情報処理学会論文誌 Vol.60 No.7 1294–1303 (July 2019)

現物給付型サービスに向けた秘密計算プロトコルの提案

坂崎 尚生[1,a)]

受付日 2018年9月20日, 採録日 2019年4月9日

概要：本稿では，高額医療・高額介護合算療養費制度等における自己負担額の世帯合算および現物給付型サービスの仕組みを電子的に実現する方法について，セキュリティ面からの検討を行う．より具体的には，暗号化状態のまま計算が可能な秘密計算と呼ばれる技術を上記仕組みに適用するための要件を定義し，その要件をすべて満たす秘密計算プロトコルを提案する．

キーワード：秘密計算，Paillier 暗号，準同型暗号，高額医療・高額介護合算療養費制度，総合合算制度，現物給付

Proposal of Secret Computation Scheme for Benefit-in-kind Services

Hisao Sakazaki[1,a)]

Received: September 20, 2018, Accepted: April 9, 2019

Abstract: On April 1, 2008, Japanese government implemented the high-cost medical care benefit system. In this research, we study the security towards computerization of the high-cost medical care benefit system. Our aim is to establish secure tabulation-technology for the high-cost medical care benefit system. To this end, first, we modeled the features of the high-cost medical care benefit system and defined the security requirements based on this model system. Then, we designed a secure tabulation-scheme that meets all of the security requirements.

Keywords: secret computation, Paillier cryptosystem, homomorphic encryption, high-cost medical care benefit system, total aggregate system, benefit in kind

1. はじめに

2008 年 4 月 1 日より始まった高額医療・高額介護合算療養費制度は，世帯内の同一の医療保険の加入者について，毎年 8 月から 1 年間にかかった医療保険と介護保険の自己負担額を世帯合算し，基準額を超えている場合は，加入している医療保険の窓口に申請することにより，その超えた金額を受け取ることができる現金給付型の制度である[*1]．

この制度導入により，同一世帯において医療と介護でかかった費用負担を緩和することができる．

また，2015 年 10 月に政府与党により見送りになってしまったが，医療費と介護費だけではなく，障害・保育にかかわる費用も世帯合算させる総合合算制度の導入も検討されていた [7]．総合合算制度では，患者が支払った医療費等の自己負担額をマイナンバー等を利用して集計し，医療機関等がその世帯合算値を確認することにより，基準額に達した患者は，受診時の窓口負担をしなくても済む現物給付型のサービスも検討されていた．なお，総合合算制度は，消費税軽減税率の財源確保のため，現時点では見送りになっているが，2017 年の衆議院選では，総合合算制度の導入を公約していた政党もあり [9], [10]，総合合算制度について再検討の声もあがっている．

本稿では，高額医療・高額介護合算療養費制度または総合合算制度における現物給付型サービスの電子化に向け

[1] 株式会社日立製作所研究開発グループシステムイノベーションセンタ
Research & Development Group, Center for Technology Innovation - Systems Engineering, Hitachi, Ltd., Yokohama, Kanagawa 244–0817, Japan
[a)] hisao.sakazaki.qc@hitachi.com

[*1] 高額医療・高額介護合算療養費制度では，事前に全国健康保険協会の各都道府県支部から健康保険限度額適用認定証を取得し，医療機関に同認定証を提出することにより，受診時の窓口負担をしなくても済む現物給付型のサービスも受けることができる．

© 2019 Information Processing Society of Japan

て，セキュリティ面より検討を行う．より具体的には，現物給付型サービスのモデル化を行い，そのモデルを用いて同サービスに求められる要件を定義する．さらに，その要件をすべて満たす方法の一例として，マルチユーザ環境で利用可能な秘密計算プロトコルを提案する．

本稿の構成は次のとおりである．2 章では医療費等合算制度の電子化に向けた現物給付型サービスのモデル化と要件定義をする．3 章ではサービスモデルのセキュリティを検討するうえでの前提条件と攻撃者モデルについて説明する．4 章では現物給付型サービスに既存のセキュリティ技術を適用するうえでの課題を説明する．5 章で要件をすべて満たす秘密計算プロトコルを提案し，6 章で本稿をまとめる．

2. サービスモデルと要件定義

政府検討資料 [7] では，医療機関等のサービス事業者が該当者の世帯合算値を確認する具体的な方法についてまでは明記されていない．つまり世帯合算値を確認する方法として，サービス事業者が能動的に該当者の世帯合算値を取得して確認するのか，または，現状の高額医療・高額介護合算療養費制度の現物給付型サービスのように，利用者があらかじめ取得した限度額適用認定証を会計時にサービス事業者に提出させることで，受動的に該当者の世帯合算値を確認するのかまでは，明記されていない．

しかし，マイナンバー制度の目的 [8] の 1 つとして添付書類の削減を掲げていることから，現物給付型サービスの将来像としては，限度額適用認定証のような添付書類の提出を必要としない前者の仕組みの方が望ましいと考える．

それゆえ，本稿では，現物給付型サービスにおける世帯合算値を確認する方法として，医療機関等のサービス事業者が能動的に該当者の世帯合算値を取得する仕組みをあるべき姿として検討を行った．

2.1 サービスモデル

図 1 に医療費等合算制度の電子化に向けた現物給付型サービスの想定モデルを記す．現実の制度では，社会保険診療報酬支払基金や国民健康保険団体連合会が実務主体とサービス事業者の間に入り，診療報酬の審査等の業務を行っているが，本稿では，簡略化のため，実務主体は審査済の診療報酬を管理しているとして，社会保険診療報酬支払基金等の審査業務をモデル化の対象外とする．

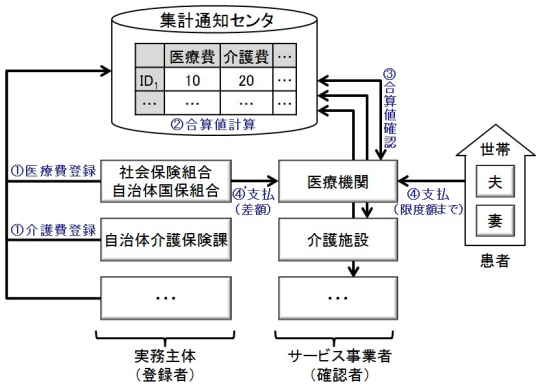

図 1 サービスモデル
Fig. 1 Service model.

