Computer Security Symposium 2017
23 - 25 October 2017

階層的秘密分散法の新しい構成法と評価

島 幸司[1]　土井 洋[1]

概要：参加者をレベルに分割し，そのレベルで分割された参加者のグループ間で秘密を共有する階層的秘密分散法が知られている．我々は秘密分散後の秘密消去の容易性，すなわち，秘密情報の削除が必須参加者のシェアの削除で保証される階層的秘密分散法に着目する．本論文では，Systematic IDA で使用される生成行列に階層のアイデアを取り入れ，任意の階層に適用可能な階層的秘密分散法を提案する．安全性の証明は別途報告するが，提案手法は 1 階層においても Chen らの秘密分散法と比較し計算効率と実装効率がよいことを示す．また，実用性を見据えて，ソフトウェア実装評価を示す．

キーワード：秘密分散法，階層的秘密分散法，IDA，ソフトウェア実装

A new construction of hierarchical secret sharing schemes and its evaluation

Koji Shima[1]　Hiroshi Doi[1]

Abstract: Hierarchical secret sharing schemes are known for how they share a secret among a group of participants partitioned into levels. We examine these schemes in terms of how easily they delete a secret after it is distributed, i.e., in cases where the reliability of data deletion depends on the deletion of each indispensable participants' share. In this paper, we apply an idea of hierarchy to the generator matrix used in a systematic information dispersal algorithm (IDA) and propose a hierarchical secret sharing scheme applicable at any level. We need to separately report the proof of perfect security, but even in a single hierarchy, our scheme is more efficient in computation and implementation than Chen et al.'s secret sharing scheme. Taking practical use into consideration, we also show our evaluation of our software implementation.

Keywords: secret sharing scheme, hierarchical secret sharing scheme, IDA, software implementation

1. はじめに

秘密情報の安全な保管は情報の盗難対策や紛失対策に見られるように情報化社会においてニーズが高い．この情報の盗難対策や紛失対策を同時に満たすような秘密情報の分散管理の方法として秘密分散法が知られている．1979 年に Blakley [1] と Shamir [2] はそれぞれ独自に (k, n) しきい値法と呼ばれる秘密分散法の概念を提案した．秘密情報を n 個のシェアに分散し，n 個のシェアの中から任意の k 個を集めれば元の秘密情報を復元でき，任意の $k-1$ 個のシェアからは元の秘密情報に関する情報が全く得られないという特徴がある．このため，シェアの一部が漏えいしても元の秘密情報は安全であり，シェアの一部が紛失しても元の秘密情報を復元できる．

一方で，参加者をレベルに分割し，そのレベルで分割された参加者のグループ間で秘密を共有する階層的秘密分散法が知られている．その中で，金庫を開けるには 3 人の従業員が必要で，少なくとも 1 人は部長といったシナリオに見られるように，最小限の高いレベルの参加者が必要とされる秘密分散法がある．このシナリオの例を $(\{1, 3\}, n)$ 階層的秘密分散法と呼ぶ．Tassa [3], [4] は導関数を導入し，Birkhoff 補間問題に注力している．

この階層的秘密分散法は秘密情報の復元に必須参加者を必要とするため，秘密消去の容易性を狙える．従来の

[1] 情報セキュリティ大学院大学
Institute of Information Security

(k,n) 秘密分散法では，秘密情報の削除が $n-k$ 個より多くのシェアの削除で保証されるが，この階層的秘密分散法では，秘密情報の削除が必須参加者のシェアの削除で保証されるからである．たとえば，実用上の想定として，緊急性によるデータ消去の保証や確実性を考えたとき，必須参加者のシェアの削除を拠り所にできる．このため，この階層的秘密分散法は情報の盗難対策や紛失対策を満たしつつ，秘密分散後の秘密消去の容易性に適した手法と言える．

1.1 秘密分散法

Shamir の (k,n) しきい値法は $k \leq n$ を満たす任意の k と n に対して実現可能であるが，秘密情報の分散および復元において，$k-1$ 次多項式を処理する必要があるため，計算コストが大きい．藤井ら [5] は排他的倫理和演算（XOR）のみを用いて秘密情報の分散および復元を行うことができる $(2,n)$ しきい値法を提案した．栗原ら [6], [7], [8] は XOR のみを用いた $(3,n)$ しきい値法と (k,n) しきい値法を提案した．また，栗原ら [9] は GF(2^n) 上の乗算コストが大きいことを踏まえ，Feng ら [10]，Blömer ら [11] の有限体の行列表現を用いて，GF(2^n) ではなく GF(2) 上の演算を実現できることを示し高速化に貢献した．Chen ら [12] は systematic IDA を通してシェアを組み立てる手法を提案した．これらの方式はすべて理想的秘密分散法である．

山本 [13] と Blakley ら [14] は符号化効率を考えたランプ型（*ramp*）秘密分散法を提案した．Krawczyk [15] は鍵ベースの暗号アルゴリズムで秘密情報を暗号化し，その暗号化データを Rabin の情報分散アルゴリズム（Information Dispersal Algorithm, IDA） [16] で分散し，鍵を秘密分散法で分散する Secret Sharing Made Short (SSMS) と呼ばれる計算量的秘密分散を提案した．栗原ら [8], [17] はランプ型秘密分散法への拡張方法を示し，高速なランプ型秘密分散法を提案した．Resch ら [18] は Rivest の All-or-nothing Transform (AONT) [19] と IDA を強化し組み合わせ，AONT-RS と呼ばれる計算量的秘密分散を提案した．五十嵐ら [20] は消失訂正符号の研究と実用化に関連し，SSMS の考えをベースに GF(2^{64}) の乗算の高速化テクニックを紹介し，高速な計算量的秘密分散を提案した．

