アドホックネットワークの証明書管理ノード方式における認証妨害対策

與坂　宜士　　長瀬　智行　　竹花　洋次郎　　吉岡　良雄

弘前大学大学院理工学研究科
〒036-8560 青森県弘前市文京町 3
E-mail：nagase@eit.hirosaki-u.ac.jp

あらまし　見ず知らずのノードが参加する無線アドホックネットワークでは、公開鍵証明書によるノード認証は重要な技術となっている。公開鍵証明書を効率よく管理する方式として、証明書管理ノード方式が提案されている。しかし、この方式は悪意ある不正ノードが偽の証明書を発行し、特定のノードの認証を妨害することが出来るという問題がある。本研究では、証明書管理ノード方式に偽の証明書を含めた場合の認証成功確率を求め、偽の証明書による認証成功確率の降下を防ぐ方法を提案する。また、無駄な証明書のやり取りを省き、通信量の削減を目指す。そして、シミュレーションの結果から提案方式が有効であるかを検討する

Authentication and Certificate Managements of Unauthorized Intrusion in Ad-Hoc Networks; Problems and Solutions

Norihito Yosaka　Tomoyuki Nagase　Youjiro Takehana　Yoshio Yoshioka

Graduate School of Science and Technology, Hirosaki University
3 Bunkyo-cho, Hirosaki-shi, Aomori, 036-8560 Japan
E-mail：nagase@eit.hirosaki-u.ac.jp

Abstract　The security problem for mobile ad hoc networks is becoming one of the major issues that need to be considered.　The security of ad hoc networks is based on certification authority (CA) which issues and manages security certificates and public keys for message encryption. This paper introduces a new method of detecting unauthorized intrusion that is used by malicious user who issues a forged certificate to join a group of ad hoc network. This method also eliminates unnecessary certificates exchange among users' nodes to reduce traffic flow inside the network.

1. はじめに

今日の技術発展により、ノート PC や携帯電話など携帯型移動端末により無線通信が普及している。それに伴い、何時でも何処でも携帯型移動端末を用いて他の端末と通信したいという要求が高まっている。現在の携帯型移動端末での通信は無線 LAN が主に利用されているが、無線 LAN を用いて通信を行うにはアクセスポイントに接続する必要があり、アクセスポイントを設置するには、設置コストがかかる、設置に時間がかかる、地理的に設置できない場所では通信が出来ないなどの問題がある。このため、時間やコスト、地理環境に影響されないアドホックネットワークへの期待が高まっている[1]。

アドホックネットワークはマルチホップ通信を利用して基地局やアクセスポイントなどのインフラストラクチャに依存しない無線ネットワークであり、将来のモバイル通信の一形態として注目されている。この特徴を利用して、災害時の緊急通信手段への利用や、渋滞などの交通状況の情報を車両間を通じてやり取りを行う車両間ネットワークへの利用などが期待されている。アド

ホックネットワークではネットワークを構成する端末が移動することでネットワークのトポロジーが頻繁に変化する。また悪意あるユーザーと緊密に通信を行うことが考えられるため、パケット改ざんやなりすましを防ぐ、あるいは検出するセキュリティの仕組みが必要である。このようなセキュリティを確立するための方法として公開鍵暗号方式を用いたノード認証がある。

アドホックネットワークにおけるノード認証方式に公開鍵分散管理方式というものがある[2]。これはネットワークを構成するノードが独自の判断で別のノードの公開鍵に証明書を発行し、自身で信頼の輪を構築するというものである。しかしこの方式はネットワーク内の全ノードの証明書を集めるため通信量およびメモリ消費量が大きいという欠点がある。それを受けて、ネットワーク内に証明書の管理を担当する証明書管理ノードを設け、そのノードの電波範囲内にいる一般ノードが発行した公開鍵証明書を代行管理するという証明書管理ノード方式が提案されている[3]。これは公開鍵分散管理方式と比べてメモリ消費量および通信量を削減できるという利点がある。しかしこの証明書管理ノード方式はネットワーク内の全ノードが正しく動作することを前提としており、悪意あるノードの偽造証明書の発行による認証妨害を想定していない。

