Vol.2016-DPS-166 No.32
Vol.2016-CSEC-72 No.32
2016/3/4

非構造化オーバーレイネットワークにおける セキュリティ方式の提案と PUCC プロトコルへの適用

加藤剛志[†1]　石川憲洋[†2]　吉田尚史[†2]

概要：センサや情報家電によるホームネットワークシステムなど，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同士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オーバー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るなどの課題がある．本稿ではそれらの課題を解決する方式として，パスワードベースの認証によるセキュアなマルチホップセッション構築を可能とするセキュリティ方式の提案を行う．また，提案方式の有効性検証のため，PUCC（P2P Universal Computing Consortium）で規定されている XML ベースのプロトコル上での設計について述べる．

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

Security method for Unstructured Overlay network and its application to PUCC Protocols

TAKESHI KATO[†1]　NORIHIRO ISHIKAWA[†2]
NAOFUMI YOSHIDA[†2]

Abstract: In recent years, various devices such as sensor, home appliances which have communication capability are being networked. We are considering security scheme on an unstructured overlay network which has a function for networking on an ad-hoc basis among devices on different physical networks. In such environment, there are some security challenges such as difficulty redistributing or updating digital certificates of many distributed devices and establishing secure session on a specific communication path via a specific device. In this paper, we propose the security method which has password based authentication/encryption function and multi-hop secure session establishment function. We also describe protocol design on XML protocol defined by PUCC (P2P Universal Computing Consortium) for verifying the proposal method.

Keywords: Overlay network, Home network, Security, PUCC, IoT, M2M

1. はじめに

近年，無線 LAN や Bluetooth などの無線技術の発展に伴い，携帯電話が従来のセルラー通信インターフェイスに加えて，ローカル通信インターフェイスを持つなど，複数の異なる無線ネットワークインターフェイスを持つモバイル端末が増加している．また，プロセッサや無線技術の進歩により，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問題となっている．本研究ではそのような通信環境において，様々なネットワークに属するデバイス同士が目的に応じてネットワークを構築し協調動作する仕組みとして非構造化オーバーレイネットワークに着目した．そのセキュリティ要求条件から認証，暗号化方式の検討を行った．その有効性を検証するため，その方式を PUCC（P2P Universal Computing Consortium）[12]で規定されている XML ベースの PUCC プロトコル上での設計を行った．本稿では，まず対象とするオーバーレイネットワークの特徴を述べ,そのセキュリティ要件ついて整理し，それに基づき提案するセキュリティ方式について述べる．さらに PUCC プロトコルの概要について説明し，PUCC プロトコル上での提案方式の設計について述べる．

†1 駒澤大学大学院グローバル・メディア研究科
The Graduate School of Global Media, Komazawa University.
†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.

2. オーバーレイネットワークの分類

本研究では Bluetooth, ZigBee など異なるネットワークに属する複数のデバイス同士が目的に応じてネットワークを構築し協調動作する仕組みとして非構造化オーバーレイネットワークに着目した．オーバー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取りを行うピュア 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，DSR（Dynamic Source Routing）のようなフラッディング等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．オーバー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
・構造化ネットワーク：分散リソース共有（CHORD[2], CAN[3]，Pastry[4]）
・非構造化ネットワーク：分散リソース共有（Freenet[5]），ファイル共有（Gnutella[6]），汎用プラットフォーム（JXTA[7]），情報家電[8]，センサネットワーク[9]

② ハイブリッド P2P 型
・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ーマルチキャスト[10,11]

ハイブリット P2P 型やピュア P2P 型のうち構造化オーバーレイネットワークでは，そのネットワークの構築に IP ネットワークのエンドツーエンド接続性，即ちインターネット上での構築を前提としている．ピュア P2P 型のうち非構造化オーバーレイネットワークでは，接続可能なデバイスが互いに探索し，アドホック的に通信リンクを確立する．また，下位トランスポートプロトコルの差異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中継することで，異種ネットワーク間の通信も実現可能である．非 IP ネットワークを含む複数の異種ネットワークで構成された環境においては，非構造型オーバーレイネットワークが適していると考える．

3. セキュリティ要件

以下に，非構造化オーバーレイネットワークのセキュリティ要件について整理する．本研究では，非構造化オーバーレイネットワークをホームネットワークやセンサネットワークに適用することを想定し，下記のようにセキュリティ要件を整理した．

