Vol.2014-CSEC-65 No.1
Vol.2014-IOT-25 No.1
2014/5/22

秘密分散システムにおける分散データの更新手法

古田 英之[†1]　須賀 祐治[†2]　岩村 惠市[†1]

概要：本論文では, 分散情報の更新手法について考える.shamir の秘密分散法では本来 k-1 の安全性を持っているが, もし, サーバが攻撃された場合, それをそのままにしていれば攻撃に対して k-2 の安全性に落ちてしまう. ゆえに, 攻撃されたということがわかった段階で, 分散情報の更新が必要になってくる．自明な方法としてある, 一度秘密情報を復元し分散するという手段では, 手間や安全面において問題があるので, 秘密情報を復元することなく全ての分散情報を更新することを考える. 従来、閾値 k を変化させ, かつ秘密情報を変化させる更新手法は提案されているが, 本手法は閾値 k や秘密情報を変化させることなく, 全ての分散情報を更新する。また, 分散情報の更新は全てシステム側で実行でき, 秘密情報の持ち主の負担はない.

キーワード：秘密分散法, 分散情報更新

Updating method of distributed data in secret sharing system

Furuta Hideyuki[†1]　Suga Yuji[†2]　Iwamura Keiichi[†1]

Abstract: A secret sharing scheme protects a secret by distributing related information among members in a group. An access structure for a secret sharing defines all subsets of members who are authorized to reconstruct the secret. We discussed the update of the shared information of all without having to restore the secret information. It can respond more efficiently even in the case of changing the threshold value k. The updating of distributed information can be run on the system side all, there is no burden on the owner of the confidential information.

Keywords: A secret sharing scheme, update of the shared information.

1. はじめに

クラウド [1] とは, クラウドコンピューティング (Cloud Computing) の略で, 従来ユーザが自身のデータを自身の端末に保有, 管理していたことに対し, クラウドではユーザはオンライン上のサーバにデータを保有, 管理する. そのため, ユーザは自身のデータを持ち歩くことなしに，ネットワークを介していつでもどこでも必要なデータにアクセスできるようになる．しかしながら, クラウドを使用する際には, 基本的にはすべてのデータがクラウドに集約されるため気を付けなければならない問題点がいくつかある. その一つは, サーバやネットワークの障害などによって，クラウドサービスが利用できなくなってしまう場合である. また, 集中的なデータの管理は様々な情報をクラウドに集約させるため攻撃の標的となりやすく, 情報流出の恐れが出てくる．さらに, ユーザがクラウドへ預けるデータには多くの個人情報や機密情報を含むため, データの安全性について不安を持つユーザも多く存在する．企業などの持つ機密情報はとても多く, これらが漏れると多くの人に様々な被害が生じてしまう．以上の問題を解決するため. クラウドシステムのプライバシ問題を解決するために「秘密分散法」を適用することが考えられている.

秘密分散法 (secret sharing scheme)[2] は, 一つの情報を複数の異なる情報に変換し, そのうちの一定数以上が集まれば元の情報が復元可能だが, その数未満では元の情報は全く復元されないという手法である. これによって, 災害な

†1 現在，東京理科大学
Presently with Tokyo University of Science
†2 現在，株式会社インターネットイニシアティブ
Presently with Internet Initiative Japan Inc.

ⓒ 2014 Information Processing Society of Japan

Vol.2014-CSEC-65 No.1
Vol.2014-IOT-25 No.1
2014/5/22

どによりデータの一部が失われても一定数以下ならば元の情報が復元でき, 一定数以上の情報が漏洩しない限り情報漏洩は起こらないという安全な情報管理システムが実現できる.

また最近, クラウドにおけるサーバの故障に対応する情報消失に対して再生成符号や再分散というものが研究されている. これらは, 故障などによりサーバが失った分散情報を, 秘密情報を復元することなく他のサーバの情報から復元するというものである.

