社団法人　情報処理学会研究報告
IPSJ SIG Technical Report

2008－MBL－44　(34)
2008－UBI－17　(34)
2008／3／6

# アドホックネットワークにおける隣接テーブルを用いた認証によるノード管理手法

田中 漢人 †　　小野里 好邦 †

† 群馬大学大学院工学研究科

センサネットワークとは、通信機器を備えたセンサ（センサノード）を用いて構成されるネットワークのことを言う。その目的は、ある領域を厳密に測定することにある。

センサネットワークの研究分野として各センサの位置決定がある。センサネットワークにおいて各センサの座標位置がなければセンサで測定した情報が無意味なものとなってしまう。故にセンサの位置は重要ものである。しかしセキュリティを考慮した位置決定法はまだ完全に確立されてはいない。

そこで、ネットワークに対する攻撃に対する防御を考慮したうえで、侵害されたノードが存在する中でもその影響を軽減するような、通信メカニズムを提案し、シミュレーションによりその性能を評価する。

# Node management method using authentication process with neighbor table in ad hoc network

Kunihito Tanaka†　　Yoshikuni Onozato†

† Department of Computer Sience Graduate School of Engineering Gunma University

The network composed with a sensor (sensor node) that has telecommunications equipment is called a sensor network. The purpose is in the strict measurement of a certain area.

There is a positional decision of each sensor as a field of research on the sensor network. Information measured with the sensor becomes meaningless if there is no coordinates position of each sensor on the sensor network. Therefore, the position of the sensor is the important one. However, the positional decision method to consider security has not been established still completely.

Then, it proposes the communication mechanism that reduces especially the influence where the violated node exists after the defense to the attack to the network is considered, and the performance is evaluated by the simulation.

## 1　研究背景

センサネットワークとは通信機器を備えたセンサ（センサノード）を用いて構成されるネットワークでのことである。人の入り込めない場所、及び入り込むのが困難な環境の状態をリアルタイムに測定し、測定結果をオンデマンドで手にいれ分析を行うのに適している。センサネットワークでは、センサノードを非常に多数、配置することを前提とすることが多い。そのような環境でセンサを多数配置し、自律的に、自己組織的にネットワークを形成させようと様々な研究がされている。

センサネットワークの研究分野としてノードの位置決定法に関する研究がある。あるエリアにセンサを配置したときセンサの位置がわからなければ得た情報を有効に活用することができない。故にセンサの位置を特定するメカニズムが必要である。しかし GPS(Global Positioning System) などの機能を全てのセンサに装備することは高価なため現実的ではない。そのため他の位置決定法が必要となる。

センサネットワークでは他のノードと協調して位置決定を行うのが主流となっているが、ノードは安全な環境に配置されるとは限らないため、敵から攻撃を受けた場合ネットワークノードの配置が崩壊してしまう危険性がある。そうならないためにセキュリティを導入するが、既存のセキュリティプロトコルでは高価なため実装することは難しい。

極力、センサハードウェアのリソースを使用せず、セキュリティを高める提案を本研究では行う。

本研究の目的は、暗号化に頼らずセキュリティメカニズムを構築することによりセンサネットワークにおけるセキュリティを高めることである。

既存のセキュリティプロトコル以外のセキュリティプロトコルが必要となるが、確立されたプロトコルはいま

だなく、現在研究がおこなわれている。その主な方法は暗号化である。暗号鍵方式には大きく非対称鍵暗号方式と対称鍵暗号方式とがあるが、非対称暗号鍵方式は高価であるためセンサネットワークのようなリソースの限られたシステムには適さない。よって多くの研究では対称鍵暗号方式を採用している。対称鍵暗号方式のみを使用し、堅固な暗号方式を構築する研究も活発に行われているが、暗号鍵が敵に知られてしまった場合ネットワークの崩壊を防ぐ術はない。

そこで本研究では暗号鍵が破られる場合を想定し、その状況下でもネットワークを機能させる方法として、隣接テーブルを用いた認証という概念の導入を提案する。認証の際の基本方針は、自身が認証しているノードが認証しているならば、そのノードは信頼できるノードであり認証する、そうでなければ認証しないというものである。本論文では、この概念を取り入れたネットワークを提案し、シミュレーションによりその性能を評価する。

