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

船曳俊介[†] 磯原隆将[†] 北田夕子[†]
竹森敬祐[††] 笹瀬巌[†]

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におけるノード認証は重要な技術であり，各ノードが独自に証明書を発行してリポジトリを管理することで，自身で信頼の輪を構築して相手認証を行う公開鍵証明書分散管理方式が提案されている．しかし，この方式は証明書の収集に時間がかかり，通信量およびメモリ消費量が大きいという欠点がある．そこで本論文では，アドホック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て，そのノードの電波範囲内にいる周囲のノードが発行した公開鍵証明書を代行管理する方式を提案する．本方式は，証明書の管理を行う証明書管理ノードの選定と，各所に点在するノード間のクラスタリングから構成される．本方式は，証明書が発行や失効されるタイミングで，証明書管理ノードのリポジトリに証明書を登録もしくは削除を行うことにより，証明書の収集に要する時間を削減でき，失効証明書リストの管理が不要になる．計算機シミュレーションにより，信頼の輪の構築成功率，証明書の管理のために使用するメモリ量，ネットワーク全体での通信量について評価を行い，メモリ量，通信量ともに，従来方式よりも削減できることを示す．

Self-organized Public Key Management with Certificate Management Nodes for Wireless Ad Hoc Networks

Shunsuke Funabiki,[†] Takamasa Isohara,[†] Yuko Kitada,[†]
Keisuke Takemori[††] and Iwao Sasase[†]

In a wireless ad hoc network that has no connection to the Internet, certificating nodes is an important technique and the self-organized public key management has been proposed. In this scheme, each node creates and manages certificates by itself. However, it needs large time to collect all certificates in the network and costs a lot of traffic after collecting all certificates since each node collects certificates periodically. In this paper, we propose representation techniques for public key certificate on wireless ad hoc network by using certificate management nodes that collect certificaties of each node in their power range. This proposed scheme is consist of selection of certificate management node and clustering technique. In the proposed scheme, we can cut time of collecting certificates since certificate management node stores or deletes certificates in its repository when certificates are issued or expired. By a computer simulation, we evaluate average traffic and memory and show that the proposed scheme can reduce the both traffic and memory than the conventional scheme.

1. はじめに

公開鍵基盤（PKI: Public Key Infrastructure）では，ノードが生成した公開鍵に対応する秘密鍵の正規所有者を保証するために，認証局（CA: Certificate Authority）と呼ばれる機関が，公開鍵に対して公開鍵証明書を発行する[1)]．近年，多数のノードが，お互いに協調してパケットを中継する，無線アドホックネットワークが注目されている．しかし，インターネットと独立した無線アドホックネットワークの場合，信頼できる第三者機関である既存の CA を利用することができないという問題がある．そのため，見ず知らずのノードによるデータの盗聴や改竄の恐れがある電子会議システムやデータ交換システムのようなアプリケーションを使用する場合，認証が重要な技術となる．そこで，ノード自身が公開鍵証明書を管理する方式として，各ノードが独自に証明書を発行し，リポジトリで

† 慶應義塾大学理工学部情報工学科
Department of Information and Computer Science, Keio University
†† 株式会社 KDDI 研究所
KDDI R&D Laboratories, Inc.

管理する公開鍵証明書分散管理方式が提案されている[2]．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，リポジトリ情報を補完し，信頼の輪の構築を通じて認証を行う方式である．しかしこの方式は，各ノードがネットワーク内のすべての証明書を集めることになり，証明書の収集に時間がかかる，通信量が増大する，証明書を保持するためのメモリ消費量がすべてのノードにおいて大きくなるという問題がある．

そこで本論文では，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て，リポジトリを構築する方式を提案する．証明書管理ノードは，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するように，クラスタリング手法を用いて選択される．本方式では，一般のノードが，証明書管理ノードに発行した証明書と失効情報を送信するだけで，証明書の収集と失効リストの管理が完了するため，隣接するノードと定期的に証明書を交換する従来方式に比べて，証明書に関する情報の収集時間を短縮できる．また，証明書を格納するリポジトリの定期的な交換が不要なため，従来方式と比較して通信量が低減される．一般のノードにおける証明書管理のためのリポジトリのメモリ使用量は，一般ノードでは自身の発行した証明書の数に対応し，証明書管理ノードでは証明書管理ノード数に反比例して削減される．計算機シミュレーションにより，信頼の輪の構築成功率，全ネットワークにおける通信量について評価を行う．また証明書の管理に必要なメモリ量について評価する．その結果，提案方式は，無線アドホックネットワーク環境において有効な証明書管理方式であることを示す．以下，2 章で従来方式とその問題点について述べ，3 章で提案方式について説明する．4 章で特性評価を行い，最後に 5 章でまとめる．

2. 従来方式：公開鍵証明書分散管理方式

2.1 従来方式の動作

ノード間の既存の認証技術として PGP[3] における信頼の輪がある．これは，ある公開鍵に対してこれを信頼する者が署名を施すことで信頼性の基準の 1 つにする技術である．PGP では，この公開鍵に有効性と信頼度という値を設けて，それぞれ 5 段階の度合いを設定している．有効性は公開鍵が証明書の発行を受けた本人に属しており，改竄もされていないことを表す指標であり，信頼度はある人に他の人を紹介してもらう際に，紹介者をどれだけ信頼するかという指標である．従来方式は，PGP の信頼の輪の概念を無線アドホックネットワークに導入したものであるが，有効性と信頼度の指標を段階分けせず，信頼できるノードの署名が正しい状態で付与されていれば有効性と信頼度がともにあると判断する．従来方式[8] における信頼の輪の構築手順はシステムは本提案でも踏襲する．

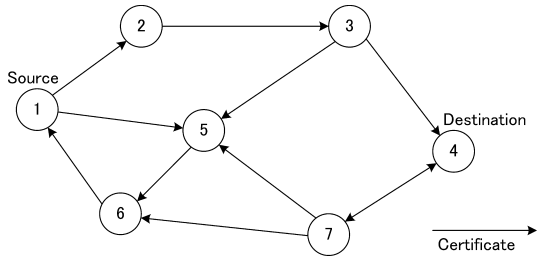

図 1 信頼の輪の構築
Fig. 1 Establishment of certificate chain.

