情報処理学会論文誌 Vol. 50 No. 2 787–797 (Feb. 2009)

無線センサネットワークにおける不正メッセージ作成元ノードの検知

清 雄一[†1] 本位田 真一[†2,†1]

大規模な無線センサネットワークでは，攻撃者がセンサを物理的に取得しセンサを不正に操作する脅威が考えられる．不正ノードは，自ら不正メッセージを発生させたり，転送されてきた正しいメッセージを改ざんしたりする等，不正メッセージの作成を行うことができる．本論文は，このように不正メッセージを作成し，そのメッセージをシンクに転送する不正ノードの特定を目的とする．既存研究は，ただ 1 つのメッセージ発生元ノードがあり，そのノードからシンクまでのルーティングパスが固定されている環境にのみ適応できる手法である．無線センサネットワークにおいては，ノードの故障率が高いため，この制限は大きな問題となる．また，既存研究は不正ノードを特定するまでに数多くの不正メッセージを必要とする．提案手法では，メッセージの転送ノードが，ノード ID と k ビットの Message Authentication Code（MAC）をメッセージに付加する．k の値が小さい場合は，トラフィック量を抑えることが可能だが，攻撃者が正しい MAC をねつ造できる可能性が高まる．だが，つねに正しい MAC を作成できるわけではない．本手法では，統計的手法を用いることにより，不正ノードである可能性が高いノードを特定する．また，論理ノードを導入することにより，ルーティングパスが頻繁に変わる環境にも対応する．数学的な解析やシミュレーションにより，既存研究と比較して少ない不正メッセージ数から不正ノードを検知できることを示す．

Detecting Sensor Nodes Creating False Messages in Wireless Sensor Networks

Yuichi Sei[†1] and Shinichi Honiden[†2,†1]

The sensor nodes in large scale sensor networks are at high risk of being captured and compromised. A compromised node can be used to create false messages by generating them on their own or by fabricating legitimate messages received from other nodes. Our goal is to locate the compromised nodes that create false messages and send them to the sink. Existing works can only be used in situations where there is one source node and a routing path from it to the sink is static. This limitation is a big problem in wireless sensor networks because of node failures. They also must receive a lot of false messages before they can locate a compromised node. In our method, each forwarding node appends its ID and k-bit message authentication code (MAC). If we set k to be small, we can reduce communication traffic. Although ttackers can create legitimate MACs with high probability in this case, they cannot create legitimate MACs every time. We detect compromised nodes by statistical method. Our method can be used in dynamic environments and can detect compromised nodes faster, because it requires the recognition of less false messages. Our mathematical analysis and the simulations we conducted prove the effectiveness of our method.

1. はじめに

無線センサネットワーク（WSN）を利用する主な目的は，イベントを検知し，ユーザに通知することである．森林火災や侵入者検知等に用いることができる．また，センサネットワークは広範囲に数多くのノードを設置することにより構成されており，興味のあるイベントを検知したならば，無線を通じて，**シンク**までマルチホップで通知する．しかし，WSN においては攻撃者がセンサノードを物理的に取得し，不正に操作する脅威が存在する．攻撃者は不正に取得したノードから，ノードの秘密鍵等のすべてのデータを取得することができる．また，悪意のあるコードを埋め込むことも可能となる．これにより，攻撃者は，不正メッセージを自ら発生させたり，他のノードから転送されてきたメッセージを改ざんしたりする等の，不正メッセージの作成を行うことができる．

現在，このような不正メッセージの検知に関しては様々な研究が行われている[7),10),17),19),22),25)]．だがこれらの研究では，不正メッセージの検知はできるが，どのノードが不正メッセージを作成したのかの特定を行うことはできない．不正ノードの検知の手法としては，次の 3 つの手法に分類することができる．コードの完全性の検証，ノードどうしによる監視，シンクからのトレースバックである．コードの完全性の検証という手法は，文献 11)，18) 等，主に challenge-response プロトコルを利用している．この手法は，通常，「怪しいノード」を検知した後，そのノードに対して不正ノードであるかどうかを判定するために用いられる．

