IoTデバイスにおけるWi-Fi接続のための パスワード更新の自動化

牧 丈晴[1]　大野 有樹[2]　串田 高幸[1]

概要：IoT デバイスの通信方式に Wi-Fi（無線 LAN）がある．IoT デバイスが Wi-Fi 接続することで測定データをサーバに送信することできるようになり，遠隔で測定データの閲覧が可能になる．また Wi-Fi は Wi-Fi ルータ 1 つで多くの IoT デバイスと接続することが可能である．IoT デバイスの管理者は Wi-Fi のパスワードを新しく設定した際に，Wi-Fi に接続されている IoT デバイスのパスワードを 1 台ずつ更新するため，時間がかかる．本稿では IoT デバイスにおける Wi-Fi のパスワードを更新する作業を自動化する手法を提案した．自動化を実現するために，予約時刻と IoT デバイスのリストを作成した．予約時刻は IoT デバイスと Wi-Fi ルータに設定することで，Wi-Fi のパスワード更新を同時に行える．IoT デバイスのリストはサーバに 24 時間以内にデータを送信した IoT デバイスの IP アドレスと MAC アドレスで構成されており，リストを参照してパスワード更新する IoT デバイスを識別する．予約時刻と IoT デバイスのリストを用いることでパスワード更新の自動化が行える．予約時刻になった場合，新しいパスワードを用いて IoT デバイスは Wi-Fi の再接続を行う．IoT デバイスが再接続を開始してから接続完了までの時間を 10 回計測した．結果 IoT デバイスの台数が 1 台のときの平均が 11.3 秒で，2 台のときの平均が 13.8 秒になった．

1. はじめに

背景

IoT（Internet of Things）を活用して農業を行うことをスマート農業という [1, 2]．スマート農業の活用例としてビニールハウスを用いたレタスの温室栽培が挙げられる．温室栽培ではレタスを発芽，生育するために室温を 20 ℃から 30 ℃の間に保つ必要がある [3]．20 ℃以下もしくは 30 ℃以上の室温で 1 日以上栽培していると，生育に悪影響を与えてしまう [4]．そのためビニールハウスの管理者は室温を測定して管理する必要がある．しかしビニールハウス内に赴いて何度も室温を測定するのは手間である．そこでビニールハウスの室温の遠隔監視が用いられる [5, 6]．ビニールハウス内に温度センサを搭載した IoT デバイスを設置する．IoT デバイスが測定した温度データをインターネットを介してサーバに送信することで，管理者が遠隔でビニールハウス内の室温を監視できる．

IoT デバイスがインターネットに接続する方法として Wi-Fi（無線 LAN）がある [7, 8]．Wi-Fi は Wi-Fi ルータと呼ばれるアクセスポイントに接続することで通信が可能になる．Wi-Fi ルータに接続するためには SSID とパスワードを入力する必要がある．SSID とは Wi-Fi ルータが出力している Wi-Fi の識別子である．Wi-Fi ルータは SSID と SSID に対応しているパスワードを設定することで，任意のデバイスのみと接続することが可能になる．また IoT デバイスも SSID とパスワードを設定することで，接続したい Wi-Fi ルータと接続することが可能になる．

Wi-Fi のパスワードを設定する理由は，情報漏洩を防ぐためである．Wi-Fi の暗号化がされていない場合，通信内容を盗聴することができる [9–11]．またパスワードを設定していない場合，だれでもネットワーク内に侵入できる [12]．ネットワーク内に侵入した際にはネットワーク内に接続している機器に保存している情報を取得することができる．例えば，業務で使用している PC が Wi-Fi を用いてインターネットと接続している場合，業務に関する顧客の情報が漏洩する．また Wi-Fi のパスワードを設定していない場合，他者に無断使用される問題がある．他者が無断使用していることにより，Wi-Fi が通信できるデータ量に限りがあるため通信速度の低下につながる場合もある[*1]．パスワードが漏洩した場合も同様に上記 2 つの問題が発生

[1] 東京工科大学コンピュータサイエンス学部
〒 192–0982 東京都八王子市片倉町 1404-1
[2] 東京工科大学大学院 バイオ・情報メディア研究科 コンピュータサイエンス専攻
〒 192–0982 東京都八王子市片倉町 1404–1

