# IoTデバイスにおけるWi-Fi接続のための暗号化キー更新の自動化

牧 丈晴[1]　大野 有樹[2]　串田 高幸[1]

**概要**：IoT デバイスはサーバーへセンサーデータを送信するためにインターネットに接続する必要がある．IoT デバイスの管理者は Wi-Fi を用いてインターネットに接続する際に，SSID と暗号化キーを入力し Wi-Fi ルーターへ接続する．不正アクセスを防ぐために Wi-Fi ルーターの暗号化キーを 1 年に 1 度変える際に 1 台ずつ手動で設定する場合，IoT デバイスの台数が増えるほど Wi-Fi ルーターの再接続をする移行作業にかかる時間が増加する．提案方式は暗号化キーと変更する予約時刻を IoT デバイスに送信して更新する．Wi-Fi ルーターに接続されている IoT デバイスに対して，管理者が設定した Wi-Fi ルーターの新しい暗号化キーをサーバーへ送信する．送られてきた暗号化キーを用いて IoT デバイスは管理者の設定した予約時刻に暗号化キーの更新を行う．更新した際に更新したことをサーバーに送信して送る．評価方法は 2 つの自動化の手法を実行時間を計測することで比較する．1 つは Wi-Fi ルーターの暗号化キーの変更を管理者のデバイスから送信して自動化する提案，もう 1 つは本提案手法であるサーバーを介した暗号化キーの取得による自動化である．

## 1. はじめに

### 背景

日本の農業では人材不足が問題視されている．人材不足を解消する手段の 1 つとして，IoT を農業に活用したスマート農業が挙げられる [1]．スマート農業では IoT デバイスが気温，水温，二酸化炭素濃度，光量を測定してサーバーにデータを送信することで，遠隔から測定したデータを確認できる [2,3]．スマート農業の活用例としてレタスの水耕栽培が挙げられる [4]．水耕栽培では 1 年を通してレタスを栽培するために，ビニールハウスで室温管理しながら行っている．レタスの商品としての質を上げるために，レタスの発育に適した温度である 15〜25 ℃の間で室温を保つ必要がある．レタス農家はビニールハウスの室温を常に把握しておく必要がある [5]．そのためレタス農家が温度センサを搭載した IoT デバイスをビニールハウスに設置することで，遠隔でビニールハウス内の室温を把握できるようになり，現地に赴く時間を削減できる．上記の使用例からスマート農業は人材不足を解消するのに一役買っている．

[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

IoT デバイスはサーバーへセンサデータを送信するためにインターネットに接続する必要がある．インターネットへの接続方法の 1 つである Wi-Fi は Wi-Fi ルーターを用いて接続する [6]．Wi-Fi ルーターには SSID と暗号化キーが設定してあり，IoT デバイスは接続したい Wi-Fi ルーターの SSID と暗号化キーを設定することで，Wi-Fi との接続が可能になる．

### 課題

不正アクセスを防ぐために Wi-Fi ルーターの暗号化キーを変える際に，Wi-Fi ルーターに接続している IoT デバイスの台数が増えるほど Wi-Fi ルーターの再接続をする移行作業にかかる時間が増加することが課題である．IoT デバイスはサーバーにセンサーデータを送信するためにインターネットへ接続する必要がある．IoT デバイスは Wi-Fi を用いてインターネットに接続する際に，管理者は SSID と暗号化キーを入力し Wi-Fi ルーターへ接続する．図 1 は IoT デバイスを 1 台ずつ Wi-Fi ルーターに接続する様子を示している．管理者は Wi-Fi に接続している IoT デバイスに IoT デバイスの管理者のデバイスを用いて新しい暗号化キーを設定して Wi-Fi ルーターに再接続する．ひとつひとつに暗号化キーを入力する必要があるため，IoT デバイスの台数が増えるほど Wi-Fi ルーターに再接続する時間が増加する．

