電子カルテと紐づけ解析可能とする生理計測データ集積の PKI とプロキシ再暗号化を用いた実現可能性確認

小林うらら† Le Hieu Hanh† 横田 治夫†

† 東京工業大学情報理工学院情報工学系 〒 152-8550 東京都目黒区大岡山 2-12-1
E-mail: †{urarak,hanhlh}@de.cs.titech.ac.jp, †yokota@cs.titech.ac.jp

あらまし 近年, 電子カルテ解析による医療支援の研究に加え, ウェアラブルな IoMT (Internet of Medical Things) 端末の普及が進んでいる. IoMT 端末で取得した生理計測データを集積し, 電子カルテと紐づけることで, 症状経過を加味した解析が可能になるが, プライバシーを担保した電子カルテとの紐付け可能な生理計測データ集積方法は確立されていない. 本研究は, 紐付けのための個人を特定可能な匿名 ID と, 医療従事者以外が解析できるように公開鍵暗号基盤 (PKI) とプロキシ再暗号化を用いて, 秘匿性レベルに応じた共有範囲を管理可能な生理計測データ集積システムを提案する. 実データセットを用いてデータ量や同時アクセス数，FHIR データフォーマットを前提にした暗号化単位といった条件を変化させて登録，取得，さらに MEWS スコアやシーケンシャルパターンマイニングの解析に要する時間を計測し, 暗号処理コストを含めた実現可能性と解析に適した暗号化単位を確認する．

キーワード 医療・ヘルスケア, IoT, プロキシ再暗号化, 生理計測データ, 電子カルテ

1 はじめに

1.1 研究背景

近年，電子カルテが広く普及しており，電子カルテ解析による医療支援の研究が進行している [1]．電子カルテ解析の例として，ある疾病の患者に対して頻出医療指示シーケンスの抽出を行うことによる，典型的な医療行為の流れであるクリニカルパスの生成支援があげられる [2].

また，心拍数や酸素飽和度といった生理計測データを測定可能なやウェアラブルな IoMT (Internet of Medical Things) 端末の普及が進んでいる [3]．IoMT 端末を利用することで，生理計測データの定期的かつ安価な記録が可能となり，遠隔医療における患者のモニタリングや，蓄積されたデータの解析が進んでいる．例えば，ウェアラブルデバイスから収集したデータを用いて初期症状から COVID-19 患者の重症度を予測する研究があげられる [4] [5] [6] [7].

IoMT 端末で取得した生理計測データを集積し，電子カルテと紐づけて解析することで，病院外での症状経過を加味した解析が可能になる．COVID-19 の場合では，入院前の体温や酸素飽和度についてのデータがあることで，より実際の症状経過に近いデータを元に解析できる．例えばこれまでは電子カルテの情報から得られる医療指示だけにシーケンシャルパターンマイニングを適応させていたのに対して，生理計測データと医療指示の両方にシーケンシャルパターンマイニングを適応させることができる．

集積した生理計測データと電子カルテとを紐付けた解析において検討すべき点として，生理計測データと電子カルテではデータの生成頻度や解析の際の読み出しの単位が違うことがある．たとえば生理計測データはウェアラブル端末が毎秒心拍数を測定しデータが毎秒生成されるのに対して，電子カルテは入院患者であれば 1 日数回診察や検査を受け，通院患者であれば数週間〜数ヶ月に 1 回診察を受けるなど，データ生成頻度が生理計測データに比べ少なく，その結果解析の際の読み出し単位も異なる．そのため，生理計測データをどのような単位で集積するかが解析の際に重要である．

生理計測データや電子カルテは医療情報なので高水準のプライバシー保護が必要であり，特に解析するためには情報の共有範囲が設定できる必要がある．記録した自身の生理計測データを，解析者には共有せず医師にだけ共有したいという場合や，解析者に共有を許可する情報と許可しない情報が存在するからである．このように秘匿性の程度に幅があるため，共有範囲を設定できるような暗号技術が好ましいと言える．

1.2 本研究の目的

本研究は，プライバシーを担保した上で，電子カルテと紐づけて共に解析できるような生理計測データ集積システムを構築することを目的とする．また電子カルテと紐付けた解析に適した生理計測データの集積単位の検討も行う．そして提案システムのプロトタイピングを作成し，実データセットを用いてデータ量や同時アクセス数，集積単位といった条件を変化させて登録，取得，解析に要する時間を計測し暗号処理コストを含めた実現可能性と解析に適した暗号化単位を確認する.

1.3 本論文の構成

本論文は以下の通りに構成される．2 節では本研究の前提となる概念を背景知識として説明し関連研究を紹介する．3 節では個人を特定可能な匿名 ID と，PKI とプロキシ再暗号化を用いた生理計測データ集積システムの概要を提案手法として述べる．4 節では 3 節の手法をプロトタイピングし，データ構造や

暗号化単位について述べる．5 節で 4 節で作成したプロトタイプの動作を確認し，データの登録，取得，解析に関する実行時間の評価を行う．最後に 6 節で本論文のまとめと今後の課題を述べる．

2 背景知識・関連研究

本節では背景知識として，電子カルテの規格である HL7 FHIR (Health Level Seven Fast Healthcare Interoperability Resources)，プロキシ再暗号化について説明を行う．さらに関連研究についても述べる．

2.1 HL7 FHIR

HL7 はコンピュータ間での医療文書情報のデータ連携を標準化するための国際規格で，テキスト形式の V2，XML 形式の V3 と CDA，Web 通信の FHIR の 4 種類がある. それぞれデータ構造のルールを定めており，FHIR のみ Web 通信での連携を前提とし RESTful API を採用している [8].

リソースは FHIR で医療における何らかの事物の抽象的な概念を表現するもので，医療情報交換を行う上でのやりとりする情報等の最小単位である [9]．リソースには情報の種類によって複数タイプがあり，患者や検査結果，診断情報，処方要求，薬剤などがある [10]．Bundle は 1 つのやりとりで，複数のリソースの集合体を使用した医療情報の処理を行う場合に使われるリソースである [9].

2.2 プロキシ再暗号化

再暗号化とはあるユーザのために暗号化された暗号文を別のユーザが復号可能な暗号文に変換することである. 特に, 第三者であるプロキシによって再暗号化が行われる暗号体系は, プロキシ再暗号化と呼ばれる [11].

