ユビキタス情報環境のための階層型公開鍵分散管理方式

中山誠也[†1] 武田敦志[†2,†1] 北形元[†1,†3]
チャクラボルティ デバシシュ[†3] 白鳥則郎[†1,†3]

ユビキタス情報環境においては，様々な計算機端末がネットワークを介してプライベートな情報のやりとりを行うことが予測されため，公開鍵暗号を用いて安全な通信を行うための仕組みが必要不可欠となる．しかし，従来の公開鍵認証基盤には利便性とスケーラビリティに関する課題が存在するため，より誰もが容易に利用可能なスケーラブルな認証基盤が必要となる．そこで本稿では，ネットワークに参加するそれぞれのノードが公開鍵を効率的に分散管理する手法を提案する．この手法ではネットワークに参加するノードの階層化を行うことで，携帯端末などの計算能力が低いノードであっても正しい公開鍵を取得し，安全な通信を行うことが可能となっている．本稿では，提案手法について説明し，シミュレーション結果を通じて提案手法のスケーラビリティを検証する．

Hierarchical Public Key Management for Ubiquitous Environments

SEIYA NAKAYAMA,[†1] ATSUSHI TAKEDA,[†2,†1]
GEN KITAGATA,[†1,†3] DEBASISH CHAKRABORTY[†3]
and NORIO SHIRATORI[†1,†3]

In ubiquitous computing environments, a lot of private data is sent through computer networks. Therefore, we need a secure communication mechanism which is enabled by public key encryption, however, conventional public key infrastructures have some problem in usability and scalability. So, we need a new scalable public key infrastructure which everyone can use easily. In this paper, we propose a distributed public key management system which has a hierarchical structure. In the proposed system, both high performance machine and low performance machine such as PDA can do secure communications. In this paper, we explain the proposed system and show scalability of this system through simulation results.

1. はじめに

情報技術の発達やコンピュータネットワークの普及に伴い，ユビキタス情報環境が実現されつつある．ユビキタス情報環境においては，コンピュータネットワークを介してセンサー端末や携帯端末などの様々な計算機端末が相互接続を行い，人や物に関する様々な情報を共有する．このようにして共有された情報を効果的に利用することで，人々にコンピュータの存在を意識させることなく，快適なサービスを提供することが可能になる．一方，ユビキタス情報環境においては，個人情報などのセンシティブな情報の通信を扱う機会も多く，公開鍵暗号を用いたセキュアな通信を行う仕組みが必要となり，公開鍵の管理と配布が不可欠となる．しかし，従来の公開鍵管理方式は利便性やスケーラビリティの点に課題があった．

そこで本稿では，利便性に優れ，スケーラブルな公開鍵分散管理方式:HiHDAM(Hierarchical Hash-based Distributed Authentication Method) を提案する．HiHDAM では，分散ハッシュテーブルを用いてノード識別子の一意性に基づいた公開鍵のスケーラブルな管理・配布を実現することに加え，ネットワークに参加する計算機端末（以降，ノードと呼ぶ）の階層化を行うことで，携帯端末やセンサーなどの計算能力が低いノードの参加をも実現する．

以下，2 章では関連研究について述べ，3 章で利便性に優れ，スケーラブルな公開鍵分散管理方式 HiHDAM を提案する．また，4 章ではシミュレーションにより HiHDAM のスケーラビリティを検証し，5 章で本稿のまとめと今後の課題を述べる．

2. 関連研究

コンピュータネットワークにおいて，安心・安全な通信を実現するためには，公開鍵暗号技術を用いた暗号通信やデジタル署名が不可欠である．ネットワークに参加するすべてのノードから，これらの技術を利用するためには，公開鍵の確実な管理と配布が必要となる．

Public Key Infrastructure(PKI)[1] は最も有名な公開鍵認証基盤である．PKI では認証局と呼ばれるサーバで各ノードの公開鍵に対して公開鍵証明書を発行する．各ノードは必要に応じて，通信相手の公開鍵証明書に付加された電子署名を認証局の公開鍵で検証するこ

