Vol.2016-DPS-166 No.4
Vol.2016-CSEC-72 No.4
2016/3/3

標数 2 の有限体上の$(\{1,k\},n)$階層的秘密分散法の研究

島幸司[†1]　土井洋[†1]

概要：秘密分散法は 1979 年に Blakley と Shamir によりそれぞれ独自に提案された．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．著者らは秘密消去の容易性を狙いに階層的秘密分散法に着目し，CSS2015 で高速な$(\{1,3\},n)$階層的秘密分散法として，排他的論理和のみを用いるものと，導関数と Birkhoff 補間を使った Tassa のアイデアを継承しつつ，標数 2 の有限体上への適応が可能なものを提案した．本稿では，後者に対する改良とその実装評価の結果を報告する．また，より一般的な$(\{1,k\},n)$階層的秘密分散法への拡張法も提案する．実装評価は CPU: Intel Celeron G1820 2.70GHz，RAM: 3.6GB の 1 台のマシン環境で測定し，秘密情報の復元に約 970Mbps の処理で実現できることを確認した．

キーワード：秘密分散法，階層的秘密分散法，導関数，Birkhoff 補間，標数 2 の有限体

A study on $(\{1,k\},n)$ hierarchical secret sharing schemes over finite fields of characteristic 2

KOJI SHIMA[†1]　HIROSHI DOI[†1]

Abstract: Blakley and Shamir independently introduced the concept of (k,n) threshold secret sharing scheme in 1979. On the other hand, hierarchical secret sharing scheme is known in the way that the secret is shared among a group of participants that is partitioned into levels. The authors look at hierarchical secret sharing schemes in the purpose of the ease of deleting the secret, and proposed fast $(\{1,3\},n)$ hierarchical secret sharing schemes in CSS2015 by the method of only using XOR and by the method that inherits Tassa's idea of using derivatives and Birkhoff interpolation, and that can apply to finite fields of characteristic 2. In this paper, we report the results of the improvement and implementation evaluation for the latter, and propose the more general extended method of a $(\{1,k\},n)$ hierarchical secret sharing scheme. Our implementation system on a PC with Intel Celeron G1820 2.70GHz and 3.6GB RAM can recover the secret in the processing of around 970Mbps.

Keywords: Secret sharing scheme, hierarchical secret sharing scheme, derivative, Birkhoff interpolation, finite fields of characteristic 2

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を分散管理するための方法として秘密分散法が知られている．1979 年に Blakley と Shamir 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[1][2]．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，$k-1$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，かつ，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．Tassa は導関数を導入し，Birkhoff 補間問題に注力している[3][4]．

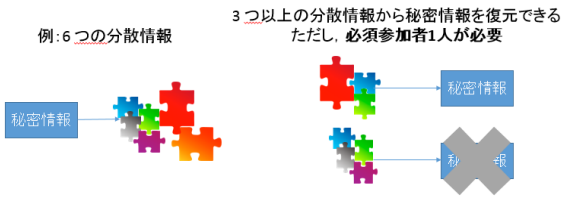

図 1　階層的秘密分散法
Figure 1　Hierarchical secret sharing scheme.

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．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データ消去の確実性について，必須参加者のシェアの削除を拠り所にできることである．著者らは文献[5]において，先行研究を継承した高速な階層的秘密分散法の実現性を提案した．最近は，階層的秘密分散法に関連して，文献[6][7]の研究がなされている．

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

Vol.2016-DPS-166 No.4
Vol.2016-CSEC-72 No.4
2016/3/3

1.1 秘密分散法の高速化

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．2005 年に藤井らは排他的倫理和演算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[8]. 2008 年に栗原らは排他的論理和演算のみを用いた$(3,n)$しきい値法を提案し，nがあまり大きくなければ，Shamir のしきい値法と比較して非常に効率の良い計算コストが得られた[9]．さらに，同年に栗原らは排他的論理和演算のみを用いた(k,n)しきい値法を提案し，nがあまり大きくなければ，Shamir のしきい値法と比較して，より効率の良い計算コストが得られることを，$(k,n)=(4,5)$を例に示している[10].

