Bluetooth over DTLS による IoT デバイスの遠隔制御システムに関する研究

163430006　岡田 真実
鈴木研究室

1. はじめに

IoT （Internet of Things）デバイスを制御するための短距離無線通信規格の 1 つとして Bluetooth がある．しかし，Bluetooth は通信可能範囲が限定されているため，宅外から宅内の機器を Bluetooth の仕組みで直接遠隔制御することができない．そのため，Bluetooth の制御メッセージに着目した遠隔制御システムが提案されている [1]．しかし，文献 [1] は従来の Bluetooth 搭載 IoT デバイスのみが制御対象となっており，近年普及が進んでいる BLE（Bluetooth Low Energy）搭載 IoT デバイスに対応しておらず，セキュリティの考慮が不十分である．

そこで，本研究では従来研究に対して，DTLS（Datagram Transport Layer Security）[2] による暗号化トンネルを利用して従来の Bluetooth および BLE の制御メッセージを遠隔地へ伝送することにより，遠隔地に存在する Bluetooth 搭載 IoT デバイスのセキュアな遠隔制御システムを提案する．また，提案システムのプロトタイプ実装および実環境における性能評価を行った．

2. 従来研究

Bluetooth はソフトウェアで構成される Host と，ハードウェアで構成される Controller で構成されており，Host と Controller の間で HCI（Host Controller Interface）メッセージを交換することで通信を行っている．文献 [1] では，Bluetooth の HCI メッセージのやりとりに着目し，HCI メッセージを UDP（User Datagram Protocol）通信を利用することで遠隔地に伝送する．この手法では，操作端末の近隣に専用のハードウェアを不要とし，かつユーザは宅内と同じ通常の Bluetooth アプリケーションにより遠隔地の IoT デバイスとの通信が可能となる．

しかし，このシステムは BLE 搭載 IoT デバイスに未対応であり，UDP 通信部のセキュリティが考慮されていない．

3. 提案システム

3.1 概要

図 1 に提案システムの概要を示す．提案システムは操作端末 CD（Control Device），CD の近隣に存在する Bluetooth 搭載 IoT デバイスを ND（Neighbor Device），宅内に設置される専用の機器 BGW（Bluetooth Gateway），BGW の近隣に存在し CD の遠隔制御対象となる Bluetooth 搭載 IoT デバイスを RD（Remote Device）から構成されている．以後，Bluetooth 搭載 IoT デバイスを単に IoT デバイスと表記する．また，BGW は Bluetooth Interface（I/F）と IP I/F を搭載している．

提案システムでは，従来研究と同様に Bluetooth プロトコルスタックにおけるホストとコントローラの間で交換されている HCI メッセージのコピーをフックし遠隔地に設置した BGW へ UDP を利用し送信する．その際，UDP 通信を保護するための標準化技術である DTLS（Datagram Transport Layer Security）を適用する．DTLS を利用したトンネル通信を行うことで，CD から BGW の Bluetooth コントローラに対してデバイス間のセキュリティを考慮しつつ HCI メッセージを届けることができる．これにより，CD は BGW の Bluetooth インタフェースを自身のインタフェースであるかのように操作することが可能となり，自身の Bluetooth ホストでだけではなく DTLS トンネル通信により BGW から RD の情報を受け取ることができる．

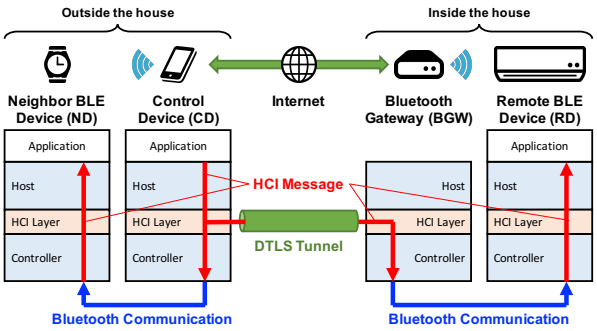

図 1: 提案システムの概要

これにより，ユーザは専用の装置を携帯する必要が無く，かつ場所の違いに影響されることなく常に単一の操作アプリケーションで同じように ND と RD を同時に探索でき，ユーザが選択した IoT デバイスを操作することができる．