プロキシ再暗号化によって, データの所有者は全共有ユーザの鍵情報を把握したり，暗号化したデータを復号したりすることなくデータを共有することができる．具体例として図 1 のように，従来の暗号化方式で A 宛てに暗号化された暗号文を B 宛ての暗号文に変換する場合とプロキシ再暗号化を比較する．この時，A の秘密鍵 sk_A を B に渡すことはセキュリティ上望ましくない．したがって A は自分の秘密鍵 sk_A を用いて一度暗号文を復号した後，B の公開鍵 pk_B を用いて再度暗号化する必要がある．プロキシ再暗号化可能な暗号では，A から B への再暗号化鍵 $rkA \to B$ を用いることでプロキシが A 宛ての暗号文を復号することができない状況で B 宛ての暗号文に変換することができる [12].

プロキシ再暗号化可能な暗号の 1 つに ElGamal 暗号をベースして Blaze らにより考案された BBS 方式がある [13]. BBS 暗号のアルゴリズムは次のように表現される [12].

（1） セットアップ

安全素数 $p = 2q + 1$，$\mathbb{Z}_p^*$ の原始元 g を決め，公開パラメータとする．

（2） 鍵生成

$a \to \mathbb{Z}$ をランダムに選び, $sk_a = a, pk_A = g^a \pmod p$ をそれぞれ

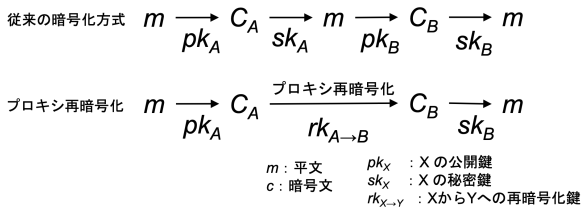

図 1: 従来の暗号化方式とプロキシ再暗号化の比較

A の秘密鍵，公開鍵とする．

（3） 再暗号化鍵を生成

sk_A, sk_B を入力として，$rk_{A\to B} = \frac{sk_B}{sk_A} = \frac{b}{a}$ を A 宛ての暗号文から B 宛ての暗号文に変換する再暗号化鍵とする．

（4） 暗号化

平文 $m \in \mathbb{Z}_p^*$ を pk_A で次のように暗号化した暗号文 c_A を A 宛ての暗号文とする．

$$c_A = (mg^s, pk_A^s) = (mg^s, g^{as}) \pmod p$$

（5） 再暗号化

c_A と　$rk_{A\to B}$ を入力として，以下のように c_B へと再暗号化を行う．

$$c_B = (mg^s, (pk_A^s)^{rk_{A\to B}}) = (mg^s, (g^{as})^{\frac{b}{a}}) = (mg^s, g^{bs})$$
$$= (mg^s, pk_B^s) \pmod p$$

（6） 復号

c_A と sk_A により以下のように復号する．

$$\frac{mg^s}{(g^a s)(sk_A^{-1})} = \frac{mg^s}{(g^{as})^{a-1}} = m \pmod p$$

2.3 関 連 研 究

2.3.1 プロキシ再暗号化を用いた医療情報管理

BBS 暗号を利用した情報共有管理システムとして児玉氏らの研究 [11] と萱原氏らの研究 [12] が挙げられる．児玉氏らは共有情報に対し，ロジカルなクラスを単位とした暗号化を行うことでクラス単位でのアクセス制御を可能にする手法を提案し，暗号化されたデータベースとユーザ間にプロキシを配置した情報共有管理システムを作成した．萱原氏らはプロキシ再暗号化可能なブロックチェーンベースの電子カルテのモデルを提案した．このモデルでは，プロキシ再暗号化を用いてユーザのアクセスレベルを設定できる．また，プロキシサーバの機能をブロックチェーン内のスマートコントラクトとして記述することで，プロキシサーバをブロックチェーンネットワークと一体化させた．

2.3.2 患者相互参照方法について

患者相互参照方法として Kho らの研究が挙げられる [14]．名前や生年月日，SSN (Social Security Number) といった患者識別情報を組み合わせてハッシュ化することで，シカゴの複数施設にまたがる電子カルテデータを紐づけている．この研究では，SSN を使用できなかった施設が存在したことや，他人の SSN を不正利用している人の存在を考慮したために，SSN の一致のみで同一患者であるかを判断していない．

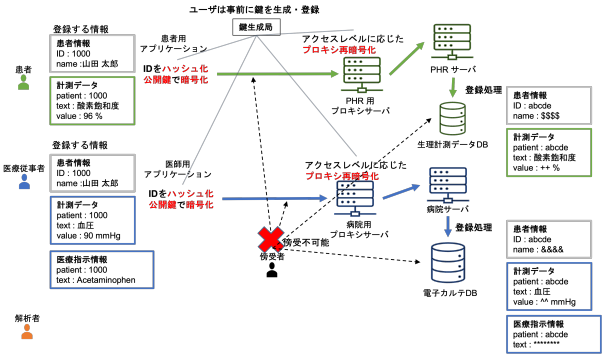

図 2: 提案システム (登録)

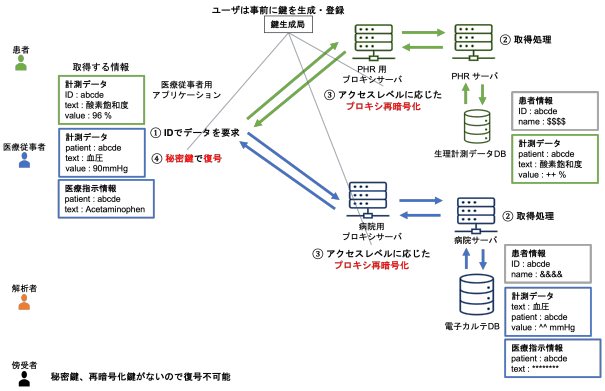

図 3: 提案システム (医療従事者による取得)

また，標準化した地域連携における診療情報共有の仕組みとして IHE PIX/PDQ (Patient ID Cross Referencing / Patient Demographic Query) がある [15]．これは地域内の患者を一意に識別する地域患者 ID を管理し，地域内の各医療施設がその地域患者 ID を参照する方法である．これらの参照方法は，医療機関ごとの医療情報を紐づけることを目的としており，病院外での生理計測データの紐付けについては議論されていない．

3 提案システム

本節ではまず紐付けのために個人を特定可能な匿名 ID と，PKI・プロキシ再暗号化を用いてプライバシーを担保した生理計測データ集積システムの概要を示す．そして，電子カルテとの紐づけ方法について，プライバシー担保のための PKI とプロキシ再暗号化について説明する．