[*1] https://eset-info.canon-its.jp/malware_info/special/detail/220901.html

© 2022 Cloud and Distributed Systems Laboratory

するため，パスワードの更新が必要になる．

課題

IoT デバイスが接続している Wi-Fi のパスワードを 1 台ずつで更新する場合，IoT デバイスの台数が多いほど更新に要する時間は多くなる課題がある．Wi-Fi ルータのパスワードを更新する際に，Wi-Fi ルータと接続している IoT デバイスも Wi-Fi のパスワードを更新する必要がある．既存の方法では IoT デバイスの Wi-Fi 接続プログラムを実行してから，Wi-Fi ルータに接続するのに 11 秒の時間を要する [13]．この方法では Wi-Fi と接続している IoT デバイスが RSSI を用いて，Wi-Fi に接続したい IoT デバイスを特定する．特定した Wi-Fi に接続したい IoT デバイスに BLE を用いてパスワードを送信する．パスワードを受け取った Wi-Fi に接続したい IoT デバイスは，このパスワードを用いて Wi-Fi に接続する．しかし上記の方法では 1 台ずつのみでしかパスワードの更新が行えない．例としてパスワード更新を行う必要がある IoT デバイスが 9 台の場合，9 台の IoT デバイスを 1 台ずつ Wi-Fi の更新を行う必要がある．そのため IoT デバイスの台数が増えるほど，更新に必要な時間は増えていく．

各章の概要

第 2 章では，本論文の関連研究を説明する．第 3 章では，本稿の課題を解決するための提案手法を説明する．第 4 章では，章の提案手法を実現するための実装を説明する．第 5 章では，第 3 章の提案手法の実験環境と実験の結果の分析を説明する．第 6 章では，提案，実験，評価が本稿の課題を解決しているかを議論する．第 7 章では，本稿の課題解決による貢献を説明する．

2. 関連研究

Junyoung Choi らは IoT デバイスを Wi-Fi ルータに安全かつ短時間で接続するために，PUP と呼ばれる安全な接続方式を提案した [13]．PUP は周囲の電波を利用して Wi-Fi 接続したい IoT デバイスを特定し，RSA 暗号化アルゴリズムを使用してパスワードを BLE 通信を用いて送信する．この研究では結果として，1 台の IoT デバイスを Wi-Fi 接続するために要する時間を 11 秒以内にすることが出来た．しかしこの研究の提案方式では，既に Wi-Fi 接続されている IoT デバイスのパスワードの更新を考慮していない．そのためパスワードの更新を行うには非効率であり，パスワードの更新に多くの時間を要する課題がある．

Mahabub Hasan Mahalat らは Wi-Fi 接続時の IoT デバイスを MAC アドレスのなりすましや認証解除、不正アクセスポイントの脅威から保護するために，PUF ベースのプロトコルを提案した [14]．PUF とは IoT デバイスに搭載されているチップの物理的な構造に生じる微小な違いを用いて，IoT デバイスを識別する技術である．この PUF を用いてワンタイム登録フェーズ、接続初期化フェーズ、ルータ認証フェーズ、およびクライアント認証フェーズの 4 つのステージで認証を行うプロトコルを開発した．これにより Wi-Fi 接続時の MAC スプーフィング攻撃や認証解除、不正アクセスポイントの脅威から保護することが出来た．しかしこの研究の提案方式では，既に Wi-Fi 接続されている IoT デバイスのパスワードの更新を考慮していない．そのため既に Wi-Fi 接続されている IoT デバイスに対してこの提案方式を行うと，パスワードの更新に必要な工程が多くなることで更新に多くの時間を要する課題がある．

Chandramohan Sudarrah らは Wi-Fi ネットワークの認証と保護に使用されている SSID とパスワードが，フィッシングや総当たり攻撃に脆弱性がある課題に対して，TOTP（時間ベースのワンタイムパスワード）を生成する自動化された Wi-Fi 接続の認証システムを提案した [15]．TOTP は共有キーと現在の時刻を使用してワンタイムパスワードを生成するアルゴリズムである．TOTP のアルゴリズムで Wi-Fi のパスワードを作成することで，Wi-Fi のパスワードをワンタイムパスワードにする．アクセスポイントは生成されたワンタイムパスワードと SSID でエンコードされた QR コードを生成する．Wi-Fi 接続したいデバイスが生成された QR コードをスキャンすることで，アクセスポイントと接続することができる．しかしこの研究の提案方式では Wi-Fi 接続を 1 台ずつ行うため複数台の IoT デバイスを Wi-Fi 接続する際に，多くの時間を要する課題がある．

3. 提案

本稿の目的は Wi-Fi ルータに接続されている IoT デバイスに対して，Wi-Fi のパスワード更新に要する時間を削減することである．そのために本稿ではパスワード更新を自動化する提案をした．本稿の提案では下記の 3 つの構成要素で成り立っている．これを下記に記述する．

- IoT デバイスのパスワード更新
- パスワード更新をすべき IoT デバイスのリストを作成
- パスワード更新のために設定した予約時刻の延長

また本稿の提案手法を適応するために IoT デバイスに対して前提条件を下記の 4 つに設定する．

- IoT デバイスは Wi-Fi を用いてインターネットと接続
- IoT デバイス全台で共通して 15 分毎に温度を測定してサーバに送信
- IoT デバイスは温度データと IP アドレス，MAC アドレスをサーバに送信
- サーバは受け取ったデータと受け取った日時をデータベースに保存