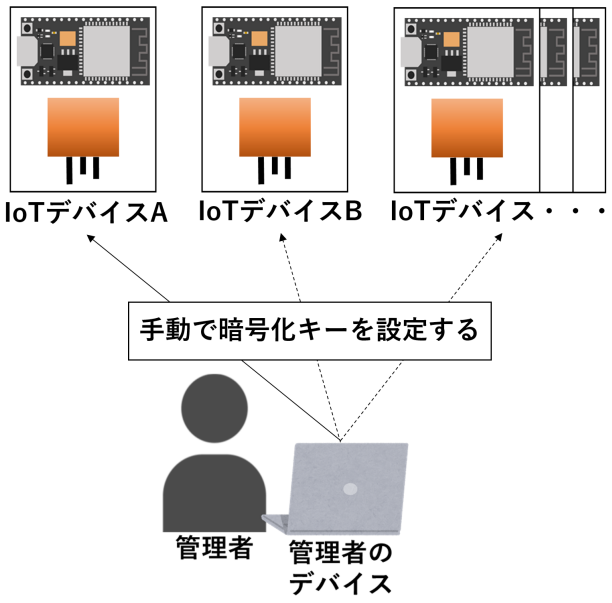


**図 1**　手動で IoT デバイスへ Wi-Fi の暗号化キーを設定

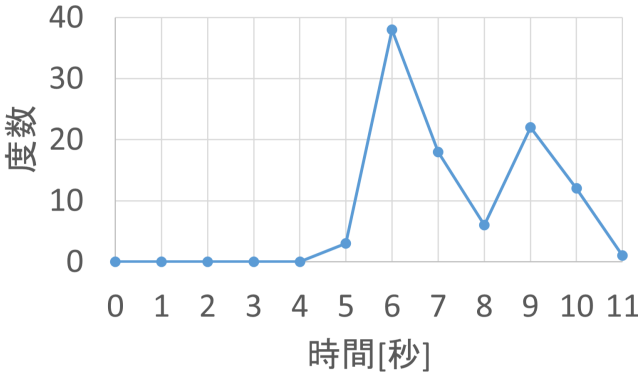


**図 2**　Wi-Fi の暗号化キーを取得して再接続する時間の度数分布

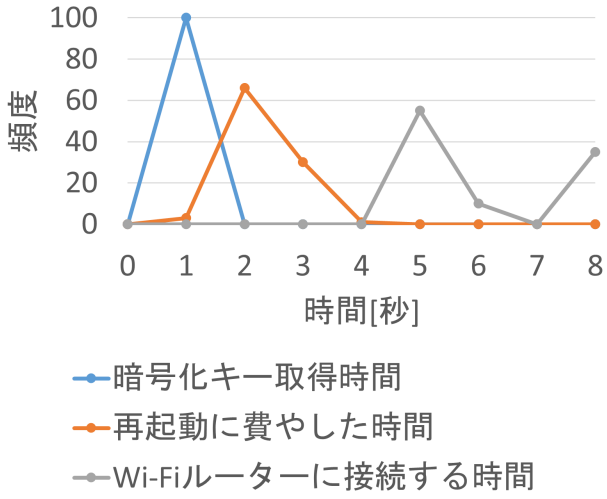


**図 3**　Wi-Fi の暗号化キーを変更する時間の内訳の度数分布

### 基礎実験

実際に IoT デバイスを Wi-Fi ルーターに接続するためにサーバーから暗号化キーを取得して Wi-FI ルーターに再接続するまでの時間を計測する．IoT デバイスとして実験では ESP32 を使用した．本基礎実験は手作業で行う暗号化キーの更新を自動化した場合にかかる時間を測定する．あらかじめ暗号化キーをサーバーに保存しておきサーバーで暗号化キーを取得できるようにする．IoT デバイスとサーバの間は HTTP で通信を行う．IoT デバイスは HTTP の GET メソッドでサーバから暗号化キーを取得する．

