ とする．

また，$k_{agg} = \Pi_{j \in S} g_{n+1-j}^\beta \in \mathbb{G}, Tr = k_{agg} \cdot H(w_l) \in \mathbb{G}$ より，S の番号の個数に関係なく集約鍵とトラップドアのサイズは $|\mathbb{G}|$ であることが分かる．つまり，構成 1 は compactness を満足する．

4.3 構成 2

構成 2 では，互いに結託しない 2 台のサーバ $\mathcal{C}_{main}$ と $\mathcal{C}_{aid}$ を利用する．それぞれのサーバには同じ暗号化キーワードを保存する．

構成 2 は次の通りである．なお，構成 2 では，トラップドア秘匿性を満足するために非対称双線型写像を利用する．さらに，Adjust アルゴリズムと Test アルゴリズムの入力では，関数 $f : \mathbb{Z}_p \times \mathbb{G} \rightarrow \mathbb{G}$ と $f_T : \mathbb{Z}_p \times \mathbb{G}_T \rightarrow \mathbb{G}_T$ を用いる．f は任意の 2 つの入力 $x \in \mathbb{Z}_p, g \in \mathbb{G}$ をとり，$g^x \in \mathbb{G}$ を出力する関数である．f_T は任意の 2 つの入力 $x \in \mathbb{Z}_p, g_T \in \mathbb{G}_T$ をとり，$g_T^x \in \mathbb{G}_T$ を出力する関数である．

$params \leftarrow Setup(1^\lambda, n)$: 双線型写像で利用するパラメータ $\mathbb{B} = (p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e(\cdot,\cdot))$ を生成する．n をデータの最大数とする．ランダムな生成元 $g \in \mathbb{G}, h \in \mathbb{H}$ とランダムな $\alpha \in \mathbb{Z}_p$ を用いて $g_i = g^{(\alpha^i)} \in \mathbb{G}(i \in \{1,2,...,n,n+2,...,2n\}), h_i = h^{(\alpha^i)} \in \mathbb{H}(i \in [1,n])$ を計算する．一方向性ハッシュ関数 $H : \{0,1\}^* \rightarrow \mathbb{G}$ を選ぶ．そして，公開パラメータ $params = (\mathbb{B}, PubK, H)$ を出力する．ここで，$PubK = (g, g_1, ..., g_n, g_{n+2}, ..., g_{2n}, h, h_1, ..., h_n) \in (\mathbb{G}^{2n} \times \mathbb{H}^{n+1})$ とする．

$sk \leftarrow KeyGen(params)$: ランダムな $\beta \in \mathbb{Z}_p$ を選び，

秘密鍵 $sk = \beta$ を出力する．

$c_{i,l} \leftarrow Encrypt(params, sk, i, w_l)$: ランダムな $t_{i,l} \in \mathbb{Z}_p$ を選び，次の暗号化キーワード $c_{i,l} = (c_{1,i,l}, c_{2,i,l}, c_{3,i,l})$ を出力する．ただし，$c_{1,i,l} = h^{t_{i,l}} \in \mathbb{H}, c_{2,i,l} = (g^{\beta} \cdot g_i)^{t_{i,l}} \in \mathbb{G}, c_{3,i,l} = e(H(w_l), h)^{t_{i,l}} / e(g_1, h_n)^{t_{i,l}} \in \mathbb{G}_T$ である．

$k_{agg} \leftarrow Extract(params, sk, S)$: データ番号集合 $S \subseteq [1, n]$ に対して，集約鍵 $k_{agg} = \Pi_{j \in S} g_{n+1-j}^{\beta} \in \mathbb{G}$ を出力する．

$Tr \leftarrow Trapdoor(params, k_{agg}, w_l)$: ランダムに $r \in \mathbb{Z}_p$ を選び，$Tr = (k_{agg} \cdot H(w_l))^r \in \mathbb{G}$ を計算する．そして，r を $r = r_{main} + r_{aid}$ と分割する．(Tr, r_{main}) を $\mathcal{C}_{main}$ に，$Tr_{aid} = r_{aid}$ を $\mathcal{C}_{aid}$ に送る．