- デバイス及びユーザ単位での認証

各デバイス間，もしくは各デバイスを使用するユーザ単位で認証できるようにする．また，ID の付与を管理する中央サーバは想定せず，各ノードが GUID 等を用いて自身でグローバルに一意な ID を生成し，そのノード内でユーザ ID が一意となるように付与することにより，ユーザ ID＋ノード ID の単位でも認証可能とする．

- 双方向認証

ノード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開始側から認証されるようにする．

- 電子証明書に依存しない

家電やセンサに関しては，電子証明書の登録や再配布も運用効率的，コスト的に困難であり，またアドホックな通信環境においては必ずしも CA（認証局）にアクセスできず証明書の正当性が確認できないと想定し，認証や暗号化には CA（認証局）を利用した電子証明書を利用しない方式とする．

- 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を想定すると，GW 装置などの特定のデバイス経由のアクセスを信頼するなど通信経路の判断が重要である．そのため通信経路を含めたマルチホップセッション単位での認証，暗号化ができるようにする．

- メッセージ認証による改ざん防止

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するため，メッセージ内にメッセージの転送を担う中間ノードが参照する部分がある．従って，経路情報を含むメッセージ全体を暗号化せず，メッセージ全体の改ざん防止のためにメッセージ認証を導入する．

4. 提案方式

前節のセキュリティ要件に従い，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 認証

前節のセキュリティ要件より，GW 装置などの特定のデバイス経由のアクセスを信頼するなど通信経路の判断が重要となるため，途中のノードも含めた通信経路を含む，ノード間のマルチホップセッションの概念を導入する．図 1 のように特定の経路（A→C→B）を経由したノード間の通信に対して，認証，暗号化を実現することを目的とする．なお，マルチホップッセッションの概念は，シングルホップセッション（例えば，A→C）のケースも含む．また，別の経路（A→D→B）の経路を用いた通信に対しては，再度，セキュアなマルチホップセッションの構築が必要となる．

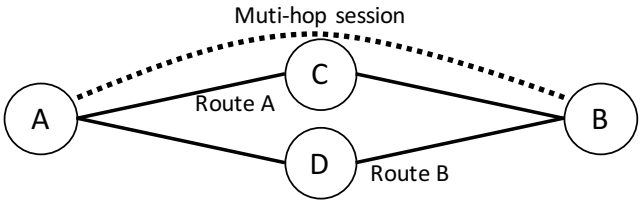

図 1 マルチホップセッションの概念
Figure 1 Multi-hop session

また，前節のセキュリティ要件より，CA（認証局）を利用した電子証明書や中央サーバでの認証を前提としないため，ユーザが設定するパスワードをベースとした認証方式の検討を行った．またノード ID はグローバルで一意とし，ノード内においてユーザ ID が一意となるようにし，ノード ID またはユーザ ID に対してパスワードが設定するもの

とする．パスワードは総当り攻撃ができないように十分長いものを利用すること前提とする．また，オーバーレイネットワークの性質上，中間ノードによるパスワード漏洩の可能性があるため，チャレンジレスポンス認証を用いることとした．各ノードは対等であることから片方向の認証は脆弱であり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開始側から認証する方式を検討した．図 2 に提案方式の認証シーケンスを示す．

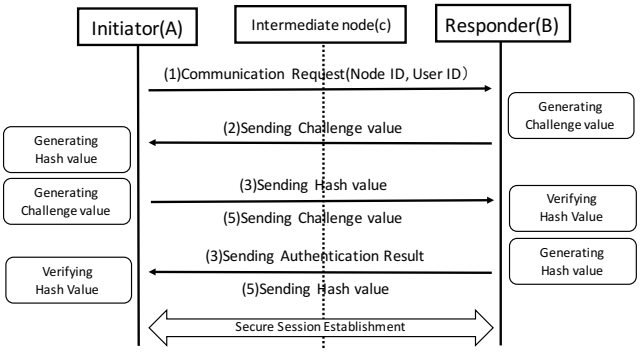

図 2　認証シーケンス

Figure 2　Authentication sequence

1. 通信開始側（A）から中間ノード（C）を経由して通信を受ける側（B）にノード ID またはノード ID に加えてユーザ ID を含む通信要求を送信する．
2. B が擬似乱数生成器によりランダム値(チャレンジ)を生成し，返却する．
3. A は受信したチャレンジとパスワードからハッシュ値を生成し B に返却する．
4. B は受信したハッシュ値と自身で生成したハッシュ値を比較し，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．
5. 続いて B は受信したチャレンジとパスフレーズからハッシュ値を生成し A に送付する．このチャレンジ値は（3）のレスポンスに含める．
6. B は送信したチャレンジと自身のパスワードからハッシュ値を生成し A に送付する．このハッシュ値は認証結果応答の（4）に含める．
7. A は受信したハッシュ値と自身で生成したハッシュ値を比較し，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，双方向認証が完了する．