図 1 に信頼の輪の構築を示す．1 から 7 がノードを示し，矢印の向きに相手を信頼しているものとする．図 1 では，ノード 1 がノード 2 を，ノード 2 がノード 3 を，そしてノード 3 がノード 4 を信頼しているため，最終的に信頼の輪はノード 1 からノード 4 へとつながっている．これらの信頼性は公開鍵証明書によって保証され，ノード 1 からノード 2 への証明書にはノード 2 の公開鍵とノード 1 の署名が含まれる．文献 2) では，各ノードは独自に信頼する他ノードの公開鍵に対して証明書を発行し，ノードごとにアドホックネットワーク内の全証明書を収集して，リポジトリを作成し，通信相手ノードまでの信頼の輪を構築することで，認証を行う方式を提案している．これは，各ノードが，移動しながら電波の届く範囲内のノードと，リポジトリを定期的に交換することで，ネットワーク内のすべての証明書を集めている．失効証明書の管理についても，各ノードは，無効な証明書の一覧を記載する失効証明書リストを作成している．各ノードは，失効証明書リスト内の情報をつねに最新に保つために，証明書を発行したノードに定期的に問い合わせることで，証明書の有効性を確認し，失効証明書リストを更新する．

2.2 従来方式の問題点

従来方式では，各ノードがネットワーク内のすべての証明書を収集するため，証明書の収集を完了するまでに大きな時間を要する．例として，ノード数が 100，証明書数が 600 枚で構成されるアドホックネットワークにおいて，リポジトリ交換の間隔を 60 秒とする場合，1 ノードが全証明書を収集するために要する時間は約 10,000 秒になることが報告されている[2]．この

収集完了までのタイムラグにより，信頼の輪を構成する証明書について，認証時における有効性について発行したノードに問い合わせる必要がある．また，ネットワーク内のノード数が多い場合，管理しなければならない証明書の数が膨大になるため，すべてのノードにおいてリポジトリサイズが増大する．一般的に，簡易端末が集まった無線アドホックネットワークの場合，端末のメモリ量が制限されることから，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．さらに，他ノードとの定期的なリポジトリ交換により，ネットワーク内の通信量が増大してしまうという問題もある．この通信量の増大は，ネットワークの輻輳やパケット衝突によるデータの損失を招く．

2.3 オンデマンド公開鍵分散管理方式

その他の公開鍵証明書の管理方式としてオンデマンド公開鍵分散管理方式[7] が提案されている．文献 7) では，各ノードは自身の発行した証明書のみをリポジトリに格納しておき，認証要求が発生した時点で通信相手ノードまでの信頼の輪を他ノードと協調して構築する．文献 7) では，オンデマンドで信頼の輪を構築するため，メモリ使用量を軽減できることや，有効な証明書を収集するために失効証明書リストの確認処理が不要になるなどの優位性を持つが，オンデマンドで証明書を収集するためにすべてのノードがダウンせずにネットワークに接続されている必要があるという欠点を持つ．

本論文では公開鍵分散管理方式を従来方式 1，オンデマンド公開鍵分散管理方式を従来方式 2 として，比較および評価を行う．

3. 提案方式：公開鍵証明書の代行管理

本論文では，アドホックネットワーク内のノードの中から，リポジトリを代行管理する証明書管理ノードを設けることで，証明書の収集に要する時間と通信量を削減する方式を提案する．提案方式において認証が必要な場合は，認証要求が発生した時点で証明書をオンデマンドで入手する．また，本提案に要する通信量の評価式として，証明書管理ノードの選定にともなう通信量を T_a，初期のクラスタリングにともなう通信量を T_b，再クラスタリングにともなう通信量を T_c，認証にともなう通信量を T_d，証明書の失効にともなう通信量を T_e として，それぞれ定式化する．なお，本提案における信頼の輪のシステムは従来方式 1 のものと同様である．

3.1 基 本 動 作

図 2 に証明書管理ノードの概念図を示す．以下，認証を行うノードを認証ノード，認証の対象となるノードを被認証ノードと呼ぶ．各ノードは，信頼できる相手の公開鍵証明書を発行し，証明書の管理を証明書管理ノードに委託する．すなわち，証明書管理ノードは，周囲のノードにおけるリポジトリとして動作する．認証要求が発生した場合，認証ノードは証明書管理ノードから信頼の輪を構築するために必要な証明書を受け取る．提案方式は，A) 証明書の収集，B) 認証処理からなる．以下にそれぞれの手順を示し，詳細については 3.2 節以降で述べる．

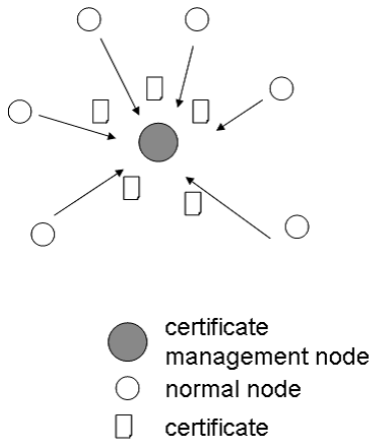

図 2 提案方式における証明書管理ノードの概念
Fig. 2 Proposal model of certificate management node.