本研究では証明書管理ノード方式において、不正ノードの偽造証明書発行による認証への影響を検討し、偽造証明書を信頼の輪から排除できる方式を提案する。また、認証時の動作を変更し、通信量の削減を目指す。そして、シミュレーションから提案方式の有効性を示す。

2. 公開鍵分散管理方式

公開鍵分散管理方式とは、ネットワークを構成する各ノードが独自に証明書を発行し、リポジトリで管理する方式である。これは、発行した証明書を電波範囲内のノードと定期的に交換し合うことでネットワーク上の証明書を全て収集し、それを用いた信頼の輪の構築を通じて認証を行うものである。

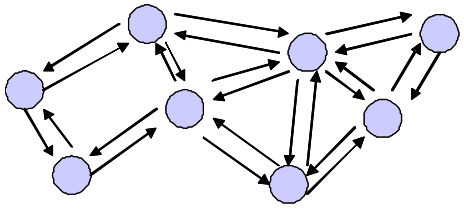

図１　公開鍵分散管理方式の概念図

信頼の輪(Web of trust)とは、ネットワークを構成するノードが信頼できる別のノードの公開鍵の正当性を保証する方式である。これはＡがＢを信頼し、ＣがＡを信頼している場合、ＣはＢを信頼できるという考えに基づいている。実際の処理は以下のようになる。

(1)ノードＡはノードＢの公開鍵を予め入手しており、自身の秘密鍵を用いてＢの公開鍵に署名を発行している（ＡはＢの公開鍵の正当性を保証している）

(2)ノードＣはノードＡの公開鍵を予め入手しており、自身の秘密鍵を用いてＡの公開鍵に署名を発行している（ＣはＡの公開鍵の正当性を保証している）

(3)ノードＣはノードＡからＡの署名が付加されたＢの公開鍵を入手する（この公開鍵が本当にＢの物であるかはまだ不明）

(4)ノードＣは自身が保持しているＡの公開鍵を用いて付加されているＡの署名を復号する

(5)復号が成功すればこの公開鍵は間違いなくＢのものであると判断できる（自身が信頼するノードＡがこの公開鍵はＢのものであると保証している）

公開鍵分散管理方式では、各ノードがネットワーク内の全ての証明書を集めるため、証明書収集が完了するまでに大きな時間を要する。また、ネットワーク内のノード数が多い場合、収集し管理する証明書の数が膨大になる。この場合、ネットワークに参加するノード全てに多大なメモリの消費を強いることになる。アドホックネットワークを構成する携帯型移動端末はメモリ量が制限されていることを想定する必要があるため、証明書の格納に必要なメモリ量は少ないことが望ましい。

3. 証明書管理ノード方式

3.1 証明書管理ノード方式の概要

証明書管理ノード方式は、公開鍵分散管理方式における証明書収集の時間短縮と証明書管理に必要なメモリ量の節約を目的として提案された証明書管理方式である。

この方式では、アドホックネットワーク内のいくつかのノードに、公開鍵証明書の管理を代行する証明書管理ノードを設け、電波が届く 1 ホップ以内のノードから発行された証明書のみを管理させる。そして一般ノードから認証要求を受けた時、他の証明書管理ノードに問い合わせを行い、証明書を収集し信頼の輪構築を行う。認証要求を行った一般ノードは証明書管理ノードから信頼の輪を構築できる証明書を受け取り、証明書の検証を行った後に、認証を行う。具体的な証明書収集と認証の処理は以下のようになる。

(A)証明書の収集

(1)各ノードは公開鍵と秘密鍵のペアを作成し、信頼できる相手の公開鍵に自身の秘密鍵で署名した公開鍵証明書を発行する
(2)手順に従い証明書管理ノードを選出する
(3)各ノードは証明書を証明書管理ノードに送信する
(4)証明書管理ノードは各ノードから受け取った証明書を自身のリポジトリに格納する

(B)認証処理

(1)認証ノードは証明書管理ノードに問い合わせを行う
(2)証明書管理ノードは認証ノードの要求に対し、被認証ノードへの信頼の輪を構築するために必要な証明書を、他の証明書管理ノードと協力しながら探索し、認証ノードに受け渡す
(3)認証ノードは証明書管理ノードから受け取った証明書を用いて被認証ノードの認証を行う

