クラウドコンピューティングに適した計算量的安全性を持つ秘密分散法

高橋 慧† 岩村 惠市†

† 東京理科大学工学部第一部電気工学科
102-0073 東京都千代田区九段北 1-14-6
takahashi@sec.ee.kagu.tus.ac.jp

あらまし 秘密分散法の活用法として，多くのユーザが参加し，それぞれのユーザの持つデータを分散させてクラウドコンピューティングに適用がさせることが考えられる．しかし Shamir の秘密分散法をクラウドシステムへそのまま適用すると各サーバが持つ分散情報のデータサイズを秘密情報より小さくすることができないため，分散する秘密情報の数が多い場合には適していない．そこで，本論文では具体的なクラウドシステムを想定し，分散情報の記憶容量の削減と計算量的な安全性を実現する新たな秘密分散法の提案を行う．また，本提案方式と従来方式を利用した場合におけるシステム構成やサーバの持つ記憶容量，ユーザの行う処理量に関する比較を行う．

Secret Sharing Scheme Suitable for Cloud Computing Having Computational Safety

Satoshi Takahashi† Keichi Iwamura†

†Tokyo University of Science
1-14-6 Kudankita, Ciyoda-ward, Tokyo 102-0073, JAPAN
takahashi@sec.ee.kagu.tus.ac.jp

Abstract Secret sharing scheme can be used as cloud computing which many users are participated and each user has distribute multiple data. However if apply shamir's secret sharing to this cloud system, it can't reduce share size less than former secret, so it doesn't suit the case when there are many secrets to distribute. Therefore, in this paper we assume a concrete cloud system and propose secret sharing scheme, it can achieve computational safety while reducing share size. Also, we compare with conventional method, about system configuration, the share size and amount of processing users.

1 はじめに

近年，新たなデータストレージ法としてクラウドコンピューティングが注目されている．クラウドコンピューティングとはユーザの持つデータをクラウドと呼ばれるネットワーク上にある複数のサーバで構成される仮想の大容量ストレージに分散・保管する技術である．また，秘匿計算を用いることで単にデータをストレージするだけでなく，クラウド上に分散・保管されたデータを用いて各サーバにユーザのデータを知られることなく任意の計算を行うことができる．

この様な秘匿計算を実現するために秘密分散の利用が注目されている．秘密分散を用いたクラウドシステムには以下のような要件が求められている．

(1) 1 つのデータを複数のサーバに分散し，こ

れらのデータのうちいくつかが破損しても元のデータを復元することができる．
(2) 秘密分散によって増加する記憶容量を少なくとも (k,n) 閾値秘密分散法を用いた時よりも抑えることができる．
(3) 分散情報が閾値以下集まった場合にも情報の部分的漏えいが生じず，少なくとも計算量的安全性を持つ．

まず (1) の要件について考える．クラウドコンピューティングは複数のサーバにより構成されており，これらのサーバのうちのいくつかが災害による停電やヒューマンエラー等によりダウンすることは十分に考えられるためこの要件は必要である．また、(2) の要件については，これからよりクラウドシステムが浸透していく上で問題になるのはユーザの増加に起因するデータ量の増加である．しかし秘密分散法の従来方式である (k,n) 閾値秘密分散法をそのまま適用した場合，分散情報のデータサイズを秘密情報よりも小さくすることができないため，容量効率が非常に悪い．このような問題に対応するために (2) の要件は必要である．また，記憶容量の削減を実現したとしても安全性が保たれていなければ意味がない．そこで (3) の要件が必要となってくる．

ここで，これまでに提案された代表的な秘密分散法をクラウドに適用することを考える．まず，(k,n) 閾値秘密分散法 [1] をクラウドに用いると，n 個のサーバにデータを分散しユーザはそのうち k 個を集めることで元の秘密情報を復元することができるため (1) の要件を満たすことができる．また，(3) についてもこの手法は情報量的安全性を持つことが知られているためこれも満たすことができる．しかしこの手法は前述の通り，各サーバが持つ分散情報のデータサイズを秘密情報よりも小さくすることができないため，(2) の要件は満たしていない．そこで，この欠点を補う手法として，ランプ型秘密分散法 [2] を用いる場合，ユーザの記憶容量を (k,n) 閾値秘密分散法に比べ，1/L 倍にすることができるが，サーバに分散する分散情報のサイズを削減するほど，閾値である k 個未満の分散情報から秘密情報の段階的な漏えいが生じるため，安全性に問題が生じてしまう．そのため，この手法は (1) 及び (2) の要件は満たしているが，(3) の要件を満たしていない．