3.1 システム概要

図 2，3 が提案システム全体を表している．本システムは，さまざまなユーザが自身のアプリケーションを通じて，医療情報を登録，取得，解析することを想定している．

ユーザの種類は，患者，医療従事者，解析者，傍受者の 4 者である．患者と医療従事者は医療情報の登録，取得を行い，解析者は医療情報の取得と解析を行う. 傍受者は医療情報を傍受しようとする．

また，扱う医療情報は，患者の生年月日や氏名などの患者情報，計測した生理計測データ，医療指示情報の 3 種類である．患者情報と生理計測データは，患者と医療従事者の両者が登録し，医療指示情報は医療従事者のみが登録する．

上記の医療情報の管理は，PHR (Personal Health Record) サーバと病院サーバの 2 種類のサーバが行う．PHR サーバは患者が登録する患者情報と生理計測データについて管理し，病院サーバは医療従事者が登録する患者情報，生理計測データ，医療指示情報について管理する．PHR サーバと病院サーバにはそれぞれ，プロキシ再暗号化を行うプロキシサーバが存在する．

3.2 電子カルテとの紐づけ

PHR サーバに登録される情報と病院サーバに登録される情報との紐付けには，個人を特定可能な ID を用いる．個人を特定可能な ID を全ての医療情報が保持し，その ID を参照することで紐付けを行う．

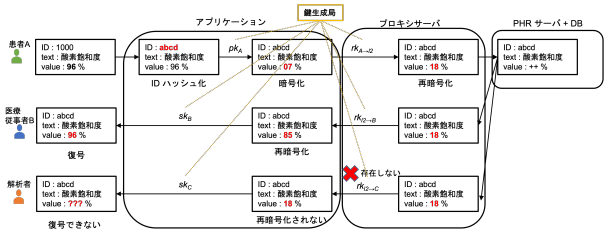

図 4: 暗号化と復号の流れ

また，2.1 節で述べた HL7 FHIR を患者が測定した生理計測データの記録の規格に用いることで，相互参照を可能にする．

3.3 プライバシー担保

プライバシーを担保するために，ハッシュ関数，PKI，プロキシ再暗号化を使用する．まずハッシュ関数によって 3.2 節で述べた個人を特定可能な ID をハッシュ化し保護する．そして，登録者の公開鍵によって医療情報を暗号化する．さらに，プロキシ再暗号化を行い医療情報の共有範囲を設定する．公開鍵による暗号化は自身のアプリケーションにて行い，プロキシ再暗号化は，PHR サーバと病院サーバそれぞれのプロキシサーバで行う．

3.4 医療情報の登録から解析まで

医療情報の暗号化と復号の流れを図 4 に示す.

3.4.1 セットアップ

ユーザはユーザ登録時に信頼できる第三者の鍵生成局に秘密鍵と公開鍵，そして患者・医療従事者・解析者といったユーザの種類に応じて必要な再暗号化鍵の作成を要求する．作成された鍵は鍵生成局が保管する．

3.4.2 医療情報の登録

患者 A が生理計測データを登録するときには，はじめに自身のアプリケーションにてデータに含まれている自身の ID をハッシュ化し，その他のデータについては自らの公開鍵 pk_A を鍵生成局から取得し暗号化する．暗号化されたデータがプロキシサーバに送信され，プロキシサーバは医療情報のアクセスレベル l を読み取り，鍵生成局から適切な再暗号化鍵 $rk_{A \to l}$ を取得する．そして $rk_{A \to l}$ を用いて再暗号化されたデータをサーバ

へ送信しデータベースに記録する．通信路上やデータベース内にはハッシュ化された ID と，暗号化されたデータしか存在しないので，傍受者が個人を特定したり平文の情報を得ることは不可能である．

3.4.3 医療情報の取得

医療従事者 B がデータを取得するときは，プロキシサーバにハッシュ化した ID を使ってデータ要求すると，プロキシサーバが復号したいデータに記録されているアクセスレベル l を読み取る．そして読み取ったアクセスレベルの再暗号化鍵 $rk_{l \to B}$ を鍵生成局から取得し再暗号化を施す．その後，再暗号化されたデータは医療従事者 B のアプリケーションに送られ自らの秘密鍵 sk_B を鍵生成局から取得し，それを用いて復号を行う．また，復号が許可されていない解析者 C がデータの復号を試みたとする．プロキシサーバが復号したいデータに記録されているアクセスレベル l を読み取り，それに適した再暗号化鍵 $rk_{l \to C}$ を鍵生成局から取得しようとするが，そのような再暗号化鍵は存在しない．その場合再暗号化は行われず，そのまま解析者 C のアプリケーションにデータが送られる．その後，自身の秘密鍵 sk_C を鍵生成局から取得し復号を試みるが，秘密鍵で復号できるように再暗号化されていないため復号は失敗する．

4 プロトタイピング

本節では，3 節で提案した生理計測データ集積システムのプロトタイプを HAPI FHIR [16] [17] を用いて作成する．HAPI FHIR とは Canada の UHN (University Health Network) が立ち上げた Java ベースの FHIR オープンソースライブラリである．

4.1 プロトタイピングの概要

本システムは大きく分けて，ユーザー用アプリケーション，プロキシサーバ，FHIR サーバ，鍵生成局の 4 つが存在する．ユーザ用アプリケーションは患者用，医療従事者用，解析者用の 3 つ存在し，プロキシサーバと FHIR サーバは PHR 用，病院用の 2 つ存在する．

本プロトタイプのユーザ用アプリケーションは Node.js によって記述されており，実行すると患者用，医療従事者用，解析者用でそれぞれローカルホストのポート番号 5100 番，6100 番，7100 番を開く．

プロキシサーバも Node.js によって記述されており，実行すると PHR プロキシサーバと病院プロキシサーバでそれぞれローカルホストのポート番号 8000 番，9000 番を開く．

FHIR サーバは，HAPI FHIR サーバを PHR サーバとしてローカルホストのポート番号 8080 番に，病院サーバはローカルホストのポート番号 9080 番にたてる．

鍵生成局は Node.js によって記述されており実行するとローカルホストのポート番号 3000 番を開く．鍵生成局ではハッシュ化された ID と共に，ユーザの秘密鍵，公開鍵，及び再暗号化鍵が保管されているため，常に信頼された第三者であることが前提条件となる．

4.2 データ構造

本研究で扱う 3 種類の医療情報はそれぞれ，患者情報は Patient リソース，生理計測データは Observation リソース，医療指示情報は Medication Request リソースで表現される．生理計測データと医療指示情報については，1 回の登録操作で複数件数の登録を行うことを想定しており，2.1 節で述べた Bundle を使用する．