本論文における各節の概要は以下のようになっている。1 節では本研究の背景および概要を述べた。2 節では既存の位置決定法などの関連研究について述べる。3 節では認証を用いたノード管理手法を提案する。4 節ではシミュレーションを行いその性能を評価する。5 節でまとめと今後の課題について述べる。

# 2 関連研究

本節ではセンサネットワークにおける研究分野である、ノードの位置決定法について述べ、その後ネットワークへの攻撃の方法について述べる。

## 2.1 ノードの位置決定法

センサネットワークにおいてセンサノードの位置決定が重要であると先に述べたが、すべてのノードに GPS を用いることなく位置決定を行う方法は様々ある [1] が、大きく 2 種類に分けられる。以下にそれを示す。

- Range-based localization ··· あるセンサノードが隣接しているノード間で距離・角度を計測することにより自身の位置を推定する方法である。Range-based localization の方法には、RSSI、TDOA 等がある。
- Range-free localization ··· 各センサノードが距離・角度を計測せず、自身の位置を推定する方法である。Range-free localization の方法には、SeRLoc[5] や、APIT[6]、DV-hop[7]、GOMASHIO[8] 等がある。

以上のような位置決定法があるが、Range-based localization による位置決定法の場合どうしても各ノードに特別な装置を装備しなければならない。例えば、信号強度を測定する機能や、二つの違う無線電波を送出する機能を持つ装置が必要となる。故に、Range-free localization による位置決定法が各センサノードの単価の観点からセンサネットワークに相応しいだろうと考える。

## 2.2 ネットワークへの侵害方法

ここでは、センサネットワークの位置決定において行われる恐れのある攻撃方法を紹介する。その攻撃は p2p ネットワーク、アドホックネットワーク等の分散システムに共通な攻撃である。センサネットワークにおけるルーティングに対して行われる攻撃方法としては Wormhole Attack[2, 3] や Sybil Attack[2, 4] がある。

Wormhole Attack とは敵が確立した Wormhole Link なるものの影響により、他のエリアの正当なノードのメッセージが聞こえてしまうというものである。センサノードの位置決定において各センサノードは周りのメッセージを受信して位置決定を行うので他のエリアのメッセージが聞こえてしまっては、全てのノードが自身の位置を誤って認識することになる。この攻撃は正当なノードのメッセージを受け取るので検出が難しい。

Sybil Attack とは敵が一つ以上のハードウェアから多数の偽造ノード ID を発行し、ある正当なノードの周りに多数のノードが居るように見せかける攻撃である。

これらの攻撃に対するセキュリティを考慮した通信メカニズムとしては LITEWORP[3] や SeRLoc[5] がある。

LITEWORP はノード配置時に隣接していたノードとのみ通信を行い、さらに規定外の行動を繰り返すノードをネットワークから排除することで Wormhole Attack を防いでいる。本研究は LITEWORP の考えを元に発展させノードが移動する場合に対応させている。

SeRLoc は locaor と呼ばれるリソース制限が弱く絶対座標を求めることができるノードをネットワークに取り入れ、鍵暗号や一方向性関数、ナンス値などを駆使し侵害ノードを見分けている。

# 3 提案手法

本節では、提案手法である認証を用いたノード管理手法について述べる。そのため、まず、隣接テーブルを作成する方法を述べ、その後認証を取り入れた隣接テーブルの更新方法について述べる。

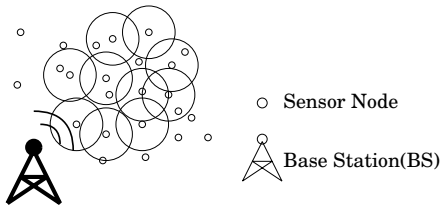


図 1　センサネットワーク

## 3.1　ネットワークモデル

本研究が想定しているネットワークモデルについて述べる。本研究でのネットワークの構成は、図 1 に示されるように、多数のセンサノードと一つの基地局を与える。センサノードはある特定の領域にランダムに配置され、自由に動くことができると仮定する。また、全てのセンサノードは全指向性アンテナを装備し、通信範囲 R は同一とする。

特別な場合にのみ、基地局から全ネットワークにメッセージがブロードキャストされるが、各ノードはたいていの場合、隣接しているノードとしか通信を行わないものとする。各ノードは、自分のセンシングで得た情報を基地局へ転送するが、その際マルチホップ通信によりメッセージを届けるものとする。