また，(1)〜(3) の条件全てを満たした方式として SCIS2010 に植松・岩村等による方式 (以降，植松方式)[3] が提案されている．この方式は，秘密情報が m 個ある場合，まず分散情報を削減するサーバ l 台 ($l = 2 \leq l \leq k-1$)(以下鍵サーバ) を決定し，秘密情報 s_1 に関する分散情報 W_{1j} を (k,n) 閾値秘密分散法と同様の手順で生成し，この値を初期値として，l 台の鍵サーバは $m-1$ 個の疑似乱数 $q_{2j} \cdots q_{mj} (j = 2 \cdots l)$ を生成する．その後，生成された疑似乱数から係数 $a_{i1} \cdots a_{ik-1} (i = 2 \cdots m)$ を決定し，残りのサーバ (以下データサーバ) の持つ分散情報 $W_{2j} \cdots W_{mj} (j = l \cdots n)$ を計算する．この方式を用いることで最大 $k-1$ 台のサーバが持つ情報を疑似乱数生成のための初期値 W_{1j} 及び鍵 key_j のみとすることができる．またこの方式の安全性は利用する疑似乱数生成器に依存するため，利用する疑似乱数生成器が計算量的安全性を持つ場合には，植松方式も計算量的安全性を持つ．しかしこの方式はユーザ毎に疑似乱数生成を行うため鍵サーバは各ユーザごとに初期値 W_{1j} 及び鍵 key_j を持つ必要があり，クラウドコンピューティングのような参加ユーザの人数が多い場合には多くのデータを持つ必要がある．また，ユーザがある秘密情報 s_i のみを復元したい場合にも鍵サーバは初期値 W_{1j} と鍵 key_j を用いて s_i に関する分散情報となる疑似乱数 q_{ij} が出力されるまで疑似乱数生成を繰り返す必要があるため，冗長な計算を行う必要がある．

そこで，本論文では植松方式を改良し，上記 (1)〜(3) の要求を満たしながら多くのユーザが参加する具体的なシステムに適した方式を提案する．この手法は，利用する暗号方式が計算量的安全性を持つと仮定した場合，それぞれの秘密情報に関する独立性及び，秘密情報に関する計算量的安全性を保ったまま分散情報の削減を行うことができるため，上記クラウドシステムに適した秘密分散法であるといえる．

以下，本論文の構成を示す．第 2 章において前述の従来手法である植松方式の構成やこの方

式をクラウドシステムに適用した際の諸問題について説明を行う．第 3 章では植松方式を改良し，クラウドシステムへの適用へより適した秘密分散法である提案方式の構成を示し，4 章ではその評価を行う．

2　従来方式

本章では，複数の秘密情報の分散に適した秘密分散法の従来方式である植松方式についての説明を行う．

なお，本論文全体を通じて秘密分散に関する計算は秘密情報を s，ユーザが秘密情報を分散するサーバの台数を n としたとき，$s < p$ かつ $n < p$ である素数 p による (mod p) 上で行うものとする．また，分散する秘密情報の個数を m とし，簡単のため全秘密情報と分散情報のサイズを同じとし，$|s|$ と表す.