1.2 階層的秘密分散法

Tassa [3], [4] は最小限の高いレベルの参加者が必要とされる階層的秘密分散法を提案した．導関数を用いて階層化を実現し，Birkhoff 補間を用いて秘密情報を復元する．Selçuk ら [21] は多項式を切り詰める truncated 版関数を用いて階層化を提案した．島ら [22], [23] は藤井ら [5] の手法をベースした高速な階層的秘密分散法を提案した．また，彼らは [22], [24] は Tassa の手法を標数 2 の拡大体に適用した階層的秘密分散法を提案し高速化を示した．

別の階層的な設定として，Shamir [2] はより重要な参加者にはより多くのシェアを与えることを提案した．Tassa が指摘するように，Shamir の手法は参加者の部分集合の中で表現されるそれぞれのレベルで関係づけられたしきい値の加重平均で決まるため，低いレベルの参加者の部分集合が十分に大きいときは，低いレベルの参加者のみで秘密情報を復元できてしまう課題がある．また，Simmons [25] と Brickell [26] は必要な参加者が各レベルに関連付けられるしきい値の中の最大値で決まるため，最小限の高いレベルの参加者が必要とされるシナリオは実現できない．

1.3 本研究の貢献

本論文では，階層型 IDA を導入することによる階層的秘密分散の構成法を提案する．提案手法は理想的な階層的秘密分散法であり，アクセス構造が (k,n) しきい値秘密分散法よりも複雑でありながら，そのアルゴリズムは Chen らの手法よりもシンプルである．また，Chen らの構成の安全性は攻撃者が高々 $k-1$ 個のシェアしか利用できないことに依存する．しかし，階層的な構造を考える場合，低階層の k 以上のシェアを利用できる考察が必要であり，Chen らの手法をそのまま多階層に適用できない．詳細は 4.3 節で述べる．本研究の貢献は次のように要約できる．

- 任意の階層に適用可能な階層的秘密分散法を提案する．安全性の証明は紙面の都合で別途報告する．
- IDA で使用する生成行列に階層のアイデアを取り入れ，前述の課題を解決する．階層的秘密分散のシステム構成が 1 階層のとき，この生成行列が Chen らの手法で使用される生成行列と同じ効率を満たす．
- シンプルなアルゴリズムを提供する．階層的秘密分散のシステム構成が 1 階層のとき，Chen らの手法と比較し計算効率と実装効率がよい．

2. 準備

2.1 表記法と定義

- $\oplus$ はビット単位の XOR を表す．
- $\oplus_{j=a}^{b} c_j$ は $c_a \oplus \cdots \oplus c_b$ を表す．
- $||$ はバイナリ列の連結を表す．
- $||_{j=a}^{b} c_j$ は $c_a || \cdots || c_b$ を表す．
- $H(X)$ は確率変数 X のエントロピーを表す．
- $\mathbf{v}[j]$ はベクトル $\mathbf{v}$ の j 番目の要素を表す．
- $\mathbf{v}[0] \cdots [n-1]$ は n 個の要素を持つベクトル $\mathbf{v}$ を表す．

2.2 完全秘密分散法

Beimel [27] は文献の定義 2 と定義 3 において，完全秘密分散法は次の条件を必要とすることを示している．

- [正当性] アクセス構造の中に属する権限を持つすべての集合 B は秘密に関する情報を得る．
- [完全性] アクセス構造の外に属する権限を持たないすべての集合 T は秘密に関する情報を一切得ない．

言い換えれば，ある与えられた確率分布の中での秘密情報に関する確率変数を S，ある与えられた確率分布の中での権限を持つすべての集合 B のシェアに関する確率変数を S_B，ある与えられた確率分布の中での権限を持たないすべての集合 T のシェアに関する確率変数を S_T としたとき，完全秘密分散法は次の条件を必要とする．

- [正当性] $H(S|S_B) = 0$.
- [完全性] $H(S|S_T) = H(S)$.

2.3 理想的秘密分散法

Blundo ら [28], [29]，栗原ら [6], [7], [8] の文献から，n 人の参加者集合を $\mathcal{P} = \{P_1, \cdots, P_n\}$，秘密情報の集合を $\mathcal{S}$，参加者 P_i のシェアとして可能性のある集合を $\mathcal{W}_i$ とする秘密分散法が与えられたとき，その情報率を $\rho = \dfrac{H(S)}{\max\limits_{P_i \in \mathcal{P}} H(W_i)}$ と定義する．ここで，S と W_i はそれぞれ $s \in \mathcal{S}$, $w_i \in \mathcal{W}_i$ によって誘起される確率変数とする．$\mathcal{S}$ と $\mathcal{W}_i$ がどちらも一様な確率分布に従うとき，

$$\rho = \frac{\log_2 |\mathcal{S}|}{\max\limits_{P_i \in \mathcal{P}} \log_2 |\mathcal{W}_i|}$$

を測定できることが知られており，$\rho = 1$ を満たす完全秘密分散法を理想的秘密分散法という．すなわち，各シェアのビット長は秘密情報のビット長よりは小さくできないが，これらのビット長が等しい場合，理想的秘密分散法となる．また，Tassa [4] が定義 1.1 で述べるように，この情報率を階層的秘密分散法に適用できる．

2.4 ランプ型秘密分散法

本稿では，(k, L, n) ランプ型秘密分散法 [13] に関して，Jackson ら [30] の文献を参照し，文献 [12] の定義を用いる．

定義 2.1. $(t_0, t_1; n)$ ランプ型秘密分散法は秘密情報を n 個のシェアに分散し，任意の t_1 個を集めれば元の秘密情報を復元でき，t_0 個以下のシェアからは元の秘密情報に関する情報が全く得られない．

この定義より，$(k-1, k; n)$ ランプ型秘密分散法は Shamir の (k, n) しきい値法と同じである．$(0, k, n)$ ランプ型秘密分散法は元の秘密情報に関する情報が全く得られないという完全性の強制力がなく，任意の k 個で復元できる正当性のみを持つ．$(t_0, t_1; n)$ ランプ型秘密分散法が線形ならば，t_0 個のシェアが集まった後は続くシェアごとに $\frac{1}{t_1 - t_0}$ ビットの秘密情報が得られる．

2.5 情報分散アルゴリズム

(k, n) IDA [16] は長さ $|F|$ のファイル F を長さ $|F|/k$ の n 個の断片に分割し，任意の k 個の断片で復元できる．分散後のデータサイズは $n|F|/k$ である．1 に近い n/k を選べるため，スペース効率が良い．IDA は完全性は要求されず正当性のみであるから [12], [15]，(k, n) IDA は $(0, k; n)$ ランプ型秘密分散法と等価である [12].