アクセスレベルは $l1$，$l2$，$l3$ の 3 タイプを用意している．それぞれのデータの共有範囲は表 1 のようになっている．アクセスレベルも FHIR フォーマット内に記述し登録する．

4.3 暗号化単位

4.2 節のようにデータを記述すると，暗号化の単位は患者情報については各情報ごと，リソースごとの 2 パターン，生理計測データと医療指示情報については，Bundle ごとを加えた 3 パターンが検討できる．暗号化単位と集積単位との関係は，各情報ごと・リソースごとは 1 件ずつ集積することに対応し，Bundle ごとは複数件ずつ集積することに対応する．Observation リソースについてそれぞれ暗号化単位を変化させた例を図 5 に示す．各情報ごと，及び，リソースごと暗号化する際には，4.2 節で述べた Patient リソースを参照するための項目と，データ取得時に日時の大小比較を可能にするために Date 項目は暗号化の対象に含めないこととする．Bundle ごとに暗号化する場合は，複数リソースをまとめて暗号化したものを Basic リソースを利用して記述する．Basic リソースは FHIR でまだ定義されていない概念を表現するカスタムリソースである．実際に実装する際には図 5 のように FHIR のフォーマットも含めて暗号化するのではなく，アプリケーションやプロキシサーバが json 形式で受け取った値だけを暗号化した後，FHIR のフォーマットに整形する．

暗号化単位を変化させることで共有範囲設定の粒度が変化する．例えば患者情報について各情報ごとに暗号化した場合とリソースごとに暗号化した場合を比較すると，各情報ごとに暗号化した場合は，名前，生年月日といった各情報それぞれについて共有範囲を設定することができる．一方でリソースごとに暗号化した場合は，それらは全て同じ共有範囲が設定される．また，各情報ごとに暗号化した場合は，生理計測データについて計測値だけを暗号化するのか，または，計測値に加えて計測内容も暗号化するのかといった暗号化の有無も区別することができる．

生理計測データを Bundle ごとに暗号化した場合は，複数件登録すると，それらは全て同じ共有範囲が設定される．また，日時情報も暗号化するので，日時を使った検索をすることができない．しかし，複数リソースをまとめて暗号化して Basic リソースを作成する際に，例えば同じ日付のものをまとめた場合などは Basic リソースの created というキーに日付を登録することができ，日付を検索パラメータとして使用できる．

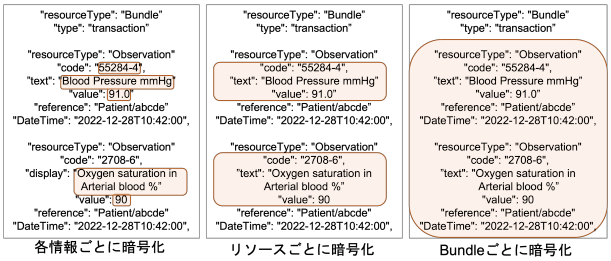

図 5: 暗号化単位

表 1: アクセスレベルと共有範囲

アクセスレベル	共有範囲
l1	暗号化した本人のみが復号可能
l2	患者及び医療従事者のみが復号可能
l3	患者，医療従事者，解析者が復号可能

表 2: 実験環境

CPU	AMD Opteron(tm) Processor 4184
Memory	32GB
OS	Ubuntu Server 17.10 x86-64
Docker	Docker version 20.10.22, build 3a2c30b
HAPI FHIR	HAPI FHIR 6.2.0
FHIR	4.0.1
Node.js	v16.19.0

5 実　　験

本節ではデータの登録，取得，解析に関する実行時間の評価を行う．暗号化・復号を施す情報のデータ量や同時アクセス数によって，データの登録，取得にかかる時間がどのように変化するか測定することで，暗号化・復号化のオーバーヘッドを確認し提案システムの実現可能性について検討する．また，暗号化の単位を変化させて解析で必要とされるデータを取得するのにかかる時間を測定し，どの暗号化単位が解析に適しているのかを検証する．なお実験に際し動作確認を行ったところ，アクセスレベルに応じて暗号化および復号が適切に行われていることが確認できた．本実験の実行環境を表 2 に示す．本実験で用いた ID，生年月日，名前の患者情報はオープンソースの合成 FHIR データジェネレーターによって作成されたものを使用した [18].

5.1 登録・取得数と時間の関係

5.1.1 実 験 方 法

一度の登録・取得操作で登録・取得するデータ件数を変化させて，各情報ごと，リソースごと，Bundle ごとに暗号化した場合それぞれの全体処理時間と各フェーズごとの所要時間を計測する．フェーズは暗号化，復号に対し以下の 3 フェーズを考える．

- 登録 ：アプリケーションでの暗号化，プロキシサーバでの再暗号化，サーバでの DB へのリソース登録
- 取得 ：サーバでの DB からのリソース取得処理，プロキシサーバでの再暗号化，アプリケーションでの復号

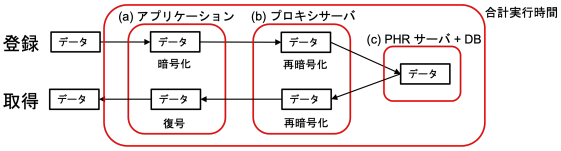

図 6: 実行時間を計測する各フェーズ

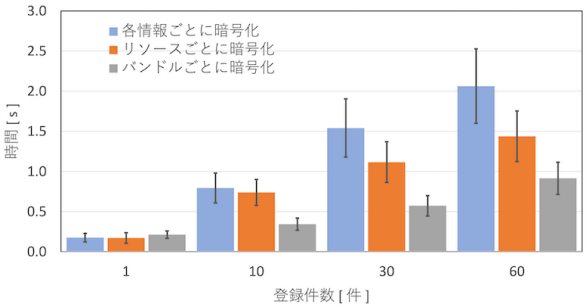

図 7: 一度に登録する件数と登録処理合計時間の関係

各フェーズを図示すると図 6 のようになり，登録全体処理時間はユーザがアプリケーションに登録要求をしてからサーバが DB への登録処理を終えたレスポンスを取得するまで，取得全体処理時間はユーザがアプリケーションにデータ取得要求をしてから取得したデータを手元で復号し終えるまでである．

