不正成功確率を0にできる検証機能付き秘密分散法

岩村惠市[1]　井上穂乃香

概要：Shamir の(k,n)閾値秘密分散法（以降，Shamir 法）は$k-1$個までの分散値漏洩に対して耐性を持つが，復元において 1 個でも偽の分散値が含まれると正しい結果を復元できない．そのために，復元結果が正しいかどうかを検証する機能を持たせた秘密分散法が多数提案されている．また，従来の検証方式では著者らが知る限り不正成功確率を 0 にできるものはない．不正成功確率とは偽の分散値を含んで復元する場合，偶然の一致を含み不正が検証できない確率を指す．一方，Shamir 法はリード・ソロモン符号（以降 RS 符号）に基づく構成法も知られており，偽の分散値に対する耐性を持たせることができるが，変換処理などが必要であり効率的でない．そこで本論文では，RS 符号化をせず Shamir 法をそのまま用いて不正成功確率を 0 にできる検証機能付き秘密分散法を提案する．この手法は$n>k$であればサーバは秘密情報の分散値以外持つ必要がなく，復号において秘密情報の復元を複数回行うだけであるので単純であり，CDV モデルと OKS モデルのどちらにも適用できる．

キーワード：Shamir 法，秘密分散，検証可能，不正成功確率

Verifiable Secret Sharing Scxheme that can reduce the false success probability to zero

KEIICHI IWAMURA[1]　　HONOKA INOUE

Abstract: Shamir's (k,n) threshold secret sharing scheme (Shamir's scheme) is resistant to leakage of shares up to k-1. However, at the reconstruction phases, the correct secret cannot be achieved even with one false share. Therefore, a number of verifiable secret sharing methods have been proposed. In addition, to the best of our knowledges, none of the conventional methods can reduce the false success probability to zero. False success probability is defined as the probability that a false share cannot be detected when it appears during reconstruction, including coincidence cases. Meanwhile, Shamir's scheme is known that can be constructed based on Reed-Solomon codes (RS codes), which let shares have a resistance to false shares but it is not efficient because of the conversion process required. Therefore, we propose verifiable secret sharing scheme that can reduce the false success probability to zero without RS coding. With this scheme, servers does not need to have secret information other than shares of the secret if n>k. Furthermore, it is simple as it only requires multiple reconstructions of the secret, and it is also applicable to both the CDV and OKS models.

Keywords: Secret Sharing, Verifiable, Server ID, Cloud

1. はじめに

データを秘匿して保存する手法の一つとして秘密分散法が知られている．(k,n)閾値秘密分散法と呼ばれる手法は 1 個の秘密情報をn個に分散して，n台のサーバに預け，その中からk個$(k \leqq n)$の分散値を集めることで元の秘密情報を復元できるが，k個未満の分散値からは一切秘密情報に関する情報を得ることができないという情報理論的安全性を持つ[1]．このような秘密分散法として，Shamir による(k,n)閾値秘密分散法（以降，Shamir 法）が良く知られている．

ただし，秘密分散法の問題点として，攻撃者がサーバを乗っ取るなどして，復元時に偽の分散値を出力した場合，正しい秘密情報が復元されず，そのままでは復元者はそれを検証することができないという点がある．そのため，秘密分散法における復元情報の検証法が多数提案されてきた[3][4][5][6][7][10]．これらは各サーバが秘密情報の分散値以外に，秘密情報と特別な関係を持たせた認証子と呼ばれる情報の分散値を持たせて検証する場合が多い．しかし，認証子は秘密情報と特別な関係を持たせるため用いる体や攻撃者数等に制限を生じさせる．また，サーバが認証子に関

1　東京理科大学　〒125-8585　東京都葛飾区新宿 6-3-1
Tokyo University of Science
6-3-1, Niijuku Katsushikaku, Tokyo 125-8585, Japan*

する分散値を保存するために，サーバの記憶容量が増加する．また，不正サーバまで特定できる従来方式は少ない．

一方，著者らが知る限り，従来方式で不正成功確率を0にできるものはない．不正成功確率とは偽の分散値を含んで復元する場合，偶然の一致を含み不正が検証できない確率を指す．すなわち，従来方式では復元された偽の秘密情報と偽の認証子が定められた関係を満たす場合不正が成功し，その確率を不正成功確率としている．

そこで，本論文では Shamir 法の復元情報に対する検証能力を検討する．その結果，分散においては秘密情報を分散するだけで認証子のような別の情報を必要とせず，$n > k$であれば復元回数を増すだけで，復元結果を効率的に検証できることを示す．また，ある条件を満たせば偽の分散値を出力するサーバを特定することも可能であり，不正成功確率を0にすることができるという従来方式にない大きな特徴をもつ．

以降，2 章では従来方式，3 章では本研究を理解するために必要な関連研究を示し，4 章において不正成功確率を 0 にできる Shamir 法を用いた検証方式を示す．5 章において考察を行う．

2. 従来方式

以下では,用いる有限体の要素数をpとする.

