

IoT センサ情報を管理する ブロックチェーンシステムの構築

小八重潤¹ 下園幸一² 升屋正人²

概要： IoT センサ情報を従来の中央集権型の IT システムで管理する場合、サイバー攻撃の被害を受け情報を窃取されるリスクがあることや、データが改ざんされていないことの保証が難しいという問題がある。一方、分散型台帳技術とも呼ばれるブロックチェーンは、分散型のネットワークに取引の内容と順序を記録するものであり、データの対改竄性が高く、特定の機器が攻撃を受けても被害は小さい。そこで本研究ではこの特性を生かし、IoT センサ情報を管理するブロックチェーンシステムを構築した。

キーワード： 改ざん防止, 分散システム, Arduino, Ethereum, スマートコントラクト

Blockchain Based System for Gathering IoT Sensor Data

JUN KOBAE¹ KOICHI SHIMOZONO² MASATO MASUYA²

Abstract: When managing IoT sensor information with a conventional centralized IT system, there are problems that there is a risk of information being stolen due to the damage of cyberattacks and it is difficult to guarantee that the data has not been tampered with. On the other hand, blockchain, which is also called distributed ledger technology, records the contents and order of transactions in a distributed network, and the data is highly tamper-proof, and even if a specific device in the system is attacked, it can keep the damage at a minimum. Therefore, in this research, we constructed a blockchain system that manages IoT sensor information by taking advantage of this characteristic.

Keywords: Tamper Protection, Distributed System, Arduino, Ethereum, Smart Contract

1. はじめに

IoT が普及し、高度なサービスが登場するにつれて、従来のクライアント・サーバ型の中央集権型 IT システムの技術的弱点が課題になると言われている[1]。中央集権型 IT システムでは、クライアントとなる大量の IoT デバイスから膨大なデータが送信されてトラフィックが増加すると、通信回線やサーバに多くの負荷がかかる。その結果、多くの IoT システムにとって重要なリアルタイム性が損なわれ、場合によってはサーバに障害が発生して通信の継続性が失われる。また、サーバが被害に遭うとシステム全体に影響があるため、サイバー攻撃に対しては、中央にきわめて堅牢なセキュリティ対策を講じる必要がある。IoT のサービスの高度化を進める上で、これらの課題の解決は不可欠である。分散型

システムであるブロックチェーンはその解決につながる技術の一つと考えられている[2]。

ブロックチェーンは、一般社団法人日本ブロックチェーン協会によると「電子署名とハッシュポインタを使用し改竄検出が容易なデータ構造を持ち、且つ、当該データをネットワーク上に分散する多数のノードに保持させることで、高可用性及びデータ同一性等を実現する技術」とされている[3]。信頼できない多数のノードからなる分散システムにおいて合意形成を図ることで、結果の改竄や紛失を防ぐ技術でもある。ビットコインを初めとする暗号通貨の事例が広く知られているが、ブロックチェーン自体は、さまざまな分野に応用ができる技術であり、IoT における情報共有に関する研究の例[4]もある。しかし、ブロックチェーンが IoT に応用できることは示されているものの、その概念に関する

¹ 鹿児島大学大学院理工学研究科
Graduate School of Science and Engineering, Kagoshima University
² 鹿児島大学学術情報基盤センター
Computing and Communications Center, Kagoshima University