1.2 本研究の貢献

秘密分散法は様々な視点で研究されている．たとえば，分散後のデータの総量が元データの数倍になるストレージ面の課題への対策，ランプ型秘密分散法，プロアクティブ秘密分散法がある．事業継続計画対策の一環としても利用されるクラウドサービスに目を向けると，秘密情報の分散管理が求められるが，同時に反応速度も求められる．一方では，運用リスクや情報管理に対する懸念からオンプレミスの選択もあるように，秘密分散法が利用される領域は固定的ではなく，求められる要素が様々という事情がある．こうした秘密分散法とその周辺の課題の中で，ビッグデータに向けた性能向上の必要性が求められることからも，本研究では高速化に着目している．

また，秘密消去の容易性を狙うシナリオから，有益と考えられるレベルの高い権限者が最低でも 1 名を満たす，高速かつメモリ使用量が少ない階層的秘密分散法について検討する．具体的には，導関数と Birkhoff 補間を使った Tassa のアイデアを継承し，標数 2 の有限体上への適用を考慮した$(\{1,k\},n)$階層的秘密分散法を提案する．なお，藤井らの排他的論理和の手法への適用は著者らの文献[5]がある．

2. 準備

2.1 標数 2 の有限体の演算

$\mathrm{GF}(2^4)$を例に述べる. 原始多項式x^4+x+1の根 α から生成すると，$\alpha^{2^4-1}=\alpha^{15}=1$である．加算と減算は排他的論理和であり，乗算は$\alpha^i\alpha^j=\alpha^{(i+j) \bmod 15}$，除算は$\alpha^i$の逆元$\alpha^{-i}=\alpha^{(15-i) \bmod 15}$の積で演算する. たとえば, $\alpha^4+\alpha+1=0$から$\alpha^4=\alpha+1$であり，指数表現α^4はベクトル表現(0011)で表され，10 進数表現 3 で表せる．

2.2 原始多項式

使用する原始多項式は表 1 のとおりである．

表 1　原始多項式
Table 1　Primitive polynomials.

拡大体	原始多項式
$\mathrm{GF}(2^8)$	$x^8+x^4+x^3+x+1$
$\mathrm{GF}(2^{16})$	$x^{16}+x^{12}+x^3+x+1$
$\mathrm{GF}(2^{32})$	$x^{32}+x^2+x+1$
$\mathrm{GF}(2^{64})$	$x^{64}+x^4+x^3+x+1$
$\mathrm{GF}(2^{128})$	$x^{128}+x^7+x^2+x+1$
$\mathrm{GF}(2^{256})$	$x^{256}+x^{10}+x^5+x^2+1$

2.3 行列式の性質

本研究で必要なn次行列式の性質は次のとおりである．行を列に言い換えても同じことが言える．

- 二つの行の交換は行列式の値の符合だけ変わる．
- ある行のc倍をほかの行に足しても，行列式の値は変わらない．
- 転置行列の行列式の値はもとの行列の行列式の値と変わらない．
- 二つの行が一致するならば，行列式の値は0である．
- 行列式のある行をc倍すると，行列式はc倍になる．

Vandermonde の行列式の値は次のとおりである．

$$\begin{vmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^n \end{vmatrix} = \prod_{i>j}(x_i - x_j)$$

3. 関連研究

Tassa 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次のように定義する．

$\boldsymbol{U}$をn人の参加者集合とするとき，階層レベルiの参加者集合$\boldsymbol{U}_i$と表現したm階層を考える．

$$\boldsymbol{U}=\bigcup_{i=0}^{m}\boldsymbol{U}_i,\ \boldsymbol{U}_i\cap\boldsymbol{U}_j=\emptyset,\ 0\leq i<j\leq m$$

$\boldsymbol{k}=\{k_i\}_{i=0}^{m}, 0<k_0<\cdots<k_m$とするとき，$(\boldsymbol{k},n)$階層的秘密分散法は次のアクセス構造を満たすように各参加者$u\in\boldsymbol{U}$にシェアを割り当てる．

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

このため，$(\{1,3\},n)$階層的秘密分散法は$\boldsymbol{k}=\{1,3\}$の 2 階層で構成され，$\boldsymbol{U}_0$の必須参加者は 1 人以上，復元に$\boldsymbol{U}_0\cup\boldsymbol{U}_1$

から 3 人以上の協力が必要な階層的秘密分散法を意味する.

一方で，Tassa は文献[3][4]で Simmons とその後 Brickell が Tassa とは別の階層的な設定を検討したことを述べている. Simmons が解決するシナリオは任意の 2 人の副社長か，代替手段として，任意の 3 人の上級窓口係が電子資金決済できるもので, Tassa の階層的秘密分散法におけるアクセス構造のしきい値条件を論理積から論理和で考えている．しかしながら，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.

3.1 Tassa の階層的秘密分散法の実現方式

Tassa の実現方式は低いレベルの参加者だけでは秘密情報は復元できず，かつ理想的秘密分散法である．Shamir の(k,n)しきい値法のように，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し，最大しきい値を用いて，$k=k_m$とする．各参加者$u\in\boldsymbol{u}$は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$\boldsymbol{u}_i$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.