Wi-Fi ルーターへの再接続の時間は 3 つに分けられ，暗号化キー取得時間，再起動に費やした時間，Wi-Fi ルーターに接続する時間となる．暗号化キー取得時間は RTC で計測した．再起動に費やした時間は差分から求めた．NTP で計測した全体にかかった時間を a とする．暗号化キー取得時間を b とする．Wi-Fi ルーターに接続する時間を c とする．このとき先起動に費やした時間は $a-(b+c)$ と算出した．暗号化キー取得時間と Wi-Fi ルーターに接続する時間は ESP32 内蔵の RTC で計測した．

図 2 は基礎実験の結果を示している．縦軸は度数を表している．横軸は時間を表し，単位に秒をとる．実験は Wi-Fi ルーターへの再接続の時間を 100 回計測した．結果として，4 秒から 5 秒までは 3 回，5 秒から 6 秒までは 38 回，6 秒から 7 秒までは 18 回，7 秒から 8 秒までは 6 回，8 秒から 9 秒までは 22 回，9 秒から 10 秒までは 12 回，10 秒から 11 秒までは 1 回となった．Wi-Fi の再接続は最小で 4 秒，最大で 11 秒かかることが分かった．

図 3 は基礎実験の結果の内訳を示している．暗号化キー取得時間は 1 秒にも満たさないが，再起動に費やした時間は 0 秒から 4 秒，Wi-Fi ルーターに接続する時間は 4 秒から 8 秒とばらつきがあった．再接続までの時間にばらつきがあるのは再起動に費やした時間と Wi-Fi ルーターに接続する時間にばらつきがあるためである．

### 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，本研究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と実験方法を説明する．第 5 章では，第 3 章の提案手法が課題を解決しているかを判断するための評価手法と分析手法を説明する．第 6 章では，提案，実験，評価が本研究の課題を解決しているかを議論する．第 7 章では，本研究の課題解決による貢献を説明する．

## 2. 関連研究

Jiafu らは，IoT デバイスの動的なリソース管理のためのプラットフォームを提案した [3]．この提案は IoT デバイスを相互で通信して IoT デバイスの管理をしていた．しかし，新たにインターネットへ繋ぎなおすための手間を考慮

しないシステムのため，Wi-Fi の暗号化キーが変わったときの手間に関する利点がない．

Marco らは，Wi-Fi を用いた家庭および個人の監視システムのアーキテクチャを提案した [7]．この提案は消費電力が ZigBee より高いものの Wi-Fi ルーターが普及しているためインストールを大幅に短縮できる利点がある．しかし，Wi-Fi ルーターへの接続や暗号化キーを変更したときの再接続を考慮していないため，Wi-Fi ルーターへの接続の手間を考慮する必要がる．

Kranthi らは，家庭菜園や畑の灌漑に必要な水を自動で供給するシステムを提案した [8]．この提案は ZigBee を用いて IoT デバイス間の通信をして，近隣の Wi-Fi ルーターを検索し認定された Wi-Fi ルーターへ接続する．しかし，Wi-Fi ルーターを認定する方法が記載されていないため Wi-Fi ルーターを認定する手順が必要になる．

## 3. 提案方式

Wi-Fi ルーターに接続されている IoT デバイスに対して，管理者が設定した Wi-Fi ルーターの新しい暗号化キーを送信する．送られてきた暗号化キーを用いて IoT デバイスは管理者の設定した予約時刻に自動で Wi-Fi ルーターと IoT デバイスの更新を行う．

### 前提条件

- センサデータの測定間隔は IoT デバイス全台で共通して 15 分毎
- IoT デバイスは Wi-Fi を用いてインターネットと接続
- センサデータをサーバーに送信するたびに NTP を用いて時刻の同期

### 提案方式

図 4 は IoT デバイスへの暗号化キーの送信を示している．(1)(2) は暗号化キーの登録を示しており，(3)(4) はサーバーからの暗号化キーの送信を示している．暗号化キーはサーバーを通して IoT デバイスに送信される．Wi-Fi ルーターへ接続されている IoT デバイスはサーバーにセンサーデータを送信しているデバイスであると定義する．Wi-Fi ルーターに接続している IoT デバイスはサーバーに登録された暗号化キーをサーバーから取得する．サーバーから暗号化キーを取得する理由としては，Wi-Fi ルーターに接続されている IoT デバイスを識別するためである．暗号化キーを渡すデバイスを限定するために，サーバーで識別して IoT デバイスのみに暗号化キーを送信する．

