ブロックチェーン技術とプロキシ再暗号を用いた医療データ管理システムに関する研究

ブロックチェーン技術とプロキシ再暗号を用いた医療データ管理システムに関する研究

池田　良磨 [a)]・岡崎　直宣 [b)]・山場　久昭 [c)]・油田　健太郎 [d)]

Medical Data Management System Using Blockchain and Proxy Re-Encryption

Ryoma IKEDA, Naonobu OKAZAKI, Hisaaki YAMABA, Kentaro ABURADA

Abstract

In Japan, our medical data is distributed and managed. Those data will not to be using when visiting medical institution. Since it is important to know the patient's past medical history in medical case, it is necessary to be able to disclose those data at the medical institution where patient is consulted. So, we propose a method that allows patients to autonomously manage medical data and share it with medical institutions. By managing medical data by patients, it is possible to disclose various data such as healthcare information acquired by the patient, history of medical practices received before, and medicines taken by patients to medical institutions trusted by patients. It also has the advantage of encouraging patients to participate in research. However, managing medical data using a general centralized network concentrates costs on the administrator. In addition, it is difficult to obtain patient consensus because all authority is delegated to the administrator. Therefore, in this research, we used a blockchain that realized a decentralized network. There are some precedents for medical data management using blockchain, and there are issues regarding encryption and permission. In this study, encryption and permission were realized by using proxy re-encryption with attack resistance. In the evaluation experiment, from the viewpoint of usability, the effect of proxy re-encryption used in this study on transactions was investigated by focusing on the execution time. From the results, it was confirmed that the cryptographic method used in this study does not have a fatal effect on transactions.

Keywords: Blockchain, Proxy re-encryption, Medical data management

1. はじめに

医療業界において昨今ではデータ管理が紙ベースのカルテから電子カルテに発展するなど、IT 化が進んでいる。またこれに伴い医療行為の標準化や業務の効率化、医療データの 2 次利用が進んでいる。特に患者データの 2 次利用に関して、日本ではかけはし研究班による WeAreHere[1)] や J-RARE[2)] 等希少疾患の患者を対象とした患者主体の情報登録プラットフォームが登場し、疾患にあたっての原因究明や新薬開発のような研究活動に利用できる体制が整備されはじめている。こうした観点から患者データは患者が主体となって管理運用し、自らの医療データから情報提供することでその対価を得る体制を整えるべきである、とした動きも現れている [3)]。

日本において医療データに関する問題として個人データの分散化が挙げられる。一生のうちに行う健康診断や受診記録など個人データがあらゆる機関で分散し同一人物のデータの収集が困難である。また複数の団体に所属する者であれば同一期間内に同じ内容の健康診断を複数回受ける場合もあり非効率な状況も見受けられる。医師は罹患した患者のこれまでの病歴や遺伝子情報などを考慮したうえで治療行為を行う必要があるため、治療を行う前に調査が必要になる。また医療行為の委託を他の医療機関に行うに当たって紹介状の作成やこれまでの医療履歴を作成して送付する必要がある。これら操作は医師に対して負担がかかる。

医療データ管理に関する問題点から本研究では医療データを患者個人が自律的に管理し、分散している個人データを容易に収集可能にすることを目的とする。ここで、自律的とは自身で取得したヘルスケア情報や過去に受けた健康診断結果、医療履歴といった情報を情報の所有者が自ら管理し、自身の判断で主治医に提供したり紹介先の医療機関へ開示することができることと定義する。医療データを個人で収集することができれば紹介先の医療機関への情報開示により医師の委託業務を軽減することができると考える。また希少疾患に関する研究でもデータの収集が容易になると考える。希少疾患の実態を把握するためには、患者情報を集約する必要があり、希少疾患に関した患者レジストリが構築されれば疾患及びその疾患に罹患した患者の実態を明らかにでき、研究開発を促進することができる [4)]。またこれは患者の研究参加を促すことにも期待できる。患者の医療研究への参加は、研究開発をす

[a)] 工学専攻機械・情報系コース大学院生
[b)] 情報システム工学科教授
[c)] 情報システム工学科助教
[d)] 情報システム工学科准教授

すめるうえでの新たな視点と価値を獲得でき、患者の不安・疑問点の解消や医療に対する理解など様々な恩恵がある[5]。個人データが作成された時点で作成された医療機関で保存され他の医療機関においてもアクセスが可能となる医療データ共有プラットフォームについて考える。このとき分散化された医療データは一か所に収集することができる。

本研究では、ブロックチェーン技術を利用した医療データ管理システムのモデルを提案する。提案にあたって患者データは患者自身が自律的に管理を行うことを想定する。医療データをオフチェーン上で管理し医療データに対するアクセス権限をブロックチェーン技術を用いて管理する。アクセス権限は文献[6]においてブロックチェーンのスマートコントラクト上にプロキシ機能を実装しプロキシ再暗号を用いてデータの暗号化とアクセスコントロールを同時に実現した手法を参考に、プロキシ再暗号を用いた手法で制御する。ここで、電子カルテ等を含む医療データは患者の個人情報や個人に関する重要な情報を含んでいることから医療データ管理システムは真正性、見読性、保存性に加え診療録等の個人情報を電気通信回線で伝送する間の個人情報の保護が求められる[7]。本研究では結託攻撃と CCA(Chosen Ciphertext Attack) に耐性をもつプロキシ再暗号を採用して実装を行った。

評価実験では実装したモデルのトランザクションに関して実行時間を計測した。

以下に、本論文の構成を示す。2. 章でブロックチェーンを用いた医療データ管理に関する研究を紹介する。3. 章では、本研究において関連する要素技術について解説する。4. 章では、本研究における提案モデルについて説明する。5. 章では、提案システムに関する評価実験について説明し、その実用性について議論する。最後に 6. 章でまとめと今後の課題、展望について述べる。

2. 関連研究

本章では、関連研究について説明する。最初に、ブロックチェーン技術を活用した医療データ管理手法の先行研究について紹介する。最後に、紹介した文献と本研究との差異について述べる。

2.1 先行研究

2.1.1 遺伝アルゴリズムと離散ウェーブレットの活用

Hussein らの研究[8]では、遺伝アルゴリズムと離散ウェーブレット変換を用いたブロックチェーンベースの電子カルテ管理手法を提案している。Hussein らはブロックチェーンに記録するログからユーザの履歴を改ざん不可能な形で実現でき、分散台帳に患者のデータが保存されることによって医者の患者に対する医療の説明責任を果たすことを可能にすると考えブロックチェーンを使用している。離散ウェーブレット変換を用いてユーザ鍵を生成することでセキュアな鍵管理を実現している。また遺伝アルゴリズムを用いて生成されたブロックに含まれるトランザクションの検証を行いブロックチェーン上での処理時間を既存の手法よりも高い性能で抑えている。医療データをオフチェーンで管理していることもあり、よりスケーラビリティ、堅牢性、攻撃耐性を担保したシステムの実装を行った。

2.1.2 Medical data management with privacy

Tian らの研究[9]ではユーザのリボケーションが生じた際でも正当なユーザが再構築可能な共有鍵を確立するブロックチェーンベースの医療データ管理を提案している。診療及び治療プロセスといったデータを暗号化し共有鍵を用いてブロックチェーンに保存する。共有鍵は SIFF(Sibling Intractable Function Families)[10] を用いて確立している。SIFF は文字列の 2 つのセットが衝突する場合、ほかの衝突する文字列を見つけることが計算上不可能である性質をもった暗号化の概念である。SIFF を用て共有鍵を生成することでデータの可用性とプライバシー要件を満たし、ブロックチェーンに医療データを保存することで整合性を満たしている。評価実験では通信コスト、処理時間のコストを計測し、そのシステムの効率を示している。

2.1.3 analyzing the performance of blockcahin-based PHR

Roehrs らの研究[11]では散在している PHR(Personal Health Record) や EHR(Electronic Health Record) の統合する PHR モデルの実装を行っている。ブロックチェーン技術と open EHR 相互運用性を用いて実装したプロトタイプに関して、レコードの統一された一覧を評価することに加え、応答時間、CPU 使用量、メモリ独有率、ディスク、ネットワーク使用量のパフォーマンス要件について性能評価を行っている。

2.1.4 クラウドストレージに保存する手法

Duboviskaya らは患者の医療情報をクラウドストレージに暗号化を施した上で保存し、そのメタデータとアクセスコントロールポリシーをプライベート型ブロックチェーンである hyperledger を用いて記録するシステムを提案した[12]。Duboviskaya らは複数の医療機関を含めた医療データの 1 次利用、研究者や解析者を含めた医療データの 2 次利用、保険会社や薬局といった非医療機関を含めた EHR(Electoric Health Record) 的利用の 3 つのシナリオにおいてブロックチェーンを電子カルテに適応した際の真正性や透明性を検証した。また、放射線治療を行っているがん患者に対してノード数が 4 つの小規模なブロックチェーンネットワークを構築し、患者の医療情報のアップロードや変更、読み取りといったトランザクションがセキュリティやプライバシーを考慮したうえで機能することを実験を通して確認した。

2.1.5 プロキシ再暗号の活用

文献[6]で萱原らは、プロキシ再暗号を用いて医療データの共有範囲を決定できるユーザのアクセス権限を付与し、ブロックチェーン上に医療データに加え記録している。萱原らはプロキシ再暗号の機能をスマートコントラクト上で実装することでプロキシサーバを別途用意する必要性をなくしているとともに、プロキシ再暗号には BBS 暗号[13]を採用し、暗号化及び復号における処理時間と実装した hyperledger 上での処理時間の評価を行い実用性を示している。しかしより精密なセキュリティ評価の必要性とスケーラビリティやユーザのリボケーションなどの課題を残している。