3.2 通信シーケンス

事前の準備として，ユーザはあらかじめ BGW との間でユーザ認証処理を行う．なお，自宅の NAT/ファイアウォール（FW）に対して宅外から BGW に対して DTLS 通信ができるよう，ポートフォワーディングの設定を行っているものとする．

提案システムでは CD と BGW の双方向認証を行うため，CD が宅外のネットワークに接続すると，BGW に対して DTLS ハンドシェイクを開始し，両者は自身の公開鍵証明書を交換して共通鍵を生成する．CD および BGW は DTLS の枠組み内で双方向認証が行われ，DTLS ハンドシェイクが完了するとユーザ認証も完了し，DTLS トンネルが生成される．

ユーザ認証処理による DTLS トンネル生成後，CD 側で Bluetooth 通信を行う操作アプリケーションを起動して近隣の Bluetooth 機器の探索を開始すると，Bluetooth ホストからコントローラに向けて HCI メッセージが渡される．ここで，HCI 層において HCI メッセージを複製し，元の HCI メッセージはそのままコントローラに渡して，CD は自身の Bluetooth インタフェースを用いて近隣の IoT デバイス ND を探索する．ND を発見した場合は通常の Bluetooth プロトコルスタックの処理手順により探索結果がアプリケーションに渡される．一方，複製された HCI イベントメッセージはユーザ認証処理で構築した DTLS トンネルを利用して BGW へ伝送される．BGW は CD から HCI メッセージを受信すると，自身の Bluetooth コントローラへ HCI メッセージを渡し，BGW の近隣に存在する IoT デバイス機器 RD を探索する．RD を発見した場合は逆の手順により HCI メッセージを CD 側へ伝送する．DTLS トンネルから受信した HCI メッセージが CD の HCI 層へ渡されると，HCI メッセージに記載された BD（Bluetooth Device）アドレス，DTLS トンネルの送信元

IP アドレスとポート番号[1] をトンネルテーブルに記録し，HCI メッセージを Bluetooth ホストへ渡す．以上の処理により，操作アプリケーションに RD の情報も渡される．

発見した IoT デバイスを Bluetooth 通信で操作する際は，操作したい IoT デバイスの BD アドレスを宛先とした HCI メッセージがホストからコントローラに向かって渡されるため，宛先の BD アドレスを用いてトンネルテーブルを検索することで RD および ND の判別を行う．宛先が RD の場合は該当するエントリが見つかるため HCI メッセージを複製せずにフックし，DTLS トンネルを利用して BGW へ伝送する．以後は機器探索時と同じ手順で HCI メッセージを処理する．一方，宛先が ND の場合はトンネルテーブルに該当するエントリが存在しないため，HCI メッセージをフックせずに自身のコントローラへ渡し，通常の Bluetooth の仕組みで ND と Bluetooth 通信を行う．

4. 実装

提案システムを実現するためには，Bluetooth プロトコルスタックにおける HCI 層で HCI メッセージの複製を BGW へ伝送し，BGW 側から受け取った HCI メッセージを CD のプロトコルスタック戻す処理が必要である．そこで，提案システムを実現するために，Linux PC を利用し，Linux に実装されている Bluetooth プロトコルスタック BlueZ のカーネルモジュールを拡張した．また，DTLS トンネルの構築，ユーザ認証処理および HCI メッセージの伝送を行うために HCI Forwarder デーモンを実装した．図 2 にモジュール構成を示す．

CD 側の BlueZ に実装されている HCI 層における HCI メッセージの送信処理部に HCI メッセージが格納されているソケットバッファを複製する処理を追加した．その際，従来の Bluetooth 規格の HCI メッセージだけでなく，BLE の HCI メッセージもフックすることで両規格に対応させた．また，複製した HCI メッセージを Netlink ソケットを用いて，ユーザランドで動作している HCI Forwarder デーモンへ渡す処理を追加した．

HCI Forwarder デーモンは CD と BGW においてバックグラウンドで動作し，拡張した Bluetooth カーネルモジュールとのメッセージ交換を行う Netlink ソケットおよび DTLS トンネルを用いたメッセージ送受信を行うためのデータグラムソケット，トンネルテーブルを作成した．また，DTLS ハンドシェイクおよび暗号化や認証処理を行うために，OpenSSL ライブラリ（version 1.0.2g）を利用した．なお，DTLS のプロトコルバージョンは 1.2 を採用した．

