Vol.2015-DPS-162 No.42
Vol.2015-CSEC-68 No.42
2015/3/6

クラウドサービス利用者の安心感を高める軽量秘密計算法の実クラウドにおける実験

韓　嘯公[†1]　宮西 洋太郎[†2]　北上 眞二[†1]　浦野 義頼[†1]　白鳥 則郎[†1]

クラウドコンピューティングを利用する場合，利用者はプログラムもデータもクラウド事業者に保管，管理，実行を委ねることになる．プログラムやデータは企業競争力の源泉であり，競争相手に対して，企業は厳しく秘密を守りたい．このような点から利用者のセキュリティに関する不安が払拭できない．本研究では，クラウド事業者の社内からの不注意や悪意によるプログラム（処理ノウハウ）やデータの漏洩，不正使用についての対抗策を検討する．その対抗策の１つとして，セキュアマルチパーティ法による秘密分散法，秘密計算法が研究され，実用化されている．この方法の特性として，加減算については，容易に実現できるが，乗除算については，パーティ間での煩雑なデータ交換が必要である．我々は乗除算についても，効果的な工夫と制限を設けることにより，効率的に計算結果が得られる方法の提案を行っている．本稿では，この提案方式について，実クラウド環境において実験し，その有効性を示す．

An Experiment in Real Clouds for a Lightweight Secure Multi Party Computation Method to Increase User's Sense of Safety

XIAOGONG HAN[†1]　YOHTARO MIYANISHI [†2]　SHINJI KITAGAMI[†1]
YOSHIYORI URANO[†1]　NORIO SHIRATORI[†1]

Users of cloud computing services could not wipe away the anxiety about the data and programs may be abused or leaked, because the users hand them over to a cloud provider. Countermeasures should be studied against the abuse or leakage caused by cloud providers' careless or intentional crime. The "secret sharing and secret computation by secure multi parties method" is practically used in some applications, but it requires complicated data exchange between the parties in the cases of multiplication and division. We newly proposed a "lightweight secure multi party computation method" that can efficiently compute even in these cases, by providing effective restrictions. In this paper, we shows the usefulness of the proposed method through the experiment in a real cloud environment.

1. はじめに

近年，クラウドコンピューティングが普及しつつある．クラウドコンピューティングは計算能力を柔軟に拡大，縮小でき，総合的に情報システムのコストを下げうるなどの長所がある反面，データの保管や処理(プログラム)を外部企業(クラウド事業者)に委託することにより，データや処理方法の不正使用や漏洩のリスクをもつという短所がある．この短所ゆえに，委託するデータやプログラムの性質によっては，利用者は十分な安心感をもって利用することができない．

本稿では，既に提案している「軽量秘密計算法」について，実クラウドを用いて実験を行い，有効性を示す．具体的には，平均値，分散，標準偏差，相関係数の 4 つの統計計算処理，および１つの個別計算処理について，軽量秘密計算法の動作を実クラウド環境において検証し，評価と考察を行う．実験により，単独のクラウドからデータが不正使用されたり漏洩されたりしても，データおよび計算結果の秘密が確保されることが実クラウド環境で確認できた．

†1 早稲田大学
WASEDA University
†2 (株)アイエスイーエム
ISEM, Inc.

2. 技術的課題と研究目的

2.1 技術的課題

クラウドを利用する場合，データやブログラムをクラウド事業者に委託することとなり，クラウドに保管されているデータやプログラムがクラウド事業者内部に起因して不正使用されたり，または漏洩される等のセキュリティリスクの問題がある．

上記のセキュリティリスクに関する不安に対して，利用者は現在では大略，次のような 2 つの対策によっている．
(1)企業秘密であるデータやプログラムにはパブリッククラウドを利用しない（本稿の対象外）．
(2)クラウドを利用する場合には，信頼性の高いクラウド事業者を選択し，政府発行のガイドラインにそった利用，運用を行う[1][2].

上記のガイドラインでは，クラウドにおけるデータの暗号化が推奨されている．データの暗号化は，クラウド外部から不正にデータを読み取ろうとする攻撃には有効な対策であるが，クラウド内でプログラム実行時，暗号データは復号され，平文となり，クラウド事業者にはデータが明らかとなる（ただし，後述の完全準同型暗号方式によれば，この限りではない）．

さらには，データベースとプログラムが同一クラウドに配置されていれば，上記の復号時に使用した暗号鍵がクラウド事業者に知られることとなり，データベース全体が明らかとなってしまう．従って暗号化しても，セキュリティリスクが存在することになる．

このリスクに対しては，SLA(Service Level Agreement)などの人間的な「約束ごと，契約」によって防止する対策はあるが，そのような方法では技術またはシステムによる解決策としては十分とは言いがたい．

3. 従来の技術的な対応

セキュリティリスクに対して技術またはシステムによる解決策として，有望な技術には，秘密分散法[3]および秘密計算法がある[4].