2.1 秘密情報を指数乗する認証子方式[3][4]

秘密情報sを二乗した$a = s^2$を認証子として，sとaを秘密分散し，復元結果が$a = s^2$の関係を満たせば不正無しとする[3]. しかし，$GF(2^2)$においては秘密情報の検証ができない. なぜならば，攻撃者が秘密情報の分散値と認証子の分散値を改竄すると，偽の秘密情報$s' = s + \Delta s$と偽の認証子$a' = a + \Delta a$が復元される．この際,攻撃者が不正を成功させるには$a' = (s')^2$とする必要がある．しかし，$GF(2^2)$においては，攻撃者は$\Delta s = 2s \cdot \Delta s + (\Delta s)^2 = (\Delta s)^2 = 0$となるような誤差を生成することができるため，攻撃者の不正成功確率は 1 となる.よって，$GF(2^2)$は用いられないという制限を持つ. また，$GF(2^2)$を用いなくても偽の分散値を用いた復元結果が偶然$a' = (s')^2$となる場合があるため，不正成功確率は $1/p$となる.また,秘密情報に対する分散値の他に認証子に対する分散値を保管するため，各サーバの記憶容量は 2 倍になる.同様に,方式[4]も$GF(3^m)$の時には，攻撃者の不正成功確率が 1 になるため，任意の体を設定できないという制限を持つ.

2.2 秘密情報をビット列分解する認証子方式[5][6]

方式[5]は，$GF(2^{2m})$における検証に特化したものである．秘密情報sを半分に分解し，ビット列$s = (s_1, s_2)$を生成し，秘密情報sと認証子$a = s_1 \cdot s_2$を秘密分散する. 復元された秘密情報s'と認証子a'が，$a' = s_1 \cdot s_2$であれば，復元結果は正当であるとし，一致しなければ不正とする. 問題点としては，任意のビット列に適応できないことがある. 方式[5]では，ビット長が偶数，かつ，秘密情報が$s \in GF(2^{2m})$時のみに有効である. また，方式[6]も，秘密情報をNビットに分解するが，$s \in GF(2^{Nm})$時のみ有効であり，任意の体を設定できない. さらに，認証子の分散値も保存するため記憶容量が増加する．また，不正成功確率は$1/p$である.

2.3 分散値に関する関数や乱数使用の認証子方式[7]

[7]は秘密情報を操作した認証子を用いないため用いる体 に制限はないが，攻撃者数が制限される.秘密情報sを$k-1$次の多項式$f(x)$で分散し，$h(x) = f(x) + bg(x)$が$k-3$次式 となるようなスカラ量bと$k-1$次式の$g(x)$を設定し，秘密情 報sを$g(x)$でも分散する(これが認証子に相当する). 復元時 に復元者はラグランジュの補間公式を用いて$f(x)$と$g(x)$を 復元する.そして，$f(x) + bg(x)$が$k-3$次式となる乱数bが あれば,秘密情報sを復元する.問題点としては，攻撃者数が$k-1$人の場合，秘密情報が漏洩するため最大$k-2$人に制限 されることや,認証子の分散値に対する記憶容量が増加す ることである．また，不正成功確率は $1/p$である.

2.4 認証子をサーバがブロードキャストしあう方式[8]

各サーバは,受け取った秘密情報のシェアs_iを$s_i = s_{i,0} + s_{i,1}$に分割する. $s_{i,1}$を各サーバが保管し，$s_{i,0}$を検証用の情報とする.各サーバは，$A_i(0) = s_{i,1}$となるような$k-1$ 次式の$A_i(x)$を用意する.さらに,各サーバは$B_i(0) = b_{i,0}$となるように乱数$b_{i,0}, b_{i,1}, \cdots, b_{i,k-1}$を用いて$k-1$ 次式の$B_i(x)$も用意する.そして，$C_{i,j} = g^{s_{i,j}} h^{b_{i,j}}$を計算し，$A_i(j), B_i(j), C_{i,j}$を各サーバがブロードキャストしあう.復元時には,各サーバが収集した$A_i(j), B_i(j), C_{i,j}$から，$g^{s_{i,j}} h^{b_{i,j}} \bmod p = \prod_{l=1}^{k} C_{i,l}^{j^l}$が成り立つかどうかを検証する.成り立てば，$s_{i,0}$が正しいと判断し,各サーバは秘密情報$s$の復元を行う. [8]では以上のようにサーバに大きな負担を強いる.また，検証は離散対数問題に基づいて行われる.秘密情報を含む$g^{s_{i,j}}$は直接公開されないため，情報理論的安全性が実現されるとしているが，検証については計算量的安全性になると考えられる.不正成功確率については記述されていないが，$1/p$と考えられる.

