Vol.2013-CSEC-61 No.17
Vol.2013-IOT-21 No.17
2013/5/10

証明書管理ノード方式における証明書収集処理の改良と評価

綱渕隼介[†]　　與坂宜士[†]　　長瀬智行[†]

アドホックネットワークは，地理環境に影響されずにインフラストラクチャに依存しないネットワークを構築できるという利点がある．しかしながら，マルチホップ通信を使用するという特性上，悪意のあるユーザーが通信に介在して盗聴や改ざんを行うというような危険性が存在する．この問題を公開鍵証明書によって解決する方式の一つに証明書管理ノード方式がある．この方式では，アドホックネットワークの認証処理における，証明書のやり取りで発生する通信量を削減している．本研究では，証明書管理ノード方式の証明書収集処理を改良し，更に通信量を削減することを目的としている．そして，提案方式をシミュレーションによって評価を行い，その有効性を示す．

Improvement and Evaluation of Certificates Collection process for Certificates management node

SYUNSUKE TSUNABUCHI[†]　NORIHITO YOSAKA[†]
TOMOYUKI NAGASE[†]

The wireless Ad hoc network is an independent and a self-configuring network. The mobility functions of ad hoc network, such as routing, connection's decisions and security managements are performed by the nodes themselves independently. Despite the benefits of using Ad hoc network, it can be vulnerable to security attacks. The security of ad hoc networks is based on certification authority (CA) which issues and manages security certificates and public keys for message encryption. Therefore, the security issue of Ad hoc network has become more imperative. This paper introduces a method for detecting unauthorized intrusion that is used by malicious user who issues a forged certificate to join a group of ad hoc network Reducing traffic flow of the certificates in the network is also considered.

1. はじめに

アドホックネットワークはマルチホップ通信を利用した, 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．しかしながら，アドホックネットワークでは, 悪意のあるユーザーが通信に介在することで, 盗聴や改ざんが可能であるなどのセキュリティ上の問題が存在する．これまで，数多くのアドホックネットワークのセキュリティ向上を目的とした提案がなされているが, その中の一つに証明書管理ノード方式というものがある[1]．証明書管理ノード方式では，アドホックネットワークを構成するノードが，個別の判断で，公開鍵証明書を発行し，相手ノードを認証する．発行した証明書は，証明書管理ノードと呼ばれるノードが管理する．

各ノードは認証のための証明書が必要になったときに，証明書管理ノードに証明書を要求することで証明書を入手する. 証明書管理ノードが持つ証明書が不十分であった場合は，証明書管理ノード同士が連絡を取り合い，認証に必要な証明書を収集する．

このように, あるノードに証明書を一括管理させることでその他一般ノードのメモリ消費量を削減している．

本研究では，証明書管理ノードが証明書を収集する際の，証明書管理ノード同士の連絡の取り合いの方法を改良し，認証時に発生する通信量を削減することを目的としている. そして, 提案方式をシミュレーションによって評価し，その結果から，提案方式の有効性について考察する．

1. 証明書管理ノード方式

1.1. 証明書管理ノード方式の概要

証明書管理ノード方式は，公開鍵証明書分散管理方式[2]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

[†] 弘前大学
Hirosaki University

Vol.2013-CSEC-61 No.17
Vol.2013-IOT-21 No.17
2013/5/10

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出来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．