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対して，自身の秘密鍵で署名した公開鍵証明書を発行する．

(2) 3.2 節で提案する手順に従い証明書管理ノードが選出される．

(3) 各ノードは証明書を証明書管理ノードに送信する．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合せを行う．

(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．必要な証明書の具体的な探索手順に関しては 3.4 節で説明する．

(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．

3.2 証明書管理ノードの選定法

A)-(2) に述べた証明書管理ノードの選出方法について述べる．提案方式におけるアドホックネットワークは

フラッディングの際の通信量を軽減するために，OLSR（Optimized Link State Routing）プロトコル[4)~6)]を用いることを前提とする．また，OLSR プロトコルにおいては，フラッディングの際に情報を再送信する中継ノードへのなりやすさを示す値である Willingness 値というものを各ノードが持つ．Willingness 値は 0 から 7 の範囲で示され，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．これらをふまえて，証明書管理ノードの選出は，証明書の収集効率を高めるために Willingness の値を基準に判断し，ルーティングの核となるノードが率先して証明書管理ノードを務めるようにする．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 $will_i$ を 1 ホップ内にいるノードすべて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 $will_i$ を自身の値と比較し，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．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．そのため，周囲に等しい Willingness 値を持つ PC が多数存在した場合でも，立候補の時刻はランダムであり，Willingness 値が異なる場合と同様に最も立候補が早かったノードが証明書管理ノードになるため，証明書管理ノードの収束に必要な時間は変わらない．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になる．しかし，後から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，3.3.2 項で後述するクラスタリングの再構築によって証明書管理ノード数は制限される．

このときの証明書管理ノードの選定にともなう通信量 T_a は，ネットワーク内にいる N 個のすべてのノードが自身の Willingness 値を 1 ホップ以内にいるノードに対してブロードキャストすることによって，Willingness 値の交換を行うため，

$$T_a = N \cdot S_{message}$$

と表される．ここで N はネットワーク全体のノード数，$S_{message}$ は 1 回のメッセージサイズである．

3.3 クラスタリング

証明書管理ノードが 1 つしか存在しない場合，ネットワーク全体の証明書がその 1 つの証明書管理ノードに集中して，証明書管理ノードのメモリ使用量が膨大になってしまう．また通信量の観点からも，アクセス集中によって輻輳やパケット損失が発生してしまう．そのため，複数の証明書管理ノードが管理することによって証明書を収集する範囲を分担することを考える．この場合，各ノードと証明書管理ノードとの結び付きを決定するクラスタリングが必要となる．クラスタは証明書管理ノードごとに，証明書管理ノードと，その電波範囲内に存在する一般ノードで構成される．**図 3** にクラスタの構成を示す．

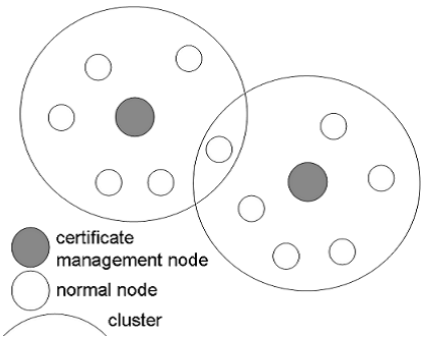

図 3 クラスタリング
Fig. 3 Grouping model of cluster.

3.3.1 初期のクラスタリング

証明書管理ノードと一般ノードの関係が定まっていない状態からのクラスタリングの手順について，**図 4** に初期のクラスタリングの過程を示す．

(1) 証明書管理ノードに立候補したノード i は，電波範囲内のノードに識別子 ID_i^{manage} を添えて，立候補情報を電波範囲にブロードキャストする．

(2) 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．そのため，周囲に等しい Willingness 値を持つ PC が多数存在した場合でも，立候補の瞬間はランダムであり，Willingness 値が異なる場合と同様に最も立候補が早かったノードが証明書管理ノードになるため，証明書管理ノードの収束に必要な時間は変わらない．立候補情報を受信したノードは立候補せずに，そのノードのクラスタに所属する．各ノードが立候補情報を受信できない場合，電波範囲内に立候補するノードが存在しないと判断して，自ら立候補する．一度クラスタに所属したノードは，クラスタが重複しないように，後から他の立候補情報を受信しても無視する．立候補情報が無視された証明書管理ノードは，一定時間経過しても識別子と証明書を受信できないノードに関しては，自身立候補情報が無視されたと判断する．各ノードは，受信した ID_i^{manage} を自身が所属するクラスタの証明書管理ノードの情報として保存する．以上の作業によりクラスタが形成される．

(3) 立候補によって決定した証明書管理ノードは，立候補情報とは別に，自身が証明書管理ノードであると

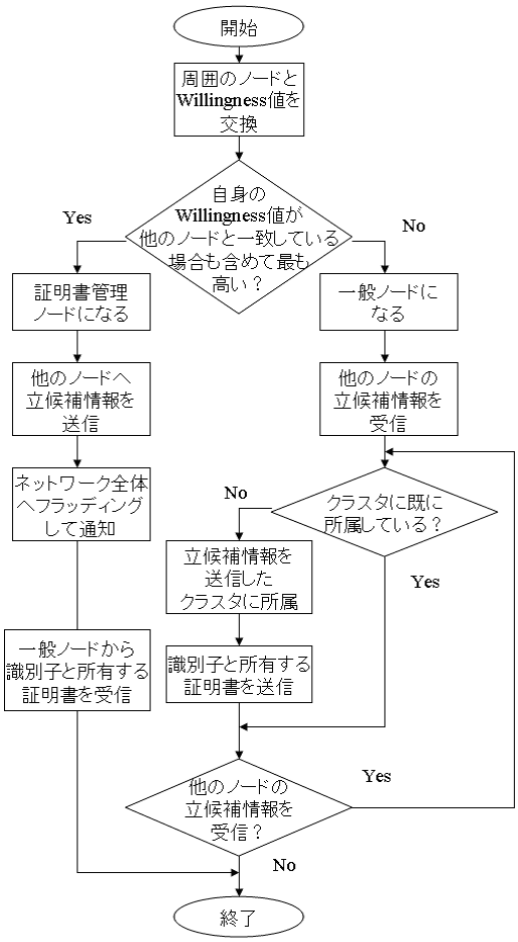

図 4 初期のクラスタリングの過程
Fig. 4 Initial grouping process of cluster.