提案方式

本稿の提案方式は 3 つの構成要素で成り立っており，こ

© 2022 Cloud and Distributed Systems Laboratory

れらによってパスワード変更に要する時間を削減すること実現可能となる．提案方式の 3 つの構成要素の詳細を述べる．

IoT デバイスのパスワード更新

図 1 は IoT デバイスのパスワード更新の流れを表したものである．最初に管理者がサーバ側に組み込まれているパスワード更新プログラムを実行する．パスワード更新プログラムによってサーバは新しいパスワードと予約時刻を自動で作成する．パスワードはランダムに出力されたアルファベットの大文字，小文字，数字の 8 桁で構成されている．予約時刻はパスワードを作成した日時から 16 分遅らせた日時で作成する．16 分という数字はセンサデータの測定間隔の 15 分に 1 分足したものであり，詳細の説明はパスワード変更のために設定した予約時刻の延長の章で行う．パスワードと予約時刻の作成が完了し，IoT デバイスと Wi-Fi ルータに予約時刻とパスワードを送信する．IoT デバイスと Wi-Fi ルータは受け取った予約時刻とパスワードを自身のストレージに保存する．保存が完了した場合，サーバに向けて保存完了の通知を送信する．現在時刻が予約時刻に到達したら，IoT デバイスと Wi-Fi ルータはストレージに保存してあるパスワードに更新する．

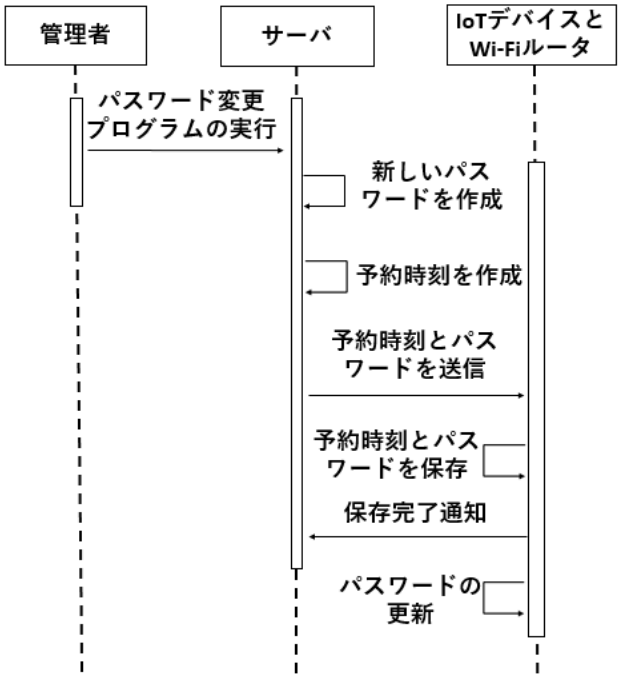

図 1　IoT デバイスのパスワード変更の流れ図

次に予約時刻を設定する理由について説明する．予約時刻を設定することで IoT デバイスと Wi-Fi ルータの Wi-Fi のパスワード更新タイミングを合わせることできる．これによって図 2 のように，IoT デバイス A だけが先に Wi-Fi のパスワードを更新することで，Wi-Fi ルータと接続出来なくなることを防ぐ．

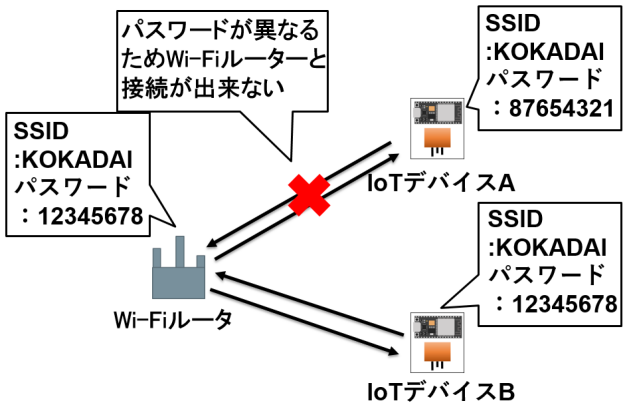

図 2　予約時刻を設定しない場合

パスワード更新をすべき IoT デバイスのリストを作成

Wi-Fi ルータに接続されている IoT デバイス群の中で Wi-Fi のパスワードを更新する IoT デバイスのリストを作成する．リストを作成して参照することで，新しいパスワードを送信すべき IoT デバイスを選定できる．作成するリストは IoT デバイスの IP アドレスと MAC アドレスで構成されている．IP アドレスはサーバが Wi-Fi を通して IoT デバイスにデータを送信する際に用いられる．MAC アドレスは IoT デバイスを識別するために用いられる．IoT デバイスの Wi-Fi 接続では動的 IP アドレスが用いられる [16]．そのため Wi-Fi ルータを再起動した際に，IoT デバイスに割り振られている IP アドレスも変わる可能性がある．これによって IP アドレスでの IoT デバイスの識別は出来ないため，MAC アドレスを用いて IoT デバイスの識別を行う．