一度の登録操作で登録する件数とその処理時間の関係はデータの送信頻度に影響する．例えば，毎秒計測して生成されたデータを生成された直後に 1 件ずつサーバへ送信する場合，1 秒に 1 回 1 件のデータの登録処理が行われるのに対して，30 件のデータをまとめて送信する場合は，30 秒に 1 回 30 件のデータの登録処理が行われ，前者の状況では登録処理は 1 秒以内，後者では 30 秒以内に行われる必要がある．

実験データにはウェアラブルデバイスで毎秒計測された心拍数のデータセットを使用した [19] [20]．本データセットは 1 件あたり暗号化を施す部分のバイト数が各情報ごと・リソースごとに暗号化した場合は 25 byte，Bundle ごとでは 53 byte であった．公開パラメータの安全素数 $p = 902,723$ を用いて，一度の登録操作で登録する件数は 1，10，30，60 件と変化させた．また，取得については 10，100，300，600 件のデータを一度に取得した．それぞれの件数に対して実験を 10 回繰り返し，平均所要時間と本実験における 95 %信頼区間を記録した．なお，これ以降の実験では全て同じ安全素数を用い，10 回実験を繰り返し平均所要時間と本実験における 95 %信頼区間を記録した．

5.1.2 実験結果と考察

登録の合計所要時間を図 7，取得の合計所要時間を図 8 に示す．

まず登録件数と処理時間については，最も時間がかかる各情報ごとに暗号化した場合でも 0.2 秒で 1 件, 2 秒で 60 件登録することができており，今回実験に用いた心拍数データのように，毎秒生成されるデータを生成された直後に 1 件ずつ登録することが可能だとわかった．

表 3: 各情報ごとに暗号化した場合の暗号化の際の各フェーズの所要時間の割合 [%]

件数	1	10	30	60
暗号化	0.34	0.78	0.97	1.46
再暗号化	0.98	0.33	0.73	0.88
サーバの DB への登録処理	78.2	93.0	95.6	95.1

表 4: リソースごとに暗号化した場合の暗号化の際の各フェーズの所要時間の割合 [%]

件数	1	10	30	60
暗号化	0.88	1.04	1.71	2.61
再暗号化	0.64	0.83	0.96	1.60
サーバの DB への登録処理	57.8	93.1	93.5	92.4

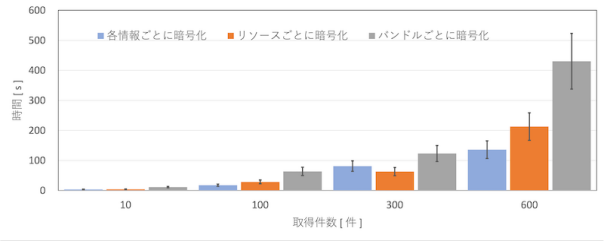

図 8: 一度に取得する件数と取得処理合計時間の関係

表 5: 各情報ごとに暗号化した場合の復号の際の各フェーズの所要時間の割合 [%]

件数	1	10	30	60
サーバの DB からの取得処理	2.78	0.43	0.082	0.0049
再暗号化	0.14	0.23	0.15	0.19
復号	62.0	94.8	98.4	98.1

表 6: リソースごとに暗号化した場合の復号の際の各フェーズの所要時間の割合 [%]

件数	1	10	30	60
サーバの DB からの取得処理	1.22	0.18	0.084	0.029
再暗号化	0.13	0.11	0.23	0.15
復号	96.7	95.2	97.2	98.6

また表 3，4 から登録についてはサーバでの登録処理が大部分を占めるため, 暗号化におけるオーバーヘッドの問題は無視できると言える．なお所要時間を計測した各フェーズ以外に鍵の取得や，アプリケーションとプロキシサーバ，プロキシサーバとサーバ間の通信などの時間があるため表 3，4 の値を合計しても 100 %になっていない．

暗号化単位について比較すると，Bundle ごと，つまり複数件まとめて登録するのが最短だということがわかった．サーバでの登録処理時間が Bundle ごとに暗号化する場合が最も短いことが影響しており，これは複数件のデータをまとめて 1 件のデータとして登録するためだと考えられる．

次に取得については図 8 の結果から，同時に取得処理をするユーザが 1 人の場合，今回実験に用いたデータ 600 件を各情報

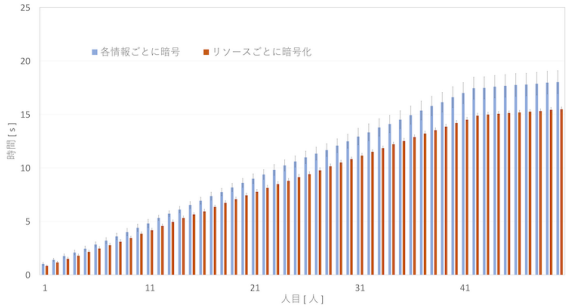

図 9: 50 人が一度に 10 件ずつ追加した時の各人の全体処理時間

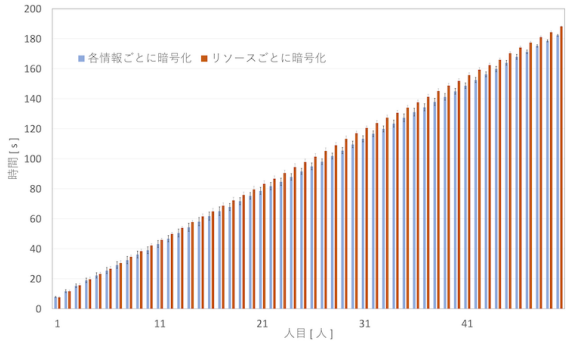

図 10: 50 人が一度に 10 件ずつ取得した時の各人の全体処理時間

ごとに暗号化した場合は約 2 分で，リソースごとでは約 3 分半，Bundle ごとでは約 7 分で取得可能であることがわかった．医療従事者が診察前や最中に患者の生理計測データを確認するといった状況でも利用可能であるといえる．

復号化の際の所要時間については表 5，6 かから，アプリケーションでの復号が取得時間の大部分を占めることがわかった．これは 2.2 節で示したアプリケーションでの復号で極めて大きな素数のモジュラ逆数を算出する際の計算量の多さによるものと考えられる．