4.2 暗号化通信

提案方式では電子証明書を用いた公開鍵暗号を使用しないため，事前に設定及び共有されているパスワードを用いたパスワードベース暗号[13]により生成する共有鍵を利用してメッセージの暗号化を実現する．図 3 に提案方式における共有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生成したランダム値（チャレンジ）をソルト値とし認証フェーズで決定された繰り返し回数によりパスワードから鍵を生成することで，相互認証されたノード間で共通の鍵が生成できる．

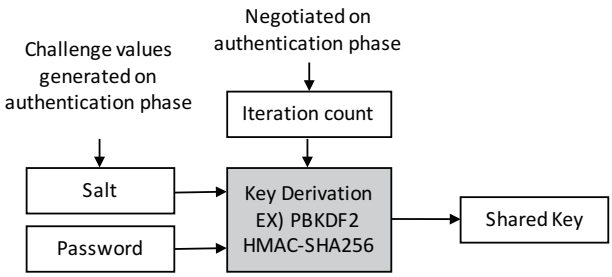

図 3　共有鍵生成手順

Figure 3　Shared key generation method

実際のデータの暗号化，復号化にはセッション鍵を用いる．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．セッション鍵は共有鍵により暗号化され,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するようにする．

4.3 メッセージ認証

オーバーレイネットワークを流れるメッセージは，中間ノードが参照する必要がある経路情報などの平文が含まれるため，メッセージ全体の完全性と送信者の正当性を担保する必要がある．提案方式では HMAC-SHA256 等用いて，パスワードとメッセージによるメッセージ認証符号により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の適用先として取り上げる PUCC プロトコルについて概要を説明する．

PUCC プロトコルは PUCC（P2P Universal Computing Consortium）[12]において策定されている通信プロトコルである．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オーバレイネットワーキングプロトコルを提供する事を目標としている．本研究では，非構造化ネットワークプロトコルの一例として PUCC プロトコルを取り上げ，提案方式の有効性の検証のため，本プロトコル上で提案方式の設計を行った．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，非構造型オーバー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，IEE1394，HTTP 等に対応しており，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 型ネットワークアーキテ

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，PUCC プロトコルは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，オーバレイネットワーキング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，レイヤー構成プロトコルを設計したり，複雑なアプリケーションプロトコルを追加するなど，フォーマットの拡張が容易である．

5.2 PUCC プロトコルスタック

PUCC プロトコルのスタック構成を図 4 に示す．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及び転送など，ノード ID に基づくオーバレイルーティング機能を有している．また，様々な下位トランスポート層とのバインドとしている．この PUCC Core Protocol の上位のプロトコルとして，ノードの探索やマルチキャストグループへの参加，エラーの通知などを行うための機能別の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 PUCC ノードの発見とその PUCC ノードへのサービス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 の応用アプリケーションについては，個別に PUCC Application Protocols として拡張定義される．

この階層構造や拡張性により，異種トランスポートプロトコルへの対応や，様々なアプリケーションへの適用が可能である．

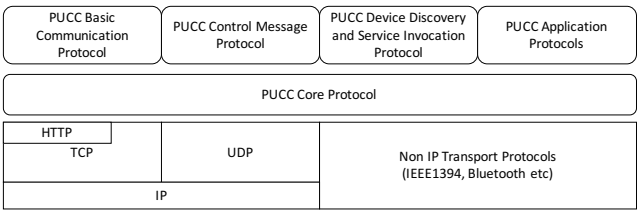

図 4　PUCC プロトコルスタックの構成
Figure 4　PUCC Protocol Stack

5.3 PUCC コネクションとセッション

PUCC プロトコルでは, 各ノードは一意のノード ID を持っており，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 5）．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 メッセージ及び 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間またはノード上のユーザ間で構築が可能になっている．

本研究では PUCC のマルチホップセッションの概念を用いて，セキュアな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．