また、サーバの故障対策ではなくサーバの構成の変化に対応して閾値 k や秘密情報を変化させて情報を更新する方法が提案されている. しかし、この方法はサーバ構成を変化させない、すなわち閾値 k を変化させない場合に適用できない. 閾値 k を変化させない情報更新は以下の状況で有用となる. 秘密分散法では k-1 台までのサーバが攻撃されても情報漏えいしないが, 1 つのサーバが攻撃された場合, 分散情報をそのままにしていればそのシステムは k-2 のサーバ攻撃耐性しか持たない. よって, 1 つのサーバに対する攻撃が判明した時点で, サーバ構成を変えずに全サーバの分散情報を更新する必要がある. 本手法は閾値 k や秘密情報を変化させることなく, 全ての分散情報を更新する。また, 分散情報の更新は全てシステム側で実行でき, 秘密情報の持ち主の負担はない.

2. 従来方式

2.1 再分散

? Shamir 秘密分散の分散処理によって生成された K この分散データから，これらを開示することなく新たに異なる分散データを生成する処理を再分散という.

次の 2 つの条件を満たす秘密分散法を，shamir((k,n) 閾値) 秘密分散法と呼ぶ.

(1) 任意の k 個の分散情報から，元の秘密情報 s を復元することができる.

(2) k-1 個以下の分散情報からは，秘密情報 s に関する情報は一切得ることができない.

Shamir の提案した多項式補間による方法では，以下の様にして (k,n) 閾値秘密分散法を実現する.

[分散]

(1) $s < p$ かつ $n < p$ である素数 p を選ぶ.

(2) $GF(p)$ の元から，異なる n 個の $x_i (i = 1, \cdots, n)$ を選びだし，ユーザ ID とする.

(3) $GF(p)$ の元から，$k-1$ 個の乱数 $a_l (l = 1, 2, \cdots, k-1)$ を選んで，以下の式を生成する.

$$W_i = s + a_1 x_i + a_2 x_i^2 + \cdots + a_{k-1} x_i^{k-1} \qquad (1)$$

(4) 上記式 (19) の x_i に各ユーザ ID を代入して，分散情報 W_i を計算し，各ユーザにこれらのユーザ IDx_i と分散情報 W_i を配付する.

[復号]

(1) 復号に用いる分散情報を $W_{if} (f = 1, 2, \cdots, k)$ とする. また, その分散情報に対応するユーザ ID を x_{if} とする.

(2) 分散式に x_{if} と W_{if} を代入し，k 個の連立方程式を解いて，s を得る．s の復元の際には，Lagrange の補間公式を用いると便利である.

(1) 記号の定義

- p : 素数
- $GF(p)$: p を位数とする有限体．本稿では $GF(p)$ の任意の元のサイズは一定と見なす.
- κ : セキュリティパラメータ (推奨地 : $\kappa \geqq 128$)
- $V(R) \in GF(p) : R \in \{0,1\}^k$ をシードとして確定的に得られる疑似乱数
- $|x|$: データ x のビット長
- $x||y$:x と y のデータ連結
- P_i : 分散データの保管やマルチパーティ計算を行う i 番目の主体

(2) Lagrange 補間

$n_i (1 \leqq \mathrm{i} \leqq \mathrm{K})$ および n を 0 以上 p 未満の互いに異なる整数としたとき，ある $K-1$ 次多項式 $f_{(x)} = \sum_{i=0}^{K-1} a_i x^i (a_i \in GF(p)))$ の K 個の座標 $(n_i, f_{(n_i)})$ から，以下の Lagrange 補間式により $f_{(n)}$ を求めることができる.

$$f_{(x)} = \sum_{i=1}^{K} f(n_i) L_i(x) \qquad (2)$$

$$L_i(x) = \prod_{j=1, j \neq i}^{K} \frac{x - n_i}{n_j - n_j}$$

(3) 再分散

(a) $P_i (1 \leqq \mathrm{i} \leqq \mathrm{K})$ は以下を行う.

(i) $\sum_{j=1}^{K} r_{i,j}$ となる乱数 $r_{i,j} \in GF(p)$ を生成し，$r_{i,j}$ を P_i に送信する.

(ii) $s_i = f(i) L_i(K+!) + \sum_{j=1}^{K} r_{j,i}$ を計算し，$f(K+1)$ の分散データとして P_{K+1} に送信する.

(b) P_{K+1} は $f(K+1) = \sum_{i=1}^{K} s_i$ を得る.