†1 東京大学
The University of Tokyo
†2 国立情報学研究所
Nnatinal Institute of Informatics

我々が提案する手法では，シンクにおいて，不正ノードを高確率で検知できるものであり，つまり，怪しいノードを検知することができる．したがって，我々が提案する手法とコードの完全性の検証の手法は同時に利用することができる．ノードどうしによる監視を行う手法は，共謀攻撃に弱いという欠点がある．なぜなら，監視を行うノード自体も，攻撃者によって操られている可能性があるからである（3 章で議論を行う）．3 つめの手法は，シンクからのトレースバックである．現在 WSN で提案されている手法[20),23)]は，メッセージ発生元ノードが 1 つだけであり，そのノードからシンクまでのパスが固定されているという限られた環境においてしか利用することができない（このことは文献 20)，23) において記述されている）．だが，WSN においては，ノードの故障率が高いことを通常想定するため[15)]，この特性は大きな制約となる．

本論文の目的は，不正メッセージを作成しシンクまで転送する，不正ノードを検知することである．不正メッセージの検知自体は本論文の対象外であるが，上述したような手法を用いることができる．我々は不正ノードを検知するために，パケットマーキングの手法を用いる．メッセージを転送する各ノードは，k ビットの *Message Authentication Code* (*MAC*)[6)] をメッセージに付加する．通常，MAC のビット長は 64 ビットから 128 ビット程度である[3)]．すべてのメッセージのすべての転送ノードが 64 ビットの MAC を付加すると，発生するトラフィック量がセンサネットワークでは実現困難なほどに膨大になってしまうため，このままではセンサネットワークで利用することはできない[20)]．提案手法ではトラフィック量を削減するため，MAC のビット長である k の値を小さな値（たとえば 1 ビット）に設定する．もし k の値が小さいと，攻撃者が高い確率で MAC を正しくねつ造することが可能となる．だが，すべての MAC を正しくねつ造するこはできない．したがって，いくつかの不正メッセージを受け取ることにより，統計学的手法を用いることで，不正ノードの検知が可能である．また，提案手法の有効性をシミュレーションによって検証する．

本論文の構成は下記のとおりである．2 章において，想定環境を述べる．関連研究を 3 章で紹介し，その課題を議論する．4 章では，我々が提案する手法を述べ，5 章において提案手法の評価を行う．提案手法に関する議論を 6 章で行い，最後に 7 章において本論文の結論を記す．

2. 想 定 環 境

本章では，想定する WSN のモデルと，不正メッセージ攻撃の定義を行う．

2.1 WSN のモデル

多数の小さなセンサノードから成り立つ WSN を想定する．ノードはユーザが指定したイベントを検知することができる．イベントを検知したノードはそれをシンクに通知する．本研究においては，1 度配備された後，センサノードは移動しないものとする．また，コストの制約から，センサは耐タンパハードウェアを装備していない．センサはバッテリ駆動であり，計算性能は低いものとする．また，本研究においては，ノードから発生するメッセージの宛先はシンクであるとする．いい換えると，本研究における目的は，不正メッセージを作成し，それをシンクに送る不正ノードを検知することである．イベント情報を収集するシンクは，十分な計算性能とデータストレージを持ち，セキュリティ侵害を受けないものとする．