2.5 3 つ以上の認証子を用いる方式[10]

秘密情報のディーラは,秘密情報の分散値$d_{i,0} = f(x_i)$だけ でなく,任意の乱数$d_{i,1} \sim d_{i,k-1}$を生成し，合わせて$n \times k$次元 ベクトルを用意する$(i = 1, \ldots, n)$.さらに，乱数$g_{j,i}$を用意し て以下の$b_{j,i}$を計算し$(j = 1, \ldots, n, i \neq j)$, $(g_{j,i}, b_{j,i})$の組を$n-1$個つくり，その組をn台のサーバに配布する.よって，n台 のサーバは秘密情報の分散値に加えて$2(n-1)$個の情報を保管する.

$$b_{j,i} = g_{j,i}d_{i,0} + \alpha_j d_{i,1} + \cdots + \alpha_j^{k-1} d_{i,k-1} \ (i \neq j).$$

そして，復元時には各参加者が$d_{i,j}$と$(g_{j,i}, b_{j,i})$を用いて検証することで分散値の正当性と不正者を特定する．[10]では各サーバが分散値以外の検証用情報(認証子)を多数保存する必要があるため記憶容量が増加することが問題である．また，不正成功確率はほぼ$1/p$としている．

3. 関連研究

3.1 Shamir の秘密分散法

［分散］

秘密情報sからn個の分散値を計算し，n台のサーバにその分散値を各々配布する．Shamir 法は任意の体$GF(p^m)$に適用できる．以下に，素体$GF(p)$における手法を示す．

1. $s < p$かつ$n < p$である任意の素数pを選択する．
2. $GF(p)$から異なるn個の値$x_1, \ldots, x_n$を選択し，サーバ ID とする．
3. $GF(p)$から，$k-1$個の乱数$a_1, \ldots, a_{k-1}$を生成し，下記$k-1$次多項式$f(x)$を作る．

$$f(x) = s + a_1 x + \cdots + a_{k-1} x^{k-1} \pmod p \qquad (1)$$

4. $f(x)$にn台のサーバ ID $x_1, \ldots, x_n$を代入し，n個の分散値 $W_1, \ldots, W_n$を計算する．

$$W_i = f(x_i) \pmod p \qquad (2)$$

5. 各サーバに分散値を配布する．
   ただし，サーバ ID $x_1, \ldots, x_n$は，公開情報とする．

［復号］

1. 復号に用いる分散値を$W_i (i = 1,2, \cdots, k)$とする．また，その分散値に対応するサーバ ID を$x_i (i = 1,2, \cdots, k)$とする．
2. 復元者は，k個の(x_i, W_i)を集め，分散式にx_iとW_iを代入し，Lagrange の補間公式を用いてsを得る．

3.2 RS 符号に基づく秘密分散法

Shamir 法はリード・ソロモン符号を用いることで実現可能であることが知られている[11]．これに基づき，情報の分散化を行う．以下に，$GF(p)$における手法を示す．

$GF(p)$において，αを原始元とする．符号長$n = p-1$のリードソロモン符号を考える．まず生成行列Gと検査行列Hは，

$$G = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \cdots & & \alpha^{(n-1)} \\ 1 & \alpha^2 & \cdots & & \alpha^{2(n-1)} \\ 1 & \vdots & \ddots & & \vdots \\ 1 & \alpha^{k-1} & \cdots & & \alpha^{(k-1)(n-1)} \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & \alpha & \cdots & \alpha^{(n-1)} \\ 1 & \alpha^2 & \cdots & \alpha^{2(n-1)} \\ 1 & \vdots & \ddots & \vdots \\ 1 & \alpha^{n-k} & \cdots & \alpha^{(n-k)(n-1)} \end{bmatrix}$$

となる．これらにより情報系列$\boldsymbol{A} = (a_0, a_1, \ldots, a_{k-1})$に対して RS 符号化を実行すると，$\boldsymbol{w} = (w_1, \ldots, w_n) = \boldsymbol{A}G$となる．また，多項式表現をすると，それぞれ以下のように表せる．

$$f(x) = a_0 + a_1 x + \cdots + a_{k-1} x^{k-1}$$
$$W(x) = w_0 + w_1 x + \cdots + w_{n-1} x^{n-1}$$