2.2 本研究との差異

関連研究としてブロックチェーンを活用した医療データの管理を提案している文献を挙げた。本研究では実際の医療データをブロックチェーン上には保存せず、医療機関のサーバなどオフチェーンでの管理を行うことを想定している。スケーラビリティの観点から大容量のデータの共有はブロックチェーンに不向きであることからオフチェーン上で管理を行っている。文献 [6]、[9] はブロックチェーン上で医療データを共有している為本研究とは異なる。文献 [9] では暗号化したデータに対してその共有鍵を少数人のグループで共有するが、医療データが増加すれば患者の管理する秘密鍵も増えることになる。本研究では医療データの量によらず患者が管理する秘密鍵は 1 人あたり 1 つである。またブロックチェーンの医療応用において、様々な課題 [3] が挙げられるが本研究ではアクセス権限、暗号化に着目して研究を行っている。文献 [11] では収集し統合することに着目しているため本質的に異なる。文献 [8] では患者データに対してのアクセス権限に関して、共有しているグループ内でのタスクを経て新たにユーザを共有グループに入れる手法を用いている。本研究では患者のデータは患者の決定により提供を行う事を想定しているため文献 [8] とは異なる。文献 [12] では医療データを共通鍵暗号で暗号化し、暗号化に使用した共通鍵を共有しているがリボケーションが生じた際に鍵を再発行したうえで再度暗号化を行う必要があり非効率である。また本研究では CCA-secure 及び結託攻撃耐性を有したプロキシ再暗号の活用を行っておりよりセキュリティ面で高い性能を有している点で文献 [6] とは異なる。

3. 関連技術

本章では、本研究において関連する技術について概説する。最初にプロキシ再暗号に関して説明し、その後ブロックチェーン技術について述べる。

3.1 暗号技術

最初に、暗号技術の要素技術について説明する。その後、本研究で使用するプロキシ再暗号について述べる。

3.1.1 双線形写像

以下に、双線形写像と双線形群について簡単に説明する。

$\mathbb{G}_1$ と $\mathbb{G}_2$、及び $\mathbb{G}_T$ をそれぞれ素数位数 p の乗法群とする。g_1 を $\mathbb{G}_1$ の生成元、g_2 を $\mathbb{G}_2$ の生成元とし、e を双線形写像 $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ とする。このとき双線形写像 e は以下の特性を持つ。

1. 双線形性：すべての $u \in \mathbb{G}_1$、$v \in \mathbb{G}_2$ 及び a、$b \in \mathbb{Z}_p$ において、
   $e(u^a, v^b) = e(u, v)^{ab}$ を満たす。
2. 非退化：$e(g_1, g_2) \neq 1$。

$\mathbb{G}_1$ と $\mathbb{G}_2$ から群作用を効率的に計算可能で、かつ $\mathbb{G}_T$ の群作用と双線形写像
$e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ がともに効率的に計算可能なものが存在する場合、$\mathbb{G}_1$ 及び $\mathbb{G}_2$ を共に双線形群とする。

3.1.2 階層型 ID ベース暗号

ID ベース暗号 (IBE : Identity Based Encryption) システム [16] [17] はユーザの email アドレスなど任意の文字列を公開鍵として使用できる公開鍵暗号方式である。中央機関はユーザの鍵発行依頼に応じ、マスター秘密鍵を用いてユーザの ID に対応した秘密鍵を発行する。

階層型 ID ベース暗号 (HIBE : Hierarchical IBE)[18] [19] は組織階層を反映した IBE の一般化である。階層木レベル k の識別子は木構造における自分の子にあたる識別子に対応した秘密鍵を発行できる。この時、意図した異なる識別子向けのメッセージは復号できない。最初に HIBE が提案されたのはランダムオラクルモデルの双線形ディフィーヘルマン仮定に基づいてセキュリティを担保した Gentry と Silverberg[19] のものである。また、その後 Boneh と Boyen はランダムオラクルのない双線形ディフィーヘルマン仮定に基づいた効率的な（ID を選択可能な）HIBE を提案した [14]。

A HIBE System with Constant Size Ciphertext
本研究で使用する HIBE について簡単に説明する。本研究では Dan Boneh らの階層の深さによらない暗号文サイズや復号コストを発揮する HIBE を用いる [20]。選定理由については第 4. 章で述べる。

$\mathbb{G}_1$ 及び $\mathbb{G}_2$ を素数位数 p の双線形群とする。また $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ を双線形写像とする。ここで、公開鍵（ユーザ識別子）を ID とし、階層の深さ k における ID のベクトルの各要素を $\mathbb{Z}_p^{*k}$ に属する整数とすると、ID$= (I_1, ..., I_k) \in \mathbb{Z}_p^{*k}$ と表せる。以下に HIBE システムの流れを記述する。

Setup: 階層木の最大の深さを l としたときのシステムパラメータを生成する。ランダムな生成元 $g \in \mathbb{G}_2$ を生成し、ランダムな $\alpha \in \mathbb{Z}_p$ を選ぶ。ここで、$g_1 = g^\alpha$ としておく。ランダムなパラメータ $g_2, g_3, h_1, ..., h_l \in \mathbb{G}_1$ をそれぞれ生成する。公開パラメータとマスター鍵をそれぞれ、

$$params = (g, g_1, g_2, g_3, h_1, ..., h_l), \qquad msk = g_2^\alpha.$$

とする。

KeyGen(from msk): $params$、msk、ID を入力とし、識別子 ID $= (I_1, ..., I_k) \in \mathbb{Z}_p^{*k}$ の秘密鍵 d_{ID} を生成する。ただし、ID ベクトルの要素数は $k < l$。ランダムな値 $r \in \mathbb{Z}_p$ を選択し、

$$d_{ID} = (g_2^\alpha \cdot (h_1^{I_1} \cdot ... \cdot h_k^{I_k} \cdot g_3)^r, g^r, h_{k+1}^r, ..., h_l^r) \in \mathbb{G}^{2+l-k}.$$

を出力する。

KeyGen(from parent): ID の要素数 $k-1$ をインクリメントする。つまり階層木の深さ $k-1$ をインクリメントする。このとき既に ID $= (I_1, ..., I_{k-1})$ の秘密鍵 $d_{ID|k-1}$ は生成されており、$k < l$ である。新たに生成された ID $= (I_1, ..., I_{k-1}, I_k) \in \mathbb{Z}_p^{*k}$ の秘密鍵 d_{ID} を生成する。

$$d_{ID|k-1} = (g_2^\alpha \cdot (h_1^{I_1} \cdot ... \cdot h_{k-1}^{I_{k-1}} \cdot g_3)^{r'}, g^{r'}, h_k^{r'}, ..., h_l^{r'})$$

$$= (a_0、a_1、b_k、...、b_l)。$$

としたとき、ランダムな $t \in \mathbb{Z}_p$ を選択し、秘密鍵 $d_{ID|k-1}$、params、ID をインプットとして以下のように d_{ID} を生成する。

$$d_{ID} = \begin{array}{l}(a_0 \cdot b_k^{I_k} \cdot (h_1^{I_1} \cdot ... \cdot h_k^{I_k} \cdot g_3)^t、\\ a_1 \cdot g^t、b_{k+1} \cdot h_{k+1}^t、...、b_l \cdot h_l^t)。\end{array}$$

Encrypt: $params$、ID、平文 M をインプットとして暗号文 CT を出力する。ここで、$M \in \mathbb{G}_T$ であり、ID $= (I_1、... 、I_k) \in \mathbb{Z}_p^{*k}$ とする。ランダムな $s \in \mathbb{Z}_p$ を選び、以下を出力する。

$$CT = (e(g_1、g_2)^s \cdot M、g^s、(h_1^{I_1} \cdot ... \cdot h_k^{I_k} \cdot g_3)^s)$$

$$\in \mathbb{G}_T \times \mathbb{G}_2 \times \mathbb{G}_1。$$

Decrypt: ID $= (I_1、..., I_k) \in \mathbb{Z}_p^{*k}$ で暗号化された暗号文 $CT = (A、B、C)$ を秘密鍵 $d_{ID} = (a_0、a_1、b_k、...、b_l)$ を用いて復号する。復号の式は以下の通りである。

$$A \cdot e(a_1、C)/e(B、a_0) = M。$$

ここで、$A = e(g_1、g_2)^s$ であり、

$$\frac{e(a_1、C)}{e(B、a_0)} = \frac{e(g^r、(h_1^{I_1} \cdot ... \cdot h_k^{I_k} \cdot g_3)^s)}{e(g^s、g_2^\alpha \cdot (h_1^{I_1} \cdot ... \cdot h_k^{I_k} \cdot g_3)^r)}$$

$$= \frac{1}{e(g、g_2)^{s\alpha}} = \frac{1}{e(g_1、g_2)^s}。$$

であることから復号が可能となっている。

3.1.3 マルチホップ一方向 ID ベースプロキシ再暗号

本研究で使用したマルチホップ一方向 ID ベースプロキシ再暗号 (MUIBPRE : Multi-use Unidirectional Identity Based Proxy Re-Encryption)[21] について記述する。

プロキシ再暗号

プロキシ再暗号 (RRE : Proxy Re-Encryption) とは暗号文の変換によって復号を委譲する方法である。

具体例として、通常の公開鍵暗号方式を用いて Alice の公開鍵で暗号化された暗号文を Bob の秘密鍵で復号できる暗号文に変換する場合について述べる。まず、暗号文の変換を行うにあたって一度暗号文を Alice の秘密鍵 sk_{Alice} で復号する必要がある。この時にセキュリティの観点から sk_{Alice} を Bob、或いは第 3 者に渡すのは好ましくない。よって暗号文の変換は Alice が行う。ここで、サーバ（例えば mail Server）などの中間者 (proxy) を介してデータを送る場合、暗号文の変換は proxy が行う必要がある。Alice は sk_{Alice} を proxy に渡す必要があるがこれは上述したように好ましくない。さらに proxy は 1 つのデータに対して復号と暗号化をそれぞれ行う必要があり非効率である。