2.1　植松方式

植松方式は，最大 $k-1$ 台の鍵サーバについて秘密情報 s_1 に関する分散情報 $W_{1j}(j = 1 \cdots k-1)$ を初期値とし，鍵 key_j を利用して生成した疑似乱数 $q_{1j} \cdots q_{mj}$ が他の秘密情報 $s_2 \cdots s_m$ の分散情報となるように係数 $a_{i1} \cdots a_{ik-1}(i=2 \cdots m)$ を決定することでこれらの鍵サーバの分散情報を初期値 W_{1j} と鍵 key_j のみに削減する方式である．具体的にはこの方式は以下のような手順で構成される．

[分散]

1. ディーラは $a_{11}, \cdots, a_{1k-1}$ を任意に定め，$W_{11}, \cdots, W_{1n}$ を計算する
2. サーバ $j=j_1, j_2, \cdots, j_n$ から任意の $L-1$ 台 ($1 \leq L \leq$ k) を選び，$j_l(1 \leq l \leq L-1)$ とする (分散情報を 1 つと疑似乱数生成のための鍵だけ持てば良い鍵サーバとなる).
3. ディーラは鍵サーバに key_{j_l} を鍵として定める.
4. $i=2$ として，鍵サーバ x_{j_l}($1 \leq l \leq L-1$) は W_{1j_l} を初期値，key_{j_l} を鍵として疑似乱数生成器を用いて疑似乱数 q_{ij_l} を生成する.
5. ディーラはまず，$a_{iL}, \cdots, a_{ik-1}$ を任意に定め，$W_{ij_1} = q_{ij_1}, W_{ij_2} = q_{ij_2}, \cdots, W_{ij_{L-1}} = q_{ij_{L-1}}$ として式を連立させ，$a_{i1}, \cdots, a_{iL-1}$ を求める.
6. (2) で選ばれなかった $n-(L-1)$ 台のデータサーバが持つ分散情報 $W_{ij_L}, \cdots, W_{ij_n}$ を，$a_{i1}, \cdots, a_{ik-1}$ を用いて計算する.
7. $i = 3, 4, \cdots, m$ について，(4),(5),(6) を行う.

[復元]

1. 秘密情報 s_i を復元する場合，鍵サーバ $x_{j_l}(1 \leq l \leq L-1)$ は初期値 W_{1j_l}，鍵 key_{j_l} を利用して疑似乱数 q_{ij_l} を生成する.
2. 全ての秘密情報に関する分散情報を持つデータサーバ $x_{j_L} \cdots x_{j_k}$ は対応する分散情報 $W_{ij_L} \cdots W_{ij_k}$ を送信する.
3. 秘密情報 s_i を復元するユーザは受け取った疑似乱数 q_{ij_l} および分散情報 $W_{ij_L} \cdots W_{ij_k}$ を利用し，(k,n) 閾値秘密分散法と同様の手順で秘密情報 s_i を復元する.

これよりこの方式では，最大 $k-1$ 台の鍵サーバの持つ分散情報を最小で初期値して利用する分散情報 W_{1j} 及び鍵 key_j のみを持ち，その他の $n-k+1$ 台のデータサーバは m 個の秘密情報全てに関する分散情報を持つこととなる.

2.2　記憶容量

ここで，鍵のサイズを正の実数 t を用いて $|key_i| = |s_i|/t$ と表すとする時，鍵サーバに必要な記憶容量は

$$|key_i| + |W_i| = (1 + \frac{1}{t})|s_i| \qquad (1)$$

となり，全ての秘密情報に関する分散情報を持つデータサーバに必要な記憶容量は，

$$m \times |W_i| = m \times |s_i| \qquad (2)$$

となる．これより，鍵サーバが l 台であるとして，提案方式においてシステム全体で必要な記憶容量は，

$$(\frac{1}{t}+1)\times l|s_i|+((n-l)\times m|s_i|) \quad (3)$$

となる．

ここで，従来の (k,n) 閾値秘密分散法において，システム全体で必要な分散情報に関する記憶容量は，

$$nm|W_i| = nm|s_i| \quad (4)$$

であるので，提案方式においてシステム全体で必要な記憶容量と比較すると秘密情報の個数である m が十分大きい場合，容量比は以下の様になる．

$$\frac{(1+\frac{1}{t})\times l+m\times(n-l)}{mn} \simeq 1-\frac{l}{n} \quad (5)$$