暗号化単位について比較すると，Bundle ごとに暗号化した場合の取得処理時間が最大で，これは復号量が大きいためである．

5.2 同時アクセス数と時間の関係

5.2.1 実 験 方 法

同時に複数人が本システムにアクセスすることを想定し，1 回の操作で 10 件のデータ登録・取得する要求を 50 個同時に出したときの各フェーズでの所要時間を，各情報ごととリソースごとの 2 つの暗号化単位で計測した．本実験でも 5.1 節で用いた心拍数のデータを用いた．なお，Bundle ごとに暗号化するパターンは複数件のデータを 1 件のリソースとしてまとめて登録するため，各情報ごと，リソースごとに暗号化する場合と条件が異なるために今回は実験をしていない．

5.2.2 実験結果と考察

50 人が一度に 10 件ずつ追加したときの合計の所用時間を図 9，50 人が一度に 10 件のデータ取得をしたときの合計の所用時間を図 10 に示す．

まずデータ登録については，図 9 から最後の人にかけて実行時間が累積することがわかった．これはサーバでのリソース登録処理時間が最後の人にかけて増加していくことが原因であった．このことから，サーバでの登録処理において登録処理が終了する前に次の登録要求が到着がしてしまい待ち時間が発生していることが考えられる．多くの人が同時にアクセスすることは十分考えられる状況なので，DB への同時書き込み数を増やすなどの対策が必要である．

次にデータ取得については，図 10 から最後の人にかけて実行時間が累積することがわかった．これはサーバでのリソース取得処理時間が最後の人にかけて増加していくことも原因の 1 つであるが，最も大きな原因はアプリケーションでの秘密鍵取得・復号が並列に実行できない実装になっていることであった．ただ，本実験では 1 つのデバイスで複数の復号要求を処理しているが，実際は各人のデバイスのアプリケーションで復号処理を行うはずなので実行時間は累積しないと考えられる．

5.3 MEWS スコア計算のための取得時間

MEWS (Modified early warning score) スコアとは修正早期警戒スコアと呼ばれ，血圧，酸素飽和度，呼吸数，意識レベルなどの指標を使用して，患者の重症度を表すスコアを計算し患者急変を予知するために使用される [3]．本実験では，各情報ごと，リソースごと，Bundle ごとの 3 つの暗号化単位で登録されたデータそれぞれについて，MEWS スコア計算に必要なデータ取得にかかる時間を，取得する患者数を変化させて測定し，患者数による変化と各暗号化単位による違いを明らかにする．

5.3.1 実 験 方 法

MEWS スコアの計算方法はいくつかあるが，本実験では Venki ら [3] が採用していた方法で，収縮期血圧，心拍数，酸素飽和度，体温，呼吸数を用いてスコアを計算した．また，この 5 種類のデータに加えて MEWS スコアの計算に使用しない拡張機血圧と脈拍のデータも含めた 7 種類のデータを用いた．これらのデータは自分でそれぞれの正常値を参考に作成した．シナリオとして，6 時，9 時，12 時の 3 時間おきに，上記 7 種類の項目を 50 人の患者が測定し，12 時の計測データを使って MEWS スコアを計算するために必要データを取得してくることを想定した．

各情報ごとに暗号化するパターンでは，計測値だけを暗号化したデータを登録したので，計測項目コードと計測日時を取得時の検索パラメータとして使用でき，12 時に測った収縮期血圧，心拍数，酸素飽和度，体温，呼吸数の計測値のみを復号する．リソースごとに暗号化するパターンでは，計測値と計測項目を暗号化してデータを登録したので計測日時のみを検索パラメータとして使用でき，12 時に測ったデータを全て復号した後に必要な 5 種類のデータかどうかを判定した．Bundle ごとに暗号化するパターンは，計測値，計測項目，日時を全て暗号化し，1 回で計測する 7 種類の項目を 1 つの Basic リソースにまとめて登録し，1 人あたり 6 時，9 時，12 時分の 3 つのリソースを登録した．このパターンでは取得時に検索パラメータが指定できないので，全てのデータを取得し復号した後，対象の日時，項目であるかを判断した．各暗号化単位における各計測内容の暗号化データ量と実際に復号するデータ量は表 7 の通りである．

表 7: 各暗号化単位の復号データ量 [byte/人]

暗号化単位	復号量
各情報ごと	15
リソースごと	233
Bundle ごと	1,263

表 8: 各情報ごとに暗号化したときのデータ取得人数と時間の関係 [ms]

取得人数	1	2	5	10	25	50
サーバの DB からの取得処理	3,658	4,641	4,897	7,520	19,378	35,285
再暗号化	1	1	1	3	5	9
復号	454	318	748	1,505	3,404	7,124
合計	4,174	4,989	5,702	9,087	22,826	42,504

表 9: リソースごとに暗号化したときのデータ取得人数と時間の関係 [ms]

取得人数	1	2	5	10	25	50
サーバの DB からの取得処理	1,279	1,567	2,012	2,279	6,557	12,072
再暗号化	3	4	7	18	38	77
復号	2,704	5,025	12,199	23,867	69,316	120,026
合計	4,044	6,725	14,258	26,686	66,929	132,264

表 10: Bundle ごとに暗号化したときのデータ取得人数と時間の関係 [ms]

取得人数	1	2	5	10	25	50
サーバの DB からの取得処理	28	35	52	93	265	449
再暗号化	25	35	52	93	265	449
復号	14,548	28,331	69,721	138,730	346,194	692,248
合計	14,686	28,535	70,121	139,428	347,355	694,081

5.3.2 実験結果と考察

各情報ごとに暗号化したとき，リソースごとに暗号化したとき，Bundle ごとに暗号化したときの MEWS スコア計算のためのデータ取得人数と時間の関係を表 8，9，10 に示す．

表 8 から各情報ごとに暗号化した場合は 50 人の患者データを約 43 秒で取得できており，ネットワークコストを除外すると，COVID-19 自宅療養者 50 人の MEWS スコアを 1 分ごとに計算できることがわかり，これは現実的に妥当な頻度であると考える．

また暗号化の単位については一部のデータのみが必要な場合やリアルタイム性が必要な場合は，暗号化の単位を各情報ごととした上で計測値のみ暗号化するのが適切であるといえる．

5.4 解析のための取得・ソート時間

提案システムを使った生理計測データと電子カルテとを紐付けた解析の例としてシーケンシャルパターンマイニングを検討する．シーケンシャルパターンマイニングを適用するためにはデータが日時順に並んでいることが必要となる．暗号化単位を変化させて生理計測データと医療指示情報が日時順に並んだデータの取得にかかる時間を計測した．

5.4.1 実 験 方 法

まず生理計測データと医療指示情報の 2 種類のデータを作成する．生理計測データについてはオープンソースの COVID-19

表 11: 各日の作成したデータ件数 [件]