プロキシ再暗号は Alice の公開鍵で暗号化された暗号文を proxy が sk_{Alice} を必要とせず、且つ 1 度の処理で Bob の秘密鍵で復号できる暗号文に変換することができる。公開鍵暗号とプロキシ再暗号の流れの比較は図 1 の通りである。

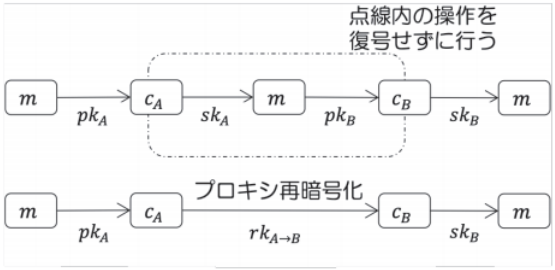

図 1. 公開鍵暗号とプロキシ再暗号の比較（[6] より引用）

MUIBPRE

本研究で使用した MUIBPRE について記述する。本研究では Jun Shao らの提案した MUIBPRE[21] を用いる。Jun Shao らは CPA(Chosen Plaintext Attack)-secure な NaHIBE(Non-anonymous Hierarchical Identity-Based Encryption) を MUIBPRE へ拡張するスキームを提案した。その際に CCA(Chosen Ciphertext Attack)-secure と耐結託耐性を有することを証明した。MUIBPRE ではプロキシ再暗号全体の中に NaHIBE、SIG(Signature)、SKE(Symmetric Key Encryption) の 3 つを使用している。それぞれの暗号、署名は以下の機能をもつ。

NaHIBE

- KeyGen （公開パラメータ、マスター秘密鍵の生成を行う。）
- Extract（ユーザの秘密鍵を生成する。本研究では Key Gen From Master が該当）
- Delegate（復号権の委譲。本研究では Key Gen From Parent が該当）
- Enc （暗号化）
- Dec （復号）

SIG

- G (鍵の発行。署名検証用鍵 svk と署名鍵 ssk をそれぞれアウトプット)
- S （ssk を用いてメッセージの署名を行う）
- V (svk とシグネチャを用いて署名検証を行う)

SKE

- KeyGen （共通鍵の生成）
- Enc (暗号化)
- Dec （復号）

論文中では Waters09[22] と呼ばれる Dual System Encryption を用いてその例を示したが本研究では上述した HIBE system with Constant Size Ciphertext を使用する。

Multi-use。

Multi-use とは暗号化したデータを再暗号化できる回数のことである。Single-use では暗号化したデータを再暗号化するとそれ以降再暗号化できない。一方 Multi-use は再暗号化を何

度でもできる。文献 [21] で提案された暗号手法では再暗号化する度に暗号文のサイズが増加する。しかし本研究では 1 つのデータの再暗号化の回数は高々 2 回であるためこの問題は無視できる。

unidirectional。
プロキシ再暗号には一方向性 (unidirectional) と双方向性 (bidirectional) が存在する。双方向性は Alice の暗号文を Bob の暗号文に変換するときに使用する再暗号鍵を、Bob の暗号文から Alice の暗号文に変換するときにも使用できる。一方向性では Alice から Bob、Bob から Alice への暗号文の変換にそれぞれ異なる鍵が必要となる暗号方式である。本研究ではアクセス権として再暗号鍵を管理し、要求などによってその権限の内容が異なるため、一方向性のプロキシ再暗号を使用した。

CCA-secure。
CCA とは攻撃者が選択した暗号文を正規のユーザに復号させ、元の暗号文と得られた平文から暗号鍵を推測し、同じ暗号鍵を用いて作られた暗号文を解読しようとする攻撃である。文献 [21] では意図しない攻撃者が復号オラクルにアクセスできても決して平文を入手することはできないとしており、CCA-secure を担保している。

耐結託攻撃
結託攻撃は、悪意のあるユーザとプロキシが結託することで意図した秘密鍵の入手を行う攻撃である。文献 [21] ではプロキシはユーザの秘密鍵を所有しないためこの攻撃への耐性を示している。再暗号の再にユーザの秘密鍵を用いらず、秘密鍵から新たに生成した鍵を Sub-privatekey としてその場限りで使用することにより結託攻撃に耐性がある。

アルゴリズム
以下に MUIBPRE の流れを記述する。

Setup: システムパラメータ (NaHIBE、SIG、SKE) をセットする。NaHIBE は CPA-secure を有した階層型 ID ベース暗号である。SIG は偽造困難で署名鍵を one-time でしか使用しない署名スキームである。SKE は CCA-secure な、鍵を one-time でしか使用しない共通鍵暗号方式である。

KeyGen: セキュリティパラメータ 1^{λ} をインプットとして、
NaHIBE.KeyGen$(1^{\lambda}) \rightarrow (mpk、msk)$。
このとき、mpk は公開パラメータ、msk はマスター秘密鍵である。

Extract: ID をインプットとして、その ID に対応した秘密鍵を生成する。
NaHIBE.Extract$(ID) \rightarrow d_{ID}$

ReKeyGen: 委譲元であるユーザの ID 及びその秘密鍵 d_{ID}、委譲先のユーザの ID' をインプットとして再暗号鍵を生成する。
NaHIBE.Delegate$(d_{ID}、ID) \rightarrow d_{ID,\ ID}$、
$d_{ID,\ ID}$ を $d^{(1)}_{ID,\ ID}$、$d^{(2)}_{ID,\ ID}$ に分割する。このとき、$\mathcal{F}(d^{(1)}_{ID,\ ID}、d^{(2)}_{ID,\ ID}) = d_{ID}$、$\mathcal{F}$ は関数である。また、以下を計算する。

NaHIBE.Enc$(mpk、(ID'、ID'、svk)、sk) \rightarrow A$、

SKE.Enc$(sk、d^{(2)}_{ID,\ ID}) \rightarrow B$、

SIG.S$(ssk、(A、B)) \rightarrow C$

$(svk、ssk)$ は SIG におけるランダムな署名鍵のペアで、sk は SKE におけるランダムな共通鍵である。それぞれ使用直前に作成しておく。なお、それ以降でこの鍵を新たに暗号化、署名することには使用しない。(one-time)

$rk^{(1)} = d^{(1)}_{ID,\ ID}$、$rk^{(2)} = (A、B、C、svk)$

として再暗号鍵を $rk = (rk^{(1)}、rk^{(2)})$ とする。

Enc: 平文 m と暗号化に用いる ID をインプットとして暗号文を作成する。

NaHIBE.Enc$(mpk、(ID、ID、\overline{svk})、\overline{sk}) \rightarrow \overline{A}$、

SKE.Enc$(\overline{sk}、m) \rightarrow \overline{B}$、

SIG.S$(\overline{ssk}、(\overline{A}、\overline{B})) \rightarrow \overline{C}$。

$(\overline{svk}、\overline{ssk})$ は SIG におけるランダムな署名鍵のペアである。$\overline{sk}$ は SKE におけるランダムな共通鍵である。それぞれ ReKeyGen で作成したものとは因果関係を持たず、使用直前で作成される。なお、この鍵を新たに暗号化、署名に使用することはない。ここで、暗号文は $(\overline{A}、\overline{B}、\overline{C}、\overline{svk})$ である。

ReEnc: ID により作成された暗号文 $(\overline{A}、\overline{B}、\overline{C}、\overline{svk})$、再暗号鍵 $(rk^{(1)}、rk^{(2)})$ をインプットとする。最初に、暗号文の要素 $\overline{A}$ が ID ベクトル $(ID、ID、\overline{svk})$ 下で暗号化されたものであるかを検証する。

SIG.V$(\overline{svk}、(\overline{A}、\overline{B})、\overline{C})$。

検証に失敗した場合、$\perp$ を返す。次に再暗号の手順を示す。手順は大きく 4 フェーズに分ける。

1. NaHIBE.Dec$(\overline{\mathcal{F}}(rk^{(1)}、d)、\overline{A}) \rightarrow \widehat{\overline{A}}$
   ここで、$\overline{\mathcal{F}}$ はある関数、d はユーザの秘密鍵スペースからランダムに選んだパラメータである。
2. NaHIBE.Enc$(mpk、(ID'、ID'、svk')、sk') \rightarrow A'$
   SKE.Enc$(sk'、(\overline{A}、d)) \rightarrow B'$
   ここで、$(svk'、ssk')$ は SIG におけるランダムな署名鍵のペアである。また sk' は SKE におけるランダムな

共通鍵である。それぞれ ReKeyGen や Enc で作成した鍵とは因果関係を持たず、使用直前で作成する。なお、これらの鍵を新たに暗号化、署名に使用することはない。

3. SIG.S(ssk'、($\widehat{A}$、$\overline{B}$、$\overline{C}$、$\overline{svk}$、$rk^{(2)}$、A'、B')) → C'
4. 再暗号化された暗号文をアウトプットする。再暗号化された暗号文は以下に示す。

$$(\widehat{A}、\overline{B}、\overline{C}、\overline{svk}、rk^{(2)}、A'、B'、C'、svk')$$

$$= (\widehat{A}、\overline{B}、\overline{C}、\overline{svk}、A、B、C、svk、A'、B'、C'、svk')$$

もし $\widehat{A}$ の値が 1 つでなくとも、$\widehat{A}$ の整合性チェックを行う手法は存在しなければならない。整合性確認手法がなければ再暗号化された暗号文の有効性の検証に CCA-secure な SKE を使用できない。

