情報処理学会論文誌 Vol.59 No.2 392–403 (Feb. 2018)

非構造化P2P/オーバレイネットワークにおけるセキュリティ方式の提案およびPUCCプロトコルを用いた実装と評価

加藤 剛志[1,a)] 石川 憲洋[2,b)] 吉田 尚史[2,c)]

受付日 2017年4月29日, 採録日 2017年11月7日

概要：従来のPCやスマートフォンに加えて，情報家電やセンサ，ウェアラブル端末など，様々なデバイスが通信機能を搭載しネットワーク化され始めている．本研究では異なるネットワークに属する複数のデバイスどうしがアドホックにネットワークを構築する仕組みとして非構造化P2P/オーバレイネットワークに着目し，そのセキュリティ方式について検討した．情報家電やセンサなど多種多様なデバイスがネットワーク化されると，認証や暗号化のための電子証明書の登録や再配布が困難であったり，特定のGW装置などを経由したアクセスのみ信頼する必要があったりするなどの課題がある．本研究ではそれらの課題を解決する方式として，新たにセキュアマルチホップセッションという概念を考案し，P2P/オーバレイネットワークのマルチホップセッション上で多段階認証および暗号化通信を実現するセキュリティ方式を提案する．提案方式をPUCC（P2P Universal Computing Consortium）で規定されているPUCCプロトコル上で設計および実装し，その性能評価を行い，提案方式の有効性を確認した．

キーワード：オーバレイネットワーク，ホームネットワーク，セキュリティ，PUCC，IoT，M2M

Security Mechanism for Unstructured P2P/Overlay Network and Its Implementation and Evaluation Using PUCC Protocols

Takeshi Kato[1,a)] Norihiro Ishikawa[2,b)] Naofumi Yoshida[2,c)]

Received: April 29, 2017, Accepted: November 7, 2017

Abstract: In recent years, various devices such as home appliances, sensor devices and wearable devices, which have communication capability, communicate with each other. We study on security mechanism for an unstructured P2P/overlay network, which has an ad-hoc networking function among devices across heterogeneous physical networks. In such environments, there are some security challenges such as difficulty of redistributing and/or updating digital certificates of many various devices across heterogeneous networks, and establishing a secure session between devices via a reliable device (e.g., HGW). This paper newly proposes the concept of secure multi-hop session, and proposes a security mechanism, which executes multi-step authentication/encryption functions over a multi-hop session on an unstructured P2P/overlay network. We also design and implement the security protocol based on the proposed method using PUCC protocols defined by PUCC (P2P Universal Computing Consortium), and evaluated its performance and confirmed its feasibility.

Keywords: overlay network, home network, security, PUCC, IoT, M2M

[1] 駒澤大学大学院グローバル・メディア研究科
Graduate School of Global Media, Komazawa University, Setagaya, Tokyo 154–8525, Japan
[2] 駒澤大学グローバル・メディア・スタディーズ学部
Faculty of Global Media Studies, Komazawa University, Setagaya, Tokyo 154–8525, Japan
[a)] 3715201t@komazawa-u.ac.jp
[b)] isic@komazawa-u.ac.jp
[c)] naofumi@komazawa-u.ac.jp

1. はじめに

近年，無線LANやBluetoothなどの無線技術の発展にともない，携帯電話が従来のセルラー通信インタフェースに加えてローカル通信インタフェースを搭載したり，ホームゲートウェイ（HGW）が無線LANとBluetoothを搭載したりするなど，複数の異なる無線ネットワークインタ

フェースを持つデバイスが増加している．また，プロセッサ技術の進歩によりデバイスの小型化や低価格化が進み，従来の PC やスマートフォンに加えて，センサや情報家電によるホームネットワークシステムや，ヘルスケアデバイスや眼鏡型，時計型端末をはじめとしたウェアラブルデバイスなど，様々なデバイスが通信機能を搭載し，ネットワーク化され始めている．近い将来，多くのセンサやプロセッサが様々な物に埋め込まれて互いに通信を行う，いわゆる IoT: Internet of Thing [1] が実現されていくと考えられるが，そのような通信環境におけるセキュリティが大きな課題となっている．

本研究では，様々な異種ネットワークに属するデバイスどうしがネットワークを構築し相互通信する仕組みとして非構造化 P2P/オーバレイネットワークに着目し，そのセキュリティ方式について検討した．ホームネットワークやセンサネットワークなど，様々なデバイスが GW 装置などを介して接続される環境において，新たにセキュアマルチホップセッションという概念を考案し，P2P/オーバレイネットワークのマルチホップ通信上で多段階認証および暗号化通信を実現する方式を提案する．提案方式を PUCC (P2P Universal Computing Consortium) [2] で規定されている XML ベースの P2P/オーバレイネットワーキングプロトコルである PUCC プロトコル上で設計および実装し，その性能評価を行い，提案方式の有効性を確認した．

本論文の構成は以下のとおりである．2 章において P2P/オーバレイネットワークの分類について述べ，3 章においてそのセキュリティ要件について整理する．4 章において本論文の提案方式について述べる．5 章において PUCC プロトコルの概要を説明し，6 章において PUCC プロトコル上での提案方式の設計および実装について述べる．7 章で性能評価，8 章で関連研究について述べ，9 章でまとめを行う．

2. P2P/オーバレイネットワークの分類

本章では本研究で対象とする非構造化 P2P/オーバレイネットワークについて述べる．P2P/オーバレイネットワークはアーキテクチャ的に分類すると，ルーティングや認証を担う中央サーバが存在するハイブリッド P2P 型，中央サーバを持たず個々のノードが対等にやりとりを行うピュア P2P 型があり，さらにピュア P2P 型では，DHT などをベースとした構造化ネットワークを構築するもの，ネットワークに特定の構造を持たずフラッディングなどを用いて目的のノードを探索する非構造化ネットワークを構築するものに大別される．P2P/オーバレイネットワークの分類と代表的な事例を以下に示す．

(1) ピュア P2P 型ネットワーク
  (a) 構造化ネットワーク：分散リソース共有 (CHORD [3]，CAN [4]，Pastry [5])
  (b) 非構造化ネットワーク：分散リソース共有 (Freenet [6])，ファイル共有 (Gnutella [7])，汎用プラットフォーム (JXTA [8])，ホームネットワーク [9]，センサネットワーク [10]

(2) ハイブリッド P2P 型ネットワーク
  (a) 分散リソース共有 (インスタントメッセージ・グループウェア)，アプリケーションレイヤマルチキャスト [11], [12]

構造化ピュア P2P 型ネットワーク，ハイブリッド P2P 型ネットワークでは，そのネットワークの構築に各ノード間のエンドツーエンド接続性が必要であり，インターネットのような IP ネットワーク上ですべてのデバイス間が接続可能な状態が前提となる．ピュア P2P 型のうち非構造化ネットワークでは，デバイスどうしが IP ネットワーク上で直接的に接続できない環境においても，中間ノードを介して接続可能なデバイスどうしが互いに探索し，アドホック的に通信経路を確立することができる．下位トランスポートプロトコルによるルーティングをアプリケーション層で吸収し，各中間ノードがバケツリレー式にデータを中継することで，IP ネットワークだけでなく非 IP ネットワーク含めた異種ネットワーク間の通信も実現可能である．