これより，$l<n$ であるため，提案方式は (k,n) 閾値秘密分散法よりもサーバが持つべき記憶容量を削減することができる．また，l が大きいほど (k,n) 閾値秘密分散法に比べシステム全体として持つべき記憶容量を削減することが可能である．また，ここでは証明は省略するが，安全性についても用いる疑似乱数が計算量的に安全ならば，植松方式も計算量的に安全であることが証明できる．

2.3 クラウド適用における問題点

しかし，この方式をクラウドコンピューティングの様な多くのユーザが参加するシステムに適用することを考えた場合，鍵サーバは一人のユーザに対して一つの分散情報と鍵を保管する必要があるため，ユーザが r 人おり，各ユーザが疑似乱数生成を行う場合には持つべき情報の数は $r\times(1+1/t)|s|$ となってしまう．また復元の際にも，植松方式の場合，ある秘密情報 s_i のみの復元を考えた場合，この秘密情報の分散情報となっている疑似乱数 q_{ij} が出力されるまで鍵サーバは繰り返し初期値 W_{1j} 及び鍵 key_j を用いて疑似乱数生成を行い，ユーザに送信する形となる．そのため，鍵サーバは平均で $m/2$ 回の疑似乱数生成を行う必要がある．

これより植松方式をそのままクラウドシステムに適用することは適切ではないと考えられる．

3 提案方式

前述の植松方式では，まず疑似乱数生成のための初期値 W_{1j} を計算し，その初期値を利用して疑似乱数 $Q=[q_{i1}\cdots q_{ik}](i=1\cdots m)$ を生成した．それに対して，本提案方式は計算量的に安全な暗号方式を利用するという仮定の下，あるユーザの分散情報を秘密情報の ID を暗号化することで生成する疑似乱数系列に等しくなるように全ての分散式の係数列を決定する．

3.1 提案方式の概要

本方式では分散情報を削減する鍵サーバはそれぞれ固有の鍵を持ち，ユーザから送られてきたユーザ ID をその鍵を用いて自身の暗号装置で暗号化し，ユーザは自身の持つデータ ID を利用して疑似乱数を生成する．これにより本方式では鍵サーバは暗号文生成のための鍵のみを持てば良いと言う事になる．

以下に本方式を用いた場合のクラウドシステムの構成図を示す．ここで，前述の鍵情報のみを持つ鍵サーバを $x_1,\cdots,x_l$ とし，すべての秘密情報に関する分散情報を持つデータサーバを $x_{l+1}\cdots x_n$ とする．また秘密情報をクラウドに預けるユーザは r 人とし，ユーザ ID$ID[y](y=1,\cdots,r)$ が割り振られており，それぞれのユーザが m 個の秘密情報 $s_{1j}\cdots s_{mj}(j=1\cdots r)$ を持つとし，これらの秘密情報にもそれぞれデータ ID$dID[s(i)]$ が割り振られているものとする．この様なシステムに植松方式を適用した場合，

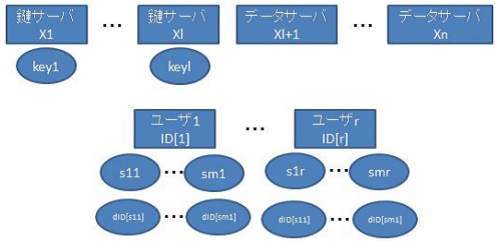

図 1: システム構成図

前述の通り，図中の鍵サーバは各ユーザごとに疑似乱数生成のための初期値W_{1j}と鍵key_jを持つ必要があり，ユーザが多数いる場合には多くの情報を持つ必要がある．また，疑似乱数を生成する際にも，目的の秘密情報に関する分散情報となる疑似乱数が出力されるまで，平均$m/2$回疑似乱数生成を繰り返す必要があるが，提案方式では各ユーザはデータ ID$dID[s(i)]$を持つため，それぞれの秘密情報ごとに疑似乱数を生成することができる．

3.2 提案方式の構成

まず，それぞれの秘密情報$s_i(i = 1 \cdots m)$について以下の分散式を生成する．

$$W_{ij} = s_i + a_{i1}x_j + a_{i2}x_j^2 + \cdots + a_{ik-1}x_j^{k-1} \quad (6)$$

