Vol.2019-MPS-125 No.12
2019/9/25

拡大体を用いた秘密分散法の計算量に関する考察

松田健[†1]　佐藤大樹[†2]　園田道夫[†3]

概要：データを安全に保管して利用するための方法として，データを分散管理することができる秘密分散法が知られている．秘密分散法のアルゴリズムの一種である，(k, n)しきい値法は有限体上の連立一次方程式の計算が必要であり，四則演算時に予め計算ルールとなる演算表を構成する必要がある．有限体の拡大体の性質を利用することで，乗法と除法の演算時に，拡大体の元のべき表現を用いることで演算表を構成せずに連立方程式の演算ができることが知られているが，素体と拡大次数によって，現実的に計算にかかる時間がどの程度であるかという情報を得ることは，現状では困難である．本研究では，(k, n)しきい値法に関する演算において，拡大体の次数，連立方程式の計算量，および，実際にかかる計算時間の関連性について具体的な問題を考えることで調査し，考察する．

キーワード：秘密分散，拡大体，実行時間

Investigation on calculation amount of Secret Sharing Method with Extended Field

TAKESHI MATSUDA[†1]　DAIKI SATO[†2]　MICHIO SONODA[†3]

Abstract: A secret sharing method that can manage and distribute data is known as a method for storing and using data safely. The (k, n) threshold method, which is one of secret sharing algorithm, requires calculation of simultaneous linear equations over a finite field, and it is necessary to construct an operation table that becomes a calculation rule in advance when performing four arithmetic operations. It is known that simultaneous equations can be calculated without constructing a calculation table by using the original power representation of the expansion field when calculating multiplication and division by using the properties of the expansion field of the finite field. It is difficult to obtain information about how much time is actually required for calculation by the prime field and the extended order. In this study, we investigated and considered the relationship between the order of the extension field, the computational complexity of the simultaneous equations, and the actual computation time in the operation related to the (k, n) threshold method.

Keywords: Secret sharing method, extended field, execution time

1. はじめに

クラウドサービスやIoT機器を利用することにより収集されるデータや個人情報や機密情報などの重要データを安全かつ確実に保管し，それらのデータを適切に利活用できる仕組みが重要である。データを安全に保管する仕組みとしては，データのバックアップや暗号化などが基本的である。秘密分散は，このようなデータのバックアップと暗号化を効率よく実現する手法と捉えることも可能である．一般的に，秘密分散を実現するには計算コストがかかることが知られており，排他的論理和演算のみで秘密分散法を実装する方法[1]や，演算結果の改ざんを検知可能な3パーティ秘匿関数計算プロトコル[2]など，関連研究について様々なものが提案されている．秘密分散や情報の復元にかかる計算量や実際にかかる計算時間については，問題の設定や使用するコンピュータリソースの影響も考慮する必要があり，正確に情報を知ることは困難である．しかしながら，これらの情報について，ある程度見積もることができれば，実際に実装する際に有益な情報になり得ると考えられる．本研究は，Shamir の(k, n)しきい値法において，秘密情報の分散時の計算時間の情報を特定の環境で実際に計測し，k, nの値，および体の標数と計算時間の関係が予測できるかどうか検討した．

†1 長崎県立大学情報セキュリティ学科
†2 中央大学理工学部
†3 国立研究開発法人情報通信研究機構

2. Shamir の(k,n)しきい値法

秘密分散法には様々なアルゴリズムが存在するが，本研究では，Shamir の(k, n)しきい値法を用いて計算時間を計測する．秘密分散法には，秘密情報の分散を実行するフェーズと，秘密情報の復元フェーズの二つのフェーズが存在する．n 人の参加者に対して，$k \leqq n$ なる自然数を考え，k 人が持つ情報から元の情報を復元する方法を以下にまとめる．標数 $p > 0$ の有限体を $GF(p)$で表す．$p \geqq n+1$ とする．

[秘密情報の分散]

$x_i \in GF(p)$ $(i=1,2,\cdots,n)$を公開されている情報とし，$GF(p)$から $k-1$ 個のランダムな元 r_j を選ぶ．分散させる情報を S とし，k 個のシェア w_i,を以下の式を用いて計算する．

$$w_{i1} = S + r_1 x_{i1} + r_2 x_{i1}^2 + \cdots + r_{k-1} x_{i1}^{k-1}$$

$w_{i2} = S + r_1 x_{i2} + r_2 x_{i2}^2 + \cdots + r_{k-1} x_{i2}^{k-1}$
…
$w_{i\,k} = S + r_1 x_{ik} + r_2 x_{ik}^2 + \cdots + r_{k-1} x_{ik}^{k-1}$