$Tr_i \leftarrow Adjust(params, i, S, Tr)$: 2 台のサーバで $pub_i = \Pi_{j \in S, j \neq i} g_{n+1-j+i} \in \mathbb{G}$ を計算する．$\mathcal{C}_{aid}$ は $f(r_{aid}, pub_i) = pub_i^{r_{aid}}$ を $\mathcal{C}_{main}$ に送る．次に，$\mathcal{C}_{main}$ は $(f(r_{main}, pub_i)) \cdot (f(r_{aid}, pub_i)) = pub_i^{r_{main}} \cdot pub_i^{r_{aid}} = pub_i^{r}$ と $Tr_i = Tr \cdot pub_i^{r} \in \mathbb{G}$ を計算する．

$b \leftarrow Test(params, Tr_i, S, c_{i,l}, f_T(r_{main}, c_{2,i,l}^{\#}), f_T(r_{aid}, c_{2,i,l}^{\#}), f_T(r_{main}, c_{3,i,l}), f_T(r_{aid}, c_{3,i,l}))$: 2 台のサーバで $pub = \Pi_{j \in S} h_{n+1-j} \in \mathbb{H}$ and $c_{2,i,l}^{\#} = e(c_{2,i,l}, pub)$ を計算する．$\mathcal{C}_{aid}$ は $f_T(r_{aid}, c_{2,i,l}^{\#}) = e(c_{2,i,l}, pub)^{r_{aid}}$ を $\mathcal{C}_{main}$ に送る．次に，$\mathcal{C}_{main}$ は $(f_T(r_{main}, c_{2,i,l}^{\#})) \cdot (f_T(r_{aid}, c_{2,i,l}^{\#})) = e(c_{2,i,l}, pub)^{r_{main}} \cdot e(c_{2,i,l}, pub)^{r_{aid}} = e(c_{2,i,l}, pub)^{r}$ を計算する．それに加えて，$\mathcal{C}_{aid}$ は $f_T(r_{aid}, c_{3,i,l}) = c_{3,i,l}^{r_{aid}}$ を $\mathcal{C}_{main}$ に送る．そして，$\mathcal{C}_{aid}$ は $(f_T(r_{main}, c_{3,i,l})) \cdot (f_T(r_{aid}, c_{3,i,l})) = c_{3,i,l}^{r_{main}} \cdot c_{3,i,l}^{r_{aid}} = c_{3,i,l}^{r}$ を計算して，次の等式：$\frac{e(Tr_i, c_{1,i,l})}{e(c_{2,i,l}, pub)^r} \stackrel{?}{=} c_{3,i,l}^{r}$. が成り立つかどうかを判定して $true$ もしくは $false$ を出力する．

ここで，構成 1 と同様に，構成 2 は正当性と compactness を満足する．

5. 満足する安全性

本節では，3.2 節で定式化した KASE の安全性を満足することを帰着アルゴリズムを構成することで証明できることを説明する．本稿では，紙面の都合上，それぞれの安全性について証明のアイデアについてのみ説明する．

5.1 構成 1

定理 1 ((ϵ', n)-キーワード秘匿性) 構成 1 は，2 節の (ϵ, n)-BDHE 仮定が成り立てば (ϵ', n)-キーワード秘匿性を満足する．ただし，$\epsilon \geq \epsilon'$ とする．

本稿では，(ϵ', n)-キーワード秘匿性を多項式時間で解くことができるアドバンテージ ϵ' のアルゴリズム $\mathcal{A}$ が存在すれば，(ϵ, n)-BDHE 問題を多項式時間で解くことができるアドバンテージ ϵ のアルゴリズム $\mathcal{B}$ を構成した．$\mathcal{B}$ が (ϵ, n)-BDHE 問題で受け取る入力のうち，$(g, g^{\alpha}, ..., g^{\alpha^n}, g^{\alpha^{n+2}}, ..., g^{\alpha^{2n}})$ を KASE の公開パラメータに，h, Z をチャレンジ暗号化キーワードに割り当てる．もし $Z = e(g^{\alpha^{n+1}}, h)$ ならば，チャレンジ暗号化キーワードは KASE の正当性を満足する構成となり，$\mathcal{A}$ は (ϵ', n)-キーワード秘匿性を多項式時間で解くことができる．つまり，キーワードの情報が $\mathcal{A}$ に判明するため，$Z = e(g^{\alpha^{n+1}}, h)$ であることが $\mathcal{B}$ に判明し，$\mathcal{B}$ は (ϵ, n)-BDHE 問題を解くことができる．