3.1 秘密分散法(secret sharing)，電子的割符技術(electronic tally method)

秘密分散法の 1 つに(k, n)閾値法がある[5]. この方法は，データを秘匿するため，秘密にすべきデータを k-1 次多項式によって変換して秘密データ（「シェア」と称される）とし，n 個のサーバに分散配置する．復元は，k 個のサーバからデータを得て連立方程式を解くことによりなされる．電子的割符技術は，あるデータをビットレベルで秘密データに加工する技術である．

秘密分散法は個人情報(プライバシー情報)の匿名化や電子投票に応用されている．またセキュリティ問題に応用したシステムも実用化されている[6][7].

3.2 秘密計算法(secure computation)

秘密計算法の 1 つに，完全準同型暗号を使用した秘密計算法がある[8]．この方法では，クラウド側は暗号化されたデータをそのまま演算処理して，暗号化された計算結果を返答する．この返答を受けて，ユーザ側で暗号化された計算結果を復号して，平文の計算結果とする．よってクラウド側での，利用者データの秘密が保たれる．

上記の方法は，1 つの数値データをビット列として表現するなど，通常よりも長いビット長を必要とし，まだ実用のシステムに適用するレベルには達していない[4]とされているが IBM 社からライブラリ提供の広報もなされている[9]．今後に注目したい．

秘密計算法は，上記の完全準同型暗号を用いる方法のほかに 3.4 に述べるセキュアマルチパーティによる計算法もある[4]．後者は現時点では限定された機能で実用化されている[10].

3.3 データ・プログラムの分割による方法

全体としてのデータやプログラムを分割し，別々のクラウドに配置することにより，部分的に秘密が破られても，全体としてのデータやプログラム(処理方法)の秘密を守るという提案も我々は行っている[11][12]. 本稿ではこの方式を，「複数クラウドへの分割法」と称する．

3.4 セキュアマルチパーティ法

秘密計算法の 1 つであるセキュアマルチパーティ法では，秘密の保持と信頼性を両立させるために，秘密にしたい数値データ a，b（0 以上 m 未満の整数）について，n 個のサーバに分散させ，それらのサーバのうち，異なる k 個のサーバからデータを得て，元の数値 a, b を復元する仕組みとなっている(本稿では，k out of n と略する場合がある)．本研究での提案方式を説明する上で，本稿では，既存方式の説明が必要である．文献[4]に従い，k=2，n=3 の場合について，基本的動作を紹介する．

数値 a，b を，乱数を用いるなどで，a=a0+a1+a2，b=b0+b1+b2 となるように，（a0，a1，a2）及び（b0，b1，b2）にユーザサーバにおいて分割し，（a0，a1），（b0，b1）を外部サーバ 0 に，（a1，a2），（b1，b2）を外部サーバ 1 に，（a2，a0），（b2，b0）を外部サーバ 2 に分散配置する．これらのデータは, それぞれ a, b の「シェア」と称される．

この a と b についての分割，分散配置を図 1 に示す．c については，乗算用であり，本節後半に述べる．

このような準備の下に，ある計算要求が発生したとする．まずは，a と b の加算（減算も同様）の場合，図 1 において, サーバ n (n=0, …, 2)では an と bn の加算及び an+1 と bn+1 の加算を計算する（n+1 は mod 3 の計算）．それぞれの結果は，サーバ n に対して，秘密が保持されている．それぞれの結果がユーザ側（ユーザサーバ）に返答される．3 個のサーバのうち 2 個から計算結果が得られれば（すなわち，2 out of 3），求める結果が得られる．ユーザ側では，それぞれの結果を加算することによって，計算要求の結果を得ることができる．

$$(a0+b0)+(a1+b1)+(a2+b2)=(a0+a1+a2)+(b0+b1+b2)$$
$$=a+b$$

係数のかかった重み付け加減算についても同様である．

次に，計算要求が乗算の場合，動作は複雑になるが，概略，次のようなプロセスで結果を得ることができる．この場合にも，3 個のサーバのうち, 2 個のサーバから計算結果を得ることにより，要求された計算結果を得ることができるが，途中のプロセスでは，3 個のサーバはすべて正常に動作していることが必要である．

(1)サーバ 0 において

1)ランダム整数（0 以上 m 未満）c0，r1，r2 を生成する．

2)$c1 = (a0+a1)\cdot(b0+b1) - r1 - r2 - c0 \pmod m$を生成する．

3)サーバ 1 に（c1，r1）を送る．

4)サーバ 2 に（c0，r2）を送る．

5)サーバ 0 にシェアとして（c0，c1）を保持する．

(2)サーバ 1 において

1)$y = a1\cdot b2+a2\cdot b1+r1 \pmod m$を計算する．

2)y をサーバ 2 に送る．

(3)サーバ 2 において

1)$z = a2\cdot b0+a0\cdot b2+r2 \pmod m$を計算する．

Vol.2015-DPS-162 No.42
Vol.2015-CSEC-68 No.42
2015/3/6

2)z をサーバ 1 に送る.