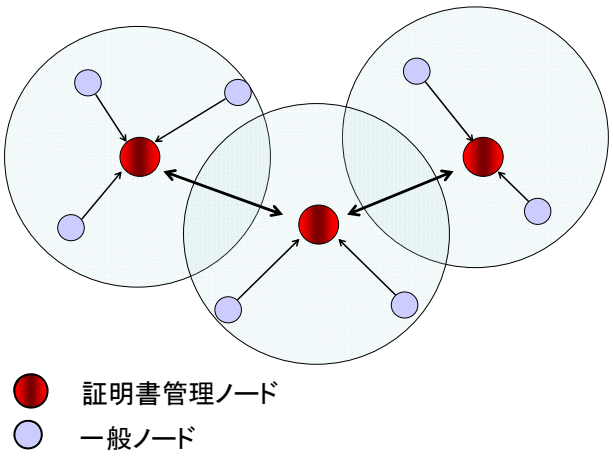

図 1　証明書管理ノードの概念
Figure 1 The Concept of Certificate management node

2.2.　ルーティングプロトコル

証明書管理ノード方式では,OLSR(Optimized Link State Routing)というルーティングプロトコルの使用を前提としている[3]．OLSR は，Proactive 型に属するルーティングプロトコルである．OLSR の大きな特徴は，MPR（Multi Point Replay）集合を用いたフラッディング（ローカルブロードキャスト）の効率化である．フラッディングとは,宛先不明のノードへパケットを送信,あるいはネットワーク上のすべてのノードにパケットを送信する手段である．単純なフラッディングのアルゴリズムは以下のようになる．

1. 送信元ノードは，自分に繋がるすべてのノードにパケットを送信する
2. 同様に各ノードは，受け取ったパケットを，自分に繋がるすべてのノードに再送信する
3. 一度送信したパケットを再び受信した場合はそのパケットを破棄する

単純なフラッディングのアルゴリズムでは無駄なパケットが多く流れてしまうという問題がある．そこで OLSR では MPR 集合を用いて無駄な再送信の削減を図っている．

2.3.　MPR 集合

MPR 集合とは，OLSR プロトコルにおいて，フラッディングの際に再送信を行うノードの集まりのことである.フラッディングを行ったノードから MPR に選ばれているノードのみが再送信を行うことで,無駄なパケットの再送信を削減し，フラッディングの効率化を実現している(図 2)．各ノードは，Willingness と呼ばれる 0 から 7 の範囲で定義されている値が設定されている．この値は，各ノードの再送信への積極性を表している.この値が高いほど再送信に対して積極的であり,他ノードから MPR に選ばれやすくなる．各ノードは，受信した Willingness から隣接ノードのどれを MPR として選択するかを決定する.証明書管理ノードで方式では,この Willingness を基準として証明書管理ノードを選定する．

2.4.　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時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 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

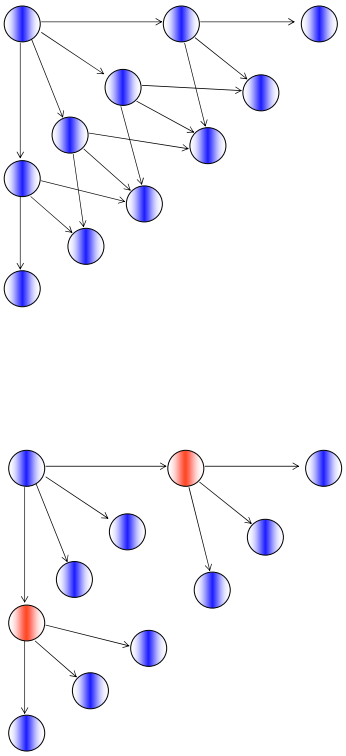

図 2　　MPR 集合によるフラッディング
Figure 2 The MPR for flooding control

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 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 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った Willingness と自身の値を比較し最も値が高い場合は自

Vol.2013-CSEC-61 No.17
Vol.2013-IOT-21 No.17
2013/5/10

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

周囲に等しい Willingness を持つノードが複数存在した場合，証明書管理ノードへの立候補は，立候補が早かったノードを優先する．伝播遅延により立候補がすれ違ってしまった場合には，複数のノードが一時的に証明書管理ノードとなる．これは後に，他の立候補情報を受け取ることで，自身の電波範囲内に他の証明書管理ノードが存在していることが判明するため，証明書管理ノードの数は制限される．

2.5. 認証

一般ノードがネットワーク上の特定のノードを認証したい場合，一般ノードは自分を管理する証明書管理ノードに認証要求を送信する．認証要求を受けた証明書管理ノードは，認証に必要な証明書の収集を行い，信頼の輪構築に成功したとき，収集した証明書を一般ノードに受け渡す．