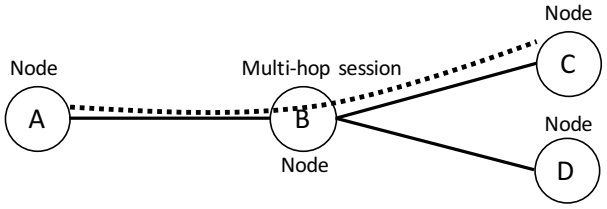

図 5　PUCC コネクションとセッション
Figure 5　PUCC Connection and Session

6. プロトコル設計

本章では 4 章で示した提案方式の実現可能性の検証のため, PUCC[12]で規定されている PUCC プロトコル上での設計を行った．

6.1 認証シーケンス

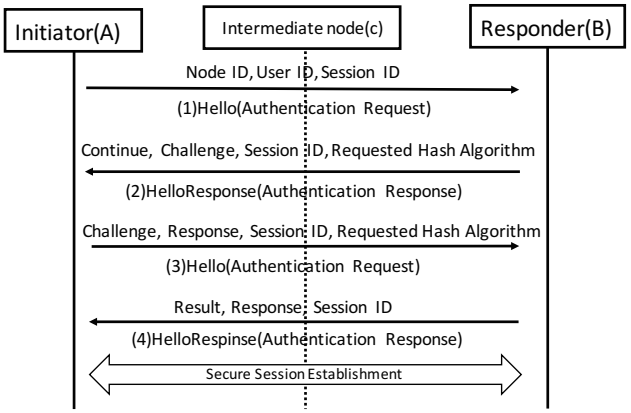

図 6　認証シーケンス
Figure 6　Authentication sequence

図 6 に示すようにリクエスト・レスポンスのシーケンスを 2 回実施することで相互認証を行う．通信開始側（A）から Node ID, User ID 及び Session ID を含む認証要求（Hello メッセージ）を送信する．次に通信を受ける側（B）は，認証処理継続のフラグ（Continue），チャレンジ，Session ID 及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，通信を受ける側の認証のためのチャレンジ，Session ID 及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，繰り返し回数，セッション鍵更新回数）も合わせて通知する．最後に，通信を受ける側（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合わせて通知する．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 7 に，XML で記述されあ 1 回目の Hello メッセージの例を示す．

```
<Core xmlns="Namespace of PUCC Core Protocol">
 <MsgType>Request</MsgType>
 <MsgID>12345.2002-12-20T16:15:32Z@968742ab-f9bb-4305-9900-f98e56f12352</MsgID>
 <Destination>
  <Target>874542ab-a5c6-4305-8745-f98e56f12547</Target>
 </Destination>
 <Source>968742ab-f9bb-4305-9900-f98e56f12352</Source>
 <ComType>Unicast</ComType>                                   Session ID
 <SessionID>12345.2002-12-20T16:15:32Z@968742ab-f9bb-4305-9900-f98e56f12352</SessionID>
 <MsgBody protocol="Namespace of PUCC Basic Communication Protocol">
  <Hello xmlns="Namespace of PUCC Basic Communication Protocol">
   <InitiatorUserID>urn:pucc:user:A</InitiatorUserID>
   <ResponderUserID>urn:pucc:user:B</ResponderUserID>
   <Authentication/>
  </Hello>
 </MsgBody>
</Core>
```

図 7 Hello メッセージの例

Figure 7　Example of Hello message

6.2 暗号化通信

暗号化アルゴリズムとしては AES（Advanced Encryption Standard）[14]あるいは Camellia[15]の 128bit 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[16]に従う.データの暗号化は, W3C XML Encryption Scheme[17]に準拠して実行する．全ての暗号化要素は EncryptedData 要素内に設定される．EncryptionMethod 要素には使用する暗号化アルゴリズムを指定する．ds:KeyInfo 要素内には，Encryptedkey 要素がありその配下の 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 8 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 要素内には共有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データが設定される．