本モデルにおけるステークホルダは，以下である．

- **実務主体（登録者）**：医療費制度や介護費等制度等の実務を行う者．医療費制度における社会保険組合や自治体の国保組合，介護費制度における自治体の介護保険課等がこれにあたる．現状，実務主体は，レセプト等の診療報酬明細書より，個人が支払った医療費や介護費等を把握しており，それらの情報は各々の実務主体によって管理されている．本サービスモデルでは，図 1① のように，実務主体が本人（患者）に代わって個人が支払った医療費や介護費等の情報を集計通知センタにオンラインで登録する．
- **集計通知センタ**：本サービスモデルにおいて新設するクラウド上のデータセンタ（仮）．各実務主体から送られてきた医療費や介護費等の個人支払情報を収集し，マイナンバー等の情報を基に世帯ごとの合算値を計算する（図 1②）．また，サービス事業者からの依頼に応じて，該当者の自己負担額の世帯合算値をサービス事業者に通知する．
- **サービス事業者（確認者）**：医療や介護等のサービスを行う医療機関や介護施設等．診療費を患者に窓口請求をする際，その患者の世帯合算値をインターネットを介して集計通知センタに確認し（図 1③），その限度額までの自己負担額を患者に窓口請求する．なお，本来の自己負担額と患者が支払う窓口請求額とで差が生じた場合，サービス事業者は，その差額を実務主体に請求し，実務主体の保険料より，その差額分がサービス事業者に支払われる（図 1④’）．
- **患者**：医療や介護等のサービスを受ける者．サービス事業者が算出した窓口請求額を支払う（図 1④）．

本サービスモデルでは，図 1 のように “複数” の実務主体（登録者）がデータを集計通知センタに登録し，“複数” のサービス事業者（確認者）が集計通知センタに問い合わせて該当者の世帯合算値を確認する，という特徴がある．

2.2 要件定義

本節では，前記サービスモデルをベースとして，実務主体，集計通知センタおよびサービス事業者に求められる要件を整理した．なお，ステークホルダのうち，患者に対しては，サービス事業者が算出した窓口請求額を支払うだけなので，ここでは要件を求めないこととする．

【要件 1】 実務主体は，個人が支払った医療費や介護費等の情報を集計通知センタに登録することが主な役割である．医療費や介護費の支払情報は，患者個人のプライバシ

© 2019 Information Processing Society of Japan

にかかわる情報であり，本人に代わって代理登録を行う実務主体は，それら個人情報を一貫して守る責務がある．

本サービスモデルでは，個人情報は，集計通知センタを中継して，サービス事業者にわたる．それゆえ，実務主体の直接的な管理から離れるとはいえ，実務主体は，それらの情報を部外者に漏らすことなく，一方で，権限のあるサービス事業者には，正しく閲覧できるようにしておく必要がある．

すなわち，本サービスモデルでは，医療費や介護費等の個人情報を実務主体からサービス事業者まで，End-to-Endで暗号化することが望ましい．

【要件 2】 集計通知センタは，世帯合算値を計算し，サービス事業者に通知することが主な役割である．集計通知センタ自体は，実務主体から送られてくる個々のデータの値や世帯ごとの合算値を知る必要はない．役所等の職員が興味本位に個人情報を不正閲覧するような事件が多発する状況 [11], [12] に鑑みると，知る必要のない情報は，たとえ集計通知センタの運用者や管理者でも，閲覧できない仕組みにしておくことが望ましい．

また, 2009 年に起きた米国クレジットカード漏洩事件 [13] では，加盟店との通信を SSL で暗号化していたにもかかわらず，カード会社のサーバ上で復号された瞬間をマルウェアにより搾取されクレジットカード情報が漏洩している．このようなマルウェア対策という観点からも，個人情報が集まる集計通知センタでは，個人情報を平文のまま管理せず，つねに暗号化して管理する等の安全対策を施しておくことが望ましい．

一方で，集計通知センタは，サービス事業者の要求に応じて，世帯合算値を計算しなければならない．それゆえ，本サービスモデルでは，暗号化状態のまま演算処理ができる必要がある．

【要件 3】 サービス事業者は，集計通知センタに該当者の世帯合算値を確認し，その限度額まで患者に窓口請求を行う．本サービスモデルでは，サービス事業者は複数存在する．しかし，十分な安全管理が行える事業者からそうでない事業者まで存在する．このようにガバナンスが効かないサービス事業者側に，過度なセキュリティ要件を求めることは，得策とはいえない．それゆえ，あるサービス事業者からの情報漏洩が，システム全体に波及しない仕組みにしておくことが肝要である．たとえば，全サービス事業者にシステム共通の秘密鍵を配布するような仕組みにしてはならない．

また，必ずしもすべてのサービス事業者が，プログラムソフトの導入・管理等のリソースを割けられるわけではない．

ゆえに，ブラウザのみで本サービスを利用できる等，ITリテラシーが低いサービス事業者でも容易にかつ安全に本サービスを使えるような仕組みにしておくことが望ましい．

3. 前提条件と攻撃者モデル

一般的にインターネット上の脅威として，データの盗聴，改竄，なりすまし等があげられるが，ここでは，本サービスモデルに対し，以下の状況を前提とする．

3.1 前提条件

(1) ステークホルダ間の通信路は，IPsec や SSL/TLS 等によって盗聴から守られている．
(2) 各々のステークホルダ間は，相互認証されていて，なりすましによる脅威からも守られている．
(3) Web ページ間の遷移は，正しくセッション管理されており，セッション乗っ取り等の対策が施されている．
(4) 実務主体の職員は，正しくデータを集計通知センタに登録する．
(5) 集計通知センタは，正しく合算値を計算をする．
(6) 集計通知センタの Web ページ（JavaScript 含む）は，改竄対策もされている．
(7) 集計通知センタの秘密鍵は，Hardware Security Module（HSM）等を用いて保護・管理されている．
(8) サービス事業者は，業務上必要な該当者の世帯合算値のみを取得する．