本研究では非構造化 P2P/オーバレイネットワークに着目し，その適用先の 1 つとして，**図 1** に示すように，スマートフォンなどを利用した遠隔からの家電制御や家電メーカからの家電の遠隔診断などを想定し，そのセキュリティ方式について検討した．本適用例では外部ネットワークから HGW を介しマルチホップで各家電デバイスにアクセスする．たとえば外部のスマートフォンから宅内の情報家電にアクセスする際には，スマートフォンから HGW を介して情報家電の認証を行い，マルチホップ通信によりセキュアな遠隔制御を行う．家電メーカからの家庭内の家電の遠隔診断を行う場合も，メーカ側の診断サーバから HGW を介して家電の認証を行い，マルチホップ通信によりセキュア遠隔診断を行う．マルチホップ通信であることから，双方向でやりとりする場合でも HGW を介することにより，宅内の機器へのグローバル IP の付与は必要なく，HGW で

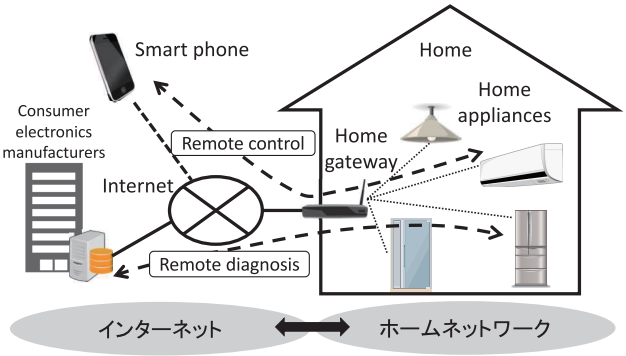

図 1 ホームネットワーク
Fig. 1 Home network.

不正なアクセスを検知することにより，セキュリティ上のリスクを軽減することができる．また，非構造化 P2P/オーバレイネットワークを利用することにより，IP ネットワークだけでなく，非 IP ネットワーク（NFC，Bluetooth，ZigBee など）も含めた異種ネットワーク間の通信も実現可能となる．

3. セキュリティ要求条件

3.1 前提条件

本研究ではホームネットワークや，ビルや工場などの屋内のセンサネットワークへの適用を前提として，**図 2** に示すようなネットワーク構成を想定する．GW 装置（HGW やセンサ GW など）に相当する中間ノードを介して，2〜3 ホップ程度の異種ネットワークが接続される階層型ネットワークで，GW 装置配下のネットワークは，多くても数百ノード以下のあらかじめ把握されている機器（情報家電やセンサデバイス）で構成された比較的信頼のある環境を想定する．インターネットなどのグローバルネットワーク側からは事前に取り決められた機器（家電メーカのサーバやユーザ自身のスマートフォンなど）からのリモートアクセスだけを許容するが，グローバルネットワークであることから不正なノードが接続してくる可能性がある．この前提条件をもとに，セキュリティの要求条件を整理した．

3.2 要求条件

(1) 経路情報を含めたセッション管理

情報家電やセンサデバイスは，他のデバイスと異なり性能が比較的低かったり，ソフトウェアの更新などが頻繁に行えなかったりする場合が想定されるため，HGW やセンサ GW 経由の通信のみを信頼するなど安全面での考慮が必要である．通信経路の中継デバイスの認証などを含めて多段階で認証を行う仕組みの導入が要求される．

(2) 認証局を利用した電子証明書（PKI）に依存しない認証方式

各家庭などに設置する情報家電や多数のセンサデバイスへの電子証明書の登録や再配布は運用効率面，コスト的面から困難である．また，非 IP ネットワークのような通信環境においては，認証局にアクセスすることができず，電子証明書の正当性が確認できない場合がある．そのため，認証や暗号化には認証局を利用した電子証明書（PKI）を利用しない方式が要求される．

(3) 双方向認証

P2P/オーバレイネットワークにおいては，各デバイスは対等であることから双方向認証を基本とし，リプレイアタック防止のため通信を開始するデバイス側から認証できることが必要である．

(4) 暗号化および改ざん防止

P2P/オーバレイネットワークでは，各中間ノードがバケツリレー式に通信を中継するため，メッセージのペイロード部は暗号化が可能だが，ルーティング情報は暗号化できない．したがって，暗号化されたペイロード部を含めメッセージ全体の改ざん防止のためにメッセージ認証が必要である．

(5) 通信性能への影響低減

本研究で対象とするデバイスは，情報家電やセンサデバイスなど，サーバや PC と異なり性能が比較的低いため，通信性能への影響を最小限に抑える必要がある．

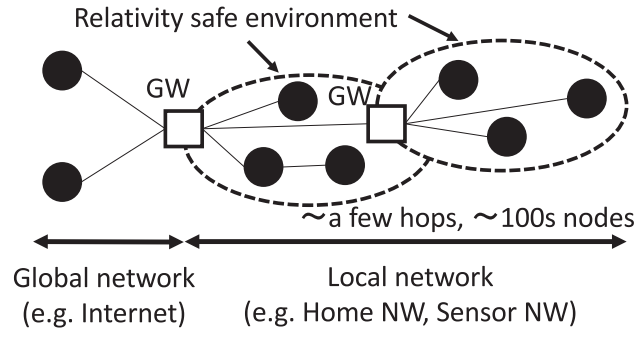

図 2 想定するネットワーク環境
Fig. 2 Assumed network environment.

4. 提案方式

3 章で述べたセキュリティ要求条件に基づき，本研究で提案する認証，暗号化，メッセージ認証方式について述べる．

4.1 セキュアマルチホップセッション

3.2 節のセキュリティ要求条件 (1) に基づき，メッセージの通信経路を含む，マルチホップセキュアセッションの概念を提案する．**図 3** においてノード A からノード B に対してセキュアセッションを確立する場合，まず中間ノードに対してセキュアセッションを確立する（AC 間）．次に，ノード A からノード B へ経路 R_i（A → C → B）単位でマルチホップセキュアセッションを確立する．送受信対象が同一であっても，別の経路 R_j（A → D → B）を用いた通信を行う場合は，再度，経路 R_j でのセッション確立が必要となる．これにより，特定の中間ノードで認証されたアクセスのみ認証を許可するなど，ネットワーク上での多段階認証を実現することが可能となる．

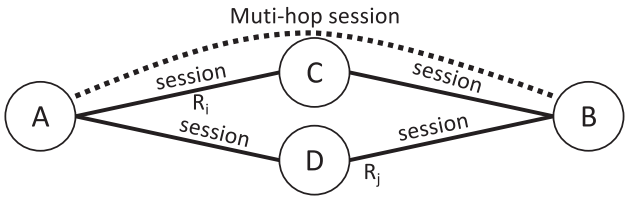

図 3 マルチホップセッションの概念
Fig. 3 Multi-hop session.