2.6 誤り訂正符号と消失訂正符号

誤り訂正符号はデータ転送や保存においてエラーが発生したとき，元の情報を復元できること保証するために冗長な情報に符号化する方法である．符号語が k 個の情報記号と符号化により付加された $n-k$ 個の検査記号が明確に区別できる符号を systematic と呼ぶ．消失訂正符号は誤り訂正符号であり，(k, n) IDA と等価であるとみなせる [12].

3. 関連研究

Tassa [3], [4] は次のアクセス構造を定義する．

定義 3.1. $\mathbf{k} = \{k_i\}_{i=0}^{m}$, $0 < k_0 < \cdots < k_m$ とする．k は最大しきい値 k_m を用いて $k = k_m$ とする．最小限の高いレベルの参加者が必要とされる $(\mathbf{k}, n)$ 階層的秘密分散法は次のアクセス構造 Γ で与えられる．

$$\Gamma = \left\{ \mathcal{V} \subset \mathcal{U} : \left| \mathcal{V} \cap \left(\bigcup_{j=0}^{i} \mathcal{U}_j \right) \right| \geq k_i, \forall i \in \{0, 1, \cdots, m\} \right\}$$

$\mathcal{U}$ は n 人の参加者集合とし，すべての $0 \leq i < j \leq m$ について $\mathcal{U}_i \cap \mathcal{U}_j = \emptyset$ となる $\mathcal{U} = \bigcup_{i=0}^{m} \mathcal{U}_i$ で構成される．アクセス構造 Γ を満たすように各参加者 $u \in \mathcal{U}$ にシェアを割り当てる．$\mathbf{k} = \{1, 3\}$ とすれば，2 階層で構成され，$\mathcal{U}_0$ の必須参加者は 1 人以上，復元に 3 人以上の協力が必要な $(\{1, 3\}, n)$ 階層的秘密分散法を意味する．

3.1 Tassa の階層的秘密分散法

定義 3.1 を満たす理想的な階層的秘密分散法である．Shamir の (k, n) しきい値法と同じように，大きな有限体上の $k-1$ 次多項式 $p(x)$ の定数を秘密情報とする．各参加者 $u \in \mathcal{U}$ は同じ u と表現される有限体上の識別子が与えられ，自身の階層の位置に依存する何らかの j 階導関数値 $p^{(j)}(u)$ をシェアとして受け取る．より重要な参加者はより小さい i 番目の参加者集合 $\mathcal{U}_i$ に所属し，より低い j 階導関数を用いたシェアを得る．導関数を適切に選ぶことで，階層的秘密分散法の要求するアクセス構造を満たし，権限を持つ部分集合が協力して秘密情報の復元を試みる．

3.2 Systematic IDA と理想的秘密分散法

Resch ら [18] は AONT-RS の実現に Rabin の IDA に変更を加え，systematic な消失訂正符号を用いる．Systematic IDA の採用により，最初の k 個の符号語は符号化する必要がなく性能を改善させる．

Chen ら [12] は systematic IDA を用いて理想的秘密分散法を構成する．IDA はランプ型秘密分散法であるが，秘密情報と乱数の XOR 値，その使用した乱数をそれぞれ systematic IDA に通し，その出力に巡回シフトを施してシェアを組み立て理想的秘密分散法を実現する．

3.2.1 Systematic IDA

(k,n) systematic IDA は $Share^{\mathrm{IDA}}$ と $Recover^{\mathrm{IDA}}$ の 2 つのアルゴリズムから構成される．

$Share^{\mathrm{IDA}}$ はデータ M を入力として n 人の参加者に分散するための符号語を出力する．M は k 個の要素を持つ列ベクトル $\mathbf{M}$ に展開され，各要素は L ビットの長さを持つ $\mathrm{GF}(2^L)$ の要素として扱われる．生成行列または分散行列と呼ばれる行列 $\mathbf{G}=[g_{(i,j)}]_{i=1,j=1}^{n\ \ k}$ は公開される $n\times k$ 行列であり，最初の k 行は $k\times k$ 単位行列である．残りの $n-k$ 行は $\mathbf{G}$ を構成する n 行の任意の k 行が線形独立になるように決める．n 個の要素を持つ列ベクトル $\mathbf{C}=\mathbf{G}\cdot\mathbf{M}$ を符号語とする出力を得る．$\mathbf{G}$ の最初の k 行が単位行列であるから，$\mathbf{M}$ の各要素 $\mathbf{M}[i]\in\mathrm{GF}(2^L)$，$\mathbf{C}$ の各要素 $\mathbf{C}[i]\in\mathrm{GF}(2^L)$ を用いて次式を得る．

$$\mathbf{C}=\mathbf{G}\cdot\begin{pmatrix}\mathbf{M}[0]\\ \vdots\\ \mathbf{M}[k-1]\end{pmatrix}=\begin{pmatrix}\mathbf{M}[0]\\ \vdots\\ \mathbf{M}[k-1]\\ \mathbf{C}[k]\\ \vdots\\ \mathbf{C}[n-1]\end{pmatrix}$$

生成行列 $\mathbf{G}$ は Vandermonde 行列に行列の基本変形を施した行列にする．Plank ら [31] は $g_{(i,j)}=(i-1)^{j-1}$ とする $n\times k$ 行列の Vandermonde 行列を準備し，その行列に行列の基本変形を施し，最初の k 行を単位行列にすることを述べている．行列の基本変形は行列の階数を変えないからである．なお，$g_{(i,j)}=x^{j-1}$ とするとき，すべての x が異なれば Vandermonde 行列の行列式は 0 ではない．