3.2 攻撃者モデル

本稿で想定する攻撃者は，実務主体の職員，集計通知センタの職員，サービス事業者の職員，患者および悪意のある第三者（マルウェア等を含む）とする．また，本稿で扱う脅威は，データの盗聴とし，その他の脅威は，上記前提の下，検討の対象外とする．また，攻撃対象は，新設される集計通知センタに管理されている個人情報とする．なお，実務主体やサービス事業者内で管理されている個人情報も盗聴の脅威にさらされているが，それらは，現状のサービス形態ででも起こりうる脅威である．当然，それらの対策も施さねばならないが，本稿では，集計通知センタの新設によって影響される脅威に主眼を置き，実務主体やサービス事業者への攻撃は，検討の対象外とする．

上記状況に鑑み，本稿では，攻撃者が故意・過失により集計通知センタ内の個人情報を不正閲覧することを脅威とし，その対策を講じる（**図 2**）．

4. 既存技術の適用上の課題

データの機密性を確保する一般的な方法として，データの暗号化がある．特にクラウド等のデータセンタを中継してデータが送受信される場合には，送信者から受信者までを End-to-End で暗号化をすることにより，データセンタ職員による不正閲覧やデータセンタに仕掛けられたマルウェアによる情報漏洩事故等を防ぐことができる．

一方，本サービスモデルでは，集計通知センタは，サー

© 2019 Information Processing Society of Japan

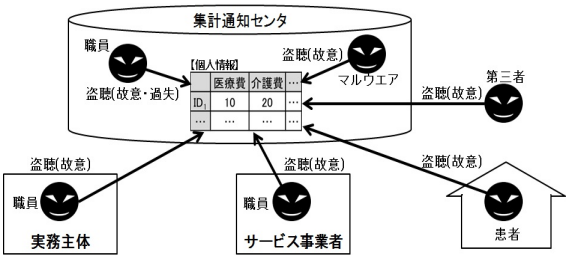

図 2 攻撃者モデル
Fig. 2 Attacker model.

ビス事業者の依頼に応じて，該当者の世帯合算値を計算する必要があり，End-to-End で個々のデータが暗号化されてしまうと，集計通知センタでは個々のデータから合算値を計算することが難しくなる．

上記課題を解決するために，本稿では，暗号化状態のまま算術演算をすることができる秘密計算と呼ばれる技術に着目する．暗号理論では古くから入力データを秘匿しつつ，正しい値を計算できる秘密計算の研究が進められており，加法準同型暗号に基づく Paillier 暗号 [1] や乗法準同型暗号に基づく RSA 暗号 [2] 等が知られている．また，最近では，Somewhat 準同型暗号 [3] や Craig Gentry による完全準同型暗号に基づく方式 [4] が提案されている．

本サービスモデルでは，自己負担額の世帯合算値を安全に計算することが主目的であるので，既存技術のうち，暗号化状態で加法を行うことができる Paillier 暗号 [1] をベースに説明するが，加法準同型の性質を持つ Somewhat 準同型暗号 [3] や完全準同型暗号 [4] 等を用いても実現できる．

4.1 Paillier 暗号

まず，準備として，加法準同型の性質を持つ Paillier 暗号 [1] について説明する．本稿では公開鍵 pk での Paillier 暗号化関数を $\mathtt{ENC}_{pk}()$ と表記し，秘密鍵 sk での Paillier 復号関数を $\mathtt{DEC}_{sk}()$ と表記する[*2]．

Paillier 暗号は Pascal Paillier が 1999 年に提案した公開鍵暗号方式であり，合成数剰余判定仮定の下で加法準同型性を持つ IND-CPA 安全な方式である．

Paillier 暗号では $n = p \cdot q$（p，q は素数）とする．また，$k \in Z_n^*$ を任意に選び，$g = 1 + k \cdot n \bmod n^2$ とする．そして，$pk = (n, g)$ を公開鍵とし，$sk = (p, q)$ を秘密鍵とする．

このとき，平文 m の暗号化は以下である．

$$c = \mathtt{ENC}_{pk}(m) = g^m \cdot r^n \bmod n^2.$$

なお，$r \in Z_n^*$ は任意の乱数である．

また，暗号文 c の復号化は以下である．

$$m = \mathtt{DEC}_{sk}(c) = \frac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n.$$

[*2] Somewhat 準同型暗号 [3] や完全準同型暗号 [4] 等を用いる場合，$\mathtt{ENC}_{pk}()$，$\mathtt{DEC}_{sk}()$ としてそれぞれ対応する暗号化関数，復号化関数を当てはめて扱えばよい．

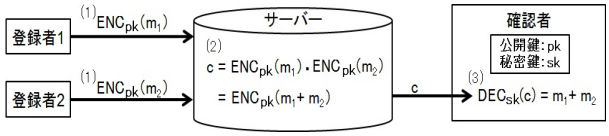

図 3 一般的な Paillier 利用モデル
Fig. 3 Typical usage model of Paillier cryptosystem.

なお，$\lambda = \mathrm{lcm}(p-1, q-1)$ であり，L は以下である．

$$L(x) = \frac{x-1}{n} \bmod n^2.$$

Paillier 暗号の特徴は，加法に関して準同型の性質を持つことであり，平文 m_1，m_2 に対して以下が成り立つ．

$$\mathtt{ENC}_{pk}(m_1) \cdot \mathtt{ENC}_{pk}(m_2) = \mathtt{ENC}_{pk}(m_1 + m_2),$$
$$\mathtt{ENC}_{pk}(m_1)^{m_2} = \mathtt{ENC}_{pk}(m_1 \cdot m_2).$$

一般的に Paillier 暗号は，**図 3** に示すように，(1) 複数の登録者が確認者の公開鍵を用いてデータを暗号化してサーバに送り，(2) サーバが秘密計算を行って暗号化計算結果を確認者に送り，(3) 確認者が自身の秘密鍵で復号して計算結果を取得する，という使い方をする．

4.2 Paillier 暗号の適用上の課題

前述のとおり Paillier 暗号は，加法に関して準同型の性質を持つ．それゆえ，Paillier 暗号の適用により，個々のデータがたとえ End-to-End で暗号化されていても，集計通知センタでは世帯合算値を計算することが可能になる．