2.2 再生成符号

?? 故障サーバの修復問題，つまり，分散データの再生成問題について，必要となる 6 個のパラメータ（n，k，d，α，β，B) を示しながら説明する.

(1) 符号化と保存

サイズ B のデータを符号化し，サイズ α の分散データを n 個作成する．その後，n 個のノードに分散データを伝送する．各ノードでは，サイズ α の分散データを保存する．また，各ノード保存できるデータサイズも α とする

(2) 復元

ⓒ 2014 Information Processing Society of Japan

データコレクターは，元データを復元するために，ネットワーク上のn個のノードの中から任意のk個のノードにアクセスし，各ノードに保存されているサイズαの分散データすべて，またはその一部をダウンロードする．そして，収集した分散データから元データを復元する．

(3) 再生成

故障ノードを修復するために置き換えられた新しいノードは故障していないノードの中から任意にd個のノードにアクセスし，それぞれのノードからサイズβの再生成用データをダウンロードする．したがって，合計でサイズd βのデータをダウンロードすることになり，これを修復バンドワイドという．各ノードは，置き換えられた新ノードからの要求に従って，保存しているサイズαの分散データからサイズβの再生成用データを作成する必要がある．具体的には，α個の分散データの線形結合により再生成用データを作成する．置き換えられた新ノードは，収集した合計サイズd βの再生成用データから故障ノードが保存sていたサイズαの分散データと同じ分散データを再生成し保存する．

(4) メッセージ行列

分散符号化する元データとなるサイズBのメッセージについて説明する．サイズBのメッセージっを以下に示す対称行列の成分に配置することで，メッセージ行列を構成する．

$$S^{(l)} = \begin{bmatrix} u_{1,1}^{(l)} & u_{1,2}^{(l)} & \dots & u_{1,\alpha}^{(l)} \\ u_{2,1}^{(l)} & u_{2,2}^{(l)} & \dots & u_{2,\alpha}^{(l)} \\ \vdots & \vdots & \ddots & \vdots \\ u_{\alpha,1}^{(l)} & u_{\alpha,2}^{(l)} & \dots & u_{\alpha,\alpha}^{(l)} \end{bmatrix}, l = 1, 2 \quad (3)$$

と定義する．ただし，対称行列であることより，任意のi, j ,1≤ i, j $\leq \alpha$に対し,$u_{i,j}^{(l)} = u_{j,i}^{(l)}$である．

2個の対称行列を縦方向に並べた2 $\alpha \times \alpha$メッセージ行列を

$$A = \begin{pmatrix} S^{(1)} \\ S^{(2)} \end{pmatrix} \quad (4)$$

と定義する．

(5) 秘密分散法の適用された再生成符号?

本節では，故障ノードの分散データを再生成する際に収集する再生成用データにおける秘密分散において議論する．結論として，サイズ2αの再生成用データ$d_{h1}, \dots, d_{h2\alpha}$から秘密情報は全く漏れない．また，任意の2個のノード$i_(1), i_(2)$が保存する合計サイズ2αの分散データから秘密情報は全く漏れない．そのために，サイズBのメッセージに秘密情報と乱数を対応させる． メッセージ行列Mのj列目,! $\leq j \leq \alpha$に着目し，2個の対称行列の対角成分を$u_{j,j}^{(1)}$と$u_{j,j}^{(2)}$を並べたベクトルを

$$\underline{u}_{j,j} = [\underline{u}_{j,j}^{(1)}, \underline{u}_{j,j}^{(2)}]^t \in F_q^2, j = 1, \dots, \alpha \quad (5)$$

と定義する．一方，各対角成分より下方の$\alpha - l$個の成分を並べたベクトルを

$$\underline{u}_{S,j}^{l} = \underline{u}_{secret,j}^{l} = [\underline{u}_{j+1,j}^{(l)}, \underline{u}_{j+2,j}^{(l)}, \dots, \underline{u}_{\alpha,j}^{(l)}]^t \in F_q^{\alpha-j} \quad (6)$$