ここで，x_jは各サーバに割り当てられるIDであり，W_{ji}はサーバIDx_jを利用して生成した秘密情報s_iの分散情報である．また，それぞれの秘密情報に割り振られているデータ ID$dID[s(i)]$は秘密情報のデータサイズよりも小さいものとし，これらのデータIDと秘密情報s_iの間には以下の関係があるとする．

$$H(s_i|dID[s(i)]) = H(s_i) \quad (7)$$

本提案方式はユーザが鍵情報のみを持つ鍵サーバに自身のユーザ ID$ID[y](y = 1, \cdots, r)$を送信し，それを受け取った鍵サーバが自身の持つ鍵key_jと受け取ったユーザIDを用いて$Eid(y, j) = Enc(ID[y], key_j)(j = 1 \cdots l)$を生成してユーザに送信する．ここで$Enc(a, b)$はaをbという鍵を用いて暗号化する処理を表すとする．ユーザがこれを用いて自身のデータ ID$dID[s(i)]$を暗号化し暗号化結果
$q_{ij} = Enc(dID[s(i)], Eid(y, j))$
をm個の秘密情報全てについて生成したのち，これらが鍵サーバの分散情報に対応するように$W_{1j} = q_{1j}, W_{2j} = q_{2j}, \cdots, W_{mj} = q_{mj}$としてそれぞれの秘密情報$s_i(i = 1 \cdots m)$に関する分散係数$A' = [a_{i1}, \cdots a_{ik-1}]^T$を求める．一般的に書くと提案方式の構成は以下のようになる．

ここで，分散情報を削減する鍵サーバの台数はクラウドシステム構成時に決定しており，l台$(2 \leq l \leq k)$とする．

[分散]

(1) ユーザは自身の ID$ID[y](y = 1, \cdots, r)$を鍵サーバ$x_1, \cdots, x_l$に送信する．

(2) $ID[y]$を受け取った鍵サーバは自身の持つ暗号装置と鍵key_jを利用して$Eid(y, j) = Enc(ID[y], key_j)(j = 1, \cdots, l)$を生成し，ユーザに送信する．

(3) これを受け取ったユーザは自身の秘密情報に関するデータ ID$dID[s(i)](i = 1 \cdots m)$を用いて疑似乱数
$q_{ij} = Enc(dID[s(i)], Eid(y, j))$
を生成する．

(4) ユーザは(3)で生成した疑似乱数系列$Q = [q_{1j}, \cdots, q_{mj}]^T$及び鍵サーバのID系列

$$X' = \begin{bmatrix} x_1 & \cdots & x_1^{k-1} \\ \vdots & \ddots & \vdots \\ x_l & \cdots & x_l^{k-1} \end{bmatrix} \quad (8)$$

を用いて以下の式から分散係数行列$A'(i) = [a_{i1}, \cdots, a_{ik-1}]$を生成．

$$A'(i) = X'^{-1}Q \quad (9)$$

(5) また，ユーザはデータサーバ$x_{l+1}, \cdots, x_n$に関する分散情報$W_{il+1}, \cdots, W_{in}$を(4)で生成した係数行列を利用して(k,n)閾値秘密分散法と同様の手順により算出する．

(6) ユーザはデータサーバに生成した分散情報$W_{1j}, \cdots, W_{mj}$を送信する．

[復元]

(1) 秘密情報s_iを復元するユーザは全てのサーバ$x_1, \cdots, x_n$から任意のk個のサーバを選択し，選択したサーバに対して自身の$ID[y]$及び秘密情報s_iのデータ ID$dID[s(i)]$を送信する．

(2) 鍵サーバの中で，$(ID[y], dID[s(i)])$を受け取ったサーバは自身の持つ鍵key_j及び暗号装置を利用して

$Eid(y,j) = Enc(ID[y], key_j)$
を生成し，疑似乱数
$q_{ij} = Enc(dID[s(i)], Eid(y,j))$
を生成してユーザに送信する．