4.2 PSK（Pre-Shared Key）によるパスワード認証

3.2 節のセキュリティ要求条件 (2) に基づき，事前に設定された PSK（Pre-Shared Key）およびチャレンジレスポンス方式を利用した認証方式を用いる．本研究で想定するネットワーク環境では，ホームネットワーク環境など，多くても 100 台程度の機器であることから，市販の無線 LAN ルータのようにあらかじめ機器などに PSK を設定しておき，エンドユーザにより GW 装置へ設定するなど，マニュアル設定が可能な PSK を用いることが現実解であると考える．PSK はノード ID 単位，またはユーザ ID 単位で設定するものとし，総当たり攻撃ができないように十分長いものを利用することを前提とする．後述するように，本研究では 128 bit 以上の AES [13] や 256 bit 以上のハッシュ関数を用いていることから，ランダム生成した ASCII 文字（英字の大文字，小文字，数字および記号 14 字）で 21 文字以上 [14] であれば等価安全性 [15] として 128 bit 安全性が担保できる．

4.3 マルチホップ認証プロトコル

3.2 節のセキュリティ要求条件 (3) に基づき，チャレンジレスポンス方式を応用したマルチホップセキュアセッションを確立する認証プロトコルを提案する．本プロトコルでは，マルチホップで接続されているノードどうしが認証を行う場合，通信開始側と中間ノードの間で認証を行わないと，通信開始側から送信される認証要求を，中間ノードが通信開始を受ける側に転送を行わない．**図 4** に認証シーケンスを示す．事前に中間ノードとの間（AC 間）で下記と同様の手順で，セキュアセッションが構築されていることとし，マルチホップ経路 R_i（A → C → B）上で，下記の手順でチャレンジレスポンス認証を行う．

(1) 通信開始側（A）から中間ノード（C）を経由して通信開始を受ける側（B）に ID_A（ノード ID またはノード ID ＋ユーザ ID）および経路情報 R_i を含む通信要求を送信する．中間ノード（C）は通常のチャレンジレスポンス方式と異なりセキュアセッションが確立された相手からの認証要求しか転送しない．

(2) B が擬似乱数生成器によりランダム値（CV_i）を生成し，ID_B（ノード ID またはノード ID ＋ユーザ ID）および経路情報 R_i を含む応答を返却する．

(3) A は受信したチャレンジと事前に設定される PSK_A からハッシュ値 H(PSK_A, CV_i) を生成し B に返却する．あわせてチャレンジ値 CV_j を B に通知する．

(4) B は受信したハッシュ値 H(PSK_A, CV_i) と自身で生成したハッシュ値 H(PSK_B, CV_i) を比較し，一致した場合 A は認証される．一致しない場合，B は認証されない．B は A に認証結果を返却する．あわせて B は同様にチャレンジ値 CV_j を生成して A に通知する．

(5) A は受信したハッシュ値 H(PSK_B, CV_j) と自身で生成したハッシュ値 H(PSK_A, CV_j) を比較し，一致した場合は認証成功とし，双方向認証セッションが確立する．認証に失敗した場合は B に結果を通知し，通信は切断される．

以上で，マルチホップ経路間の双方向認証が完了する．

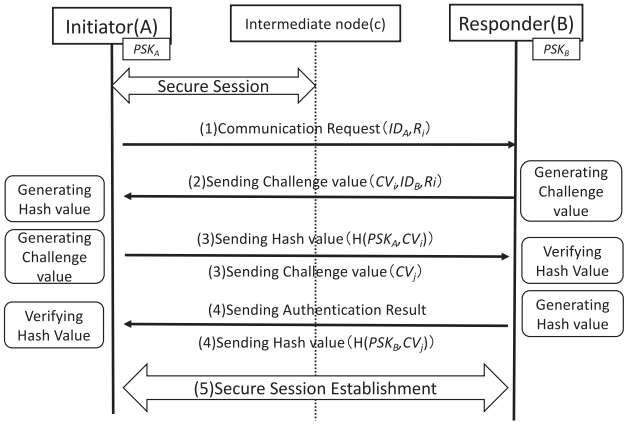

図 4 認証シーケンス
Fig. 4 Authentication sequence.

4.4 暗号化およびメッセージ認証

前章のセキュリティ要求条件 (4) に基づき，暗号化およびメッセージ認証の検討を行った．暗号化については，事前に設定および共有されている PSK から，チャレンジ値 CV_j をソルト値としてパスワードベース暗号 [16] を利用して共通鍵 CK を生成する方式を提案する．**図 5** に共通鍵生成の手順を示す．鍵生成はパスワードベース暗号における PBKDF2 などの鍵生成機能を用いる．認証フェーズにおいて受信側が生成したチャレンジ値 CV_j をソルト値とし事前に決定された繰返し回数により PSK を生成することで，相互認証されたノード間で共通鍵 CK が生成できる．実際のデータの暗号化，復号化にはセッション鍵 SK_i を用いる．セッション鍵は擬似乱数生成器により双方のノードでそれぞれ生成する．**図 6** に示すようにセッション鍵は共通鍵 CK により暗号化され，セッション鍵で暗号化されたデータとともに送信する．セッション鍵は認証フェーズで決定された回数使用されると更新されるようにする．さらに，メッセージ全体の完全性と送信者の正当性を担保す

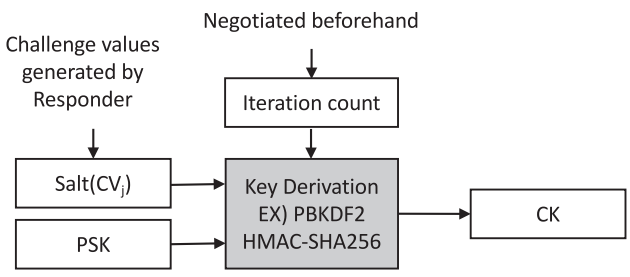

図 5 共通鍵生成手順
Fig. 5 Common key generation method.

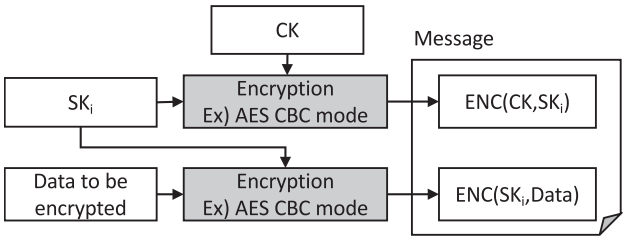

図 6 暗号化通信手順
Fig. 6 Encrypted communication method.

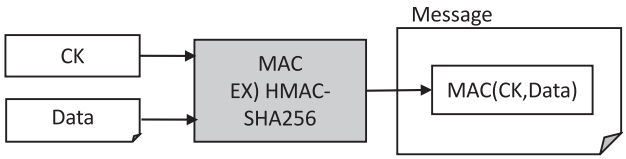

図 7 メッセージ認証方式
Fig. 7 Message authentication method.