$l = 1, 2,$と定義する．そして$\underline{u}_{S,j}^{(1)}$と$\underline{u}_{S,j}^{(2)}$を並べたベクトルを

$$\underline{u}_{S,j} = \underline{u}_{secret,j} = [(\underline{u}_{s,j}^{(1)})^t, (\underline{u}_{j,j}^{(2)})^t]^t \in F_q^{2(\alpha-j)} \quad (7)$$

$j = 1, \dots, \alpha - 1,$と定義する．最後に，$\alpha - 1$個の$\underline{u}_{S,1}, \underline{u}_{S,2}, \dots, \underline{u}_{S,\alpha-1}$つを並べたベクトルを

$$\underline{u}_S = \underline{u}_{secret} = [\underline{u}_{s,1}^t, \dots, \underline{u}_{s,\alpha-1}^t]^t \in F_q^{\alpha(\alpha-1)} \quad (8)$$

と定義する．

(6) 再生成

故障したノードをノードfとする．置き換えられた新ノードは，故障ノードfが保存していた分散データ$\underline{c}_f$を再生成するために，故障していない任意のd = 2 α個のノード$h_1, \dots, h_{2\alpha}$のそれぞれからサイズβ = 1の再生成用データ

$$\underline{c}_i = [\underline{\psi}_i^t M]^t = \sum_{l=1}^{2} x_i^{(l-1)\alpha} S^{(l)} \underline{\phi}_i \quad (9)$$

をダウンロードする．ここで，$\underline{\phi}_i = [1, x_i, x_i^2, \dots, x_i^{\alpha-1}]^t \in F_q^\alpha$ ただし，各ノードh_pから符号化された再生成用データ$d_{h1}, \dots, d_{h2\alpha}$をダウンロードするには，事前に，故障したノードfに対応する要素情報 を，新ノードからノードh_pに知らせる必要がある． 新オードは，収集したサイズ2αの再生成用データ$d_{h1}, \dots, d_{h2\alpha}$と故障したノードfに対応する$F_q$要素情報$x_f$を用いて，次の2段階の手続きにより，分散データ$\underline{c}_i$を一意に再生成できる．まず，2個の長さ$\alpha$の列ベクトル$\underline{f}_1, \underline{f}_2$を次のように定義する

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \underline{\psi}_{h1} \\ \vdots \\ \underline{\psi}_{h2\alpha} \end{bmatrix}^{-1} \begin{bmatrix} d_{h1} \\ \vdots \\ d_{h2\alpha} \end{bmatrix} \quad (10)$$

このとき，故障したノードが保存していた分散データ$\underline{c}_f$は次の計算で求まる．

$$\underline{c}_i = \sum_{l=1}^{2} x_f^{(l-1)2} \underline{f}^{(l)} \quad (11)$$

上記の設定より，α 個のベクトル $\underline{u}_{1,1},\ldots,\underline{u}_{\alpha,\alpha}$ の合計 2α 個の成分に独立な乱数を対応させる．一方，ベクトル $\underline{u}_S$ の合計 $\alpha(\alpha-1)$ 個の成分に秘密情報を対応させる．乱数も秘密情報も有限体 F_q の要素である．

2.3 閾値 k や秘密情報を変化させての更新法

? 閾値 k や秘密情報 S を変化させる方法での分散情報の更新手法を示す. 変更後の閾値を k_r, 秘密情報を S_r とする.

分散情報は以下の式で作られる.

$$\begin{bmatrix} 1 & x_{i_1} & {x_{i_1}}^2 & \ldots & {x_{i_1}}^{k-1} \\ 1 & x_{i_2} & {x_{i_2}}^2 & \ldots & {x_{i_1}}^{k-1} \\ \vdots & \vdots & & & \vdots \\ 1 & x_{i_k} & {x_{i_k}}^2 & \ldots & {x_{i_1}}^{k-1} \end{bmatrix} \begin{bmatrix} S \\ a_1 \\ \vdots \\ a_{k-1} \end{bmatrix} = \begin{bmatrix} f(x_{i_1}) \\ f(x_{i_2}) \\ \vdots \\ f(x_{i_k}) \end{bmatrix} \quad (12)$$

(1) $k_r < k$

新しい閾値が元の閾値より小さくなる場合.