各ノードは、以下の隣接テーブルを作成し所持する。各テーブルは主ノードの通信範囲内にいる他のノードの ID を表にしたものであり、その作成、更新の方法を 3.2.1 節で説明する。

- 存在テーブル ··· 自身の通信範囲内に存在しているノードのテーブル
- 認証テーブル ··· 自身の通信範囲内に存在している認証ノードのテーブル
- 完全認証テーブル ··· 自身と自身の認証ノードの認証テーブルを結合したテーブル

各ノードは移動の際に各テーブルの更新をし、その際に新たなノードを発見した場合には認証によりその新ノードを認証テーブルに含めるかどうかを決める。このときの基準は完全認証テーブルを参照し、自身の認証ノード内に新ノードを認証しているノードが存在しているかどうかによって決める。認証しているノード（共通認証ノード）が存在していれば新ノードを自身の認証ノードへ加え、存在していなければ自身の認証ノードへは加えないものとする。認証を取り入れることでノードは以下の 3 種類に分類できる。

- authenticated node··· 認証されネットワークとして機能しているノード
- unapproved node··· 認証されずに孤立したノード
- isolated node··· 物理的に孤立したノード

基地局では、その認証テーブルを全てのノードから集め、統合することにより全てのセンサノードの位置を決定できると仮定する。

## 3.2　隣接テーブルの作成と更新

### 3.2.1　隣接テーブルの作成

この方法は各テーブルの作成および存在テーブルの更新に用いられる。ノード $n_0$ が各テーブルを作成するときを例にあげて説明する。実際のネットワークノード配置を図 2 に示す。

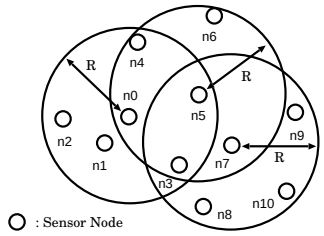


図 2　ノードの初期配置

1. $n_0$ はビーコン信号をブロードキャストする。
2. ノード $n_0$ のブロードキャストを聞いたノード $n_1, n_2, n_3, n_4, n_5$ は、自身の ID を付け加え、すぐに応答する。
3. それを受信したノード $n_0$ は、周りにノード $n_1, n_2, n_3, n_4, n_5$ があると知り、表表 1 のような隣接テーブルを作成する。これが存在テーブルとなる。

| 主ノード | 存在ノード |
|---|---|
| $ID_0$ | $ID_1, ID_2, ID_3, ID_4, ID_5$ |

表 1　ノード $n_0$ の存在テーブル

4. 次にノード $n_0$ は隣接しているノードに対して、認証テーブルを自身 $(n_0)$ に送信するよう要求する。
5. そのリクエストを受け取ったノードは、自身が保持している認証テーブルをノード $n_0$ へ転送する。持っていなければ、ノード $n_0$ のように手順 1 から順序よく存在テーブルを作成していき、手順 3 が終わると、存在テーブルを認証テーブルへとコピーする。その後 $n_0$ へ認証テーブルを転送する。

6. 周りのノードの認証テーブルを受け取ったノード $n_0$ は自身と認証ノードの認証テーブルを結合し完全認証テーブルを作成する。表 2 にノード $n_0$ の完全認証テーブルを示す。

| 主ノード | 認証ノード |
|---|---|
| $ID_0$ | $ID_1, ID_2, ID_3, ID_4, ID_5$ |
| $ID_1$ | $ID_0, ID_2, ID_3, \cdots$ |
| $ID_2$ | $ID_0, ID_1, \cdots$ |
| $ID_3$ | $ID_0, ID_1, ID_7, ID_8, ID_5 \cdots$ |
| $ID_4$ | $ID_0, ID_5, ID_6, \cdots$ |
| $ID_5$ | $ID_0, ID_3, ID_4, ID_6, ID_7 \cdots$ |

表 2　ノード $n_0$ の完全認証テーブル

このようにして完全認証テーブルを作成し、BS から認証テーブルの要求があった場合には、BS へ認証テーブルを送信する。周りの認証ノードが変化した場合には、3.2.2 節に基づいて認証テーブル及び完全認証テーブルの更新を行う。なお、存在テーブルの更新は手順 1 から手順 3 を用いて更新する。　ネットワークへのノード配置後、各ノードがすぐに上のアルゴリズムを実行すれば、その時点で全ての存在テーブルは正当であるといえる。そこでその時点での存在テーブルを認証テーブルへとコピーする。ノードが移動した場合、各テーブルを更新しなければならないが、敵の影響により侵害されたノードがネットワークに入り込んでいるかもしれない。そのため各ノードは、安全に認証テーブルの更新を行わなければならない。3.2.2 節で認証テーブルの更新方法について述べる。