しかし，Paillier 暗号は，図 3 に示すように，複数の登録者がデータを暗号化してサーバに登録することはできるが，その暗号化データを復号できるのは，秘密鍵を保有する確認者のみである．この性質は，Paillier 暗号だけの特徴ではなく，他の準同型暗号 [2], [3], [4] も持っている．

一方で本サービスモデルでは，登録者だけではなく，確認者も複数存在することが特徴である．しかし，要件 3 より，全確認者に同一の秘密鍵を配布することは，望ましくない．すなわち，本サービスモデルへの適用には，複数存在する確認者にシステム共通の秘密鍵を配布することなく，各確認者が暗号化計算結果を復号できる仕組みが必要である．

4.3 関連研究 1（準同型暗号の拡張）

一般的に準同型暗号 [1], [2], [3], [4] は，一組の鍵ペア（公開鍵，秘密鍵）を用いて暗号化・復号化を行う方式であり，先に指摘したように，ある公開鍵で暗号化したデータを異なる秘密鍵で復号することはできない．また，異なる公開鍵で暗号化したデータどうしを準同型演算することもできない．

これらの課題を解決するために準同型暗号の拡張に関する研究もされており，2012 年には，López-Alt らによって

© 2019 Information Processing Society of Japan

Multi-Key 完全準同型暗号 [5] と呼ばれる拡張方式が提案された．López-Alt らの方式は，主に後者の課題を解決する完全準同型暗号である．なお，López-Alt らの方式では，異なる公開鍵で暗号化したデータどうしの演算を可能とするが，1 度の演算で利用できる公開鍵の個数に制限がある．また，異なる公開鍵で暗号化したデータどうしの演算結果を復号する場合，該当公開鍵に対応するすべての秘密鍵を必要とし，それゆえ，それら秘密鍵の保有者が対話型マルチパーティプロトコルを用いて協力しながら演算結果を復号する，といった特徴がある．

一方，本稿での課題は，López-Alt らとは異なり，主に前者の方を扱う．本稿では，複数存在する確認者にシステム共通の秘密鍵を配布することなく，ある公開鍵で暗号化したデータを各確認者が復号できる仕組みを提案する．

5. 要件を満たすプロトコルの提案

5.1 ラフスケッチ

提案方式のポイントは，2 つある（図 4）．

1 つ目のポイントは，「集計通知センタ」を「集計センタ」と「通知センタ」の 2 つに分け，暗号化データの管理と秘密鍵の管理を分担させたことである．

2 つ目のポイントは，サービス事業者が合算値の計算を依頼する際，サービス事業者がマスク値（乱数）を動的に生成し，マスク値を知っているサービス事業者のみが真の合算値を取得できるようにしたことである．

次節で，提案方式の詳細を説明する．

5.2 提案方式の詳細

本方式は，鍵生成フェーズ，データ登録フェーズ，合算値確認フェーズの 3 フェーズに分かれる．以下，フェーズごとに詳細を説明する．

【鍵生成フェーズ】

本方式では，2 つに分けたセンタのうち，通知センタ A が，Paillier 暗号における公開鍵 pk_A と秘密鍵 sk_A を生成し，公開鍵 pk_A を実務主体 R およびサービス事業者 V に公開する．なお，秘密鍵 sk_A は，FIPS140-2 Level 3 [15] 相当の Hardware Security Module（HSM）等を用いて安全に管理され，各攻撃者から守られているとする[*3]．

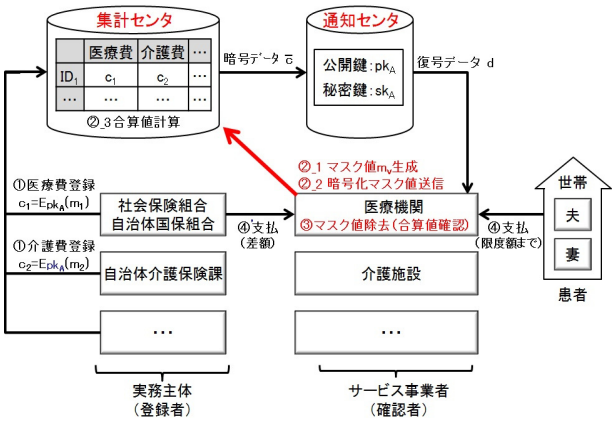

図 4 提案方式概略図

Fig. 4 Outline of the proposed method.

【データ登録フェーズ】

実務主体 R は，支払額データ $m \in Z_n^*$ に対し，通知センタ A の公開鍵 pk_A で支払額データ m を Paillier 暗号を用いて暗号化する．

$$c = \mathrm{ENC}_{pk_A}(m).$$

そして実務主体 R は，暗号化データ c が何に対するデータであるかの属性情報（たとえば “ID_1” の “医療費” のデータであることを示す情報）を添えて，暗号化データ c を集計センタ S に送信し，集計センタ S は，指定された属性情報に従い，暗号化データ c を DB に登録する．

なお，実務主体 R と集計センタ S との通信は，IPsec や SSL/TLS により通信路は暗号化され，相互認証もされているとする．また，実務主体の職員は，正しいデータを登録し，暗号化データには改竄対策も施されているとする．

【合算値確認フェーズ】

サービス事業者 V と集計センタ S と通知センタ A との通信路も暗号化され，かつ相互に認証されているとする．

また，サービス事業者 V と集計センタ S と通知センタ A との間は，正しくセッション管理されているとする．

合算値確認フェーズは，さらに細かく，「演算依頼処理」「秘密計算処理」「合算値取得処理」からなる（図 5）．

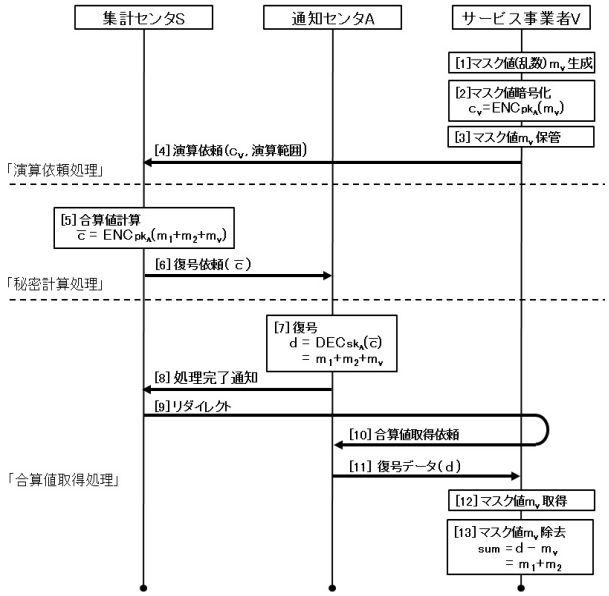

図 5 合算値確認フェーズ

Fig. 5 Total value confirmation phase.