日付	生理計測データ	医療指示情報
2020 年 3 月 25 日	969	0
2020 年 3 月 26 日	800	0
2020 年 3 月 27 日	202	0
2020 年 3 月 28 日	800	36
2020 年 3 月 29 日	270	9
2020 年 3 月 30 日	800	6
2020 年 3 月 31 日	0	6
2020 年 4 月 1 日	437	6
2020 年 4 月 2 日	334	31
2020 年 4 月 3 日	383	6
2020 年 4 月 4 日	707	10

患者がウェアラブルデバイスで測定した生理計測データセット [21] の中から，COVID-19 の発症日が記録されていて発症前後の生理計測データがあるユーザを 1 人選び，2020 年 3 月 25 日から 2020 年 4 月 4 日までの心拍数データの一部を使って作成した．各情報ごとに暗号化する方法では計測値のみを暗号化した．Bundle ごとに暗号化するパターンでは同じ日付のデータを約 100 件ずつまとめて 1 つのリソースとし計測日の情報も追加して登録した．医療指示情報については,「COVID-19 感染の診療への影響調査と予測モデル開発」に関する研究の研究協力機関である医療機関 A から提供された，COVID-19 に対する実電子カルテデータ中の医療指示データを参考にデータ作成した．Bundle ごとに暗号化するパターンでは各日の全データをまとめて 1 つの Basic リソースとし，医療指示日の情報を Basic リソースに付与して登録した．作成した生理計測データと医療指示情報の件数は表 11 の通りである．

2020 年 3 月 27 日と 28 日のデータを取得・復号して日時順に並び替えるまでの時間を計測した．なお参考にした電子カルテ情報には時刻情報はなかったので，医療指示情報は同日の全生理計測データの後ろに挿入することとした．各情報ごと・リソースごとに暗号化した場合は，日付を検索パラメータとし，リソース id 順にソートした生理計測データと医療指示情報をそれぞれ取得し復号した．今回生理計測データは日時順に登録をしたのでリソースの id 順にソートされたものを取得することで日時順のデータを取得することができる．最後に取得・復号した生理計測データと医療指示情報を日付け順になるようにソートした．Bundle ごとに暗号化した場合は，日付情報を追加して登録したため日付を検索可能なので日付を検索パラメータとしてデータを取得した．その後復号しデータを日時順にソートした．本実験で実際に復号したデータ量は，各情報ごとに暗号化したパターンでは 2,627 byte，リソースごとに暗号化しがパターンでは 24,743 byte，Bundle ごとに暗号化したパターンでは 49,437 byte だった．

5.4.2 実験結果と考察

各情報ごと，リソースごと，Bundle ごとに暗号化したときの各フェーズの処理時間を表 12 に示す．

まずソート時間は，各情報ごと・リソースごとに暗号化する場合はソート済みの医療指示情報と生理計測データをソート，Bundle ごとに暗号化する場合はソートされていない医療指示情報と生理計測データを一括でソートするのにかかった時間だが，95 %信頼区間も考慮すると両者に大きな差はなかった．各情報ごと・リソースごとに暗号化する場合はサーバでのリソース取得時に各種リソース内のソートを実施しているが，本実験に追加して各情報ごとに暗号化したデータを日時の指定のみ行いリソースの id 順にソートしないで取得する場合のサーバ処理時間を 10 回計測し平均値を出したところ，医療指示情報取得は 125 ms，生理計測データ取得は 1,704 ms であった．このことからサーバの取得処理時のソートに医療情報は 20 ms, 生理計測データは 560 ms かかっていることがわかり，ソート自体はアプリケーションにて一括で実施する方が高速にできると考えられる．

表 12: 暗号化単位と SPM 適応のためのデータ取得時間 [ms] の関係

	各情報ごとに暗号化（ms）	リソースごとに暗号化（ms）	Bundle ごとに暗号化（ms）
全体実行時間	41,983	338,529	680,435
医療指示情報取得全処理時間	9,630	10,486	17,443
医療指示情報取得	145	133	67
医療指示再暗号化	11.5	9.5	35.9
医療指示情報復号	8,748	9,625	17,224
生理計測データ取得全処理	41,953	338,500	680,428
生理計測データ取得処理	2,264	2,072	1,466
生理計測データ再暗号化	57.0	147.7	308.5
生理計測データ復号	32,314	327,986	662,958
ソート時間	7.3	7.6	6.7

また，サーバでのリソース取得時間は Bundle ごとに暗号化されたものを取得する方法が最短で，本実験では医療指示情報を Bundle ごとに暗号化する際に暗号化する内容に日付情報も含んでいたが，日付情報を暗号化対象から外し Basic リソースの日付情報を利用することで復号量が減って処理時間が減少すると考えられるので，時刻情報が不要な医療指示情報は同じ日付をまとめて Bundle とし暗号化するのが適切だといえる．

一方で時刻情報が必要な生理計測データは，日時情報を暗号化対象から外し Basic リソースの日付情報を利用することはできないので Bundle ごとに暗号化すると復号量が大きくなり復号時間が増加するので，各情報ごとに暗号化するのが適切だといえる．

6 おわりに

6.1 まとめ

紐付けのための個人を特定可能な匿名 ID と, 医療従事者以外が解析できるように公開鍵暗号基盤 (PKI) とプロキシ再暗号化を用いて, 秘匿性レベルに応じた共有範囲を管理可能な生理計測データ集積システムを提案した．また生理計測データの集積単位と対応して，FHIR データフォーマットに則り暗号化の単位を各情報ごと，リソースごと，Bundle ごとの 3 つ検討した．

提案システムのプロトタイピングを行い，生理計測データを集積して暗号化や共有範囲を考慮した復号が正しく行われることを確認した．そして実データセットを用いて暗号化・復号を施すデータ量や同時アクセス数，暗号化単位をといった条件を変化させたときの登録・取得時間，さらに MEWS スコア計算や

シーケンシャルパターンマイニング適応に必要なデータ取得に要する時間を計測することで，暗号化や復号にかかるオーバーヘッドを確認して提案システムの実現可能性を確認するとともに，どの暗号化単位を用いるのが妥当かを検討した．

その結果実現可能性については，データの登録・取得共に実用に耐えられる実行時間であること，データ登録時の暗号化のオーバヘッドは無視できるがデータ取得ではアプリケーションでの復号が大部分を占めること，複数人アクセスした場合は最後の人にかけて処理時間が累積していくことがわかった．