### 3.2.2　認証テーブルの更新

ここでは、ノードの移動に伴う認証テーブルの更新について述べる。ノードが移動した後、各テーブルを更新しなければならないが先のアルゴリズムの手順 1 から 3 ように作成すれば、安全に作ることはできない。なぜなら、ネットワーク内に侵害されたノードが存在しているかもしれないからである。故に、新しく通信範囲内に入ってくるノードに対して認証を行う。

本研究で認証に必要なことは、共通認証ノードの存在と各ノードが自身の通信範囲内にいるノードを常に監視することである。あるノードが移動し、他のノードの通信範囲内に入るとき、両方のノードと認証しているノードつまり共通認証ノードが無ければ、正当なノードと認めないものとする。

例として、図 3 から図 4 のようにノード $n_0$ が移動した場合を考える。図 3 では、ノード $n_0$ が移動するが、他のノードは移動していないとする。移動の結果、ノード $n_0$ が図 4 の位置まで移動してきたとする。

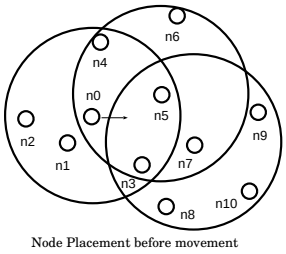


図 3　移動前のノード配置

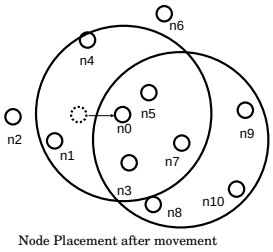


図 4　移動後のノード配置

1. ノード $n_0$(移動するノード) は移動前にこれから移動することをブロードキャストで周囲のノードへ通知した後移動する。
2. 受信したノード $n_0$ の隣接ノードは移動後にノード $n_0$ と隣接しているかどうかを確認し、隣接していないならば自身の各テーブルからノード $n_0$ を削除し、削除したことを隣接ノードに通知する。
3. ノード $n_0$ は移動後に存在ノードの更新を行う。
4. ノード $n_0$ は、新たなノードの通信範囲内に入ったことを感知し、自身の前回の完全認証テーブルを参照し、移動する前に認証していたノードがノード $n_7$ を認証していたかを調べる。
   つまり、移動前のノード $n_0$ とノード $n_7$ との共通認証ノードが存在するか調べる。
5. 共通認証ノードが存在すれば、ノード $n_7$ を新ノードとして認め認証テーブルに加え、共通認証ノードが存在しなければ、ノード $n_7$ を自身の認証テーブルに加えない。
6. 認証ノードに変化があればその変化を表 3 の変更テーブルを用いて周囲へとブロードキャストする。
7. ノード $n_1, n_4, n_3, n_5$ がこれを聞き、ノード $n_1, n_4$ が自身の完全認証テーブルを更新する。
8. ノード $n_7$ も同様にして新ノードをノード $n_0$ とした変更テーブルを作成、ブロードキャストする。

| 新ノード | 共通認証ノード |
|---|---|
| $ID_7$ | $ID_3, ID_5$ |

表 3　変更テーブル

9. ノード $n_7$ のもともとの認証ノードである、ノード $n_3, n_5, n_8, n_9, n_{10}$ がこれを聞き、ノード $n_8, n_9, n_{10}$ が自身の完全認証テーブルを更新する。
10. ノード $n_3, n_5$ は二つの変更テーブルを受け取って初めて自身の完全認証テーブルを更新する。
11. ノード $n_0, n_7$ はお互い、更新した認証テーブルを交換する。

以上の手順で隣接ノードの認証を行うことで攻撃の影響を軽減できることを 3.3 節で述べる。

## 3.3　侵害されたノード

ノードが侵害された場合、以下の二つのことに焦点を絞る。

- 一つのノードが侵害されたときの影響
- 二つのノードが侵害されたときの影響

### 3.3.1　一つのノードが侵害されたときの影響

あるノード $n_x$ が侵害され、挙動が侵害者に操られていると仮定し、実際のネットワークのノード配置を図 5 とする。　ノード $n_x$ が侵害される前に、全てのノードの完全認証テーブルはすでに完成しており、正当である。表 6 に、ノード $n_x$ の本来の完全認証テーブルを示す。