一方，Shamir は文献[2]で，階層的秘密分散法の実現方法として，より重要な参加者にはより多くのシェアを与えることで達成することを提案している．しかし，Tassa が文献[3][4]で指摘するように, 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.

3.2 多項式補間

秘密情報の復元で連立方程式を解く代わりに多項式補間を利用すると，計算量削減により高速化に貢献する．しかし，シェアに導関数値が含まれると，Lagrange 補間や Newton 補間では秘密情報を復元できない．Hermite 補間は導関数値を含めた補間手法として知られているが，シェア$f'(x_1)$と同時に$f(x_1)$の値もシェアとして与えられる必要があるため，シェア配布の観点で制限が入る．Birkhoff 補間はこの制限を解消しうる.

(1) Birkhoff 補間

$\mathcal{G}=\{g_0,g_1,\dots,g_N\}$を線形独立な[a, b]で$n$回連続微分可能なℝ上の関数列とし，線形結合$P=\sum_{k=0}^{N}a_k g_k, a_k\in\mathbb{R}$を$\mathcal{G}$の多項式と呼ぶ．次式の$m\times(n+1)$補間行列

$$E=\left[e_{i,j}\right]_{i=1,j=0}^{m\quad n},\qquad m\geq 1,\ n\geq 0,$$

は要素$e_{i,j}$が 0 または 1 であり，かつ，$\sum e_{i,j}=N+1$である．ただし，Eは空行を含まない．すなわち，すべての$j=0,\dots n$に対して，$e_{i,j}=0$となる行iは含まないとする.

今，[a, b]でm個の異なる点の集合$X=\{x_1,\dots,x_m\}, x_1<\dots<x_m$が与えられているとする. Birkhoff 補間問題[11][12]とは，$\langle E,X,\mathcal{G}\rangle$の組と与えられたデータ$c_{i,j}$により，次の条件を満たす多項式$p$を見つけることである.

$$p^{(j)}(x_i)=c_{i,j},\quad e_{i,j}=1 \tag{1}$$

式(1)は$N+1$個の等式からなる．次の行列式が 0 以外のときに限り，$\langle E,X,\mathcal{G}\rangle$の組が$c_{i,j}$の各集合に対して唯一の解を持つ.

$$D(E,X,\mathcal{G})=\det[g_0^{(j)}(x_i),\dots,g_N^{(j)}(x_i);e_{i,j}=1] \tag{2}$$

式(2)は$(N+1)\times(N+1)$行列式の一つの行だけを示している．すなわち，$e_{i,j}=1$の(i,j)の組に対応した行を示している．また，行の並びは$i<i'$または$i=i',j<j'$のときに(i,j)が(i',j')より先に並ぶ辞書的な順番とする.

$(N+1)\times(N+1)$行列を$A(E,X,\mathcal{G})$と表現すると，$D(E,X,\mathcal{G})$は次式で表せる.

$$D(E,X,\mathcal{G})=\det\big(A(E,X,\mathcal{G})\big)=|A(E,X,\mathcal{G})|$$

データ$c_{i,j}$が$c_{i,j}=p^{(j)}(x_i)$で与えられたとき，補間多項式は次式で与えられる.

$$p(x)=\sum_{j=0}^{N}\frac{D\big(E,X,\mathcal{G}_j\big)}{D(E,X,\mathcal{G})}\cdot g_j(x)$$

$\mathcal{G}_j$は$\mathcal{G}$のg_jをpに置き換えた関数の集合で，たとえば，$\mathcal{G}_1=\{g_0,p,g_2,\dots,g_N\}$である.

(2) Birkhoff 補間の具体例

$g_0(x)=1,g_1(x)=x,g_2(x)=x^2$において，すなわち，$\mathcal{G}=\{1,x,x^2\}$において，次のように$X$と$E$が与えられたとする.

$$X=\{1,2,3\},\qquad E=\begin{pmatrix}1&0\\1&0\\0&1\end{pmatrix}$$

言い換えれば，次の値を満たす多項式$p(x)=\sum_{j=0}^{2}a_j x^j$を探すことである.