図 2 に，ノード 1 からノード 7 への認証の様子を示す．ノード1は所属するクラスタの証明書管理ノードaに認証元の識別子と認証相手の識別子を送り，証明書の問い合わせをする．証明書管理ノード a は問い合わせを受けて，自身が所持する証明書のみで認証元から認証相手までの信頼の輪を構築出来るかどうかを判断する．構築が出来ないと判断すると，作成した証明書管理ノードリストを基に，他の証明書管理ノードに順番に証明書の問い合わせを行う．問い合わせを受けた証明書管理ノードは自身が管理している証明書全てを受け渡す．証明書管理ノード a は受け取った証明書と自身の持つ証明書とを合わせて再び 1 から 7 への信頼の輪の構築の成否を調べる．その結果信頼の輪が構築出来ることが分かると，認証に必要な証明書をノード 1 に送る．ノード 1 は受け取った証明書によって認証相手のノード 7 の認証を行う．

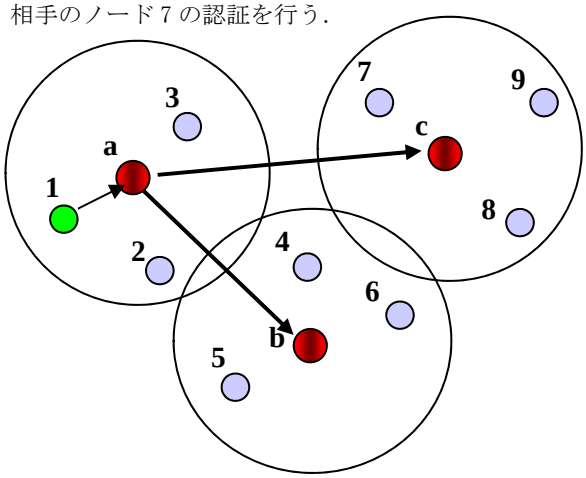

図 3　認証処理
Figure 3 Certificates processing

このときの認証にともなう通信量 Td は，認証ノードが証
者リストと公開鍵リスト両方に同じノードが現れた場合，信頼の輪が構築可能と判断出来る．

明書管理ノードに通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，

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

となる．ここで，$Smessage$ は一回のメッセージサイズ，$Scerti$ 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，$relayk$ 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から他の証明書管理ノードへ問い合わせたときの中継ノード数，$nrequest$ 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．

2. 提案方式

証明書管理ノード方式では，ネットワーク全体の通信量と，各ノードの証明書管理のためのメモリ消費量の削減することを目的としている．証明書管理ノードは，認証ノードからの認証要求に応じて，証明書収集を代行し，他の証明書管理ノードから認証に必要な証明書を取り寄せる．しかしながら，証明書管理ノード方式では，問い合わせ先の証明書管理ノードが管理している証明書を全て受け取るために，実際には認証に使用されない証明書を収集する可能性が高い．これにより，無駄な通信量が発生している．

そこで，提案方式では，認証ノードと被認証ノードに関わる証明書だけを収集する．すなわち，認証ノードが署名を付加している証明書と，被認証ノードの公開鍵が埋め込まれた証明書を指定して収集する．認証ノードと被認証ノードに関連する証明書に限定して，収集することで，信頼の輪に使用される可能性が高い証明書だけを収集する．提案方式では，署名者リストと公開鍵リストというものを新たに設けている．署名者リストは，認証ノードが署名を付加している証明書に埋め込まれている公開鍵の発行者のノード番号と，そのノードが署名を付加している証明書に埋め込まれている公開鍵の発行者のノード番号を記載する．一方，公開鍵リストは，被認証ノードの公開鍵が埋め込まれた証明書に署名を付加しているノードの番号と，そのノードの公開鍵が埋め込まれた証明書に署名を付加しているノードの番号を記載する．このようにして，認証ノードを始点，被認証ノードを終点とし，両端から信頼の輪を構築していく．認証ノードと