$Recover^{\mathrm{IDA}}$ は k 個の残存する符号語の要素 $\mathbf{C}'$ を入力としてデータ M を復元する．$\mathbf{C}'$ は k 個の要素を持つ列ベクトルである．k 個の残存する要素に対応する $\mathbf{G}$ の行から新しい $k\times k$ 行列 $\mathbf{G}'$ を生成する．逆行列 $(\mathbf{G}')^{-1}$ を用いて，$\mathbf{M}=(\mathbf{G}')^{-1}\cdot\mathbf{C}'$ によりデータ M を得る．

3.3 Chen らの秘密分散法

Chen ら [12] の (k,n) 秘密分散法を示す．参加者は P_x $(x=0,\cdots,n-1)$ とする．Systematic IDA を用いるため，生成行列 $\mathbf{G}$ は公開されている．

3.3.1 分散アルゴリズム

$F=\mathrm{GF}(2^L)$ 上の (k,n) 秘密分散法において，秘密情報 $s\in\{0,1\}^\lambda,\lambda=L\cdot k$ が与えられる．s が k 個の L ビットからなる要素で構成される $\mathbf{s}\in F^k$ と展開できる．秘密情報が λ ビットに満たない場合は λ ビットにパディングする．

アルゴリズムを表 1 に示す．Step 1 で乱数 $r_1,\cdots,r_{k-1}\in\{0,1\}^\lambda$ を生成し，$\mathbf{r}_1,\cdots,\mathbf{r}_{k-1}\in F^k$ と展開する．Step 2 で s とすべての乱数 r_i の XOR から $s'\in\{0,1\}^\lambda$ を生成し，$\mathbf{r}_0\in F^k$ と展開する．Step 3 でそれぞれの $\mathbf{r}_i$ を $Share^{\mathrm{IDA}}$ に渡す．この結果 n 個の要素を持つ列ベクトル $\mathbf{R}_0,\cdots\mathbf{R}_{k-1}\in F^n$ が得られる．Step 4, 5 で参加者 P_x はシェア $w_x\in\{0,1\}^\lambda$ を受け取る．

Step 3, 4 を詳しく述べる．Step 3 は次のような $k\times n$ 行列 $\mathbf{M}$ を生成すると表現できる．

$$\mathbf{M}=\begin{pmatrix}\mathbf{R}_0^{\mathrm{T}}\\ \mathbf{R}_1^{\mathrm{T}}\\ \vdots\\ \mathbf{R}_{k-1}^{\mathrm{T}}\end{pmatrix}=\begin{pmatrix}\mathbf{R}_0[0][1]\cdots[n-1]\\ \mathbf{R}_1[0][1]\cdots[n-1]\\ \vdots\\ \mathbf{R}_{k-1}[0][1]\cdots[n-1]\end{pmatrix}$$

行列 $\mathbf{M}$ の j $(j=1,2,\cdots,k)$ 行目の要素を左に $j-1$ 回巡回シフトすると，次の行列 $\mathbf{M}'$ を表現できる．

$$\mathbf{M}'=\begin{pmatrix}\mathbf{R}_0[0] & [1] & \cdots & [n-2] & [n-1]\\ \mathbf{R}_1[1] & [2] & \cdots & [n-1] & [0]\\ \vdots & \vdots & & \vdots & \vdots\\ \mathbf{R}_{k-1}[k-1] & [k] & \cdots & [k-3] & [k-2]\end{pmatrix}$$

参加者 P_x は行列 $\mathbf{M}'$ の $x+1$ 列目の要素を連結したシェア w_x を秘密裏に得る．

表 1 (k,n) しきい値分散アルゴリズム
Table 1 (k,n) threshold distribution algorithm.

Input: $s\in\{0,1\}^\lambda$
Output: $(w_0,\cdots w_{n-1})$
1: for $i\leftarrow 1$ to $k-1$:
$\quad\mathbf{r}_i\leftarrow r_i\xleftarrow{\$}\{0,1\}^\lambda$
2: $\mathbf{r}_0\leftarrow s'\leftarrow s\oplus\{\oplus_{j=1}^{k-1}r_j\}$
3: for $i\leftarrow 0$ to $k-1$:
$\quad\mathbf{R}_i\leftarrow Share^{\mathrm{IDA}}(\mathbf{r}_i,\mathbf{G})$
4: for $i\leftarrow 0$ to $n-1$:
$\quad w_i\leftarrow\|_{j=0}^{k-1}\mathbf{R}_j[i+j(\mathrm{mod}\ n)]$
5: return $(w_0,\cdots w_{n-1})$

3.3.2 復元アルゴリズム

アルゴリズムを表 2 に示す．秘密情報の復元に協力する参加者 P_i $(i=t_0,\cdots,t_{k-1})$ のシェアを入力とする．Step 1 でシェア w_i を k 個の要素に展開する．$k\times n$ 行列 $\mathbf{M}$ で考えると，各行の n 個の要素の中で k 個の要素が展開される．シェア生成時に要素が巡回シフトされているから，各行の展開される k 個の要素のインデックスは異なる．これは各行で使用する $Recover^{\mathrm{IDA}}$ の $\mathbf{G}'$ が異なることを示唆する．Step 2, 3 で Step 1 の各行で展開された k 個の要素を列ベクトル $\mathbf{R}'_0,\cdots\mathbf{R}'_{k-1}$ と表現し，$Recover^{\mathrm{IDA}}$ に渡す．この結果 s' と $r_1,\cdots,r_{k-1}$ を復元できる．Step 4, 5 でこれらの復元された値の XOR で秘密情報 s を復元できる．

3.3.3 安全性

行列 $\mathbf{M}$ が次の要件を満たし，完全性を満たすことを示している．実際，行列 $\mathbf{M}$ は巡回シフトを用いて構成されるから要件を満たす．

表 2　(k,n) しきい値復元アルゴリズム
Table 2　(k,n) threshold recovery algorithm.