いう情報を，識別子 ID_i^{manage} を添えて，ネットワーク内の全ノードにフラッディングして伝える．フラッディングは，通信量を最小にするため，OLSR プロトコルの MPR（Multiple Point Relay）フラッディングにより行う．この作業により，各ノードはすべての証明書管理ノードを把握でき，どのノードが証明書管理ノードであるかについて記載した証明書管理ノードリストを各自で作成する．証明書管理ノードリストは 3.3.2 項で記述する再クラスタリングによって，証明書管理ノードが変更されるつど，フラッディング情報によって更新される．

(4) クラスタの形成後，各ノードは，自身の発行した証明書と自身の識別子 ID_i を，所属するクラスタの証明書管理ノードに受け渡す．本提案では証明書管理ノードの信頼性に関しては考慮しないため，証明書管理ノードが一般ノードから信頼されていない場合でも，一般ノードは証明書管理ノードに証明書を受け渡す．

このときの初期のクラスタリングにともなう通信量 T_b は，立候補したノードが周囲のノードにブロードキャストして通知する情報の通信量，証明書管理ノードになったことを全体にフラッディングして通知する通信量，および各ノードが所属するクラスタの証明書管理ノードへの通知と証明書の通信量より，

$$T_b = N_{manage} \cdot S_{manage} + N_{manage} \cdot S_{flood} + \sum_{i}^{N_{normal}} (S_{message} + S_{certi} \cdot certi_i)$$

と表される．ここで，N_{manage} は証明書管理ノードの総数，S_{flood} は MPR 集合のフラッディングによる通信量，N_{normal} は一般ノードの総数，S_{certi} は証明書 1 枚あたりのサイズ，$certi_i$ は各ノードが所有する証明書の枚数を表す．また，S_{flood} は定数ではなく，各ノードによって異なる分布を持つ．

3.3.2 クラスタリングの再構築

無線アドホックネットワークの場合，ノードの位置はたえず変化する．そのため，以下に示す 2 通りの状況において，証明書管理ノードおよびクラスタを変更し，最適なクラスタを再構築する必要がある．なお，Willingness 値は各ノードの CPU の処理速度やメモリ量によって定められる固有の値であるため，ノードの配置状況が変化した場合でも，Willingness 値は変化しない[4]．

パターン 1 一般ノードの電波範囲から証明書管理ノードが外れてしまった場合

図 5 に一般ノードに注目したクラスタの再構築の過程を示す．証明書管理ノードと一般ノードの移動によって，お互いが電波範囲内に存在しなくなった場合，一般ノードは所属するクラスタを変更する必要がある．一般ノードが認証の際に証明書管理ノードまで問合せができない場合，一般ノードは証明書管理ノードが電波範囲から外れたと判断する．一般ノードは，新たに所属するクラスタを探索するため，電波範囲内のノードに孤立したことを通知するメッセージをブロードキャストする．一般ノードの電波範囲内に他の証明書管理ノードが存在して，孤立情報を受け取った場合，証明書管理ノードは受け入れを許可するというメッセージを一般ノードに返信する．新たなクラスタが決定した一般ノードは，初期のクラスタリングと同様に，自身の識別子とすべての証明書を証明書管理ノードに送付する．電波範囲内に他の証明書管理ノードが存在せず，受け入れ可能メッセージが返ってこない場合は，自ら証明書管理ノードに立候補する．証明書管理ノードに立候補した一般ノードは，初期のクラスタリングの手

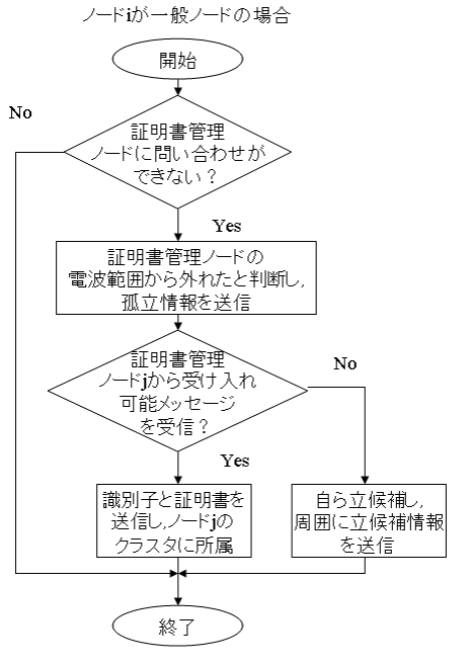

図 5 一般ノードに注目したクラスタの再構築の過程
Fig. 5 Re-grouping process of cluster with user node.

順と同様に電波範囲内に立候補情報を送信し，周囲の孤立した一般ノードを自身のクラスタに所属させる．一般ノードが別の証明書管理ノードに証明書を再登録したときに，今まで所属していたクラスタの証明書管理ノードの情報も送信することで，再登録を受け付けた証明書管理ノードと今まで所属していたクラスタの証明書管理ノードが通信し，今まで所属していたクラスタの証明書管理ノードは自身が管理する一般ノードが電波範囲から外れたことを検知する．検知した証明書管理ノードは電波範囲から外れた一般ノードの識別子の情報と，該当する一般ノードが発行した証明書を削除する．

このとき再クラスタリングにともなう通信量 T_c のうち，パターン 1 にともなう通信量 T_{c1} を一般ノードが他のクラスタに所属する場合における通信量 T_{c11}，一般ノードが自ら証明書管理ノードに立候補する場合における通信量 T_{c12} に分ける．T_{c11} は，周囲のノードにブロードキャストする孤立情報の通信量，証明書管理ノードが送信する許可通知の通信量，および証明書管理ノードに送信する識別子と証明書の通信量より計 3 回のメッセージ送信が行われるため，

$$T_{c11} = 3S_{message} + S_{certi} \cdot certi_i$$

と表される．T_{c12} は孤立情報と立候補情報の通信量より，計 2 回のメッセージ送信と 1 回のフラッディングが行われるため，

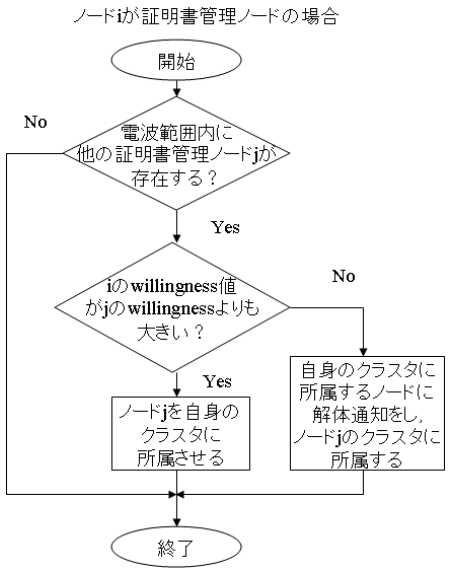

図 6 証明書管理ノードに注目したクラスタの再構築の過程
Fig. 6 Re-grouping process of cluster with certificate management node.