提案の流れを図 5 を用いて説明する．

(1) では管理者が Wi-Fi ルーターの新しい暗号化キーを考えて，サーバーに送信する．このとき，予約時刻もサーバーへ送信する．(2) では管理者から送られた暗号化キーと予約時刻ををストレージに保存する．(3) では IoT デバイスは新しい暗号化キーをサーバーに要求する．(4) ではサーバーが内部のストレージから暗号化キーと予約時刻を取得する．(5) では暗号化キーと予約時刻を送信サーバーから」IoT デバイスへ送信する．(6) ではサーバーから受け取った暗号化キーと予約時刻をストレージにファイル形式で保存する．(7) ではサーバーから受け取った予約時刻になるとき新しい暗号化キーで Wi-Fi ルーターに接続する．

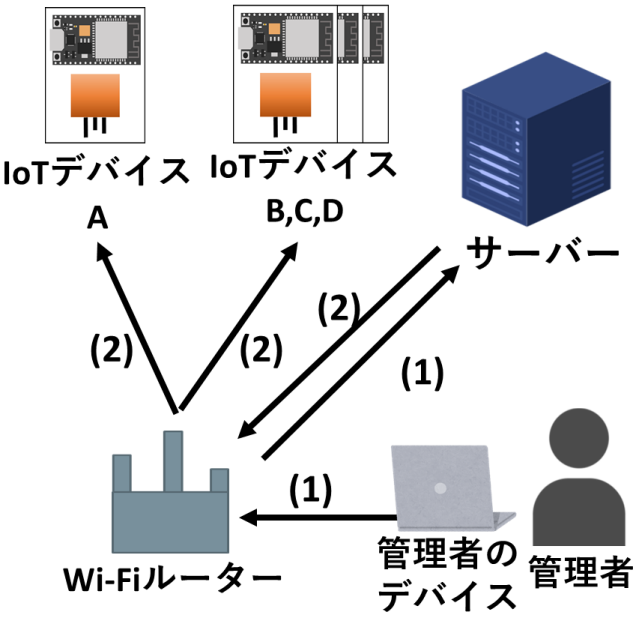


図 4　暗号化キーの送信

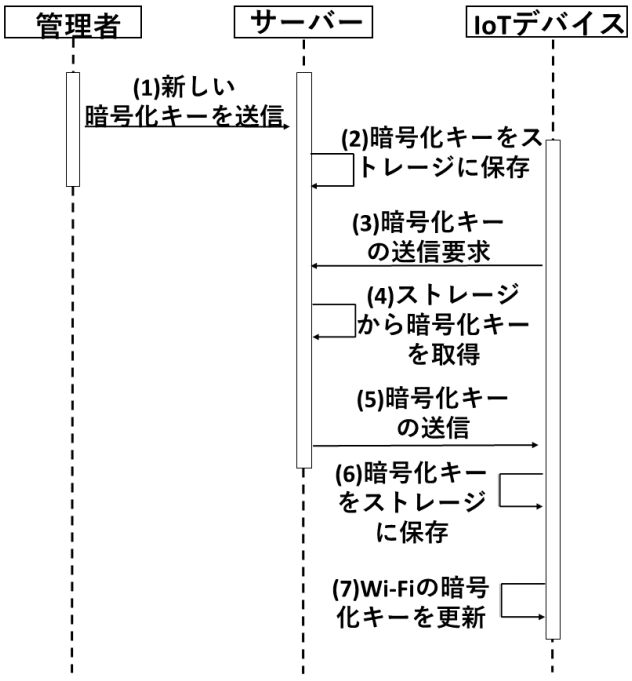


図 5　提案の流れ図

### ユースケース・シナリオ

ビニールハウスでレタスを水耕栽培してる際に，ビニールハウスの室温を測定して Wi-Fi ルーターを経由してサー

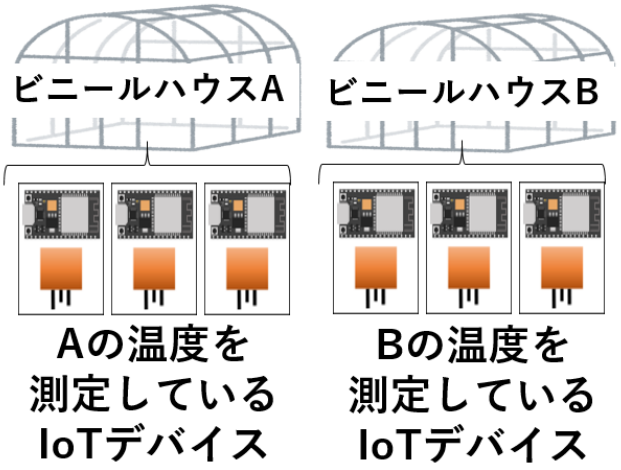


**図 6** IoT デバイスとビニールハウスの配置

バーに送信する IoT デバイスを想定する．ビニールハウスの大きさは幅 7.4m・奥行 50m・高さ 3m である．図 6 のようにビニールハウス 1 棟につき 3 個の IoT デバイスが設置されており，3 個の IoT デバイスを用いてビニールハウス 1 棟の室温を測定している．