ここで，情報系列の後ろに$n-k$シンボルの 0 を付け加えてn次元ベクトル$\boldsymbol{A'} = (a_0, a_1, \ldots, a_{k-1}, 0, \cdots, 0)$とすると，これは符号語$\boldsymbol{w}$をフーリエ変換したものとなっている[12]．

$$A'_j = \sum_{i=0}^{n-1} w_i \alpha^{ij} \ (j = 0,1, \cdots, n-1)$$

$\boldsymbol{A'}$の多項式表現は，k次から$n-1$次の係数が$a_k = a_{k+1} = \cdots = 0$であることから，$\boldsymbol{A}$と同様に$f(x)$により表現できる．さらに，上式は符号多項式に根を代入した$W(\alpha^j)$に他ならない．したがって，以下のように書き換えられる．

$$\boldsymbol{A'} = (W(1), W(\alpha), \ldots, W(\alpha^{n-1}))$$

これをフーリエ逆変換すると，$\boldsymbol{A'}$から$\boldsymbol{w}$が復元できる．

$$\boldsymbol{w} = (f(1), f(\alpha), \ldots, f(\alpha^{n-1}))$$

$a_0 = s$と置くことで，これはすなわち RS 符号に基づいて情報が分散化され，3.1 の Shamir 法が実現されている．

3.3 Lagrange の補間公式

$x_j (j = 1,2, \cdots, n)$をサーバ ID とする．一般に，x_jは公開情報であり，W_jはサーバx_jに対応する分散値である．よって，k台のサーバが出力した分散値$W_i (i = 1, \ldots, k)$とそのサーバ ID であるx_iがわかれば下記式が得られ，$x = 0$を代入することにより秘密情報sが得られる．

$$W(x) = \sum_{i=1}^{k} W_i \prod_{z \neq i} \frac{(x - x_z)}{(x_i - x_z)} \qquad (3)$$

しかし，悪意ある攻撃者が，自らの持つ分散値W_iを偽の分散値$W_i + \Delta W_i$として送信した場合は以下のようになり，誤差が生じる．ただし，$f_i(x) = \prod_{z \neq i} (x - x_z)$とする．

$$W'(0) = W(0) + \sum_{i=1}^{k} \Delta w_i \frac{f_i(0)}{f_i(x_i)} \qquad (4)$$

例えば，1 回目にサーバx_iが偽の分散情報$W_i + \Delta W_i$を送信し，2 回目にサーバx_jが偽の分散情報$W_j + \Delta W_j$を送信した場合，以下のように秘密情報が復元される．

$$W'(0) = W(0) + \Delta w_i \frac{f_i(0)}{f_i(x_i)} \qquad (5)$$

$$W''(0) = W(0) + \Delta w_j \frac{f_j(0)}{f_j(x_j)} \qquad (6)$$

このとき，攻撃者は公開されているサーバ ID $x_1, \ldots, x_n$から，$W'(0)$と$W"(0)$が一致するよう誤差$\Delta W_i, \Delta W_j$を式(8)のように調整することで偽の秘密情報$S' = W'(0) = W"(0)$を攻撃者の意図によって復元させることができる．

$$W'(0) - W''(0) = \Delta W_i \frac{f_i(0)}{f_i(x_i)} - \Delta W_j \frac{f_j(0)}{f_j(x_j)} = 0 \qquad (7)$$

$$\Delta W_i = \Delta W_j \frac{f_j(0)}{f_j(x_j)} \cdot \frac{f_i(x_i)}{f_i(0)} \qquad (8)$$

4. 提案方式

4.1 アルゴリズム

(k,n)Shamir 法を用いる．また，n台のサーバ ID $x_1,\dots,x_n$は公開されており 0 でなく全て異なる．また, $n>k$とする．

[分散]

1. オーナは(k,n)Shamir 法を用いて秘密情報sをn台のサーバに秘密分散する．

[復元・検証]

1. 復元者はn台のサーバから分散値を得て，$k+1$台のサーバからなるサーバ組に対して組み合わせを変えて，k回の復元を行う．
2. 全ての復元結果が一致しない場合，そのサーバ組は不正サーバを含むとし，一致する場合そのサーバ組を構成するサーバは正当としてその復元結果を採用する．
3. 異なる$k+1$台のサーバ組に対して同様にk回の復元を繰り返す．
4. n台のサーバ中，全ての$k+1$台のサーバの組合せを尽くした結果，同じサーバ組でk回の復元が一致したサーバ組に含まれるサーバを正当なサーバとし，それ以外のサーバを不正サーバとする．

4.2 提案方式の安全性

提案方式は(k,n)Shamir 法をそのまま用いているため, 攻撃者が$k-1$台までのサーバ情報しか得られない場合，情報漏洩（機密性）対する安全性は保証される．また，完全性に対する安全性は以下となる. 攻撃者はn台中$e(<k)$台のサーバを乗っ取り，偽の分散値$W_j+\Delta w_j(j=1,\dots,e)$を出力するとする．秘密情報$s$の復元に対しては以下が言える．

[定理 1]

同じ秘密情報に対する分散値をもつ$k+1$台のサーバを用いて組合せを変えてk回の復元を Lagrange 補間によって行う．その中に偽の分散値を出力する$k-1$台までの不正サーバがある場合, k回の復元結果を一致させることはできない．よって，k回の復元結果が一致すればその$k+1$台のサーバに不正サーバは無く，その復元結果は正しい．一致しなければその$k+1$台のサーバに不正サーバを含む.