$$T_{c12} = 2S_{message} + S_{flood}$$

と表される．

パターン 2 証明書管理ノードが密集した場合

図 6 に証明書管理ノードに注目したクラスタの再構築の過程を示す．証明書管理ノードが，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を検出した場合，証明書管理ノードは互いに，Willingness 値の情報を交換し，Willingness 値の低い方の証明書管理ノードは一般ノードとなり，自身のクラスタを解体する．クラスタの解体が決定した証明書管理ノードは，自身のクラスタに所属するノードに解体を通知し，管理する証明書や所属ノードの識別子を破棄して，もう一方の証明書管理ノードのクラスタに所属する一般ノードになる．このとき，証明書管理ノードになったときと同様に，証明書管理ノードではなくなったことを示す情報を，識別子 ID_i^{manage} を添えて，ネットワーク内のノードにブロードキャストする．解体によって孤立した一般ノードは，パターン 1 に述べた手順により，クラスタリングの再構築を行う．証明書管理ノードどうしの Willingness 値が同じ場合は，解体のメッセージの通知が早かった方を優先して解体する．隠れ端末問題によりメッセージが受信できなかった場合，一定時間経過後にメッセージの再送信を促す通知をし，再び通信を試みる．

ここでパターン 2 にともなう通信量 T_{c2} は，証明書管理ノードどうしで willingness 値を交換する通信量，クラスタを解体するために証明書管理ノードが自

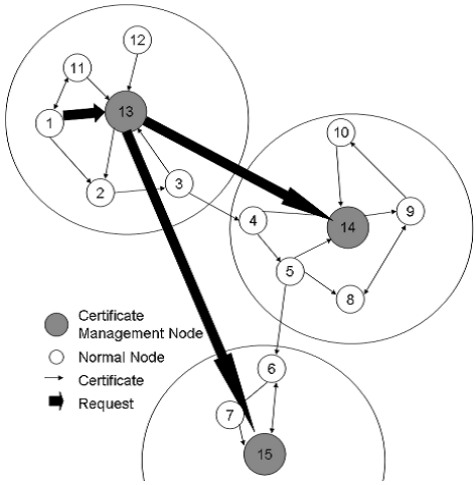

図 7 認証作業
Fig. 7 Certificate query.

身のクラスタに所属するノードに通知する解体情報通信量，および新たに所属するクラスタに送信する識別子と証明書の通信量より，

$$T_{c2} = 2S_{message} + S_{message} + S_{message} + S_{certi} \cdot certi_i$$

と表される．

3.4 認　　証

本提案では，認証要求が発生した時点で証明書をオンデマンドで入手する．**図 7** に認証作業を示す．ノード 1 が認証ノードであり，ノード 7 が被認証ノードである場合を例に，認証の手順を以下に示す．

(1) ノード 1 は，所属するクラスタの証明書管理ノード 13 に，認証元ノード識別子と認証相手ノード識別子に関する情報 (ID_1, ID_7) を送り，証明書の問合せをする．

(2) 証明書管理ノード 13 は，自身のリポジトリに登録されている証明書を探索し，自身の持つ証明書のみでは信頼の輪が構築できないことを確認すると，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証明書の問合せを行う．問合せは，証明書管理ノードリストに記載されている順に 1 回ずつ行い，複数のノードを経由して問合せを行う．図 7 では 14, 15 の順番に問合せを行う．認証ノードからの認証要求を受けた証明書管理ノードが他の証明書管理と信頼関係がない場合でも，他の証明書管理ノードに問合せを行う．

(3) 証明書管理ノード 14 は，問合せを受信後，自身のリポジトリに格納されている全証明書を，送信元の証明書管理ノード 13 に返信する．証明書管理ノード 13 は，証明書管理ノード 14 からの証明書を受信後，自身の持つ証明書と合わせて，再び 1 から 7 への信頼の輪が構築できるか，証明書を探索する．信頼の輪が構築できないことを確認すると，次は証明書管理ノード 15 へ，証明書の問合せを行う．この作業は，信頼の輪が構築できるまで続けられる．

(4) 証明書管理ノード 15 は，問合せを受信後，自身のリポジトリに格納されている全証明書を，送信元の証明書管理ノード 13 に返信する．証明書を受信した証明書管理ノード 13 は，証明書管理ノード 14 からの証明書と自身の持つ証明書とを合わせて，再び 1 から 7 への信頼の輪が構築できる証明書を探索する．その結果，$(1 \to 2 \to 3 \to 4 \to 5 \to 6 \to 7)$ という信頼の輪が構築できることが分かると，認証に必要な証明書をノード 1 に送る．

(5) ノード 1 は，受け取った証明書によって被認証ノードであるノード 7 の認証を行う．

ここで，ノードのメモリ消費量を考慮して，認証が成功した後，認証を行ったノードと証明書管理ノードは，受け取った証明書を削除する．ネットワーク内の全証明書を収集しても信頼の輪がつながらない場合，証明書管理ノードは，認証ノードに対して，認証不可能であることを伝える．

このときの認証にともなう通信量 T_d は，認証ノードが証明書管理ノードに送信する問合せ情報の通信量，証明書管理ノードから他の証明書管理ノードへの問合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，

$$T_d = S_{message} + S_{certi} \cdot certi_j + \sum_{k}^{n_{request}} (S_{message} + 2S_{certi} \cdot certi_k)$$