†1 東北大学大学院情報科学研究科
Graduate School of Information Sciences, Tohoku University
†2 東北文化学園大学知能情報システム学科
Department of Intelligent Information Systems, Tohoku Bunka Gakuen University
†3 東北大学電気通信研究所
Research Institute of Electrical Communication, Tohoku University

とで，通信相手の公開鍵の正当性を確認する．PKI では，ノードの利用者と認証局の管理者の社会的な信頼関係に基づいて公開鍵証明書の発行を行うため，公開鍵証明書を取得するためには複雑な手続きを必要し，利便性に課題がある．また，PKI は公開鍵証明書の集中管理を行うため，スケーラビリティにも問題がある．そのため，膨大な数のノードがネットワークに参加することが予想されるユビキタス情報環境に PKI を適用すると，公開鍵証明書の発行・管理に莫大なコストが必要となる．そこで，利便性に優れ，スケーラブルな公開鍵の管理と配布を実現する方法が必要とされている．

サーバを必要としない公開鍵の管理方式として Pretty Good Privacy(PGP)[2] がある．PGP は信頼の輪と呼ばれる各ノードの利用者間の信頼関係を活用することにより，信頼できるノードを介して新たな公開鍵を収集することで，認証局を必要としない分散型の公開鍵管理を実現している．そのため，ノードの利用者と認証局の管理者との社会的な信頼関係の確認も不要となり，各ノードの利用者間の合意によって新たな公開鍵の登録をすることが可能となる．しかし，PGP は新たな公開鍵を検索するための仕組みを持たず，計画的な信頼の輪を形成することが出来ないため，任意のノードの公開鍵を入手するために，全てのノードの公開鍵を入手する必要がある．そのため，各ノードは公開鍵を管理するために多大なメモリ量を必要とし，必要な公開鍵を入手するために多くの通信メッセージを必要とする．そのため，ユビキタス情報環境に対する PGP の適用は難しい．

サーバを必要とせず，かつ効率的に公開鍵の管理と配布を実現する公開鍵分散管理手法が提案されている[3)−5)]．これらの手法では，ネットワークのルーティング情報などに基づき新しい公開鍵を自動的に収集することで，ノードが必要とするメモリ量と通信メッセージ量を削減している．しかし，ルーティング情報などのネットワーク特有の情報を利用するため，適用できるネットワークが限られてしまうという問題がある．

適用可能なネットワークを限定せず，効率的に公開鍵の分散管理を実現する手法として Hash-based Distributed Authentication Method(HDAM)[7] がある．HDAM は，信頼の輪と分散ハッシュテーブル（DHT)[6] を用いることで，確実な公開鍵の管理・配布を実現している．しかし，参加するすべてのノードは，ほかのノードによる公開鍵の取得の仲介を行わなければならず，すべてのノードに一定の負荷が生じてしまうため，センサーや携帯端末などの低性能な計算機端末の参加に対応することが難しくなっている．そのため，多様な計算機端末の参加が予想されるユビキタス情報環境へそのまま適用することは難しい．

これらの手法に対し，本稿で提案する階層型公開鍵分散管理方式 : HiHDAM は，ネットワークに参加するノードの階層化を行うことにより，公開鍵暗号の暗号処理を行うには計算能力が不十分な低性能端末でも参加可能な公開鍵分散管理方式を実現する．

3. 提案：Hierarchical HDAM(HiHDAM)

3.1 HiHDAM の概要

ユビキタス情報環境において公開鍵暗号を用いたセキュアな通信を行うためには，利便性に優れ，スケーラブルであることに加え，センサーや携帯端末などの低性能端末でも参加できる公開鍵管理方式が必要である．そこで本稿では，ネットワークに参加している任意のノード間の認証と公開鍵の効率的な分散管理を実現する HDAM に親ノード・子ノードの概念を取り入れることで，低性能端末でも参加可能であるスケーラブルな公開鍵分散管理方式を提案する．