[*3] HSM を用いても通知センタ職員内の HSM 管理者のみは，秘密鍵に直接アクセスすることができる．それゆえ，1 人の管理者に権限を集中させない運用（職務分掌・複数人操作）を行う等，HSM 管理者による不正対策も別途実施されているとする．

© 2019 Information Processing Society of Japan

「演算依頼処理」

サービス事業者 V は，演算範囲（たとえば “ID_1” の “医療費” と “介護費” の “和” を知りたい旨を示す情報）を指定して，集計センタ S に計算依頼をする．このとき，サービス事業者 V は，マスク値（乱数）$m_v \in Z_n^*$ を任意に選び，そのマスク値 m_v を通知センタ A の公開鍵 pk_A で Paillier 暗号を用いて暗号化する．

$$c_v = \mathtt{ENC}_{pk_A}(m_v)$$

そして，マスク値 m_v をローカルに保存しつつ，暗号化マスク値 c_v を演算範囲とともに集計センタ S に送信する．

「秘密計算処理」

集計センタ S は，指定されている演算範囲を基に該当する暗号化データを選択する．ここでは，ID_1 の医療費に対する暗号化データを “$c_1 = \mathtt{ENC}_{pk_A}(m_1)$” とし，$ID_1$ の介護費に対する暗号化データを “$c_2 = \mathtt{ENC}_{pk_A}(m_2)$” として説明する．

まず，集計センタ S は，要求されている指示に従い暗号化状態で演算を行う．このとき暗号化データは以下になる．

$$\begin{aligned} c_1 \cdot c_2 &= \mathtt{ENC}_{pk_A}(m_1) \cdot \mathtt{ENC}_{pk_A}(m_2) \\ &= \mathtt{ENC}_{pk_A}(m_1 + m_2). \end{aligned}$$

また，集計センタ S はサービス事業者 V から送られてきた暗号化マスク値 c_v を用いて，先の演算結果に対してさらに加法演算を行う．このとき，暗号化計算結果は以下になる．なお，集計センタ S は，下記計算を正しく行うとする．

$$\begin{aligned} \bar{c} &= (c_1 \cdot c_2) \cdot c_v \\ &= \mathtt{ENC}_{pk_A}(m_1 + m_2) \cdot \mathtt{ENC}_{pk_A}(m_v) \\ &= \mathtt{ENC}_{pk_A}(m_1 + m_2 + m_v). \end{aligned}$$

集計センタ S は，暗号化計算結果 $\bar{c}$ を通知センタ A に送り，通知センタ A に暗号化計算結果の復号依頼を行う．

「合算値取得処理」

通知センタ A は，集計センタ S から送られていた暗号化計算結果 $\bar{c}$ を自身の秘密鍵 sk_A を用いて Paillier 暗号の復号処理を行う．

$$\begin{aligned} d &= \mathtt{DEC}_{sk_A}(\bar{c}) \\ &= \mathtt{DEC}_{sk_A}(\mathtt{ENC}_{pk_A}(m_1 + m_2 + m_v)) \\ &= m_1 + m_2 + m_v. \end{aligned}$$

そして通知センタ A は，復号処理が完了した旨を集計センタ S に通知し，完了通知を受信した集計センタ S では，サービス事業者 V を通知センタ A の合算値取得ページにリダイレクトさせる．

通知センタ A のシステムにリダイレクトされたサービス事業者 V は，通知センタ A から復号データ d を取得する．

そしてサービス事業者 V は，ローカルに保管していたマスク値 m_v を取得し，復号データ d からマスク値 m_v を引いて求める合算値 sum を取得する．なお，マスク値 m_v は，合算値 sum 取得後に消去する．

$$\begin{aligned} sum &= d - m_v \\ &= (m_1 + m_2 + m_v) - m_v \\ &= m_1 + m_2. \end{aligned}$$

5.3 適用モデルの整理

ここでは，本方式の適用モデルを整理する．

本方式は，実務主体 R が登録した暗号化データを「集計センタ S」,「通知センタ A」,「サービス事業者 V」の 3 者が協力して処理を行う．なお，この 3 者が協力する本方式の適用モデルは，複数考えられる．

本稿では，サービス事業者 V を起点とし，集計センタ S で計算した暗号化計算結果が通知センタ A を経由してサービス事業者 V に戻るモデルを検討対象として**図 6** に示す 3 つの適用モデルを検討した．図中の山括弧内の数字は，経路の順序を示している．

なお，5.2 節の提案方式のベースは，適用モデル 3 である．提案方式をブラウザ上で実装する場合，⟨1⟩ で集計センタ S にアクセスしているサービス事業者 V に対し，⟨3⟩ で通知センタ A から復号データ d をサービス事業者 V に直接配信するのは困難である．そこで本方式の適用では，**図 7** の適用モデル 3’ のようにサービス事業者 V を通知センタ A にリダイレクトさせ，サービス事業者 V が復号データ d を通知センタ A から取りに行くことで実現する．

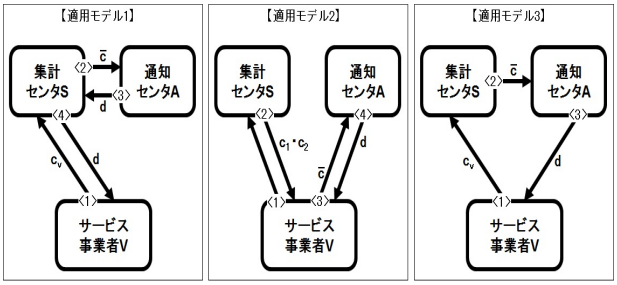

図 6 適用モデル
Fig. 6 Usage model.