$$p(1) = c_{1,0}, p(2) = c_{2,0}, p'(3) = c_{3,1}$$

具体的に$p(1) = 15, p(2) = 29, p'(3) = 23$が与えられたとすると，次の多項式$p(x)$を得る．

$$D(E,X,\mathcal{G}) = \begin{vmatrix} g_0(x_1) & g_1(x_1) & g_2(x_1) \\ g_0(x_2) & g_1(x_2) & g_2(x_2) \\ g_0'(x_3) & g_1'(x_3) & g_2'(x_3) \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 0 & 1 & 6 \end{vmatrix} = 3,$$

$$D(E,X,\mathcal{G}_0) = \begin{vmatrix} p(x_1) & g_1(x_1) & g_2(x_1) \\ p(x_2) & g_1(x_2) & g_2(x_2) \\ p'(x_3) & g_1'(x_3) & g_2'(x_3) \end{vmatrix} = \begin{vmatrix} 15 & 1 & 1 \\ 29 & 2 & 4 \\ 23 & 1 & 6 \end{vmatrix} = 21,$$

$$D(E,X,\mathcal{G}_1) = \begin{vmatrix} g_0(x_1) & p(x_1) & g_2(x_1) \\ g_0(x_2) & p(x_2) & g_2(x_2) \\ g_0'(x_3) & p'(x_3) & g_2'(x_3) \end{vmatrix} = \begin{vmatrix} 1 & 15 & 1 \\ 1 & 29 & 4 \\ 0 & 23 & 6 \end{vmatrix} = 15,$$

$$D(E,X,\mathcal{G}_2) = \begin{vmatrix} g_0(x_1) & g_1(x_1) & p(x_1) \\ g_0(x_2) & g_1(x_2) & p(x_2) \\ g_0'(x_3) & g_1'(x_3) & p'(x_3) \end{vmatrix} = \begin{vmatrix} 1 & 1 & 15 \\ 1 & 2 & 29 \\ 0 & 1 & 23 \end{vmatrix} = 9,$$

$$p(x) = \sum_{j=0}^{2} \frac{D(E,X,\mathcal{G}_j)}{D(E,X,\mathcal{G})} \cdot g_j(x) = 7 + 5x + 3x^2$$

十分大きい素数pにおいて，$f(x) = 3x^2 + 5x + 7 \bmod p$でシェアが分散される階層的秘密分散法を考える．秘密情報はシェア$f(1), f(2), f'(3)$が集まると，次式で計算できる．

$$s = f(0) = \frac{D(E,X,\mathcal{G}_0)}{D(E,X,\mathcal{G})} = \frac{21}{3} = 7$$

3.3 著者らの階層的秘密分散法の実現方式

文献[5]で，導関数を用いた$(\{1,3\},n)$階層的秘密分散法を示した．標数 2 の拡大体上で微分すると，次数が偶数の項の結果は消えてしまうので，単に拡大体を適用するだけではなく工夫が必要である．そこで，3 次多項式をランダムに選択し，n人の参加者に各シェアが配布されると同時に，$u \in \boldsymbol{U}_0$の参加者に相当する一つのシェアはグローバルに共有されるようにした．

その後の研究で，行列式の計算を効率よく行うために，グローバルに共有されるシェアを$f(x_1 = 1)$で配布し，計算量を削減できることを確認した．その結果，必須参加者 1 人を含む 3 人が復元に協力するとき，必須参加者 2 人を含む 3 人が復元に協力するときのそれぞれについて，秘密情報の復元に必要な演算回数（単位は回数）は表 2 のとおりである．なお，大きなファイルを分散・復元するときは，小さなサイズに分割して分散・復元するが（4.3 節参照），二回目以降の計算量は参加者の識別子に関する初回の演算結果を再利用できることから小さくなる．

表 2　復元の演算回数

Table 2　The number of recovery operations.

	必須参加者 1 人			必須参加者 2 人		
	乗算	加算	除算	乗算	加算	除算
初回	9	8	1	13	12	1
二回目以降	4	4	1	6	5	1

しかしながら，二つの課題があることを認識している．一つは 4.1 節で後述するように，$(\{1,k\},n)$階層的秘密分散法への一般化ができないことである．もう一つは上述したように，グローバルに共有されるシェアが存在するため，ストレージ面で優位性がないことが挙げられる．

4. 提案方式

権限レベルの高い参加者を実現するために，n人の参加者集合$\boldsymbol{U}$を必須参加者のレベルとそれ以外のレベルに分け，$\boldsymbol{k} = \{k_i\}_{i=0}^{1}$の$(\boldsymbol{k},n)$階層的秘密分散法をモデルにする．その構成方法として，標数 2 の有限体上への適用を考慮した$(\{1,k\},n)$階層的秘密分散法を提案する．すなわち，$\boldsymbol{k} = \{1,k\}$の 2 階層で構成され，$\boldsymbol{U}_0$の必須参加者は 1 人以上，復元に$\boldsymbol{U}_0 \cup \boldsymbol{U}_1$から k人以上の協力が必要な階層的秘密分散法である．アクセス構造は次のように定義される．

$$\boldsymbol{U} = \bigcup_{i=0}^{1} \boldsymbol{U}_i, \boldsymbol{U}_0 \cap \boldsymbol{U}_1 = \emptyset,$$
$$\Gamma = \left\{ \boldsymbol{V} \subset \boldsymbol{U} : \left| \boldsymbol{V} \cap \left(\bigcup_{j=0}^{i} \boldsymbol{U}_j \right) \right| \geq k_i, \forall i \in \{0,1\} \right\},$$
$$k_0 = 1, k_1 = k \qquad (3)$$