提案手法では，効率的に公開鍵を分散管理するために仮想的にオーバーレイネットワークを構築する．その際，高性能端末を親ノードとし，ハッシュリング上に配置する．親ノードは DHT を効果的に用いて信頼の輪を形成することにより，公開鍵の安全で効率的な管理を実現する．また，低性能端末は子ノードとし，ハッシュリング上に配置せず，親ノードを介して必要な公開鍵を入手できるようにする．これにより，子ノードはハッシュリング上で行われる公開鍵の分散管理や公開鍵の配布の中継を行わなくてよいため，これに伴う暗号の復号化や電子署名の検証を行う必要はなくなり，子ノードの負担を最小限に抑えることが可能となる．

3.2 公開鍵の分散管理

図 1 に HiHDAM による公開鍵の分散管理の例を示す．ここで，A〜G は親ノードを示し，a1,e1,e2 はそれぞれノード A とノード E の子ノードを示す．また，$i.hash$ は親ノード i のハッシュ値，$i.parent$ は子ノード i の親ノード，K_i はノード i の公開鍵を示す．親ノードはそれぞれの識別子から一方向ハッシュ関数で決められたハッシュ値を基に，1 から N までの指標を円形に配置したハッシュリング上に仮想的に配置される．子ノードは，同様の方法で得られたハッシュ値を基に決められた親ノードを介して HiHDAM に参加する．そして，親ノードはハッシュリングにおいて自身の位置から正の方向に $2^k(k = 0, 1, 2, \cdots)$ 以上離れたノードのうち最も近い位置に配置されたノードの公開鍵と，自身の子ノードの公開鍵を管理する．最大ハッシュ値が N のとき，各親ノードが管理する公開鍵の最大数は $log_2 N$ となる．図 1 の場合，A が管理する公開鍵は以下の 4 個となる．

- 正の方向に $2^1(2^0)$ 以上離れたノードの中で最も近い位置に配置されたノードである B の公開鍵 K_B

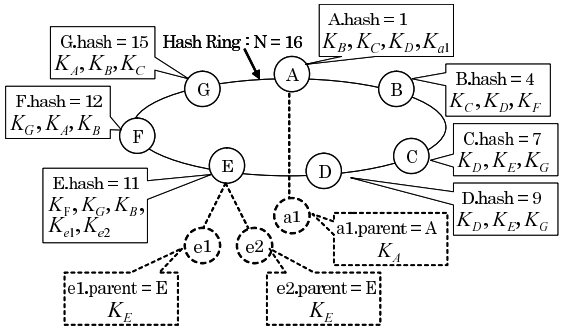

図 1　公開鍵の分散管理

- 正の方向に 2^2 以上離れたノードの中で最も近い位置に配置されたノードである C の公開鍵 K_C
- 正の方向に 2^3 以上離れたノードの中で最も近い位置に配置されたノードである D の公開鍵 K_D
- 自身の子ノードである a1 の公開鍵であるノード a1 の公開鍵 K_{a1}

また，子ノードは自身の親ノードの公開鍵のみを管理することとなり，図 1 の場合，a1 が管理する公開鍵は自身の親ノードである A の公開鍵 K_A1 個のみとなる．

3.3　信頼の輪と DHT を用いた公開鍵の入手

子ノード s が子ノード d の公開鍵を保持していない状態で，d から s への認証要求が行われるなどし，公開鍵の入手が必要になった場合，以下の手順により，s は d の公開鍵を入手する．

1. s は自身の親ノード S に d の公開鍵 K_d を要求する．
2. S が公開鍵 K_d を保持している場合，S は s へ K_d を送信する．
3. S が公開鍵 K_d を保持していない場合，S は d の親ノードである D の公開鍵 K_D を保持しているか確認する．
   - 3.1. S が公開鍵 K_D を保持している場合，S は D に対し，d の公開鍵 K_d を要求する．
   - 3.2. S が公開鍵 K_D を保持していない場合，S は自身が公開鍵を保持しているノードの中から，ハッシュリング上で最も D に近い位置に配置され，かつ D より小さいハッシュ値を持つノード S^t に対し，D の公開鍵 K_D を要求する．
   - 3.3. S^t が公開鍵 K_D を保持している場合，S^t は S へ公開鍵 K_D を送信する．
   - 3.4. S^t が公開鍵 K_D を保持していない場合，S^t が公開鍵を保持しているノードの中から，ハッシュリング上で最も D に近い位置に配置され，かつ D より小さいハッシュ値を持つノード S' の公開鍵 $K_{S'}$ を S に送信する．S は S' の公開鍵 $K_{S'}$ を入手し，手順 3.2 へ戻る．
4. S は D に対し d の公開鍵 K_d を要求する．
5. D は S に対し，d の公開鍵である K_d を送信する．
6. S は受信した公開鍵 K_d を s へ送信する．