(3) データサーバの中で，$(ID[y], dID[s(i)])$ を受け取ったサーバはこれらの ID に対応する分散情報 W_{ij} をユーザに送信する．

(4) サーバにより生成された分散情報及び疑似乱数を受け取ったユーザはそれを利用して (k,n) 閾値秘密分散法と同様の手段で，秘密情報 s_i を復元する．

これより提案方式では各ユーザの ID$ID[y]$ 及び秘密情報 s_i に割り振られた $dID[s(i)]$ を利用することで，鍵サーバが持つ情報は疑似乱数生成のための key_j のみでよく，この鍵情報を鍵サーバが安全に保管するという前提を置いた場合，すべてのユーザに共通で利用することができるため，ユーザの人数及び秘密情報の数に依存せず常に鍵情報のみを持つだけでよいということになる．

また，提案方式と植松方式の大きな違いとして，植松方式の場合疑似乱数 $q_{1j}, \cdots, q_{mj}$ を生成する際，初期値 W_{1j} 及び鍵 key_j から生成するため，あるユーザが秘密情報 s_i のみを復元することを考えた場合，鍵サーバはこの秘密情報に対応する疑似乱数 q_{ij} が出力されるまで，平均 $m/2$ 回の疑似乱数生成を行い，ユーザに送信するため，冗長的な計算が必要であった．それに対し，提案方式では鍵サーバはユーザから受け取ったユーザ ID$ID[y]$ を暗号化 Eid(y,j) を生成し，それを用いて疑似乱数 $q_{ij} = Enc(dID[s(i)], Eid(y,j))$ の生成を行うため，植松方式に比べ計算量の削減を行えるほか，マルチパーティ計算などを行う際にサーバに与える情報量を必要最低限に抑えることができる．

4 評価

ここでは，提案方式を実際のクラウドシステムに適用した場合を想定し，各サーバの持つ記憶容量及び分散情報の安全性について考察を行う．

4.1 記憶容量

提案方式を用いた場合，鍵情報のみを持つ鍵サーバとすべての秘密情報に関する分散情報を持つデータサーバの二通りのサーバが考えられる．ここで，鍵サーバについては前述より一つの鍵 key_j のみとなり，必要な記憶容量は鍵サイズである $|key| = |s|/t$ となる．また，データサーバは $m \times |s|$ の記憶容量が必要となる．これより，提案方式において鍵サーバを $l(2 \leq l \leq k-1)$ 台として提案方式を利用して構成されたシステム全体で必要な記憶容量は以下のようになる．

$$\frac{|s|}{t} \times l + (n-l) \times (m \times |s|) \tag{10}$$

ここで，提案方式では鍵サーバの記憶容量はユーザの参加人数に依存せず常に鍵情報一つ分であるため，クラウドシステムに参加するユーザの数を r 人とすると提案方式を利用した場合に必要な記憶容量は

$$\frac{|s|}{t} \times l + r \times (n-l) \times m \times |s| \tag{11}$$

となる．

これに対して従来方式である植松方式を同様のシステムに用いた場合の記憶容量は式 (3) より以下のようになる．

$$(\frac{1}{t}+1)r \times l|s_i| + r \times (n-l) \times m|s_i| \tag{12}$$

ここで，提案方式及び植松方式において鍵サーバが持つ記憶容量について注目する．すると植松方式の場合鍵サーバに必要な記憶容量は以下の様になる．

$$(\frac{1}{t}+1)r \times l|s_i| \tag{13}$$

次に提案方式の場合には以下のようになる．

$$\frac{1}{t} \times l|s_i| \tag{14}$$

これより，それぞれの方式において鍵サーバにおける容量比は

$$\frac{\frac{1}{t}}{(\frac{1}{t}+1)r} = \frac{1}{(1+t)r} \tag{15}$$

となる．これより提案方式を用いる場合，秘密情報をクラウドに預けるユーザが多ければ多いほど鍵サーバが持つデータ量は小さくなるということがわかる．

4.2 安全性

ここでは，鍵サーバはそのとりうる最大数である $k-1$ 台であるとする．

まず，鍵サーバ $x_1,\cdots,x_{k-1}$ の持つ情報について考える．これらのサーバはユーザから送信されるユーザ ID$ID[y](y=1,\cdots,r)$ 及び自身の持つ鍵 key_j を利用して以下を生成する．