定理 2 ((ϵ', n)-集約鍵偽造不可能性) 構成 1 は，2 節の (ϵ, n)-DHE 仮定が成り立てば (ϵ', n)-集約鍵偽造不可能性を満足する．ただし，$\epsilon = \epsilon'$ とする．

本稿では，(ϵ', n)-集約鍵偽造不可能性を多項式時間で解くことができるアドバンテージ ϵ' のアルゴリズム $\mathcal{A}$ が存在すれば，(ϵ, n)-DHE 問題を多項式時間で解くことができるアドバンテージ ϵ のアルゴリズム $\mathcal{B}$ を構成した．$\mathcal{B}$ が (ϵ, n)-DHE 問題で受け取る入力のうち，$(g, g^{\alpha}, ..., g^{\alpha^n}, g^{\alpha^{n+2}}, ..., g^{\alpha^{2n}})$ は KASE の公開パラメータに割り当てる．そして，$\mathcal{A}$ は i^* 番目を含むデータの検索が可能な集約鍵を出力する．もし偽造した集約鍵が正当性を満足すれば，$\mathcal{B}$ はその集約鍵を用いて $g^{\alpha^{n+1}}$ を出力することができる．従って，$\mathcal{B}$ は (ϵ, n)-DHE 問題を解くことができる．

5.2 構成 2

本節では，構成 2 が (ϵ', n)-トラップドア秘匿性を満足することを説明する．なお，構成 2 は (ϵ', n)-キーワード秘匿性と (ϵ', n)-集約鍵偽造不可能性も満足する．これは，(ϵ, n)-BDHE 仮定と (ϵ, n)-DHE 仮定を非対称群形式へと拡張した仮定を用いることで，構成 1 の場合と同様に証明が可能である．

定理 3 ((ϵ', n)-トラップドア秘匿性) 構成 2 は，ランダムオラクルモデルの存在下で 2 節の XDH 仮定が成り立てば (ϵ', n)-トラップドア秘匿性を満足する．ただし，$\epsilon \geq \epsilon'$ とする．

本稿では，(ϵ', n)-トラップドア秘匿性を多項式時間で解くことができるアドバンテージ ϵ' のアルゴリズム $\mathcal{A}$ が存在すれば，XDH 問題を多項式時間で解くことができるアドバンテージ ϵ のアルゴリズム $\mathcal{B}$ を構成した．なお，構成 1 と異なり，構成 2 は互いに結託しない 2 台のサーバを利用する．そこで，$\mathcal{B}$ は片方のサーバ $\mathcal{C}_{aid}$ の動作も含めた上で証明を行う．$\mathcal{B}$ が XDH 問題で受け取る入力のうち，g^a, Z はチャレンジトラップドアに，g^b は KASE の秘密鍵に割り当てる．もし $Z = g^{ab}$ ならば，チャレンジトラップドアは KASE の正当性を満足する構成となり，$\mathcal{A}$ は (ϵ', n)-トラップドア秘匿性を多項式時間で解くことができる．つまり，キーワードの情報が $\mathcal{A}$ に判明するため，$Z = g^{ab}$ であることが $\mathcal{B}$ に判明し，$\mathcal{B}$ は XDH 問題を解くことができる．

6. 考察

本節では，提案方式の性能について議論する．まず，本稿では提案方式の実装評価を行った．次に，提案方式の計算コストとストレージコスト，および安全性について関連研究と理論的に比較した．

実装評価 本稿では，双線型写像を計算するためにバージョン 0.94[*1]の mcl ライブラリを用いた．このライブラリでは，C++を用いる．また，本稿では BLS12-381 曲線を利用する．暗号処理は Mac OS Mojave 上で行った．CPU は 1.4 GHz Intel Core i5-4260U で，メモリは 4 GB 1600 MHz DDR3 である．なお，mcl ライブラリでは通信のための関数が実装されていないため，2 台のサーバ間の通信遅延は評価していない．