また，親ノード S が子ノード d の公開鍵を入手する場合には，上で示した手順 3 から手順 6 までの動作を行う．親ノード S が親ノード D の公開鍵を入手する場合には，手順 3.2 から手順 3.4 までの動作を行うことで解決できる．同様に，子ノード s が親ノード D の公開鍵を入手する場合には，手順 1 から手順 3 までの動作を通して s の親ノードであるノード S がノード D の公開鍵 K_D を入手した段階で，s に対し公開鍵 K_D を送信することで解決できる．最大ハッシュ値が N のとき，公開鍵の入手に必要となる通信データ量は $O(log_2 N)$ となる．その際，ネットワークで送受信されるメッセージの大部分を占める公開鍵の中継に関してはすべて親ノードが行い，子ノードがメッセージを処理しなければならないのは，自身の公開鍵を要求された場合と，自身が要求した公開鍵が送信されてくる場合のみとなる．これにより子ノードの負担を最小限に抑えることが出来，低性能端末でも参加可能な分散管理を実現出来る．

3.4　ノードの参加手順

ノード n が HiHDAM に参加するときの前提条件として，n はすでに HiHDAM に参加している任意のノード g と公開鍵を交換済みであるものとする．n が親ノードだった場合の HiHDAM に参加する手順を以下に述べる．

1. n はハッシュリング上で自身の正側に隣接するノード $n.successor$ の公開鍵を g から取得する．
2. n が認証すべきノードを計算し，それらの公開鍵を $n.successor$ から取得する．
3. $n.successor$ は自身の公開鍵を配布したノードに対して，信頼の輪の再構築を通知する．
4. 再構築の通知を受けたノードは自身が認証すべきノードを再計算し，それらのノードの公開鍵を取得する．n の公開鍵は $n.successor$ から取得する．

n が子ノードだった場合は，n は g を通して n の親ノード $n.parent$ の公開鍵を取得する

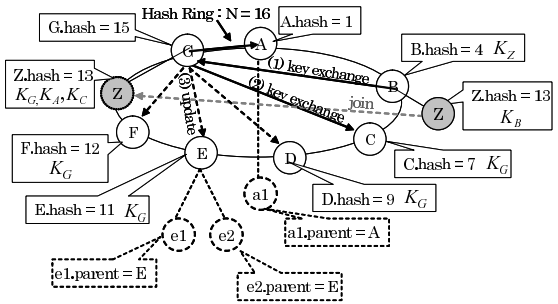

図 2 ノードの参加手順

ことで HiHDAM に参加する．その際，n の親ノード $n.parent$ は，一方向性ハッシュ関数を用い n の持つ固有の情報から一意に決定される．

図 2 にノードが HiHDAM に参加する際の例を示す．この例では，親ノード Z がノード B を介してネットワークに参加する．以下，本例をもとに処理手順を述べる．

1. Z はハッシュリング上で自身の正側に隣接するノード G の公開鍵 K_G を B から取得する．
2. Z は自身が認証するノード G,A,C の公開鍵を G から取得する．
3. G は自身の公開鍵を保有しているノード F,E,D,C に対して，信頼の輪の再構築を通知する．この通知をうけたノードのうち，F,E,D は Z を認証するために Z の公開鍵 K_Z を G から取得する．

以上の手順により親ノード Z は HiHDAM へ参加する．最大ハッシュ値が N のとき，新しい親ノードがネットワークに参加するために必要な通信データ量は $O(log_2 N)$ となる．