[証明]

Lagrange の補間公式より正しい復元結果とのずれΔyは，

$\Delta y=\sum_{i=1}^{k}\Delta w_i\frac{f_i(0)}{f_i(x_i)}=\sum_{i=1}^{k}\Delta w_i\prod_{z\neq i}\frac{(x_z)}{(x_i-x_z)}$で表される．ただし, $f_{zi}(x)=\prod_{z\neq i}\frac{(x-x_z)}{(x_i-x_z)}$は基底多項式であるため各々独立である．$k+1$台中$k-1$台のサーバからの分散値が偽で，それを用いてサーバの組合せを変えたk回の復元を行うとする．このとき, そのk回の復元結果を一致させることができれば不正が成功する．よって，以下が成り立てばよい．ただし，$f_{zi}(0)=f_{zi}$と表し，Δw_iが用いられない場合$f_{zi}=0$とする．

$$\begin{bmatrix} f_{11} & \dots & f_{1k-1} \\ \vdots & \dots & \vdots \\ f_{k1} & \dots & f_{kk-1} \end{bmatrix}\begin{bmatrix} \Delta w_1 \\ \vdots \\ \Delta w_{k-1} \end{bmatrix}=\begin{bmatrix} \Delta y \\ \vdots \\ \Delta y \end{bmatrix} \quad (9)$$

ここで，Δw_iを未知数とすると，$k-1$個の未知数に対して, 全て異なる基底多項式を用いてk個の方程式を成り立たせる解は存在しない．すなわち，基底多項式f_{zi}による式(9)を成立させる解は$\Delta w_i=0,\Delta y=0$以外にない．よって, 不正成功確率は 0 となる．

[系 1]

$n>k$であれば$k+1$台のサーバの組合せを変えてk回以上の復元が可能である．

[証明]

${}_{k+1}C_k=\frac{(k+1)!}{k!}=k+1>k$となるため，$k$回以上の復元が可能である．$n=k$の場合 ${}_nC_k$=1 であるので，nは$k+1$以上必要である．

[定理 2]

偽の分散値を出力する不正サーバの数をeとすると，$n-e\geq k+1$であれば不正サーバをすべて特定でき，正しい復元結果を得ることができる．

[証明]

不正サーバの数をeとすると，正当なサーバは$n-e$台存在する．$n-e=k+1$であれば，そのサーバ組による復元結果は全て一致する．よって，$n-e\geq k+1$であれば正当なサーバを特定できる．それに含まれないサーバは不正サーバとなる．また，一致した復元結果は定理 1 より正しいことが言える．

[系 2] $e<k$より$n\geq 2k$であれば，必ず不正サーバを検出でき，正しい復元結果を得ることができる．

[系 3] 最大の復元回数は ${}_nC_k$ となる．

5. 考察

5.1 具体例及び提案方式の特徴

簡単な例を示す．$n=3,k=2$とすれば，$k+1=3$台のサーバ中の2台の組合せを変えて式(10)～(12)が復元される．ただし，不正サーバは$k-1=1$個であるので，x_3を偽の分散値$W'_3=W_3+\Delta w_3$を出力する不正サーバとする．また，$W_{ij}(0)$をサーバx_i,x_jによる復元結果とする．

$$W_{12}(0)=W(0) \quad (10)$$

$$W_{13}(0)=W(0)+\Delta w_3\frac{(-x_1)}{(x_3-x_1)}=W(0)+\Delta w_3 f_{13} \quad (11)$$

$$W_{23}(0)=W(0)+\Delta w_3\frac{(-x_2)}{(x_3-x_2)}=W(0)+\Delta w_3 f_{23} \quad (12)$$

ここで，$k=2$回の復元で$W_{12}(0),W_{13}(0)$が復元されたとすると，$\Delta w_3=0$でなければ$W_{12}(0)=W_{13}(0)$とならない．$W_{12}(0),W_{23}(0)$が復元された場合も同様である．また，$W_{13}(0),W_{23}(0)$が復元された場合，以下となる必要がある．

$$\Delta w_3 \frac{(-x_1)}{(x_3-x_1)} = \Delta w_3 \frac{(-x_2)}{(x_3-x_2)} \qquad (13)$$

しかし，x_iは全て異なるので$\frac{(-x_1)}{(x_3-x_1)} \neq \frac{(-x_2)}{(x_3-x_2)}$であり，$\Delta w_3 = 0$以外では$W_{13}(0) = W_{23}(0)$とならない．すなわち，式(9)に対応させると式(14)～(16)となり，式(14)～(16)は$\Delta w_3 = \Delta y_i = 0 (i = 1,2,3)$以外では成り立たず，定理 1 に示すように$\Delta w_3 \neq 0$であれば，2 回の復元結果は必ず一致しないことがわかる．

$$\begin{bmatrix} 0 \\ f_{13} \end{bmatrix} [\Delta w_3] = \begin{bmatrix} \Delta y_1 \\ \Delta y_1 \end{bmatrix} \qquad (14)$$

$$\begin{bmatrix} 0 \\ f_{23} \end{bmatrix} [\Delta w_3] = \begin{bmatrix} \Delta y_2 \\ \Delta y_2 \end{bmatrix} \qquad (15)$$

$$\begin{bmatrix} f_{13} \\ f_{23} \end{bmatrix} [\Delta w_3] = \begin{bmatrix} \Delta y_3 \\ \Delta y_3 \end{bmatrix} \qquad (16)$$