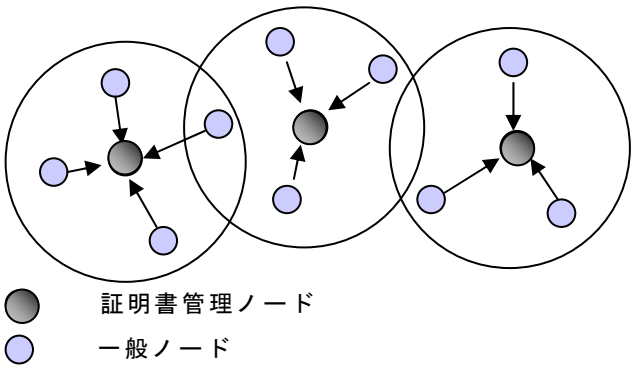

図 2　証明書管理ノード方式の概念

このときの認証にともなう通信量 *Td* は、認証ノードが証明書管理ノードに送信する問い合わせ情報の通信量、証明書管理ノードから他の証明書管理ノードへの問い合わせと受け取る証明書の通信量、およびその証明書を認証ノードに受け渡す通信量より、

$$Td = Smessage + Scerti \cdot certij + \sum_{k}^{nrequest} (Smessage + 2Scerti \cdot certik) \quad (1)$$

と表される。ここで、*Smessage* は認証要求や問い合わせなどの通信量、*Scerti* は証明書一枚あたりのサイズ、*certij* は認証ノード i が所属するクラスタの証明書管理ノードが所有する証明書数、*nrequest* は他の証明書管理ノードに問い合わせを行った回数、*certik* は問い合わせを受けた証明書管理ノード k が所有する証明書数である。

このようにして証明書管理と証明書の収集、そして信頼の輪構築を特定のノードに一任させることで、メモリ消費を抑えることが出来るようになり、性能の低いノードもネットワークに参加できるようになる。

3.2 証明書管理ノードの選定法

証明書管理ノード方式はフラッディングの際の通信量を軽減するため、OLSR を用いることを前提としている。

OLSR プロトコルには Willingness という値が各ノードに設定されている。Willingness 値は 0 から 7 の値で示され、値が高いほど中継ノードになりやすい。中継ノードになりやすいノードは周囲から経由されやすいため証明書の収集は容易となる。よって、証明書の選定は Willingness 値を基準に判断する。証明書管理ノードの選定方法を以下に示す。

(1)各ノードは自身の Willingness 値を 1 ホップ内の全てのノードにブロードキャストする
(2)各ノードは周囲のノードから受け取る Willingness 値と自身の値を比較し最も値が高い場合は自身がルーティングの基点になるべきノードであると判断し証明書管理ノードに立候補する

4. 証明書管理ノード方式の問題点

証明書管理ノード方式はネットワークを構成する全てのノードが、決められた動作を間違いなく行うことを前提としている。よって、悪意あるノードが偽造した証明書を証明書管理ノードに

紛れ込ませた場合、特定のノードの認証を妨害することが出来る。

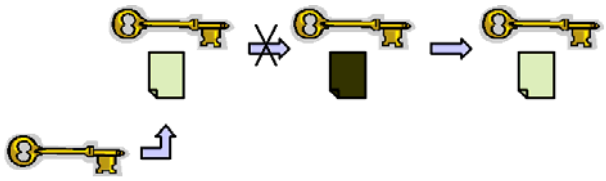

図 3　検証失敗の一例

他の証明書管理ノードから証明書を収集し構築された信頼の輪に、公開鍵あるいは電子署名を偽造された証明書が含まれていた場合、それを受け取った認証ノードは検証に失敗し、認証が不可能になる。他の証明書管理ノードに問い合わせをする順番は個々のノードが作成する証明書管理ノードリストを基に決められるため、偽の証明書が信頼の輪構築に使われるような問い合わせの順番になった場合、正しい証明書を収集する前に証明書収集を終了してしまうため、ネットワークトポロジーに変化が無い限り認証が不可能になってしまう。