Input: $(w_{t_0},\cdots w_{t_{k-1}})$
Output: s

1: for $i \leftarrow 0$ to $k-1$:
　　$||_{j=0}^{k-1}\mathbf{R}_j[t_i+j(\mathrm{mod}\ n)] \leftarrow w_{t_i}$
2: for $i \leftarrow 0$ to $k-1$:
　　$r_i \leftarrow \mathbf{r}_i \leftarrow Recover^{\mathrm{IDA}}(\mathbf{R}'_i,\mathbf{G}'_i)$
3: $s' \leftarrow r_0$
4: $s \leftarrow s' \oplus \{\oplus_{j=1}^{k-1} r_j\}$
5: return s

要件 3.1. 復元に協力する参加者が多くとも $k-1$ 人のとき，その参加者集合は行列 $\mathbf{M}$ のすべての列について，k 個の要素の中の少なくとも *1* つの要素に関する情報が得られない．

4. 提案方式

定義 3.1 を満たす $F=\mathrm{GF}(2^L)$ 上の $(\mathbf{k},n)$ 階層的秘密分散法を提案する．$\mathrm{GF}(p)$ も適用可能である．秘密情報 $s\in\{0,1\}^{\lambda},\lambda=L\cdot k$ が与えられる．s が k 個の L ビットからなる要素で構成される $\mathbf{s}\in F^k$ と展開できる．秘密情報が λ ビットに満たない場合は λ ビットにパディングする．次の項目を満たす生成行列 $\mathbf{G}$ を使用する．

- アクセス構造を満たす参加者集合のみが復元できるように階層化に対応する．
- $n\times k$ 行列である．

$\mathbf{G}$ は公開され，システムで一意に決まるテーブルと考えてよいから，シェアのみで秘密情報の復元ができる．また，$\mathbf{G}$ は systematic IDA のために行列の基本変形を施すが，行列の基本変形を施す前の生成行列で議論できる．この $\mathbf{G}$ を階層型生成行列と呼ぶことにする．また，この $\mathbf{G}$ を使った IDA を階層型 IDA と呼ぶことにする．

4.1 参加者の識別子と実在しない参加者

参加者 P_x $(x=0,\cdots,n-1)\in\mathcal{U}$ は識別子 $x\in F$ を持つとする．一般性を失うことなく，参加者 P_x は次の階層に属すると考えてよい．$0<l_0<\cdots<l_m=n$ とする．

$$
\begin{aligned}
&P_0,\cdots,P_{l_0-1}\in\mathcal{U}_0,\\
&P_{l_0},\cdots,P_{l_1-1}\in\mathcal{U}_1,\\
&\ \ \vdots\\
&P_{l_{m-1}},\cdots,P_{l_m-1}\in\mathcal{U}_m,
\end{aligned}
$$

実在しない参加者 $0\in\mathcal{U}_0$ を割り当てる．このあと述べる u_x は常に 0 が割り当てられるとする．

4.2 階層型生成行列

$n\times k$ 生成行列 $\mathbf{G}$ を考える．P_x は $\mathbf{G}$ を構成する $x+1$ 行目に対応する．$\mathbf{G}=[g_{(x,j)}]_{x=0,\ j=1}^{n-1\ \ k}$ と表すことにする．

$\mathbf{G}$ に階層化を導入する．アクセス構造を満たさない k 個のシェアに対応する $\mathbf{G}$ の行から生成される $k\times k$ 行列 $\mathbf{G}'$ が $\det(\mathbf{G}')=0$ となるように $\mathbf{G}$ を生成する．$P_x\in\mathcal{U}_i$ $(i=0,\cdots,m)$ とする $\mathbf{G}$ は

$$
g_{(x,j)}=\begin{cases} u_x^{j-1-k_{i-1}} & (j>k_{i-1})\\ 0 & (j\le k_{i-1})\end{cases}
$$

で与えられる．$g_{(x,j)}\in F, k_{-1}=0$ とする．

例として，$(\{2,3,5\},n)$ 階層的秘密分散法の $\mathbf{G}$ を示す．

$$
\mathbf{G}=\begin{pmatrix}
1 & u_0 & u_0^2 & u_0^3 & u_0^4\\
\vdots & \vdots & \vdots & \vdots & \vdots\\
1 & u_{l_0-1} & u_{l_0-1}^2 & u_{l_0-1}^3 & u_{l_0-1}^4\\
0 & 0 & 1 & u_{l_0} & u_{l_0}^2\\
\vdots & \vdots & \vdots & \vdots & \vdots\\
0 & 0 & 1 & u_{l_1-1} & u_{l_1-1}^2\\
0 & 0 & 0 & 1 & u_{l_1}\\
\vdots & \vdots & \vdots & \vdots & \vdots\\
0 & 0 & 0 & 1 & u_{n-1}
\end{pmatrix}
$$

$\mathbf{G}$ の u_x は F 上の 0 を除く 2^L-1 個が与えうるが，与え方によっては，本来復元できる参加者集合にもかかわらず，復元に対応する $\mathbf{G}'$ が $\det(\mathbf{G}')=0$ になり復元できないことがある．これは Tassa [3], [4] が述べる Birkhoff 補間問題と同様の議論を展開できる．

たとえば，$(\{1,3\},5)$ 階層的秘密分散法の $\mathbf{G}$ として $\mathbf{G}_1$ を $u_x=1,2,3,4,5$ で与えると，$\mathbf{G}_1'$ の一つが $\det(\mathbf{G}_{p_1}')=0$ となり，アクセス構造を満たせない．$u_x=1,2,4,5,6$ とする $\mathbf{G}_2$ を与える必要がある．このようなアクセス構造を満たす $\mathbf{G}$ を一つ求めることができればよい．

$$
\mathbf{G}_1=\begin{pmatrix}1&1&1\\1&2&4\\0&1&3\\0&1&4\\0&1&5\end{pmatrix},\mathbf{G}_1'=\begin{pmatrix}1&1&1\\1&2&4\\0&1&3\end{pmatrix},
$$

$$
\mathbf{G}_2=\begin{pmatrix}1&1&1\\1&2&4\\0&1&4\\0&1&5\\0&1&6\end{pmatrix}
$$

4.3 IDA に階層化を適用する課題

(k,n) IDA は任意の k 個の要素が与えられると，分散元のメッセージだけではなく n 個の要素すべてが得られる．生成行列 $\mathbf{G}$ が公開されているため，残りの n 個の要素も