提案方式の実装評価結果を図 1-6 に載せる．なお，2 つの提案方式間において，Setup アルゴリズム，Encrypt アルゴリズム，Extract アルゴリズムは共通しているため，図 1-3 には構成 1 のみの結果を載せている．

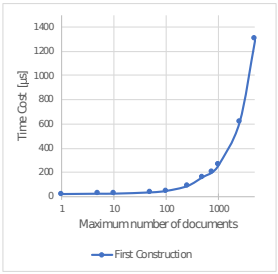

図 1: Setup の実行時間

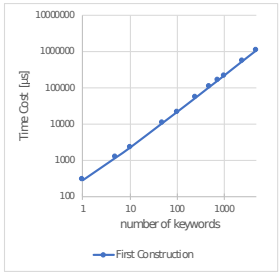

図 2: Encrypt の実行時間

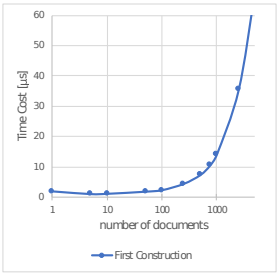

図 3: Extract の実行時間

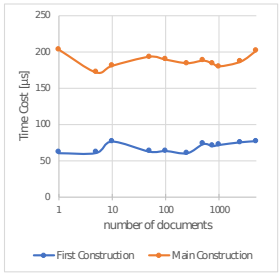

図 4: Trapdoor の実行時間

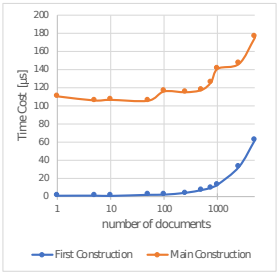

図 5: Adjust の実行時間

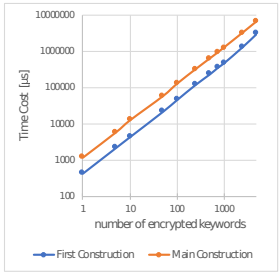

図 6: Test の実行時間

[*1] mcl library: `https://github.com/herumi/mcl`

Trapdoor アルゴリズムを除き，いずれのアルゴリズムも線形増加している．Trapdoor アルゴリズムに関しては，n に依存しない実行時間となっている．

また，全体のアルゴリズムの中で Encrypt アルゴリズムと Test アルゴリズムが他のアルゴリズムの 100 倍以上の実行時間となっているが，これは双線型写像の計算に大きな時間が必要だからである．しかし，$n = 5000$ に対して，Adjust アルゴリズムを含めた Test アルゴリズムの計算時間は構成 1 において 3 秒程度，構成 2 においては 6 秒となっている．従って，検索時間もそれと同じ程度だと言える．Encrypt アルゴリズムは 1 秒程度となっていて，実用可能な実行時間で提案方式の運用が可能だと考えられる．

計算コストとストレージコスト 提案方式と関連研究の計算コストとストレージコストの分析結果を表 1,2 に載せる．表 1,2 には，単数の保有者設定の方式を示している．

構成 1 における計算コストとストレージコストは，Cui ら [8] の方式と Li ら [13] の方式と等価もしくは少なくなっている．構成 2 においても，Encrypt アルゴリズムは既存方式と等価である．Adjust+Test アルゴリズムは既存方式と比べてコストが増加しているが，負荷が特に大きい $\mathbb{G}$ 上でのスカラー倍算と G_T 上でのべき乗算がデータ数に対して定数時間であるため，十分に実用的な計算量と考えられる．ストレージコストにおいても，Cui らの方式などと比べて $2|\mathbb{Z}_p|$ 分だけ増えているが，これは $\mathbb{G}$ とサイズが変わらない．従って，Zhou らの方式と比べて提案方式の計算コストとストレージコストは等価な状態を達成している．