生理計測データと電子カルテを紐付けた解析に適した暗号化単位については，内容や時刻情報で選別したい場合やリアルタイム性が求められる場合は各情報ごとに暗号化し，そうでない場合は同じ日付でまとめて Bundle として登録し Bundle ごとに暗号化するのが適しているとわかった．

6.2 今後の課題

実現可能性の評価のために，許可される DB への同時書き込み数を増加させたときに，複数人が同時にデータ登録・取得をしても現実的に妥当な時間内に処理が終了できるかを検証していきたい．加えて今回は実験において想定したシナリオが限定的であったが，実際の病院システムに即したアクセス数やデータサイズで本システムを利用する際のシステム構成を検討し，実験・評価をする必要がある．

また傍受者とプロキシが結託した場合など今回用いた暗号方式で想定される脅威についての耐性についてセキュリティ評価を行う必要があると考えられる．

さらに，本研究で用いた BBS 暗号方式は大小比較ができないために，Bundle ごとに暗号化した際には日時情報を検索パラメータをして用いることができなかったが，大小比較が可能な暗号方式を利用することでこの課題を解決することができる [22]．暗号化されたまま日時の大小比較を行い取得する方法と，現在のように全てデータを取得・復号して比較する場合ではどちらの方がコストが小さいのか検証することも今後の課題である．

文　献

[1] 横田治夫. 電子カルテデータ解析-医療支援のためのエビデンス・ベースド・アプローチ. 共立出版, 2022.

[2] Li Yuqing, Le Hieu Hanh, 松尾亮輔, 山崎友義, 荒木賢二, 横田治夫. シーケンシャルパターンマイニングに基づく多病院間の頻出治療パターンの比較. 第 14 回データ工学と情報マネジメントに関するフォーラム予稿集, 2022.

[3] Venki Balasubramanian, Rehena Sulthana, Andrew Stranieri, G Manoharan, Teena Arora, Ram Srinivasan, K Mahalakshmi, and Varun G Menon. A secured real-time iomt application for monitoring isolated covid-19 patients using edge computing. In *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 1227–1234, 2021.

[4] Gatha Varma, Ritu Chauhan, Madhusudan Singh, and Dhananjay Singh. Pre-emption of affliction severity using hrv measurements from a smart wearable; case-study on sars-cov-2 symptoms. *Sensors*, Vol. 20, No. 24, 2020.

[5] Muzhe Guo, Long Nguyen, Hongfei Du, and Fang Jin. When patients recover from covid-19: Data-driven insights from wearable technologies. *Frontiers in Big Data*, Vol. 5, , 2022.

[6] Haytham Hijazi, Manar Abu Talib, Ahmad Hasasneh, Ali Bou Nassif, Nafisa Ahmed, and Qassim Nasir. Wearable devices, smartphones, and interpretable artificial intelligence in combating covid-19. *Sensors*, Vol. 21, No. 24, 2021.

[7] Alexey Ponomarev, Konstantin Tyapochkin, Ekaterina Surkova, Evgeniya Smorodnikova, and Pavel Pravdin. Heart rate variability as a prospective predictor of early covid-19 symptoms. *medRxiv*, 2021.

[8] 日本 HL7 協会 HL7Japan. Hl7 とは. http://www.hl7.jp/whatis/. (Accessed on 01/04/2023).

[9] Hl7 fhir に関する調査研究. https://www.mhlw.go.jp/stf/newpage_15747.html. (Accessed on 01/04/2023).

[10] Resourcelist - fhir v4.3.0. https://www.hl7.org/fhir/resourcelist.html. (Accessed on 01/04/2023).

[11] 児玉快, 横田治夫. データやユーザの効率的な追加・削除が可能な秘匿情報アクセス手法. 第 7 回データ工学と情報マネジメントに関するフォーラム論文集, 2015.

[12] 萱原正彬, 本田祐一, 山田達大, Le Hieu Hanh, 串間宗夫, 小川泰右, 松尾亮輔, 山崎友義, 荒木賢二, 横田治夫. 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. 第 11 回データ工学と情報マネジメントに関するフォーラム論文集, 2019.

[13] Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In Kaisa Nyberg, editor, *Advances in Cryptology — EUROCRYPT'98*, pp. 127–144, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[14] Abel N Kho, John P Cashy, Kathryn L Jackson, Adam R Pah, Satyender Goel, Jörn Boehnke, John Eric Humphries, Scott Duke Kominers, Bala N Hota, Shannon A Sims, et al. Design and implementation of a privacy preserving electronic health record linkage tool in chicago. *Journal of the American Medical Informatics Association*, Vol. 22, No. 5, pp. 1072–1080, 2015.

[15] ePHDS 委員会, 日本 PACS 研究会日本 IHE 協会. 地域医療連携情報システム構築ハンドブック 2011. https://www.ihe-j.org/file2/material/IHE-XDS-Handbook-2011-Main.pdf. (Accessed on 01/04/2023).

[16] Hapi fhir - the open source fhir api for java. https://hapifhir.io/.

[17] hapifhir/hapi-fhir-jpaserver-starter. https://github.com/hapifhir/hapi-fhir-jpaserver-starter.

[18] Synthea. https://synthetichealth.github.io/synthea/#about-landing. (Accessed on 01/04/2023).

[19] Xiao Li, Jessilyn Dunn, Denis Salins, Gao Zhou, Wenyu Zhou, Sophia Miryam Schüssler-Fiorenza Rose, Dalia Perelman, Elizabeth Colbert, Ryan Runge, Shannon Rego, et al. Digital health: tracking physiomes and activity using wearable biosensors reveals useful health-related information. *PLoS biology*, Vol. 15, No. 1, p. e2001402, 2017.

[20] Brinnae Bent, Ke Wang, Emilia Grzesiak, Chentian Jiang, Yuankai Qi, Yihang Jiang, Peter Cho, Kyle Zingler, Felix Ikponmwosa Ogbeide, Arthur Zhao, et al. The digital biomarker discovery pipeline: An open-source software platform for the development of digital biomarkers using mhealth and wearables data. *Journal of clinical and translational science*, Vol. 5, No. 1, 2021.

[21] Welltory/hrv-covid19: Covid-19 and wearables open data research. https://github.com/Welltory/hrv-covid19. (Accessed on 01/17/2023).

[22] 淺野将希, Le Hieu Hanh, 横田治夫. 暗号化された情報の大小比較可能な検索処理の trie 型分岐構造管理による高速化手法. 第 12 回データ工学と情報マネジメントに関するフォーラム論文集, 2020.