表 3　提案手法の行列 $\mathbf{M}$
Table 3　Matrix $\mathbf{M}$ for our scheme.

$$\mathbf{M} = \left(\begin{array}{ccc|c|ccc} \overbrace{\mathbf{R}_0[0] & \cdots & [l_0-1]}^{\mathcal{U}_0} & \cdots & \overbrace{[l_{m-1}] & \cdots & [n-1]}^{\mathcal{U}_m} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \mathbf{R}_{k-1}[0] & \cdots & [l_0-1] & \cdots & [l_{m-1}] & \cdots & [n-1] \end{array} \right)$$

計算可能だからである．また，$(\mathbf{k}, n)$ 階層型 IDA も同様に $k_m - k_{j-1}$ $(j = 0, \cdots, m)$ 個の要素が与えられた階層 j の要素はすべて復元されうる．この結果，分散元のメッセージ k 個の $k_m - k_{j-1}$ 個が復元される．また，要件 3.1 を満たす巡回シフトは階層型 IDA では効果がない．この考察の下，2.2 節の正当性と完全性を満たすアルゴリズムを提示する．

4.4　分散アルゴリズム

ディーラは参加者 P_x にシェア $w_x \in \{0,1\}^\lambda$ を秘密裏に配布する．アルゴリズムを表 4 に示す．表 1 との差分は下線で示す．Step 1 から Step 5 を通して表 3 の $n \times k$ 行列 $\mathbf{M}$ が表現される．参加者 P_x は行列 $\mathbf{M}$ の $x+1$ 列目の要素を連結したシェア w_x を得る．

表 4　提案手法の分散アルゴリズム
Table 4　Our proposed distribution algorithm.

Input: $s \in \{0,1\}^\lambda$
Output: $(w_0, \cdots w_{n-1})$
1: for $i \leftarrow 1$ to $k-1$:
$\quad \mathbf{r}_i \leftarrow r_i \xleftarrow{\$} \{0,1\}^\lambda$
2: $\mathbf{r}_0 \leftarrow s' \leftarrow s \oplus \{\oplus_{j=1}^{k-1} r_j\}$
3: $(\mathbf{r}'_0 \ \cdots \ \mathbf{r}'_{k-1}) \leftarrow (\mathbf{r}_0 \ \cdots \ \mathbf{r}_{k-1})^{\mathrm{T}}$
4: for $i \leftarrow 0$ to $k-1$:
$\quad \mathbf{R}_i \leftarrow Share^{\mathrm{IDA}}(\underline{\mathbf{r}'_i}, \mathbf{G})$
5: for $i \leftarrow 0$ to $n-1$:
$\quad w_i \leftarrow \|_{j=0}^{k-1} \mathbf{R}_j[\underline{i}]$
6: return $(w_0, \cdots w_{n-1})$

Step 3 は $\mathbf{r}_0, \cdots, \mathbf{r}_{k-1}$ を転置して構成される行列について，各列ベクトルを $\mathbf{r}'_0, \cdots, \mathbf{r}'_{k-1}$ と読み替える．

$$(\mathbf{r}_0 \ \cdots \ \mathbf{r}_{k-1})^{\mathrm{T}} = \begin{pmatrix} \mathbf{r}_0[0] & \cdots & \mathbf{r}_0[k-1] \\ \vdots & \ddots & \vdots \\ \mathbf{r}_{k-1}[0] & \cdots & \mathbf{r}_{k-1}[k-1] \end{pmatrix} = (\mathbf{r}'_0 \ \cdots \ \mathbf{r}'_{k-1})$$

4.5　復元アルゴリズム

アルゴリズムを表 5 に示す．秘密情報の復元に協力する参加者 P_i $(i = t_0, \cdots, t_{k-1})$ のシェアを入力とする．表 2 との差分は下線で示す．巡回シフトがないため，各行の $Recover^{\mathrm{IDA}}$ で使用する $\mathbf{G}'$ は共通にできる．Step 2 で Step 1 の各行で展開された k 個の要素を列ベクトル $\mathbf{R}'_0, \cdots \mathbf{R}'_{k-1}$ と表現し，$Recover^{\mathrm{IDA}}$ に渡す．すべての $\mathbf{r}'_i$ が復元されると，Step 3, 4 で s' と $r_1, \cdots, r_{k-1}$ を知ることができる．Step 5, 6 でこれらの復元された値の XOR で秘密情報 s を復元できる．

表 5　提案手法の復元アルゴリズム
Table 5　Our proposed recovery algorithm.

Input: $(w_{t_0}, \cdots w_{t_{k-1}})$
Output: s
1: for $i \leftarrow 0$ to $k-1$:
$\quad \|_{j=0}^{k-1} \mathbf{R}_j[\underline{t_i}] \leftarrow w_{t_i}$
2: for $i \leftarrow 0$ to $k-1$:
$\quad \underline{\mathbf{r}'_i} \leftarrow Recover^{\mathrm{IDA}}(\mathbf{R}'_i, \underline{\mathbf{G}'})$
3: $r_i \leftarrow \underline{\mathbf{r}_i \leftarrow \mathbf{r}'_i}$
4: $s' \leftarrow r_0$
5: $s \leftarrow s' \oplus \{\oplus_{j=1}^{k-1} r_j\}$
6: return s