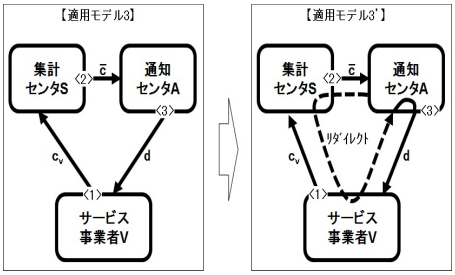

図 7 適用モデル 3 の実装方法
Fig. 7 Implementation method of Model 3.

© 2019 Information Processing Society of Japan

以後，本稿では，適用モデル 3 を適用モデル 3' として検討する．

まず適用モデル 1 についてである．適用モデル 1 では，⟨3⟩ で集計センタ S は，通知センタ A から復号データ d を受け取っている．この場合，もし，集計センタ S が ⟨2⟩ でマスク値を入れずに（あるいは任意のマスク値を入れて）通知センタ A に $\bar{c}$ の復号依頼をすると，集計センタ S は，復号データ d から求める合算値を得ることができる．この攻撃は，復号データ d を集計センタ S が受け取れるがゆえに成立する．したがって本方式の適用では，集計センタ S には，復号データ d を取得する権限（復号データ取得権限）を与えず，サービス事業者 V のみにその権限を与えるべきである．

次に適用モデル 2 についてである．適用モデル 2 では，⟨2⟩ で集計センタ S は，$c_1 \cdot c_2 = E(m_1 + m_2)$ をサービス事業者 V に送り，⟨3⟩ でサービス事業者 V 自身で暗号化マスク値 c_v を加算した $\bar{c} = (c_1 \cdot c_2) \cdot c_v = E(m_1 + m_2 + m_v)$ を計算して復号権限を有する通知センタ A に復号を依頼する．この場合，もし，あるサービス事業者 V が集計センタ S に仕掛けたマルウェア等により，集計センタ S で管理されている各暗号化データ $c = E(m)$ を取得できたとすると，そのサービス事業者 V は，通知センタ A にそれら暗号化データ c の復号を頼むことで，合算前の個々の個人情報（元データ）を不正に得ることができる．

これは，サービス事業者 V から送られてくる暗号化データ $\bar{c}$ が「集計センタ S からの暗号化合算データ $c_1 \cdot c_2$ にサービス事業者 V が c_v を加算した，正規手順に従った暗号化データ $\bar{c}$」であるのか，それとも「サービス事業者 V が正規手順に従わずに不正取得した暗号化データ c」であるのか，通知センタ A が判断できないために生じる．それゆえ，本方式の適用では，サービス事業者 V には，通知センタ A に暗号化データ $\bar{c}$ の復号を依頼できる権限（復号依頼権限）を与えず，集計センタ S のみにその権限を与えるべきである．

適用モデル 1 および適用モデル 2 の分析から，本方式の適用には，復号依頼権限，復号権限，復号データ取得権限をそれぞれ集計センタ S，通知センタ A，サービス事業者 V に分けることが重要と分かる．

最後に適用モデル 3' についてである．適用モデル 3' は，復号依頼権限，復号権限，復号データ取得権限をそれぞれ集計センタ S，通知センタ A，サービス事業者 V に分け与えている．

なお，適用モデル 3' でも適用モデル 1 での攻撃方法を用いて，集計センタ S が，マスク値を入れずに（あるいは任意のマスク値を入れて）通知センタ A に $\bar{c}$ の復号依頼をし，通知センタ A からのリダイレクト先を自分で参照することで同様の攻撃が可能であるが，通知センタ A は，復号データ d を取得できる者を該当サービス事業者 V のみに制限し，集計センタ S によるリダイレクト先データの参照を通知センタ A が拒否することで，集計センタ S に復号データ d をわたさない仕組みにすることができる．したがって適用モデル 3' では，前提条件 2 である各ステークホルダ間の相互認証を正しく実行し，権限に応じたアクセス制御をすることが重要である．

また，適用モデル 3' では，集計センタ S が通知センタ A に復号依頼を行う．ゆえに，万が一，集計センタ S で管理されている暗号化データ c が流出した場合でも，通知センタ A では，集計センタ S 以外からの復号依頼を拒否する仕組みにすることで，攻撃者が不正に暗号化データ c を取得した場合でも個人情報（元データ）を保護することができる．

5.4 関連研究 2（乱数によるマスク化手法）

本方式で行われている Paillier 暗号化データに対する乱数によるマスク処理は，Pathak らの手法 [6] でも用いられている．

Pathak らの手法 [6] では，秘密鍵を管理する Alice と，Alice に値（平文）を知らせずに演算結果を取得したい Bob との 2 者間で乱数によるマスク処理が用いられている[*4]．

Pathak らの手法 [6] は，Alice と Bob の 2 者間でのモデルであるのに対し，本稿は，集計センタ S，通知センタ A，サービス事業者 V の 3 者間モデルであり，乱数によるマスク化手法の 3 者間モデルへの適用方法を提案したものである．

なお，Pathak らの 2 者間モデルの手法 [6] を本稿の現物給付型サービスに直接適用することはできない．なぜなら Pathak らの手法 [6] を直接適用する場合，秘密鍵を管理している通知センタ A が Alice に該当し，演算結果を取得するサービス事業者 V が Bob に該当する．この場合，集計センタ S が暗号化合算値 $c_1 \cdot c_2$ をサービス事業者 V である Bob に送り，Bob が暗号化合算値に乱数でマスクした値 $\bar{c}$ を計算し，通知センタ A である Alice に $\bar{c}$ の復号を依頼することになる．これは，適用モデル 2 の型であり，サービス事業者 V に復号依頼権限を与えることになる．したがって，Pathak らの手法 [6] を直接適用することはできない．

現物給付型サービスへの適用には，本稿で示す 3 者間モデルを採用し，復号依頼権限，復号権限，復号データ取得権限を 3 者に分け与えることが重要である．

5.5 考察

本稿で想定する攻撃者は，集計センタおよび通知センタの職員，実務主体者，サービス事業者，患者および悪意のある第三者（マルウェア等を含む）であり，攻撃対象およ

[*4] 正確には，Pathak らは，隠れマルコフモデルの処理を暗号化状態で実行する方法を提案しており，その処理途中で 2 者間での乱数によるマスク処理が用いられている．

© 2019 Information Processing Society of Japan

表 1 ソース概要（JavaScript）
Table 1 Source code (JavaScript).