パスワード更新をすべき IoT デバイスはサーバに保存されている温度データの送信した日時で判断する．温度データを最後に送信した日時が 24 時間以内の IoT デバイスのみでリストを作成する．温度データを最後に送信した日時が 24 時間を過ぎている IoT デバイスは，停止している IoT デバイスと判断する．24 時間という数字は本稿のユースケースであるビニールハウスでのレタス栽培を参照した．レタスは 20 ℃以下もしくは 30 ℃以上の室温で 1 日以上栽培していると，生育に悪影響を及ぼす [4]．そのため少なくとも 1 日 1 回はビニールハウス内の室温を測定する必要がある．これにより IoT デバイスは少なくともでも 1 日 1 回温度データを測定して，サーバに送信することが機能要件となる．この機能要件から 24 時間以上経過して新しいパスワードと予約時刻を取得できない IoT デバイスは，停止した IoT デバイスと定義する．

パスワード更新のために設定した予約時刻の延長

IoT デバイスがサーバに温度データを送信しているタイミングで，サーバから新しいパスワードと予約時刻が送られてきた場合，IoT デバイスはデータを受け取る準備がで

© 2022 Cloud and Distributed Systems Laboratory

きていないため新しいパスワードと予約時刻の受信に失敗してしまう．新しいパスワードと予約時刻の取得に失敗した場合，Wi-Fi ルータ側でパスワードが更新されてしまうため IoT デバイスは Wi-Fi 接続が出来なくなる．これにより温度データの送信が出来なくなってしまう．そのためサーバから送られてくる新しいパスワードと予約時刻の受信に失敗した IoT デバイスのために，予約時刻の延長をする必要ある．しかし，上限無く予約時刻を延長するとパスワード更新が永遠に出来ない．そのため予約時刻の上限を設定する必要がある．そこで予約時刻の上限を本稿のユースケースであるビニールハウスでのレタス栽培を参照して，24 時間以上経過して新しいパスワードと予約時刻を取得できない IoT デバイスを，停止した IoT デバイスと定義する．

図 3 は予約時刻の延長の分岐を表した図である．初めにサーバでパスワード更新プログラムが実行されて，パスワードと予約時刻が IoT デバイスに送信される．次にパスワード更新をすべき IoT デバイスのリストを参照して，パスワード更新をすべき IoT デバイス全台から応答が来たかを確認する．全台から応答が来ていた場合 IoT デバイスはパスワード更新を実行する．全台から確認応答が来ていない場合，パスワードと予約時刻を IoT デバイスに送信してから 24 時間経過しているかを確認する．24 時間経過していた場合 IoT デバイスはパスワード更新を実行する．24 時間経過していない場合，現在設定されている予約時刻を 16 分延長して再度 IoT デバイスにパスワードと予約時刻を送信する．16 分という数字はセンサデータの測定間隔の 15 分に 1 分足したものである．16 分に設定することで IoT デバイスが温度データをサーバに送信するタイミングをずらすことが出来る．これによって温度データの送信タイミングが原因で，新しいパスワードと予約時刻の取得に失敗した IoT デバイスは，新しいパスワードと延長した予約時刻を受信することが出来る．

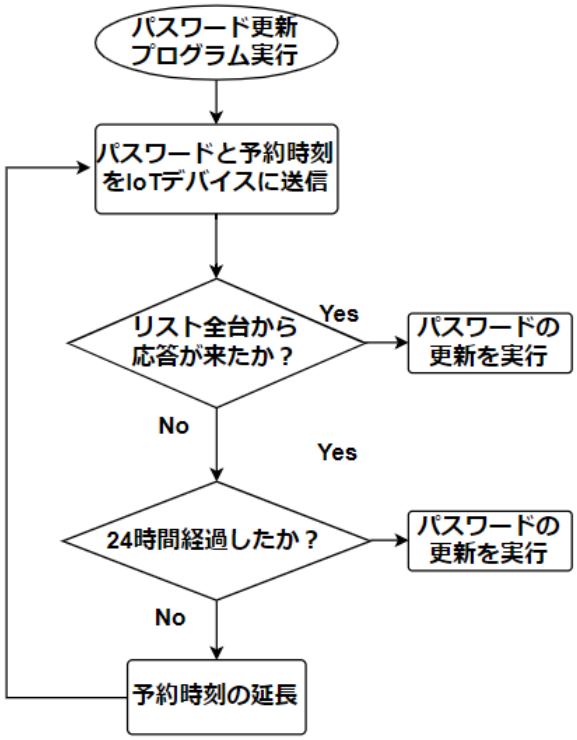

図 3　予約時刻の延長の分岐図

ユースケース・シナリオ

本稿のユースケース・シナリオを図 4 に表す．ユースケース・シナリオはビニールハウスでレタス栽培の温室栽培をしている農家を想定している．農家はビニールハウス内の室温を温度センサを搭載した IoT デバイスを用いることで，遠隔で監視している．ビニールハウス内に設置されている IoT デバイスが，測定した室温を Wi-Fi ルータを通してサーバに送信する．レタス農家はスマートフォンを用いてサーバに保存されている室温を遠隔で閲覧することが可能になる．