4.6　正当性と完全性

アクセス構造 Γ の下で $\mathbf{M}$ の各行について復元されるべき分散元のメッセージ k 個の要素が復元される必要がある．j 行目の復元されるべきデータは $k-1$ 個の乱数 $\mathbf{r}'_{j-1}[1], \cdots, \mathbf{r}'_{j-1}[k-1] \in F$，これらの乱数と $\mathbf{s}[j-1] \in F$ の XOR で生成される $\mathbf{r}'_{j-1}[0]$ である．このため，$k-1$ 個の要素からは秘密情報 $\mathbf{s}[j-1]$ に関する情報を全く得ることができない．

また，Systematic IDA を用いると行列 $\mathbf{M}$ の最初の k 列は分散元のメッセージそのもので構成される．したがって，行列 $\mathbf{M}$ の最初の k 列に対応する参加者のシェアが集まると，$Share^{\mathrm{IDA}}$ を使用せずとも秘密情報を復元できるが，行列 $\mathbf{M}$ の 1 列目は必ず最上位階層 $\mathcal{U}_0$ の参加者のシェアに割り当てられるから，必須参加者を含まない参加者だけで秘密情報を復元することは依然としてできないと言える．正当性と完全性の証明の詳細は別途報告する．

4.7 計算コストの比較

表 6 は復元の計算コストを示す．一般的に秘密情報のサイズは 1MB のファイルサイズのように L ビットを超える．事前計算はその復元で一度だけ計算すれば十分なものである．$k \times k$ 行列式は LU 分解，すなわち三角行列を用いて $\mathcal{O}(k^3)$ である．三角行列の変換には数回の除算が発生する．$k \times k$ 逆行列はガウスの消去法を用いると $\mathcal{O}(k^3)$ と数回の除算が発生する．$k \times k$ 行列の乗算は $\mathcal{O}(k^3)$ である．提案手法は除算の回数が少なく，最適化なしにおいて高速化される．

表 6　復元の計算コスト
Table 6　Computational costs for recovery.

Tassa の手法

事前計算	$k \times k$ 行列式 1 回
1 回の復元	$k \times k$ 行列式 1 回，除算 1 回

提案手法

事前計算	$k \times k$ 逆行列 1 回
1 回の復元	$k \times k$ 行列の乗算 1 回，XOR $k-1$ 回

Tassa の手法は GF(p) 上の Birkhoff 補間を用いて秘密情報を復元するときの計算コストである．GF(p) と GF(2^L) の違いに依存する計算コストは除外する．このため，提案手法も GF(p) 上の計算コストは表 6 と同じと考えてよい．

5. ソフトウェア実装

$\mathbf{k} = \{k_i\}_{i=0}^{m}$ の $(\mathbf{k}, n)$ 階層的秘密分散法について，いくつかの $\mathbf{k}$ を与えて実装を行い，888,710 バイトのファイルを復元する実験を行った．測定環境は表 7 の汎用 PC の環境 1 台を準備した．

表 7　測定環境
Table 7　Test environment.

CPU	Intel ® Celeron ® Processor G1820
	2.70GHz × 2, 2MB cache
RAM	3.6GB
OS	CentOS 7 Linux 3.10.0-229.20.1.el7.x86_64
言語	C 言語
コンパイラ	gcc 4.8.3 (-O3 -flto -DNDEBUG)

GF(2^L) の演算について，加算は排他的論理和，乗算は Russian Peasant Multiplication アルゴリズム，除算は $x^{-1} = x^{2^L-2}$，シフト演算は左に 1 シフトする演算を用いる．本実験では，GF(2^L) の演算は GF(2^8) の使用に限定し，かつ GF(2^8) の乗除算を事前計算するルックアップテーブル方式を用いた．具体的には，乗算の結果を 2^{16} バイト分の配列に，除算の結果を 2^{16} バイト分の配列にそれぞれ格納し，乗除算の計算が発生すればその配列を参照する．GF(2^8) の演算で用いた原始多項式は $x^8+x^4+x^3+x^2+1$ である．実験結果を表 8 に示す．参加者数は $(|\mathcal{U}_0|, \cdots, |\mathcal{U}_m|)$ で表現し，参加者数の合計は $|\mathcal{U}_0| + \cdots + |\mathcal{U}_m|$ である．

表 8　実験結果
Table 8　Results of our experiments.

階層 $\mathbf{k}$	参加者数	復元速度（Mbps）
{1,3}	$(2, 3)$	857
{2,4}	$(3, 4)$	562
{2,3,5}	$(3, 3, 3)$	373
{2,4,6,10}	$(2, 3, 6, 4)$	108
{3,7,11,14,17}	$(3, 4, 5, 4, 4)$	34.9

5.1 生成行列の決定における課題

$\mathbf{G}$ は n 行の任意の k 行が線形独立になるように決める必要がある．このような u_x の与え方は試行錯誤による．多くのケースですぐに $\mathbf{G}$ を決定できるが，たとえば，GF(2^8) 上の $(\{2, 4, 6, 10\}, 15)$ 階層的秘密分散において，$|\mathcal{U}_0| = 3, |\mathcal{U}_1| = 3, |\mathcal{U}_2| = 3, |\mathcal{U}_3| = 6$ で $\mathbf{G}$ を見つけることができていない．

6. おわりに

高速化と秘密消去の容易性に着目し，IDA で使用される生成行列に階層化のアイデアを取り入れ $(\mathbf{k}, n)$ 階層的秘密分散法を提案した．提案手法は理想的秘密分散法である．

Chen らの手法をそのまま多階層に適用できない分析から，階層化に必要な要素の考察と解決のための改良を経て階層化の実現を達成した．また，階層的秘密分散のシステム構成が 1 階層のとき，その生成行列は Chen らの手法で使用される生成行列と同じ効率を満たし，提案手法は Chen らの手法と比較し計算効率と実装効率がよいことを示した．