5. 評価

図 3 に示す環境において，プロトタイプ実装した提案システムの動作検証および性能評価を行った．Ubuntu 17.04 をインストールしたラップトップ PC を CD および BGW とし，Bluetooth4.0+EDR/LE 対応 USB アダプタ（BUFFALO 社製 BSBT4D09BK を装着した．CD はスマートフォンをテザリングして 4G LTE 回線によりインターネットに接続し，BGW は研究室 LAN に設置してブロードバンドルータ（BBR）を通じてインターネットに接続した．CD の近隣には EDR 対応の Bluetooth スピーカー（日本電話施設社製 OmniBeat を ND として，また BGW の近隣には BLE 対応の学習リモコンユニット（ラトックシステム社製 REX-BTIREX1 を RD としてそれぞれ配置し，CD から直接 RD を探索できない位置関係とした．

上記の環境において，公開鍵証明書を用いたユーザ認証処理を行った後，CD 上で Bluetooth 機器を探索するコマンド hcitool inq を実行し，ND および RD を探索できるか確認し，Wireshark を用いて CD が最初に HCI コマンドを発行してから ND および RD の情報が記載された最初の HCI イベントを受け取るまでに要した時間を計測した．また，公開鍵証明書の RSA 公開鍵長は 2048bit，ダイジェストアルゴリズムは SHA-256 として生成したものを用いた．

[1]BGW が設置されたホームネットワークのゲートウェイに当たる NAT の WAN 側グローバル IP アドレスと，事前にポートフォワード設定したポート番号．

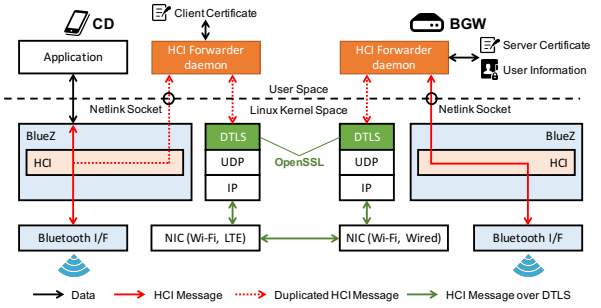

図 2: モジュール構成

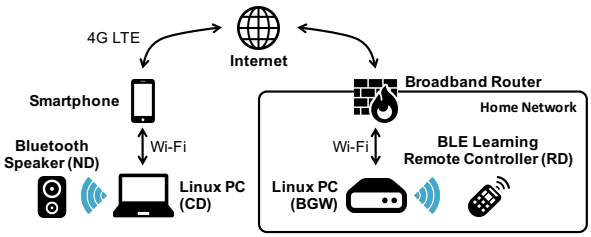

図 3: 測定環境

表 1: ユーザ認証時間と機器探索時間の測定結果

	Min [ms]	Avg [ms]	Max [ms]
User authentication	255.16	332.74	445.71
ND discovery	122.99	685.49	974.40
RD discovery	102.79	489.92	907.39

表 1 に提案システム適用時におけるユーザ認証時間と機器探索時間を 10 回測定した結果を示す．提案システムにおける CD および BGW 間で行われたユーザ認証処理は 445.71 ミリ秒で完了しており，ユーザが宅外のネットワーク接続した時に 1 回だけ発生する処理であることを考えると，実用上問題にならない．また，ND および RD の探索時間はそれぞれ Bluetooth の仕様で定義されている探索間隔 2.56 秒の範囲で収まっていることから，実用上問題ないと考えられる．以上の結果から，提案システムを適応しても Bluetooth のタイムアウト時間内に遠隔地の機器探索が可能であることを確認した．

6. まとめ

本研究では，遠隔地にある Bluetooth 搭載 IoT デバイスを仮想的に発見，接続および通信するシステムを提案した．また，提案システムのプロトタイプ実装を行い，実環境において動作検証および通信遅延を評価した．結果，Bluetooth で規定されているタイムアウト時間内に CD が遠隔地の RD を発見できることを確認した．

参考文献

[1] Tsuda, K., et al.: Proposal for a Seamless Connection Method for Remotely Located Bluetooth Devices, *Proc. of ICMU 2014*, pp. 78–79 (2014).

[2] Rescorla, E., et al.: Datagram Transport Layer Security Version 1.2, RFC 6347, IETF (2012).