## 4. 実装と実験方法

### 実装

実装は大きく 3 つに分けられ，Wi-Fi ルーターの暗号化キーの更新，IoT デバイスの暗号化キーの更新，IoT デバイスが暗号化キーを取得がある．Wi-Fi ルーターの暗号化キーを更新は管理者が Wi-Fi ルーターへ暗号化キーと変更する予約時刻を登録する．IoT デバイスの暗号化キーの更新は同様に管理者がサーバーへ暗号化キーと変更する予約時刻を登録する．IoT デバイスが暗号化キーを取得は IoT デバイスがサーバーから暗号化キーを受け取る．

図 7 は本提案手法の実装のソフトウェア構成図を示している．Wi-Fi ルーターの暗号化キーの更新は I から Ⅲ に示している．I では管理者が管理者のデバイスを用いて予約時刻プログラムに送信している．Ⅱ では同様に管理者が管理者のデバイスを用いて暗号化キーを送信している．Ⅲ では予約時刻プログラムが予約時刻になったとき暗号化キー更新プログラムを実行し，暗号化キー保存プログラムから暗号化キーを更新する．

IoT デバイスの暗号化キーの更新は (1) から (6) に示している．(1) では管理者は管理者のデバイスを用いてサーバーの暗号化キー保存プログラムへ暗号化キーを送信している．(2) では送信した暗号化キーをストレージに保存している．(3) では管理者は管理者のデバイスを用いてサーバーの予約時刻プログラムへ予約時刻を送信している．(4) ではサーバーから IoT デバイスへ予約時刻プログラムを通して予約時刻を送信している．(5) では受け取った予約時刻をもとに予約時刻プログラムが Wi-Fi 更新プログラムを