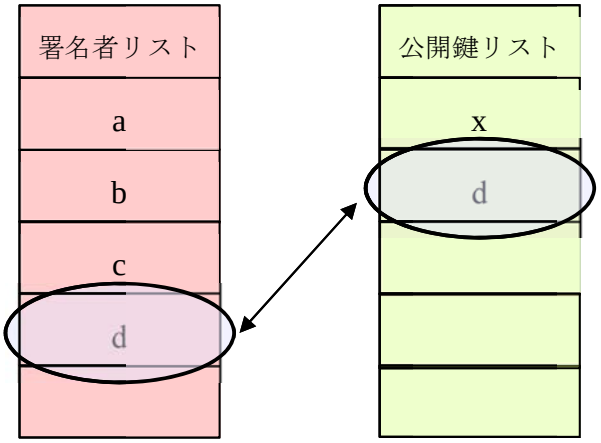

図 4　a→b→c→d→x の信頼の輪構築
Figure 4 Building web of trust

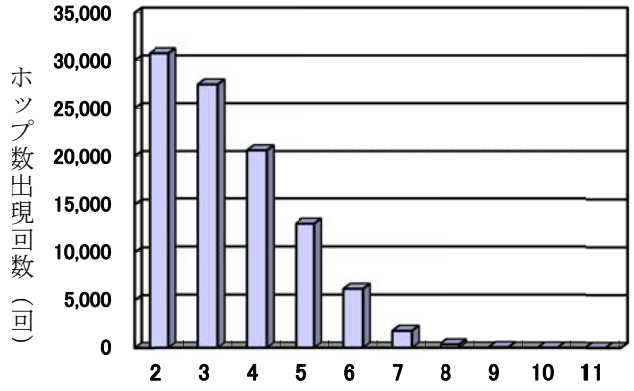

図 5　ホップ数出現頻度
Figure 5 Number of hops

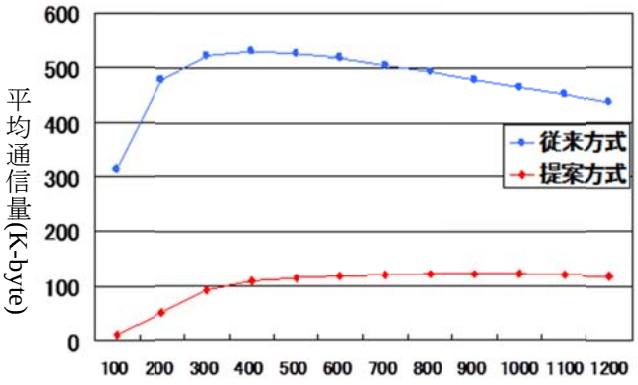

図 6　平均通信量の比較
Figure 6 Comparison of average traffic

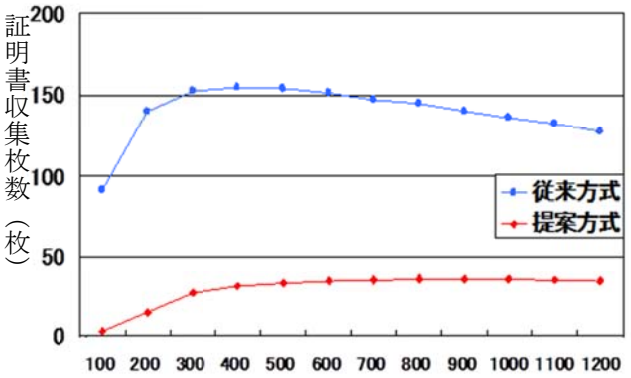

図 7　証明書収集枚数の比較
Figure 7 Comparison of the collected certificates

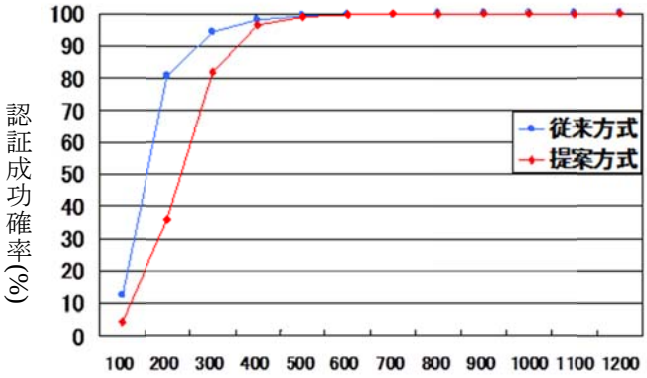

図 8　認証成功確率の比較
Figure 8 Probability of authentication success

となる．ここで，*Pkey* は認証ノードの公開鍵，*Scerti* は証明書一枚あたりのサイズ，*certit* は認証ノードが受け取る認証に必要な証明書数，*relayk* は認証要求を受けた証明書管理ノードから他の証明書管理ノードへ問い合わせたときの中継ノード数，*nrequest* は他の証明書管理ノードに問い合わせを行った回数，*certsk* は問い合わせを受けた証明書管理ノード *k* が所有する信頼の輪に関連する証明書数である．*Pkey* については参考文献[4]~[7]の仕様を踏襲している．この提案方式にともなう認証処理において発生する通信量 *Td"* は，認証ノードの認証要求による通信量，認証要求を受けた証明書管理ノードが各証明書管理ノードから収集する証明書の通信量，信頼の輪構築後に認証ノードが受け取る認証に必要とされる証明書の通信量より，

$$Td'' = Smessage + Pkey + Scerti \cdot certit + relayk \cdot \sum_{k}^{nrequest} (Smessage + Scerti \cdot certsk) \cdots (2)$$