5. 提案方式

5.1　偽の証明書による認証妨害への対策

証明書管理ノード方式の認証が不可能になる問題は、証明書管理ノードが検証を行う機能を有しておらず、認証ノードに偽の証明書を渡してしまうため、本当は信頼の輪が構築できないにもかかわらず構築成功と判断し、認証動作を終了してしまうために発生する。よって提案方式では、認証ノードが証明書管理ノードに対して認証要求をする際に自身の公開鍵を同時に送信し、証明書管理ノードに検証を行わせる。また、認証ノードが検証に失敗した場合、偽の証明書を受け取ったと判断し、認証ノードが自ら他の証明書管理ノードに問い合わせを行い直接証明書を収集する。

提案方式では、認証ノードが認証要求を証明書管理ノードに送る際に、同時に認証ノードの公開鍵を送付する。証明書管理ノードは信頼の輪構築の際に、認証要求と同時に送られた公開鍵を使って信頼の輪構築に使用する証明書の検証を行う。これにより、証明書管理ノードでも検証を行うことが出来るようになり、証明書管理ノードが偽の証明書を認証ノードに受け渡すことを防ぐことが出来る。

この方式で認証ノードが偽の証明書を受け取るには二通りの状況が考えられる。

パターン１　自身を管理する証明書管理ノードが悪意ある不正ノードだった場合

パターン２　他の一般ノードが自身に対して証明書を送信してきた場合

パターン１のように、自身を管理する証明書管理ノードが悪意ある不正ノードだった場合、自身に対して偽の証明書を送る、あるいは必要な証明書を送らないなどの認証妨害が可能になる。

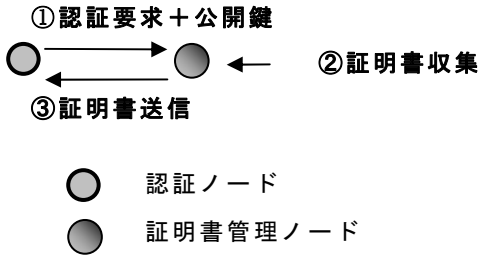

図 4　提案方式における証明書管理ノードの動作

また認証ノードに送られてくる証明書は、証明書管理ノードから送られてきたものであるという保証が無いためパターン２のように他のノードが証明書管理ノードに成りすまして認証ノードに偽の証明書を送ることが可能である。

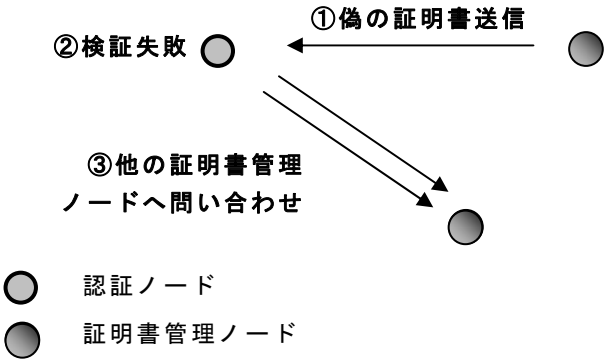

図 5　認証ノードが検証に失敗した場合の動作

提案方式では、認証ノードが受け取った証明書の検証に失敗した場合、認証妨害を受けたと判断し、自ら他の証明書管理ノードに問い合わせを行い証明書を収集する。これは、自身を管理する証明書管理ノードからの認証妨害だった場合への対策である。この時、証明書管理ノードは認証ノ

ードの動作とは独立して証明書の収集と信頼の輪構築の作業を続行し、信頼の輪構築に必要な証明書が集まった時点で、認証ノードに証明書を渡す。これにより、認証妨害が無い場合には証明書管理ノードから正しく必要な証明書だけを受け取ることができ、認証妨害があった場合でも、証明書管理ノードが正しく動作しているならば、証明書管理ノードから正しい証明書を受け取ることが出来る。

5.2 通信量の削減

証明書管理ノードは信頼の輪構築後、認証ノードに認証に必要な証明書を受け渡す際に、収集した証明書全てを受け渡す。認証に必要な証明書は信頼の輪構築に使われた証明書のみとなるので他の証明書を送る際に無駄な通信量が発生する。よって提案方式では、証明書管理ノードが信頼の輪構築後、認証ノードに証明書を受け渡す際に、必要な証明書だけを受け渡すようにする。これにより無駄な通信量の削減が出来る。