6.3 メッセージ認証

PUCC プロトコルのメッセージは，ルーティングに必要な要素（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[18]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毎に行うことから 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 9 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．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．

```
<Core xmlns="Namespace of PUCC Core Protocol">
 <MsgType>Request</MsgType>
 <MsgID>12345.2002-12-20T16:15:32Z@968742ab-f9bb-4305-9900-f98e56f12352</MsgID>
 <Destination>
  <Target>874542ab-a5c6-4305-8745-f98e56f12547</Target>
 </Destination>
 <Source>968742ab-f9bb-4305-9900-f98e56f12352</Source>
 <ComType>Unicast</ComType>
 <SessionID>12345.2002-12-20T16:15:32Z@968742ab-f9bb-4305-9900-f98e56f12352</SessionID>
 <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element"
xmlns="http://www.w3.org/2001/04/xmlenc#">
  <EncryptedMethod Algorithm="AES-256"/>
  <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">      Encrypted Session Key
   <EncryptedKey>
    <CipherData>
     <CipherValue>r7us902Ws</CipherValue>
    </CipherData>
   </EncryptedKey>
  </ds:KeyInfo>
                                                                  Encrypted data
  <CipherData>
   <CipherValue>A23B456C56XuysjfhShskeSplaedSarkfbseSkrgksANje49Ud9Js2</CipherValue>
  </CipherData>
 </EncryptedData>
</Core>
```

図 8 暗号化メッセージの例

Figure 8　Example of encrypted message

```
<Core xmlns="Namespace of PUCC Core Protocol">
 <MsgType>Request</MsgType>
 <MsgID>12345.2002-12-20T16:15:32Z@968742ab-f9bb-4305-9900-f98e56f12352</MsgID>
 <Destination>
  <Target>874542ab-a5c6-4305-8745-f98e56f12547</Target>
 </Destination>
 <Source>968742ab-f9bb-4305-9900-f98e56f12352</Source>
 <ComType>Unicast</ComType>
 <SessionID>12345.2002-12-20T16:15:32Z@968742ab-f9bb-4305-9900-f98e56f12352</SessionID>
 <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
   <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
   <SignatureMethod Algorithm="HMAC-SHA-256"/>
   <Reference URI="xpointer(/Core)"></Reference>
  </SignedInfo>                                                  Message Authentication Code
  <SignatureValue>NK8A3AS5USH874GH</SignatureValue>
 </Signature>
 <MsgBody protocol="Namespace of PUCC Control Message Protocol">
 <Diagnose xmlns="Namespace of PUCC Control Mesage Protocol">
  <DiagnoseData>104353920000</DiagnoseData>
  <DiagnoseDestination type= " NodeID">968985ab-e6aa-5842-1234-
f98e56f15687</DiagnoseDestination>
 </Diagnose>
 </MsgBody>
</Core>
```

図 9　署名メッセージ例

Figure 9　Example of signature

7. ユースケース

本研究で想定するユースケースの適用例としては, 図 10 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ーからの家電の遠隔診断などを想定している．外部ネットワークからホームゲートウェイを介

しマルチホップで各家電デバイスにアクセスするオーバーレイネットワークを構築する．例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ホーム GW を介して情報家電と認証を行い，マルチホップ通信により遠隔制御を行う．家電メーカーからの家電の遠隔診断を行う場合も，メーカー側の診断サーバから GW を介して家電の認証を行い，マルチホップ通信により遠隔診断を行う．双方向でやり取りする場合でも GW を介したマルチホップ通信のため，宅内の機器に必ずしもグローバル IP を付与しなくてもよくセキュリティ上のメリットがある．また，IP ネットワークだけでなく，非 IP ネットワーク（NFC，Bluetooth，ZigBee）も含めた異種ネットワーク間で接続性の確保が必要となる場合においても，オーバーレイネットワークを活用する利点がある．

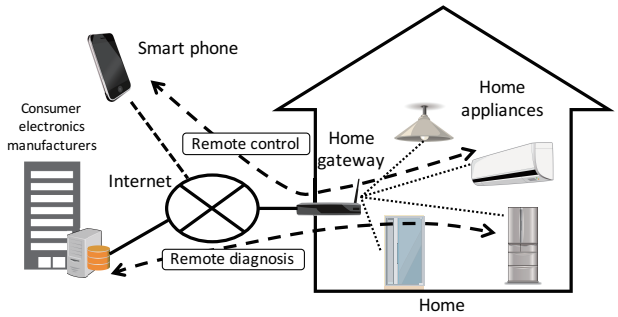

図 10 ホームネットワーク
Figure 10 Home network

8. 関連研究

オーバー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を利用した方式がある．公開鍵を利用した方式[19]として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認証を使った方式[20]としては，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取ることで認証を行う．公開鍵を利用した方式では，公開鍵を生成，登録する煩雑さがあり，また信頼の輪をベースとした認証となるため，必ずしも信頼性が保証出来ないなどの問題点がある．また，ワンタイムパスワードを使った方式としては，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，同じパスフレーズが登録されているノード同士が認証されていることから十分に長いパスフレーズが設定されていれば，信頼できるノードのみとの認証や暗号化通信が可能となる．また，通信開始時に毎回，パスフレーズベースでの暗号鍵等を生成するため状態を持っておらず，同期が外れてしまう問題は発生しない．