(4)サーバ 1，2 において

1)c2 = y+z+a2･b2 を計算する.

2)サーバ 1 にシェアとして（c1，c2）を保持する.

3)サーバ 2 にシェアとして（c2，c0）を保持する.

以上の準備のもとで，異なる 2 個のサーバから得られる c0，c1，c2 を加算することによって，a と b の積 a･b が得られる.

c0+c1+c2

=c0+(a0+a1)･(b0+b1) – r1 – r2 – c0

+(a1･b2+a2･b1+r1)+(a2･b0+a0･b2+r2)+a2･b2

=a0･(b0+b1+b2)+a1･(b0+b1+b2)+a2･(b0+b1+b2)

=(a0+a1+a2)･(b0+b1+b2)

=a･b

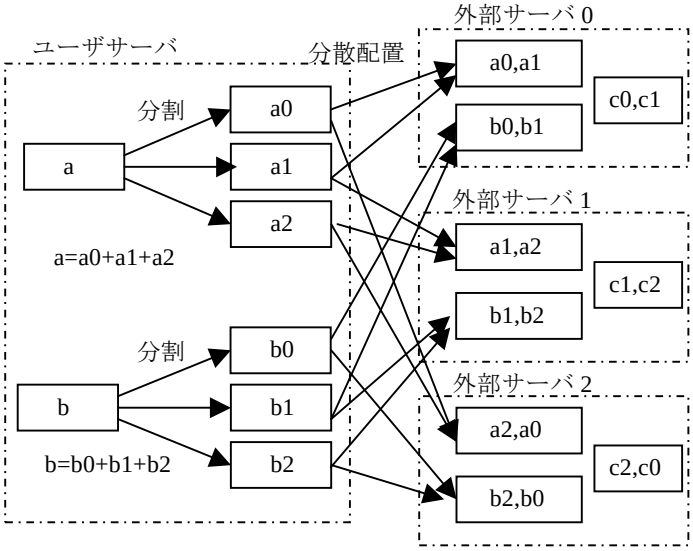

図 1　秘密にすべきデータの分割と分散配置

セキュアマルチパーティ法では，上記のようにデータが a，b（RDB（Relational Database）ならば，属性に相当する）と 2 つの場合には，あらかじめ（c0，c1），（c1，c2），（c2，c0）を計算し，その結果を 1 組のシェアとして，各サーバに保持すればよい．しかし，データが a，b,... と多くなるに従い，2 項の乗算の場合に限っても，組み合わせ数の 2 分の 1 の数の組をシェアとして保持する必要がある．データ種類（属性）の数を p とすると，シェアの組の数 q は

$$q = p\cdot(p - 1)/2$$

となる. RDB の場合，これに行の数が乗算される. この（c0，c1），（c1，c2），（c2，c0）をシェアとして保持する方式では，データの量が過大となる問題がある.

これを避けるには，計算要求があるたびに，(1)から(4)のサーバ間での連携処理を行うことが必要である. これは，ネットワークに関連する問題（エラー処理など）や処理時間が問題となる可能性がある.

さらに，除算や，3 項の計算を含めた一般的な計算式に対して，およびサーバ数の増減に対して，検討が必要である（上記の動作説明は，2 out of 3 に強く依存した説明になっていて，2 out of 2 や一般的な k out of n といった構成に対して，更なる検討が必要である）.

3.5 軽量秘密計算法

3.4 で述べた問題点を回避するため，我々は軽量秘密計算法(lightweight secure multi party computation)を提案した[13][14]. その本質的な要点は，①加減算用の秘密分散データ（シェア）に加えて，②乗除算用の秘密分散データを保持させる．加減算用のシェアの和は元のデータになるように（既存方式と同じ），乗除算用のシェアは積が元のデータになるように，分割する.

従来のセキュアマルチパーティ法では，「分割」という用語は使われていないが，シェアを作成することを明示するために軽量秘密計算法では，分割と称している.

この提案による分割と分散配置を図 2 に示す.

まず従来と同様に，数値 a，b を，乱数などを用いて，加減算用に a=Sa0+Sa1+Sa2，b=Sb0+Sb1+Sb2 となるように，（Sa0，Sa1，Sa2）及び（Sb0，Sb1，Sb2）にユーザサーバにおいて分割し，（Sa0，Sa1），（Sb0，Sb1）を外部サーバ 0 に，（Sa1，Sa2），（Sb1，Sb2）を外部サーバ 1 に，（Sa2，Sa0），（Sb2，Sb0）を外部サーバ 2 に分散配置する．S は summation(和)を意味する.