るため，**図 7** に示すように，共通鍵 CK を用いてメッセージ認証符号 MAC($CK, Data$) を生成し，メッセージ認証を行うこととした．

5. PUCC プロトコルの概要

本章では，提案方式の有効性検証のために，本方式の適用先とした PUCC プロトコルの概要について述べる．PUCC プロトコルは PUCC（P2P Universal Computing Consortium）[2] において策定されている P2P/オーバレイネットワーキングプロトコルである．PUCC では，PC だけでなく，携帯電話や情報家電などの様々なデバイスをターゲットとして，異種ネットワーク間を経由した通信を可能とする P2P/オーバレイネットワーキングプロトコルを規定している．以下に，PUCC プロトコルの特徴について述べる．

5.1 PUCC プロトコルの特徴

PUCC プロトコルは，非構造化 P2P/オーバレイネットワークプロトコルであり，下位層に依存しない ID 体系，ルーティング機構を持ち，下位トランスポートプロトコルとして TCP/IP，Bluetooth，IEEE1394，HTTP などに対応しており，様々なネットワーク間を経由したシームレスな相互通信を実現している．さらにピュア P2P 型アーキテクチャにより，中央サーバを介さずに各ノードが直接通信を行い，ノードの自動探索や動的ルーティングを行う．さらに，各ノードの持つサービスの自動発見や遠隔制御の機能を持ち，デバイス間の動的なサービス発見と実行を可能としている．PUCC では，プロトコルとデバイスメタデータを XML により記述している．XML は，汎用的なツリー構造データを扱う仕組みであり，名前空間を用いてスキーマを用意することで，レイヤ構成のプロトコル設計が可能であり，複数のアプリケーションプロトコルを追加するなど，プロトコルの拡張が容易である．

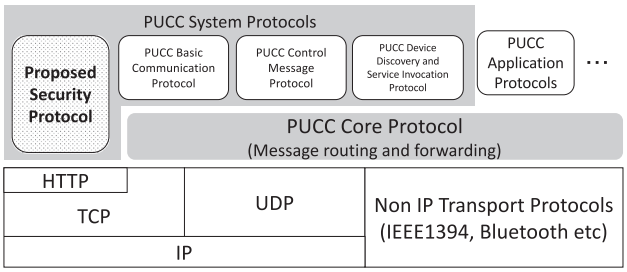

図 8 PUCC プロトコルスタックの構成
Fig. 8 PUCC protocol stack.

5.2 PUCC プロトコルスタック

PUCC プロトコルスタック構成を**図 8** に示す．図のように 2 レイヤとなっており，PUCC Core Protocol は特定の下位トランスポートネットワークに依存しない PUCC メッセージの送受信および転送など，ノード ID に基づくオーバレイルーティング機能を有し，様々な下位トランスポート層とのバインディングを可能としている．PUCC Core Protocol の上位の PUCC システムプロトコルでは，ノードの探索やエラーの通知，デバイス探索などの，PUCC アプリケーションに共通な基本的な通信機能を提供するプロトコルを規定している．PUCC Basic Communication Protocol は，ノード間のセッションの構築やリソース情報の交換を行う．PUCC Control Message Protocol は隣接ノードの探索やエラーメッセージの転送制御を行う．PUCC Device Discovery and Service Invocation Protocol は特定の機能を持つノードの発見とそのノードの遠隔機能実行などの制御を行う．また，ストリーミングやプリンティングサービスなどの PUCC アプリケーション向け機能については，個別に PUCC アプリケーションプロトコルとして拡張定義できるようになっている．

本論文で提案するプロトコルは，PUCC アプリケーションに共通に提供できる基本的な通信機能を提供するため，PUCC システムプロトコルとして設計および実装を行った．

5.3 PUCC セッション

PUCC プロトコルでは，隣接ノード間で仮想的な通信リンクを構築し，目的のノードへの通信は各中間ノードがバケツリレー式に中継を行っている（**図 9**）．最初に，PUCC Control Message Protocol の Lookfor メッセージにより PUCC ノードを発見するとそれを隣接ノードとして登録し，最初のメッセージ転送先として使用する．また，経路上の目的ノードを探索した後，各ノード間では PUCC Basic Communication Protocol の Hello メッセージおよび Bye メッセージにより，複数の中間ノードを経由したマルチホップセッションの確立，破棄が可能となっており，払い出されるセッション ID により中間ノードがメッセージ

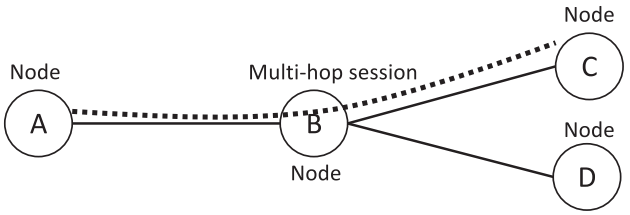

図 9 PUCC セッション
Fig. 9 PUCC session.

の転送ルートを一意に識別することができる．セッションはノード単位またはノード上のユーザ単位で構築が可能になっている．

本研究では PUCC のセッションの概念を用いて，セキュアマルチホップセッションを確立するプロトコルの設計を行った．

6. プロトコル設計

本章では，4 章で述べた提案方式の PUCC プロトコル上での設計方針について述べる．既存プロトコル機能への大きな影響がなく，提案方式の機能提供ができるように，図 8 に示すように PUCC システムプロトコルの 1 つとして，PUCC Security Protocol を設計した．上位レイヤでは PUCC Basic Communication Protocol と同様にセッション構築機能を提供し，同時に認証を行うことでセキュアなセッション構築を行うことができる．また，下位レイヤでは，上位レイヤでの認証結果に基づき，PUCC Core Protocol へ暗号化およびメッセージ認証機能を提供できるようにした．提案プロトコルによるセキュアセッション構築後は，他のシステムプロトコルおよびアプリケーションプロトコルは従来どおり PUCC Core Protocol 上においてセキュア通信が可能となる．

6.1 認証シーケンス

図 10 に示すようにチャレンジレスポンス認証を PUCC Basic Communication Protocol のセッション確立シーケンスの拡張方式として実装した．事前に AC 間では認証済みであることとする．まず，通信開始側（A）から Node ID，User ID および Session ID を含む認証要求（Hello メッセージ）を中間ノード（C）を経由して送信する．中間ノード（C）はすでに AC 間で認証済みであることから，その通信開始を受ける側（B）に転送する．次に通信開始を受ける側（B）は，認証処理継続のフラグ（Continue），チャレンジ，Session ID および要求ハッシュアルゴリズムを含む認証応答（HelloResponse メッセージ）を送信する．次に，通信開始側（A）は，チャレンジに対するレスポンス，通信開始を受ける側の認証のためのチャレンジ，Session ID および要求ハッシュアルゴリズムを含む認証要求（2 回目の Hello メッセージ）を送信する．この際，暗号化やメッセージ認証に必要なパラメータ値（ハッシュアルゴリズム，暗号アルゴリズムなど）もあわせて通知する．最後に，通信開始を受ける側（B）は，認証結果，チャレンジに対するレスポンス，Session ID を含む認証応答（2 回目の HelloResponse メッセージ）を送信する．この際，暗号化やメッセージ認証に必要なパラメータのうちサポートしているものをあわせて通知する．認証結果には成功であれば Success，失敗であれば NotAuthencated を設定する．通信開始を受ける側（B）も認証成功であれば，セキュア通信が開始される．通信開始を受ける側（B）の認証が失敗すると，通信開始側から切断通知（Bye）が送信されセッションは破棄される．**図 11** に，XML で記述された 2 回目の Hello メッセージの例を示す．

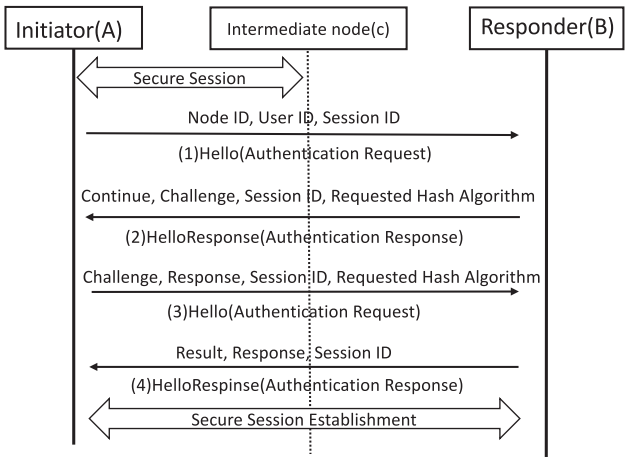

図 10 認証シーケンス
Fig. 10 Authentication sequence.