レタス農家はビニールハウスを 3 棟用いてレタス栽培を行っている．ビニールハウス 1 棟の大きさは幅 7.4m・奥行 50m である．ビニールハウスにつき IoT デバイスが 3 台設置されている．図 4 のように IoT デバイスは奥行 50m の内，両端と中央に設置されており 25m の間隔がある[*2]．レタス農家が Wi-Fi パスワードをインターネット上に漏洩した際，顧客情報の漏洩と通信速度の低下を防ぐために，パスワードの更新を行う必要がある．パスワードの更新を行うためには設置してある IoT デバイスを全台回収して，1 台ずつパスワードの更新をしなければならない．しかし奥行 50m あるビニールハウス 3 棟から設置されている IoT デバイス全台回収するのは時間的に負担となる．そのため本稿では稼働してある IoT デバイスを回収することなく自動でパスワードの更新を行う．これによりパスワードの更新に要していた時間を削減することが可能になる．

4.　実装

図 5 は本稿で実装したソフトウェア構成図である．図 5 に記されている同名のプログラムは同じ動作を行う．下記にプログラムの説明を記述する．

予約時刻の延長：サーバ

予約時刻の延長を行うプログラムである．Wi-Fi ルータと IoT デバイスから送られて来た保存完了の通知と，パスワード更新すべき IoT デバイスのリストを用いることで，パスワードの更新を行うための予約時刻を延長する．また延長した予約時刻を再度 Wi-Fi ルータと IoT デバイスに送信する．

リスト作成：サーバ

[*2] https://no-chi.com/greenhouse-cost/

© 2022 Cloud and Distributed Systems Laboratory

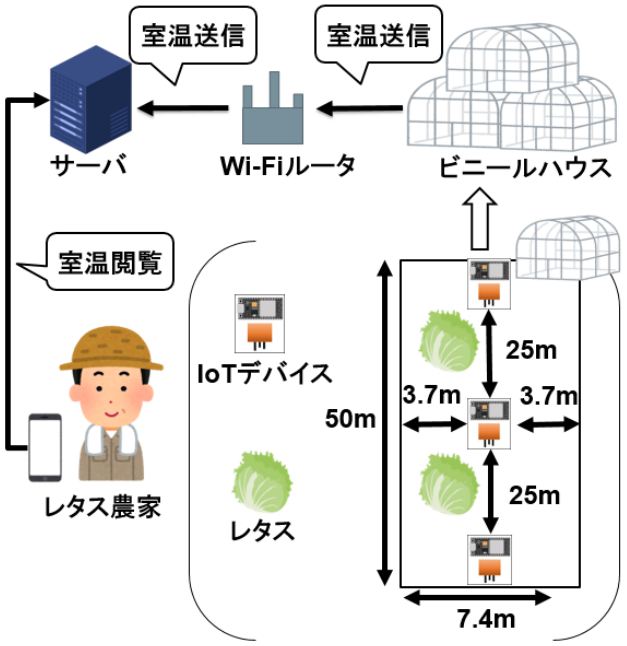

図 4　ユースケース

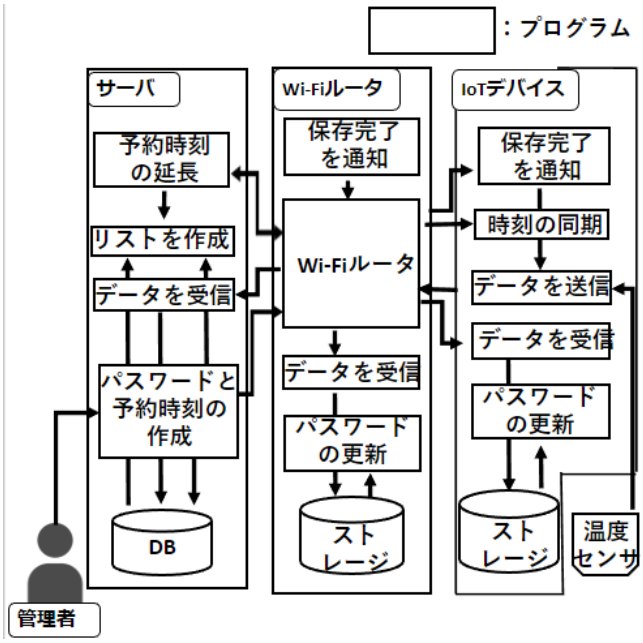

図 5　ソフトウェア構成図

更新すべき IoT デバイスのリストを作成するプログラム．

データ受信：サーバ，Wi-Fi ルータ，IoT デバイス

IoT デバイスかサーバから送られてくるデータを受信するプログラムである．受信したデータは DB かストレージに保存される．

パスワードと予約時刻の作成：サーバ

管理者が実行したらパスワードと予約時刻の作成を作成するプログラムである．作成したパスワードと予約時刻をパスワード更新すべき IoT デバイスのリストを参照して，IoT デバイスと Wi-Fi ルータに送信する．また複数台に向けて送信する際，1 台ずつ 4 秒の間隔を空けてパスワードと予約時刻を送信する．