また，$n = 4$とすると以下も復元できる．

$$W_{14}(0) = W(0) \qquad (17)$$

$$W_{24}(0) = W(0) \qquad (18)$$

$$W_{34}(0) = W(0) + \Delta w_3 \frac{(-x_4)}{(x_3-x_4)} \qquad (19)$$

この場合，$n \geq 2k$となるので，系 2 が言える．すなわち，$W_{12}(0) = W_{41}(0) = W_{42}(0)$となるため，$x_1, x_2, x_4$のサーバ組は正当であり，$x_3$が不正サーバと判定できる．また，系 3 より最大の復元回数は，$n = 3$の場合 ${}_3C_2$=3 回，$n = 4$の場合 ${}_4C_2$=6 回となる．

また，$k = 3, n = 4$とし，x_3, x_4を不正サーバとすると，式(20)～(23)が復元される．

$$W_{123}(0) = W(0) + \Delta w_3 \frac{(-x_1)}{(x_3-x_1)} \frac{(-x_2)}{(x_3-x_2)} = W(0) + \Delta w_3 f_{123} \qquad (20)$$

$$W_{124}(0) = W(0) + \Delta w_4 \frac{(-x_1)}{(x_4-x_1)} \frac{(-x_2)}{(x_4-x_2)} = W(0) + \Delta w_4 f_{124} \qquad (21)$$

$$W_{134}(0) = W(0) + \Delta w_3 \frac{(-x_1)}{(x_3-x_1)} \frac{(-x_4)}{(x_3-x_4)} + \Delta w_4 \frac{(-x_1)}{(x_4-x_1)} \frac{(-x_3)}{(x_4-x_3)}$$
$$= W(0) + \Delta w_3 f_{143} + \Delta w_4 f_{134} \qquad (22)$$

$$W_{234}(0) = W(0) + \Delta w_3 \frac{(-x_2)}{(x_3-x_2)} \frac{(-x_4)}{(x_3-x_4)} + \Delta w_4 \frac{(-x_2)}{(x_4-x_2)} \frac{(-x_3)}{(x_4-x_3)}$$
$$= W(0) + \Delta w_3 f_{243} + \Delta w_4 f_{234} \qquad (23)$$

3 回の復元で$W_{123}(0), W_{134}(0), W_{234}(0)$が復元されたとし，$\Delta w_3, \Delta w_4$に関連する部分を取り出すと以下となる．

$$\begin{bmatrix} f_{123} & 0 \\ f_{143} & f_{134} \\ f_{243} & f_{234} \end{bmatrix} \begin{bmatrix} \Delta w_3 \\ \Delta w_4 \end{bmatrix} = \begin{bmatrix} \Delta y_4 \\ \Delta y_4 \\ \Delta y_4 \end{bmatrix} \qquad (24)$$

しかし，式(24)から構成される 3 つの方程式は 1 次独立であるので，$\Delta w_3 = \Delta w_4 = \Delta y_4 = 0$以外では解を持たない．また，他の組み合わせが復元されても同様である．ただし，$k - 1 = 2$回の復元ではf_{zyi}で構成される行列は 2×2 の行列となり逆行列が計算されるため解をもつ．すなわち，$\Delta w_3, \Delta w_4$を調整することによって復元結果を一致させることができる．よって，復元回数はk回とする必要がある．以上より，上記定理及び系は実証された．

また，提案方式は復元結果が正しいかを検証するだけであれば，復元・検証の手順 1,2 を行うだけでよい．正しくない場合，$n - e \geq k + 1$であれば復元・検証の手順 3,4 を実行することによって全ての不正サーバを特定することができ，正しい復元結果を得ることができる．$n - e \geq k + 1$でない場合，正しいサーバを含んでいても全ての復元結果が異なるため，不正サーバを特定できない．

また，$n = k$の場合，系 1 からk個の方程式が立てられないため，不正検証ができない．一般に，Shamir 法は$n = k$のときサーバ欠損耐性がなく，$n > k$のときサーバ欠損耐性があることが知られている．それと同様に，Shamir 法は$n = k$のとき不正検証できず，$n > k$のとき不正検証が可能になると言える．