$|\boldsymbol{U}_0| \geq 2$もアクセス構造を満たすが，特に述べない限り，代表として，$|\boldsymbol{U}_0| = 1, |\boldsymbol{U}_1| = n - 1, n \geq 3$を考えるとする．

4.1 $(\{1,k\},n)$階層的秘密分散法への一般化の考察

3.3 節で示した手法をそのまま$(\{1,k\},n)$階層的秘密分散法に一般化することを考察する．

k次多項式fをランダムに選択し，$\mathcal{G} = \{g_0, g_1, \ldots, g_k\}, k \geq 3$において，$g_0(x) = 1$, $g_1(x) = x, \ldots, g_k(x) = x^k$とすると，

$$g_0'(x) = 0, g_1'(x) = 1, g_2'(x) = 0, g_3'(x) = x^2, \ldots,$$
$$g_k'(x) = \begin{cases} 0 \ (k\text{は偶数}) \\ x^{k-1} \ (k\text{は奇数}) \end{cases},$$

である．必須参加者 1 人を含むk人が復元に協力するとき，必須参加者のシェアを$f(x_2) = c_2$，グローバルに共有されるシェアを$f(1) = c_1$，残りの参加者のシェアを$f'(x_3) = c_3, \ldots, f'(x_{k+1}) = c_{k+1}$とすると，$k$が偶数$(k \geq 4)$のときは，

$$D(E,X,\mathcal{G}_0)=\begin{vmatrix} c_1 & 1 & 1 & 1 & 1 & 1 & 1 \\ c_2 & x_2 & x_2^2 & x_2^3 & \cdots & x_2^{k-1} & x_2^k \\ c_3 & 1 & 0 & x_3^2 & \cdots & x_3^{k-2} & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ c_{k+1} & 1 & 0 & x_{k+1}^2 & \cdots & x_{k+1}^{k-2} & 0 \end{vmatrix},$$

であるから，$D(E,X,\mathcal{G}_0)$はc_1,c_2が演算結果に反映されない．なぜならば，c_1やc_2に対する余因子をA_{11},A_{21}とすると，

$$A_{11}=(-1)^{1+1}\begin{vmatrix} x_2 & x_2^2 & x_2^3 & x_2^4 & \cdots & x_2^{k-1} & x_2^k \\ 1 & 0 & x_3^2 & 0 & \cdots & x_3^{k-2} & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & 0 & x_{k+1}^2 & 0 & \cdots & x_{k+1}^{k-2} & 0 \end{vmatrix}$$

$$=\frac{x_2^2}{x_2^4}\begin{vmatrix} x_2 & x_2^4 & x_2^3 & x_2^4 & \cdots & x_2^{k-1} & x_2^k \\ 1 & 0 & x_3^2 & 0 & \cdots & x_3^{k-2} & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & 0 & x_{k+1}^2 & 0 & \cdots & x_{k+1}^{k-2} & 0 \end{vmatrix}=0,$$

$$A_{21}=(-1)^{2+1}\begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & x_3^2 & 0 & \cdots & x_3^{k-2} & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & 0 & x_{k+1}^2 & 0 & \cdots & x_{k+1}^{k-2} & 0 \end{vmatrix}=0,$$

だからである．すなわち，A_{11},A_{21}両方とも 2 列目と 4 列目が一致するため，行列式の性質により行列式の値は 0 である．したがって，$D(E,X,\mathcal{G}_0)$ の演算結果にはc_1が反映されない上に，必須参加者のシェア$f(x_2)=c_2$も反映されないことを意味する．kが奇数（$k\geq 5$）のときも同様にこの議論が成立する．これが 3.3 節で示した課題の一つである$(\{1,k\},n)$階層的秘密分散法への一般化ができない理由である．