2.2 不正メッセージ攻撃

攻撃者は WSN 内のセンサノードをセキュリティ侵害することができる．センサがセキュリティ侵害されると，秘密鍵等センサ内のすべてのデータが攻撃者に漏えいする．また攻撃者はセンサに悪意のあるコードを埋め込み，不正メッセージを作成することができる．不正メッセージの作成とは，自ら不正メッセージを発生させることと，転送されてきたメッセージを改ざんすることを意味する．このような不正メッセージは，ユーザに誤った情報を与えることになる．また，センサの有限なリソース（バッテリ等）を無駄に費やすことにもなる．したがって，このような不正ノードをできる限り早く検知することが求められる．セキュリティ侵害を受けたノードによる脅威は，正しいイベントの破棄等も存在する．だがこれらの脅威は他の研究[5),15),21)] 等で対策をとられており，本論文中では焦点を当てない．

3. 関 連 研 究

本章では，不正ノードの検知に関する関連研究を紹介する．

3.1 コードの完全性を検証

正しいコードのみを保持していなければ時間がかかる計算を行わせ，結果を得られるまでの時間を計測することで，不正コードを埋め込まれているかどうかを検証する手法がいくつか提案されている[9),11),12)]．時間ではなく，正しいコードのみを保持していなければ現実的に計算不可能な値を出力させる手法も存在する[18)]．これらの手法は通常，他の手法を用いて，「不正である可能性が高いノード」を検知し，そのノードに対して利用する手法である．なぜならば，この手法では多くのトラフィック量や計算コストが発生するからである．これらの手法の著者たちも，不正である可能性が高いノードを検知できる他の手法との併用を勧めている．

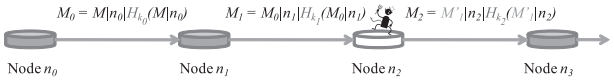


図 1　Naive algorithm
Fig. 1　Naive algorithm.

本論文で提案する手法は，不正である可能性が高いノードを検知する手法であり，これらの関連研究との併用が可能である．

3.2　ノードどうしによるモニタリング

隣接ノードの通信を傍受することによって不正ノードを検知する手法が提案されている．Watchdog[8] は，メッセージ転送の不正な行動の検知に重点を置いている．Watchdog では，メッセージ送信者が，メッセージ受信者が正しくメッセージを他のノードに通信するかどうかを監視する．もしメッセージ受信者がメッセージを破棄したり改ざんしたりすると，メッセージ送信者はネットワーク内の残りのノードに，そのメッセージ受信者が不正ノードであることを通知する．また，文献 14) や文献 16) では複数ノードが共同で監視することにより不正ノード検知の精度を向上させている．

これらの手法はネットワークに参加しているノードどうしがお互いを監視する手法である．監視役のノード自身もセキュリティ侵害を受けている可能性があるため，共謀攻撃に弱いという問題がある[24]．もしシンクを介することなく，ノードどうしでメッセージの送受信を行う環境を想定するのであれば，このような手法が必要となる．だが本論文においては，メッセージの宛先がシンクである環境を想定しているため，不正ノード検知のタスクをシンクに与えることができる．また，シンクはセキュリティ侵害を受けないという想定を行っているため，共謀攻撃にも頑健な，不正ノード検知の手法を提案することが可能となる．

3.3　シンクからのトレースバック

DoS 攻撃の対策手法等，インターネットの分野でトレースバック手法が数多く提案されている[1),13)]．これらの手法は，ルータが信頼でき，比較的高性能であるという想定を置いている．WSN においては，センサノードがルータの役目を果たしているため，このような想定を置くことはできない．そのため，インターネットの分野で用いられている手法を WSN に適用することはできない．

WSN におけるトレースバック手法として，**Naive approach** が文献 20) で紹介されている．基本的な動作を**図 1** に示す．各ノードがそれぞれ異なる ID n_i を持ち，また，シンクと秘密鍵 k_i を共有している．全ノードとシンクで共有されているセキュアハッシュ関数を H で表す．$H_{k_i}(m)$ は，ハッシュ関数 H と鍵 k_i を用いて計算された，メッセージ m の MAC を意味する．また，ストリームの結合を | で表す．図 1 において初期メッセージ M は，イベントタイプやイベントを検知したノードの ID，場所等を含んでいる．初期メッセージ M を作成した後，ノード n_0 は $M|n_0$ の MAC を k_0 を用いて計算し，メッセージ $M_0 = M|n_0|H_{k_0}(M|n_0)$ を作成し，隣接ノード n_1 に転送する．ノード n_1 は $M_0|n_1$ の MAC を鍵 k_1 を用いて計算し，メッセージ M_1 を作成する．