次に乗除算用に, 数値 a, b を, 乱数などを用いて, a=Pa0･Pa1･Pa2，b=Pb0･Pb1･Pb2 となるように，（Pa0，Pa1，Pa2）及び（Pb0, Pb1, Pb2）にユーザサーバにおいて分割し，（Pa0, Pa1），（Pb0，Pb1）を外部サーバ 0 に，（Pa1，Pa2），（Pb1，Pb2）を外部サーバ 1 に，（Pa2，Pa0），（Pb2，Pb0）を外部サーバ 2 に分散配置する．このような準備の下に，ある計算要求が発生したとする．加減算については，従来の方法と同様である.

計算要求が乗算の場合，各サーバ n では，Pan と Pbn の乗算および Pan+1 と Pbn+1 との乗算を行う（n+1 は mod 3 の計算）．P は product（積）を意味する.

3 つのサーバのうち，2 つのサーバから計算結果が得られれば, Pa0･Pb0, Pa1･Pb1, Pa2･Pb2 がそろうこととなる. これらの乗算は,

(Pa0･Pb0)･(Pa1･Pb1)･(Pa2･Pb2)

＝(Pa0･Pa1･Pa2)･(Pb0･Pb1･Pb2)

＝a･b

となり，元のデータ a，b の乗算となる.

除算 a/b についても，同様に処理を行うことができる. 計算要求が除算の場合，各サーバ n では，Pan と Pbn の除算および Pan+1 と Pbn+1 との除算を行う（n+1 は mod 3 の計算）.

3 つのサーバのうち，2 つのサーバから計算結果が得られれば，Pa0/Pb0，Pa1/Pb1，Pa2/Pb2 がそろうこととなる. これらの乗算は,

(Pa0/Pb0)･(Pa1/Pb1)･(Pa2/Pb2)

Vol.2015-DPS-162 No.42
Vol.2015-CSEC-68 No.42
2015/3/6

＝(Pa0･Pa1･Pa2)/(Pb0･Pb1･Pb2)

＝a/b

となり，元のデータ a，b の除算となる．

いずれの場合も各サーバが知りうるのは，3 つの計算結果のうち 2 つのみであり，計算前のデータと同様に，計算結果のデータについても秘密が保持されている．また，提案方式は，k，n について，単純な計算方式であるので，自由に選択し，システムを構成することができる．

