情報処理学会論文誌 Vol.53 No.1 385–402 (Jan. 2012)

DNSSECリソースレコードを用いたアドホックネットワークノード間公開鍵認証方式

鈴木 茂哉[1,a)] 石原 知洋[2] ビル マニング[3] 村井 純[4]

受付日 2010年12月24日, 採録日 2011年10月3日

概要：携帯電話のような携帯デバイスは，インターネットへ接続する機能を持つが，デバイス間で直接通信するネットワーク機能をあわせて持つものもある．本論文ではデバイス間の直接通信によりその場で構成されるネットワーク部分をアドホックネットワークと呼ぶ．アドホックネットワークは，ファイル交換等に有用と考えられるが，アクセス制御のための認証操作が煩雑なため活用されていない．認証操作が煩雑なのは，アドホックネットワーク環境で相手を認証する際，信頼できる第三者を仮定できず，パス・キーの手入力等主たる通信路以外の通信路（アウトオブバンド通信）に頼る必要があるからである．本論文では，アドホックネットワークにおけるノード間認証方式として，アウトオブバンド通信に頼らない方式を提案する．提案方式では，本方式参加ノードそれぞれが，ノード自身の識別のために必要な公開鍵（NK）と公開鍵に至る信頼の連鎖（NKCT）を事前に用意する．検証者は，立証者と自身の持つ NKCT 双方を組み合わせることで，必要な信頼の連鎖を構成できる．信頼の連鎖の確保により，認証が可能となる．また，信頼の連鎖の確保に DNSSEC リソースレコードを用い，運用性を高めた．本研究の評価では，検証ライブラリとともに，簡単なスマートフォンアプリケーションと，サーバ用認証モジュールを実装し，実験により本方式の効率性と有効性を示した．

キーワード：インターネット，アドホックネットワーク，DNSSEC，公開鍵基盤，認証

Public Key based Authentication Scheme for Ad-hoc Network Nodes Using DNSSEC Resource Records

Shigeya Suzuki[1,a)] Tomohiro Ishihara[2] Bill Manning[3] Jun Murai[4]

Received: December 24, 2010, Accepted: October 3, 2011

Abstract: Devices such as notebook PCs or mobile phones may have some ad-hoc network functionality for direct communication, in addition to connectivity to the Internet or an intranet. However, the scheme is not used for services like file exchange, because access control is cumbersome. It is impossible to assume the presence of a trusted third party, so the system should rely on some out-of-band communication such as pass-key entry by a human. In this paper, we propose a scheme applicable to node-to-node authentication in an ad-hoc network which does not rely on such out-of-band communication. In our scheme, a node prepares the information necessary to prove its authenticity – its own public key (NK) and chain of trust toward the public key (NKCT). The verifier can establish the chain of trust by combining both the prover's and its own NKCT. By establishing the chain of trust, the verifier can authenticate the prover. We improved ease of operation by providing this chain of trust using DNSSEC Resource Records. We have implemented a library, and also created a simple smart phone application and authentication modules for Unix server systems for evaluation. Our tests proved efficiency and effectiveness of the proposed scheme.

Keywords: Internet, ad-hoc network, DNSSEC, public key infrastructure, authentication

1. はじめに

近年，携帯型 PC を仕事に持ち歩き，いわゆるモバイル環境で用いる人は多い．職場から持ち出し，職場以外の環境で会議に参加したり，自宅で用いたりと，活用場所は様々である．携帯電話等によるデータ通信ネットワークや無線 LAN ホットスポット等を活用すれば，インターネットにアクセスすることもできる．

一方，PC や携帯電話のような携帯デバイスは，上記のように，インターネットやイントラネット等のインフラストラクチャとしてのネットワークに通信する機能に加え，ノード間で局所的に直接通信する機能を持つものがある．たとえば，PC 間，あるいは PC と携帯電話の間の通信は，Bluetooth [1] のような形で標準化されている．Bluetooth の場合，マウス等の無線接続や，ヘッドセットへの音声出力等に加え，ファイル転送も可能である．

しかし，会議等に集まった人々が持参した PC の間でファイルのやりとりをするとき，このような，一時的，すなわちアドホックに作成して使われるアドホックネットワークによる通信は用いられない（本論文では，PC や携帯電話のような携帯デバイス間において，その場で設定してデバイス間で直接にやりとりする一時的に作成可能な，Bluetooth 等の直接通信によって構成されるようなネットワークを，アドホックネットワークと定義する）．そのような場合は，何らかの方法でインターネットにアクセスし，インターネット上のファイル共有サービスやメールを経由させる方法や，USB メモリの手渡しによる方法が用いられているのが現状である（以下，これら既存の手段を，「外部サービスによるファイル交換」と表記する）．

アドホックネットワークが活用されないのは，通信時に簡単かつ確実なアクセス制御手段を用意できないことが原因と考えられ，この実現を妨げているのは認証の煩雑さである．一般にノード間認証は，通信を行う 2 つのノードのほかに，信頼できる第三者にあたるノードが間に立つか，なんらかの手段で秘密を共有する必要がある．アドホックネットワークにおける通信のように，信頼できる第三者を仮定できない環境では，主たる通信路（インバンド – in-band）以外の通信路を用いるアウトオブバンド通信（out-of-band）を用いて秘密の共有をするのが定法である [2]．携帯電話と Bluetooth ヘッドセットのような携帯デバイス間の結び付け（ペアリング）を例にあげると，アウトオブバンド通信を確保する方式として，パス・キーの手入力 [1]，カメラを用いた視覚経由（visual channel）[3]，音声の活用 [4]，振動に対するボタン操作 [2] 等が標準化あるいは提案されている（以下，「外部操作による認証方式」とする）．Ion らは，これらの方式による携帯デバイスのペアリング操作を実際の利用形態に近く，セキュリティ強度が異なる複数の操作手順により比較調査しているが，最低限のセキュリティを確保できる手順で，パス・キーの入力を含め，操作に最低 15 秒以上かかっていると報告している [2]．

ユーザが，会議の場でファイル交換するとき，外部サービスによるファイル交換方式と，外部操作による認証方式それぞれの手間を比較することを考える．操作の煩雑さや所要時間を比較すれば，認証が必要な直接通信よりは外部サービスによるファイル交換の方を選択するのは自然である．15 秒の操作が必要なペアリングよりは，差せば使える USB メモリが好まれるのは当然であろう．

ここで，簡便な認証を妨げているのは，認証に参加する 2 つのノード間に信頼関係を構築できないからである．アドホックネットワークにおいて，認証のための操作を容易に，あるいは操作の必要をなくすためには，信頼できる第三者が不要で，かつ，アウトオブバンド通信に頼らない手法が必要である．これを実現し，ごく簡単な操作で認証と許可の操作ができるようなアクセス制御方式を提供できれば，外部サービスによる方式よりも手軽に利用できる可能性がある．

本論文では，アドホックネットワークのような，信頼できる第三者が存在しない環境におけるノード間認証において，アウトオブバンド通信に頼る必要のない方法を提案する．立証者が，自身の公開鍵と識別名の結び付きを証明するために十分な情報を事前に自ノードに保存し，立証時に，検証者に渡す．このことにより，検証者が，立証者の識別名と，識別名に結び付いた鍵の所有を，検証可能な形で第三者に頼らずインバンド通信のみにより入手し，認証に利用できる．同時に，この応用において X.509 PKI よりも適した方式として，Domain Name System（DNS）[5] のセキュリティ機能拡張である DNSSEC [6] のリソースレコード（Resource Record – 以下 RR と表記）[7] の信頼の連鎖としての活用を提案する．

この論文での貢献は，以下のとおりである：

- 信頼の連鎖を構成する署名ひと揃いを，認証における立証者自身が自ノードに保存し，検証者に提示することで，信頼の連鎖に共通部分を持つノード間であれば，第三者に依存せず，検証できることを示した．
- この方式の実現のために，DNSSEC 関連 RR を信頼の連鎖として活用可能であり，X.509 による証明書よ

[1] 慶應義塾大学大学院政策・メディア研究科
Graduate School of Media and Governance, Keio University, Fujisawa, Kanagawa 252–8520, Japan
[2] 東京大学大学院総合文化研究科
Graduate School of Arts and Sciences, The University of Tokyo, Meguro, Tokyo 153–8902, Japan
[3] 南カリフォルニア大学情報科学研究所
USC Information Sciences Institute, Marina del Rey, CA, 90292, United States of America
[4] 慶應義塾大学環境情報学部
Faculty of Environment and Information Studies, Keio University, Fujisawa, Kanagawa 252–8520, Japan
[a)] shigeya@wide.ad.jp

り運用が容易であることを示した．

次章以降の記述は，以下のように構成されている．2 章では，認証と信頼の連鎖の関係と信頼の連鎖確立への公開鍵暗号の活用について整理する．3 章で提案の詳細を説明し，4 章において関連研究について示す．5 章で試験実装について説明する．6 章で提案手法を評価し，最後に 7 章で議論をまとめる．

2. 認証と公開鍵暗号による信頼の連鎖の確保

この章では，互いの信頼を確認できないノードの間での通信におけるアクセス制御を，信頼できる第三者が存在しない状態で適用するために必要な技術と必要な情報について整理する．以下，ノード間アクセス制御と認証，認証と立証者検証者間の信頼関係について整理した後，公開鍵暗号と認証方式，信頼の連鎖の確保と認証の可否，公開鍵基盤による信頼の連鎖の確保について説明する．つづけて，公開鍵基盤の代表例である X.509 PKI，および，本論文での提案で信頼の連鎖として用いる DNSSEC について説明する．

2.1 ノード間アクセス制御と認証

アクセス制御を実装するには，アクセス元の認証，すなわち，アクセス元が誰であるかを検証し，何らかの識別名として認識する必要がある．Lampson [8] は，計算機資源に対するアクセス制御を，**図 1** のように，アクセス元（Source）からの操作要求に対し，認証（Authentication）によりアクセス元が誰であるかを得たうえで，アクセスルールを適用し資源へのアクセスを許可（Authorization）すると定義している．アクセス元は，認証において，資源へのアクセス権限を持つ相手に対し，自分の身元を相手が理解し区別できる表現形式（識別名）で示す必要がある．アクセス元は，自身が識別名で示されるノードであることを立証し，認証する側は，そのノードが，その識別名で示されるものであるかどうか検証する．ひとたび認証ができたなら，識別名に応じて決められたアクセスルールを適用し，資源へのアクセスを許可する．

2.2 認証と立証者検証者間の信頼関係

認証の実現は，立証者が提示した識別名を検証者が信頼できると判断できるかどうかに依存する．識別名を 2 者間で立証・検証するためには，検証者から立証者に対する信頼の確立が必須である．逆に，なんらかの方法により信頼関係を確立できれば認証は可能である．

2 者間の信頼関係の確立を視点とすると，以下の 3 種類の方法をとりうる．

(1) 信頼できる第三者が検証の場に存在する場合，立証者と検証者を含む三者の信頼関係を用いる．
(2) 立証者と検証者の間で，何らかの秘密を，信頼できる直接的通信手段により共有し，用いる（アウトオブバンド通信）．
(3) 立証者の識別名をある時点で検証した第三者（中間検証者）が存在し，中間検証者による検証結果（中間検証結果）が手元にあり，検証者が中間検証者を信頼できるとする．このとき，検証者により，中間検証結果が中間検証者によるものであることを検証できれば，中間検証結果をよりどころとできる．

いずれの方法も，2 者間の信頼関係を複数組み合わせ，信頼関係を連鎖的に拡張できる．上記 (3) の方式の実現手段として，中間検査結果に公開鍵暗号を用いることができる．そして，公開鍵暗号を用いた信頼関係の連鎖の拡張を提供しているのが公開鍵基盤である．

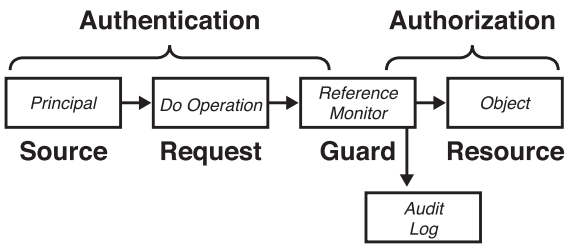


図 1 アクセス制御のモデル（Lampson [8] による）
Fig. 1 Access control model (Lampson [8]).