研究が大半で、実際にブロックチェーン技術を用いて IoT システムを構築し、検証した例は知られていない。

そこで本研究では、IoT センサの情報を管理するブロックチェーンシステムを構築し、IoT におけるブロックチェーン技術の有効性を評価することにした。

2. システムの構成

本研究で構築するブロックチェーンシステムは、センサの識別情報、温度、時刻を Web サーバに送信する IoT センサ、データを受け取り表示する Web サーバ、ブロックチェーンに接続して情報の記録・読み出しを行うスマートコントラクト、ブロックチェーンプラットフォーム、そして、ブロックチェーンに記録した情報を検索して表示する Web アプリケーションから構成される (図 1)。IoT センサを接続した Web サーバに、スマートコントラクトを実装する形となり、これらを併せたものが 1 つのノードとなる。Web サーバに複数の IoT センサを接続することができるため、1 つのノードの内部で複数のセンサの情報を集約した上で、ブロックチェーンネットワークに接続することができる。クライアント・サーバシステムとブロックチェーンを階層的に組み合わせたシステムと言える。分散台帳を構成するノード単位に局所的にクライアント・サーバシステムを用いることで、従来の中央集権型システムの弱点を回避しつつ、多数の IoT センサの効率的な利用を実現した。なお、情報を検索して表示する Web アプリケーションにもブロックチェーンに接続するためのスマートコントラクトを実装した。

ブロックチェーンのプラットフォームとしては Ethereum[5]を使用した。また、Ethereum の仮想通貨の出し入れに使う仕組み (ウォレット) としては、取引における手数料を自由に設定することができる MetaMask を使用した。

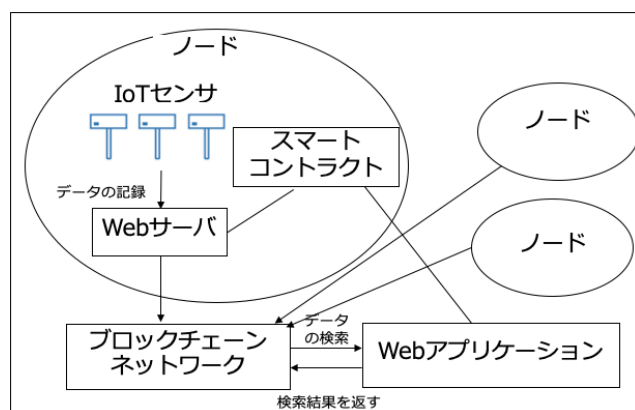


図 1 システム構成

Figure 1 System Configuration

2.1 IoT センサ

本研究のシステムの IoT センサとして、温度センサと無線 LAN 機能を内蔵し Arduino 環境で開発を行うことができ

る M5Stack BASIC を使用した。M5Stack 本体にはセンサで取得した温度を確認のためアナログ・デジタル両形式で表示することとし、表示部は[6]のソースコードを参考にプログラムを作成した (図 2)。また、取得した温度と IoT デバイスの識別番号を Web サーバに無線 LAN で送信する部分は M5Stack-Core-ESP32 用のスケッチ例の HelloServer.ino を参考にプログラムを作成した。

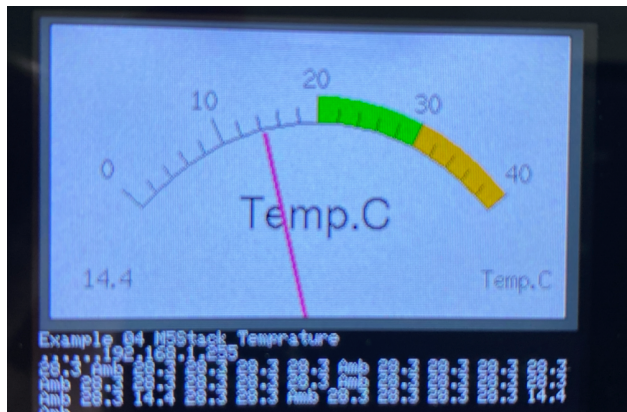


図 2 温度を取得し情報を表示するとともに送信する IoT センサとして用いた M5Stack BASIC の画面

Figure 2 The Screen of M5Stack Basic used as an IoT sensor that acquires temperature, displays information, and transmits it.