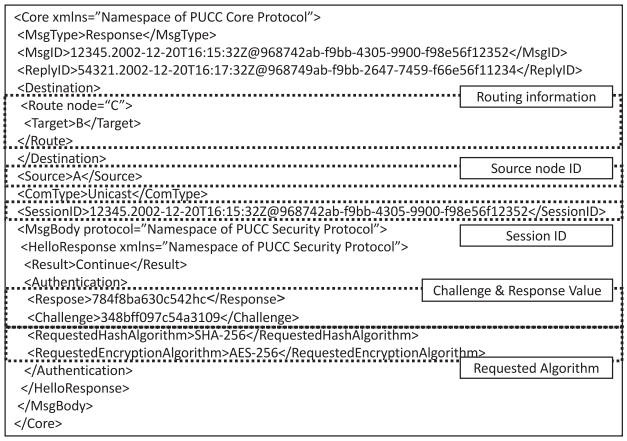

図 11 Hello メッセージの例
Fig. 11 Example of Hello message.

6.2 暗号化通信

共通鍵生成は PKCS #5 [16] に従って設計した．事前に共有されている PSK を用い，セキュアマルチポップセッション構築時に受信側が送信したチャレンジ値をソルトとして鍵生成アルゴリズム PBKDF2 を用いて鍵生成を行

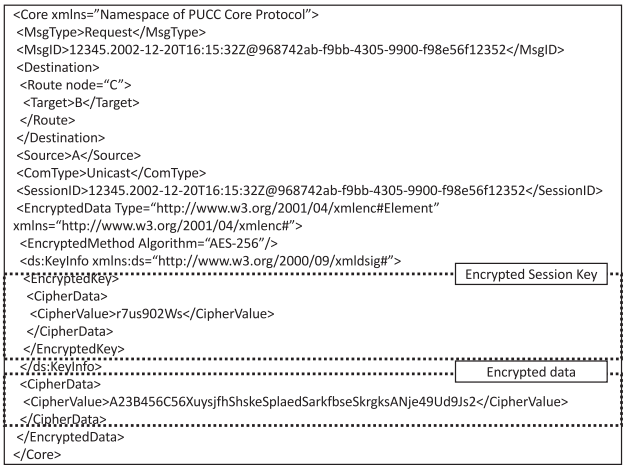

図 12 暗号化メッセージの例
Fig. 12 Example of encrypted message.

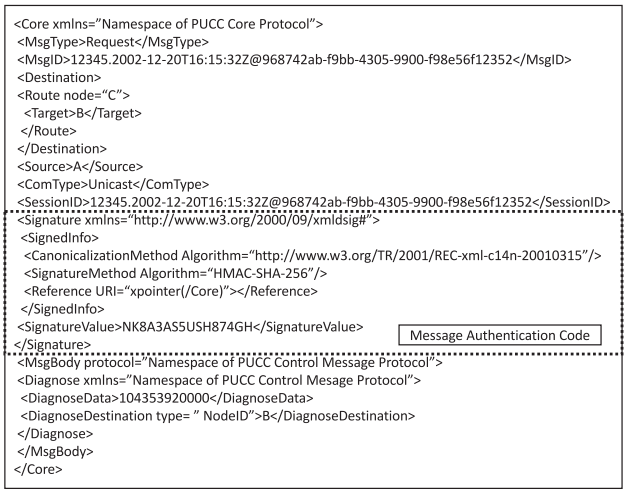

図 13 署名メッセージ例
Fig. 13 Example of signature.

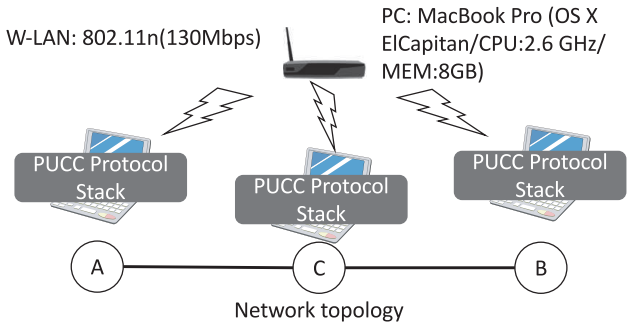

図 14 評価環境
Fig. 14 Evaluation environment.

う．繰返し回数は 10,000 回とした．暗号化アルゴリズムとしては AES [13] あるいは Camellia [17] の 128 bit 以上のブロック暗号化アルゴリズムを用いる．ブロック暗号化モードは CBC（Cipher Block Chaining）を用い，IV（Initial Vector）は各暗号時に更新する．また，最終暗号化ブロックのパディングは PKCS #7 [18] に従う．データの暗号化は，W3C XML Encryption Scheme [19] に準拠して実行する．すべての暗号化要素は EncryptedData 要素内に設定される．それぞれ EncryptionMethod 要素には使用する暗号化アルゴリズム，ds:KeyInfo 要素内の Encryptedkey 要素の配下の CiperData 要素内の CipherValue 要素に暗号化に使用したセッション鍵を設定する．さらに暗号化されたデータは CiperData 要素内の CiperValue 要素に設定される．

図 12 に暗号化メッセージの例を示す．例では MSGBody 要素が暗号化され，EncryptedData 要素に置き換わっている．EncryptedData 要素内には共通鍵により暗号化されたセッション鍵とセッション鍵により暗号化されたペイロードが設定される．

6.3 メッセージ認証

PUCC プロトコルのメッセージは，中間ノードが参照する必要があるルーティング情報要素（Destination，TraceRoute，HopCount など）は平文で指定される．そのため，メッセージ認証は，暗号データ要素も含めたメッセージ全体に適用する．メッセージ認証対象の要素は XML Signature Scheme [20] に従って正規化される．メッセージへの署名はメッセージ全体に適用され，かつ検証もメッセージごとに行うことから Enveloped 署名を用いる．署名要素は Signature 要素内に設定される．SignedInfo 要素内には，正規化アルゴリズムを指定する CanonicalizationMethod 要素，署名アルゴリズムを指定する SignatureMethod 要素，署名の適用先を指定する Reference 要素がある．また，SignatureValue 要素にメッセージ認証コードを Base64 エンコーディングしたものを設定する．**図 13** にメッセージ認証コードによる署名後のメッセージ例を示す．この例ではメッセージ全体に対して署名が適用され，Signature 要素はメッセージの一部として Core 要素配下に設定されている．SignatureValue 要素にメッセージ認証コードが設定される．

7. 実装および評価

前章で述べた提案方式に基づき PUCC プロトコル上で設計したセキュリティプロトコルを実装し，3.2 節で述べたセキュリティ要求条件 (5) に基づき提案方式による通信シーケンス実行時間への影響についての定量評価を行った．評価環境は**図 14** に示すように，3 台の MacBook Pro（CPU: 2.6 GHz/MEM: 8 GB）にインストールした Java（JDK1.8）で実装された PUCC プロトコルスタック上に提案方式のプロトコルを実装し，各 PC は無線 LAN（802.11n）インフラストラクチャモードで接続した．またアルゴリズムや各種パラメータ設定については，等価安全性で 128 bit 安全性を担保できるよう**表 1** に示すとおりとした．

以上の環境において，提案方式の PUCC プロトコルへ

表 1 使用アルゴリズムおよび各種パラメータ設定
Table 1 Algorithm and parameter setting.