保存完了を通知：Wi-Fi ルータ，IoT デバイス

サーバにパスワードと予約時刻を保存できたことを通知するプログラムである．Wi-Fi ルータと IoT デバイスがパスワードと予約時刻をストレージに保存出来た際に実行される．

Wi-Fi ルータのルーティング：Wi-Fi ルータ

アクセスポイントになるプログラムである．アクセスポイントになることで IoT デバイスと Wi-Fi 接続することが出来る．Wi-Fi 接続されている IoT デバイスから送られてくるデータをサーバに送信する．またサーバから送られてくるデータを IoT デバイスに送る役割もある．

パスワードの更新：Wi-Fi ルータ，IoT デバイス

パスワードの更新を行うプログラムである．ストレージに保存されている予約時刻が現在時刻と一致した際に実行される．ストレージに保存されているパスワードを用いてパスワードの更新を行う．

時刻の同期：IoT デバイス

サーバから時刻を取得して ESP 内の時刻と同期するプログラムである．サーバにデータを送信する際に，IoT デバイスの内蔵時計を現在時刻と同期する．

データ送信：IoT デバイス

データを Wi-Fi ルータを経由してサーバに送信するプログラムである．温度センサで測定した温度データを送信する．

5.　実験と分析

実験環境

本実験は Wi-Fi のパスワード更新にかかる時間を計測する．実験環境図が図 6 である．ESP32 と温度センサを搭載した IoT デバイス 2 台と Wi-Fi ルータにした ESP32 を搭載した IoT デバイス，Wi-Fi ルータ（ASUS TUF-AX5400），サーバを用いて実験を行う．Wi-Fi ルータにした IoT デバイス X が出力しているパスワード付きの Wi-Fi と IoT デバイス A と B が接続する．これにより，IoT デバイス A と B は IoT デバイス X を経由してサーバにデータを送信する構成により，IoT デバイス X を Wi-Fi ルータに見立てることが出来る．これにより IoT デバイス X が出力している Wi-Fi のパスワードを更新することで，Wi-Fi ルータのパスワード更新を疑似的に再現することが出来る．あらかじめ予約時刻を設定し，予約時刻になった場合，新しいパスワードを用いて IoT デバイスは Wi-Fi の再接続を行う．IoT デバイスが再接続を開始してから IoT デバイスを新しいパスワードで Wi-Fi に再接続するまでの時間を測定する．上記の測定を IoT デバイスが 1 台のときと 2 台のと

© 2022 Cloud and Distributed Systems Laboratory

きで行い比較する．

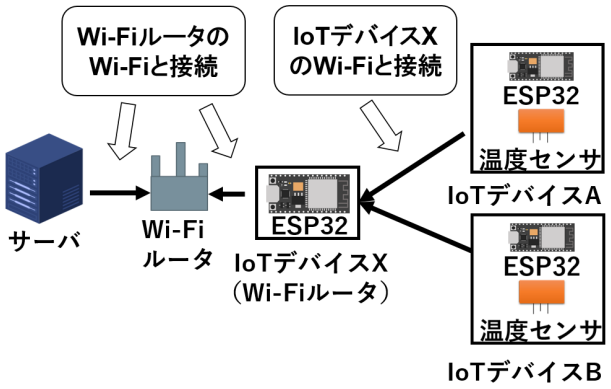

図 6　実験環境の構成図

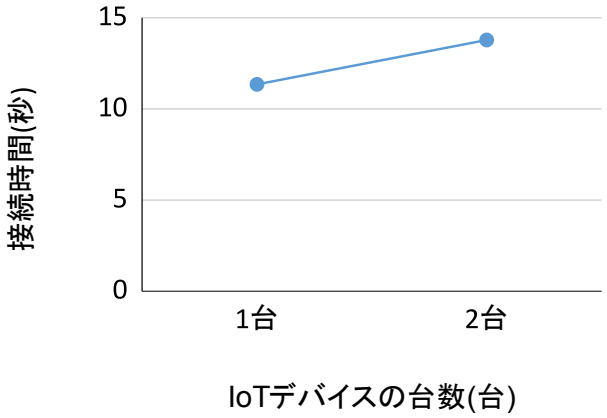

図 8　新しいパスワードを受け取るまでの時間

実験結果と分析

図 7 は予約時刻になり IoT デバイスが新しいパスワードを用いて Wi-Fi の再接続を開始してから完了するまでの時間を計測している．t_1 は予約時刻を登録した時刻である．t_2 は t_1 で登録した予約時刻を示している．Wi-Fi と IoT デバイスは予約時刻に Wi-Fi のパスワードを変更するプログラムを実行する．t_3 と t_4 は Wi-Fi への再接続が完了した時刻を示している．再接続が完了した時刻は，サーバーが接続確認の応答を受け取った時刻である．1 台のときは 1 台のみで 2 台のときは 2 台全てが再接続したときを示している．