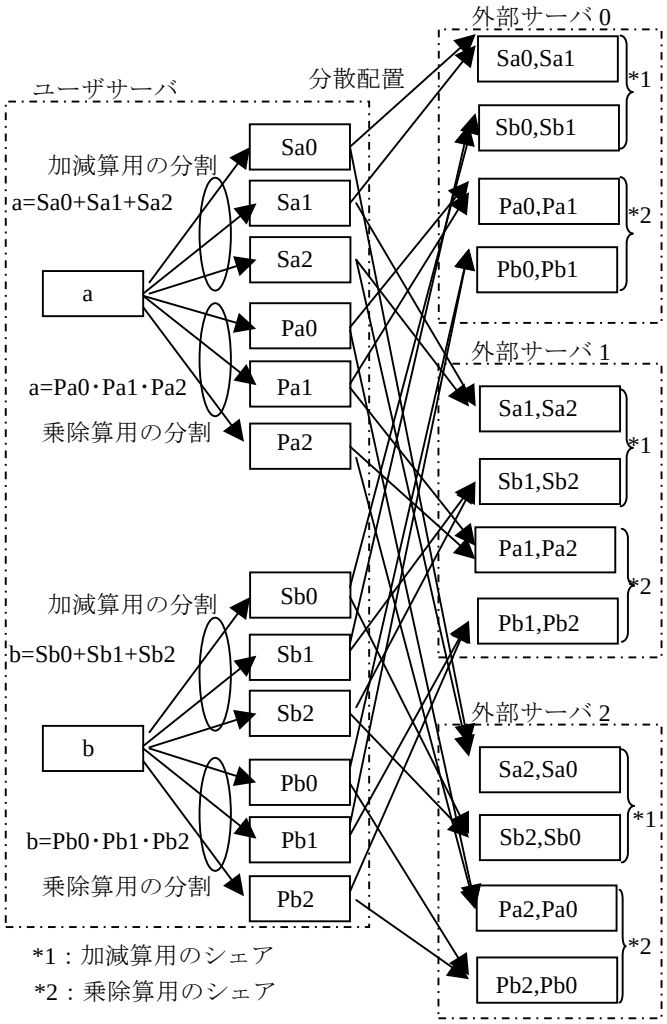

図 2　軽量秘密計算法におけるデータの分散配置

4. 実験システム

4.1　概要

軽量秘密計算法を評価するための本稿の実験システムを図 3 に示す．実験システムにおいて，サーバとして 2 つの商用サービスの「サクラインターネットの VPS (Virtual Private Server)サービス」，研究室のパソコン（win7, Java, MySQL），および実インターネット環境を用いて実験を行った．実験では，まずクラウド 1 とクラウド 2（サクライインターネット VPS）において，元のデータ x のシェア x1, x2 について，各クラウドで f(x1)，f(x2)の処理を行う．次に, 各クラウドで演算した結果をユーザサーバに転送する．最後に, ユーザサーバで統合処理を行い, 最終結果を得る．

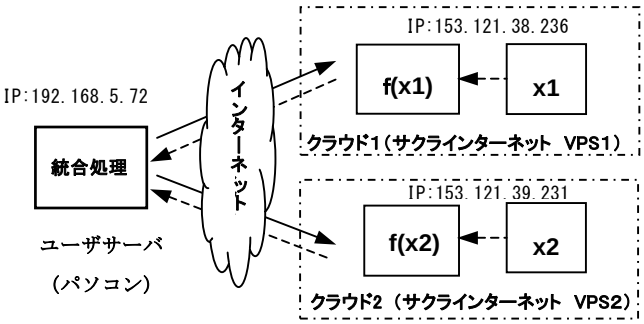

図 3　実験概要図

4.2 計算例

計算例として学生の成績評価システムを想定する．

この学生成績評価システムは以下の 2 つの計算機能を有するものとする．

(1)統計計算処理として, 全科目の平均値, 分散値(統計の), 標準偏差値，例えば，数学成績と物理成績の間の相関係数

(2)個別計算処理として，例えば，ある学生の数学と物理成績の加重和（d1･x1+d2･x2）など．

図 4 は実験システムの表示画面の例である．画面は，シェア作成用の乱数の入力領域と計算結果の表示領域からなる．

システムは，プログラミング言語として Java を用いて，DBMS は MySQL を用いた. 学生 50 人の十科目の成績データは予めシステム内に設定している．

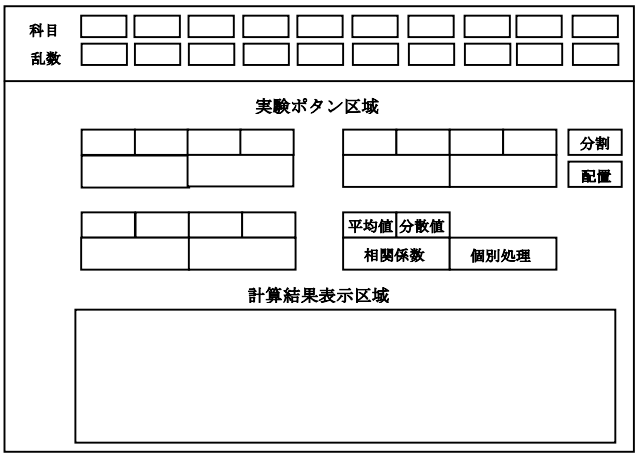

図 4：実験システムの画面表示の例

4.3 ユーザサーバでの準備段階の処理

データの秘密分割は乱数を利用し，元のデータ（本稿は学生成績）の分割・配置のことである．今回実験の外部用クラウドサーバーは 2 台であるので，2 つのシェアに分割することになる．