Hash Algorithm	HMAC-SHA-256
PSK length	128 bit 相当（ASCII 21 文字以上）
Key Derivation	PBKDF2/HMAC-SHA-256
Iteration count	10,000
Salt length	128 bit
Encryption Algorithm	AES-128/CBC mode

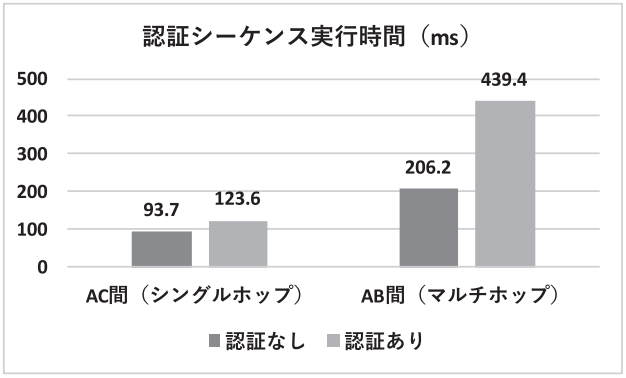

図 15 認証シーケンス実行時間
Fig. 15 Authentication sequence time.

の影響を測定した．測定は 10 回行い，その平均値を評価に用いた．

7.1 認証シーケンス実行時間

提案方式による認証の有無によるシーケンス実行時間の定量評価を行った．

図 15 に示すように，認証を行うことで通信開始までの処理時間がシングルホップ認証において約 30 ms 増となった．これは，PUCC プロトコルにおいて通信を開始する際には，Hello メソッドによる 1 回のリクエスト/レスポンスにより通信経路を確立するが，提案方式での認証を行うことで，2 回のリクエスト/レスポンスが実行されているためである．さらにマルチホップ（2 段階）認証においては約 230 ms の増となった，これは認証シーケンスに加え AC 間のセキュアセッションでの暗号・復号化処理時間（計 4 回）が加算されるためである．2 ホップのネットワークにおいて，認証の有無により処理時間が倍となっているが，これはセッション構築時のみに影響するものである．セッションの維持期間を十分に長くすることで通信全体への影響を小さくすることが可能である．

7.2 メッセージ送受信処理時間

次に，暗号化およびメッセージ認証（MAC）の有無によるメッセージ送受信時間への影響について比較を行った．メッセージサイズは 1 Kbyte と固定して送信側の送信処理時間，受信側の受信処理時間の測定を行った．**図 16** に示すように暗号化および MAC なしの場合は，送受信処理とも に 1 ms 程度となっているが，暗号化ありの場合は，送受信ともに 20 ms 程度の処理時間がかかっている．また，MAC の有無でも比較を行ったが，ほとんど影響は見られなかった．これにより，提案方式の暗号化処理は，メッセージの送受信時間に影響があるが，MAC は大きな影響がないことが分かった．

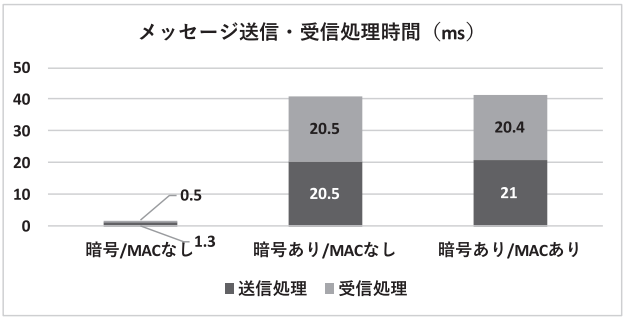

図 16 メッセージ送受信処理時間
Fig. 16 Message sending and receiving processing time.

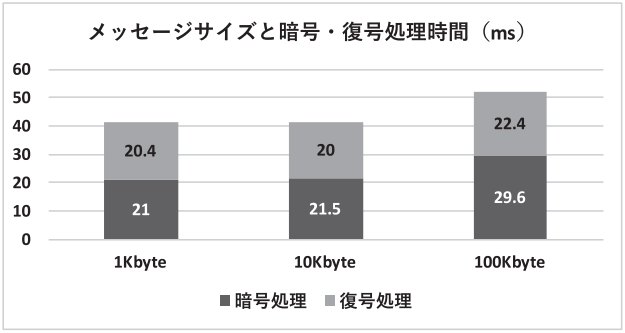

図 17 メッセージサイズと送受信時間比較
Fig. 17 Relation between message size and processing time.

7.3 メッセージサイズによる影響

暗号化およびメッセージ認証（MAC）のメッセージサイズによるメッセージ送受信時間への影響について評価を行った．**図 17** に示すように，各メッセージサイズにおいて暗号・復号処理時間はともに大きな増加はなく 20 ms 程度であり，100 Kbyte 程度まではメッセージサイズによる送受信時間への影響はそれほど大きくないことが分かった．

7.4 デバイス性能による影響

提案方式のデバイス性能による影響を確認するため，OpenSSL コマンドの speed オプションにより評価で用いた MacBook Pro および Raspberry Pi 3 Model B（Raspbian 8.0/CPU: 1.2 GHz/MEM: 1 GB）の AES 128 bit CBC モードでの暗号化処理性能の比較を行った．**図 18** に示すように，性能測定に用いた MacBook Pro よりも Raspberry Pi 3 では暗号化処理に 1.60〜1.81 倍の時間がかかっている．提案方式では通信時間に対して暗号化処理時間が支配的であることから，Raspberry Pi などの小型デバイスでは

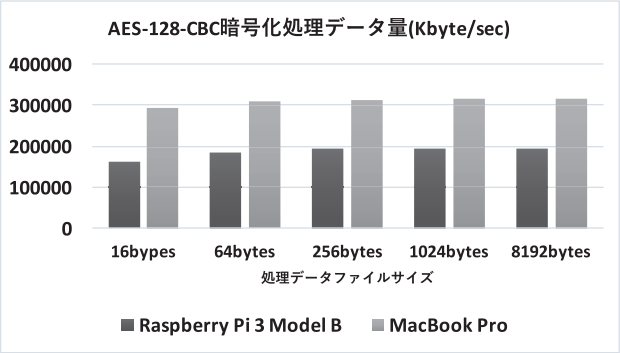

図 18 デバイスによる AES 暗号化処理性能の比較
Fig. 18 Encryption performance comparison with Macbook with Raspberry Pi.