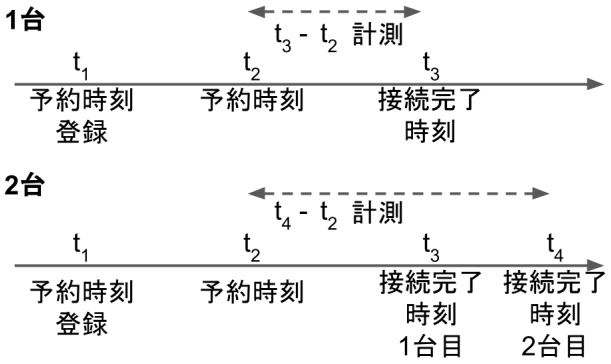

図 7　新しいパスワードを受け取るまでの時間

図 8 は IoT デバイスがサーバーから新しい Wi-Fi のパスワードを受け取ってからサーバーに受け取り完了通知が届くまでの時間を計測した結果である．

縦軸は時間を表し，単位は秒である．横軸は IoT デバイスの台数を表し，横軸の単位は台である．実験を 10 回行った結果，IoT デバイスの台数が 1 台のときの平均が 11.3 秒で，2 台のとき平均が 13.8 秒という結果になった．

上記の結果より 1 台のときと 2 台のときとでは 2.5 秒の差があった．予約時刻は同一であるにも関わらず，2.5 秒のズレがなぜ起きるかのかを分析する．

まず，RTC 自体にズレが生じていた場合である．この場合，時刻同期をしてから予約時刻までの時間を伸ばすことで IoT デバイスの再接続までの時間が変化した場合 RTC 自体にズレが生じたと分かる．

また，時刻同期をする際にズレが生じた場合である．この場合，予約時刻を迎えてから何秒後にプログラムを実行したかを計測することで時刻同期にズレが生じていたと分かる．

最後に，Wi-Fi の実装に用いたソケット通信の切り替えに時間がかかったことでズレが生じた場合である．現状，サーバで再接続の確認応答を受け取る時刻までを測定している．サーバだけでなく，Wi-Fi や IoT デバイスで再接続した時刻を記録する．IoT デバイスにおける Wi-Fi の再接続完了までの時刻とサーバーにおける Wi-Fi の再接続完了までの時刻を比較することで，ソケット通信の切り替えによるズレが生じたと分かる．

6.　議論

本稿では 24 時間以上サーバにデータを送信していない IoT デバイスを停止している IoT デバイスと判定している．停止している IoT デバイスはパスワード更新が行えない．本提案における停止している IoT デバイスの判定は 2 度行っている．1 つはパスワード更新をすべき IoT デバイスのリストを作成している際と，もう 1 つは予約時刻の延長する際である．最初にリストを作成している際に，サーバに保存されているセンサデータの更新日時が 24 時間以上経過している IoT デバイスを，停止している IoT デバイスと判定する．次にリスト内の IoT デバイス全台共通に予約時刻の延長に 24 時間の上限をつけることで停止している IoT デバイスの判定をしている．しかし，リスト作成の際に用いたセンサデータの更新日時を，予約時刻の延長の上限に参照できていない．そのため停止している IoT デバイスの判定に 24 時間以上要する場合が発生する．例とし

© 2022 Cloud and Distributed Systems Laboratory

て，リスト作成時にセンサデータの更新日時が 8 時間前の IoT デバイスがある．IoT デバイスはセンサデータの更新日時が 24 時間以内であるため，パスワード更新をすべき IoT デバイスのリストに入れられる．次に予約時刻の延長の上限として IoT デバイス全台共通で 24 時間が割り振られる．そのため，上記の IoT デバイスは停止の判定を 8 時間に 24 時間を足して 32 時間要していることになる．

解決策として，予約時刻の上限を IoT デバイス 1 台ずつで設定することで予約時刻の延長を短くする．予約時刻の上限はパスワード更新をすべき IoT デバイスのリストを作成するときにサーバ側で行う．サーバが IoT デバイス 1 台ずつの温度データを最後に受信した日時を用いて，予約時刻の上限を作成する．予約時刻の上限は温度データを最後に受信した日時から 24 時間後に設定する．例えば，温度データを最後に受信した日時が 8 時間前だった場合，予約時刻の上限を 16 時間後に設定する．これによって停止している IoT デバイスの判定を 24 時間で行える．