演算依頼処理		
[1] マスク値（乱数）生成	:	$m_v \leftarrow$ **window.crypto.getRandomValues();**
[2] マスク値暗号化	:	$c_v \leftarrow g$**.modPow(**m_v, n^2**).multiply(**r**.modPow(**n, n^2**)).mod(**n^2**);**
[3] マスク値 m_v 保存	:	**localStorage.setItem("key",** m_v**);**
[4] 演算依頼	:	**POST** c_v**;**

※ パラメータ g，n，r は，4.1 節参照

合算値取得処理		
[12] マスク値 m_v 取得	:	$m_v \leftarrow$ **bigInt(localStorage.getItem("key"));**
[13] マスク値 m_v 除去	:	$sum \leftarrow d$**.subtract(**m_v**);**

※ パラメータ sum，d は，5.2 節参照

び脅威は，攻撃者が故意・過失により集計センタ内の個人情報を不正閲覧することである．

提案方式において，新設された集計センタでは，加法演算が可能な Paillier 暗号で暗号化されたデータのみしか扱っていない．それゆえ，Paillier 暗号が安全と仮定すると，たとえ集計センタの職員であっても攻撃者は，集計センタから個人情報を不正閲覧することはできない．

また，新設されるもう 1 つのセンタである通知センタには，マスク値が加算された暗号化データが集計センタから送られてくる．通知センタは，暗号化データを管理していないので，個人情報に関するデータは，集計センタから送られてくるマスク値付暗号化データのみである．

通知センタは，Paillier 暗号の秘密鍵を管理しているので，そのマスク値付暗号化データを復号することができるが，復号した値は，マスク値が加算されたワンタイムパッドになっている．なお，秘密鍵は，Hardware Security Module（HSM）等を用いて安全に管理されているとする．

Paillier 暗号で暗号化したマスク値は安全と仮定すると，パッドに用いられたマスク値は，マスク値を生成したサービス事業者のローカル環境のみに存在する．

それゆえ，攻撃者は，サービス事業者のローカル環境のみに存在するマスク値を知らない限り，通知センタから正しい個人情報を取得することはできない．

つまり提案方式では，実務主体者が暗号化した個人情報は，権限のあるサービス事業者まで 1 度も平文に戻ることなく，End-to-End で暗号化されたままで処理される．すなわち要件 1，2 を満たす．

もちろん，本方式では，暗号化データを管理している集計センタと秘密鍵を管理している通知センタが結託すると，暗号化データの機密性を保てなくなる．しかし，独立行政法人情報処理推進機構（IPA）“組織における内部不正防止ガイドライン [14]” によると，内部不正防止対策として「やりにくくする」「やると見つかる」「割に合わない」「その気にさせない」「言い訳させない」ことが基本原則とされており，結託を試みる場合，それは同時に相手センタから不正を告発されるリスクが高まることにつながる．それゆえ，本方式のようにセンタを分けることにより，センタ職員による内部不正に対して「やりにくくする」「やると見つかる」という効果が期待できる．

また提案方式は，サービス事業者側には，管理が煩雑となる秘密鍵を使わずに，集計センタの公開鍵 pk_A と使い捨てのマスク値 m_v による簡単な処理のみで暗復号を実現していることも特徴である．

すなわちサービス事業者側では，公開情報と使い捨て値による簡単な処理しか行わないので，JavaScript のみでも容易に実装が可能である．

ガバナンスが効かないサービス事業者側に，過度なセキュリティ要件を求めることは，得策ではない．提案方式では，サービス事業者側は，JavaScript によってブラウザのみで実装することができ，これにより専用ソフトウェアを配布することもなく，IT リテラシの低いサービス事業者でも容易にかつ安全に利用することが可能となる．これは，要件 3 を満たす．

参考までに**表 1** に JavaScript のみで実装したサービス事業者側のソースの概要を記す．

本実装では，`window.crypto.getRandomValues()` を用いてマスク値 m_v を生成した．また，生成したマスク値 m_v は，`localStorage.setItem()` で Web ブラウザ（ローカル環境）に保存し，`localStorage.getItem()` で保存先から取得することができる．

数値演算は，BigInteger.js [16] を用いて，JavaScript 上での多倍長演算を行った．

いずれの処理もブラウザのバージョン等に依存することもあるが，表 1 に示すように，容易に実装することができる．

6. まとめ

本稿では，高額医療・高額介護合算療養費制度等における現物給付型サービスの仕組みを電子的に実現する方法について，サービスのモデル化を行い，そのサービスモデルに基づいて要件定義をした．また，その要件をすべて満たす秘密計算プロトコルを提案した．

© 2019 Information Processing Society of Japan

提案方式では，実務主体者（登録者）が暗号化した個人情報は，権限のあるサービス事業者（確認者）まで1度も平文に戻ることはなく，End-to-End で暗号化されたまま演算処理される．

また，本方式においてサービス事業者側では，公開情報と使い捨て値による簡単な処理しか行わないので，JavaScript のみでも容易に実装が可能であり，IT リテラシの低いサービス事業者でも容易にかつ安全に利用することができる．

本方式は，現物給付型サービスの電子化モデルにおいて新設される集計センタおよび通知センタからの情報漏洩対策技術であり，3.1 節の前提条件の下で成り立つ方式である．

たとえば，通知センタが，通信路上で実務主体が登録する暗号化された支払データ c やサービス事業者が送信する暗号化マスク値 c_v を得た場合，通知センタは，Paillier 暗号の秘密鍵を管理しているため，それら暗号化データから元のデータを復元できてしまう．それゆえ，通知センタに通信路上に流れる暗号化データを取得させないために，ステークホルダー間では，暗号化された通信路が必要である（前提条件 1）．なお，通知センタとの通信路となる「収集センタ–通知センタ間」および「サービス事業者–通知センタ間」は，必ずしも通信路暗号が必要なわけではない．