Dec: 暗号文、秘密鍵をインプットとして平文をアウトプットする。暗号文は 2 種類あるためそれぞれの手順を記述する。

- インプットされた暗号文が ($\overline{A}$、$\overline{B}$、$\overline{C}$、$\overline{svk}$) であった場合。ただし、この時暗号文は ID によって暗号化されたものであるため d_{ID} を用いて復号を行う。
  - 最初に暗号文の検証を行う。暗号文の要素 $\overline{A}$ が ID ベクトル (ID、ID、$\overline{svk}$) 下で暗号化されたものであるかを検証する。SIG.V($\overline{svk}$、($\overline{A}$、$\overline{B}$)、$\overline{C}$)。検証に失敗した場合、⊥ を返す。
  - NaHIBE.Dec(d_{ID}、$\overline{A}$) → $\overline{sk}$。
  - SKE.Dec($\overline{sk}$、$\overline{B}$) → m。
- ($\widehat{A}$、$\overline{B}$、$\overline{C}$、$\overline{svk}$、A、B、C、svk、A'、B'、C'、svk') の場合。ただし、この時暗号文は ID' 宛に再暗号化されたものであるため $d_{ID'}$ を用いて復号を行う。
  - A' が ID ベクトル (ID'、ID'、svk') 下で暗号化されたものであるかを検証する。
    SIG.V(svk'、($\widehat{A}$、$\overline{B}$、$\overline{C}$、A、B、C、svk、$rk^{(2)}$、A'、B')、C')。検証に失敗した場合、⊥ を返す。
  - A が ID ベクトル (ID'、ID'、svk) 下で暗号化されたものであるかを検証する。
    SIG.V(svk、(A、B)、C)。検証に失敗した場合、⊥ を返す。
  - NaHIBE.Dec($d_{ID'}$、A') → sk'、
    NaHIBE.Dec($d_{ID'}$、A) → sk。
  - SKE.Dec(sk'、B') → ($\overline{A}$、d)、
    SKE.Dec(sk、B) → $d^{(2)}_{ID、ID}$。
  - SIG.V($\overline{svk}$、($\overline{A}$、$\overline{B}$)、$\overline{C}$)。検証に失敗した場合、⊥ を返す。
  - NaHIBE.Dec($\overline{\mathcal{F}}^{-1}(d^{(2)}_{ID、ID}、\overline{A})$) → $\breve{\overline{A}}$。入手した $\breve{\overline{A}}$ と $\widehat{A}$ を結合して $\overline{sk}$ を得る。ここで、$\overline{\mathcal{F}}^{-1}$ は $\overline{\mathcal{F}}$ の逆演算子である。
  - SKE.Dec($\overline{sk}$、$\overline{B}$) → m。

ここで、暗号文 ($\overline{A}$、$\overline{B}$、$\overline{C}$、$\overline{svk}$) を 1st-level ciphertext ($\overline{A}_1$、$\overline{B}_1$、$\overline{C}_1$、$\overline{svk}_1$) と考えることができる。一方、再暗号化された暗号文 ($\widehat{A}$、$\overline{B}$、$\overline{C}$、$\overline{svk}$、A、B、C、svk、A'、B'、C'、svk') は 2nd-level ciphertext ($\widehat{A}_1$、$\overline{B}_1$、$\overline{C}_1$、$\overline{svk}_1$、A_2、B_2、C_2、svk_2、A'_2、B'_2、C'_2、svk'_2) と考えることができる。

3.2 ブロックチェーン

次に、ブロックチェーン技術について記する。ブロックチェーン技術はビットコインなど仮想通貨の基盤となる技術で近年注目を集めている。また、ロジスティクス分野や医療分野など様々な産業分野での応用が検討されており [23]、今後もその応用範囲は拡大していくと予想される。本稿では最初にブロックチェーンの概要について説明する。続いてブロックチェーン技術の特徴や種類について説明し、本研究で使用したブロックチェーンプラットフォームについて紹介する。

3.2.1 ブロックチェーン技術の概要

ブロックチェーンは、2008 年に Satoshi Nakamoto の研究 [24] で発表された仮想通貨であるビットコインの基盤となる技術のことを指す。ブロックチェーン技術は中央機関を必要としない技術であり、分散台帳技術とも呼ばれている。通常の分散型データベースシステムなどでは必ず信頼できる中央機関が存在し企業間や個人間の取引の仲介を行い、取引の正当性を担保していたが、ブロックチェーン技術では取引の正当性を担保したまま中央機関を介さない取引を実現している。

ユーザ間で取引を行う事を想定する。まず送金側のユーザは送金量や宛先のアドレス、自身の署名などの情報を含んだトランザクションを発行する。発行されたトランザクションは相互接続された次のノードに渡される。トランザクションを受け取ったノードはそのトランザクションの検証を行う。検証して問題がなかった場合、またその次のノードに送信され検証が行われる。最終的にマイナーによりブロックとしてブロックチェーンの一部となり送金処理が完了する。

ブロックチェーンは既存の複数の要素技術を組み合わせて実現した技術である。以下にブロックチェーンを構成する要素技術について記述する。

P2P(ピア・ツー・ピア) ネットワーク

P2P はインターネットに接続した PC やサーバなどが相互にコミュニケーションをとるネットワークの形態を指す。P2P は通常のクライアント-サーバ型のファイル取得方式とは異なり、ネットワークに参加する各ノードがそれぞれファイルを保持する。そのため P2P に参加しているすべてのノードは対等である。

コンセンサスアルゴリズム

ブロックチェーンでは、ブロックチェーン上のデータの完全性や合意形成のために、コンセンサスアルゴリズムを用いている。具体的には、Proof of Work(PoW) や Proof of Stake(PoS)、Practical Byzantine Fault Tolerance などがある。ビットコインでは、コンセンサスアルゴリズムとして PoW が利用されている。PoW では、あるブロックのハッシュ値が特定の条件を満たす Nonce を計算することでブロックを生成するアルゴリズムである。PoW により生成されたブロックはノード

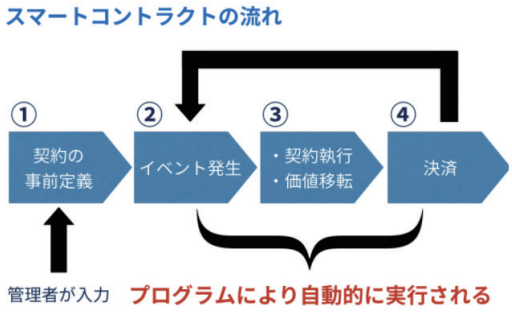

図 2. スマートコントラクトの流れ（[26] より引用）

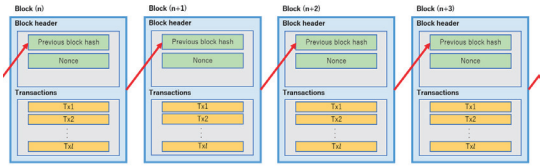

図 3. ブロックチェーンの台帳の構造イメージ

間のブロックの送受信によりビットコインネットワーク全体に伝搬され、独立にその有効性が検証される。ブロックが有効な場合、ブロックチェーンの台帳として保存される。

電子署名

例えば、ビットコインなどでは電子署名の技術を応用したマルチシグネチャーと呼ばれる、複数の署名を利用した手法が利用されている [25]。送金を行う際に秘密鍵が複数に分割されており一定数の鍵を合わせる必要がある署名である。この場合、何らかの要因で秘密鍵が保存された端末のうち 1 つから鍵が流出しても、ほかの複数の鍵も同時に入手しなければ攻撃が成立しない。

スマートコントラクト

スマートコントラクトは一般にブロックチェーンネットワーク上で動作するプログラムを指す。ブロックチェーン上で行われる契約（コントラクト）の自動化を行う。取引プロセスを自動化することで決済期間の短縮や不正防止、仲介者を介さないことによるコスト削減に寄与すると期待されている。また、スマートコントラクト自体はブロックチェーン上に保存されるため、後述するような耐改ざん性も保証される。スマートコントラクトの流れは図 2 に示す。

ブロックチェーン技術は、それぞれの取引データを台帳としてブロックに保存する。この時、ブロックにはヘッダ情報として前のブロックのハッシュ値や Nonce、作成者などの情報が含まれる。前のブロックのハッシュ値を記録することで Genesis Block(ブロックチェーンネットワークにおける最初のブロック) まで遡ることができる仕組みになっている。このようにチェーン (鎖) のようにブロックが繋がっているようにみえることからブロックチェーンと呼ばれるようになったと考えられる。ブロックチェーンの台帳の構造イメージを図 3 に示す。

3.2.2 ブロックチェーン技術の特徴

ブロックチェーンの特徴として、以下の 4 つが挙げられる [27]。

Decentralized

従来の集中型トランザクションシステムでは、各トランザクションを中央の信頼できる機関（中央銀行など）を通して検証する必要があり、必然的に中央サーバのコストやパフォーマンスのボトルネックが生じる。ブロックチェーン技術では、データ共有やトランザクションの正当性検証のために、中央となる第三者機関が存在しない。データ共有やトランザクションの検証はブロックチェーンに参加する各ノードが行うため、その計算コストやリソースを分散化することが可能である。

Persistency

トランザクションはマイナーによって迅速に検証が行われ、無効なトランザクションは許可されない。ブロックチェーン上のデータの完全性の維持やネットワーク全体での合意形成のために、PoW や PoS などのコンセンサスアルゴリズムが利用される。これらのコンセンサスアルゴリズムにより、ブロックチェーンに含まれるトランザクションを改ざんすることは困難である。例えば、PoW ではブロックを生成するためには膨大な計算資源が必要となる。仮にあるブロックに含まれるトランザクションを改ざんしようとした場合、そのブロックのハッシュ値も変更される。この時に改ざんしたブロック以降のブロックも再度計算しなおす必要がある。ブロックを生成するためには膨大な計算資源が必要となるため、ブロックチェーンの改ざんは困難である。