2.3 公開鍵暗号と認証方式

公開鍵暗号を用いると，特定の鍵（以下，公開鍵暗号の鍵ペアをセットにして「鍵」と略記する）を運用しているエンティティ（立証者）が，その鍵の実際の所有者であることを，なんらかの情報に対する立証者による署名を検証することで確認できる．鍵の所有が確実であれば，特定の公開鍵の所有者に対する確実な認証が可能である．一方，用いられた公開鍵自身の信頼性を何らかの方法で検証する必要があると同時に，公開鍵自身によって対象を識別することは困難であるので，立証者が示した公開鍵に結び付けられた識別名を得る必要がある．このために X.509 PKI [9], [10], [11] や PGP [12] 等の公開鍵基盤を利用し，検証者自身が信頼する公開鍵（証明書あるいは信頼点）を起点とし，検証対象となる公開鍵に至る信頼の連鎖を用意することで，識別名を得ることができる．

2.4 信頼の連鎖の確保と認証の可否

公開鍵基盤を用いて通信相手を識別するには，立証者から示された公開鍵に対して，検証者が信頼する信頼点を起点として信頼の連鎖を検証する．そのために，検証の対象となる立証者の証明書に対して，検証者の証明書を起点として到達するために必要な証明書を用意する．ここで，公開鍵証明書による証明の本質は，対象となる鍵に対する証明者の保持する鍵による署名である．以下，証明書とは，鍵に対する証明者による署名を示すこととする．

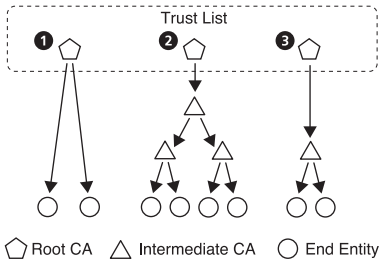


図 2 複数ルート階層型 PKI（RFC4158 [13] をもとに加筆）
Fig. 2 Multi-rooted hierarchical PKI (Based on RFC4158 [13]).

検証は，検証者が設定している信頼点を起点とし，立証者の証明書に至る，1 つあるいは複数の証明書で構成される信頼の連鎖が用意できて，はじめて成功する．信頼の連鎖を構成する証明書が検証時に不足している場合，検証できない．このような不足は，立証者の証明書へ到達可能な信頼点（証明書）と，検証者が利用可能な信頼点（証明書）の間にギャップがある場合に起こる．

たとえば，X.509 PKI における信頼点と認証局（Certificate Authority – CA）の信頼関係を示した**図 2** において，❶のルート認証局のみ信頼している検証者は，❸の木に属するエンドエンティティを検証できない（注：図中の矢印は証明者から被証明者への信頼を示す．文中の “❶” 等の表記は，図中の黒丸に白抜き文字の表記に対応している）．

一方，入手経路によらず，何らかの形で信頼の連鎖を構成する公開鍵証明書を揃えられれば，検証が可能となる．すなわち，検証時に，公開鍵の検証に十分な信頼の連鎖を揃え，かつ，到達可能な信頼点を信頼の連鎖の起点とできれば，対象となる公開鍵に結び付いた信頼できる識別名を得られる．公開鍵が識別できれば，当該公開鍵に対応する秘密鍵による署名を検証できる．これらの組合せにより，通信相手の公開鍵の入手，公開鍵に結び付いた識別名と公開鍵に対応する秘密鍵の保持の検証ができるので，公開鍵暗号を認証に用いることができる．

2.5 公開鍵基盤による信頼の連鎖の確保

公開鍵基盤は，公開鍵証明書によって，ある公開鍵が証明者によって検証されたことを示す．公開鍵基盤の機能は，以下のとおりである．

- ある鍵を持つ証明者により，ある識別名で示される被証明者の特定の鍵の所有を，被証明者の公開鍵に対する証明者による署名で示すこと
- 上記の署名情報と，被証明者の公開鍵や付随する情報を証明書として提供すること
- 必要に応じ，証明書の失効情報や，活用の判定に必要となる支援情報を提供すること

公開鍵証明書には，対象となる公開鍵に加え，公開鍵への証明者による署名と，有効期限等署名の有効性を示すための付加情報が含まれる．信頼の連鎖を検証するには，必要となる公開鍵証明書を入手し，信頼できる公開鍵証明書を起点とし，確認対象の公開鍵までの信頼の連鎖を入手した証明書から発見し，連鎖を構成するすべての証明書の有効性を検証する [14]．検証が成功すれば，証明書あるいは証明書の組合せにより識別名が明らかとなる．信頼の連鎖を構成するためには様々な方法がある．たとえば，X.509 PKI ではとりうる構成を規定しておらず，運用上様々な構成が可能であるが [13]，実際は，階層型で，かつ，信頼点（ルート認証局）が複数ある構成で運用されている（図 2）．このため，X.509 PKI を活用するアプリケーションは複数のルート認証局を信頼し用いている（図中 “Trust List”）．

2.6 X.509 PKI

X.509 PKI [9] は，X.509 公開鍵証明書を運用するための基盤である．ある立証者（サーバやクライアント等のエンドエンティティ）が保持する鍵が，特定の識別名に結び付いていることを証明する公開鍵証明書を運用する．公開鍵証明書は，認証局によって発行される．認証局は，他の認証局の公開鍵を署名し，公開鍵証明書を発行することで，認証局間の信頼関係を確立する．そして，認証局は，エンドエンティティが用いる，エンドエンティティ証明書を発行する．X.509 証明書は，設定された信頼点（証明書）を起点とし，信用の連鎖を検証しつつたどることにより検証できる．これらの信頼関係は単一方向であり，組合せにより，木構造，あるいは，有向グラフ状の信頼関係が構築される．当初，X.509 証明書は，X.500 ディレクトリ [15] のアクセス制御のために作られた標準であり，X.500 ディレクトリに合致する単一のルートの木構造との併用が想定されていた．この後，X.509 証明書が X.500 ディレクトリ以外に使われるようになるに従い，単一のルート認証局による構成で運用できなくなった．このため，X.509 PKI では，技術的には単一の CA を根元とした 1 本の信頼関係の木を作成することも可能であるが，実際の運用上，単一の信頼点となる認証局は存在せず，X.509 PKI を活用するアプリケーションが複数の認証局を信頼し用いているのが現状である．これにともない，証明書の利用者は，信頼点として，1 以上の認証局の鍵を信用することを決める．たとえば，ウェブブラウザの場合には，事前にいくつかの認証局の公開鍵証明書がインストールされている．鍵の失効判定については，Certificate Revocation List（CRL）を確認するか，オンライン証明書失効確認プロトコルである OCSP [16] を用いる．公開鍵証明書には CRL や OCSP レスポンダの所在が記載されており，ネットワーク経由でアクセスする．なお，それぞれの X.509 証明書（そして署名）が検証できたことの意味，その信頼性についての判断は，証明書を活用するユーザあるいはアプリケーションが決定するものであって，X.509 PKI はその決定を支援する機構を提供しているにすぎない．

2.7 DNSSEC

DNSSEC [7] は，DNS での情報の単位となるリソースレコード（RR）とドメイン名との対応付けを，公開鍵暗号によって担保する DNS 機能拡張である．担保されるのは，以下の 3 点である．

- 特定のドメイン名（Fully Qualified Domain Name – FQDN）の特定のタイプの RR 群（RRset）の構成（値と数）が，ある鍵を持つものによって，ある時点で確認されたことを示す．
- 特定の名前の特定のタイプの RR が存在しなかったことを，ある鍵を持つものによって，ある時点で確認されたことを示す（NSEC RR による非存在証明 [6]）．
- 上記を活用し，名前空間を構成する木の根元（ルート）に近い上位ゾーンの鍵の持ち主により，子にあたるゾーンの管理者が用いている鍵が正当であることを署名によって示す．ゾーンは DNS サーバにおける DNS 名前空間の管理単位であり，ゾーン単位で管理委譲する．上位ゾーンから子ゾーンの管理者の鍵に対して署名することにより，根元から下位のゾーンへと向かって，ゾーンの委譲と並んで，ゾーン所有者の鍵の正当性が示される．

これらを実現するために，公開鍵，署名，検証を支援するための情報等を，他の DNS の情報と同様に RR として実装している．

DNSSEC を用いると，手元に得た RR 群が，ゾーンの管理者が設定したとおりであることを検証できる（Origin Data Integrity の検証）．DNSSEC における信頼点は，理想的には DNS のルートゾーンであるが，ゾーンに登録された公開鍵であれば，DNS 名前空間を構成するツリーのどの部分のものであっても，信頼点として活用できる．ただし，そのような信頼点を選んだ場合，検証できるドメインの範囲は限定的となる．2010 年 7 月 15 日に DNS のルートゾーンが署名されたが [17]，検証可能なゾーンがまだ少ないのが現状である．

DNSSEC が必要となった背景は，ドメイン名から情報を解決するために RR を入手する際，通常は，RR をキャッシュする DNS リゾルバを利用するが，この RR を保持するキャッシュ（以下 RR キャッシュ）が汚染されるリスクがあるためである [18]．ここで，注目すべきなのは，RR のキャッシュ汚染へのリスク対策を目的として設計されているため，DNSSEC 機能を構成する署名や公開鍵等の RR は，信頼点を起点として，検証したい鍵にいたる信頼の連鎖を構成するのに足る有効な RR がそろっていれば，検証が可能であり，それぞれの RR の入手経路は問われない点である．構成要素の入手経路を問わない点は X.509 PKI についても同様であるが，構成要素である RR を X.509 証明書と比べ，より細かい単位で組み合わせることができる．

3. 提案

アドホックネットワークにおいて，あるノードが，別のノードに対して，自身の公開鍵，その公開鍵に結び付いた識別名と公開鍵に対応する秘密鍵の所有を立証し，認証に利用する方法を提案する．本章では，提案方式で対応しようとしている課題，提案方式の適用環境，提案方式の概略に続き，提案方式について説明する．

3.1 提案方式が対応する課題

アドホックネットワークのような，信頼できる第三者を仮定できない環境で，ノード間の信頼関係を確立を考える．信頼の確立には，秘密の共有が必要であり，手法としては，事前に通信相手を仮定して秘密を共有するか，アウトオブバンド通信を用いる方法がある．

しかし，通信相手を仮定すると，通信相手を限定することになる．また，アウトオブバンド通信に頼ると，操作が煩雑である．

認証における操作の煩雑さを回避するためには，アウトオブバンド通信に頼らず，なんらかの方法で信頼の連鎖を確保する必要がある．

3.2 提案方式の適用環境

本提案方式は，PC やスマートフォンのような，端末間での無線による直接通信機能を持ったモバイル端末で利用されることを想定する．さらに，これらの端末が，以下の 4 種類のネットワーク環境で使われると仮定する．