シンクが最終的にメッセージ $M_{n_r} = M_{n_{r-1}}|n_r|H_{k_r}(M_{n_{r-1}}|n_r)$ を受け取ったとき，シンクはメッセージの改ざんがなかったか検証プロセスを開始する．シンクは共有ハッシュ関数と全ノードの秘密鍵を保持している．まずシンクは $M_{n_{r-1}}|n_r$ の MAC を鍵 k_r を用いて計算する．そして，メッセージ M_{n_r} に含まれている値と比較し，同一であるかどうかの検証を行う．同一であった場合，1 つ前のホップ $r-1$ のノード ID をメッセージ M_{n_r} から取得し，再び MAC の同一性の検証を行う．このプロセスを，シンク側で計算する MAC と異なる MAC を発見するまで，または，すべての検証が完了するまで繰り返し行う．異なる MAC を発見した場合，中継ノードのいずれかのノードがメッセージを改ざんしたということである．最後に検証が成功したノードとその 1 ホップ隣接ノードのいずれかのノードが，メッセージを改ざんしたと結論づけることができる．またすべての MAC の検証が成功した場合，最後の検証を行ったノード ID とその 1 ホップ隣接ノードのいずれかのノードが，このメッセージ M_{n_r} の発生元ノードであると結論づけることができる．このメッセージが不正メッセージである場合（前述したように，不正メッセージ自身の検知は本論文の対象外である），メッセージ発生元ノードが不正ノードとなる．詳細な証明に関しては，文献 20) に記述されている．また，本論文においては，シンクの検証プロセスにおいて最後に検証が成功したノードを Last Verified Node（LVN）と定義する．

不正ノードは，不正メッセージに対して正しい MAC を付加するかどうかを選択することができる．例として図 1 を考える．不正ノード n_2 はメッセージ M_1 を M_1' に改ざんし，その後，$M_1'|n_2$ の MAC を計算している．この場合，LVN はノード n_2 である．一方，仮にノード n_2 がこのメッセージに任意の MAC を付加した場合，LVN はノード n_3 となる．本論文中では，簡単のため，不正ノードはつねに不正メッセージに対して正しい MAC を付加すると想定する．この想定が誤っていた場合，不正ノードの隣接ノードを不正ノードであると誤判断する可能性がある．この制限は，前述したようにトレースバック手法を用いる既存研究と同様のものである[20),23)]．

Naive approach は 1 つの不正メッセージから不正ノードを特定することができる．しか

し，すべてのメッセージ中継ノードがすべてのメッセージに対して MAC（たとえば 64 ビット）を付加するため，メッセージオーバヘッドが大きくなるという欠点がある．巨大な WSN ではこの手法をこのまま用いることは困難である[20]．

PNM[20] は，メッセージ中継ノードが確率的にメッセージにノード ID と MAC を付加することによりトラフィック量の発生を抑えている．十分な数の不正メッセージを受け取ることにより，シンクはメッセージが通ってきたノードのパスを特定することができる．シンクは，ルーティングパスを特定することにより，不正ノードを特定する．たとえば，あるメッセージに付加されたノード ID が，3，5，10 であったとする．この場合，このパス上にノード ID がそれぞれ 3，5，10 であるノードが，この順番で存在していることが判明する．ここで，仮にノード 5 とノード 10 の間に不正ノードが存在し，このメッセージを改ざんしたとする．この場合，シンクにおいて，ノード 5 とノード 8 の間に存在するいずれかのノードが，メッセージを改ざんしたということが判明する．このような不正メッセージを数多く受け取ることにより，シンクは完全なルーティングパスを特定し，どのノードがメッセージを改ざんしていたのかの判断を行う．だが，ルーティングパスに変更があると，ルーティングパスの特定を行うことができない．また，メッセージ発生元が複数あり，お互いのパスに共通に含まれるノードが存在するような場合においても，パスの特定を行うことができない．したがって，メッセージ発生元が 1 つのみであり，メッセージがつねに同じパスを通ってシンクまで届けられるような環境でしか利用することができない．また，不正ノードを検知するまでに大量の不正メッセージがシンクまで届けられる必要がある．