Anonymity

ブロックチェーンネットワークにおいて、あるノードは一つ以上のアドレスで表現される。例えば、ビットコインネットワークでは、「1NYXCdriKyqLjgSffmmC6xysxCMg3LVvpW」のような形式のアドレスを複数所有することが可能である。これらのアドレスは一般に、個人に結びつかない。そのため、ブロックチェーンネットワークでは、匿名でトランザクションを発行することが可能である。

Auditability

ブロックチェーンネットワークでは、不正なブロックやトランザクションを排除するため、ブロックチェーンネットワーク参加者である各ノードがブロックやトランザクションを検証する必要がある。そのためブロックやトランザクションは「検証可能」および「追跡可能」である。

3.2.3 ブロックチェーンの種類

本項では、ブロックチェーンの種類について述べる。ブロックチェーンはその特性から、パブリック型、コンソーシアム型、プライベート型の 3 種類に分けられる。ビットコインは、パブリック型ブロックチェーンの初めての実装例であり、インターネット上の不特定多数が参加できるブロックチェーンとして普及してきた。一方コンソーシアム、プライベート型ブロックチェーンは企業や組織間での特定の取引等を扱うために用いられる。コンソーシアム、プライベート型ブロック

Table 1 Comparisons among *public blockchain*, *consortium blockchain* and *private blockchain*

Property	*Public blockchain*	*Consortium blockchain*	*Private blockchain*
Consensus determination	All miners	Selected set of nodes	One organisation
Read permission	Public	Could be public or restricted	Could be public or restricted
Immutability	Nearly impossible to tamper	Could be tampered	Could be tampered
Efficiency	Low	High	High
Centralised	No	Partial	Yes
Consensus process	Permissionless	Permissioned	Permissioned

図 4. ブロックチェーンの種類とその特徴 ([27] より引用)

チェーンは信頼できる企業、組織のみが参加することを前提としているため、PoW とは異なる軽量なコンセンサス方式が採用される。またマイニングにおけるインセンティブが不要である利点もある。しかしパブリック型よりも分散化の利点が小さくなる欠点も抱える。図 4 にブロックチェーンの種類とそれぞれの特徴を示す。

図 4 から、それぞれの形態によって特徴の「度合い」が異なることがわかる。最も特徴的であるのはブロックチェーン参加者、つまり台帳を共有するノードを誰に想定するかである。本研究では台帳を医療機関によって管理することを想定している。よってコンソーシアム型ブロックチェーンを活用することが適切であると考える。

3.2.4 ブロックチェーンプラットフォーム

ブロックチェーンプラットフォームには Bitcoin Core をはじめとして、Ethereum[29] や Hyperledger Fabric[30] などが存在する。他にも、Factom[31] や GemHealth[32]、PokitDok[33] など様々なプラットフォームが存在する。Ethereum はスマートコントラクトを利用した分散型アプリケーション (Dapps: Decentralized applications) を構築するためのプラットフォームとして作成された。Ethereum ブロックチェーンを利用した分散型のサービスとしては、uPort[34] が挙げられる。

Hyperledger Fabric は、Linux Foundation によって 2015 年に開始された Hyperledger プロジェクトの 1 つであるブロックチェーン基盤の実装フレームワークである。Hyperledger はオープンソースのブロックチェーンプラットフォームであり、広範囲なビジネス用途のブロックチェーン基盤を提供することを目的としたプロジェクトである。本研究ではコンソーシアム型のブロックチェーンを想定しているため、Hyperledger Fabric を使用する。

4. 提案手法

本項では、本研究における提案手法を説明する。1. 章で述べたように、医療において発生する患者データは医療機関が独自に保有し、これらデータは外部医療に活用することは難しい。また研究などへの 2 次利用なども考慮すると医療データの他機関との共有が行えるプラットフォームが必要となる。

本研究では医療データ管理に際しそのアクセス権限をブロックチェーンで管理する手法を提案する。医療データ本体自体は文献 [12] で使用したようにクラウド上 (医療機関の専用サーバなど) に保存する。先行研究では医療データをブロックチェーンの台帳として管理する手法を提案するものが多いが、医療データには X 線画像などのような大容量のデータも含まれるため、スケーラビリティも考慮すると好ましくないためである。そこで必要とされる要件にアクセス権限や暗号化が存在する。本研究では文献 [6] を参考にプロキシ再暗号を用いてその両方を実現する。ここで、医療データは一般に個人情報でありプライバシー保護の観点からその管理は厳重に行わなければならない。よってプロキシ再暗号として使用する暗号技術はより攻撃耐性を有している必要がある。本研究では文献 [21] で提案した手法を用いて文献 [20] で提案されている HIBE を CCA-secure、耐結託性を有したプロキシ再暗号へ変換した暗号手法を実装した。また文献 [20] は MUIBPRE が CCA-secure であるための必要条件である CPA-seture であるため採択した。

表 1. ユーザの操作可能範囲

ユーザ	閲覧	挿入
患者	可	不可
医療従事者	可	可
解析者	可	不可

4.1 ブロックチェーン参加者

本研究において登場するブロックチェーン参加者について記する。まず Hyperledger における Peer の役割は医療機関、鍵生成局 (信頼できる第三者機関) が行う。医療機関のサーバに患者の医療データを保存し、台帳としてその医療データに対するアクセス権限を記録し共有する。このときアクセス権限はプロキシ再暗号における再暗号鍵であるため鍵生成局が発行をする。またこれら台帳を保有せずブロックチェーンを介して医療データにアクセスするノードをユーザとする。ユーザは医療従事者や患者、解析者などの個人を想定する。しかし医療行為や医療データに対する解析は複数人である場合が多いため医療従事者や解析者はその限りではない。医療従事者として、主に主治医やそれに属する看護師、技師などが存在しそれとともに担当外の医師も存在する。現行システム [35] ではある患者に対する、主治医が見ることのできる医療データの範囲と看護師が見ることのできる範囲、担当外の医師が見ることのできる医療データの範囲は物理的に等しい。よって医療従事者のクラス分けは本研究では行わない。それぞれのユーザの医療データに対して行う操作を閲覧、挿入として表 1 にその操作可能範囲を示す。

4.2 提案手法の概要

本節では提案手法の概要について説明する。提案手法の概略図を図 5 に示す。ユーザは自身のアプリケーションを通してブロックチェーン上のアクセス権を用い医療データの取得を行ったり、鍵生成局にアクセス権限の作成依頼を行う。(1) ユーザが医師或いは解析者など他のユーザから医療データに関するアクセス権限の作成依頼を受けた場合、鍵生成局へその request を送信する。鍵生成局はユーザからのアクセス権限作成依頼を取得するとその依頼の検証を行い (デジタル署名)、(2) 再暗号鍵の発行を行う。この再暗号鍵は医療機関からなるブロックチェーンネットワークへストアし、(3) それぞれの医療機関ノードが検証を行って台帳に保存する。(4) 台帳に権限情報が保存されたらその鍵を用いてクラウドにある医療データへアクセスし、中身を取得することができる。また、

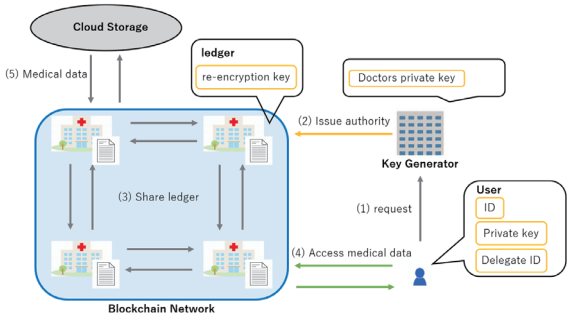

図 5. 提案手法の概略

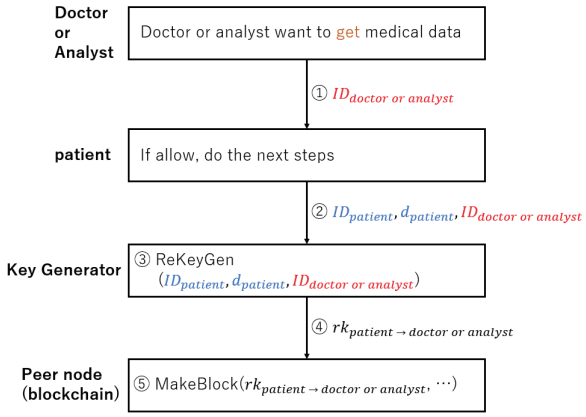

図 6. 医療データ閲覧権限作成手順

医療データ作成権限を作成した際にはその鍵を用いて特定のユーザのデータを作成し、クラウドに保存可能である。(5) 医療データはすべてクラウド上で保管されスマートコントラクトを介してのみアクセス可能である。本研究では、医療データに対する操作を閲覧、挿入としているが、それぞれの権限作成方法を示す。