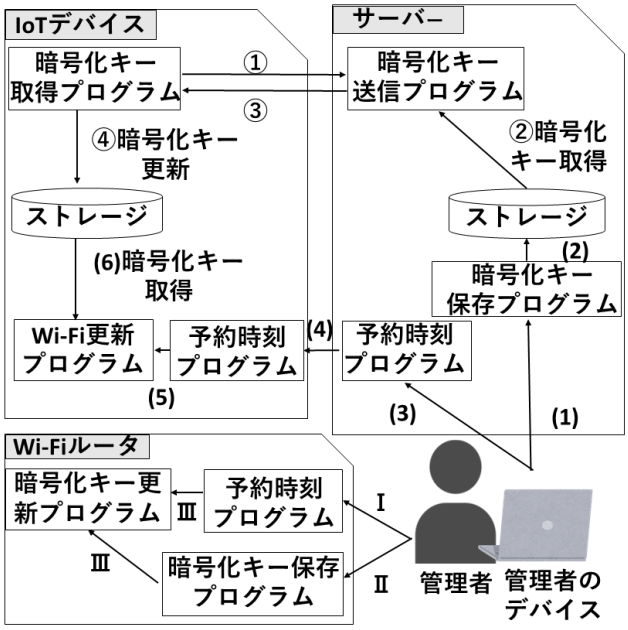


**図 7** 実装のソフトウェア構成図

実行する．(6) では Wi-Fi 更新プログラムが IoT デバイスのストレージから暗号化キーを取得する．

IoT デバイスが暗号化キーを取得は①から④に示している．①では暗号化キー取得プログラムがサーバーに暗号化キーの要求をする．②では暗号化キー取得プログラムがストレージから暗号化キーを取得する．③では IoT デバイスへ暗号化キー取得プログラムが暗号化キーを送信する．④暗号化キー取得プログラムが IoT デバイスのストレージへ暗号化キーを更新する．

### 実験環境

表 1 は実験対応表である．実験に使用する機器はサーバー側とクライアント側に分かれる．サーバー側は VMware ESXi を使用し，VM でサーバーを構築する．VM は Ubuntu 20.04.2 LTS で構築する．サーバー側の実装は Python 3.8.10 を使用し，Flask2.0.2 で HTTP サーバーを構築する．使用した言語は MicroPython 1.17.0 である．接続完了のメッセージは HTTP でサーバー側に送信した．クライアント側は 10 台の ESP32 を使用して実験をする．

**表 1** 実験対応表

| | |
|---|---|
| VM | Ubuntu 20.04LTS |
| サーバー | Flask 2.0.2 |
| マイクロコントローラ | ESP32 |
| 使用言語 | MicroPython 1.17.0 |

## 5. 評価手法と分析手法

IoT デバイスへ SSID と暗号化キーを入力して Wi-Fi に接続する状況を想定して評価する．2 つの自動化の手法を

実行時間を計測することで比較する．1 つは Wi-Fi の暗号化キーを変更する作業を自動化した提案，もう 1 つは本提案手法であるサーバーを介した暗号化キーの取得による自動化である．

1 つ目の Wi-Fi の暗号化キーを変更する作業の自動化は既に繋がっている Wi-Fi を用いて遠隔で Wi-Fi ルーターの新しい暗号化キーを送信する方法である．新しい暗号化キーを受信した IoT デバイスは次の起動時に新しい暗号化キーを使用して Wi-Fi ルーターへ接続する．時間の計測は IP アドレスを用いて IoT デバイスへ接続する時間から新しい暗号化キーを用いて Wi-Fi ルーターへ接続するまでの時間を計測する．

2 つ目の本提案手法の実装は実装の章の通りにする．時間の計測は新しい暗号化キーをサーバーに入力してから各 IoT デバイスが新しい暗号化キーで Wi-Fi ルーターへ接続するまでとする．

## 6. 議論

本提案手法は暗号化キーを変更したときの Wi-Fi ルーターへの再接続を自動化した．再接続する際に Wi-Fi に接続されている IoT デバイス全台が，サーバーから暗号化キーを取得できることを条件にしている．ただし，実際は通信エラーによって暗号化キーの取得に失敗した IoT デバイスを考慮する必要がある．解決方法として Wi-Fi の暗号化キーを変更する前に，暗号化キー取得に関する IoT デバイスのリストを作成する．

(1) サーバーと通信が出来ている IoT デバイスのリスト
(2) 暗号化キーを保存したことの確認応答がある IoT デバイスのリスト

上記の 2 つのリストを作成する．リスト (1) からリスト (2) の差を求めた結果が 1 以上の場合，暗号化キー取得に失敗した IoT デバイスが存在することになる．