また，従来の検証方式では攻撃者が秘密情報を知っていることを前提とする CDV モデルと，知らないことを前提とする OKS モデルに分類される．それに対して提案方式は Shamir 法を実行しているだけと言えるため，分散では特別な処理は行われず，復元では全サーバから得た分散値を用いて最大 ${}_nC_k$ 回の復元を行うだけである．よって，攻撃者が秘密情報を知っていても細工をすることができず，CDV モデルと OKS モデルのどちらにも適用できると考えられる．

5.2 加法的秘密分散に関する考察

加法のみを用いて秘密情報を分散する加法的秘密分散法がある．加法的秘密分散法は$n = k$の場合に限定されるが，それを$n = k + 1$に拡張した複製秘密分散法がある．以下に加法的秘密分散法を用いた複製秘密分散法を示す．

[分散]

1. オーナは乱数$a_i (i = 1, \ldots, k + 1)$を生成して秘密情報$a$を以下のように構成する．
$$a = a_1 + a_2 + \cdots + a_k + a_{k+1}$$
2. オーナは$k + 1$台のサーバ$S_i (i = 1, \ldots, k + 1)$に$a_i, a_{i+1}$を配布する．ただし，$S_{k+1}$は$a_{k+1}, a_1$をもつ．

[復元]

1. 復元者は$k + 1$台のサーバ中，k台を選択してその分散値を得て復元を行う．ただし，S_jが選択されない場合，S_{j-1}がS_jの代わりにa_jを出力するとする．

$k + 1$台のサーバ中に$k + 1 - t\ (t \geqq 2)$台の不正サーバがある場合，正当なサーバ数はt台となる．ここでは簡単のためにS_1～S_tを正当なサーバとし，S_{t+1}～S_{k+1}を不正サーバとする．$k + 1$台中復元に用いられるのは，k台である．よって，①S_1が用いられない場合（不正サーバS_{k+1}がa_1を出力する），②S_1以外の正当なサーバが用いられない場合（正当なサーバが代わりに用いられる），③S_{t+1}が用いられない場合（正当なS_tがa_{t+1}を出力する），④S_{t+1}以外の不正サーバが用いられない場合（不正なサーバが代わりに用いられる）に分けられる．ここで，S_{k+1}が出力するa_1とS_{t+1}が出力するa_{t+1}は正しい値が出力されるとし，不正サーバは結託しており，他の不正サーバが出力する偽の分散値$a_j + \Delta a_j (j =$

$t+2,\ldots,k-1$)を知るとする．そのとき，$k+1$台のサーバの組合せを変えてk回の復元を行うと，以下の式が成り立つ．ただし，$k=2$の場合を除く．

$$\begin{bmatrix}1 & \ldots & 1\\ \vdots & \ldots & \vdots\\ 1 & \ldots & 1\end{bmatrix}\begin{bmatrix}\Delta a_{t+2}\\ \vdots\\ \Delta a_{k+1}\end{bmatrix}=\begin{bmatrix}\Delta y\\ \vdots\\ \Delta y\end{bmatrix} \qquad (25)$$

式(25)における 2 行目以降の式は 1 行目の式に対して独立ではなく従属の関係になることから，任意のΔa_jに対して式(25)は常に成立する．すなわち，定理 1 が成立しない．よって，加法的秘密分散または複製秘密分散は復元の繰り返しによって復元結果の検証はできない．

例えば，$k=3, t=2$の場合，S_3, S_4が不正サーバとなる．ここで，サーバの組合せを変えてk回の復元を行うと以下の内の３つが復元される．ただし，代わりに出力される分散値は（）で表し，不正な分散値にはダッシュを付ける．また，a_{xyz}はサーバS_x, S_y, S_zの組み合わせによる復元結果を表す．

$$a_{234}=(a_1+)a_2+a_3+a'_4$$

$$a_{341}=a_1(+a_2)+a_3+a'_4$$

$$a_{412}=a_1+a_2(+a_3)+a'_4$$

$$a_{123}=a_1+a_2+a_3(+a'_4)$$

上記においてどの３つをとっても一致しており，かつ不正な値を含むことがわかる．

ただし，$k=2$の場合，正当なサーバのみの組み合わせが存在することから，$k=2$の場合のみ不正が検出される．

よって，加法的秘密分散法を用いた複製秘密分散法では代わりの分散値を出力するサーバと出力されるサーバの 2 台を攻撃者が乗っ取ることができれば不正検出できないようにすることができる．

5.3 従来方式との比較

提案方式と従来方式との性能比較は以下のようになる．

不正成功確率：

従来方式は偽の分散値に対して秘密情報と認証子の関係が偶然に一致することが考えられるため，不正成功確率は０にならず，$1/p$となる．それに対して提案方式は$k+1$台のサーバ中，不正サーバ数が$k-1$台以下であれば不正成功確率を０にすることができ安全である．

分散時の処理：

従来方式は秘密情報の他に少なくとも認証子に関する分散処理や認証子生成に関する処理を必要とする．それに対して提案方式は秘密情報を 1 回分散するだけであり，他の処理を必要とせず効率的である．