(1)加減算用のシェア作成

x = x * r + x * (1 - r) であるので,

x *r および x * (1 - r) に分割し，シェアとする．

ここで，x は元のデータ，r は乱数．r は 0，1 を避け，例えば，-9.0 < r < -2.0, または 2.0 < r <9.0 を選ぶ．

Vol.2015-DPS-162 No.42
Vol.2015-CSEC-68 No.42
2015/3/6

(2).乗除算用のシェア作成

x = √x * r * √x * (1/ r)　であるので,

√x * r および√x * (1/ r) に分割し，シェアとする．

ここで，x は元のデータ，r は乱数．r は 0，1 を避け，例えば，0.2 < r < 0.9, または 2.0 < r <9.0 を選ぶ．

4.4 各サーバでの統計処理

4.4.1 平均値の計算

科目 g，学生 i の得点を xgi と，人数 n すると，科目 g の平均値 E(xg)は，次式で得られる．

$$E(xg)=(xg1+xg2+\cdots+xgn)/n$$

図 5 は平均値計算処理概要図である．各サーバでは以下の処理を行う．科目 g について全科目を行う．

ユーザサーバ：科目 g（g の範囲は 10）を指定して，クラウド 1，2 に平均値計算プログラムを要請する．

クラウド 1：科目 g について，xg1s の縦方向への総和を求め，結果を要求元（ユーザサーバ）に返答する．

クラウド 2：科目 g について，xg2s の縦方向への総和を求め，結果を要求元（ユーザサーバ）に返答する．

ユーザサーバ：クラウド 1，クラウド 2 からの計算結果を加算し，学生数（この例では 50）で除算して，平均値 E(xg)とする．

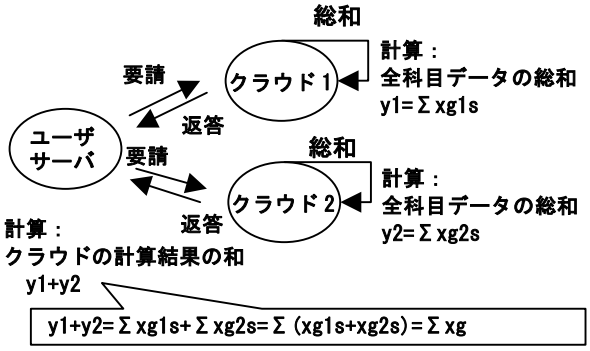

図 5　平均値計算処理概要図

4.4.2 分散値の計算

分散は，公式：$\sigma^2=E((x-E(x))^2)=E(x^2)-E(x)^2$ により，4.4.1 で求めた平均値 E(x)を利用できるので，自乗平均，すなわち自乗の総和（縦方向）$E(x^2)$を求めればよい．

図 6 は分散値計算処理概要図である．

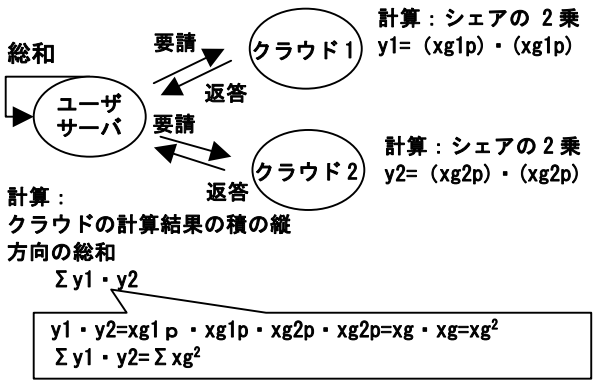

図 6　分散値計算処理概要図

各サーバは，以下の処理を行う．

ユーザサーバ：科目 g を指定して，クラウド 1，2 に分散値計算プログラムを要請する．

クラウド 1：学生 ID ごとに，科目 g について，xg1p の 2 乗を計算し，結果を要求元（自社サーバ）に返答する．

クラウド 2：学生 ID ごとに，指定科目 g について，xg2p の 2 乗を計算し，結果を要求元（自社サーバ）に返答する．

ユーザサーバ：学生 ID ごとに，クラウド 1，クラウド 2 からの計算結果を乗算し，全学生について（縦方向への）総和をもとめ，学生数（この例では 50）で除算して，自乗平均値 $E(xg^2)$とする．

4.4.3 標準偏差の計算分散処理概要図

標準偏差値 σ は分散値 σ^2 の平方根であるので，計算方法は上記分散の計算方法が同じである．

4.4.4 相関係数の計算

科目 g と科目 h の相関係数を計算する．相関係数 ρgh は「xg と xh の共分散」を「g の標準偏差と h の標準偏差の積」で除したものである．

すなわち，公式：$\rho gh=E(xg\cdot xh)/(\sigma g\cdot \sigma h)$　により，4.4.3 において求めた，各科目の標準偏差 σg，σh を利用できるので，科目間の積 xg･xh の総和（n で除して平均）を求めればよい．

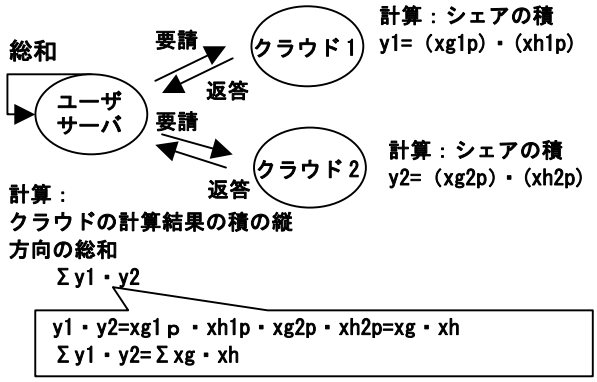

図 7 相関係数計算処理概要図

図 7 は相関係数係数処理概要図である．各サーバでは以下の処理を行う．

ユーザサーバ：科目 g，h を指定して，クラウド 1，2 に相関係数計算プログラムを要請する

クラウド 1：学生 ID ごとに，指定科目 g，h について，xg1p と xh1p の積を計算し，結果を要求元（ユーザサーバ）に返答する．

クラウド 2：学生 ID ごとに，指定科目 g，h について，xg2p と xh2p の積を計算し，結果を要求元（ユーザサーバ）に返答する．

ユーザサーバ：学生 ID ごとに，クラウド 1，クラウド 2 からの計算結果を乗算し，全学生について（縦方向への）総和をもとめ，学生数（この例では 50）で除算して，共分散 E(xg･xh)を求める．