- 自宅や職場等の信頼できるネットワークに接続（インターネットアクセス可能）
- 携帯電話データ通信ネットワークが使える場合，それを用いた通信（インターネットアクセス可能）
- 他の組織や会議場等でのビジタとして利用（インターネットアクセス可能）
- なんらかの無線技術を用いた近隣ノード間直接通信（インターネットアクセス不可）

これらのうち最初の 3 種類は，用意されたインフラを利用することでインターネットへアクセスできる．一方，最後の近隣ノード間直接通信では，インターネットへアクセスできないが，直接通信さえできれば，他のインフラに依存せず，いつでも，どこでも，利用できる．以下，インターネットアクセスが可能な状態をオンライン状態，不可能な状態をオフライン状態と表記する．

ここで，以下のような特徴が想定できる：

- それぞれのモバイル端末は，つねにオンラインとは限らない．
- ただし，自宅や職場に戻るタイミング等で，オンラインになれる．すなわち，長期間オフラインのままであることはない．

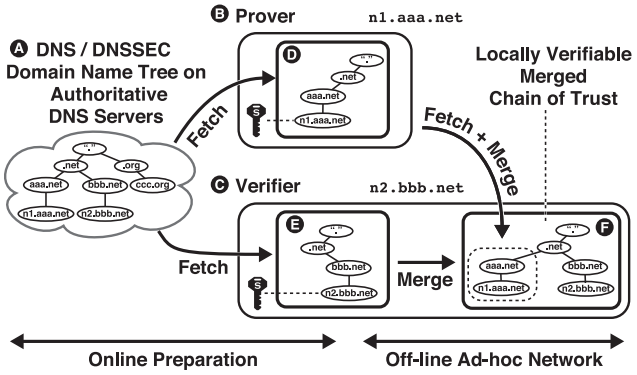


図 3 提案方式の概略
Fig. 3 Overview of the proposal method.

このような環境下で，オフライン状態で，あらかじめ相手を仮定せずに，通信相手を認証する方式を考える．

3.3 提案方式

本提案は，認証における識別名の立証において，検証者が必要とする情報のうち，立証者が用意できるものを立証者が事前に用意する方式である．オンライン時に，立証者自身の識別名を検証者に立証するために必要な信頼の連鎖を，立証者自身が保存する．そして，検証時に検証者に対して，この情報を提示する．検証者は，立証者が提示した信頼の連鎖と，検証者自身が持つ信頼の連鎖を組み合わせる．構成した信頼の連鎖にギャップがなければ，オフライン環境においても，検証が成功する．検証者と立証者の間の情報の共有のみによって検証できるため，オフライン環境下で存在を仮定できない信頼できる第三者に頼る必要がない．

2 章での議論のように，アドホックネットワーク環境で公開鍵暗号に基づく方式では，信頼の連鎖の確保が課題である．これを，立証者が自身の鍵（ノード鍵 – Node Key – NK．以下，NK の公開鍵を NK 公開鍵，NK の秘密鍵を NK 秘密鍵，NK は公開鍵秘密鍵のペアとする）を検証可能な信頼の連鎖を NK とともに常時保持するようにし，必要に応じて検証者に渡すことで，検証に十分な信頼の連鎖の確保を実現する．検証者は，受け取った信頼の連鎖に自身の持つ信頼の連鎖を組み合わせ，NK 自身の識別名を確認するとともに，NK による署名を検証する．

信頼の連鎖は，X.509 PKI や PGP ではなく，対象となる条件において鍵の交換手続きや証明コストといった運用の点で有利な DNS/DNSSEC RR を用いて確保した．ただし，DNS/DNSSEC RR を用いているが，名前の解決には用いておらず，ゾーン間の権限委譲と利用する公開鍵の信頼関係のみを活用している．

図 3 に本提案の概略を示す．DNSSEC により検証可能な DNS の名前ツリー（Ⓐ）から，立証者（Prover – Ⓑ），検証者（Verifier – Ⓒ）それぞれが，自身の持つノード鍵に至る信頼の連鎖（ⒹおよびⒺ）を事前に取得する．検証時は，立証者・検証者双方のノードが持つ信頼の連鎖を統合し検証することで，検証を必要とする鍵までの信頼の連鎖を確保する（Ⓕ）．

本提案方式による情報の追加および検証は，おおまかに以下の 3 つの部分で構成されている．

(1) NK の作成と，ゾーン管理者による所属ゾーンへの追加，署名，公開
(2) NK 公開鍵に至る信頼の連鎖の，通常の DNSSEC 仕様の範囲での検証
(3) NK 秘密鍵を唯一保有する当該ノードによって追加された RR とその署名を，NK 公開鍵で検証する，本提案の独自仕様

この (2)，(3) の検証結果により，立証者は，当該 FQDN に結び付けられた NK 秘密鍵を保持している，すなわち，当該 FQDN で示されるノードであることを検証者に対して立証できる．

以降の項で，前提条件，事前準備，立証準備，立証情報の取得と検証，信頼の連鎖の有効期間と更新タイミング，本提案方式の活用方法とユーザから見た操作，本提案方式のセッション確立プロトコルへの組み込みについて，順に説明する．ノードは，立証者・検証者双方の機能を持つことができるため，役割を立証者と検証者に分け，かつ，公開鍵を検証するのに十分な情報のやりとりの部分に特化して説明する．各ノードは，同等の情報を事前準備で持つが，検証者で検証に関係のない情報については説明を省略している．

3.3.1 前提条件

立証者（Prover），検証者（Verifier）ともに，以下の 4 点を前提とする．

(1) それぞれのノードを示す FQDN が属するゾーンで，DNSSEC が運用されており，インターネット上のノードから，そのゾーンに属する RR を入手できること
(2) それぞれのノードで，DNSSEC 検証リゾルバが設定されている，すなわち，信頼点（Trust Anchor）が設定されていること
(3) ノードを示す FQDN の検証が，少なくとも，設定された信頼点を起点として成功すること
(4) ノードは，随時，インターネットへ接続する機会があり，接続性がない状態が継続しないこと

3.3.2 事前準備（オンライン状態での信頼の連鎖の確保—NK と NKCT の準備）

事前準備では，NK を用意し所属ゾーンに登録した後，ノードの DNSSEC 検証リゾルバで設定されている信頼点を起点として，NK 公開鍵を確認するために必要な RR をすべて自ノードに保存する．これらの RR をまとめて，Node Key Chain of Trust – NKCT と呼ぶ．

事前準備は，ノードがオンライン状態であるときに行う．

図 4 に本提案の事前準備の手順を示す．各参加ノードの事前準備は，以下のとおりである．DNS でのラベルとは，あるゾーン内で，登録される RR を示す FQDN を識別するのに足る文字列のことをいう．そして，同一ゾーン中の同一ラベルを持つ同一型の RR は，まとめて RRset と呼ばれる．以下，型 x の RR を x-RR と表記する．

❶ NK を生成する．

❷ NK 秘密鍵（図中 “S” と表記の鍵）をノードに保存する．

❸ NK 公開鍵（図中 “P” と表記の鍵）を RR(*DNSKEY-RR*) として用意する．

❹ 用意した *DNSKEY-RR* を所属ゾーンの管理者に渡し，ゾーンへの登録，ゾーン署名鍵（ZSK）による署名，通常の DNS ゾーン中の RR として公開の 3 点を依頼する．ゾーン管理者は，この 3 点の操作を行う．同じゾーン中に複数の本提案方式参加ノードの鍵を登録できる．このとき，それぞれの NK 公開鍵は，異なる FQDN を持つ．ここで，NK 公開鍵を署名するために，運用上のメリットから鍵署名鍵（KSK）ではなく，ZSK を用いている．通常，ゾーン委譲ポイントにおける鍵あるいは鍵に相当する RR に対する署名は ZSK を用いる．また，NK 公開鍵に類するゾーンに属するレコードを KSK で署名してしまうと，ZSK/KSK を運用上分離した意味が失われる（6.2 節参照）．したがって，NK の署名には ZSK を用いるのが適切である．

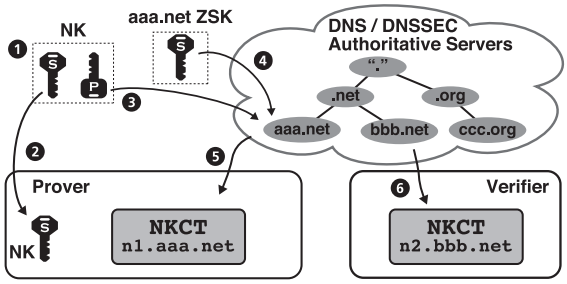


図 4 提案方式の事前準備手順
Fig. 4 Preparation steps of the proposal method.

❺ DNSSEC 検証リゾルバを用い，NKCT に必要な RR をすべて自ノードに保存する．NKCT を構成する RR はインターネットで公開されている情報なので，通常の DNS リゾルバを用いて取得できる．保存する際，自身の NK 公開鍵に至る署名を，DNSSEC 検証リゾルバに設定されている信頼点を起点にして検証する．このとき，RFC5011 [19] に基づいて運用している場合は，鍵無効（revoke）ビットの確認やロールオーバ処理を行い，取得時点で有効な RR を揃える．通信が正常に行われなかったり，攻撃を受けている等の事情により検証が成功しなかったりした場合は，NKCT 全体を破棄し，状況に応じて再取得を試みる．取得が成功しない限り，検証者に対して立証できない．

NKCT を構成する RR の一例として，n1.aaa.net というノードに対するルートを信頼点とした場合の NKCT の一覧を**表 1** に示す．NKCT を構成するのは，信頼の連鎖を構成する *DNSKEY-RR*（公開鍵），*DS-RR*（親ドメインからの使用鍵指示レコード），*RRSIG-RR*（RR に対する署名結果）の 3 種の RR である．NKCT は，それぞれの RR の持ち主のラベル（FQDN）が含まれているが，信頼の連鎖の構築を目的としており，名前解決を目的としない．表中の各行が，1 つの RR と *RRSIG-RR* に用いられる鍵を示している．それぞれの RR に対し，*DNSKEY-RR* 以外の対象は所属するドメインの ZSK で署名が作成され，*DNSKEY-RR* は RRset 全体に対して KSK で署名された結果が用いられる．親ドメインから子ドメインへの委譲点での署名は，移譲先の子ドメインの KSK のハッシュを示す *DS-RR* に対する親ゾーンの ZSK の署名で示される．一番右の列で示された鍵が，署名に用いられる公開鍵である．表中では，黒丸のアルファベットで対応関係を示した．この表にない RR は信頼の連鎖の検証に必要がないので，NKCT には含まれない．なお，非存在証明のための NSEC RR [6] による整合性は，NKCT を保存する前に確認するため NKCT には含めない．NKCT が検

表 1 NKCT/LCT を構成する DNS RR と，それぞれが署名に用いる鍵
Table 1 Resource records for NKCT/LCT and the keys to sign the RRs.