9. まとめと今後の課題

本稿では，非構造化オーバーレイネットワークにおけるセキュリティ方式として，パスフレーズベースの認証，暗号化，メッセージ認証の提案を行った．オーバーレイネットワーク上で，エンド・エンド間の認証・暗号化を可能とするマルチホップセッション上での認証・暗号化方式を提案した．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をソルト値として暗号鍵を生成する方式の提案を行った．さらに提案方式の実現可能性の検証のため，PUCC プロトコル上での設計について述べた．今後の課題を以下に述べる．

- Self-certifying ID[21]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．上記の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．

- 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．

- 実装による性能／有用性評価

提案方式を PUCC プロトコル上で実装し，その性能評価，有用性の検証などを実施する予定である．

参考文献

1) Kevin Ashton, "That 'Internet of Things' Thing," RFID Journal, July 2009.
2) Ion Stoica, et al. “Chord:A Scalable Peer-to-Peer Lookup service for Internet Applications,” ACM SIG- COMM Computer Communication, pp.149-160, 2001.
3) S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, “A scalable content addressable network,” in Processings of the ACM SIGCOMM, 2001, pp. 161–172.
4) A. Rowstron and P. Druschel, “Pastry: Scalable, distributed object loca- tion and routing for large-scale peer-to-peer systems,” in Proceedings of the Middleware, 2001.
5) http://freenet.sourceforge.net/
6) Gnutella protocol development, http://rfc-gnutella.sourceforge.net/index.html
7) JXTA, https://jxta.kenai.com
8) Norihiro Ishikawa, Takeshi Kato, Hiromitsu Sumino, Kazuhiro Kitagawa and Nobuo Saito: Recent Activities in PUCC and Its Application to Integrated Home Network Control and Management, 7th

IEEE Consumer Communications and Networking Conference (CCNC 2010), Januraty 2010.
9) Masatoshi Ogura, Hiroshi Mineno, Norihiro Ishikawa, Tomoyuki Osano, Tadanori Mizuno, “Automatic GUI Generation for Meta-data Based PUCC Sensor Gateway,” 12th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems (KES2008), pp.159-166, Sep.2008.
10) Yang-hua Chu, Sanjay G. Rao and Hui Zhang, "A Case For End System Multicast", Proceedings of ACM SIGMETRICS, Santa Clara,CA, Jun. 2000.
11) D. Pendarakis, S. Shi, D. Verma and M. Waldvogel, “ALMI: An Application Level Multicast Infrastructure,” Proceedings of the 3rd USNIX Symposium on Internet Technologies and Systems (USITS '01)
12) Norihiro Ishikawa, Takeshi Kato, Hiromitsu Sumino, Singo Murakami and Johan Hjelm: PUCC Architecture, Protocols and Applications, 4th IEEE Consumer Communications and Networking Conference(CCNC 2007), January 2007.
13) PKCS #5: Password-Based Cryptography Specification, RFC2898
14) Advanced Encryption Standard, FIPS PUB 197
15) Camellia, http://info.isl.ntt.co.jp/crypt/camellia/index.html
16) PKCS #7: Cryptographic Message Syntax, RFC2315
17) XML Encryption Syntax and Processing, W3C Recommendation
18) XML Signature Syntax and Processing, W3C Recommendation
19) 武田敦志, 北形元, 松島悠, 木下哲男, 白鳥則郎, "P2P ネットワークのための分散ハッシュ型認証手法", 第 6 回科学技術フォーラム FIT2007 情報技術レターズ 5, LM-009, pp.433-436, 2007.9.
20) 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. 31-36, Oct 2006.
21) A. Venkataramani, J. F. Kurose, D. Raychaudhuri, K. Nagaraja, M. Mao and S. Banerjee, ”MobilityFirst: A Mobility-Centric and Trustworthy Internet Architecture,” ACM SIGCOMM. Computer Communication Review (CCR), Volume 44 Issue 3, Pages 74-80, July 2014.