Vol.2015-DPS-162 No.42
Vol.2015-CSEC-68 No.42
2015/3/6

4.5 各サーバでの個別計算処理

個別計算処理として，加算，除算の混合した計算を行う．

図 8 は個別計算処理概要図である．図により，

たとえば，y=d･x1＋e･x2＋f･x1／x2 の場合は

y11= d･x11s＋e･x21s

y12= d･x12s＋e･x22s

それぞれ結果 y11，y12 をユーザサーバに返答させる．

y2 =f･x1／x2 をクラウド 1，クラウド 2 で行い，

y21=f･x11p／x21p

y22=f･x12p／x22p

それぞれの結果 y21, y22 をユーザサーバに返答させる．

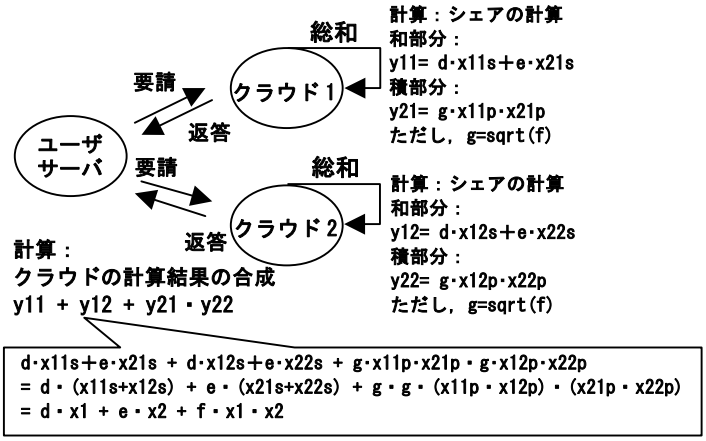

図 8 個別処理概要図

ユーザサーバでは，クラウド 1 からの y11 とクラウド 2 からの y12 を加算し，y3 とする．

y3=y11＋y12

除算部分は，クラウド 1 からの y2 とクラウド 2 からの y2 の積を行い y4 とする．

y4=y21･y22

最終的な結果 y は，y3 と y4 の和である．

y=y3+y4

5. 評価と考察

5.1 評価

軽量秘密計算法の実クラウド環境での実験を行い，想定とおりの結果が得られ，有効性を確認した．

次に，方式の評価として，安心感，機能（除算・実数演算），信頼性，性能（転送回数・処理），精度，構築の容易性という 6 つ観点について，軽量秘密計算法，従来のマルチパーティ法および複数クラウドへの分割法の間で比較し，実験結果と従来の検討に基づいて，定性的な評価を行う．表 1 に比較評価を示す．

安心感：軽量秘密計算法と従来のマルチパーティ法はデータ本体を分割しシェアとし，元の値を隠蔽しているので，安全性の順位が高いと評価する．複数のクラウドへの分割法は暗号を利用しないから，他の二種類方法と比較し，順位が低いと評価する．

機能（除算・実数演算）：従来のマルチパーティ法は除算の機能が無いこと，および実数演算機能が無いことから，他の 2 つ方式により順位が低いと評価する．

信頼性：信頼性は，複数のクラウドを利用したとき，例えば 2 out of 3 の場合，3 つのクラウドのうち，2 つが正常動作していれば，システムとして，機能するという意味での信頼性である．

信頼性の比較において，例えば軽量秘密計算法では，2 out of 3 の場合，2 つのクラウドが正常動作していれば，システムとして機能するが，従来のマルチパーティ法では，2 out of 3 の場合でも，3.4 で述べたように，結果を得る途中のプロセスで，3 つのクラウドが正常動作していることが必要である．その分，信頼性は，軽量秘密計算法よりも低い．

性能：性能について，クラウド間でのデータの転送回数と処理量という視点から評価する．従来のマルチパーティ法は 3.4 で述べたように，計算プロセス途中でのデータ分散･配置の煩雑なやりとりが多いので，性能の順位が低い．軽量秘密計算法は事前の準備が簡単し，処理中でもデータの転送回数が少ないので，順位が高いと評価する．