と表される．ここで，$certi_j$ 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 j が所有する証明書数，$n_{request}$ は他の証明書管理ノードに問合せを行った回数，$certi_k$ は問合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．

3.5 証明書の失効

秘密鍵の漏洩などにより，証明書を失効する必要がある場合，各ノードは，証明書の失効情報を証明書管理ノードに送る．失効情報を受け取った証明書管理ノードは，受信した失効情報に該当する証明書を破棄し，管理する証明書の状態をつねに最新に保つ．また，クラスタリングの再構築により，ノードの所属するクラスタが変更された場合，証明書管理ノードはそれまで所属していたノードの証明書をすべて削除して，古い証明書を残さない．提案方式では 3.4 節で述べたよ

うに，認証要求が発生した時点でオンデマンドで証明書を収集するため，各ノードが証明書管理ノードに問い合わせて得られる証明書は，つねに最新の状態のものであり，認証処理において証明書の有効性を検証するする必要はなく，失効証明書リストの管理が不要となる．

本方式における失効作業は，各ノードが証明書を管理する証明書管理ノードに失効情報を送信して，該当する証明書が削除されるだけで達成される．失効完了に必要な時間は 1 ホップの距離におけるメッセージの送信時間であり，これは無視してよいほど微小な時間で完了できる．

このときの証明書の失効にともなう通信量 T_e は，証明書を失効する必要のあるノードが証明書管理ノードに失効情報を通知する通信量のみであるから，

$$T_e = S_{message}$$

と表される．

3.6 ノードがダウンした場合の対策

従来方式 1 ではノードごとにすべての証明書を管理していたため，いくつかのノードがダウンしても認証処理への影響は少ない．しかし提案方式では，証明書管理ノードがダウンした場合，その証明書管理ノードのクラスタに所属する一般ノードは認証を行うことができなくなる．具体的な例として 500 m × 500 m の正方形の範囲内に電波の到達範囲が半径 100 m のノードがランダムな位置に 100 ノード存在する場合，1 つの証明書管理ノードがダウンすると約 5 個の一般ノードが認証できなくなる．

そこで認証処理の信頼性を確保するために，証明書管理ノードの二重化という方式と，迅速な再クラスタリングによる回復という 2 つの方式が考えられる．ここで前者を適用する場合，証明書管理ノードの電波範囲内に別のノードを立候補させることになり，提案プロトコルの大きな変更を要してしまう．一方，後者を適用する場合，証明書管理ノードからの待ち時間を短く設定するのみで，周囲のノードが素早く証明書管理ノードへと立候補することができ，迅速な復旧による信頼性の確保を実現できる．たとえば，アドホックネットワークにおける通信速度が 1 Mbps，再クラスタリングに要する 1 つのメッセージが約 0.1 Kbyte，証明書のサイズが 1 Kbyte である場合，待ち時間を 1 秒に設定すると，タイムアウト後の孤立情報の送信時間，および他の証明書管理ノードから受け入れ可能情報を受信する通信時間はそれぞれ約 0.001 秒となり，証明書 1 枚を送信する時間は約 0.01 秒となる．したがって，ノードの内部処理を加えても認証要求を発信してから再クラスタリングまで 2 秒以内に完了でき，証明書管理ノードがダウンしている場合でも，迅速な証明書管理ノードの復旧を図ることができる．

3.7 不正ノードについての考察

提案方式において，ノードがクラスタリングの際に交換する Willingness 値は自己申告であるため，不正ノードが故意に高い Willingness 値を申告して証明書管理ノードになる場合が考えられる．その場合，クラスタリングを行わない従来方式に対して，提案方式では，各ノードが認証の問合せを行っても，不正な証明書管理ノードが信頼の輪の構築に必要な証明書を受け渡さないという被害が生じる．不正な証明書管理ノードが虚偽の証明書を受け渡す場合，認証ノードの通信相手の公開鍵証明書であると偽ることによるなりすまし攻撃が考えられる．また本方式では，不正な証明書管理ノードが信頼の輪の構築が可能であるにもかかわらず，構築不可能であるという虚偽の情報を認証ノードへ返信する攻撃を受けた場合，認証ノードが通信相手のノードを認証できなくなるというという被害が考えられる．このような不正ノードが存在する場合，メッセージの送信回数，送信する証明書の枚数の変化は，送信されるべき情報が送信されない場合による影響のみでほとんど変わらないため，通信量に対しての影響はない．こうした不正ノードを特定する必要はあるが，本提案は効率の良い証明書の管理手法に注目しており，不正ノードを特定する機能は有していない．不正ノードの特定を支援する既存技術として階層型認証機構[8] があるが，本提案モデルへの適用はできないため，新たな仕組みの提案が今後の課題である．

4. 特 性 評 価

提案方式の有効性を検証するために，信頼の輪の構築成功率，証明書の管理のために使用するメモリ量，全ネットワークにおける通信量の評価を行う．

4.1 証明書の収集時間

従来方式 1 では，各ノードが隣接するノードとリポジトリの交換を繰り返すことで，ネットワーク内のすべての証明書を収集するため，大きな時間を要する．これに対して提案方式では，証明書の収集は各ノードが証明書管理ノードに証明書を送信する作業のみで達成され，従来方式 1 で必要とされる，各ノードが個別にリポジトリ交換を実施しなくてすむため，証明書の収集時間は短縮できる．提案方式における証明書の収集は，アドホックネットワークにおける通信速度が 1 Mbps，証明書のサイズが 1 Kbyte，各ノードの証明書保持数が平均 4 枚である場合，単純に各ノードが 4

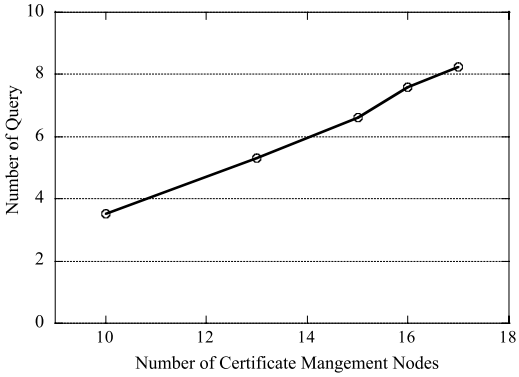

図 8 信頼の輪を構築するための平均問合せ回数
Fig. 8 Average request number of query to make certificate chain.