3.5 ノードの離脱手順

親ノードがネットワークから離脱する場合，離脱ノードはハッシュリング上で隣接するノードに離脱通知を行い，この通知に従って隣接ノードは離脱ノードが存在しない信頼の輪を再構築する．このとき，離脱ノードに属していた子ノードは，隣接ノードの子ノードとして HiHDAM に参加し直す．また，予告なく親ノードが離脱した場合，隣接ノードがノードの離脱を検知し，隣接ノードは離脱ノードが存在しない信頼の輪を再構築し，離脱ノードの子ノードを自身の子ノードとして管理する．最大ハッシュ値が N のとき，親ノードの離脱に必要な通信データ量は $O(log_2 N)$ となる．

子ノードがネットワークから離脱する場合，自身の親ノードに離脱を通知することで，ネットワークから離脱する．離脱通知を受信した親ノードは，その子ノードの公開鍵を破棄する．予告なく子ノードが離脱した場合には，親ノードが離脱を検知し，その子ノードの公開鍵を破棄する．

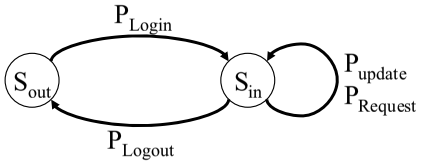

図 3 ノードエージェントの状態遷移図

4. 評　　価

4.1 シミュレータの概要

HiHDAM の特性を評価し，提案手法の有効性を示すために，Java を用いて P2P ネットワークの動作シミュレータを設計・実装した．このシミュレータは，ネットワークに参加するノードをエージェント（ノードエージェント）として実現したもので，ノードエージェント間でメッセージを送受信することにより，ネットワークへの参加時，ネットワークからの離脱時，公開鍵の更新時，及び公開鍵の取得要求が行われた際のノード間の通信をシミュレートする．

図 3 にノードエージェントの状態遷移図を示す．ノードエージェントはネットワークに参加していない状態 (S_{out}) とネットワークに参加している状態 (S_{in}) を持つ．ノードエージェントは状態 S_{out} の時に確率 P_{Login} でネットワークに参加し，状態 S_{in} に移行する．同様に状態 S_{in} の時は確率 P_{Logout} でネットワークから離脱し，状態 S_{out} に移行する．また，状態 S_{in} の時は確率 P_{Update} で自身の公開鍵を更新し，確率 $P_{Request}$ で乱数に従って決定された受信ノードに対して公開鍵を要求するメッセージを送信する．このとき，乱数で決定される受信ノードの割合はベキ分布に従うものとする．これは，一般的に P2P ネットワークにおいて，隣接ピア数とピア数の関係[8] とコンテンツの人気度とリクエスト数の関係[9] がそれぞれべき乗則となることを考慮したためである．

4.2 実験シナリオ

ノードエージェントの設定パラメータとして，2つのパラメータを用意した．実験で用いたパラメータを表 1 に示す．parameter A はノードがネットワークへの参加と離脱を頻繁に繰り返す場面を想定しており，parameter B はネットワークに参加したノードの離脱が

表 1　ノードエージェントの設定パラメータ

parameter	P_{Login}	P_{Logout}	P_{Update}	$P_{Request}$
A	1.0	0.45	0.05	0.50
B	1.0	0.01	0.01	0.98

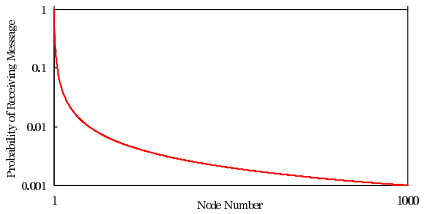

図 4　ノード数 1　　のときのメッセージ受信割合

表 2　メッセージ受信割合におけるノードの分布

histogram	Node Number	
	1–300	301–1000
A	Parent Node	Child Node
B	Child Node	Parent Node