NKCT or LCT	Label (FQDN)	Data RR (signing target)	Key to sign the RR/RRset for the RRSIG
NKCT	.	DNSKEY(KSK) Ⓐ	entire DNSKEY RRset signed by
NKCT	.	DNSKEY(ZSK) Ⓑ	KSK ”.” Ⓐ
NKCT	net.	DS	ZSK ”.” Ⓑ
NKCT	net.	DNSKEY(KSK) Ⓒ	entire DNSKEY RRset signed by
NKCT	net.	DNSKEY(ZSK) Ⓓ	KSK ”net.” Ⓒ
NKCT	aaa.net.	DS	ZSK ”net” Ⓓ
NKCT	aaa.net.	DNSKEY(KSK) Ⓔ	entire DNSKEY RRset signed by
NKCT	aaa.net.	DNSKEY(ZSK) Ⓕ	KSK ”aaa.net.” Ⓔ
NKCT	n1.aaa.net.	DNSKEY(NK) Ⓖ	ZSK ”aaa.net” Ⓕ
LCT	n1.aaa.net.	A	NK ”n1.aaa.net” Ⓖ

証でき，かつ，NSEC RR で，NKCT 関連 RR の存在が確認できれば，NKCT の存在は明らかである．

作成した NKCT 全体の有効期間は，NKCT を構成するすべての署名のうち，最も早く到達する有効期限までとなる．NKCT の有効期間を最大限確保するために，ノードがインターネットに接続している間は，DNS RR 取得時の TTL に従い，NKCT を構成する RR が一部でも無効化された時点で，無効化された RR を取得し，NKCT を再検証する．更新タイミングと頻度の関係については，3.3.5 項で整理した．

❻ 検証者も，立証者と同様に NKCT を用意しノードに保存する．

なお，DNSSEC の署名は RRset に対するもので，用いられた鍵と署名の有効期限で区別されて保存される．これにより，ある RRset に対して，本来のゾーン管理者が ZSK で署名した結果とともに，NK で署名した結果をゾーンの RR の一部として管理しても，DNSSEC の仕様上問題がない．また，RFC4034 [7] では，*DNSKEY-RR* に DNS インフラストラクチャに直接関係ない鍵を導入することが禁止されているが，本提案は DNS インフラストラクチャに直接関係あると解釈して利用している．

3.3.3 立証準備（立証環境における LCT の作成）

立証が必要な環境におかれた時点で，立証準備する．以下で手順を述べる．ある NK が，特定の FQDN に結び付けられていることを示すとともに，立証者の NK 秘密鍵の保有を立証するために必要な情報の集まりを，Local Chain of Trust（以下 LCT）と呼ぶ．LCT は，NKCT に対して追加するなんらかの RR と，追加された RR に対する NK 秘密鍵による署名である *RRSIG-RR* を含む．立証環境で追加する RR を NK 秘密鍵を用いて署名することで，立証者が NK 公開鍵に対応する NK 秘密鍵の保持を立証する．追加する RR は，NK と同一のラベルを持ち，タイプは DNSSEC と DNS の構造を示す RR タイプ以外であればよい．ここでは，立証環境で得られる情報の 1 つである IP アドレスを示す *A-RR* あるいは *AAAA-RR* を用いる．IP アドレスを用いるのは必須条件ではないが，アドホックネットワーク環境で得られる情報の 1 つであり，RR タイプとして定義されており，かつ，実際の通信時に用いられるアドレスの合致の確認も可能な点でメリットがある．**図 5** に本提案の立証と検証の手順を示す．

以下，立証者が LCT を作成する手順を示す．この例では，ネットワークのローカルアドレスを持つ *A-RR* レコードを署名対象としている．なお，本提案では，オフライン状態での検証を目的としているが，提案方式の検証自体はオンライン状態で行われても成立する．LCT は，LCT を構成する情報である NKCT あるいは署名対象に変化があった時点で，作成あるいは更新する．

❼ ネットワークインタフェースから IP アドレスを随時取得する．

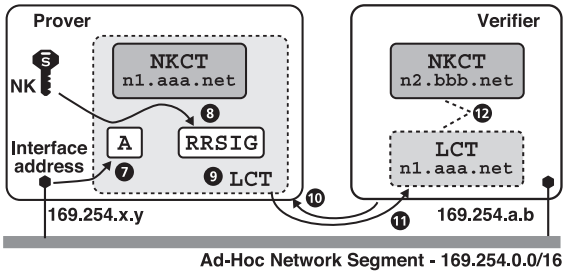


図 5 提案方式の立証–検証手順
Fig. 5 Prove-verify steps of the proposal method.

❽ ネットワークインタフェースの IP アドレスの割当てや変更が起きたとき，IP アドレスに対応する *A-RR* または *AAAA-RR* を用意する．RR を用意した後，NK 秘密鍵で署名し，署名 RR(*RRSIG-RR*) を作成する．署名の際は，署名の有効期間として，NKCT の有効期間内で想定される検証者が LCT を受け取り検証を完了するまでに必要な時間を考慮し，検証が成功する期間を最低期間として設定する．通常は，署名時の時刻から，NKCT の最長有効期間までを有効期間とする．なお，ここで用意する *A-RR* あるいは *AAAA-RR* は，3.3.2 項で述べたように，NKCT には含まれないので LCT に含まれる唯一の *A-RR* あるいは *AAAA-RR* となる．

❾ 追加作成した RR と NKCT をまとめ，LCT として保存する．LCT は，NKCT を構成する RR に加え，*A-RR* あるいは *AAAA-RR* と，それに対応する *RRSIG-RR* で構成される（表 1）．なお，想定された状況以外での検証の成功を導くリスクがあるため，作成された LCT は，DNSSEC 本来の仕様のとおりに動作する DNSSEC 検証リゾルバでの検証用の情報（あるいはキャッシュ）と厳密に区分して扱い，DNSSEC 本来の検証に用いない．

3.3.4 立証情報の取得と検証

検証者は，立証者から立証のための情報（LCT）を入手し，LCT を検証する．検証は，オンライン・オフラインどちらの状態でも，立証者が LCT を準備しているのであれば，可能である．以下に手順を示す．

❿ 検証者は，立証者に LCT を要求する．

⓫ 検証者は，LCT を立証者から受け取る．LCT 授受のプロトコルに特に指定はなく，転送される情報を後で検証することから，暗号化や保護も不要である．5 章の実装例では，HTTP を用いている．HTTP（TCP）であれば対向する接続先のアドレスを知ることができるのでノードに署名されて渡された *A-RR* あるいは *AAAA-RR* どちらかのタイプの RR と一致することを確認する．なお，得られたアドレスとノードとの一致

は，TCP コネクションのセキュリティ上の安全性の範囲に限定されていることに留意する．ここで重要なのは署名の検証結果であり，アドレス自身の値の検証は副次的である．単純なアドレスだけの認証では，リプレイアタックのリスクがある．

⓬ 署名に用いられた秘密鍵に対応する公開鍵を立証者から受け取った LCT から発見し検証した後，必要に応じて自身の NKCT と組み合わせ，信頼点からの信頼の連鎖を検証する．検証は，NK の属するゾーンから DNS ゾーン階層の根元方向に向かって順次進め，自身の持つ信頼点と位置が一致した時点で成功とする．たとえば，n1.aaa.net の所属するゾーンである，aaa.net から親の net，最後はルートゾーン（.）と順に確認する．なお，検証手続きでは，立証者の NKCT と LCT，検証者の NKCT に含まれる RR 以外を用いない．ここで，立証者の NKCT と検証者の NKCT による信頼の連鎖が交差していないと検証は成功しない．たとえば，図 4 を例にあげるなら，aaa.net と bbb.net は，net とルートを共有しているので，信頼点に net，ルートどちらを選択している場合においても成功する．一方，aaa.net と ccc.org に属するノードが検証する場合，ルートを信頼点として NKCT を保持している場合は検証は成功するが，いずれかのノードがルートでなく，.net あるいは.org を信頼点として選んで NKCT を構成している場合，交差がないため，検証は失敗する．

この段階でノードの検証済みの識別名である FQDN に一致する公開鍵を検証者が入手でき，かつ立証者の秘密鍵の保持も確認できたことになる．以降，公開鍵と識別名をセッションを確立するための認証手段としてノード間で活用できる．

3.3.5 信頼の連鎖の有効期間と更新タイミング

作成した信頼の連鎖（NKCT）には，有効期間がある．DNSSEC の署名と NKCT の更新は，DNS RR の TTL（Time To Live）パラメータ [20] と密接な関係があり，NKCT の有効判定や更新も，それに従う必要がある．

本提案方式では，オンライン状態では DNS の TTL に従って随時 NKCT を更新し，オフライン状態では NKCT 自身の有効期間を用いて署名の有効性を判断し，TTL を無視する．一方，NKCT に含まれる RR の TTL に応じて，NKCT の有効期限と独立に，更新が望まれるタイミングが決まる．以下，NKCT の有効期限，DNS の TTL，想定される更新頻度について，それぞれ説明する．

NKCT 自身の有効期限は，NKCT を構成する署名のなかで，最も早く有効期限に到達する期限である．DNSSEC における署名レコードには，署名の有効期限が絶対時間で指定されているので，NKCT の有効期限も絶対時間となる．

通常の DNS の運用において，DNS のゾーン管理者は，TTL の指示により，RR がリゾルバのキャッシュに保存される期間を制御できる．DNSSEC の場合，鍵と署名の更新タイミングを制御するパラメータとなる．TTL の値は，権威サーバから DNS キャッシングリゾルバが RR を取り込んだ時点からカウントダウンされる相対的な時間を示し，キャッシングリゾルバがキャッシュから RR を消去するタイミングを DNS 権威サーバが指示する．レコードの更新頻度が高いと想定される場合は，ゾーン管理者は短い TTL を設定し，RR キャッシュに保持された RR を頻繁に無効化することで，RR の権威サーバからの取得頻度を高める．DNSSEC の運用について書かれた RFC4641 [21] では，署名の更新が行われた際の置き換えを迅速にするため，TTL を署名の有効期間の数分の 1 以下に設定することを推奨している．たとえば，親ゾーンから子ゾーンに対する DNSSEC における証明書に相当するのは DS レコードであるが，この有効期間は，文献 [22] によれば，最低数日から数週間とされている．実際，すでに署名されている TLD である.com の DS レコードの署名には，7 日強の有効期間と，2 日間強の TTL が設定されている[*1]．

NKCT は，有効期間に達する前に更新する必要がある．DNSSEC の鍵管理者の視点では，TTL で示されている期間は，仮に新しい鍵が用意されたとしても，クライアントが更新しなくても問題ない期間を示している．すなわち，TTL で示される期間においては，手元に保存された NKCT は運用上問題なく使うことができる．仮に，ガイドラインに従っている現在の.com の DS のように，有効期限が 7 日程度，TTL が 2 日間である場合は，NKCT は 2 日に 1 度更新する必要がある．なお，RFC5011 [19] の Active Refresh 仕様（Section 2.3）に従った鍵更新をする場合は，仕様に従い，NKCT の有効期間は短く，かつ，TTL をより少なくなるよう算定することになる．これに従った場合は，オフライン中の有効期間も半減し，更新頻度は倍加する．

3.3.6 本提案方式の活用方法とユーザから見た操作

本提案方式を用いれば，立証者が特定の識別名（FQDN）を持つノードであることを検証できるとともに，同じ FQDN に結び付いた鍵の所有者であることを示せる．同時に，FQDN は階層構造を持っていることから，特定のドメインに属したノードであることも明らかとなる．このことから，検証者は FQDN や，その一部を用い，自身の持つアクセス制御の対象となっている資源への立証者からのアクセス制御に利用できる．なお，本提案では，オフライン状態での検証を目的としているが，NKCT の準備がオンライン状態で行われている限り，提案方式の検証自体はオンライン状態で行われても成立する．