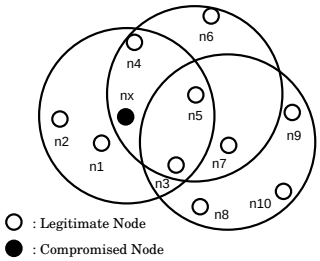


図 5　実際のノード配置図

この場合、侵害者の目的は、侵害されたノードを使用して偽造した認証テーブルを周りに送信する、もしくは BS へ送信することとする。

認証テーブルの更新は、3.2.2 節に従い、更新をする上で、変更テーブルをブロードキャストしなければならない。仮に従わず、ノード $n_x$ が偽造した認証テーブルを BS へ送信しようとした場合、周りのノードはそのメッセージが以前の認証テーブルと異なることを感知する。故に ノード $n_x$ の全ての認証ノードはメッセージを次のノードへ中継せず、ノード $n_x$ へ警告する。よって、認証テーブルの更新は、3.2.2 節に従わなければならない。

| 主ノード | 認証ノード |
|---|---|
| $ID_x$ | $ID_1, ID_2, ID_3, ID_4, ID_5$ |
| $ID_1$ | $ID_x, ID_2, ID_3, \cdots$ |
| $ID_2$ | $ID_x, ID_1, \cdots$ |
| $ID_3$ | $ID_x, ID_1, ID_7, ID_8, ID_5 \cdots$ |
| $ID_4$ | $ID_x, ID_5, ID_6, \cdots$ |
| $ID_5$ | $ID_x, ID_3, ID_4, ID_6, ID_7 \cdots$ |

図 6　ノード $n_x$ の本来の完全認証テーブル

そこで、ノード配置を狂わせるため任意のノード $n_y$ を選択し、新ノードとして自身の認証テーブルに含めたいとする。このとき実際には、その任意のノード $n_y$ はノード $n_x$ から遠く離れているとする。

ノード $n_x$ は、3.2.2 節の手順 5 で、変更テーブルを偽造し、新ノードとしてノード $n_y$ を含める際に共通認証ノードの存在が必要となる。ここで注意することはノード $n_x$ の認証ノードはノード $n_x$ の一番最近の認証ノードが誰かを知っているということである。

ここで、以下のように共通認証ノード $n_z$ をも任意にでっち上げ、変更テーブルをブロードキャストした場合、周りのノードは以前のノード $n_x$ の認証ノードを知っているため、そのでっち上げた共通認証ノード $n_z$ は存在せず、偽造メッセージだと結論づける。周りのノードはノード $n_x$ への警告を行う。

また、ノード $n_x$ が本当に認証しているノードが新ノードを認証していないにもかかわらず、つまり共通認証ノードではないにもかかわらず共通認証ノードとして $n_3$ とでっち上げ、変更テーブルをブロードキャストしたた場合、ノード $n_3$ はノード $n_y$ と実際には認証していないため、認証していないことを知っているノード $n_3, n_4, n_5$ がノード $n_x$ に警告する。例え、ノード $n_y$ が実際に近くにあり、ノード $n_3$ を認証していたとしても、ノード $n_x$ のビーコン信号に対するノード $n_y$ の応答をノード $n_3$ が聞いていないため、ノード $n_3$ がノード $n_x$ に警告する。

また、本来のネットワーク配置が図 5 であるにも関わらず、削除ノードとしてノード $n_2$ を指定した場合、ノード $n_2$ はノード $n_x$ に警告し、素直に自身の認証テーブルから $n_x$ を削除する。そして、ノード $n_x$ を削除したことを周囲のノードに通知する。これによる、ネットワーク配置の影響は少ないと考えられる。

逆の場合を考える。実際にはノード $n_2$ がノード $n_x$ の

通信範囲から離れたにも関わらず、ノード $n_x$ が削除ノードとしてノード $n_2$ を指定しなかった場合は、ノード $n_2$ とノード $n_x$ の共通認証ノードであるノード $n_1$ がノード $n_x$ のビーコンに対するノード $n_2$ の応答を聞いていないためノード $n_x$ に警告する。