多項式 $f(x)$ の $a_j(k_r \le j \le k-1)$ を公表する. その後以下を行うことによって分散情報を更新する.

$$\begin{bmatrix} 1 & x_{i_1} & {x_{i_1}}^2 & \ldots & {x_{i_1}}^{k_r-1} \\ 1 & x_{i_2} & {x_{i_2}}^2 & \ldots & {x_{i_1}}^{k_r-1} \\ \vdots & \vdots & & & \vdots \\ 1 & x_{i_k} & {x_{i_k}}^2 & \ldots & {x_{i_1}}^{k_r-1} \end{bmatrix} \begin{bmatrix} S \\ a_1 \\ \vdots \\ a_{k_r-1} \end{bmatrix} = \begin{bmatrix} f(x_{i_1}) - \sum_{j=k_r}^{k-1} a_j x_{i_1}^j \\ f(x_{i_1}) - \sum_{j=k_r}^{k-1} a_j x_{i_2}^j \\ \vdots \\ f(x_{i_1}) - \sum_{j=k_r}^{k-1} a_j x_{i_{k_r}}^j \end{bmatrix} \quad (13)$$

上記式により, 閾値 k_r の分散情報が生成される.

(2) $k \le k_r < n$

新しい閾値が元の閾値より大きく n より小さくなる場合.

新しい秘密情報 $S_r \neq S$ と乱数 $b_j(1 \le j \le n)$ を置く.

$$\begin{bmatrix} x_1 & x_1^2 & \ldots & {x_1}^n \\ x_2 & x_2^2 & \ldots & {x_2}^n \\ \vdots & \vdots & & \vdots \\ x_n & x_n^2 & \ldots & {x_n}^n \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} f(x_{i_1}) - S_r \\ f(x_{i_1}) - S_r \\ \vdots \\ f(x_{i_1}) - S_r \end{bmatrix} \quad (14)$$

とすると $f_r(x) = S_r + \sum_{j=1}^{n} b_j x^j$ と表せ. 分散情報の更新が可能となる.

(3) $k \le n < k_r$

新しい閾値が元の閾値よりも n よりも大きくなる場合.

新しい秘密情報 $S_r \neq S$ と乱数 $b_j(1 \le j \le k_r-1)$ を置く.

$$\begin{bmatrix} x_1 & x_1^2 & \ldots & {x_1}^{k_r-1} \\ x_2 & x_2^2 & \ldots & {x_2}^{k_r-1} \\ \vdots & \vdots & & \vdots \\ x_n & x_n^2 & \ldots & {x_n}^n \\ x_{n+1} & x_{n+1}^2 & \ldots & {x_{n+1}}^{k_r-1} \\ \vdots & \vdots & & \vdots \\ x_{k_r-1} & x_{k_r-1}^2 & \ldots & {x_{k_r-1}}^{k_r-1} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \\ b_{n+1} \\ \vdots \\ b_{k_r-1} \end{bmatrix} = \begin{bmatrix} f(x_1) - S_r \\ f(x_2) - S_r \\ \vdots \\ f(x_n) - S_r \\ y_{n+1} - S_r \\ \vdots \\ y_{k_r-1} - S_r \end{bmatrix} \quad (15)$$

とすると $f_r(x) = S_r + \sum_{j=1}^{n} b_j x^j$ と表せ. 分散情報の更新が可能となる.

3. 提案方式

これまで述べてきたように従来の方式で考慮されていたのは，サーバが持つ分散情報が自然災害やヒューマンエラーによって消失した場合にそのデータを元に戻す、または閾値 k を変化させてサーバ構成を変えるということに重点が置かれていた．今回は，攻撃者がサーバを攻撃し分散情報を盗んだ場合について考える．shamir の秘密分散法では本来 k-1 の安全性を持っているが，もし，サーバが攻撃された場合，攻撃者にとっては k-2 の安全性に落ちてしまう．ゆえに，攻撃されたということがわかった段階で，分散情報の更新が必要になってくる．従来方式である閾値 k や秘密情報を変化させての更新法では, 閾値 k を変更しなければならなかったり, 秘密情報を変更しなければならなかったりした. 本提案方式では, 閾値 k や秘密情報を変えることなく分散情報の更新を行う.