表 3　実験で用いたシナリオ

scenario	parameter	histogram
no. 1	A	A
no. 2	A	B
no. 3	B	A
no. 4	B	B

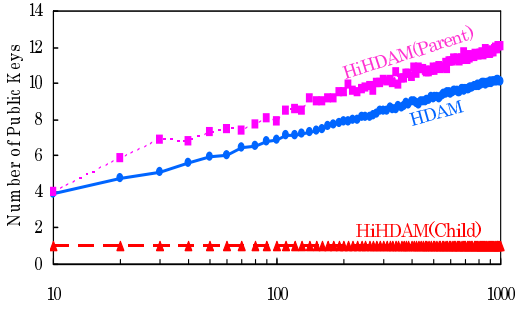

図 5　ノード数と公開鍵数の関係

あまり行われない場面を想定している．

今回，提案手法における親ノードの割合はノード全体の 3 割とする．また，提案手法においては，$P_{Request}$ で生じる公開鍵の要求メッセージの受信ノードの種類の偏りによって，結果にあたえる影響に差異が生じる可能性を考慮し，べき分布における値の大きい上位 3 割が親ノードの場合を histogram A，値の小さい下位 3 割が親ノードの場合を histogram B とし，計測を行った．ノード数が 1000 のときのメッセージ受信割合を示したべき分布を図 4 に示し，表 2 に各分布について示す．

今回，ノードエージェントの設定パラメータとメッセージ受信割合におけるノードの分布の組み合わせを 4 つのシナリオとして用意した．実験で用いたシナリオを表 3 に示す．

4.3　従来手法との比較

提案手法の有効性を示すために，HiHDAM と従来手法 (HDAM) をそれぞれシミュレータ上に実装し，比較評価を行った．

各ノードが管理する公開鍵数

図 5 に，ネットワークに参加しているノードの数と，各ノード 1 台あたりが最低限管理しなくてはならない公開鍵の数の関係を示す．従来手法の場合，ネットワークに参加するノードが増加するに従い，すべてのノードが保持する公開鍵の数も増加していくことがわかる．これに対し提案手法においては，親ノードの保持する公開鍵の数はネットワークに参加するノードの数とともに増加していくが，子ノードが保持しなくてはならない数は，参加ノードの数にかかわらず常に 1 個だけだった．これは，子ノードは自身の親ノードの公開鍵だけを持つことで，ネットワークへの参加が可能となっているためであり，提案手法は低性能端末のノードでも参加できるスケーラブルな分散管理方式であるといえる．

各ノードが処理するメッセージ数

ネットワークに参加しているノード数と各ノード 1 台あたりが処理するメッセージ数の関係を分析する．すべてのノードが図 3 で示した状態遷移を繰り返し行ったとき，各ノード 1 台あたりが 1 ステップあたりに受信したメッセージの個数の平均を，表 3 のシナリオ 1 からシナリオ 4 までのそれぞれについて，図 6 から図 9 に示す．

既存手法では，ネットワークに参加するノードの数が増加した場合や，ノードの参加と離

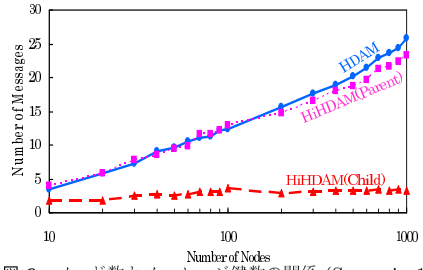

図 6　ノード数とメッセージ鍵数の関係 (Scenario 1)

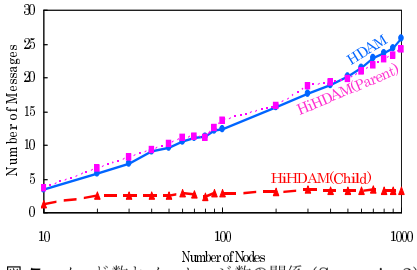

図 7　ノード数とメッセージ数の関係 (Scenario 2)