4.2 $(\{1,k\},n)$階層的秘密分散法の提案

4.1 節で示した課題と，3.3 節で示した標数 2 の拡大体上で微分すると，次数が偶数の項の結果は消えてしまう課題を踏まえ，$k-1$個の変数を持つ多項式fをランダムに選択し，$\mathcal{G}=\{g_0,g_1,\dots,g_k\}$において，

$$g_0(x)=1,\ g_1(x)=x,\ g_2(x)=x^3,\dots,g_{k-1}(x)=x^{2(k-2)+1},$$

と設定し，秘密情報の復元に Birkhoff 補間を適用する．ここで，$k-1$次多項式をランダムに選択するわけではないことに注意する．すると，導関数は次のとおりである．

$$g_0'(x)=0,g_1'(x)=1,g_2'(x)=x^2,\dots,g_{k-1}'(x)=x^{2(k-2)},$$

(3) **分散アルゴリズム**

n人の参加者$P_{i,j}\in\boldsymbol{u}_j,1\leq i\leq|\boldsymbol{u}_j|,j\in\{0,1\}$に$x_{i,j}\neq 0$とするシェア$f^{(j)}(x_{i,j})$を秘密裏に送る．

(4) **復元アルゴリズム**

アクセス構造(3)を満たすk人が復元に協力する．たとえば，必須参加者 1 人を含むk人が復元に協力するときは，$\boldsymbol{u}_0$から 1 人，$\boldsymbol{u}_1$から$k-1$人が必要である．

必須参加者 1 人を含むk人が秘密情報sの復元に協力し，$f(x_1)=c_1,f'(x_2)=c_2,f'(x_3)=c_3,\dots,f'(x_k)=c_k$とすると，

$$D(E,X,\mathcal{G}_0)=\begin{vmatrix} c_1 & x_1 & x_1^3 & \cdots & x_1^{2(k-2)+1} \\ c_2 & 1 & x_2^2 & \cdots & x_2^{2(k-2)} \\ \vdots & \vdots & \vdots & & \vdots \\ c_k & 1 & x_k^2 & \cdots & x_k^{2(k-2)} \end{vmatrix},$$

$$D(E,X,\mathcal{G})=\begin{vmatrix} 1 & x_1 & x_1^3 & \cdots & x_1^{2(k-2)+1} \\ 0 & 1 & x_2^2 & \cdots & x_2^{2(k-2)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 1 & x_k^2 & \cdots & x_k^{2(k-2)} \end{vmatrix},$$

$$s=f(0)=\frac{D(E_A,X,\mathcal{G}_0)}{D(E_A,X,\mathcal{G})},$$

と秘密情報sを復元でき，復元には確かに必須参加者のシェアが反映される．

4.3 分割されたデータの分散方法

GF(2^l)上の演算を考える．たとえば，1MB のデータを分散・復元するときは，lビットに分割してそれぞれを分散・復元する．具体的には，lビットに分割されたそれぞれの断片に対してランダムな多項式を生成し，シェアを作成する．多項式の生成で使用した乱数はシェア配布後に削除できる．

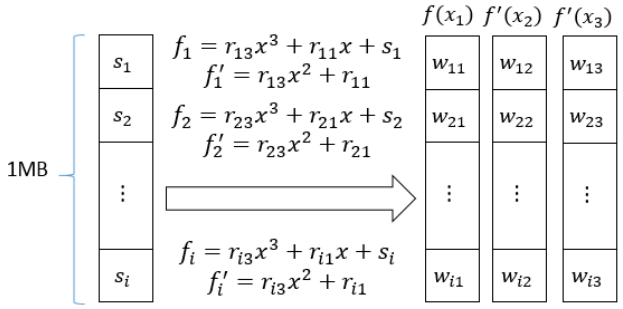

図 2　1MB データの分散

Figure 2　Distribution of the data size of 1MB.

4.4 $(\{1,3\},n)$階層的秘密分散法の計算効率

$k=3$について計算効率を述べる．2 変数を持つ多項式$f(x)=a_2x^3+a_1x+s\in$ GF(2^l)をランダムに選択する．$\mathcal{G}=\{g_0=1,g_1=x,g_2=x^3\}$として，Birkhoff 補間を適用する．必須参加者 1 人を含む 3 人が復元に協力するときの補間行列をE_A，必須参加者 2 人を含む 3 人が復元に協力するときの補間行列をE_Bとする．

$$E_A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}, \qquad E_B = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

(5) **必須参加者 1 人を含む 3 人が復元に協力する**

補間行列E_Aと$X = \{x_1, x_2, x_3\}$ が与えられるから，次式のように計算される．

$$D(E, X, \mathcal{G}_0) = \begin{vmatrix} c_1 & x_1 & x_1^3 \\ c_2 & 1 & x_2^2 \\ c_3 & 1 & x_3^2 \end{vmatrix}$$

$$= c_1(x_2^2 + x_3^2) + c_2 x_1(x_1^2 + x_3^2) + c_3 x_1(x_1^2 + x_2^2) \qquad (4)$$