[情報の復元]

以下の式に w_{i1}, x_{i1} を代入して，連立一次方程式を計算して S と $r_1, r_2, \cdots, r_{k-1}$ を計算する．

$w_{i1} = S + r_1 x_{i1} + r_2 x_{i1}^2 + \cdots + r_{k-1} x_{i1}^{k-1}$
$w_{i2} = S + r_1 x_{i2} + r_2 x_{i2}^2 + \cdots + r_{k-1} x_{i2}^{k-1}$
…
$w_{ik} = S + r_1 x_{ik} + r_2 x_{ik}^2 + \cdots + r_{k-1} x_{ik}^{k-1}$

3. 実験と計算時間の推定

以下の仕様の PC を用いて，秘密情報の分散フェーズの実行時間を計測する．Sの値は 5，ランダムに決める rの値は全て 1 として計算を進める．

MacBook Pro (15-inch, 2017)
プロセッサ　3.1 GHz Intel Core i7
メモリ　16 GB 2133 MHz LPDDR3

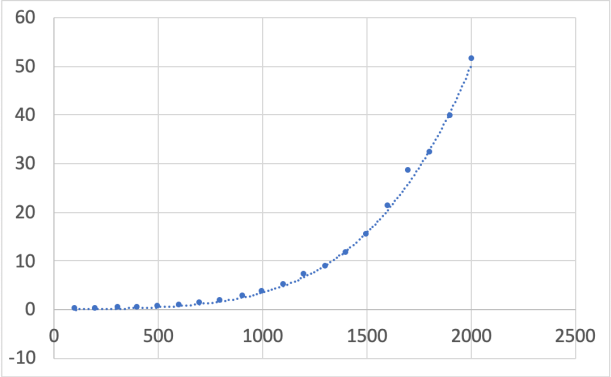

図 1 表 1 の p の値が 2003 以下の値の p と time の値
Figure 1 p and time values with a p of 2003 or less in Table 1

図 1 は，表 1 のデータに 4 次曲線を当てはめたときの散布図であり，曲線の方程式は，

$$y = 3.72*10^{-12}x^4 - 1.83*10^{-9}x^3 + 1.83*10^{-6}x^2 + 0.0006x - 0.1108$$

である．決定係数の値は 0.9978 である．この近似曲線で p=2503 以降の場合の時間を推定すると，

(p, time) = (2503, 125.537)
= (3001, 263.853)
= (3499, 494.768)
= (4001, 856.283)

となる．p の値が大きくなると計算時間の実際の値との差が大きくなることがわかる．

表 1　p, n, k と実行時間の関係
Table1　Relationship between p, n and k with execution time.

p	n	k	time[s]
101	100	99	0.007437
199	198	197	0.029303
307	306	305	0.098723
401	400	399	0.209213
499	498	497	0.416716
601	600	599	0.710076
701	700	699	1.155543
797	796	795	1.68329
907	906	905	2.533662
997	996	995	3.566696
1103	1102	1101	4.838234
1201	1200	1199	6.994625
1301	1300	1299	8.672607
1399	1398	1397	11.5618
1499	1498	1497	15.130483
1601	1600	1599	21.062002
1699	1698	1697	28.248213
1801	1800	1799	32.021135
1901	1900	1899	39.510783
2003	2002	2001	51.36557
2503	2502	2501	113.920948
3001	3000	2999	203.948909
3499	3498	3497	328.07963
4001	4000	3999	536.424088

4. 考察と今後の課題

表 1 は拡大体を用いない場合の計算時間の結果であるが，p の値が小さい場合かつ拡大次数が小さな場合は計算時間に大きな影響があるものとは考えにくい．素数 p の値が大きく，かつ S の値としてある程度実運用に近い値のものを考慮した際の実行時間について調査を進めることが今後の課題である．

参考文献

[1] 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“．情報処理学会, Computer Security Symposium,8C-2,2005.
[2] 千田浩司，五十嵐大，濱田浩気，高橋克巳: “エラー検出可能な軽量 3 パーティ秘匿関数計算の提案と実装評価“．情報処理学会論文誌, Vol.52, No.9, pp.2674-2685, 2011.