以上のようにして、ノードが一つ侵害された場合のネットワークの影響を少なくすることが出来る。

### 3.3.2 二つのノードが侵害されたときの影響

図 7 を例として使用する。

まず、エリア A ではノード $n_x$ が、エリア B ではノード $n_v$ が侵害され、二つのノードで Wormhole Link を確立したとする。共謀するためには、侵害されたノードだけの通信リンクが必要だからである。

ノード $n_x$ の挙動に注目する。ノード $n_x$ は、ノード

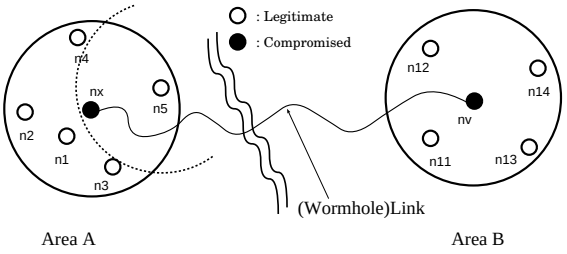


図 7　二つのノードが侵害された場合

$n_v$ と隣接していることを主張したい。つまり、ノード $n_x$ は認証テーブルにノード $n_v$ を加えるため、3.2.2 章の手順により認証を行い、認証テーブルを更新する必要がある。

そこでノード $n_x$ は、認証テーブルを改変し、新たにノード $n_x$ の認証ノードに加わったノード $n_v$ が正当なものであると周りのノードを騙すため、ノード $n_v$ との共通ノードを変更テーブルに付加する。この時、ノード $n_x$ が本当に認証しているノードを共通認証ノードとして指定しなければ、周りのノードは偽造メッセージだと気付く。よって、正当なノードであるノード $n_5$ の ID を共通認証ノードに含めるとする。

これをブロードキャストするが、当然ノード $n_5$ はノード $n_v$ を認証していないので、ノード $n_5$ は偽造メッセージだと気付く。

ノード $n_5$ は、$n_x$ が嘘をついていると判断し、ノード $n_x$ への警告をおこなう。一方、エリア B のところでもノード $n_v$ の周りのノードがノード $n_v$ に対する警告をおこなうだろう。　このようにして、ネットワーク内に侵害されたノードが二つあった場合も、そのようなノードを分離してネットワークの整合性を保つことができる。

ただし、述べたようなメカニズムでは侵害されたノードが検出できない状況がある。それは侵害されたノード同士が比較的近くにいるときである。

例えば、図 8 のように侵害されたノードがお互い近く

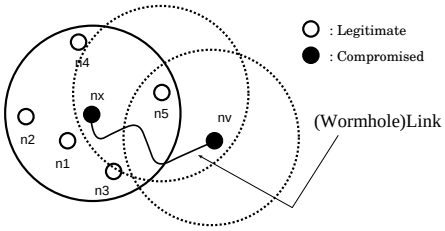


図 8　検出できないときの状況

に居るときその共通認証ノード $n_5$ は異常に気づけない。なぜなら、怪しい挙動を見せないためである。

前に述べたように、ノード $n_x$ は認証テーブルを更新するためにまずビーコン信号を送出する。Wormhole Link でノード $n_v$ と繋がっているため、ノード $n_v$ はノード $n_x$ がビーコン信号をブロードキャストしたことを知ることができる。そして $n_v$ は、あたかもノード $n_x$ のビーコン信号が届いたかのように振る舞い、ノード $n_x$ へビーコン信号の応答をする。もちろんこの応答は届かないが共通認証ノードであるノード $n_5$ を騙すための演技のため応答を返すのである。このような状況でノード $n_5$ がノード $n_v$ の演技を見破ることは出来ないだろう。

このような状況への対策は今後考慮する必要がある。

## 4　シミュレーション

本節では先に述べた提案手法に基づいてシミュレーションプログラムを作成し、その結果を考察する。

### 4.1　シミュレーション

先に述べた提案手法に沿って WindowsXP Professional 上で Java 言語を用いてシミュレーションプログラムを作成した。

シミュレーションは以下のように動作する。ある大きさのエリアにノードの座標をランダムに割り当て、座標の比較を行い、ある通信範囲内にいるノードを存在ノードとしてテーブルに記録する。ある時間にノードは一つしか動かない様にし、ノードが移動するとその近傍にいるノードがノード認証を行い、共通認証ノードがあれば認証テーブルの更新を行う。この一つのサイクルが終わってから、次の移動するノードを一様にランダムに決め同