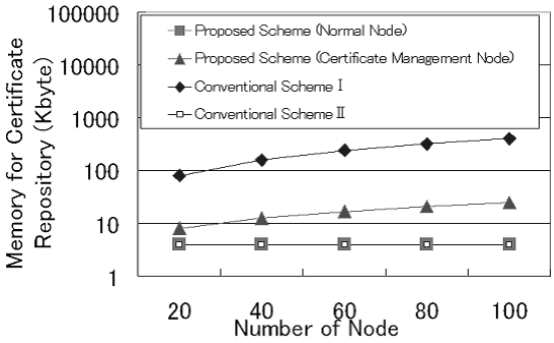

図 9 証明書の管理に使用するメモリ量
Fig. 9 Memory size for certificate repository.

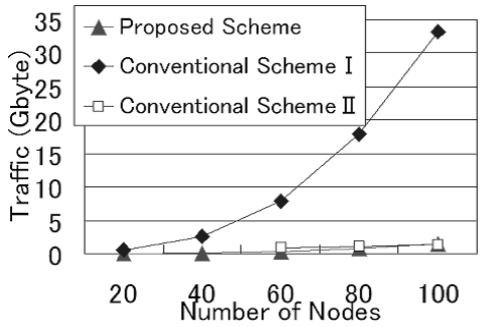

図 10 ネットワーク内における通信量
Fig. 10 Traffic load on ad hoc network.

枚の証明書を送付するのに 0.05 秒程度を要するため，送受信の確認処理を含めても 1 秒未満で完了する．これに対して従来方式では，2.2 節に示される条件において 10,000 秒の時間を要する[2]．ここで，不正なノードが証明書の送信を意図的に遅らせた場合には，遅らせた時間だけ対応する証明書管理ノードによる証明書の収集は遅れるが，従来方式 1 では，全ノードから証明書を直列的に収集する必要があるため，存在する不正ノードの数に比例して証明書の収集が遅れる．

4.2 信頼の輪の構築成功率

信頼の輪が確実に構築されるためには，文献 7) でノード数が 100 以下の場合，1 ノードが少なくとも平均 4 ノード信頼すればよいことが分かっている．以後のシミュレーションでは，ノード数が 100 の場合に 1 ノードが平均 4 ノードを信頼してほぼ確実に信頼の輪の構築が成功するシステムを想定する．

図 8 に信頼の輪を構築するための，他の証明書管理ノードへの平均問合せ回数を示す．図 8 より平均 4 ノード信頼するシステムの場合，証明書管理ノード数の半分以下の問合せ回数で，信頼の輪がつながることが分かる．

4.3 使用メモリ量

図 9 に証明書の管理に使用するメモリ量を示す．横軸は全体の総ノード数を示している．また，各ノードの電波範囲は 100 m，シミュレーション範囲は 500 m × 500 m の正方形とする．ノードの移動は，範囲内の座標をランダムに目的地として決定し，目的地に向けて一定速度で移動し，到着後に再び新たな目的地を設定するランダムウォークモデルを用いる．証明書の管理は，証明書管理ノードのみが行うので，一般のノードが使用するメモリ量は，従来方式 2 と同様に自身の発行した証明書の容量のみである．また，証明書管理ノード自体も，複数で全ネットワークの証明書を分担して管理するので従来よりもメモリの使用量が軽減される．たとえば全体のノード数が 100 の場合，証明書管理ノードは約 16 個選定され，各証明書管理ノードのメモリ使用量は全証明書を収集する従来方式の 1/16 となる．また，ネットワークにおけるノードの数が増えるほど，使用メモリ量の差は大きくなることが分かる．したがって提案方式は，ノード密度の高いネットワークにおいて，特に有効であることが分かる．

4.4 平均通信量

図 10 にネットワーク内における証明書の管理に要する通信量を示す．各ノードの平均移動速度は 10 m/s とし，シミュレーション時間は 1,000 秒とした．提案方式は，ノードの移動によって発生するクラスタリングの再構築に必要な通信量が 100 ノードで約 2 GByte となり全体の通信量の 3/4 を占めるが，従来方式 1 における失効証明書の確認と証明書収集のための定期的なリポジトリ交換がなくなるため，従来方式 1 よりも通信量は軽減されて従来方式 2 とほぼ同等の値を示した．また，ネットワーク内のノード数が増える場合，提案方式では通信量が一定の割合で増加する．これに対して従来方式では，一度にリポジトリを交換す

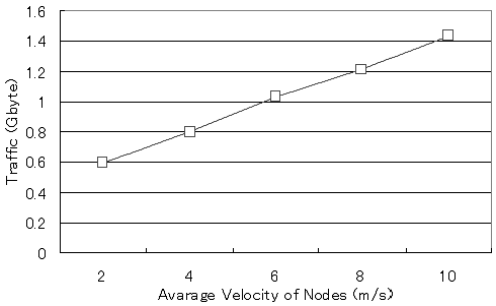

図 11 各ノードの平均移動速度を変化させた場合の通信量
Fig. 11 Traffic load versus average node speed.