4.2.1 医療データ閲覧権限作成

例えば患者の医療データに対して医療従事者や解析者が閲覧をする際に作成される。このとき、患者のデータはクラウド上に患者の ID で暗号化した状態で保存されている。これにより患者は権限情報の作成などを行わずに自分の情報を閲覧できる。患者の医療データ作成手順を図 6 に示す。まず医療従事者あるいは解析者がデータの閲覧を行いたい患者に閲覧権限の作成依頼を行う。①作成依頼をする場合は自身の ID も依頼に添える。②権限作成依頼を受けた患者は鍵生成依頼を自分の秘密鍵と一緒に鍵生成局へ送信する。このとき鍵生成局は依頼の真正性を検証する。③検証に問題がなければ医師或いは解析者の ID と患者の秘密鍵を用いて再暗号鍵の作成を行う。このときこの再暗号鍵は患者データを医師データへと変換する鍵が作成される。④作成した再暗号鍵をブロックチェーンネットワークに参加してあるノードに送信し、⑤台帳として共有、管理する。本研究で使用するプロキシ再暗号は unidirectional であるため、医師データを患者データへと変換する際にこの鍵は使用できない。

4.2.2 医療データ挿入権限作成

医療データ挿入権限の作成は閲覧権限を作成した際の手順とほぼ同じである。比較を容易にするため、図 7 に医療データ挿入権限作成の手順を示す。閲覧権作成と異なる点として

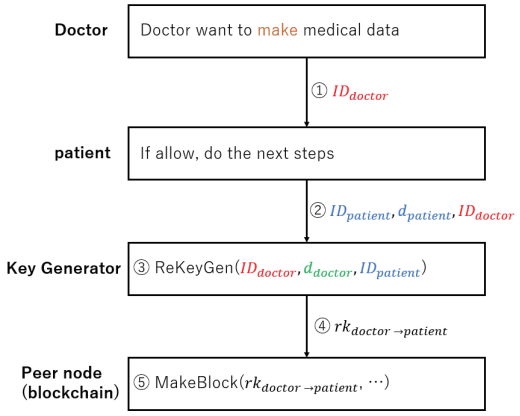

図 7. 医療データ挿入権限作成手順

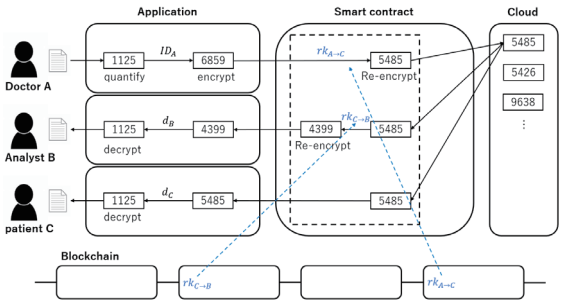

図 8. 暗号化及び復号の手順

再暗号鍵の方向がある。挿入権限は医師データを患者データへ変換する再暗号鍵である。よって作成に必要になる秘密鍵が患者ではなく医師のものとなる。医師の秘密鍵は鍵生成局が登録し保持しておりユーザは自身の身元さえ証明できれば③鍵生成局は医師の秘密鍵を用いて挿入権限の作成ができる。

4.2.3 医療データ暗号化及び復号

医療データの暗号化及び復号の手順を図 8 に示す。まず患者の医療データが作成されると医師は自分の ID で暗号化する。その後暗号化されたデータを患者宛の暗号文へ変換する。このとき権限（再暗号鍵）がなければ暗号文の変換は行われず医療データ作成は失敗する。再暗号化されたデータは医療機関のサーバなどクラウドで管理される。また医療データは患者宛に変換されている為、患者 C が自身の医療データを取得したいと考えればいつでも取得可能となっている。解析者や医師が患者の医療データにアクセスしたい場合、ブロックチェーンに保存されている権限を用いてクラウド上で管理されている医療データを再暗号化する。このとき権限がなければ再暗号化は行われない。また、適切な再暗号鍵を用いて再暗号が行われなければ自身の秘密鍵を使って医療データを復号することはできない。再暗号化されることにより権限を持った医師或いは解析者は患者のデータを取得することができる。また、図 8 では患者の医療データは 1 度再暗号化されて管理される。患者が自身の医療データを取得する際には再暗号化は行われず、医師または解析者が患者の医療データを取得する際には更に再暗号化される。再暗号化されたデータの用途は復号のみを想定している。このことから本研究では 1 つの医療データの再暗号化回数は高々 2 回であることがわかる。

表 2. トランザクション定義

トランザクション	実行者	機能
AddReKey	鍵生成局	再暗号鍵生成
AddEncTreatment	ユーザ（医療従事者）	医療データ挿入
DectyptTreatment	ユーザ	医療データ閲覧

4.3 トランザクション

本研究におけるブロックチェーンのトランザクションについて定義する。表 2 にトランザクションを記す。本研究では医療データに対して行う操作を挿入及び閲覧のみを想定している。また再暗号鍵生成により生成される再暗号鍵をブロックチェーンに記録しておくため、トランザクションは 3 つとする。

4.4 ID の委譲に関して

本研究で使用した MUIBPRE において、ID の委譲は ReKeyGen で行っている。この機能は鍵生成局のもとで行うため、患者は自身の秘密鍵を鍵生成局へ送信する必要がある。本研究ではユーザと鍵生成局は安全な通信路を用いて行うことを前提としているが、幾度も秘密鍵を送信することは好ましいとは言えない。そこで ReKeyGen の機能における delegate(KeyGenFromParent) 関数のみをユーザの手元で行うよう改良を加えた。これによりユーザは新たに生成した秘密鍵を鍵生成局へ送信することで自身の秘密鍵を送らず再暗号鍵を生成することができる。ここで、生成された秘密鍵から委譲基となった秘密鍵は割り出すことはできない。仮に何等かの手法で送信した秘密鍵を盗聴された場合、その鍵に関する暗号文を再度暗号化しなおし、漏洩した鍵とその ID を破棄することで鍵は効力を失う。さらにユーザは委譲した新たな秘密鍵を鍵生成局へ送信したのち、破棄できる。これは作成した秘密鍵で復号できる暗号文は、委譲基となった秘密鍵と委譲し新たに追加した ID ベクトルを用いれば復号できるからである。よって患者は自身の ID、秘密鍵、委譲した ID さえ保持すればよい。

4.5 ブロックチェーン技術を用いる理由

本研究では医療行為等により発生する患者の医療データは患者のデータとして管理する、という観点で研究を行っている。ここで、従来の中央集権型の分散型データベースにおいてもこのシステムは実現できる。また「ねんきんネット」[36] や「マイポータル」[37] のような既存のデータベースに患者自身でアクセス可能とする仕組みを作りこむことでも実現不可能ではないと考えられる。しかし、従来の中央集権的なデータベースでは特定の管理者に全権限を委ねることになりコンセンサスを得にくい。その点ブロックチェーン技術は非中央集権型ネットワークで特定の個人或いは管理者が全権限を得ることはあり得ない。また本研究では医療データに対して暗号化を行い、その復号権をブロックチェーン上に保存している。このとき医療データはセンシティブな情報でアクセス権限は厳重に管理せねばならないと考える。ブロックチェーンは強い改ざん耐性を有しまたアクセス履歴として過去のトランザクションも全て記録することから、医療データに対して不正アクセスが行えず、誰がいつ何をしたかなどのようなログも改ざんされない形で確認できる。この強い改ざん耐性と非中央集権型であるという点に着目し、ブロックチェーンを用いることとした。

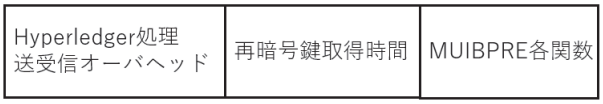

図 9. トランザクション実行時間

表 3. 実験環境

ホスト OS	Windows 10
ゲスト OS	Ubuntu Desktop 18.04.3 LTS
ゲスト OS メモリ	4GB
ゲスト OS プロセッサコア	2

表 4. トランザクション使用関数

トランザクション	実行者	使用関数
AddReKey	鍵生成局	ReKeyGen
AddEncTreatment	ユーザ（医療従事者）	Enc、ReEnc
DecryptTreatment1	ユーザ（患者）	2nd-Dec
DecryptTreatment2	ユーザ（医療従事者、解析者）	2nd-ReEnc、3rd-Dec

5. 評価実験

本章では、評価実験について述べる。評価実験では、本研究で提案したシステムのトランザクションの実行時間について性能評価を行った。図 9 に、トランザクションの実行時間について定義する。

本実験ではそれぞれに所要する実行時間を計測する。この時、Hyperledger 処理及び送受信に要するオーバーヘッドは本研究においては議論しない。よって実験内容は MUIBPRE 各関数と再暗号鍵取得に要する時間である。

また本評価実験は VMware を利用し、Windows10 上にゲスト OS として Ubuntu Desktop 18.04.3 を動作させる仮想環境を用いた。実験環境を表 3 に示す。

5.1 MUIBPRE 実行時間

本研究におけるトランザクション定義は表 2 に示した。ここで、定義したトランザクションで使用する関数を更に表 4 に示す。DectyptTreatment について、2 つに分けた理由としてそれぞれ実行者によって使用関数が異なるためである。2nd-Dec は再暗号化された暗号文を復号する関数とし、3rd-Dec は再暗号化された暗号文を更に再暗号化した暗号文を復号する関数である。また、2nd-ReEnc は再暗号化した暗号文を更に再暗号化する。クラウド上に保存する暗号文は患者の委譲秘密鍵によって復号できるため、患者は自身の秘密鍵を使って復号するのみでよい。医療従事者、解析者は一度再暗号化して自身の秘密鍵で復号する必要があるため再暗号化を行っている。ここで、AddEncTreatment における Enc は厳密にはトランザクション外のユーザの手元で行う。また暗号文復号用の関数もユーザの手元で行うため、厳密にはトランザクション実行には使われない。本実験では暗号化及び復号を行うにあたって 1、10、200、300、1000 文字のテキストファイル、jpeg を用いて実験を行った。ただしこの時、1000 文字のテキストファイルは 2.9kB であり、jpeg は 53.3kB の jpeg 形式の画像ファイルである。また本実験における実装環境、パラメータを表 5 に示す。