提案方式の認証にともなう通信量 *Td'*は、認証ノードが証明書管理ノードに送信する問い合わせ情報の通信量、証明書管理ノードが他の証明書管理ノードへの問い合わせと受け取る証明書の通信量、および信頼の輪構築に必要な証明書を認証ノードに受け渡す通信量、認証妨害があった場合の認証ノードから他の証明書管理ノードへの問い合わせと受け取る証明書の通信量より、

$$Td' = Smessage + Pkey + Scerti \cdot certit + \sum_{k}^{nrequest}(Smessage + Scerti \cdot certik) + \sum_{k'}^{n'request}(Smessage + Scerti \cdot certik') \quad (2)$$

と表される。ここで、*Pkey* は認証ノードが証明書管理ノードに送信する自身の公開鍵のサイズ、*certit* は証明書管理ノードが認証ノードへ受け渡す信頼の輪構築に必要な証明書の枚数、*n'request* は認証ノードが自ら他の証明書管理ノードに問い合わせを行った回数、*certik'*は認証ノードからの問い合わせを受けた証明書管理ノード k' が所有する証明書数である。

6. 提案方式の評価

証明書一枚あたりのサイズを 1Kbyte、ネットワーク上の総ノード数を 100 としてシミュレーションを行い、提案方式の評価を行う。以下、従来方式とは証明書管理ノード方式のことを指す。求めるデータは、従来方式と提案方式における、偽の証明書があった場合の認証成功確率、またそれぞれの認証時の平均通信量、そして提案方式における、平均検証回数である。

従来方式と提案方式において、ネットワーク上の正しい証明書数が 100、200、300、400 の場合の、偽の証明書の数に対する認証成功確率を図 6、図 7 に示す。

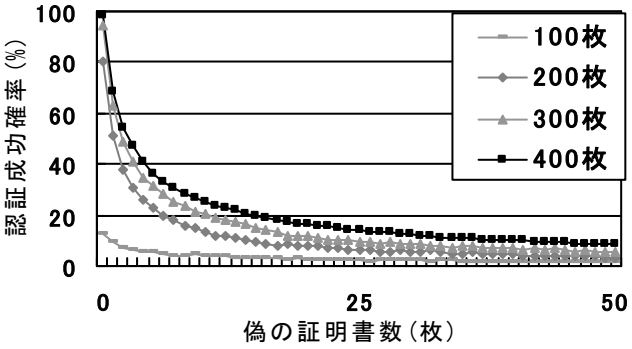

図 6　従来方式における認証成功確率

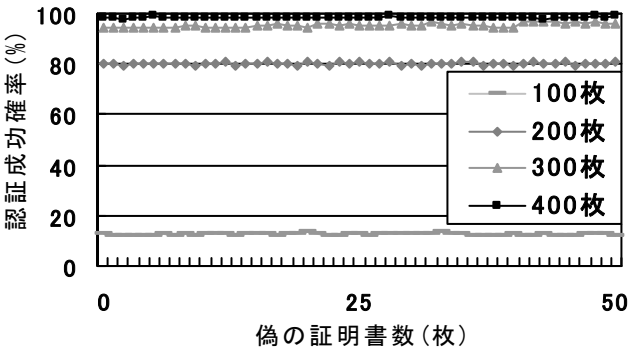

図 7　提案方式における認証成功確率

次に、従来方式と提案方式における、認証時の平均通信量を図 8 に示す。ここでの横軸はネットワーク上の正しい証明書数を示している。

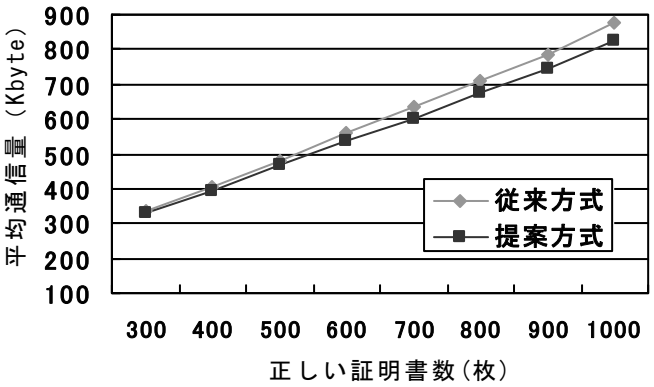

図 8　ネットワーク内における平均通信量

また提案方式では、証明書の検証を証明書管理ノードでも行っている。さらに、偽の証明書の検証に失敗する度に新たに信頼の輪構築を行い、再度検証を行うため、検証の回数が増える。そこで、提案方式においてネットワーク上の偽の証明書数に対する認証時の検証回数の平均を図 9 に示す。