M5Stack Basic測定データ

```
No. 年/月/日 時:分:秒 °C
001.2022/01/10, 16:36:32, 8.9
001.2022/01/10, 16:47:00, 9.1
001.2022/01/10, 16:57:00, 8.9
001.2022/01/10, 17:07:00, 8.7
001.2022/01/10, 17:17:00, 8.1
001.2022/01/10, 17:27:00, 7.5
001.2022/01/10, 17:37:00, 7.6
001.2022/01/10, 17:47:00, 7.1
001.2022/01/10, 17:57:00, 7.6
001.2022/01/10, 18:07:00, 6.9
001.2022/01/10, 18:17:00, 6.1
001.2022/01/10, 18:27:00, 5.9
001.2022/01/10, 18:37:00, 5.6
001.2022/01/10, 18:47:00, 4.9
001.2022/01/10, 18:57:13, 4.9
```

図 3 取得したセンサ情報を Web サーバの Web ページで表示している例

Figure 3 An example of displaying the acquired sensor information on the Web page of the Web server.

2.2 データを受け取り表示する Web サーバ

IoT センサで取得したデータを、PHP を用いて構築した Web サーバで受信し、表示する (図 3)。Web サーバ上でスマートコントラクトを動作させ、IoT センサからのデータを取得すると同時にデータをブロックチェーンに記録する。Ethereum を利用するにあたり、Web サーバ 1 台につき、デ

ータを記録するときに手数料を支払う MetaMask アカウントを1つ割り当てることになる。

2.3 ブロックチェーンに接続するスマートコントラクト

スマートコントラクトの開発環境として REMIX IDE(v0.21.2)を使用した。IoT センサからのデータの取得をトリガーにして Ethereum でトランザクション（仮想通貨の取引：本システムではデータの記録と呼び出し）を実行するスマートコントラクト（自動実行プログラム）を[7]を参考に Solidity で作成した。スマートコントラクトには IoT の識別番号を MetaMask アカウントと紐付けて新規登録する関数、識別番号が登録されているかどうかを確認する関数、データを読み込む関数を実装し、作成したスマートコントラクトを Ethereum のテスト用ネットワークである Ropsten Test Network に接続した。スマートコントラクトのアドレスを取得し、Web サーバと Web アプリケーションから呼び出せるようにすることで、データを取得すると同時に Ethereum のブロックチェーンネットワークに接続してデータを記録できるようにした。

トランザクションの際には手数料がマイナー（トランザクションの検証と記録を自らのコンピュータで行う参加者）へ支払われる。Ethereum における手数料（Gas Fee）は、Gas を単位として内部通貨とは別に設定されており、1 Gas あたりの単価（Gas Prime）とトランザクションの際にかかる最大の仕事量（Gas Limit）の積で決定される。これらはトランザクションの実行者が設定し、Gas Prime を高く設定するほど、処理が優先的に行われるためトランザクションにかかる時間が短くなる。Gas Fee と Ethereum の内部通貨である ETH の交換比率は日々変動しているため、そのたびに支払う ETH も変動する。表 1 に設定した Gas Prime と Gas Limit、2022 年 2 月 7 日時点での 1 回のトランザクションに支払う手数料（ETH）、ブロックの記録時間を示す。

表 1 トランザクションにかかる手数料と時間

Table 1 Fee and time required for a transaction.

Gas Prime	2100000000
Gas Limit	3000000
手数料（ETH）	0.0063
トランザクションにかかる時間（s）	15

2.4 ブロックチェーンに記録した情報を検索して表示する Web アプリケーション

ブロックチェーンに記録した IoT センサ情報を管理するため、接続ブロックチェーンに記録した日時と識別番号、もしくは日時のみ、識別番号のみを検索するアプリケーションを作成した（図 4）。スマートコントラクトを実装しているがブロックチェーンにデータを書き込むわけではないため、検索結果を表示する際にトランザクションは発生しない。