以下に、それぞれのトランザクションで使用する関数の計

表 5. 実験パラメータ

実装言語	Golang
NaHIBE	Hierarchical Identity based Encryption with Constant Size Ciphertext[20]
SKE	AES
SIG	wots
ID depth	10
ID max depth	ID depth + 10
ID order	6.50005E+76

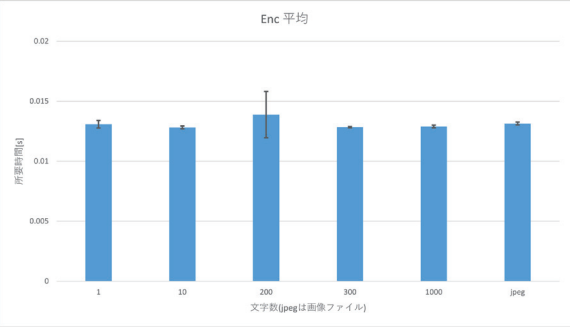

図 10. Enc

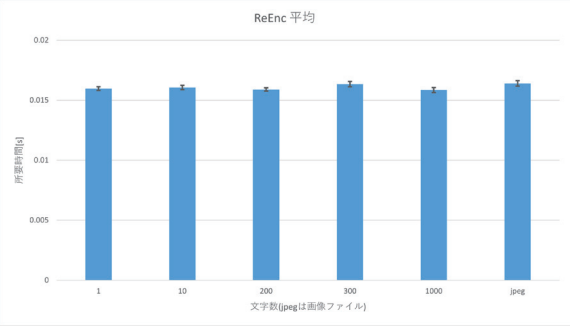

図 11. ReEnc

測結果について記述する。また、観測結果の図はすべて 10 回計測した平均とその 95%信頼区間を記録している。

5.1.1 AddReKey

ユーザの医療データに対するアクセス権限 (再暗号鍵) を発行するトランザクションである。本実験では、暗号化及び復号は行わない為、サイズの異なるテキストファイルおよび画像ファイルは使用していない。実験ではランダムな再暗号鍵を 10 回生成し、その平均所要時間を算出した。結果は標本平均 0.027194[s]、標準誤差 0.000223[s] である。ユーザビリティの観点から、再暗号鍵作成依頼を受けその作成を行うにあたって 1 秒にも満たない時間で処理を終了できる点で、関数として十分に高速であると考える。

5.1.2 AddEncTreatment

医療データを追加するトランザクションである。トランザクション実行前に医療従事者は自身のアプリケーションで医療データを暗号化しなければならない (Enc)。この時に使用する ID は自身の ID である。暗号化したデータを送信してトランザクションが実行される。図 10 に Enc の実行時間の結果を記す。次に、ReEnc の処理時間を図 11 に示す。

5.1.3 DecryptTreatment1

DecryptTreatment1 は患者が自身の医療データを取得する際に用いるトランザクションである。この時、実際のトランザクションでは MUIBPRE の関数は用いていないため 2nd-Dec のみの評価になる。図 12 に実験結果を記す。

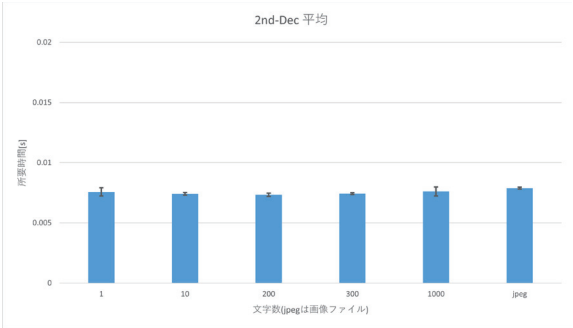

図 12. 2nd-Dec

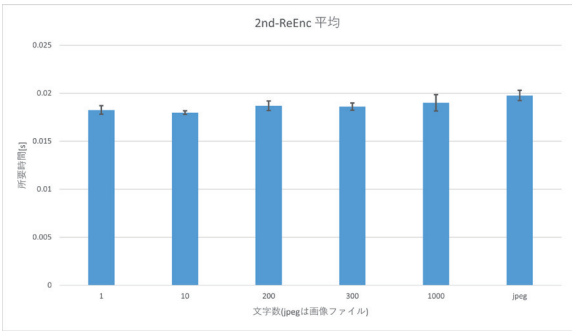

図 13. 2nd-ReEnc

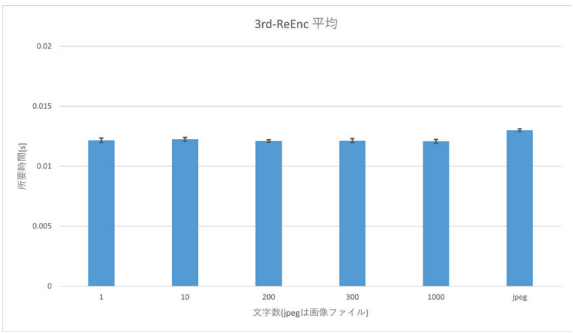

図 14. 3rd-Dec

5.1.4 DecryptTreatment2

DectyptTreatment2 は医療従事者及び解析者が患者の医療データを取得する際に行うトランザクションである。患者の医療データを取得するとき、トランザクションを実行し、得られたデータに対して 3rd-Dec を用いて復号を行う。まず図 13 に 2nd-ReEnc の実行時間を示す。次に図 14 に 3rd-Dec の処理時間を示す。

5.2 再暗号鍵取得時間

再暗号鍵の取得時間を計測した。再暗号鍵はブロックチェーン上に記録している為、記録情報が膨大になるとその検索性能に影響があると考える。本提案手法における鍵探索が必要なトランザクションとそうでないトランザクションを表 6 に示す。表のとおり、今回は再暗号鍵の取得は AddEncTreatment、DecryptTreatment2 のみである。また本評価実験はコンソーシアム型ブロックチェーンとしのプラットフォームとして広く活用されている hyperledger Fabric を利用した。10000、30000、100000 個の再暗号鍵を生成しそれぞれの個数からランダムな 1 つの鍵の取得時間を表 7 に示す。

表 6. 権限探索の有無

トランザクション	権限探索
AddReKey	×
AddEncTreatment	○
DecryptTreatment1	×
DecryptTreatment2	○

表 7. 再暗号鍵取得時間

再暗号鍵数	回数	平均取得時間 [s]
10000	100	0.000886
30000	100	0.000876
100000	100	0.001149

5.3 考察

本実験では暗号化するファイルのサイズを変えてそれぞれに所要する時間を計測した。医療データには大容量のファイルも含まれるためファイルサイズの増加に伴い変化する暗号化及び復号の所要時間を計測することは重要であると考える。ファイルサイズの増加に伴い、暗号化及び復号に所要する時間も増加する傾向にある、と予測し実験を行ったが、結果としてはファイルサイズの変化に伴った暗号化及び復号の所要時間に変化はなかった。本研究で使用した MUIBPRE は実際に暗号化を施すデータを共通鍵暗号方式で暗号化している。暗号化に利用した共通鍵を公開鍵暗号方式ベースである階層型 ID ベース暗号で暗号化することでプロキシ再暗号を実現した。ここで、一般に共通鍵による暗号化の処理は公開鍵を用いた処理と比較しはるかに短時間に終了する [38]。よって暗号化するファイルサイズの増減に伴う MUIBPRE 全体の処理時間への影響は少ないと考えられる。図 15、16 にそれぞれ暗号化、復号に伴う MUIBPRE 全体の処理時間と共通鍵暗号方式による処理時間を比較し示す。また、暗号化及び復号に伴う処理時間をそれぞれ確認してもどの処理も 1 秒に満たず終了している。以上を踏まえ、ユーザビリティを考慮すると MUIBPRE による医療データの暗号化は問題ないと考える。

再暗号鍵の取得時間について確認する。再暗号鍵はユーザの医療データに対するアクセス権限である。よって膨大な数の再暗号鍵が生成されると考えられる。このことから実験では鍵探索の時間を計測するにあたって鍵の数を増やしながら実験を行った。再暗号鍵の数の増加に伴い探索時間も増加すると考えたが表 7 から読み取れるように増加傾向はなかった。これは、hyperledger Fabric が State DB と呼ばれるデータベースを使用しており、データベース上のインデックス付きのデータを検索しているためであると考える。データベースにおけるインデックス付きデータの検索は極めて高速であり著しく多い数でない限り検索性能に影響を及ぼさないと考える。State DB は、ブロックチェーンの最新の状態を記録するデータベースである [39]。本実験ではその恩恵はなかったが、ある特定の医療データに対して行った操作をトランザクションとして記録しそのログを確認する際には大いに役立つと考える。

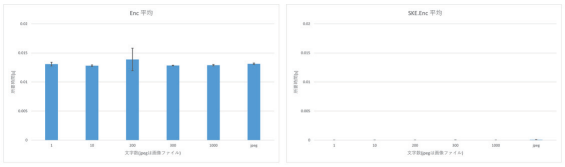

図 15. enc-comparison

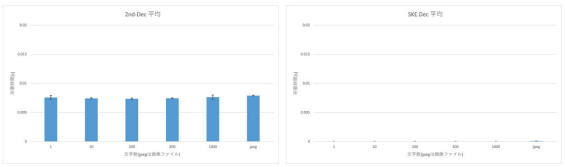

図 16. dec-comparison

6. 結論

6.1 まとめ

医療業界において、個人の医療データは各医療機関ごとに管理され、その共有性について課題を抱える。医療データを患者個人のデータとして共有化を行うことで、希少疾患に関する研究活動や新薬開発に役立たせることができ、患者の研究参加を促すと考えられる。また過去の医療履歴や日々のヘルスケア情報を容易に役立たせることができると考える。