様のことを行わせた。この操作を 10000 回行い、その後各ノードが 3 種類のどれに該当するかを確認しそれぞれの合計を出し 1 回の試行とした。同一のパラメータで 10 回試行し、その平均値を評価に使用した。

なお、今回は単純化のためネットワークへの侵害は発生しないものとし、提案手法を用いることによるネットワークの変化を評価する。　シミュレーションに使用するパラメータは表 4 のとおりである。

| | |
|---|---|
| ノード数 | 100,20,···,1100 |
| ネットワークエリア | 100 |
| 通信範囲 | 1,2,···,10 |
| 移動範囲 | 0,1,2,···,10 |
| 時間 | 10000 |

表 4　パラメータ

表 4 の各パラメータの説明をする。ネットワークエリアは 100 となっているが X 軸が 100 の長さ、Y 軸が 100 の長さの正方形状のエリアを意味する。このエリア上の座標をノードに一様にランダムに割り当てるのである。通信範囲は各ノードが使用する無線到達距離を表している。隣接ノードはこの通信範囲内のノードのことをいう。移動範囲は各ノードが X 軸、Y 軸それぞれに一度に移動できる範囲を示している例えば移動範囲 5 の場合は、各ノードは移動する際、X 軸、Y 軸をそれぞれ一様に-5,-4,···,5 の範囲で移動するものとする。時間の 10000 とはランダムに 10000 回ノードを選択し、移動させていく事を表している。

各パラメータを変化させていき、各種類ごとのノード数を評価する。評価指標として主に unapproved node の数を使用した。この値は提案手法によってネットワークに加われなくなったノードの数であり、提案手法によるネットワークの機能低下を示す数値となるからである。

なお、移動範囲 0 のときは移動しないので unapproved node の数は 0 となる。

## 4.2　実験結果

シミュレーション実験の結果は以下のようになった。

まず図 9、図 10 は共に縦軸に unapproved node の数、横軸に通信範囲をとり、移動範囲を変化させたときのグラフで、ノード数は図 9 が 500、図 10 が 1000 となっている。

これをみると通信範囲の拡大に伴い、始めは unapproved node の数が増加していき、あるところで今度は減少していることがわかる。そしてノード数が多いほど、

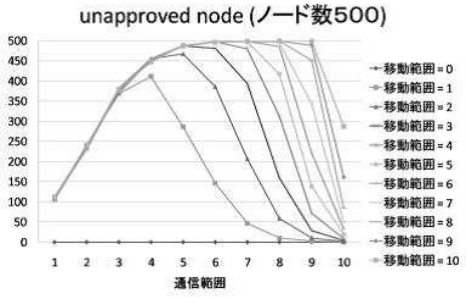


図 9　ノード数 500 の時の unapproved node の数

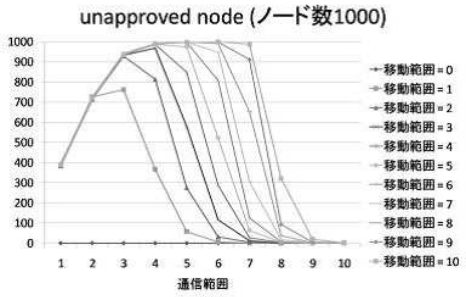


図 10　ノード数 1000 の時の unapproved node の数

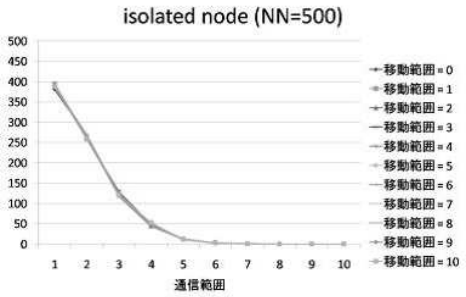


図 11　ノード数 500 の時の isolated node の数

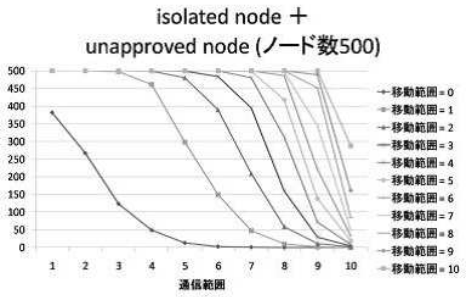


図 12　ノード数 500 の時の isolated node ＋ unapproved node の数

移動範囲が狭いほど unapproved node の減少は通信範囲の狭いところから始まることがわかる。

これは、ノード数が多いほど 1 ノードの通信範囲内に存在するノードの密度が上がり共通認証ノードを発見しやすくなるためである。また、移動範囲が広くなると、一度に遠くまで移動できてしまうため共通認証ノードが発見しづらくなることが考えられるためである。