2 ホップでの認証において約 400 ms（230 ms × 1.7 倍）の処理時間増，暗号化通信において送受信それぞれ約 36 ms（21 ms × 1.7 倍）の処理時間増となることが推測される．

以上の測定結果により，提案方式による通信シーケンス実行時間への影響についてまとめる．メッセージ認証については送受信時間に大きな影響はないことが分かった．またマルチホップ認証や暗号化については送受信時間への影響が比較的大きく，Raspberry Pi などの小型デバイスではその影響がさらに大きくなることが分かった．しかし，近年では，AES による暗号化・復号化処理の高速化のため AES-NI（Advanced Encryption Standard New Instructions）[21] のようなハードウェアアクセラレーション技術も普及し始めており，大幅な性能増も見込める [22]．将来的には AES による暗号化・復号化処理の影響は実用レベルに低減され，提案方式は家電機器やセンサデバイスなど小型デバイスにおいても十分実用的に動作すると考える．

7.5 提案プロトコルの安全性

提案プロトコルの定性評価として，想定されるリスクの洗い出しと対策について検討した．提案方式では事前に設定された PSK によりチャレンジレスポンス方式を用いてマルチホップ認証を行い暗号化通信するため，基本的にはセキュアセッションが構築された経路に不正なノードが侵入することはない．ただし，PSK を用いた認証であること，P2P/オーバレイネットワークであることから，下記に列挙するセキュリティリスクが存在する．

(1) PSK の物理的な漏洩

PSK がユーザにより管理されたり，市販の無線 LAN ルータの WPA キーのように，あらかじめ製造段階で製品に設定しておいたりする場合において物理的な漏洩が考えられる．しかし提案方式では，家電機器の PSK が漏洩した場合でも，GW 装置での認証の段階でブロックするなど，多段階認証による対策が可能である．

さらに，認証局などの第三者による身分証明を行わないことから下記のようなリスク [23] が存在する．

(2) オンライン攻撃による PSK の類推

ユーザが設定する PSK の長さやランダム性が十分でない場合，その PSK を類推して不正アクセスが行われる可能性がある．ブルートフォース攻撃やパスワードリスト攻撃に関しては，一般的なパスワードクラック対策と同様に，通信開始を受ける側で認証試行回数に上限を設けるなどの対策が必要である．

(3) オフライン攻撃による PSK の類推

特定のチャレンジ値を用いたレインボーテーブルによる攻撃 [24] も考えられる．通信開始側からの攻撃に関しては，先に通信開始側を認証することから，認証前にハッシュ値の取得ができず攻撃は不可能である．今回のユースケースでは可能性は低いが，事前に特定のチャレンジ値によるテーブルを用意することで，通信開始を受ける側からの攻撃が可能である．そのため，通信開始側において接続先（通信開始を受ける側）からの不正なチャレンジ値（明らかに短いものや専用のレインボーテーブルが存在することが明確なものなど）を検知する対策が必要である．

また，P2P/オーバレイネットワークの観点では，下記のような可用性リスクが存在する．

(4) 経路上の悪意のあるノードによる攻撃

提案方式ではセキュアセッション構築段階において，その経路の途中で認証応答を返さないなど悪意のある中間ノードが存在する可能性がある．そのような場合は別の経路を経由して認証を行うなどプロトコル実行上での対策が必要となる．

以上のように，PSK を用いた認証であること，P2P/オーバレイネットワークであることから，プロトコルの実用段階においては，上記で述べたセキュリティリスクへの対策が必要である．

8. 関連研究

オーバレイネットワーク上でのセキュリティ方式の研究としては，公開鍵を利用した方式とワンタイムパスワード方式などがある．公開鍵を利用した方式 [25] ではノード間の信頼の輪と分散ハッシュにより公開鍵を分散管理する仕組みにより認証を実現している．ワンタイムパスワード方式 [26] では，ハッシュ関数の 1 方向性に着目し，ハッシュ関数の適用回数の同期をとることで認証を行う．公開鍵を利用した方式においては公開鍵配送方式の効率化の観点，ワンタイムパスワード方式では認証方式のネットワーク耐障害性と運用コストの観点での研究であり，関連研究ではあるが本研究とは性能評価の観点が異なる．そのため，各研究で前提としている認証方式に着目してその定性的な比較を行った．比較結果を**表 2** に示す．

運用コストの観点では，公開鍵を利用した方式では，公開鍵を生成する煩雑さがあるが，各ノードで自動生成する

表 2 関連研究との比較
Table 2 Comparison with related works.

	公開鍵を利用した方式 [25]	ワンタイムパスワード方式 [26]	提案方式（PSK+マルチホップセッション）
運用コスト	○	△※同期が外れた場合	○
不正ノードへの耐性	×※信頼の輪に不正ノードが含まれる場合	△※半数以上が不正ノードの場合	○
等価安全性	○	○	○

ことにより回避可能である．また，ワンタイムパスワード方式としては，各ノード間でそれぞれハッシュの世代管理を行う必要があり，ハッシュの同期が外れてしまうと再設定が必要となるなどの問題点がある．提案方式では，PSKをもとに暗号鍵などを生成するためエンドユーザによる設定も可能であり，またワンタイムパスワードのように同期が外れてしまうという問題は発生しない．

不正ノードへの耐性の観点では，公開鍵を利用した方式では信頼の輪をベースとした認証となるため，信頼の輪の途中に不正ノードが含まれると信頼性が保証できない．ワンタイムパスワードを利用した方式については，最終的に多数決により認証の判断が行われるため，多数の不正ノードがネットワーク上に存在すると信頼性が保証できなくなるという問題がある．提案方式では，事前に設定されたPSK が漏洩しない限り，信頼性が保証される．

等価安全性の観点では，公開鍵を利用した方式，ワンタイムパスワード方式，提案方式ともに，十分に長い公開鍵およびハッシュ値を用いることで安全性が担保できる．提案方式ではPSK に十分に長い文字列（2017 年時点で NIST が推奨する 96 bit 安全性であると ASCII 文字 16 文字以上）が設定されていれば，十分な安全性が担保された通信が可能となる．また，マルチホップ通信上での多段階認証を利用することで，より安全性の高い通信を行うこともできる．

9. まとめと今後の課題

本論文では，非構造化 P2P/オーバレイネットワークにおけるセキュリティ方式として，新たにセキュアマルチホップセッションという概念を提案し，マルチホップ通信上での PSK（Pre Shared Key）を用いた多段階認証，暗号化，メッセージ認証を行う方式を提案した．暗号化においては，CA（認証局）を利用した電子証明書を使用できない環境を想定し，パスワードベース暗号を応用し，認証フェーズでやりとりするチャレンジ値をソルトとして暗号鍵を生成する方式を提案した．さらに PUCC プロトコル上でのプロトコル設計および実装し，その性能評価を行い，提案方式の有効性を確認した．以下に今後の課題を述べる．

(1) Self-certifying ID [27] の利用の検討

提案方式ではユーザの指定するパスワードをベースとした認証，暗号化方式の提案を行ったが，エンドユーザが各デバイスに十分に長いパスワードを設定することは容易ではない．また，ノード間で安全にパスワードを共有するためには，パスワードの生成・管理，配送の課題がある．Self-certifying ID により公開鍵のハッシュ値からノード ID を生成することにより，自動生成したノード ID による認証を行うことが可能となる．しかし，パスワードの配送などの問題は解決できるが，信頼モデルに課題があるため，その解決が必要である．

(2) アクセスコントロール

実際にデバイスのリモート制御を行う場合，ユーザによるアクセス権限を管理する必要がある．PUCC メタデータでは，デバイスメタデータにアクセスコントロールリスト（ACL）の定義が可能であり，ACL 方式と提案方式との連携について検討する予定である．

(3) 大規模ネットワークへの対応

本研究では，P2P/オーバレイネットワークの規模としては，ホームネットワークのように比較的小規模なものを想定して検討および評価を行ったが，原理上は大規模なネットワークにも適用可能である．今後，P2P ストリーミングのような，より大規模ネットワークに適用した場合の検討や評価も行う予定である．

参考文献

[1] Ashton, K.: That ‘Internet of Things’ Thing (online), available from ⟨http://www.rfidjournal.com/articles/view?4986⟩ (accessed 2017-11-08).