文献 23) では，PNM よりも少ない数の不正メッセージから，不正メッセージ発生元ノードを検知することができる．だが，メッセージの改ざんを検知することはできず，ルーティングパスが変化するような動的な環境では用いることができない．

4. 提案手法

提案手法ではメッセージの転送ノードが，ノード ID と k ビットの MAC を付加する．ここでトラフィック量削減のため，k の値は小さい値（たとえば 1 ビット）に設定する．PNM では，トラフィック量削減のために，各ノードが確率的にノード ID と MAC を付加する手法を用いていた．だが，不正ノードを特定するためには不正メッセージが通ったルーティングパスを特定する必要がある．ルーティングパスに変更があると，前述したようにルーティングパスの特定ができないため，ルーティングパスが変更する環境では用いることができなかった．本手法においては，すべてのノードがノード ID と MAC を付加する代わりに，トラフィック量を削減するために，MAC のビット長を k ビットに削減する．k の値が小さい場合は，不正ノードが高確率で正しい MAC をねつ造できてしまう．だが，すべての MAC を正しく作成することはできないため，シンクが複数の不正メッセージを受け取ると，統計学的手法を用いることにより，どのノードが不正を行ったノードである確率が高いかを判断することができる．また，ルーティングパスの変化に対応するため，**論理ノード**を導入する．

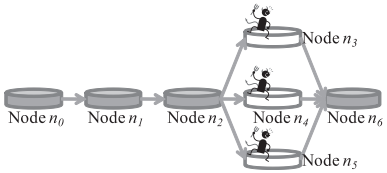


図 2　メッセージのルーティングパスの変更を利用した共謀攻撃
Fig. 2　Collusion attacks in the situation the routing path can change.

4.1 MAC のビット数の削減

Naive approach や PNM では MAC を利用している．Du らは MAC のビット長には言及していないが，WSN においては 64 ビットかそれ以上の長さを利用するのが普通である[2]．我々はこの MAC のビット長を k ビット，たとえば 1 ビットに削減する．もし k の値が小さい場合，攻撃者が高い確率で正しい MAC をねつ造することが可能となる．だが，いくつかの不正メッセージを受け取ることにより，統計学的手法を用いることでシンクは不正メッセージ作成元ノードを検知することが可能である．

たとえば図 1 の状況を考える．ノード n_2 がメッセージを改ざんしたとき，Naive approach においてはノード n_2 が LVN となる．我々の手法では，LVN となるノードの候補は，メッセージ発生元ノードから不正ノードまでのすべてのノードであり，この例の場合は，ノード n_0，n_1，n_2 のいずれかのノードである．このノードの中ではノード n_2 が最も高い確率で LVN となる．したがって，どのノードが LVN となったかの数を数えることで，不正ノードの確率が高いノードを求めることが可能である．

ルーティングパスの変化を考慮すると問題は難しくなる．たとえば，**図 2** の状況を考える．この例では，ノード n_2 の隣接ノード群であるノード n_3，n_4，n_5 が不正ノードである．この場合，LVN となる可能性が最も高いのは，正しいノードであるノード n_2 であるかもしれない．また，今後ノード n_2 の隣接ノードがこれらの不正ノード以外のノードに変わることも考えられる．この場合，誤った判断をすることなく不正ノードを検知することはさらに難しくなる．