たとえば，得られたノード名を画面に表示し，判断をユーザに促す．ユーザは当該 FQDN が実際に通信しようとし

[*1] 2011 年 6 月 24 日現在

ている相手のものであることを確認して，許可するかどうかを判断する．1 度，許可した対象ノードの FQDN を保存しておけば，認証後に自動的にアクセス許可するような方式もとれる．一方，人手を介さないドメイン単位のアクセス制御も可能である．FQDN はドメイン名であるので，特定のドメインのノードすべてに対して許可するような設定が可能である．たとえば，図 4 でのノード `n2.bbb.net` は，`aaa.net` ドメインに属するノードすべてに対して許可を与えるような設定が可能である．このアクセス制御は，検証者が属するドメイン以外のドメインに対しても適用できる．

3.3.7 本提案方式のセッション確立プロトコルへの組み込み

クライアント–サーバ間（立証者–検証者）のセッション確立へ本提案方式を組み込む方法を説明する．

LCT を用意し送受する手順によって，3 通りの方法が考えられる．

(1) セッション確立プロトコルへの組み込み
(2) LCT を受け渡し，検証後に，NK を用いてセッションを確立
(3) サーバへのアクセス時にクライアントへ LCT を要求し，検証する

まず，サービスのためのセッション確立のプロトコルに LCT の送受と検証を組み込む方法がある．これには，通信回数を減らせるというメリットがある．理想的には，立証者に検証者から渡されるなんらかの情報を nonce [23] として用いる．この場合，LCT での鍵の所有を示す情報，すなわち *A-RR* に対する署名の代わりに，渡された nonce に対する署名を用いることができる．通信回数を含め，最適な方法といえるが，セッション確立部分のプロトコルに変更が必要である．

第 2 の方法として，3.3 節で説明した手順に従って NK の検証を済ませた後に，NK を用いてセッションを確立する方法がある．この方法は，セッション確立部分の変更が必要ないことに対して，セッション確立までのノード間の通信回数が多いという欠点がある．

第 3 の方法として，接続を受けたサーバが，接続を受けたときに，クライアントに対し逆向きに接続し，LCT を要求する方法がある．クライアントは，その場で LCT を作成して返答する．

5 章では，携帯端末向けに上記 (2) の方法で実装し，検証者が LCT を入手し検証した後に，セッション確立を模したチャレンジレスポンスによる判定を行った．また，サーバ型の実装として，上記 (3) の方法を，LCT による検証機能を Unix 系 OS の PAM（Pluggable Authentication Modules）[24] により実装した．

4. 関連研究と関連標準

その場で構成されるネットワークとして，アドホックネットワークという呼称が用いられているが，この言葉で表現される研究領域は複数あり，それぞれ異なった特性を持つ．本提案に近い領域の研究として，以下の 3 点があげられる．

- 本論文が対象とするような，一時的（impromptu）ネットワーク接続
- モバイルアドホックネットワーク（MANET）
- ワイアレスセンサネットワーク（WSN）

以下，本章では，アドホックネットワークとは，上記 3 種の領域のいずれかに属するものとする．これらは，対象となるネットワークやノードの特性，求められる条件は異なるが，一定の共通点を持つ．認証方式は，共通のトピックであるが，MANET や WSN における研究が特に活発である．実現する方式としては，その場にあるノード間の通信経路（インバンド通信）を用いた方式と，ノード間通信経路以外（アウトオブバンド通信）を用いる方式があるが，ここでは明記しない限りインバンド通信のものを対象として説明する．以下では，ノード間の信頼構築に関連したもので，本提案に関連が深いものを，

(1) サーベイ
(2) 公開鍵証明書を用いる認証方式
(3) アドホックネットワークネットワーク上で鍵を生成するもの
(4) アウトオブバンド通信を用いる認証方式
(5) DNSSEC 運用
(6) 信頼点管理

の 6 種類に分類して説明する．

認証に関連するサーベイとして，van der Merwe らによる MANET の鍵管理についてのもの [25]，Miltchev らによる分散ファイルシステムに対するアクセス管理についてのもの [26]，Wang らによる WSN でのセキュリティ全般についてのものがある [27]．

公開鍵証明書を用いた認証方式では，2 章で説明したように，検証者が信頼している公開鍵（すなわち，信頼点）を起点として，複数の証明書による鍵の信頼の連鎖をたぐりながら，対象となる公開鍵証明書の検証を試みる．そのため，証明書の配付・取得のメカニズムが必要で，そこに差異がある．アドホックネットワークにおける証明書配布の方式は，大きく分けて，自立型と権威ノードを持つ方法に二分される．アプリケーションにより，向き不向きもあるため，どちらが決定的に有効とはいえない．たとえば，自立型の場合は，耐障害性に優れるが，軍事利用等におけるアクセス管理という視点では，権威ノード方式による一括管理が求められている．今回の課題では，権威ノードに頼ることができない．したがって，自立型の研究を中心に

説明する．Capkun らは，MANET における公開鍵暗号管理方式として，自立型の公開鍵検証基盤を提案した [28]．ノードそれぞれが PGP [12] に近い方法で証明書を独自に発行し，近隣ノード間で相互に交換することで，信頼の連鎖を確立する．近隣ノード間での単純な交換でない工夫を持たせる研究として，配布にルーティング情報を活用したもの [29] と，さらに重み付き有向グラフを組み合わせた方式 [30]，ノード間の信頼関係を活用した方式 [31] 等がある．加えて，完全な分散型でなく，アドホックネットワーク上の証明書管理ノードによって代行管理を行う方法 [32] もある．これらの方式は，常時，自律型ネットワークにノードが置かれていることを仮定している．一方，提案方式では，ノードがオンラインあるいはオフライン状態どちらかに置かれることを仮定し，オンライン時に自身の識別名を立証するために必要な NKCT を保持し，オフライン時にそれを活用している．さらに，検証・立証する相手に対しての直接通信を前提とし，検証者の信頼点を起点とした NKCT と，立証者の LCT を組み合わせて用いることで，検証に必要な証明書を揃えており，信頼できる第三者を必要としない点が異なる．

アドホックネットワーク上での鍵生成や認証スキームに工夫を持たせた研究としては，生成自体を ID スキームを用いて分散する手法 [33] や ID スキームに閾値暗号を組み合わせた手法 [34] がある．これらに対し，本提案方式は準備時において鍵を事前に作成し，NKCT として検証可能な状態で保存している点が異なる．

本提案と同等のユースケースを対象とした提案としたアウトオブバンド通信を用いる方式として，パスワードに基づいたグループ鍵承認機構 [35] がある．この提案では，共通の秘密としてパスワードを用いてグループ鍵を生成する方法を用いている．また，携帯電話と Bluetooth ヘッドセットのような携帯デバイス間のペアリングのための認証方式として，パス・キーの手入力 [1]，カメラを用いた視覚経由（visual channel）[3]，音声の活用 [4]，振動に対するボタン操作 [2] 等が標準化あるいは提案されている．Ion らは，これらの方式による携帯デバイスのペアリング操作の比較調査をしている [2]．これに対して，提案方式では，共通の秘密の共有や共通の秘密の入力が不要である．

本提案の目的との共通性はないが，DNSSEC を活用したゾーン管理方式として，窪田らによる DNSSEC ゾーンの自動構成による手法 [36] がある．局所的ネットワーク環境で，ドメインの管理と DNSSEC 鍵提供を DNSSEC を用いて行っている．この手法は，ゾーンに対する鍵の自動生成・管理であり，特定の管理ノードがゾーンの管理を行う．他に，DNSSEC の運用に関連した研究が発表されている [37]．本研究で関係する部分として，鍵の無効化についての議論がある．

信頼点を動的に管理するメカニズムとして，信頼サービス提供者（Trust Service Provider – 以下 TSP）のリストを提供する ETSI TS 102 231 V3.1.2 [38] がある．信頼サービスの提供場所と運用状況の一覧（以下 TSL）の提供形式が定義されており，このリストを更新でき，TSP にアクセスできる限り，信頼点として用いることが可能である．TSP として，公開鍵認証局や OCSP サーバだけでなく，キーエスクロー等も対象となっている．TSL の提供は管理のうえではメリットがある．しかし，このサービスはオンラインでのサービス提供を前提としているので，仮に検証者が TSL を持っていたとしても，オフライン状態では，立証者が提示した情報を確認できるように信頼の連鎖を確保できるとは限らない．一方，提案方式では，検証者の持つ信頼点のリストに頼るのではなく，立証者・検証者双方の信頼の連鎖を組み合わせており，第三者に依存しない．

5. 実装と実験

本提案の性能の視点での評価のため，本提案の署名と検証機能を実現したライブラリを実装した．さらに，具体的な動作を示すため，相手認証プログラムの携帯端末（iPhone）向け実装と Unix サーバでの認証機構への組み込みを試みた．

5.1 ライブラリの実装

ライブラリは，提案手法を忠実に実装しており，3.3 節で説明した手順に従って利用できる．必要なゾーン情報，信頼点の公開鍵と FQDN，自身の秘密鍵を設定後，署名と検証の 2 つの操作が可能となる．ライブラリは，検証に必要な情報を，ネットワーク経由でも，ファイル経由でも受け取れるように作成した．LCT の表現形式は，テキスト形式で，DNS RR の Presentation Format（RFC 1035 5.1 節 [5]）による RR 表現の並びとした．

5.2 携帯端末での実験実装

ユーザがどのような形で用いるのかを示すため，iPhone 上で動作するアプリケーション（NearBuddy と呼ぶ）を作成した．署名対象の RR は，ノードが得て利用している IPv4 アドレス（*A-RR*）とし，検証完了後に，実際の通信セッション確立の代わりに，立証者に対して乱数値を渡し，秘密鍵での署名の返答を検証するチャレンジレスポンスで確認した．実験条件を**表 2** に，表示例を**図 6** に示す．具体的な動作は以下のとおり．

ノード識別名の設定 NearBuddy は，実験のため，事前に用意した複数のノード鍵とゾーンがあらかじめ設定されており，用いるノード鍵を選べる．これは識別名の選択と同じ意味である．

検証情報提供機能 立証者は検証者に対し立証するための情報を提供するため，サーバとして動作し，あるポートで接続を待つ．検証に必要な情報は HTTP で検証者に渡す．追

表 2 実験設定
Table 2 Experimental configuration.

項目	説明
OS	iOS 4.2, MacOS X 10.6.5
言語	C++, Objective-C
ライブラリ	OpenSSL 1.0.0a（armv6 – C 言語のみ．アセンブラ不使用）
鍵生成	dnssec-keygen（9.7.2-P2）
署名	dnssec-signzone（9.7.2-P2）
アルゴリズム	RSA / SHA-256

図 6 NearBuddy ブラウザの画面
Fig. 6 NearBuddy browser screen.