[2] Peer to Peer Universal Computing Consortium: PUCC Version 3 Specification (online), available from ⟨http://www.pucc.jp/pucc-version3-specification-download/index.html⟩ (accessed 2017-11-08).

[3] Stoica, I., Morris, R., Karger, D., Kaashoek, M.F. and Balakrishnan, H.: Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications, *Proc. 2001 SIGCOMM Conference*, pp.149–160, ACM (2001).

[4] Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Shenker, S.: A scalable content-addressable network, *Proc. 2001 SIGCOMM Conference*, pp.161–172, ACM (2001).

[5] Rowstron, A. and Druschel, P.: Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems, *Middleware ’01*, *Proc. IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, pp.329–350, ACM (2001).

[6] The Freenet Project Inc.: Freenet (online), available from ⟨http://freenet.sourceforge.net/⟩

(accessed 2017-11-08).
[7] Kirk, P.: Gnutella protocol development (online), available from ⟨http://rfc-gnutella.sourceforge.net⟩ (accessed 2017-11-08).
[8] Oracle: JXTA Technology (online), available from ⟨http://www.oracle.com/technetwork/java/index-jsp-136561.html⟩ (accessed 2017-11-08).
[9] Ishikawa, N.: PUCC Activities on Overlay Networking Protocols and Metadata for Controlling and Managing Home Networks and Appliances, *Proc. IEEE*, Vol.101, No.11, pp.2355–2366 (2013).
[10] Ogura, M., Mineno, H., Ishikawa, N., et al.: Automatic GUI Generation for Meta-data Based PUCC Sensor Gateway, *Knowledge-Based Intelligent Information and Engineering Systems*, *KES 2008*, *Lecture Notes in Computer Science*, Vol.5179, pp.159–166 (2008).
[11] Chu, Y.-H., Rao, S.G. and Zhang, H.: A Case For End System Multicast, *IEEE Journal on Selected Areas in Communications*, Vol.20, No.8, pp.1456–1471 (2002).
[12] Pendarakis, D., Shi, S., Verma, D. and Waldvogel, M.: ALMI: An Application Level Multicast Infrastructure, *USITS '01*, *Proc. 3rd Conference on USENIX Symposium on Internet Technologies and Systems*, pp.49–60, USENIX (2001).
[13] National Institute of Standards and Technology: Advanced Encryption Standard (online), available from ⟨https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf⟩ (accessed 2017-11-08).
[14] 齋藤孝道：マスタリング TCP/IP 情報セキュリティ編，オーム社 (2013).
[15] National Institute of Standards and Technology: Recommendation on Key Management (online), available from ⟨http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf⟩ (accessed 2017-11-08).
[16] Kaliski, B.: PKCS #5: Password-Based Cryptography Specification Version 2.0 (online), available from ⟨https://www.ietf.org/rfc/rfc2898.txt⟩ (accessed 2017-11-08).
[17] Aoki, K., Ichikawa, T., Kanda, M., et al.: Specification of Camellia – A 128-bit Block Cipher (online), available from ⟨http://info.isl.ntt.co.jp/crypt/camellia/dl/01espec.pdf⟩ (accessed 2017-11-08).
[18] Kaliski, B.: PKCS #7: Cryptographic Message Syntax Version 1.5 (online), available from ⟨https://tools.ietf.org/html/rfc2315⟩ (accessed 2017-11-08).
[19] Imamura, T., Dillaway, L., Simon, E., et al.: XML Encryption Syntax and Processing Version 1.1 (online), available from ⟨https://www.w3.org/TR/xmlenc-core1/⟩ (accessed 2017-11-08).
[20] Bartel, M., Boyer, J., Fox, B., et al.: XML Signature Syntax and Processing Version 2.0 (online), available from ⟨https://www.w3.org/TR/xmldsig-core2/⟩ (accessed 2017-11-08).
[21] Gael, H.: Introduction to Intel AES-NI and Intel Secure Key instructions (online), available from ⟨https://software.intel.com/en-us/node/256280⟩ (accessed 2017-11-08).
[22] Xu, L.: Securing the Enterprise with Intel AES-NI (online), available from ⟨https://www.intel.com/content/www/us/en/enterprise-security/enterprise-security-aes-ni-white-paper.html⟩ (accessed 2017-11-08).
[23] National Institute of Standards and Technology: Electronic Authentication Guideline (online), available from ⟨http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63-2.pdf⟩ (accessed 2017-11-08).
[24] Oechslin, P.: Making a Faster Cryptanalytic Time-Memory Trade-Off, *Advances in Cryptology – CRYPTO 2003*, *Lecture Notes in Computer Science*, Vol.2729, pp.617–630 (2003).
[25] 武田敦志，北形　元，松島　悠ほか：P2P ネットワークのための分散ハッシュ型認証手法，情報科学技術レターズ，Vol.6, pp.433–436 (2007).
[26] 西田雄治，辻　貴介，清水明宏：P2P 型ネットワークへのワンタイムパスワード認証方式の適用，信学技報，Vol.105, No.281, pp.23–27 (2006).
[27] Venkataramani, A., Kurose, J.F., Raychaudhuri, D., Nagaraja, K., Mao, M. and Banerjee, S.: MobilityFirst: A Mobility-Centric and Trustworthy Internet Architecture, *ACM SIGCOMM Computer Communication Review*, Vol.44, No.3, pp.74–80 (2014).

加藤 剛志　（学生会員）

1999 年慶應義塾大学理工学部電気工学科卒業，2001 年慶應義塾大学大学院理工学研究科修士課程修了．同年株式会社エヌ・ティ・ティ・ドコモ（現，NTT ドコモ）入社．モバイルインターネット，ユビキタスコンピューティング等の研究開発を経て，モバイルサービスの企画開発等に従事．駒澤大学大学院グローバル・メディア研究科博士後期課程在籍中．

石川 憲洋　（正会員）

1980 年京都大学大学院工学研究科修了．同年日本電信電話公社（現，NTT）入社．1999 年 NTT 移動通信網株式会社（現，NTT ドコモ）入社．2010 年 4 月より駒澤大学グローバル・メディア・スタディーズ学部教授．博士（情報学）．モバイルコンピューティング，ユビキタスコンピューティング等の研究に従事．情報処理学会業績賞等を受賞．電子情報通信学会，IEEE，ACM 各会員．本会シニア会員．

吉田 尚史 （正会員）

1996 年筑波大学第三学群情報学類卒業，1998 年筑波大学大学院修士課程理工学研究科修了．2001 年筑波大学大学院博士課程工学研究科（現在，システム情報工学研究科）修了．博士（工学）．2001～2006 年慶應義塾大学大学院政策・メディア研究科特別研究教員（専任講師）．2006～2014 年慶應義塾大学 SFC 研究所上席所員（訪問）．2006 年より駒澤大学グローバル・メディア・スタディーズ学部講師，同准教授を経て，2017 年より同教授．2010 年 5～9 月まで Tampere University of Technology，Pori，Finland 研究員．ACM，IEEE-CS，日本データベース学会，電子情報通信学会各会員．