安全性 提案方式や既存方式が達成している安全性を表 3 に載せる．Cui ら [8] と Li ら [13] はキーワード秘匿性について帰着アルゴリズムを用いた安全性証明は議論していない．一方，Zhou ら [22] の方式は帰着アルゴリズムを用いた安全性証明を議論しているが，この方式は compactness を満足しない．これは，遠隔にあるセンサデバイスが暗号化するという特殊な状況を想定しているためである．これは各センサデバイスが暗号化用の鍵をさらに持つことになり，システム全体で見たときに鍵の本数がセンサ数に対して線形に増加する．このため，compactness を満たさない．また，集約鍵偽造不可能性は他の方式では議論されておらず，安全性が証明されていない．

謝辞 本研究の一部は JSPS 科研費 18K18049 およびセコム財団挑戦的研究助成の助成を受けたものです．

参考文献

[1] A. Arriaga, Q. Tang, and P. Ryan. Trapdoor privacy in asymmetric searchable encryption schemes. In *Proc. of AFRICACRYPT 2014*, pages 31–50. Springer, 2014.

[2] L. Ballard, M. Green, B. De Medeiros, and F. Monrose. Correlation-resistant storage via keyword-searchable encryption. *IACR Cryptology ePrint*

	Cui et al. [8]	Li et al. [13]	Zhou et al. [22]	構成 1	構成 2
Encrypt	$T_h + 2T_{sm} + T_a +$ $2T_p + T_{mul} + T_{exp}$	$T_h + 2T_{sm} + 2T_a + 2T_p +$ $T_{mul} + 2T_{exp} + T_x + T_{bf}$	$T_h + 4T_{sm} + T_a$	$T_h + 2T_{sm} + T_a +$ $2T_p + T_{mul} + T_{exp}$	$T_h + 2T_{sm} + T_a +$ $2T_p + T_{mul} + T_{exp}$
Trapdoor	$T_h + T_a$	$T_h + T_a$	$T_h + 3T_{sm} + T_a$	$T_h + T_a$	$T_h + T_a + T_{sm}$
Adjust+Test	$2\|S\| \cdot T_a +$ $T_{mul} + 2T_p$	$2\|S\| \cdot T_a +$ $T_{mul} + 2T_p$	$2\|S\| \cdot T_a +$ $2T_{mul} + 4T_p$	$2\|S\| \cdot T_a +$ $T_{mul} + 2T_p$	$6\|S\| \cdot T_a + 2\|S\| \cdot T_{mul} +$ $4T_{sm} + 2T_{exp} + 2T_p$

表 1: KASE の計算コスト：ハッシュ関数の計算，スカラー倍算，点加算， $\mathbb{G}_T$ 上での排他的論理和，$\mathbb{G}_T$ 上でのべき乗計算，$\mathbb{G}_T$ 上での乗算，双線型写像の計算にかかる時間をそれぞれ $T_h, T_{sm}, T_a, T_x, T_{exp}, T_{mul}, T_p$ と定義する．それに加えて，Li ら [13] の方式ではブルームフィルターを用いる．その計算コストを T_{bf} と定義する．乱数生成計算と整数加算にかかる時間は無視する．なお，表で示している Adjust+Test アルゴリズムの計算コストは 1 つのデータに対して表している．

	Cui et al. [8]	Li et al. [13]	Zhou et al. [22]	構成 1	構成 2
暗号化 キーワード	$2\|\mathbb{G}\|+$ $\|\mathbb{G}_T\|$	$2\|\mathbb{G}\|+$ $3\|\mathbb{G}_T\|$	$3\|\mathbb{G}\|$	$2\|\mathbb{G}\|+$ $\|\mathbb{G}_T\|$	$2\|\mathbb{G}\|+$ $\|\mathbb{G}_T\|$
トラップドア	$\|\mathbb{G}\|$	$\|\mathbb{G}\|$	$2\|\mathbb{G}\|$	$\|\mathbb{G}\|$	$\|\mathbb{G}\|+$ $2\|\mathbb{Z}_p^*\|$
集約鍵	$\|\mathbb{G}\|$	$\|\mathbb{G}\|$	$\|\mathbb{G}\|$	$\|\mathbb{G}\|$	$\|\mathbb{G}\|$

表 2: KASE のストレージコスト：$|\mathbb{G}|, |\mathbb{G}_T|$ は $\mathbb{G}, \mathbb{G}_T$ のサイズである．

	Cui et al. [8]	Li et al. [13]	Zhou et al. [22]	構成 1	構成 2
compactness	✓	✓		✓	✓
キーワード 秘匿性			✓	✓	✓
集約鍵 偽造不可能性				✓	✓
トラップドア 秘匿性			✓		✓

表 3: 満足する KASE の安全性：チェックマークは，証明可能安全性を達成していることを意味する．

Archive, 2005:417, 2005.