Vol.2013-CSEC-61 No.17
Vol.2013-IOT-21 No.17
2013/5/10

3. シミュレーション

証明書管理ノード方式と提案方式をC言語によるシミュレーションにて評価する．実験環境は，Windows XP Professional，Intel Celeron，CPU 2．66GHz，メモリ502MB で行った．以下，従来方式とは証明書管理ノード方式のことを指す．シミュレーションでは，証明書管理ノード同士が通信を行う際の中継ノード数をシミュレーションにて測定する．その後，証明書管理ノード方式と提案方式において，一度の認証処理で収集する証明書の枚数を測定し,(2)式から認証時に発生する通信量を測定する.最後に，証明書の収集方法を変更することで，認証の成否にどのような影響を与えるかを，認証成功確率を求めて検討する．試行回数は全て10万回としている．

4.1. 中継ノード数

最短ホップ数を求める Dijkstra 法における中継ノード数の測定をする．縦 500m，横 500m のフィールドに，100個のノード全てが一つのネットワークに参加するように配置されるとする.ノードの電波範囲は100m とする.これは，参考文献[8]の条件を踏襲している．この条件で，無作為に選んだ二つのノードを証明書管理ノードとし，二つのノード間のホップ数を計測する．このとき，二つのノードは隣接しないものとする．ホップ数の出現頻度の分布図は図 4のようになり，その平均は3．431となった．以降のシミュレーションの通信量は,これらの条件に基づいて測定する．

4.2. 証明書収集枚数

従来方式と提案方式における一度の認証処理に収集する証明書数を計測し，図5に示す．ネットワーク上のノード数を100，証明書管理ノード数を16，一つのノードが発行している証明書の平均枚数が1から12の場合をシミュレーションする．つまり，ネットワーク上に存在する証明書数が100から1200の場合をシミュレーションする.縦軸が収集した証明書の枚数，横軸がネットワーク上の証明書の総数となっている．

4.3. 通信量

従来方式と提案方式における，認証時に発生する通信量の平均を求め，図6に示す．ネットワーク上のノード数を100，証明書管理ノード数を 16，証明書一枚あたりのサイズを 1K-byte とする．ここでは，証明書収集の変更による差異を観測するために，提案方式でも，証明書管理ノードが認証ノードに証明書を受け渡す際に，収集した全ての証明書を送信するものとする．縦軸が平均通信量，横軸がネットワーク上の証明書の総数となっている．

4.4. 認証成功確率

従来方式と提案方式における，認証成功確率を求め，図7に示す．ネットワーク上のノード数を 100，証明書管理ノード数を16，ネットワーク上に存在する証明書数が100から1200の場合をシミュレーションする.縦軸が認証成功確率,横軸がネットワーク上の証明書の総数となっている．

4. 考察

図4のホップ数分布図について考察する．ホップ数が2の場合を最大の数として,それ以降は徐々に減少している．最小ホップ数は2であるので，仮に電波範囲を大きくしていくと,グラフの比重は2に偏っていくものと考えられる．

証明書収集枚数と通信量について考察する．図 5，図 6より，従来方式，提案方式共に，証明書収集枚数と通信量の増減の特性が一致していることが分かる．これは，認証処理時に発生する通信量の大部分が，証明書送信による通信量であることを示している．提案方式は，認証処理に送信される証明書の総数を削減出来ているため，通信量を大幅に削減することが出来ている．また，図6より，通信量が最大の点がある．これは，その点を境界として，送信する証明書により発生する通信量の増加量と，他の証明書管理ノードへの問い合わせ回数の減少による収集証明書数の減少量が逆転することを示している．