$Eid(y,j)=Enc(ID[y],key_j)$

この時，これらの鍵サーバがユーザから受け取る情報はユーザの ID 情報のみであり，鍵サーバ $x_1,\cdots,x_{k-1}$ はユーザの秘密情報 s_i に関して一切の有益な情報を取得することはない．また，ユーザが利用する暗号方式が計算量的安全性を持つ場合，鍵サーバから送られてきた $Eid(y,i)$ 及びデータ ID$dID[s(i)](i=1,\cdots,m)$ を用いて生成する，秘密情報 s_i に関する疑似乱数行列 $Q(i)=[q_{i1},\cdots,q_{ik-1}]^T$ について以下のことが成立する．

$$|Pr[Z(Q(i))=1]-\frac{1}{p^{k-1}}|<\varepsilon(k-1) \quad (16)$$

ここで Z は任意の多項式計算アルゴリズムであるとする [4]．

次に全ての秘密情報 $s_i(i=1,\cdots,m)$ に関する分散情報を持つデータサーバの分散情報について考える．これらのデータサーバが持つ分散情報は先ほどユーザにより生成された疑似乱数行列 $Q(i)$ を利用して以下の式から求められた係数行列 $A'(i)=[a_{i1},\cdots,a_{ik-1}]^T$ に秘密情報を加えた行列 $A(i)=[s_i,a_{i1},\cdots,a_{ik-1}]^T$ から生成される．

$$A'=X^{-1}Q(i) \quad (17)$$

ここで，鍵サーバの ID 行列 $X=[x_1,\cdots,X_{k-1}]^T$ は公開されており，上記式より係数行列 A' については以下が成立する．

$$|Pr[Z(A')=1]-\frac{1}{p^{k-1}}|<\varepsilon(k-1) \quad (18)$$

また，秘密情報 s_i を加えた行列 A については秘密情報 s_i は一切の情報を持たないそれぞれのサーバから見ると真性乱数と等価であり，以下のことが言える．

$$|Pr[Z(A)=1]-\frac{1}{p^k}|<\varepsilon(k-1) \quad (19)$$

ユーザは上記のような係数行列 A を用いてデータサーバ $X=[x_k,\cdots,x_n]^T$ の持つ分散情報 $W=[Wi1,\cdots,W_{in}]^T$ $(i=1,\cdots,m)$ を生成する．

$$W=XA \quad (20)$$

ここで，生成された分散情報について以下が成立すると仮定する．

$$|Pr[Z(W)=1]-\frac{1}{p^n}|=\kappa \quad (21)$$

この時，分散情報 W と秘密情報を含む係数行列 A の間には式 (20) より以下の関係が成立する．

$$A=X^{-1}W \quad (22)$$

ここで，各サーバの IDX は全てのユーザに公開されているため，秘密情報を含む係数行列 A を求めることは分散情報 W を求めることと等価であるということができる．つまり式 (23) より A について以下のことをいうことができる．

$$|Pr[Z(A)=1]-\frac{1}{p^k}|=\kappa \quad (23)$$

ここで，A に関しては式 (19) の関係が成立しているため，

$$\kappa<\varepsilon(k-1) \quad (24)$$

となる．これより，式 (23) は以下のように書くことができる．

$$|Pr[Z(W)=1]-\frac{1}{p^n}<\varepsilon(k-1) \quad (25)$$

ここで，提案方式において疑似乱数生成に利用する暗号装置が安全である場合，$\varepsilon(m)$ は十分小さいため分散情報 W は確率 $1/p^n$ で一様に選択されているとみなすことができるため，各サーバが自身の持つ分散情報以外の分散情報を求めることは計算量的に困難であり，利用する暗号装置が計算量的な安全性を持つ場合分散された秘密情報についても計算量的な安全性を持つといえる．