3.1 アルゴリズム

上記従来方式内で示した (k,n) 閾値秘密分散法に一定の操作を加え、分散情報の更新を行う．条件として，攻撃されたサーバの分散情報は変更されていないとする．

以下に本提案方式の簡単な構成手順を示す．ここで利用する秘密分散法は 2 章において説明した (k,n) しきい値秘密分散法を適用するものとする．

(1) 攻撃を受けていない k-1 個のサーバは各々新しい share

となる分散情報 $Wnew_i$ を独立に乱数によって定める. ただし、$Wnew_i$ は式 (16) の関係を持つとする。

$$Wnew_i = s + a_{new_1}x_i + a_{new_2}x_i^2 + \cdots + a_{new_{k-1}}x_i^{k-1} \quad (16)$$

(2) 上記 k-1 個のサーバは新しい分散情報 Wnewi と今までの分散情報 Wi との差をとり、式 (17) の値を残りの n-k+1 個のサーバに送る.

$$u_i = (a_{new_1} - a_1)x_i + (a_{new_2} - a_2)x_i^2 + \cdots + (a_{new_{k-1}} - a_{k-1})x_i^{k-1} \quad (17)$$

(3) n-k+1 個のサーバは送られてきた k-1 個の差分から式 (17) を連立させて各乱数部分の変化量 $(a_{new_i} - a_i)(i = 1, \cdots, k-1)$ を計算し自らの分散情報へ加え、新たな分散情報とする. すなわち、n-(k-1) 個のサーバは得られた $(a_{new_i} - a_i)$ に自分の ID である x_i を乗算し、元の分散情報に加えることで新たな分散値

$$Wnew_i = s + anew_1x_i + anew_2x_i^2 + \cdots + anew_{k-1}x_i^{k-1} \quad (18)$$

を得る.

(4) 以前の分散情報

$$W_i = s + a_1x_i + a_2x_i^2 + \cdots + a_{k-1}x_i^{k-1} \quad (19)$$

を消去して更新完了. この方式によって, 分散データが盗難されたことがわかったら, 秘密情報を一度も復元することなく, 容易に分散データの更新を可能とした.

3.2 計算量と通信料

本章では閾値 k や秘密情報を変化させての更新法と, 提案方式の計算量と通信量の比較を行う. 閾値 k や秘密情報を変化させての更新法では, 閾値 k が小さくなる場合と大きくなる場合とで計算量が異なる.k が小さくなる場合は,k 次方程式を解く計算が必要である.k が大きくして kr にする場合には,kr-1 次方程式を解く必要がある. これと比べて提案方式では k-1 次方程式を解くことで分散情報の更新が可能である.

次に通信料について比較する. 閾値 k や秘密情報を変化させての更新法では,n 個の分散データを受信,n 個の分散データを送信する必要がある. これに比べて提案方式の通信料は k- 1 台のサーバが n-(k-1) 台のサーバに差分を送信する必要がある.

また計算量, 通信料ともに閾値 k や秘密情報を変化させての更新法は, サーバが連立方程式を解かない場合, ユーザとサーバが通信し, ユーザが計算したのちにサーバと通信が必要であるが, 提案方式ではサーバ間の通信のみで, サーバが計算をすることで更新が可能である.

3.3 安全性

本章では提案方式の安全性について議論を行う. まず本提案方式では古い分散情報を新しい分散情報に置き換えるという方法で分散情報の更新を実現した. 具体的な分散情報の更新方法については，$k-1$ 台のサーバに関する新しい分散情報を真性乱数を用いて生成し，その後分散式における係数を決定することで，残りの $n-k+1$ 台のサーバへの分散情報の更新を行った.

ここで，本提案方式の安全性を示すため，提案方式を用いて新しく更新された分散情報が元の shamir の秘密分散法により生成された分散情報と同等に情報量的安全性を持つことを示す.