暗号化キーの取得に失敗した IoT デバイスを見つけた場合，センサデータの送信間隔を基に暗号化キーを変更する予約時刻を遅らせる．本研究では 15 分間隔でセンサデータをサーバーに送信しているので，15 分毎に予約時刻を遅らせる．予約時刻を遅らせることで，IoT デバイスは暗号化キーを取得する．しかし、上限無く予約時刻を遅らせると暗号化キーの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 1 日と設定する．理由としては，データを測定する IoT デバイスが最低でも 1 日 1 回データを測定している．なぜなら過去のセンサデータから未来のセンサデータは予測できない．そのため最低でも 1 日 1 回データを測定することが機能要件となる．この機能要件から 1 日以上経過して予約時刻を取得できない IoT デバイスは，停止した IoT デバイスと定義づけられる．

## 7. おわりに

本研究の課題は Wi-Fi ルーターの暗号化キーを変更する際に，Wi-Fi ルーターに接続されている IoT デバイスの台数が多いほど暗号化キーの更新に必要な時間が長いことである．そこで Wi-Fi ルーターの変更予定の暗号化キーを事前に IoT デバイスに送信することで，自動で暗号化キーを変更する提案をした．Wi-Fi ルーターに接続されている IoT デバイスに対して，管理者が設定した Wi-Fi ルーターの新しい暗号化キーをサーバーへ送信する．送られてきた暗号化キーを用いて IoT デバイスは管理者の設定した予約時刻に暗号化キーの更新を行う．更新した際に更新したことをサーバーに送信して送る．評価方法は 2 つの自動化の手法を実行時間を計測することで比較する．1 つは Wi-Fi ルーターの暗号化キーの変更を管理者のデバイスから送信して自動化する提案，もう 1 つは本提案手法であるサーバーを介した暗号化キーの取得による自動化である．

### 参考文献

[1] Vij, A., Vijendra, S., Jain, A., Bajaj, S., Bassi, A. and Sharma, A.: IoT and Machine Learning Approaches for Automation of Farm Irrigation System, *Procedia Computer Science*, Vol. 167, pp. 1250–1257 (online), DOI: https://doi.org/10.1016/j.procs.2020.03.440 (2020). International Conference on Computational Intelligence and Data Science.

[2] Balaji, G. N., Nandhini, V., Mithra, S., Priya, N. and Naveena, R.: IoT based smart crop monitoring in farm land, *Imperial Journal of Interdisciplinary Research (IJIR)*, Vol. 4, No. 1, pp. 88–92 (2018).

[3] Wan, J., Chen, B., Imran, M., Tao, F., Li, D., Liu, C. and Ahmad, S.: Toward Dynamic Resources Management for IoT-Based Manufacturing, *IEEE Communications Magazine*, Vol. 56, No. 2, pp. 52–59 (online), DOI: 10.1109/MCOM.2018.1700629 (2018).

[4] Bakhtar, N., Chhabria, V., Chougle, I., Vidhrani, H. and Hande, R.: IoT based Hydroponic Farm, *2018 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 205–209 (online), DOI: 10.1109/ICSSIT.2018.8748447 (2018).

[5] 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 (online), DOI: https://doi.org/10.1016/j.plantsci.2011.07.013 (2011).

[6] Juhász, K., Póser, V., Kozlovszky, M. and Bánáti, A.: WiFi vulnerability caused by SSID forgery in the IEEE 802.11 protocol, *2019 IEEE 17th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pp. 333–338 (online), DOI: 10.1109/SAMI.2019.8782775 (2019).

[7] Bassoli, M., Bianchi, V., De Munari, I. and Ciampolini, P.: An IoT Approach for an AAL Wi-Fi-Based Monitoring System, *IEEE Transactions on Instrumentation and Measurement*, Vol. 66, No. 12, pp. 3200–3209 (online), DOI: 10.1109/TIM.2017.2753458 (2017).

[8] Kumar, M. K. and Ravi, K. S.: Automation of irrigation system based on Wi-Fi technology and IOT, *Indian*

*Journal of Science and Technology*, Vol. 9, No. 17, pp. 1–5 (2016).

 