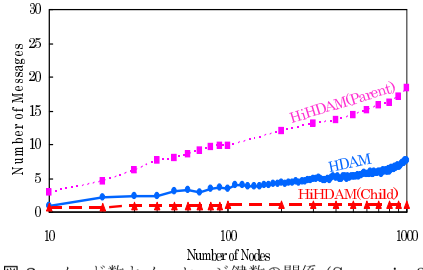

図 8　ノード数とメッセージ鍵数の関係 (Scenario 3)

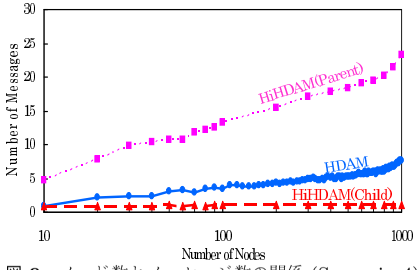

図 9　ノード数とメッセージ数の関係 (Scenario 4)

脱が繰り返されることにより，ネットワークに参加するすべてのノードに生じる負担が増加していることがわかる．これに対し，提案手法の子ノードが処理するメッセージはすべてのシナリオにおいて極めて少なく，提案手法は低性能端末のノードでも公開鍵の確実な配布を受けることが出来るスケーラブルな分散管理方式であるといえる．

5. おわりに

本稿では，信頼の輪と分散ハッシュテーブルを用いてネットワークに参加するノード全体で公開鍵を効率的に分散管理する手法 HiHDAM を提案した．HiHDAM では，ネットワークに参加するノードを階層化することで，センサーや携帯端末などの計算能力が低いノードであっても要求した公開鍵の確実な配布を受けることが出来る．コンピュータシミュレーションにより，HiHDAM は従来手法に比べ，利便性とスケーラビリティに優れていることを確認した．

現在の提案手法では，各ノードがネットワークに参加する際に自身のノードの種類を選択し，その種類は離脱するまで変化しないものであるが，今後は計算機資源に優れたと見なされる子ノードを親ノードに引き上げるなど，ノードの種類を自動的に設定する仕組みに関して検討を行う．

謝辞 本研究の一部は，情報通信研究機構 (NICT) の委託研究「ダイナミックネットワーク技術の研究開発」の助成を受けて実施したものである．

参 考 文 献

1) Housley, R., Polk, W., Ford, W. and Solo, D.: RFC 3280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List(CRL) Profile (2002).
2) Garfinkel, S.: PGP : Pretty Good Privacy, Oreilly and Associates Inc. (1994).
3) Capkun, S., Buttyan, L. and Hubaux, J.-P.: Self-Organized Public-Key Management for Mobile Ad Hoc Networks, IEEE Transactions on Mobile Computing, Vol.2, No.1, pp.52-64 (2003).
4) Kitada, Y., Watanabe, A., Sasase, I. and Takemori, K.: On demand distributed public key management for wireless ad hoc networks, Communications, Computers and signal Processing, 2005. PACRIM. 2005 IEEE Pacific Rim Conference on, pp.454-457 (2005).
5) Goold, J. and Clement, D.M.: Improving Routing Security Using a Decentralized Public Key Distribution Algorithm, Internet Monitoring and Protection, 2007. ICIMP 2007. Second International Conference on (2007).
6) Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Kaashoek, M.F., Dabek, F. and Balakrishnan, H.: Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications, IEEE/ACM Trans. Networking, Vol.11, No.1, pp.17-32 (2003).
7) Takeda, A., Chakraborty, D., Kitagata, G., Hashimoto, K. and Shiratori, N.: Proposal and Performance Evaluation of Hash-based Authentication for P2P Network, IPSJ Journal, Vol.50, No.2, pp.737-749(2009).
8) L.A.Adamic, R.M.Lukuse, A.R.Puniyani and B.A.Huberman: Search in Power-Law Networks, Physical Review E, Vol.64, pp.46135-46143, (2001).
9) A Wierzbicki, N. Leibowitz, M. Ripeanu, and R.Wozniak: Cache replacement policies for peer-to-peer file-sharing　protocols. European Transactions on Telecommunications, Special Issue on Peer-to-Peer Netoworking and Services, vol.15, no.6, pp.559-569, (2004).