い。

IoT センサ情報

識別番号

年

月

日

時

001.2022/01/10, 17:07:00, 8.7

001.2022/01/10, 17:17:00, 8.1

001.2022/01/10, 17:27:00, 7.5

001.2022/01/10, 17:37:00, 7.6

001.2022/01/10, 17:47:00, 7.1

001.2022/01/10, 17:57:00, 7.6

図 4 ブロックチェーンに記録したデータの検索結果の例
Figure 4 An example of search results of data recorded on the blockchain

3. システム負荷の評価

本システムを構成する機器のうち、データを収集しブロックチェーンネットワークに記録する Web サーバとスマートコントラクトのプログラムは同一の機器上に実装される。多数の IoT センサが接続した場合に負荷が問題となる機器であるため、この機器のシステム負荷を評価する。本研究のシステムでは Apple MacBook Air (CPU: Intel Core i5-4260U, メモリ: 8 GB, OS: macOS Catalina 10.15.5) を用いた。本システムの IoT センサと Web サーバの間の通信、及び、ブロックチェーンに記録する通信の際のメモリと CPU の使用率を、それぞれ 30 秒ごとに 5 分間計測した。通信していないときの使用率を図 5 に示す。

3.1 センサと Web サーバの間の通信時の使用率

Web サーバに情報を送信しているときのメモリと CPU の使用率を図 6 に示す。図 5 と比較してメモリ、CPU ともに変化がみられない。センサが 1 つの場合には、Web サーバとの間の通信においてメモリ、CPU ともに負荷はほとんどない。多数のセンサを接続した時の負荷はまだ調べていないのでわからないが、数個程度までであれば負荷が問題になることはないことがわかった。

3.2 ブロックチェーンに記録する通信時の負荷

ブロックチェーンにデータを記録するときのメモリと CPU の使用率を図 7 に示す。メモリ、CPU ともに通信が生じたときに使用率の変化が見られ、特に CPU はピーク時でおおよそ 20% の処理能力が必要とすることがわかった。

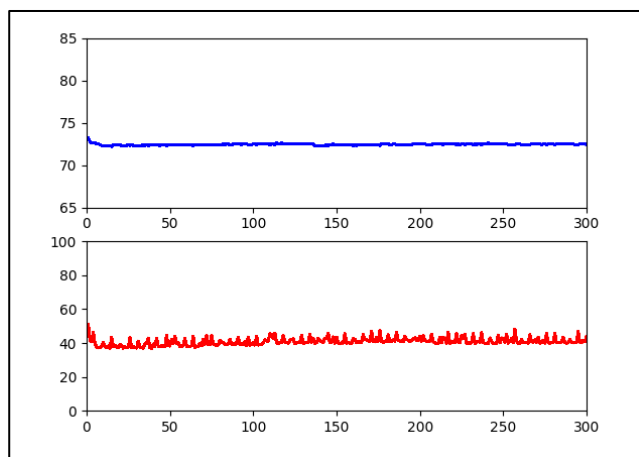


図 5 通信していないときの使用率
(上：メモリ，下：CPU)
Figure 5 Utilization rate not communicating
(Upper : memory, Lower : CPU)

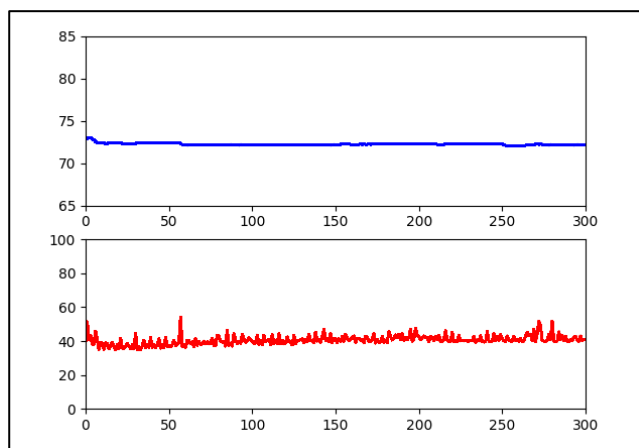


図 6 30 秒ごとにセンサから Web サーバに通信している
ときの使用率 (上：メモリ，下：CPU)
Figure 6 Utilization rate when communicating from the sensor
to the Web Server every 30 seconds
(Upper : memory, Lower : CPU)