次に、始めは unapproved node の数が増加していき、あるところで今度は減少している理由について考えてみる。まず図 11 はノード数 500 で縦軸を isolated node したものである。これをみると、isolated node は移動範囲に依存せず、通信範囲に依存していることがわかる。

そして isolated node と unapproved node を足したものが図 12 である。これはセンサネットワークとして機能していないノードの総数となる。これをみると、通信範囲の狭いうちはすべてのノードが機能していないが、通信範囲の拡大に伴い移動範囲の狭いものから機能し始めることがわかる。また、図 9 と図 11 を見比べると、図 9 での unapproved node の減少は図 11 の減少と一致することがわかる。　このことから、通信範囲の狭い時の unapproved node の増加は通信範囲の増加により isolated node が減少した分増加していることがわかる。そしてさらに通信範囲の拡大したことにより共通認証ノードを発見しやすくなるため unapproved node が減少したと考えられる。

以上のことから、共通認証ノードによる認証を取り入れた提案手法では、通常よりも広い通信範囲が必要であることがわかった。通信範囲の拡大は 3.3.2 節で述べた侵害されたノードが検出できない状況になる可能性が増大する。これを防ぐにはノード数を増やすことによりネットワークを機能させるのに必要な通信範囲を狭くするする方法がある。例えば図 9 と図 10 を比較するとノード数 500 ではほぼすべてのノードを機能させるには通信範囲が 10 では足りないが、ノード数 1000 では通信範囲 9 以上でほぼ全てのノードを機能させることができる。このようにノード数を増大させることにより通信範囲の拡大を抑えることができる。

## 5　まとめと今後の課題

本論文ではセンサネットワークの通信に共通認証ノードを用いた認証を取り入れることを提案し、シミュレーションによってその性能を評価した。その結果すべてのノードをネットワークとして機能させるには通常よりも多くのノードまた広い通信範囲が必要であることがわかった。今後の課題として、実際に侵害するノードを織り交ぜてのシミュレーションや、侵害に気づけない状況への考慮、通信コストを交えたシミュレーションとコストを抑えたアルゴリズムの考案がある。また、今回のシミュレーションではネットワークの分断を考慮にいれていないので、それを考慮する必要もある。

## 参考文献


[1] 宇地原 直史 “センサネットワークにおけるセキュリティを考慮した位置決定法の検討”, 平成 18 年度, 群馬大学工学部情報工学科 卒業論文, 平成 19 年 3 月 17 日.

[2] C. Karlof and D. Wagner, “Secure routing in wireless sensor networks: Attacks and countermeasures”, In First IEEE international Workshop on Sensor Network Protocols and Applications, May 2003.

[3] I, khalil, S. Bagchi, and N. B. Shroff, “Liteworp: A lightweight countermeasure for the wormholw attack in multihop wireless neworks”, In Proceedings of the International Conference on Dependable Systems and Networks (DSN), June-July 2005.

[4] J. Newsome, E. Shi, D. Song and A. Perrig, “The Sybil Attack in Sensor Networks: Analysis and Defenses”, In Proc, of IPSN 2004, Berkeley, CA, April 2004.

[5] L. Lazos and R. Poovendran, “SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks”, in Proceedings of WISE, Philadelphia, PA, Oct, 2004.

[6] T. He, C. Huang, B. Blum, J. Stankovic and T. Abdelzaher, “Range-Free Localization Schemes in Large Scale Sensor Network”, In Proc. of MOBICOM 2003, san Diego, CA, USA, September 2003.

[7] D.Niculescu and B. Nath, “DV Based Positioning in Ad hoc Networks”, In Journal of Telecommunication Systems, 2003.

[8] 岩谷晶子, 西尾信彦, 村瀬正名, 徳田英幸, GOMASHIO: アドホックセンサネットワークにおけるノード位置特定方式, 情報処理学会, モバイルコンピューティングとワイヤレス通信研究会 Vol.2001(108), pp.22-30, 2001