また，本方式の適用では，復号依頼権限，復号権限，復号データ取得権限をそれぞれ集計センタ，通知センタ，サービス事業者に分け与えるため，各権限に応じて互いを正しく認証することが重要であり（前提条件 2）[*5]，認証後のセッション管理も疎かにしてはならない（前提条件 3）．

なお，具体的な認証方法については，本稿の対象外とするが，本稿では，特にサービス事業者には，IT リテラシーが低い者もいることを前提にしているので，安全性を考慮しつつ，IT リテラシーが低い者でも利用可能な認証方法を用いるのが望ましい．たとえば，インターネットバンキング等でも多く利用されている ID/PW のような簡易な方式や，初期登録が比較的重い傾向があるが利用時の操作が比較的容易な FIDO [17] 等の生体認証を用いることも考えうる．

また，実務主体が誤ったデータを登録したり，集計センタが誤った演算処理をしたりすると，そもそも正しい合算値をサービス事業者に提供することができなくなる（前提条件 4，5）．これらは，サービスの信頼性の問題であり，チェック機能の強化等の対策が考えられる．

また，正規 Web サイト（JavaScript 含む）が改竄されていては，サービス事業者は，正しい合算値を得られない．それゆえ，各センタは，Web サーバをつねに監視する等，Web ページの改竄対策も施す必要がある（前提条件 6）．

また，通知センタでは，Paillier 暗号の秘密鍵を管理している．秘密鍵が漏洩するとシステム全体のセキュリティを保てなくなる．それゆえ，秘密鍵は，FIPS140-2 Level 3 [15] 相当の HSM 製品を用いる等，別途，安全管理を講じる必要がある（前提条件 7）．

また，サービス事業者が業務上必要のない患者の世帯合算値を要求した場合，サービス事業者は，個人情報を不正に取得したことになる．それゆえ，サービス事業者からの演算依頼に関するログデータを監査する仕組みを導入する等，サービス事業者の不正対策も必要である（前提条件 8）．

このようにシステム全体のセキュリティ設計には，前提となっている改竄対策やなりすまし対策等も当然必要であり，提案方式の手法も含め，複数の技術的対策や運用的対策を組み合わせて網羅的かつ効果的に設計することが重要である．

[*5] もちろん，登録権限のない者による愉快犯的なデータ登録を防ぐため，実務主体に対する認証も必要である．

参考文献

[1] Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, *EUROCRYPT '99*, Lecture Notes in Computer Science, Vol.1592, pp.223–238 (2009).

[2] Rivest, R.L., Shamir, A. and Adleman, L.: A method for obtaining digital signatures and public key cryptosystems, *Comm. ACM*, Vol.21, No.2, pp.120–126 (1978).

[3] van Dijk, M., Gentry, C., Halevi, S. and Vaikuntanathan, V.: Fully homomorphic encryption over the integers, *EUROCRYPT2010*, LNCS 6110, pp.24–43, Springer (2010).

[4] Gentry, C.: Fully homomorphic encryption using ideal lattices, *STOC2009*, pp.169–178 (2009).

[5] López-Alt, A., Tromer, E. and Vaikuntanathan, V.: On-the-Fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption, *STOC '12*, *Proc. 44th Annual ACM Symposium on Theory of Computing*, pp.1219–1234 (2012).

[6] Pathak, M., Rane, S., Sun, W. and Raj, B.: Privacy Preserving Probabilistic Inference with Hidden Markov Models, *ICASSP2011*, *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp.5868–5871 (2011).

[7] 総合合算制度の導入：厚生労働省，入手先 ⟨http://www.mhlw.go.jp/stf/shingi/2r985200000297nt-att/2r98520000029af4.pdf⟩（参照 2019-01）．

[8] マイナンバー制度：総務省，入手先 ⟨http://www.soumu.go.jp/kojinbango_card/01.html⟩（参照 2019-01）．

[9] 2017 民進党政権公約原案（参考資料），入手先 ⟨https://www.minshin.or.jp/download/37228.pdf⟩（参照 2019-01）．

[10] 希望の党：日本経済新聞，入手先 ⟨https://www.nikkei.com/edit/2017shuin/pdf/kibou.pdf⟩（参照 2019-01）．

[11] 大阪市の戸籍不正アクセスは 62 人，上司含め 188 処分外部流出はなし，入手先 ⟨http://www.sankei.com/west/news/150312/wst1503120040-n1.html⟩（参照 2019-01）．

[12] 報告：住民基本台帳情報等の目的外利用について，入手先 ⟨www.city.kobe.lg.jp/information/public/hogo/img/400300.pdf⟩（参照 2019-01）．

[13] Credit Card Processor Says Some Data Was Stolen, New York Times, January 20, 2009, available from ⟨http://www.nytimes.com/2009/01/21/technology/21breach.html?_r=3&ref=technology&⟩ (accessed 2019-01).

© 2019 Information Processing Society of Japan

[14] 組織における内部不正防止ガイドライン：IPA 独立行政法人情報処理推進機構，2017.1, 入手先 ⟨https://www.ipa.go.jp/files/000057060.pdf⟩（参照 2019-01）.
[15] FIPS PUB 140-2: NIST, available from ⟨http://csrc.nist.gov/groups/STM/cmvp/standards.html⟩ (accessed 2019-01).
[16] BigInteger.js: GitHub, available from ⟨https://github.com/peterolson/BigInteger.js/⟩ (accessed 2019-01).
[17] FIDO2 Project: FIDO Alliance, available from ⟨https://fidoalliance.org/fido2/⟩ (accessed 2019-01).

坂崎 尚生 （正会員）

1971 年生．1994 年金沢大学理学部数学科卒業．1996 年同大学大学院修士課程修了．1999 年北陸先端科学技術大学院大学情報科学研究科博士後期課程修了．博士（情報科学）．同年（株）日立製作所システム開発研究所（現，研究開発グループシステムイノベーションセンタ）入社．セキュリティ・プライバシに関する研究開発に従事．

© 2019 Information Processing Society of Japan