任意の階層で実験できる実装の下，$\mathbf{k} = \{1, 3\}$ において，850Mbps 程度の復元速度を確認できた．

参考文献

[1] Blakley, G.R.: Safeguarding cryptographic keys, *AFIPS*, Vol.48, pp.313–317 (1979).
[2] Shamir, A.: How to share a secret, *Commun. ACM*, Vol.22, No.11, pp.612–613 (1979).
[3] Tassa, T.: Hierarchical Threshold Secret Sharing, *TCC 2004*, LNCS 2951, pp.473–490 (2004).
[4] Tassa, T.: Hierarchical Threshold Secret Sharing, *Journal of Cryptology*, Vol.20, No.2, pp.237–264 (2007).
[5] 藤井吉弘, 多田美奈子, 保坂範和, 栃窪孝也, 加藤岳久：高速な $(2, n)$ 閾値法の構成法とシステムへの応用，*CSS2005*, 8C-2，pp.631–636 (2005).
[6] Kurihara, J., Kiyomoto, S., Fukushima, K. and Tanaka, T.: A Fast $(3, n)$-Threshold Secret Sharing Scheme Using Exclusive-OR Operations, *IEICE Trans. Fundamentals*, Vol.E91-A, No.1, pp.127–138 (2008).
[7] Kurihara, J., Kiyomoto, S., Fukushima, K. and Tanaka, T.: On a Fast (k, n)-Threshold Secret Sharing Scheme, *IEICE Trans. Fundamentals*, Vol.E91-A, No.9, pp.2365–2378 (2008).

[8] Kurihara, J., Kiyomoto, S., Fukushima, K. and Tanaka, T.: A New (k, n)-Threshold Secret Sharing Scheme and Its Extension, *ISC 2008*, LNCS 5222, pp.455–470 (2008).
[9] Kurihara, J. and Uyematsu, T.: A Novel Realization of Threshold Schemes over Binary Field Extensions, *IEICE Trans. Fundamentals*, Vol.E94-A, No.6, pp.1375–1380 (2011).
[10] Feng, G.-L., Deng, R.-H. and Bao, F.: Packet-loss resilient coding scheme with only XOR operations, *IEE Proc. Communications*, Vol.151, No.4 (2004).
[11] Blömer, J., Kalfane, M., Karp, R., Karpinski, M., Luby, M. and Zuckerman, D.: An XOR-Based Erasure-Resilient Coding Scheme, *ICSI Technical Report TR-95-048* (1995).
[12] Chen, L., Laing, T.M. and Martin K.M.: Efficient, XOR-Based, Ideal (t,n) threshold Schemes, *CANS 2016*, LNCS 10052, pp.467–483 (2016).
[13] 山本博資：(k, L, n) しきい値秘密分散システム，電子通信学会論文誌，Vol.J68-A，No.9，pp.945–952 (1985).
[14] Blakley, G.R. and Meadows C.: Security of Ramp Schemes, *Advances in Cryptology - CRYPTO '84*, LNCS 196, pp.242–268 (1985).
[15] Krawczyk, H.: Secret Sharing Made Short, *Advances in Cryptology - CRYPTO '93*, LNCS 773, pp.136–146 (1993).
[16] Rabin, M.O.: Efficient dispersal of information for security, load balancing, and fault tolerance, *Journal of the ACM*, Vol.36, No.2, pp.335–348 (1989).
[17] Kurihara, J., Kiyomoto, S., Fukushima, K. and Tanaka, T.: A Fast (k, L, n)-Threshold Ramp Secret Sharing Scheme, *IEICE Trans. Fundamentals*, Vol.E92-A, No.8, pp.1808–1821 (2009).
[18] Resch, J.K. and Plank, J.S.: AONT-RS: blending security and performance in dispersed storage systems, In *9th USENIX Conference on File and Storage Technologie*, pp.191–202 (2011).
[19] Rivest, R.L.: All-or-nothing encryption and the package transform, *FSE 1997*, LNCS 1267, pp.210–218 (1997).
[20] 五十嵐大, 露崎浩太, 川原祐人: SHSS: オブジェクトストレージ向けの超高速秘密分散ライブラリ, 情報処理学会, 第 70 回 CSEC 研究会 (2015).
[21] Selçuk, A.A., Kaşkaloğlu, K. and Özbudak, F.: On Hierarchical Threshold Secret Sharing, *IACR Cryptology ePrint Archive 2009*, 450 (2009).
[22] 島幸司，土井洋：階層的秘密分散法の高速化に関する研究，*CSS2015*，3C4-5，pp.1327–1334 (2015).
[23] Shima K. and Doi H.: $(\{1, 3\}, n)$ hierarchical secret sharing scheme based on XOR operations for a small number of indispensable participants, *AsiaJCIS 2016*, pp.108–114 (2016).
[24] Shima, K. and Doi, H.: A Hierarchical Secret Sharing Scheme over Finite Fields of Characteristic 2, *Journal of Information Processing*, Vol.25, pp.(undecided as of August) (2017).
[25] Simmons, G.J.: How to (really) share a secret, *Advances in Cryptology - CRYPTO '88*, LNCS 403, pp.390–448 (1990).
[26] Brickell, E.F.: Some ideal secret sharing schemes, *Advances in Cryptology - EUROCRYPT '89*, LNCS 434, pp.468–475 (1990).
[27] Beimel, A.: Secret-Sharing Schemes, A Survey, *IWCC 2011*, LNCS 6639, pp.11–46 (2011).
[28] Blundo, C., De Santis, A., Gargano, L. and Vaccaro, U.: On the information rate of secret sharing schemes, *TCS*, Vol.154, pp.283–306 (1996).
[29] Blundo, C., De Santis, A., Gargano, L. and Vaccaro, U.: On the Information Rate of Secret Sharing Schemes, *Advances in Cryptology - CRYPTO '92*, LNCS 740, pp.149–169 (1993).
[30] Jackson, W.-A. and Martin, K.M.: A Combinatorial Interpretation of Ramp Schemes, *Australasian Journal of Combinatorics*, Vol.14, pp.51–60 (1996).
[31] Plank, J.S. and Ding, Y.: Note: Correction to the 1997 tutorial on Reed-Solomon coding, *Software - Practice & Experience*, Vol.35, No.2, pp.189–194 (2005).