$D(E, X, \mathcal{G})$は$D(E, X, \mathcal{G}_0)$を$c_1 = 1, c_2 = 0, c_3 = 0$として結果を得ることができる．

$$D(E, X, \mathcal{G}) = x_2^2 + x_3^2$$

x_1^2, x_2^2, x_3^2にそれぞれ乗算 1 回の合計 3 回である．$D(E, X, \mathcal{G}_0)$は乗算 5 回，加算 5 回の演算が必要である．$D(E, X, \mathcal{G})$は乗算 0 回，加算は$D(E, X, \mathcal{G}_0)$の計算過程で得られた結果を再利用すると 0 回である．したがって，乗算 8 回，加算 5 回，秘密情報sの取得に係る除算 1 回である．

さらに，lビットに分割されたデータの復元の二回目以降は，x_2^2, x_3^2, x_4^2などの演算結果を再利用できことから，c_1, c_2, c_3に係る演算を行えばよく，秘密情報の復元に必要な計算量は乗算 3 回，加算 2 回と秘密情報sの取得に係る除算 1 回である．なお，式(4)を次のように式変形すれば，

$$D(E, X, \mathcal{G}_0) = c_1(x_2^2 + x_3^2) + \Big(c_2(x_1^2 + x_3^2) + c_3(x_1^2 + x_2^2)\Big)x_1,$$

乗算 7 回，加算 5 回であるが，lビットに分割されたデータの復元を考慮すると，式(4)の計算量が少ない．また，$x_2 \neq x_3$から$D(E, X, \mathcal{G})$は 0 ではないため，常に復元可能と言える．

(6) **必須参加者 2 人を含む 3 人が復元に協力する**

補間行列E_Bと$X = \{x_1, x_2, x_3\}$ が与えられるから，次式のように計算される．

$$D(E, X, \mathcal{G}_0) = \begin{vmatrix} c_1 & x_1 & x_1^3 \\ c_2 & x_2 & x_2^3 \\ c_3 & 1 & x_3^2 \end{vmatrix} \qquad (5)$$

$$= c_1 x_2(x_2^2 + x_3^2) + c_2 x_1(x_1^2 + x_3^2) + c_3 x_1 x_2(x_1^2 + x_2^2)$$

$D(E, X, \mathcal{G})$は$D(E, X, \mathcal{G}_0)$を$c_1 = 1, c_2 = 1, c_3 = 0$として結果を得ることができる．

$$D(E, X, \mathcal{G}) = x_2(x_2^2 + x_3^2) + x_1(x_1^2 + x_3^2)$$

x_1^2, x_2^2, x_3^2にそれぞれ乗算 1 回の合計 3 回である．$D(E, X, \mathcal{G}_0)$は乗算 7 回，加算 5 回の演算が必要である．$D(E, X, \mathcal{G})$は$D(E, X, \mathcal{G}_0)$の計算過程で得られた結果を再利用すると，乗算 0 回，加算 1 回である．したがって，乗算 10 回，加算 6 回，秘密情報sの取得に係る除算 1 回である．

さらに，lビットに分割されたデータの復元の二回目以降は，x_2^2, x_3^2, x_4^2などの演算結果を再利用できことから，c_1, c_2, c_3に係る演算を行えばよく，秘密情報の復元に必要な計算量は乗算 3 回，加算 2 回と秘密情報sの取得に係る除算 1 回である．

秘密情報の復元に必要な演算回数（単位は回数）をまとめたものを表 3 に示す．

表 3　復元の演算回数

Table 3　The number of recovery operations.

	必須参加者 1 人			必須参加者 2 人		
	乗算	加算	除算	乗算	加算	除算
初回	8	5	1	10	6	1
二回目以降	3	2	1	3	2	1

5. 実装評価

$k = 3$とする$(\{1,3\}, n)$階層的秘密分散法について，拡大体に係る演算も含めて実装を行い実測した．なお，本実装に先立ち，拡大体の演算を検証するために NTL ライブラリ[13]を使って実装し，分散・復元の調査と確認を行った．

測定環境は表 4 の汎用 PC の環境 1 台を準備した．

表 4　測定環境

Table 4　Test environment.

CPU	Intel® Celeron® CPU G1820 @ 2.70GHz×2
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)

性能に関連する gcc オプションを測定環境に示している．GF(2^l)演算について，l=8, 16, 32 でそれぞれlビットレジスタ長演算，l=64, 128, 256 でそれぞれ 64 ビットレジスタ演算を用いている．また，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 シフトする演算である．分散・復元に使用したファイルサイズは 888710 バイトである．

表 5 は必須参加者 1 人を含む 3 人が復元に協力するときのデータ復元の演算回数（単位は回数）である．演算回数は乗算 8 回，加算 5 回，除算 1 回であるから，実装したアルゴリズムによって，除算で使用される乗算や，乗算で使