まず，shamir の秘密分散法について考える．shamir の秘密分散法では以下の分散式より分散情報の生成を行った.

$$W_j = s + a_1x_j + a_2x_j^2 + \cdots + a_{k-1}x_j^{k-1} \quad (20)$$

これより shamir の秘密分散法によって求められる分散情報は真性乱数を係数とした $k-1$ 次方程式により定まっているため，これから求められる分散情報も真性乱数と等価であると言える．ゆえに以下の式が成り立つ,

$$H(s) - H(s|W_{i1}, W_{i2}, \cdots, W_{ik-1}) = 0 \quad (21)$$

これに対して，提案方式により定められる分散情報について考える．まず，最初に分散情報を更新する $k-1$ 台のサーバの持つ分散情報は真性乱数を用いて決定される．秘密情報も真性乱数であると仮定すれば, 式 (18) における a_{new_i} も真性乱数であるため、式 (18) を用いて得られる $Wnew_i(i = k, \cdots, n)$ も真性乱数と等価と考えられる。より詳細には、最初に分散情報を更新する $k-1$ 台のサーバは秘密情報 s や自分が持つ分散情報 Wi と独立に Wnewi(i=1,⋯,k-1) を定めるので下記が言える.

$$H(s) = H(s \mid Wnew_1, \cdots, Wnew_{k-1}) \quad (22)$$

$$H(s) = H(s \mid Wnew_1, \cdots, Wnew_{k-1}, W_1, \cdots, W_k) \quad (23)$$

H(s)=H(s—Wnew1,⋯,Wnewk-1) H(s)= H(s—Wnew1,⋯,Wnewk-1,W1,⋯,Wk) このとき、各サーバは従来の Wi(i=1,⋯,k-1) から式 (17) を計算して残りのサーバに送り、残りのサーバは乱数部分の変化量

$$anew_1 - a_1, anew_2 - a_2, \cdots, anew_{k-1} - a_{k-1} \quad (24)$$

を計算するが、ai(i=1,⋯,k-1) は真性乱数であり、残りのサーバはその値を知らないので、乱数の変化量から anewi(i=1,⋯,k-1) を得ることはできない. よって、手順 (3) によって得られる残りのサーバの Wnewi(i=k,⋯,n) も真性乱数と等価と言える.

ⓒ 2014 Information Processing Society of Japan

Vol.2014-CSEC-65 No.1
Vol.2014-IOT-25 No.1
2014/5/22

4. まとめ

本論文では，秘密分散法の分散情報の更新手法について提案した. 従来考えられていた更新手法では, 閾値 k を変更させる必要や, 秘密情報の変更が必要であった. 閾値 k を変えることは, 秘密分散法のシステムを変更するということになる. 本方式では, 閾値 k を変更することなくさらに秘密情報の変更も必要としないものであり, 更新前のシステムの変更を必要としないものとした. また, 従来ではユーザの関与を必要としたが, 本方式ではユーザの関与しないシステムでのみの更新を可能とした.

謝辞

本論文作成にあたり，御指導を受け賜りました姜玄浩助教授に心から感謝を致します．また，有益な助言や，アイデアを下さった東京理科大学岩村研究室の皆様に心から感謝いたします.

参考文献

[1] 菊池尚也，金井敦，谷本茂明，佐藤周行．秘密分散を用いた複数クラウドのデータ管理方式．SCIS2014 Jan.2014
[2] A. Shamir. How to share a secret. Communications of the ACM, 22, (11), pp.612-613 (1979)
[3] 千田浩司，五十嵐大，濱田浩気，菊池亮，富士仁，高橋克己．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 3B3-2，Feb.2012
[4] Dimakis, Alexandros G., et al. ”Network coding for distributed storage systems.” Information Theory, IEEE Transactions on 56.9 (2010): 4539-4551.
[5] Rashmi, Korlakai Vinayak, Nihar B. Shah, and P. Vijay Kumar. ”Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction.” Information Theory, IEEE Transactions on 57.8 (2011): 5227-5239.
[6] 栗原正純，桑門秀典．”分散ストレージにおける再生成符号と秘密分散について.” 電子情報通信学会技術研究報告. IT，情報理論 110．363 (2011) : 13-18.
[7] Yuko TAMURA, Mitsuru TADA, Eiji OKAMOTO. Update of Access Structure in Shamir’s(k,n) Threshold Scheme.SCIS’99 Jan,26-29, (1999):469-474

ⓒ 2014 Information Processing Society of Japan