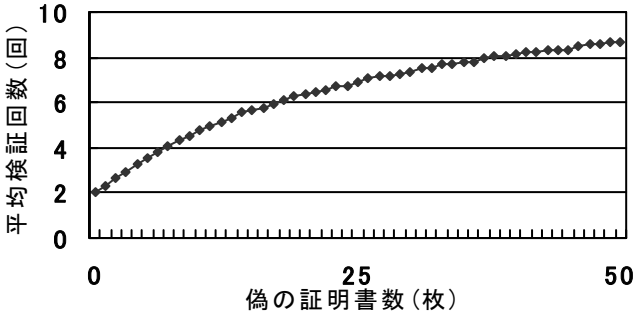

図 9　認証時の平均検証回数

7. 考察

図 6 から、従来方式では偽の証明書によって、認証成功確率が大きく降下していることが分かる。よって従来方式は、偽の証明書発行による認証妨害に脆弱であると言える。一方図 7 から、提案方式では、偽の証明書数が増加しても認証成功確率は降下しない。よって提案方式は、偽の証明書による認証妨害を防ぐことが出来ると言える。また、図 7 では、ネットワーク上の正しい証明書が 400 枚の時に、認証がほぼ成功することが分かる。シミュレーションではネットワーク上の総ノード数を 100 としているため、この条件の場合、1 ノードが平均 4 枚以上の証明書を発行していれば、認証がほぼ成功するネットワークを構築出来ると言える[4]。

図 8 から、従来方式と提案方式の認証時の平均通信量の比較を行う。正しい証明書数が 300 の時、ほぼ同値となり、ネットワーク上の証明書が増加するにしたがって、提案方式と従来方式の差が大きくなっている。よって提案方式は、証明書数が多いネットワークほど有利になると言える。

また図 9 から、提案方式は偽の証明書数が増えるにしたがって検証回数が増加することが分かる。検証回数が増えるとその分、処理速度が低下することが考えられる。よって、提案方式は従来方式に比べて、処理速度が低下すると予想される。

8. まとめと今後の課題

シミュレーションから、提案方式は偽の証明書による認証妨害を防ぐことが出来ており、通信量の削減にも成功していることが分かった。しかし、検証の回数が従来方式より増加するため、認証時の処理速度が低下すると予想される。また、処理速度に関わる要素として、ネットワーク上のノード間の距離、通信速度、ノードの移動などがあり、今回のシミュレーションではそれらを考慮に入れていない。今後は、提案方式ではどの程度処理速度が低下するのかを、ノードの移動や距離を考慮に入れた上での更に正確なシミュレーションを行い、検討していく必要がある。

また、今回は悪意ある不正ノードが偽の証明書を渡すことによる認証妨害を考えたが、その他に、悪意ある不正ノードが認証ノードに信頼の輪構築に必要な証明書を渡さないことによる認証妨害が考えられる。これは不正ノードが証明書管理ノードになることで容易く実現できる。この問題も今後の課題である。

文　献

[1] 蓮池和夫, バンディオパダイ ソンプラカシュ, 植田哲郎, “アドホックネットワークの技術的課題,”電子情報通信学会論文誌, J85-B, pp2007-2014, 2002.

[2] S.Capkun, L. Buttyan and J.-P. Hubaux,. “Self-organized Public-key Management for Mobile Ad-hoc Networks,” Vol.2, No.1, pp52-64, 2003.

[3] 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, Vol48, No.8, pp2835-2845, 2007.

[4] 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方式,” 電子情報通信学会論文誌, D-I, Vol.88, No.10, pp1571-1573, 2005.