加 RR として，接続時に用いている IP アドレスで *A-RR*，および，ノードを示すユーザ登録可能な名称を *TXT-RR* として用意する．*TXT-RR* は，ある程度自由に文字列を登録できる DNS RR である．様々な形で利用されているため，他のプロトコルと使用方法で混乱がない限り自由に使うことが可能である．これらの RRset を作成・署名し，あわせて提供する．

ノード検索と検証 iPhone では，Multicast DNS の実装である Bonjour [39] によって同一セグメント上に存在するノードで動作しているサービスを広告・発見できる．これを利用し，NearBuddy を実行しているノードを発見できる．NearBuddy を実行すると，それぞれのノードは，Bonjour で NearBuddy のサービスを広告する．並行して，NearBuddy サービスを実行しているノードを発見する．NearBuddy サービスを実行しているノードそれぞれから，検証に必要な情報を得た後，識別名とノード鍵で署名されたローカルアドレス（*A-RR*）を検証する．続けて，公開鍵暗号によるセッション確立を模擬する．検証者は乱数を生成し，立証者に送信する．立証者は受け取った乱数を NK 秘密鍵で署名し，署名を返信する．検証者は署名が当該立証者の NK によるものであることを公開鍵で確認する．検証した結果を FQDN によるノード識別名と *TXT-RR* に納められたユーザ登録の名称とともに一覧表示する．このとき，検証できた相手を緑の丸，検証できなかった相手を赤の丸で表示する．

信頼点の変更 NearBuddy は，実験用であるため，事前に用意された一覧から信頼点を選ぶことができる．選択した信頼点と，立証者から得た信頼の連鎖に交差がない場合は，認証に失敗するので，認証が失敗する状態を確認できる．

5.3 認証モジュールによる実装

LCT による検証機能を，Unix 系 OS の標準的認証フレームワークである PAM（Pluggable Authentication Modules）[24] に組み込んで実験した．

PAM は，サーバ実装で必要となる認証処理を抽象化した API として実装されている．サーバ側を PAM のライブラリを用いて認証するようにするだけで，PAM のフレームワークに従って用意された認証モジュールを自由に組み合わせることができる．

実験では，PAM のモジュールとして，認証時に，クライアント（立証者）に対して HTTP 経由で LCT を要求し検証する `pam_lct` を作成した．クライアント側には，LCT をその場で生成するウェブサーバ（LCT サーバ）を用意した．接続を受けたサーバ（検証者）は，同一サーバ上で動作する `pam_lct` に認証を要求する．`pam_lct` は，対象クライアントのアドレスを知りうるので，このアドレスの LCT サーバに対して LCT を要求する．クライアント上の LCT サーバは，立証のため，その場で LCT を作成して返答する．`pam_lct` は，得られた LCT を検証し，認証結果をサーバプログラムに返す．

PAM を実装している UNIX では，多くのサーバプログラムが PAM を用いて認証できるので，PAM を実装するだけで，PAM を利用できるサーバすべてで利用できる．実験では FTP サーバを用いて動作を確認した．

6. 評価

本提案を，課題への対応，計算回数と検証時間の計測，NKCT/LCT の情報量と想定される通信量，DNSSEC 適用の妥当性，セキュリティ上の課題，それぞれについて評価し，最後に今後の応用について議論する．

6.1 課題への対応

提案方式によって提示した課題を解決していることを確認する．

まず，アドホックネットワーク環境，すなわち，信頼できる第三者が存在しない状況での相手ノード識別名確認手法として有効である．識別名によって接続対象ノードを識別でき，かつ，識別名に紐付いた公開鍵を入手し，対応する秘密鍵の保有を署名により検証できるので，信頼できる

セッションを確立できる．そして，立証者の FQDN を識別名として検証込みで入手できるので，識別名に基づいて許可を判断できる．

アウトオブバンド通信に頼らずに，立証者・検証者間での信頼関係を確立できる．アウトオブバンド通信による方式と比べ，操作は容易であり，識別名（FQDN）に対するパターンマッチによれば，人による操作が不要となるため，操作の確実性が高い．たとえば，識別名を表示し，目視確認後にボタンをクリックするだけの操作で許可を指示できるので，携帯電話と Bluetooth ペアリングを例としたアウトオブバンド認証の調査でのパス・キー入力をともなって最低 15 秒という結果 [2] と比べると，明らかに時間もかからず容易である．ただし，アウトオブバンド認証，本提案方式のどちらの場合も，同一対象に対する検証が連続する場合は，認証結果に結び付けられたノードと，新たに認証が必要なノードとの間の同一性が確認できる範囲において，過去の認証結果を用いることが可能であり，認証を不要としたり，認証に必要な時間を短縮したりできる．一方，3.3.6 項にあるような方法でユーザに提示する際，ユーザが識別名を読み間違うリスクがあるが，ランダムな文字列より FQDN の方が可読性が高いので，有利である．

ただし，本提案では，立証者・検証者の信頼の連鎖が交差しない場合，認証できない．ここで，交差しない場合の可能性を見てみると，従来の DNSSEC の運用状況では，ルート以外の信頼点を選ぶ必要があったため，違う組織のノードどうしでの検証は，相互に相手の信頼点を含めて設定している場合を除き，不可能であったといえる．この場合，DNSSEC を運用している組織，あるいはグループ内で活用するのが限界であった．一方，現在では，DNS ルートゾーンが署名されたため [17]，DNSSEC の広がりに応じて，利用できる範囲の増加が見込める．

6.2 計算回数と検証時間の計測

サンプルとなる LCT を対象に，暗号アルゴリズムの実行回数について説明するとともに，実機測定した結果を示す．

DNSSEC では，検証にともなって RR をネットワーク経由で段階的に入手するが，本提案では，1 度の通信で RR が揃い，かつ，近隣からの通信であることから，通信時間は無視できる．この前提で，本提案の検証時間における最大コストは，当該 FQDN の信頼点から対象 RR までのすべての DNSSEC 署名検証とおおむね同等である．ここでは，比較的高性能なモバイルデバイスの 1 つだが，携帯型 PC 等と比較し計算資源の乏しい iPhone 3GS を用いて署名検証にかかる時間を測定し，利用に差し支えのない時間で終了することを検証した．信頼点には，最も検証に時間のかかるルートを選んでいる．

DNSSEC において，FQDN 全体を検証するとは，ルートから開始し，順に信頼の連鎖を確認することである．信頼の連鎖は，ドメインの管理委譲点ごとの鍵の署名によって構築されている（表 1 参照）．ドメイン管理委譲点とは，たとえば，`net` と `aaa.net` の間である．3 章での説明では，DNSSEC の仕様の詳細は説明を省いたが，DNSSEC でゾーンを署名する際は，ゾーン管理者は Key Signing Key（KSK）と Zone Signing Key（ZSK）の 2 つの鍵を用いる方式が推奨されている [21], [22]．KSK は鍵を署名するための鍵であり，ZSK はゾーンを署名する鍵である．親ゾーンで管理委譲先ゾーンの KSK に対して署名し（実際は *DS-RR* に対しての署名であるが，ここでは説明を省く），ゾーン管理者は ZSK を KSK で署名し，実際のゾーンの情報への署名は ZSK を用いる．ZSK/KSK の 2 つの鍵を用いるのは，親ゾーンへの鍵署名要求のコストが高いため，2 段階にすることで，ZSK の交換をゾーン管理者のみで行えるようにするためである [6]．このため，ドメイン管理委譲点での検証は，ZSK/KSK をセットで検証する．そして，検証時はノードに設定されている信頼点を起点に検証する．たとえば，`n1.aaa.net` という FQDN を持つノードの公開鍵を検証するには，合計 3 個のゾーンで鍵の検証を行う必要があるため，6 回の署名検証が必要である．本提案での追加操作として，ノードの FQDN 位置に収められた NK 公開鍵（*DNSKEY-RR*）の検証の後，立証者で追加のあった RRset それぞれに対し，ノード鍵を用いた署名検証が必要である．つまり，例にあげた FQDN 名に紐づけられた単一の *A-RR* の検証には合計 7 回の署名検証が必要となる．1 つの署名（*RRSIG-RR*）の検証は，署名対象データに対して，ハッシュアルゴリズムの実行を 1 度，公開鍵暗号の検証処理が 1 度必要である．一方，ノード鍵を用いて，RRset を署名するときは，ハッシュアルゴリズムの実行と，公開鍵暗号による署名処理が 1 度ずつ必要である．

本提案方式による検証ライブラリを用いて，`n1.aaa.net` のノード鍵を用いた *A-RR* に対する署名と，信頼点から *A-RR* までの検証を iPhone 3GS で実行した結果を，**表 3** に示す．表には，参考のため，本提案プロトコルによる署名操作と検証操作，それぞれの暗号アルゴリズムが実行される回数を記してある．NearBuddy に実装したベンチマークプログラムは，本提案による署名および検証にかかる時間を，指定回実行した実行時間で測定できる．それぞれ，繰返し回数を 100 回に設定し，10 回計測した後に平均をとっている．公開鍵暗号アルゴリズムは RSA，ハッシュアルゴリズムは SHA-256 を用い，鍵長はルートゾーンと TLD（.net）は 2,048 ビット，それ以外は 1,024 ビットとした．鍵長とアルゴリズム選択は文献 [22] を参考にした．結果は，署名に 35.4 ミリ秒，検証に 42.1 ミリ秒と，比較的長めの鍵を用いつつも，十分実用的な速度で動作していると結論できる．また，セッション確立の模擬のためのコストは，検証に 2.6 ミリ秒かかっている．署名は LCT 作成コストと同一である．なお，署名は，検証ライブラリの

表 3 提案方式の iPhone 3GS 上での処理速度
Table 3 Benchmark of the proposed scheme on an iPhone 3GS.

Operation	Duration	SHA-256-ops	RSA-sign-ops	RSA-verify-ops
LCT Signing	35.4 msec ± 0.46%	1	1	-
LCT Verification	42.1 msec ± 0.24%	7	-	7
Single NK Verification	2.6 msec ± 0.82%	1	-	1

表 4 本提案方式による n1.aaa.net 公開鍵検証に必要なデータ量試算
Table 4 Estimated amount of data need to verify the public key of n1.aaa.net, by the proposed scheme.

Label	*DNSKEY-RR*		*RRSIG-RR*		Data Size
	2,048 bits	1,024 bits	2,048 bits	1,024 bits	per label (bytes)
.	2		1		830
net.	2		2		1,183
aaa.net.		2	1	1	835
n1.aaa.net.		1		1	366
(LCT signature)				1	177
Total size of the LCT for n1.aaa.net					3,391

基礎となっている OpenSSL を用いているが，iPhone のプロセッサである ARM プロセッサ用にはアセンブラによって最適化されたコードがないため，C 言語の記述によっている．iOS SDK の暗号ライブラリを用いるようにすれば，所要時間の短縮を見込める．

6.3 NKCT/LCT の情報量と想定される通信量

NKCT/LCT の情報量について，記憶域の占有量，通信帯域の消費，通信頻度の点で検討する．検討材料として，表 1 で示した `n1.aaa.net` という FQDN を持つノードの LCT を構成する RR を，DNS プロトコルで用いられているバイナリ形式（いわゆる，Wire-format）[5], [7] で表現した場合のデータ量を**表 4** に示す．なお，表からは *DS-RR*, *A-RR* の個数と大きさの表記を省いているが，表での各項目の大きさは完全な LCT と一致する．

認証に用いられる情報の大きさとして，3,391 バイトという情報は小さいとはいえないが，対象領域においては問題とならない．たとえば，ワイアレスセンサネットワーク等において，記憶域の小さいノードに収めるには大きく，帯域が狭いネットワーク環境で用いるには大きすぎるといえる．しかし，本提案で対象としている，携帯型 PC やスマートフォンでは，3 K バイトという情報量は，問題にならない．そして，本提案で対象としている Bluetooth のようなノード間の局所的直接通信においては，攻撃を受けるような状況等でないかぎり，全体の帯域を使いつくすほど認証が短時間に繰り返して行われる状況は考えられない．

通信頻度という点では，準備段階での NKCT を作成する際の通信は，オンライン状態における通常の DNS リゾルバによる通信であり，通常の DNS 通信の範囲である．したがって，オンライン状態で，3.3.5 項で示したような，数日に 1 度という頻度では，問題がない．立証・検証時である，オフライン状態で NKCT から LCT を作成するときは，必要な情報を取得するための通信はいっさい行わない．LCT の通信量は，立証が必要な際に検証者からの求めに応じて送信するだけであり，認証要求の回数に一致する．なお，3.3.5 項で説明したように，RFC5011 の Active Refresh が併用される場合は，更新頻度は倍増する．しかし，Active Refresh は，更新を早めること自体が目的であるので，結果的に更新頻度が増すのはトレードオフである．