[3] F. Bao, R. H. Deng, X. Ding, and Y. Yang. Private query on encrypted data in multi-user settings. In *Proc. of ISPEC 2008*, pages 71–85. Springer, 2008.

[4] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *Proc. of ASIACRYPT 2004*, pages 506–522. Springer, 2004.

[5] D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *Proc. of EUROCRYPT 2003*, pages 416–432. Springer, 2003.

[6] D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Proc. of CRYPTO 2005*, pages 258–275. Springer, 2005.

[7] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou. Privacy-preserving multi-keyword ranked search over encrypted cloud data. *IEEE Transactions on parallel and distributed systems*, 25(1):222–233, 2014.

[8] B. Cui, Z. Liu, and L. Wang. Key-aggregate searchable encryption (kase) for group data sharing via cloud storage. *IEEE Trans. Computers*, 65(8):2374–2385, 2016.

[9] Y. G. Desmedt. Threshold cryptography. *European Transactions on Telecommunications*, 5(4):449–458, 1994.

[10] J. Herranz, F. Laguillaumie, B. Libert, and C. Ràfols. Short attribute-based signatures for threshold predicates. In *Proc. of CT-RSA 2012*, pages 51–67. Springer, 2012.

[11] A. Kiayias, O. Oksuz, A. Russell, Q. Tang, and B. Wang. Efficient encrypted keyword search for multi-user data sharing. In *Proc. of ESORICS2016*, pages 173–195. Springer, 2016.

[12] T. Li, Z. Liu, C. Jia, Z. Fu, and J. Li. Key-aggregate searchable encryption under multi-owner setting for group data sharing in the cloud. *International Journal of Web and Grid Services*, 14(1):21–43, 2018.

[13] T. Li, Z. Liu, P. Li, C. Jia, Z. L. Jiang, and J. Li. Verifiable searchable encryption with aggregate keys for data sharing in outsourcing storage. In *Proc. of ACISP 2016*, pages 153–169. Springer, 2016.

[14] K. Liang and W. Susilo. Searchable attribute-based mechanism with efficient data sharing for secure cloud storage. *IEEE Transactions on Information Forensics and Security*, 10(9):1981–1992, 2015.

[15] P. Mukti and C. J. Devesh. Mulkase-a novel approach for key aggregate searchable encryption for multi-owner data. *Frontiers of Information Technology & Electronic Engineering*, 2018.

[16] S. Patranabis, Y. Shrivastava, and D. Mukhopadhyay. Provably secure key-aggregate cryptosystems with broadcast aggregate keys for online data sharing on the cloud. *IEEE Transactions on Computers*, 66(5):891–904, 2017.

[17] R. A. Popa and N. Zeldovich. Multi-key searchable encryption. *IACR Cryptology ePrint Archive*, 2013:508, 2013.

[18] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[19] D. X. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. In *IEEE S&P 2000*, pages 44–55. IEEE, 2000.

[20] C. Van Rompay, R. Molva, and M. Önen. Multi-user searchable encryption in the cloud. In *Proc. of ISC 2015*, pages 299–316. Springer, 2015.

[21] Q. Zheng, S. Xu, and G. Ateniese. Vabks: verifiable attribute-based keyword search over outsourced encrypted data. In *Proc. of INFOCOM 2014*, pages 522–530. IEEE, 2014.

[22] R. Zhou, X. Zhang, X. Du, X. Wang, G. Yang, and M. Guizani. File-centric multi-key aggregate keyword searchable encryption for industrial internet of things. *IEEE Transactions on Industrial Informatics*, 2018.