用される加算とシフト演算が増えることがわかる．

表 5　lビットデータ復元の演算回数

Table 5　The number of recovery operations in l-bit data.

	l=8	l=16	l=32	l=64	l=128	l=256
加算	160	468	1593	6881	30078	120178
乗算	22	38	70	134	262	518
除算	1	1	1	1	1	1
シフト	176	608	2240	8576	33536	132608
コピー	41	65	113	209	401	785

そこで，$\mathrm{GF}(2^8)$のときは，事前に乗算の結果を2^{16}バイト分の配列に，同様に除算の結果を2^{16}バイト分の配列にそれぞれ保持し，その配列を演算で参照するときの実測も行った（$l=8$ テーブルと表記する）．

lビットに分割されたデータの復元の二回目以降は，必須参加者が 1 人のときも 2 人のときも乗算 3 回，加算 2 回と秘密情報の取得に係る除算 1 回であるから，$l=8$ テーブルを使用した表 6 の結果のように，必須参加者数による復元時間に優位な差はないと言ってよい．

表 6　888710 バイト復元の演算回数

Table 6　The number of recovery operations in 888710 bytes.

	必須参加者 1 人	必須参加者 2 人
加算	1777423	1777424
乗算	2666135	2666137
除算	888710	888710
シフト	0	0
コピー	2666136	2666137

表 7 は復元時間と速度の結果である．

表 7　復元時間と速度

Table 7　Time of recovery operations and speed.

	$l=8$	$l=16$	$l=32$
時間（秒）	0.169084	0.339862	0.889126
速度（Mbps）	40.1	20.0	7.63

	$l=64$	$l=128$	$l=256$
時間（秒）	1.756913	14.293181	32.646211
速度（Mbps）	3.86	0.47	0.21

	$l=8$ テーブル
時間（秒）	0.006978
速度（Mbps）	971.7

6. おわりに

導関数を用いた階層的秘密分散法に着目し，標数 2 の有限体上への適用を考慮した$(\{1,k\},n)$階層的秘密分散法を提案した．本研究の手法はストレージ面や速度においても優位であり，$(\{1,3\},n)$階層的秘密分散法の効率に関して，汎用 PC を用いた実装評価により，復元に 970Mbps 程度の処理で実現できることを確認した．安全性に関してフォーマルな証明を与えることは今後の課題である．

謝辞　本研究の一部は JSPS 科研費 25330161 の助成を受けたものである．

参考文献

[1] G. R. Blakley: Safeguarding cryptographic keys, Proceedings of the National Computer Conference 48, pp.313–317 (1979).
[2] Adi Shamir: How to share a secret, Communications of the ACM 22 (11), pp.612–613 (1979).
[3] Tamir Tassa: Hierarchical Threshold Secret Sharing, TCC 2004, LNCS 2951, pp. 473–490 (2004).
[4] Tamir Tassa: Hierarchical Threshold Secret Sharing, Journal of Cryptology 20 (2), pp. 237–264 (2007).
[5] 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).
[6] 柯陳毓弢, 穴田啓晃, 川本淳平, モロゾフ キリル, 櫻井幸一: 複数プロバイダに亘る分散ストレージのためのグループ横断秘密分散法, SCIS2016, 3A1-3 (2016).
[7] 尾崎寛之, 櫻井幸一: 秘密分散に関するもう一つの安全性問題 --不公正暗号システム・再訪--, SCIS2016, 3A1-5 (2016).
[8] 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，CSS, 8C-2 (2005).
[9] Jun KURIHARA, Shinsaku KIYOMOTO, Kazuhide FUKUSHIMA, and Toshiaki TANAKA, Members: A Fast (3,n)-Threshold Secret Sharing Scheme Using Exclusive-OR Operations, IEICE TRANS. FUNDAMENTALS, VOL.E91-A, NO.1 JANUARY (2008).
[10] Jun KURIHARA, Shinsaku KIYOMOTO, Kazuhide FUKUSHIMA, and Toshiaki TANAKA, Members: On a Fast (k, n)-Threshold Secret Sharing Scheme, IEICE TRANS. FUNDAMENTALS, VOL.E91-A, NO.9 SEPTEMBER (2008).
[11] G. G. Lorentz, K. Jetter, S. D. Riemenschneider: Birkhoff Interpolation, Encyclopedia of Mathematics and its Applications 19 (1983, 1984).
[12] G. G. Lorentz and K. L. Zeller: Birkhoff Interpolation, SIAM Journal on Numerical Analysis Vol. 8, No. 1, pp. 43-48, Mar. (1971).
[13] NTL, A Library for doing Number Theory, available from ⟨http://www.shoup.net/ntl/⟩ (accessed 2015-8-30).