サーバの記憶容量：

従来方式は秘密情報の分散値以外に少なくとも認証子の分散値や検証に必要な他の情報を保存する必要がある．提案方式は秘密情報の分散値のみ記憶すればよく記憶量を最小にできる．

復元結果の検証：

従来方式は秘密情報と認証子に関する復元処理の他に秘密情報と認証子が特定の関係を満たすかを検証するために乗算や多項式計算等を必要とする．提案方式は秘密情報を復元するだけで良い．ただし，提案方式はk回以上の復元を必要とするため，kが大きいとき，従来方式より効率が劣化する場合が考えられる．

不正サーバの特定：

従来方式で不正サーバまで特定できる方法は少ない．提案方式は$n-e\geq k+1$の条件が満たされれば不正サーバを特定することができる．特に，$n\geq 2k$とすれば必ず不正サーバを特定できる．

6. まとめ

本論文では，Shamir 法をそのまま用いて復元結果の検証を簡単に行える方法を示した．提案方式を用いれば，従来方式で復元結果を検証するために必要であった認証子等に対する処理や分散値を保存する必要はない．また，不正成功確率を 0 にすることができる．さらに，CDV モデルと OKS モデルのどちらにも適用できると考えられ，汎用性をもつ．

今後の課題は Shamir 法以外の秘密分散法に対する検証法を明らかにしていくことである．

参考文献

[1] A.Shamir . How to Share a Secret. Communications of the ACM,vol,22,no.11,pp.612-613,1979

[2] 高橋慧,岩村惠市. 非対称秘密分散法を用いたアプリケーションの検討. IPSJ SIG Technical Report, Vol.2013-CSEC-62.No.15

[3] Sergio Cabello, Carles Padr ´ o, Germ ´ an S ´ aez. Secret Sharing Schemes with Detection of Cheaters for a General Access Structure. Designs,Codes and Cryptography,25, 175-188, 2002

[4] Wakaha OGATA, Toshinori ARAKI. Cheating Detectable Secret Sharing Schemes for Random Bit Strings. IEICE TRANS. FUNDAMENTALS,VOL.E96-A, NO.11, NOVEMBER 2013

[5] H. Hoshino and S. Obana. Almost Optimum Secret Sharing Schemes with Cheating Detection for Random Bit Strings. Advances in Information and Computer Security, Lecture Notes in Computer Science vol. 9241, Springer Verlag, pp. 213-222, 2015

[6] T. Araki and W. Ogata. A Simple and Efficient Secret Sharing Scheme Secure against Cheating. IEICE Trans. Fundamentals, vol. E94-A, no. 6, June 2011,pp. 1338–1345, 2011.

[7] Xiaoyan Zhu. Efficient (k, n) Secret Sharing Scheme Secure Against k − 2 Cheaters. 2017 International Conference on Networking and Network Applications

[8] Qassim Al Mahmoud. A Novel Verifiable Secret Sharing with Detection and Identification of Cheaters' Group. I.J. Mathematical Sciences and Computing, 2016, 2, 1-13 Published Online April 2016 in MECS (http://www.mecs-press.net) DOI: 10.5815/ijmsc.2016.02.01

[9] Pedersen, T.P., 1992. Non-interactive and information-theoretic secure verifiable secret sharing. In: Advances Cryptology-CRYPTO'91, LNCS, vol. 576. Springer- Verlag,

[10] Carpentieri, A Perfect Threshold Secret Sharing Scheme to Identify Cheaters, Designs, Codes and Cryptography, 5, 183-187 (1995)

[11] R.J. McEliece and D.V. Sarwate. On Sharing Secretsand Reed- Solomon Codes. Communications of the ACM. Vol.24, pp.583-584, 1981.

[12] 今井秀樹. 符号理論. ISBN4-88552-090-8. 124, 158.

[13] 川島千種，吉田隆弘，松嶋智子. 多重符号化を利用した階層的な秘密分散法の検討. 信学技報. ISEC2007-76, pp.17-23, Sep. 2007.

[14] 川島千種，吉田隆弘，松嶋智子. 積符号化を利用した階層的な秘密分散法の検討. SCIS2008, Jan. 2008

[15] M. Carpentieri, A. De Santis, and U. Vaccaro, Size of shares and probability of cheating in threshold schemes, Lecture Notes in Computer Science 765 (1994), 118–125 (Eurocrypt '93 Proceedings).

[16] W. Ogata, K. Kurosawa and D. R. Stinson, "Optimum Secret Sharing Scheme Secure against Cheating," SIAM Journal on Discrete Mathematics, vol. 20, no. 1, pp. 79–95, 2006.

[17] Shuhong Gao. A new algorithm for decoding Reed-Solomon codes. Communications, information and network security, 2003, . pp 55-68