これらのことから，NKCT の大きさは，記憶域の占有量の点でも，通信帯域の消費のうえでも，実用上問題ない．

6.4 DNSSEC 適用の妥当性

信頼関係の確立のために，DNSSEC 関連 RR の代わりに，X.509 証明書を用いても実現可能である．そこで，X.509 PKI を用いた場合と本方式による DNSSEC の利用を，署名検証の計算コスト，鍵証明書発行（鍵署名）の経済性，鍵証明書発行（鍵署名）の運用性，鍵の無効化の 4 点で議論する．

本提案の署名検証の計算コストは X.509 PKI を使う場合と比して高くなる．ゾーン間の信頼の連鎖の 1 つを確認するためには，ZSK/KSK の 2 つの鍵を確認する必要があり，信頼の連鎖を 1 つ確認するために署名の確認が 1 つで済む X.509 PKI と比べると署名の確認回数が倍になり，不利である．一方，署名のための計算は，計算量が多いが，そのつど行うしか方法がない．X.509 PKI と比べ署名の確認回数は倍になっており，必要な CPU 処理能力や消費や電力消費の点では不利である．一方，速度の点で見ると，6.2 節で実験した検証に必要な時間，DNSSEC を用いた場合の 42.1 ミリ秒が X.509 PKI の場合，概算で 24.1 ミリ秒

になるにすぎない（署名検証 7 回のうち 3 回分が減るため，計算時間は $42.1 * (7-3)/7 = 24.1$）．時間あたりの検証回数に換算すると，X.509 PKI では毎秒約 41.5 回検証できるのに対して，DNSSEC による検証は毎秒約 23.8 回となる．ここで，認証操作はユーザの操作に応じた程度の頻度と想定できるので，この差によって大きく不利になるとは考えられない．

DNSSEC の鍵署名による証明は，X.509 PKI の証明書発行コストと比べ，経済的コストが比較的低い．X.509 のエンドエンティティ証明書を効率良く活用するには，オペレーティングシステムに組み込み済みの複数の公開鍵証明書を信頼する．この場合，これら複数の証明書のうちのいずれかをを起点として検証可能な，エンドエンティティ証明書の発行を認証局運用会社から受ける必要がある．ある認証局運用会社の例をあげると，サーバ証明書の発行料は，年間・単一の証明書あたりで，数万円から 10 万円程度でサービスされている．エンドエンティティ証明書をこれらの企業から発行を受けるためには，相応の規模の予算を確保しなければならず，単一の証明書の発行が高額であるだけでなく，運用する証明書の数も多くなるので，コストが高い．適用対象のアプリケーション（本提案）に対し，証明書の強度（これら企業が提供するサービスの信頼性）が高いため，アドホックノード認証に用いるのにはオーバスペックである．これと比べ，DNSSEC を用いた方法は，秘密鍵の保持方法等，運用方法によって運用コストに差が出るものの，通常の DNSSEC の運用と同等の運用形態となることが想定できる．したがって，X.509 エンドエンティティ証明書の活用に比べ，経済的に低コストで運用できる．

DNSSEC は，X.509 PKI を用いた場合と比べ，鍵署名を人手を介さずにするための工夫があり，運用性が高い．2 者の大きな違いは，DNSSEC が実際の運用性を強く意識して設計されている点である．ここでの運用性は，いかに人手を煩わせずに運用できるかという点であり，運用のうえで大きなウェイトを占めているのが鍵の管理である．ここでは，鍵管理における差異について議論する．あるゾーンを親とし，その親から管理委譲を受けた子ゾーンに注目すると，親ゾーンから子ゾーンへの信頼関係の構築は，親ゾーンへの署名や登録といった操作要求の必要性から，比較的コストが高い．それゆえ，更新頻度は極力減らすことが望ましい．一方，子ゾーン内に対する RR の更新とそれにともなう署名は，運用形態によるものの，少なくともゾーンの親子間の情報の更新頻度よりも高いと想定できる．仮に，単一の鍵によってゾーンの管理と，上位ゾーンからの信頼の連鎖を受ける形で構成しようとすると，頻度とコストの関係がアンバランスになる．そこで，強度や運用形態のバランスを調整するために，DNSSEC では，ゾーン署名のための鍵（ZSK）と鍵の署名のための鍵（KSK）を使い分けることが推奨されている [21], [22]．たとえば，RFC4641 [21] では，KSK では高めの強度の鍵を比較的長期間運用し，ZSK には低めの強度の鍵を短期間で切り替える運用が推奨されている．さらに，KSK 秘密鍵は常時ネットワークから切り離された場所で管理する一方，ZSK の秘密鍵はサーバ上に置いて，ゾーンの署名を自動化する手法がとれる．X.509 PKI の場合，このような運用を単一の認証局では実現できないため，強度の異なる認証局を運用する等の工夫が必要であり，運用性は DNSSEC の方が高い．

DNSSEC の鍵の無効化は X.509 PKI の方式と比較し，提案の対象とするユースケースにより適合している．DNSSEC の鍵の無効化の管理については，鍵ではなく，それぞれの署名に有効期限が設定されることに特徴がある．鍵に対する署名が有効期限に到達し，同じ鍵を用いた期限延長に相当する署名がない限り，鍵に対する署名が自動的に失効する．なお，RFC5011 [19] では，より明示的な鍵の無効化（revoke）と置き換え（rollover）のプロトコルが定義されている．鍵の失効手続きにおいて，無効な鍵すべての一覧を保持しなければならない X.509 PKI での CRL や OCSP による方法と比べ，無効化の即時性が低いというデメリットはあるが，運用が容易である．

6.5 セキュリティ上の課題

提案方式には，鍵の失効をアドホックネットワークで判定できない，匿名性が失われる，DoS に類する攻撃への耐性の 3 点でセキュリティ上の課題がある．

本提案方式は，NKCT 保存後の有効期間中に，オンライン環境で NKCT のいずれかの鍵が無効化あるいは交換され，保存された NKCT とオンラインの情報に齟齬が生じたとき，オンライン環境では無効であるにもかかわらず，保存された NKCT による検証が成功してしまう問題がある（NKCT と LCT の有効期間については，3.3.2 項手順❺，3.3.3 項手順❽および 3.3.5 項参照）．これは，アドホックネットワークのように，インターネットへの到達性がなく，最新の情報にアクセスできない以上避けられない問題である．たとえば，X.509 PKI は，鍵の有効期限の判定のほか，インターネットへの到達性がある限り，CRL や OCSP を用いて鍵の失効を判定できる．DNSSEC は，2 つの方式により，一定の時間内に鍵や署名の失効を検知できる．1 つ目の方式は，署名が TTL に従って RR キャッシュから削除されるのにともなう無効化である．2 つ目の方式は，鍵関連 RR の更新タイミングの調整と，revoke ビットを立てた鍵に対する自己署名による鍵の無効化によるものであり，RFC5011 [19] で定義されている．X.509 PKI も DNSSEC も，オフライン環境でのでの運用は前提としていないので，鍵や署名の期限切れによる時限的処理による無効化は可能であるが，能動的な情報取得が必要な突発的な鍵の無効化には対応できない．本提案固有の問題ではないので，リスクを許容できる範囲で，有効性とのトレード

オフを検討のうえ利用すべきである．

提案方式には，ノードの匿名性が失われる問題がある．本提案の設計上，立証者は，検証を求めたすべてのノードに対し LCT を引き渡さなければ認証は成り立たない．しかし，LCT にはノード自身の持つ FQDN を示す情報が含まれており，LCT をアクセスコントロールなしに渡すと，無条件に FQDN を明らかにすることになる．さらに，ノード自身の持つ MAC アドレス等の情報と FQDN が組み合わされるというリスクもある．見過ごせないリスクだが，本提案のメリットとのトレードオフとして考えることは可能である．なお，3.3.7 項での説明のようにセッションを確立する際に，LCT 受け渡しを工夫すれば，リスクを低減させることができる．

ノードに検証情報が要求される際，処理のつど，CPU 計算量の多い署名処理を行う場合，連続してサービスを要求する DoS 攻撃のリスクがある．リクエストに対するフィルタリングやレートコントロール，計算結果を適切に保持しておく方法で対策は可能である．

6.6 今後の応用

提案方式は，特に，ネットワーク接続のない環境において，信頼できる第三者に頼らずにノード間で認証する方式であるが，ネットワーク接続がある場合においても，信頼できる第三者や集中型の管理ノードに頼れないアプリケーションに適用可能である．たとえば，P2P 方式のノードに適用すれば，それぞれノードが認証可能な識別名を示し，かつ公開鍵を提供できる．また，GRID システム等の，ノード数が多く，管理ノードに頼ることが難しい環境においても，活用が可能である．

また，今回の実装では統合しなかったが，Bonjour のような，Multicast DNS [40] を用いたサービスと統合して活用することが容易である．Multicast DNS は，DNS 権威サーバに頼らずに，マルチキャストドメイン中で DNS 名前解決をするためのメカニズムである．Bonjour は Multicast DNS を活用し，DHCP サーバがなく RFC3927 [41] によって IPv4 アドレスが自動的に割り当てられるような状態であっても，そのノードが提供するサービスを近隣ノードに対して広告し，発見できる．たとえば，MacOS X の例であれば，プリンタ，Web サーバ，音楽サーバ等を発見できる．そして，リンクローカルアドレスに対して `.local` をトップドメインとする FQDN が割り当てられる．本提案による方式を統合することで，`.local` ドメインを用いずに，そのノードが実際に属する実際の FQDN を広告し，それを検証して活用できる．DNS は，ネットワークに関連した情報の提供基盤であるが，DNS の拡張である DNSSEC を用いることで，DNS RR として提供するタイプの情報は容易に提供できる．

さらに，本提案は，広義のアドホックネットワークノードへの応用が可能と考えられる．ただし，この応用については，計算量・通信量等を含め，広い範囲の検討が必要であり，本論文では，その可能性を指摘することにとどめ，今後の研究課題としたい．

7. まとめ

本論文では，アドホックネットワーク上のような信頼できる第三者が存在しない状態での通信相手認証を，公開鍵暗号を用い事前の秘密共有を必要とせず最小限の信頼点設定のみで実現する手法を提案した．

従来のノード間認証提案では，アウトオブバンド通信に頼る方法や，事前の秘密共有に頼る方法等，アドホックネットワーク上で用いるには制約があった．一方，公開鍵暗号を用いる方式は，ノードが持つ鍵と識別名の結び付きを明らかにするために，公開鍵基盤の利用が必要であるが，検証者が信頼する信頼点から立証者が持つ公開鍵に至る信頼の連鎖をアドホックネットワーク上で入手できないため，応用において制約があった．

本提案方式では，ノードの公開鍵の識別名との結び付きを立証するための情報を立証者自身が保存し，必要に応じて検証者に渡し，検証者は，それ自身が持つ信頼の連鎖と組み合わせることで検証をインバンド通信のみで可能とした．加えて，本手法で必要な公開鍵の信頼の連鎖を構成する方式として，コスト，運用性において X.509 PKI と比べて有利な DNSSEC のリソースレコードの活用を提案した．

本提案の実現性を示すために，提案方式を実現したライブラリを作成した．具体的な動作を示すため，スマートフォン上で動作する検証アプリケーションを作成すると同時に，検証に要する時間を確認した．加えて，サーバプログラム側への具体的な組み込み例として，Unix 系 OS で用いられている PAM として実装し，動作確認をした．評価では，課題への対応，計算回数と検証時間の計測，必要な情報量と通信量，DNSSEC 適用の妥当性，セキュリティそれぞれについて取り上げ，本方式の実際の運用が十分可能であることを示した．

アドホックネットワーク環境での一時的な通信というシナリオでは，そのノードを示す識別名と，識別名に結び付いた公開鍵ペアをノードが保持していることを検証できれば，認証や，セッション確立に用いることができる．提案手法は，既存のアウトオブバンド通信や事前秘密共有に頼る方式と比べ，運用性，操作性に優れており，アドホックネットワークの高度な利用が可能となる．

参考文献

[1] Bluetooth SIG: Bluetooth Specification Version 4.0, Bluetooth SIG (online), available from ⟨http://www.bluetooth.org/⟩ (accessed 2011-10-07).