4.3 応用例

提案方式ではユーザの秘密情報 s_i に関する分散情報の分散先をすべてサーバとしたが，提案

方式には分散情報を削減し，鍵情報 key_i のみを持つ鍵サーバとすべての秘密情報 m 個に関する分散情報を持つデータサーバが存在している．

ここで，鍵サーバについては単なる暗号装置で構成することができる．具体的には，これらの鍵サーバはユーザから送信されたユーザ ID$ID[y]$ を自身の持つ鍵 key_j を用いて暗号化し，$Eid(y,j) = Enc(ID[y], key_j)$ を生成し，ユーザに送り返す．これを受け取ったユーザが自身の持つ秘密情報に割り振られた $dID[s(i)]$ を用いて疑似乱数 $q_{ij} = Enc(dID[s(i)], Eid(y,j))$ の生成を暗号装置を用いて行うため，鍵サーバが持つ情報は key_j のみであり，何名のユーザがいくつの秘密情報を分散してもこのデータ量が増加することがない．これより，これらのサーバは実質的には送信された $ID[y]$ を暗号化するのみでよく，このような装置はデータの格納容量を持つサーバに比べ，はるかに安価かつ簡単に構成可能である．

提案方式ではこのような分散情報を削減することができる鍵サーバは最大 $k-1$ 台構成することができ，これらすべてをこのような装置で構成することで，従来方式を用いて構成するクラウドシステムよりもより安価に大規模なクラウドシステムを構成することができる．

また，提案方式ではユーザが鍵サーバから送られてきた $Eid(y,j)$ を用いて疑似乱数 $q_{ij} = Enc(dID[s(i)], Eid(y,j))$ を自身の持つ暗号装置を用いて生成した．この様に行うことでユーザは鍵サーバに自身のユーザ ID$ID[y]$ を送信するのみでよいため通信量の削減を行うことができる．しかし，このような処理ではユーザは暗号装置を持つ必要があり，暗号化処理を行うなど負荷が大きい．これに対して，ユーザが鍵サーバに直接データ ID$dID[s(i)]$ を送信し，鍵サーバは受け取ったデータ ID と自身の持つ鍵 key_j を用いて暗号化して，疑似乱数 $q_{ij} = Enc(dID[s(i)], key_j)$ を生成することでも提案方式を構成することができる．この場合，鍵サーバは生成した m 個の疑似乱数 $q_{1j}, \cdots, q_{mj}$ をユーザに送信する必要があるため，通信量が増加するという欠点があるが，ユーザは暗号化装置を持つ必要がなく，負荷が軽減され，かつシステム構成を簡単化することができる．また，安全性についてもデータ ID$dID[s(i)]$ と秘密情報 s_i の間には式 (7) の関係が成立するため，この情報を鍵情報が受け取ったとしても提案方式と同等の安全性を保つといえる．

5 まとめ

本論文では以下の特徴を満たすクラウドコンピューティングに適した秘密分散法の提案を行った．

- 従来の秘密分散法と同様の特性を持つ
- 分散情報を持つサーバの記憶容量削減
- 少なくとも計算量的安全性を持つ

本提方式を利用することで，最大 $k-1$ 台のサーバの持つべき情報を鍵情報のみにすることができる．これにより，従来方式に比べシステム全体で分散データ保管に関するコストを削減することができ，また，計算量的安全性を持つ暗号化装置を利用することでそれぞれの秘密譲情報について計算量的安全性を保つことができる．

参考文献

[1] A. Shamir. How to share a secret. Communications of the ACM, 22, (11), pp.612-613 (1979)

[2] G. R. Blakley. Security of ramp schemes. Crypto ’ 84, pp.242-268 (1984)

[3] 植松裕基，岩村惠市．複数の情報を有するメモリやデータベースに適した秘密分散法. SCIS2010 (2010)

[4] 岡本栄司．暗号理論入門 [第 2 版]．共立出版株式会社．pp.91-95 (2006)

[5] 千田浩司，五十嵐大，濱田浩気，菊池亮，冨士仁，高橋克巳．マルチパーティ計算に適用可能な計算量的ショート秘密分散. SCIS2012 (2012)