また，計算量についても膨大になることが考えられる．シンクが複数の不正ノードから複数の不正メッセージを受信した場合，不正ノードの候補はセンサネットワーク内の多数のノードになるからである．

4.2　提案手法の詳細

メッセージを中継するノードの振舞いは Naive algorithm と同様である．各ノードにはそれぞれ異なる IDn_i を割り当て，シンクと秘密鍵 k_i を共有する．すべての中継ノードはメッセージに自身の ID を付加し，自身の秘密鍵を用いて MAC を計算する．その MAC をメッセージに付加する．

不正ノードの検知方法は以下のようになる．まずノード $\hat{n}$ に注目して考える．ノード $\hat{n}$ が LVN となった回数を数える．また，ノード $\hat{n}$ の周辺のノードが LVN となった回数をそれぞれのノードに対して数える．この状況下のときの，ノード $\hat{n}$ が不正ノードである条件付き確率（不正メッセージを作成した条件付き確率）を計算する．この条件付き確率がある閾値（たとえば 0.999）を超えたとき，ノード $\hat{n}$ が不正ノードであると結論づける．

シンクは，不正メッセージを受け取った場合，メッセージに付加されているすべての ID と LVN の ID を記録する．受け取った不正メッセージが通ったパスを $p_i = \langle n_a, n_b, \ldots \rangle$ とする（$n_a, n_b \ldots$ はノード ID を表す）．メッセージ発生元ノードからシンクまでのホップ数を $|p_i|$ で表す．不正メッセージのルーティングパスの集合を $P = \{p_1, \ldots, p_d\}$ とする．d はシンクが受け取った不正メッセージ数である．

ルーティングパス p_i における LVN のノード ID を $L[p_i]$ とする．また，パス p_i において，ノード n を通った順番を $M_n[p_i]$ とする．パス p_i において LVN を通った順番を特に，$M_L[p_i] = M_{L[p_i]}[p_i]$ と記述する．

シンクが受け取った最新の不正メッセージについて，LVN をノード $\hat{n}$ とする．シンクはノード $\hat{n}$ を含み，$M_L[p] > M_{\hat{n}}[p]$ を満たすすべてのパスを P から抽出し，抽出したパスの集合を $P_{\hat{n}}$ とする．つまり，

$$P_{\hat{n}} = \{p_i \,|\, p_i \in P \,\&\, \hat{n} \in p_i \,\&\, M_L[p_i] > M_{\hat{n}}[p_i]\} \tag{1}$$

である．パス集合 $P_{\hat{n}}$ の i 番目のパスを $P_{\hat{n},i}$ とする．カウンタ $C(\hat{n}) = \langle c_1, \ldots, c_{N^{\max}_{\hat{n}}} \rangle$ と $C'(\hat{n}) = \langle c'_1, \ldots, c'_{N^{\max}_{\hat{n}}} \rangle$ を用意する．ここで，$N^{\max}_{\hat{n}}$ はパス集合 $P_{\hat{n}}$ において，ノード $\hat{n}$ からシンクまでの最大ホップ数 -1 を表しており，

$$N^{\max}_{\hat{n}} = \max_i(|P_{\hat{n},i}| - M_{\hat{n}}[P_{\hat{n},i}]) \tag{2}$$

である．

次に，c_i と c'_i を次のように計算する．c_i とは，パス集合 $P_{\hat{n}}$ において，ノード $\hat{n}$ よりも i ホップだけシンク側にあるノードが LVN となった回数を表している．ここで，c_i に対応するノードは変わる可能性がある（**図 3**）．c_i に対応するノードを論理ノード $\tilde{n}_i$ と呼ぶことにする．つまり，論理ノード $\tilde{n}_i$ は，パス集合 $P_{\hat{n}}$ の各パスにおいて，ノード $\hat{n}$ より i ホップだけシンク側に位置しているノードを意味する．論理ノードを導入することにより，シンクにおける計算量を削減することと，ルーティングパスの変更に対応することができる．各 c_i は次のように計算される．