精度：精度方面について，軽量秘密計算法は除算を導入し，他の二つ方法と比べて，精度が高いと評価する．

構築の容易性：複数クラウドへの分割法は，事前にデータのシェアを作成する必要がなく，構築が一番し易い．軽量秘密計算法と従来のマルチパーティ法では，事前にデータのシェアを準備する必要があり，構築の容易性が複数クラウドへの分割法により高いと評価する．

また，軽量秘密計算法では，クラウドの数を任意に選択できるが，従来のマルチパーティ法では，クラウドの数に制約があり，構築の容易性は低い．

表 1　評価表
（◎：良い，○：普通，△：良くない，×：できない，を意味する）

	1 安心感	2 機能（除算・実数演算）	3 信頼性	4 性能（転送回数・処理）	5 精度	6 構築の容易性
本研究 軽量秘密計算法	◎	◎	◎	◎	◎	○
従来のマルチパーティ法	◎	×	○	△	○	△
複数クラウドへの分割法	△	○	△	○	△	◎

5.2 考察

本稿では，統計処理の平均値，分散値，標準偏差値，属性間の相関係数を求めたり，個別処理の加減算と乗除算を含む計算処理の実験を行った．上記の 5 つの統計計算が実行できるという結果を得た．

従来の方式（マルチパーティ法と複数クラウドへの分割

法）と比較し，軽量秘密計算の方は計算前の準備のやりとりと時間が少なくて，除算と計算精度方面も有利であると考える．また，軽量秘密計算法の信頼性が高いことを検証し，ユーザの安心感を高めることができる．

ただし，従来のクラウドを利用する方法に比べ，ユーザサーバで行う処理を皆無にすることができない．4.1 で述べた統合処理をユーザサーバで行うことが必要である．

また，今回の実験の数値データタイプは浮動小数点 double 精度で行ったが，乱数の選択によっては，計算の精度に影響がでることが想定されるので，今後は，その点での研究が課題である．

6. おわりに

本研究では，クラウドサービスにおける利用者の安心感を実現す軽量秘密計算法のデータ加減算と乗除算シェアの秘密分割・秘密計算方式について実験を行い，その有効性を検証した．

今後の課題はデータ分割の大小範囲，適用する乱数の検討，計算精度の検討，データ破損の自動修復と軽量化秘密計算法の統計計算以外の適用についても検討していきたい．

参考文献

1) 経済産業省：クラウドサービス利用のための情報セキュリティマネジメントガイドラインの公表～クラウドサービスの安全・安心な利用に向けて～ (2011 年),
http://www．meti．go．jp/press/2011/04/20110401001/20110401001．html
2) 総務省：ASP・SaaS における情報セキュリティ対策ガイドライン （平成 20 年，2008 年），
http://www．soumu．go．jp/main_sosiki/joho_tsusin/policyreports/chousa/asp_saas/pdf/asp_saas_zentai．pdf
3) 千田浩司他：照合匿名化クラウドの課題と対策， 信学論（A），Vol．J96-A No．4 pp．149-156, 2013/4
4) 千田浩司：安全な情報処理を目指す秘密計算技術の研究動向と実用化に向けた取り組み， 情報処理 Vol．54 No．11 pp．1130-1134 Nov．2013
5) Shamir, A．: How to share a secret, Comm．ACM, Vol．22, No．11, pp．612-613, (1979)
6) NRI セキュア：データセンターを活用した情報漏えい防止策，
http://cloud. watch. impress. co. jp/epw/docs/news/20100301_351998．html
7) ソリトンシステム：電子割符技術 Tally-WariZen,
http://www．itmedia．co．jp/enterprise/articles/1310/07/news001．html
8) Gentry, C．: Fully Homomorphic Encryption Using Ideal Lattices, STOC2009, pp．169-178 (2009)
9) IBM：完全準同型暗号ライブラリをオープン化,
http://news．mynavi．jp/news/2013/05/09/094
10) NTT : 医療統計処理における秘密計算技術を世界で始めて実証,
http://www．ntt．co．jp/news2012/1202/120214a．html#6
11) 宮西洋太郎：クラウドコンピューティングでの高度セキュリティ実現方式の提案～情報処理委託内容の秘匿方式～， 信学技報，Vol．110 No．302 pp13-17, 2010/10
12) 韓嘯公他：クラウドサービスにおける利用者の安心感を実現するセキュリティ確保方式,
第 76 回情報処理学会全国大会，3D-7, 2014/3
13) 宮西洋太郎他：クラウドサービス利用者の安心感を高める簡易的秘密計算法の提案， 信学技報，Vol．114 No．49 SWIM2014-4 pp19-24，2014/5
14) 金岡晃他： 実数演算可能な軽量秘密計算法の一考察，CSS2014(Computer Security Symposium 2014)，Oct．2014 Sapporo pp682-687