そこで本研究では、非中央集権型で改ざん耐性を有するブロックチェーン技術を用いて医療データに対するアクセス権限を管理する手法を提案した。提案にあたって医療情報を患者が自律的に管理することを想定した。この観点から患者データを患者の秘密鍵を用いて復号できる暗号文としてクラウドに保存を行う。医療データには CT 画像や X 線画像等大容量のデータが含まれるためスケーラビリティの観点から医療データをブロックチェーン上で管理することは現実的ではないとし、医療情報に対するアクセス権限 (再暗号鍵) と操作 (トランザクション) を記録することで暗号化された医療データに対して不正にアクセスができないと考える。医療情報の暗号化及びアクセス権限の管理には MUIBPRE[21] を活用した。

評価実験では、ユーザビリティの観点から提案手法におけるトランザクション処理の実行時間を計測し評価した。それぞれのトランザクションはそこで使用している MUIBPRE[21] の各関数による処理時間及び再暗号鍵取得に要する時間を計測した。再暗号鍵取得時間の計測には hyperledger Fabric を用いてランダムな再暗号鍵を生成しその検索を行うチェーンコードを実装することで結果を得た。結果としてデータのサイズによらず暗号化、復号共に処理に膨大な時間を有さないとし、利用するにあたって問題のない性能であることを確認した。

6.2 展望と今後の課題

展望として、本研究では医療データをブロックチェーン上で管理する手法を提案したがこれは様々な分野にも応用できると考えられる。例えば介護施設において取り扱う介護情報をブロックチェーン上で適用できればその透明性を示すことができる。

本研究ではアクセス権限として再暗号鍵をブロックチェーンに保存する手法を用いている。一般的にブロックチェーンは大容量のデータを共有することには向いていないとされて

いることからブロックチェーン上に記録されるデータのサイズ及びその数、増加数について想定する必要がある。データの耐消去性についても解決する必要性がある。本研究では医療データ本体を暗号化し、オフチェーン上で管理することを想定している。よってブロックチェーンのようにデータが共有されていない為、一つの医療機関内で保管しているデータが消去された場合、復元が不可能となってしまう。また、今回扱った医療データは極めてセンシティブな情報でその扱いは慎重でなくてはならない。より精密なセキュリティ評価が必要であり、実社会での実装を想定したプロトタイプでの実験を行う必要がある。

更に実装を行う以前に患者の医療データの所有者に関する認知など法的課題も解決しなければならない。本研究では医療データを患者個人のデータとして本人が自律的に管理することを目的として研究を行ったが、現在において医療データは作成を行った医療機関で管理する情報である、とした考え方が一般である [40]。我が国の通説・判例でも主として専ら医師の裁量権に帰属するものとし、開示請求権を患者に与える法的根拠はないとしている。患者が自律的に自身の医療データを管理するにあたって、医療情報の法的位置づけや医師の医療データに対する認知などの課題を解決する必要がある。

参考文献

1) WE ARE HERE 難病患者間での情報共有を at:`https://nambyo.net/` (accessed 2020/12/28)
2) J-RARE — 難病の患者情報登録サイト — J-RARE: 難病の患者情報登録サイト at:`https://j-rare.net/`(accessed 2020/12/28)
3) 水島洋: エストニアの先進事例等を踏まえた「ブロックチェーン技術」の医療・データ管理への応用, 情報機構主催セミナー, 2019/6/14.
4) 森田端樹 西村邦裕: 患者が主体となった患者レジストリに関する検討, 厚生労働科学研究費補助金（難治性疾患等克服研究事業） 分担研究報告書 .
5) 小松康宏: 患者参加型医療が医療の在り方を変える -21世紀医療のパラダイムシフト, 国民生活研究第 *59* 号第 *2* 号 [特集]: 医療と消費者〜コミュケーションの重要性〜, 2019.
6) 萱原正彬, 本田祐一, 山田達大: ブロックチェーンとプロキシ再暗号化を用いた共有範囲設定可能な医療情報管理, *DEIM Forum 2019 D1-2.*, 2019.
7) 厚生労働省: 医療情報システムの安全管理に関するガイドライン, Vol. 5, 2017.
8) Ahmed F. Hussein, N. ArunKumar, Gustavo Ramirez-Gonzalez: A medical records managing and securing blockchain based system supported by a Genetic Algorithm and Discrete Wavelet Transform, *Cognitive Systems Research 52, 1-11.*, 2018.
9) Haibo Tian, Jiejie He, Yong Ding: Medical Data Management on Blockchain with Privacy, *Journal of Medical Systems 43* Article number: 26, 2019.
10) Zheng, Y., Hardjono, T., and Pieprzyk, J.: Sibling intractable function families and their applications, In: *International Conference on theTheory and Application of Cryptology*, pp.124-138, 1991.
11) Alex Roehrs, Cristiano André da Costa, Rodrigo da Rosa Righi, Valter Ferreira da Silva: Analyzinig the performance of a blockchain-based personal health record implementation, Journal of Biomedical Informatics Volume 92, April, 2019.
12) Alevtina Dubovitskaya, Zhigang Xu, Samuel Ryu: Secure and trustable electronic medical records sharing using blockchain, *AMIA Annual Symposium Proceedings, Vol. 2017, p. 650. American Medical Informatics Association,* 2017.
13) Matt Blaze, Gerrit Bleumer, and Martin Strauss: Divertible protocols and atomic proxy cryptography, *the proceeding of the International Conference on the Theory and Applications of Cryptographic Techniques, pp.127-144. springer*, 1998.
14) D.Boneh and X.Boyen: Efficient selective-ID identity based encryption without random oracles, In C.Cachin and J.Camenisch, editors, *Proceedings of Eurocrypt 2004, volume 3027 of LNCS, pages 223-38. Springer*, 2004.
15) S. Mitsunari, R. Sakai, and M. Kasahara: A new traitor tracing. *IEICE Transactions Fundamentals, E85-A(2):481-84*, 2002.
16) A. Shamir: Identity-based cryptosystems and signature schemes, In G. Blakley and D. Chaum, editors, *Proceedings of Crypto 1984*, volume 196 of *LNCS*, pages 47–53. Springer, 1984.
17) D. Boneh and M. Franklin: Identity-based encryption from the Weil pairing, In J. Kilian, editor, *Proceedings of Crypto 2001*, volume 2139 of *LNCS*, pages 213–29. Springer, 2001.
18) J. Horwitz and B. Lynn: Towards hierarchical identity-based encryption. In L. Knudsen, editor, *Proceedings of Eurocrypt 2002*, volume 2332 of *LNCS*, pages 466–81. Springer, 2002.
19) C. Gentry and A. Silverberg: Hierarchical ID-based cryptography. In Y. Zheng, editor, *Proceedings of Asiacrypt 2002*, volume 2501 of *LNCS*, pages 548–66, 2002.
20) Dan Boneh, Xavier Boyen, and Eu-Jin Goh: Hierarchical Identity Based Encryption with Constant Size Ciphertext, *Advances in Cryptology-EUROCRYPT 2005*, Lecture Notes in Computer Science, Springer-Verlang, 2005.
21) Jun Shao、Zhenfu Cao: Multi-use unidirectional identity-based proxy re-encryption from hierarchical

identity-based encryption, *Information Sciences 206*, 83-95, 2012.

22) B.Waters: Dual System encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions, *in: CRYPTO 2009*, 2009.

23) J. Al-Jaroodi and N. Mohamed: Blockchain in Industries: A Survey. *IEEE Access 7*, 36500-36515, 2019.

24) Nakamoto, Satoshi: Bitcoin: A peer-to-peer electronic cash system, 2008.

25) Blockchain Biz 電子署名, https://gaiax-blockchain.com/signature (accessed 2021/1/7).

26) Blockchain Biz スマートコントラクト, https://gaiax-blockchain.com/smart-contract (accessed 2021/1/7).

27) Zheng, Zibin, et al: An overview of blockchain technology: Architecture, consensus, and future trends, *IEEE 6th International Congress on Big Data*, 557-564, 2017.

28) Zheng, Zibin, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang: Blockchain challenges and opportunities: A survey, *International Journal of Web and Grid Services 14* No. 4, 352-375, 2018.

29) Ethereum Project available at:https://www.ethereum.org/ (accessed 2021/1/7).

30) Hyperledger - Open Source Blockchain Technologies available at:https://www.hyperledger.org/ (accessed 2021/1/7).

31) Factom at:https://www.factom.com/(accessed 2021/1/7).

32) Blockchain Healthcare Review at:https://blockchainhealthcarereview.com/(accessed 2021/1/7).

33) PokitDok at:https://pokitdok.com/(accessed 2021/1/7).

34) uPort.me at:https://www.uport.me/ (accessed 2021/1/7).

35) 電子カルテシステム WATATSUMI. https://www.corecreate.com/02_01_izanami.html.

36) 日本年金機構 at:https://www.nenkin.go.jp/n_net/ (accessed 2021/1/11)

37) マイポータル at:https://myna.go.jp/SCK0101_01_001/SCK0101_01_001_InitDiscsys.form

38) KEYFACTOR JUN 17, 2020 2:04:12 PM When to Use Symmetric Encryption vs. Asymmetric Encryption at:https://blog.keyfactor.com/symmetric-vs-asymmetric-encryption

39) Ledger — hyperledger-fabricdocs master documentation available at:https://hyperledger-fabric.readthedocs.io/en/release-1.4/ledger.html (accessed 2021/01/12).

40) 東洋経済 online 病院の「診療データ」は一体誰のものなのか　改正個人情報保護法施行で問われること. p2. 2017/5/30. https://toyokeizai.net/articles/-/173997?page=2 (accessed 2021/1/18).