[2] Ion, I., Langheinrich, M., Kumaraguru, P. and Capkun, S.: Influence of user perception, security needs, and so-

cial factors on device pairing method choices, *Proc. 6th Symposium on Usable Privacy and Security* (*SOUPS '10*), ACM (online), DOI: 10.1145/1837110.1837118 (2010).

[3] McCune, J., Perrig, A. and Reiter, M.: Seeing-is-believing: using camera phones for human-verifiable authentication, *Proc. IEEE Symposium on Security and Privacy*, pp.110–124 (online), DOI: 10.1109/SP.2005.19 (2005).

[4] Goodrich, M., Sirivianos, M., Solis, J., Tsudik, G. and Uzun, E.: Loud and Clear: Human-Verifiable Authentication Based on Audio, *Proc. 26th IEEE International Conference on Distributed Computing Systems* (*ICDCS 2006*), pp.1–10 (online), DOI: 10.1109/ICDCS.2006.52 (2006).

[5] Mockapetris, P.: RFC 1035: Domain Names – Implementation and Specifications, Internet RFCs (1987).

[6] Arends, R., Austein, R., Larson, M., Massey, D. and Rose, S.: RFC 4033: DNS Security Introduction and Requirements, Internet RFCs (2005).

[7] Arends, R., Austein, R., Larson, M., Massey, D. and Rose, S.: RFC 4034: Resource Records for the DNS Security Extensions, Internet RFCs (2005).

[8] Lampson, B.W.: Computer security in the real world, *IEEE Computer*, Vol.37, No.6, pp.37–46 (online), DOI: 10.1109/MC.2004.17 (2004).

[9] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R. and Polk, T.: RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, Internet RFCs (2008).

[10] Adams, C. and Lloyd, S.: *Understanding PKI: Concepts, Standards, and Deployment Considerations, 2nd edition*, Addison-Wesley Longman Publishing Co., Inc. (2002).

[11] Ford, W. and Baum, M.S.: *Secure Electronic Commerce: Building the Infrastructure for Digital Signatures and Encryption*, Prentice Hall PTR (2000).

[12] Zimmermann, P.R.: *The official PGP user's guide*, MIT Press, Cambridge Mass. (1995).

[13] Cooper, M., Dzambasow, Y., Hesse, P., Joseph, S. and Nicholas, R.: RFC 4158: Internet X.509 Public Key Infrastructure: Certification Path Building (2005).

[14] Blaze, M., Feigenbaum, J. and Lacy, J.: Decentralized trust management, *Proc. IEEE Symposium on Security and Privacy* (online), DOI: 10.1109/SECPRI.1996.502679 (1996).

[15] ITU-T: Recommendation X.500: Information technology – Open Systems Interconnection – The Directory: Overview of Concepts, Models, and Services.

[16] Myers, M., Ankney, R., Malpani, A., Galperin, S. and Adams, C.: RFC 2560: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP, Internet RFCs (1999).

[17] ICANN and VeriSign Inc.: Root DNSSEC: Information about DNSSEC for the Root Zone, ICANN and VeriSign Inc. (online), available from ⟨http://www.root-dnssec.org/⟩ (accessed 2011-10-07).

[18] Atkins, D. and Austein, R.: RFC 3833: Threat Analysis of the Domain Name System (DNS) (2004).

[19] StJohns, M.: RFC 5011: Automated Updates of DNS Security (DNSSEC) Trust Anchors, Internet RFCs (2007).

[20] Mockapetris, P.: RFC 1034: Domain Names – Concepts and Facilities, Internet RFCs (1987).

[21] Kolkman, O.M. and Gieben, R.: RFC 4641: DNSSEC Operational Practices, Internet RFCs (2006).

[22] Kolkman, O.M. and Mekking, M.: DNSSEC Operational Practices, Version 2 (draft-ietf-dnsop-rfc4641bis-06), Internet Drafts (2011).

[23] Menezes, A., Oorschot, P.V. and Vanstone, S.: *Handbook of Applied Cryptography*, CRC Press (1997).

[24] Samar, V.: Unified login with pluggable authentication modules (PAM), *Proc. ACM Conference on Computer and Communications Security* (*CCS '96*), pp.1–10 (online), DOI: 10.1145/238168.238177 (1996).

[25] van der Merwe, J., Dawoud, D. and McDonald, S.: A survey on peer-to-peer key management for mobile ad hoc networks, *ACM Comput. Surv.*, Vol.39, No.1, pp.1–45 (online), DOI: 10.1145/1216370.1216371 (2007).

[26] Miltchev, S., Smith, J., Prevelakis, V., Keromytis, A. and Ioannidis, S.: Decentralized access control in distributed file systems, *ACM Comput. Surv.*, Vol.40, No.3, pp.10:1–28 (online), DOI: 10.1145/1380584.1380588 (2008).

[27] Wang, Y., Attebury, G. and Ramamurthy, B.: A survey of security issues in wireless sensor networks, *IEEE Communications Surveys & Tutorials*, Vol.8, No.2, pp.2–23 (online), DOI: 10.1109/COMST.2006.315852 (2006).

[28] Capkun, S., Buttyan, L. and Hubaux, J.-P.: Self-organized public-key management for mobile ad hoc networks, *IEEE Trans. Mobile Computing*, Vol.2, No.1, pp.52–64 (online), DOI: 10.1109/TMC.2003.1195151 (2003).

[29] 北田夕子，荒川 豊，竹森敬祐，渡邊 晃，笹瀬 巌：無線アドホックネットワークに適したルーチング情報を用いたオンデマンド公開鍵分散管理方式，電子情報通信学会論文誌，Vol.J88-D-I, No.10, pp.1571–1583 (2005).

[30] 安田侑八，毛利寿志，高田喜朗，関 浩之：アドホックネットワークにおける PKI 証明書連鎖発見問題について，情報処理学会研究報告，Vol.2006, No.120, pp.31–38 (2006).

[31] 川端秀明，石井啓之：モバイルアドホックネットワークにおける信頼関係リストを用いた公開鍵分散管理方式の提案，電子情報通信学会技術研究報告，Vol.IN2007-235, No.2008-03, pp.455–459 (2008).

[32] 船曳俊介，磯原隆将，北田夕子，竹森敬祐，笹瀬 巌：無線アドホックネットワークの公開鍵証明書管理における証明書管理ノード方式，情報処理学会論文誌，Vol.48, No.8, pp.2835–2845 (2007).

[33] Khalili, A., Katz, J. and Arbaugh, W.: Toward secure key distribution in truly ad-hoc networks, *Proc. Symposium on Applications and the Internet Workshops*, pp.342–346 (2003).

[34] Deng, H. and Agrawal, D.: TIDS: Threshold and identity-based security scheme for wireless ad hoc networks, *Ad Hoc Networks*, Vol.2, No.2004, pp.291–307, Elsevier (2004).

[35] Asokan, N. and Ginzboorg, P.: Key agreement in ad hoc networks, *Computer Communications*, Vol.23, No.17, pp.1627–1637 (online), DOI: 10.1016/S0140-3664(00)00249-8 (2000).

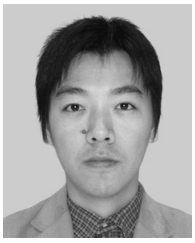
[36] 窪田 歩，三宅 優：DNSSEC ゾーンの自動生成によるパーソナルなネットワーク機器間のセキュア通信基盤の実現，電子情報通信学会技術研究報告，Vol.NS2009-170, No.2009-03, pp.153–158 (2009).

[37] Yang, H., Osterweil, E., Massey, D., Lu, S. and Zhang, L.: Deploying Cryptography in Internet-Scale Systems: A Case Study on DNSSEC, *IEEE Trans. Dependable and Secure Computing*, Vol.8, No.5, pp.656–669 (online), DOI: 10.1109/TDSC.2010.10 (2010).

[38] European Telecommunications Standards Institute (ETSI): Provision of harmonized Trust-service status information – ETSI TS 102 231 V3.1.2 (2009-12), European Telecommunications Standards Institute (online), available from ⟨http://www.etsi.org/deliver/etsi_ts/102200_102299/102231/03.01.02_60/ts_102231v030102p.pdf⟩ (accessed 2011-10-07).
[39] Apple Developer Support: Bonjour, Apple Inc. (online), available from ⟨http://www.apple.com/jp/support/bonjour/⟩ (accessed 2011-10-07).
[40] Cheshire, S. and Krochmal, M.: Multicast DNS (draft-cheshire-dnsext-multicastdns-11), Internet Drafts (2010).
[41] Cheshire, S., Aboba, B. and Guttman, E.: RFC 3927: Dynamic Configuration of IPv4 Link-Local Addresses, Internet RFCs (2005).

鈴木 茂哉 （正会員）

昭和60年より株式会社フォア・チューン代表取締役社長．平成11年放送大教養学部修了．平成11年から平成16年まで株式会社オムニサイソフトウエア代表取締役．平成16年よりAuto-ID ラボジャパン副所長．平成16年より平成17年まで慶應義塾大学SFC研究所研究員．平成17年より平成19年まで南カリフォルニア大学情報科学研究所（USC/ISI）訪問研究員．平成18年より慶應義塾大学大学院政策・メディア研究科助教．平成19年より同大学院政策・メディア研究科後期博士課程．平成22年より同大学ITC本部助教．



石原 知洋 （正会員）

平成13年日本大学理工学部物理学科卒業．平成15年慶應義塾大学大学院政策・メディア研究科修士課程修了．平成21年同後期博士課程修了．平成21年東京大学大学院総合文化研究科特任助教．平成22年慶應義塾大学大学院政策・メディア研究科博士（政策・メディア）．ドメインネームシステムおよびインターネットの運用技術に関する研究・開発に従事．

ビル マニング

昭和54年よりネットワーク関連研究に携わる．現在，南カリフォルニア大学情報科学研究所（USC/ISI）研究員．複数のRFCの著者であるとともにIETFワーキンググループチェアとして活動，現在も多くのICANN委員会に参加している．ルートDNSサーバを運用するメンバの一員である．

村井 純 （正会員）

昭和54年慶應義塾大学工学部数理工学科卒業．昭和56年同大大学院修士課程修了．昭和62年同大学院博士（工学）．昭和59年東京工業大学助手．昭和62年東京大学助手．平成2年慶應義塾大学助教授を経て平成9年より同大学教授．平成11年より平成17年まで慶應義塾大学SFC研究所所長．平成15年よりAuto-ID ラボジャパン所長．平成17年より平成21年まで慶應義塾常任理事兼慶應義塾大学環境情報学部教授．平成21年慶應義塾大学環境情報学部教授兼学部長．