次は，従来方式と提案方式における認証成功確率について比較する．図7から，提案方式は，ネットワーク上の証明書の総数が 400 以下のときに，従来方式を下回る．500以降には従来方式とほぼ同値となる．提案方式では，信頼の輪構築に使用される証明書を優先して収集することで，通信量を削減出来る一方で，ネットワーク上に十分な数の証明書が存在しない場合に，パフォーマンスが低下しやすくなる．特に証明書総数が200のときには，約2．2倍の差がついている．これは，提案方式では，ネットワーク上の収集出来ない証明書が存在することが原因だと考えられる．信頼の輪に関連するかどうかの判断に用いられる署名者リスト，公開鍵リストの情報は，他の証明書管理ノードに問い合わせる度に蓄積されていく．よって，初期に問い合わせを行った証明書管理ノードでは，関連性の情報が不足して必要な証明書を収集出来ない場合がある．この現象が，ネットワーク上の証明書数が少ない状態に起こりやすいと考えられる．

以上から，提案方式は，通信量において有利ではあるが，ネットワークが構築した直後など，各ノードが他のノードに対して発行している証明書の数が少ない状況で認証に失敗しやすいと言える．ほぼ確実に認証に成功する確率を99%以上とするならば，提案方式では，ノード数が100，平均証明書発行枚数が5枚あれば,最小の証明書発行枚数で，認証処理を成功させることが出来ると言える．すなわち，提案方式の証明書総数500の場合が最も理想的であると結論付けられる．

Vol.2013-CSEC-61 No.17
Vol.2013-IOT-21 No.17
2013/5/10

5. まとめと今後の課題

本研究では，証明書管理ノード方式おいて，証明書の収集方法を改良し，通信量を削減することを目的とした．提案方式では，認証ノードから被認証ノードまでの信頼の輪に使用される可能性の高い証明書のみを収集することで，通信量の削減と認証の成功の両立を図った．そして，提案方式における認証処理時の通信量を求めるために，証明書管理ノード間の平均中継ノード数を計測し，通信量をシミュレーションした．また，認証の成否を，認証成功確率としてシミュレーションを行い，従来方式と提案方式を比較した．その結果，提案方式は，通信量を削減出来ていることが分かった．しかし一方で，ネットワーク上の証明書の総数が少ない状態では，認証成功確率が大きく低下することが分かった．そして，提案方式では，ほぼ確実に認証に成功する場合の最低平均証明書発行枚数が 5 枚の場合が，理想的なネットワーク環境であると結論付けた．

今後の課題として，ネットワーク上の証明書の総数が少ない状態でも，認証成功確率が低下しないような改良が必要である.提案方式では,証明書の収集を終了する条件を，従来方式と同様にして，信頼の輪構築に成功，あるいは他の全ての証明書管理ノードへ問い合わせしたときとしている．この終了条件の場合，提案方式では認証に必要な証明書が収集出来ない場合が存在する．今後は，認証処理のタイムアウト条件の変更など，改良を加えて，この問題を解決する必要がある．

参考文献

1) 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)
2) Capkun, S, Buttyan, L. , Hubaux, J. -P, Self-organized public-key management for mobile ad hoc networks，IEEE Trans， Mobile Computing， Vol．2，No．1，pp52-64，(2003)
3) T．Clausen， P． Jacquet， “Optimized Link State Routing Protocol (OLSR)”， RFC3626， (2003)
4) Norihito Yosaka , Iichiro Nishimura , Tomoyuki Nagase , Authentication and Certificate Managements of Unauthorized Intrusion in Ad-hoc Networks， Problems and Solutions，The International Conference on Network-Based Information Systems, (NBiS011)，pp． 646-650，(2011)
5) 與坂宜士，長瀬智行，アドホックネットワークのセキュリティと通信効率に関する研究，第 1 回情報処理学会東北支部研究会，pp．10．1-5，(2011)
6) 與坂宜士，長瀬智行，モバイル通信における証明書管理ノード方式の効率の良い証明書収集，IEICE Technical Report，Vol．112，No． 126，pp． 83-90，2012
7) 與坂宜士，長瀬智行，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不正ノードによる認証妨害の問題と改善，IEE-Japan， 電気学会電子・情報・システム部門大会，MC3：情報セキュリティ pp．1180-1185，(2012)
8) 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).