4. まとめ

IoT センサの情報を管理するブロックチェーンシステムを構築し、構築したシステムでセンサと Web サーバの間の通信、及び、ブロックチェーンにデータを記録する通信のときのメモリと CPU の負荷を計測した。その結果、センサとの通信に際しては負荷がほとんどないこと、データをブロックチェーンに登録するときの負荷は高く、CPU 使用率はおよそ 20%であることがわかった。

多数の IoT センサが存在する場合は、Web サーバとの通信時の負荷が上昇することが予想されるが、台数に比例して上昇するのか、あるいは、複数センサが存在しても負荷の上昇は大きくないのかについて不明であるので、センサの

数を増やして今後評価を進める予定である。また、ブロックチェーン記録時の負荷については、センサの数が増えても大きくは増加しないことが予想されるが、こちらについてもセンサの数を増やして検証の予定である。

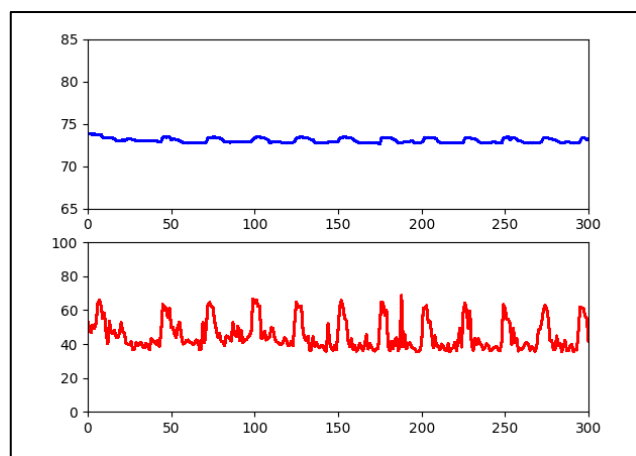


図 7 30 秒ごとにブロックチェーンにデータを記録するときの使用率 (上：メモリ，下：CPU)
Figure 7 Utilization rate when recording information on the
blockchain every 30 seconds (Upper : memory, Lower : CPU)

参考文献

- [1] 北野健太：IoT におけるブロックチェーンの適応可能性について－「つながる IoT」プラットフォームの実現に向けて－, JRI レビュー, Vol. 8, No.47, pp.27-46 (2017).
- [2] Nallapaneni, M.K. and Pradeep, K.M.: Blockchain technology for security issues and challenges in IoT, Procedia Computer Science, Vol.132, pp.1815-1823 (2018).
- [3] Japan Blockchain Association : 「ブロックチェーンの定義」を公開しました, 入手先<<https://jba-web.jp/news/642>> (参照 2022-2-5).
- [4] Haiping, S., Changxia, S., Yanling, L., Hongbo, Q. and Lei, S.: IoT information sharing security mechanism based on blockchain technology, Future Generation Computer Systems, Vol.101, pp.1028-1040 (2019).
- [5] Ethereum Foundation : イーサリアムとは?, ethereum.org, 入手先<<https://ethereum.org/ja/what-is-ethereum>> (参照 2022-2-5).
- [6] 国野亘：全部入りマイコン M5Stack で始める IoT, エレキジャック IoT, No.2, pp.66-88 (2020).
- [7] Kevin, S., Randall, K. and David, H.H.: Hands-On Smart Contract Development with Solidity and Ethereum, O'Reilly Media, Inc (2019). 中城元臣 (訳) : Solidity と Ethereum による実践スマートコントラクト開発－Truffle Suite を用いた開発の基礎からデプロイまで, オライリー・ジャパン (2021)