7. おわりに

Wi-Fi のパスワードを更新する際に，1 台ずつ IoT デバイスに設定されている Wi-Fi のパスワードも更新する必要がある．Wi-Fi 接続されている IoT デバイスが 9 台であった場合，1 台ずつ Wi-Fi 接続されている IoT デバイスのパスワードを更新するのは時間がかかる．本稿ではパスワード更新に要する時間を削減するために，IoT デバイスのリストとパスワード更新の予約時刻を作成しそれらを用いることで，パスワード更新の自動化する手法を提案した．予約時刻になった場合，新しいパスワードを用いて IoT デバイスは Wi-Fi の再接続を行う．IoT デバイスが再接続を開始してから接続完了までの時間を 1 台の場合と 2 台の場合でそれぞれ 10 回計測した．結果 IoT デバイスの台数が 1 台のときの平均が 11.3 秒で，2 台のときの平均が 13.8 秒になった．1 台の場合と 2 台の場合では 2.5 秒の差があった．提案手法により管理者は IoT デバイスを 1 台ずつパスワード更新するよりも，少ない時間でパスワード更新が行える．

参考文献

[1] Yeo, H.: Smart Farming Technology Review: Keynote Address, *2022 IEEE/ACIS 20th International Conference on Software Engineering Research, Management and Applications (SERA)*, pp. 2–2 (online), DOI: 10.1109/SERA54885.2022.9806473 (2022).

[2] Bandara, T. M., Mudiyanselage, W. and Raza, M.: Smart farm and monitoring system for measuring the Environmental condition using wireless sensor network - IOT Technology in farming, *2020 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA)*, pp. 1–7 (online), DOI: 10.1109/CITISIA50690.2020.9371830 (2020).

[3] Labouriau, L. G.: Seed germination as a thermobiological problem, *Radiation and Environmental Biophysics*, Vol. 15, No. 4, pp. 345–366 (1978).

[4] Boo, H.-O., Heo, B.-G., Gorinstein, S. and Chon, S.-U.: Positive effects of temperature and growth conditions on enzymatic and antioxidant status in lettuce plants, *Plant Science*, Vol. 181, No. 4, pp. 479–484 (2011).

[5] Amin, M. S., Rizvi, S. T. H., Iftikhar, U., Malik, S. and Faheem, Z. B.: IoT Based Monitoring and Control in Smart Farming, *2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*, pp. 1–6 (online), DOI: 10.1109/MAJICC53071.2021.9526247 (2021).

[6] Amandeep, Bhattacharjee, A., Das, P., Basu, D., Roy, S., Ghosh, S., Saha, S., Pain, S., Dey, S. and Rana, T.: Smart farming using IOT, *2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 278–280 (online), DOI: 10.1109/IEMCON.2017.8117219 (2017).

[7] Kumar, N. V. R., Praveen, B. S. B., Reddy, A. V. S. and Sam, B. B.: Study on IOT with reference of M2M and WiFi, *2017 International Conference on Information Communication and Embedded Systems (ICICES)*, pp. 1–6 (online), DOI: 10.1109/ICICES.2017.8070754 (2017).

[8] Kaur, J.: Wireless security issues and their emerging trends, *International Journal of Control Theory and Applications*, Vol. 10, No. 13, pp. 85–90 (2017).

[9] Zhong, X., Fan, C. and Zhou, S.: Eavesdropping area for evaluating the security of wireless communications, *China Communications*, Vol. 19, No. 3, pp. 145–157 (online), DOI: 10.23919/JCC.2022.03.010 (2022).

[10] Zou, Y., Zhu, J., Wang, X. and Hanzo, L.: A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends, *Proceedings of the IEEE*, Vol. 104, No. 9, pp. 1727–1765 (online), DOI: 10.1109/JPROC.2016.2558521 (2016).

[11] Callegati, F., Cerroni, W. and Ramilli, M.: Man-in-the-Middle Attack to the HTTPS Protocol, *IEEE Security & Privacy*, Vol. 7, No. 1, pp. 78–81 (2009).

[12] Lu, H.-J. and Yu, Y.: Research on WiFi penetration testing with Kali Linux, *Complexity*, Vol. 2021 (2021).

[13] Choi, J., Hur, J. and Bahk, S.: Push yoUr Password: Secure and Fast WiFi Connection for IoT Devices, *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6 (online), DOI: 10.1109/WCNC49053.2021.9417287 (2021).

[14] Mahalat, M. H., Saha, S., Mondal, A. and Sen, B.: A PUF based Light Weight Protocol for Secure WiFi Authentication of IoT devices, *2018 8th International Symposium on Embedded Computing and System Design (ISED)*, pp. 183–187 (online), DOI: 10.1109/ISED.2018.8703993 (2018).

[15] Sudar, C., Arjun, S. K. and Deepthi, L. R.: Time-based one-time password for Wi-Fi authentication and security, *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1212–1216 (online), DOI: 10.1109/ICACCI.2017.8126007 (2017).

[16] Kim, E. J., Kang, H., Jun, J. A. and Kim, N.-S.: The method of providing dynamic IP management function in a gateway for IoE, *2016 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 937–939 (online), DOI: 10.1109/ICTC.2016.7763335 (2016).

© 2022 Cloud and Distributed Systems Laboratory