るノードが増えるため，全体のノード数を N，シミュレーション範囲の面積を S，電波の届く範囲の面積を P とすると，各ノードの電波範囲に存在する平均ノード数は NP/S となることから，自分自身を除いた周囲の全ノードと一度に行う通信量は $N(NP/S-1)$ となり，通信量の増加率は $N(N-1)$ で表される二乗関数に従っている．したがって 4.3 節同様に，提案方式はノード密度の大きいネットワークにおいて，特に有効であることが分かる．次に，提案方式はノードの移動速度に依存したクラスタの再構成による通信が発生する．そこで，**図 11** に全ノード数を 100 として各ノードの平均移動速度を変化させた場合の通信量の評価結果を示す．各ノードの平均移動速度が遅い場合，クラスタリングの再構築を行う回数が少なくなるため，証明書の管理に要する通信量は少なくなる．すなわち，各ノードの移動速度が遅いネットワークにおいて，提案方式の優位性は増す．

5. おわりに

本論文では，インターネットと独立した無線アドホックネットワークにおける公開鍵証明書を管理する方式として，ネットワーク内のノードから証明書管理ノードを選定し，周囲のノードが発行する証明書を収集して，リポジトリを代行管理する方式を提案した．本方式では，一般のノードが証明書を預ける証明書管理ノードを決定するためにクラスタリングを行っている．提案方式では，一般のノードが証明書管理ノードに証明書を送信するだけで，ネットワーク内の証明書の収集が完了するため，証明書の収集時間を短縮できる．計算機シミュレーションにより，少ない問合せ回数によって，他の証明書管理ノードへ信頼の輪がつながることを確認した．証明書の管理のために使用するメモリ量，全ネットワークにおける通信量についても評価を行い，ともに従来方式よりも削減できることを示した．メモリ量に関しては，1 ノードが平均 4 枚の証明書を発行し，範囲は 500 m × 500 m の正方形の中で，各ノードの電波範囲を 100 m，全体のノード数を 100 とする場合，一般ノードのメモリ使用量は証明書 4 枚分の容量となり，証明書管理ノードのメモリ使用量は従来の 1/16 となることを示した．通信量に関しては，全体のノード数が 40 の場合，従来の通信量の 1/10，ノード数が 100 の場合，従来の通信量の 1/20 となることを示した．特に，各ノードの移動速度が遅い場合，クラスタリングの再構築を行う回数が少なくなるため，通信量が減少することを示した．以上より，提案方式はネットワーク内のノード数が増えるにつれて，メモリ量および通信量に関して従来方式との差が大きくなり，ノード密度の高いネットワークについて特に有効であることが示された．また移動速度に関して，各ノードの速度が遅い場合に有効であることが示された．

謝辞 本論文は，文部科学省 COE「アクセス網高度化光電子デバイス」プログラム，および KDDI 研究所との共同研究によって行われた．関係者各位に深謝する．

参考文献

1) IETF: RFC3280 Internet X. 509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (Apr. 2002).
2) 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 (Jan.–Mar. 2003).
3) Zimmermann, P.: *The Official PGP User's Guide*, MIT Press (1995).
4) Jacquet, P., Muhlethaler, P., Qayyum, A., Laouiti, A., Viennot, L. and Clausen, T.: Optimized link state routing protocol, Internet-draft, draft-ietfmanet-olsr-02.txt (2000).
5) Wu, B., Wu, J. and Fernandez, E.B.: Secure and efficient key management in mobile ad hoc networks, *Proc. 19th IEEE International Parallel and Distributed Processing Symposium*, p.288a (Apr. 2005).
6) Heinzelman, W.R., Chandrakasan, A. and Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks, *Proc. 33rd Hawai Internatiobnal Conference on System Sciences*, Vol.2, p.10 (Jan. 2000).
7) 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.J88-D-I, No.10, pp.1571–1573 (2005).

8) 小原奈緒子，小口正人：仮想マシンを用いたMANETにおける階層型認証機構の提案と実装，電子情報通信学会コンピュータシステム研究会，CPSY2006-43, pp.31–36 (2006).
9) 鎌田美緒，小口正人：マルチホップネットワークにおけるセキュアな通信路構築に関する制御手法，電子情報通信学会ITS研究会，ITS2006-39, pp.25–30 (2006).

(平成18年12月14日受付)
(平成19年 5 月 9 日採録)

船曳　俊介

昭和59年生．平成19年慶應義塾大学理工学部情報工学科卒業．同年三井住友海上火災保険（株）入社．在学中，インターネットセキュリティに関する研究に従事．

磯原　隆将（学生会員）

昭和56年生．平成17年慶應義塾大学理工学部情報工学科卒業．平成19年同大学大学院修士課程修了．同年KDDI（株）入社．在学中，インターネットセキュリティに関する研究に従事．

北田　夕子

昭和56年生．平成16年慶應義塾大学理工学部情報工学科卒業．平成18年同大学大学院修士課程修了．同年ゴールドマンサックス（株）入社．在学中，インターネットセキュリティに関する研究に従事．

竹森　敬祐（正会員）

平成6年慶應義塾大学理工学部電気工学科卒業．平成8年同大学大学院修士課程修了．同年KDD（株）入社．平成16年慶應義塾大学大学院博士課程修了．現在，（株）KDDI研究所勤務．主として，通信ネットワークおよびインターネットセキュリティに関する研究に従事．平成14年度電子情報通信学会学術奨励賞受賞．IEEE，電子情報通信学会各会員．

笹瀬　巌（正会員）

昭和31年生．昭和54年慶應義塾大学工学部電気工学科卒業．昭和59年同大学大学院博士課程修了．同年オタワ大学理工学部電気工学科ポストドクトラルフェロー，昭和60年同大学講師．昭和61年慶應義塾大学理工学部電気工学科助手，昭和63年同大学専任講師，平成4年同助教授，平成11年同大学理工学部情報工学科教授，現在に至る．主として，ディジタル通信，通信ネットワーク，光通信理論，マイクロ波通信，非線形通信システム，通信理論，符号理論，インターネットセキュリティに関する研究に従事．工学博士．昭和59年度IEEE COM.SOC. 学生論文賞，昭和62年度第3回井上研究奨励賞，昭和63年第1回安藤博紀学術奨励賞，昭和63年篠原記念学術奨励賞，平成8年度電子情報通信学会交換システム研究会優秀論文賞受賞．IEEE Senior Member，電子情報通信学会，情報理論とその応用学会各会員．