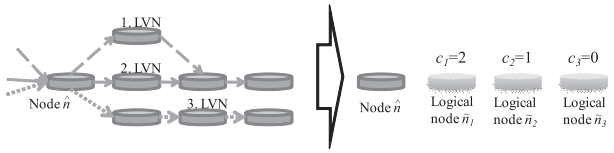


図 3　ノード $\hat{n}$ の論理ノードと c_i．矢印は不正メッセージのルーティングパスを表し，LVN の文字は，そのノードが LVN になったことを表す

Fig. 3　Logical nodes and c_i for node $\hat{n}$. Arrows represent routing paths of three false messages and the text *LVN* represents that the node became a LVN.

$$c_i = \sum_{j=1}^{|P_{\hat{n}}|} \delta_{i, M_L[P_{\hat{n},j}] - M_{\hat{n}}[P_{\hat{n},j}]} \tag{3}$$

ここで，$\delta_{i,j}$ はクロネッカのデルタを意味する．

c'_i は，論理ノード $\tilde{n}_i$ 自身が不正を行った影響により論理ノード $\tilde{n}_i$ が LVN となった期待値を表している．つまり，c'_i は，c_i から，他ノードが不正を行った影響により論理ノード $\tilde{n}_i$ が不正ノードとなった回数を差し引いたものになる．例として図 3 を用いて説明する．ノード $\hat{n}$ にとって c_1 は 2 である．このとき，この論理ノード $\tilde{n}_1$ が 2 回 LVN となった原因として，論理ノード $\tilde{n}_1$，$\tilde{n}_2$，$\tilde{n}_3$ のいずれか，または複数ノードが不正を行ったことになる．論理ノード $\tilde{n}_1$ 自身のみが 2 回以上不正を行い，これが原因で論理ノード $\tilde{n}_1$ が 2 度 LVN となった可能性がある．また，論理ノード $\tilde{n}_1$ 以外の論理ノードだけが不正を行い，これが原因で論理ノード $\tilde{n}_1$ が 2 度 LVN となった可能性もある．後者の場合，論理ノード $\tilde{n}_1$ 自身が不正を行ったために論理ノード $\tilde{n}_1$ が LVN となった回数は 0 である．以下のように計算を行うことで，論理ノード $\tilde{n}_i$ 自身が不正を行ったために論理ノード $\tilde{n}_i$ が LVN となった回数の期待値を求めることができる．

まずすべての c'_i を c_i に初期化する．整数 j を用意し，初期値を $N^{max}_{\hat{n}}$ とする．各 $c'_i (i = 1, \ldots, j-1)$ について，

$$c'_i \rightarrow \max(0, c'_i - c'_j \cdot 2^{-k\cdot(j-i)}) \tag{4}$$

のように更新する．ここで，max 関数は 2 つの引数を受け，大きいほうの値を返す関数である．この更新を $j = N_{\hat{n}}^{max}$ から $j = 1$ まで繰り返す．

パス集合 $P_{\hat{n}}$ において，ノード n_i が LVN となった回数を L_{n_i} とする．ノード $\hat{n}$ が不正ノードである確率を $Q(\hat{n})$ とすると，ベイズの定理より，

$$Q(\hat{n}) = \sum_{z=0}^{L_{\hat{n}}-1} I(z) / \sum_{z=0}^{L_{\hat{n}}} I(z) \tag{5}$$

となる．ここで，$I(z)$ は，パス集合 $P_{\hat{n}}$ において，ノード $\hat{n}$ よりシンク側に位置するノードが不正を行った影響で，ノード $\hat{n}$ が z 回だけ LVN となる確率を表す．たとえば，$I(0)$ はノード $\hat{n}$ 以外のノードが不正を行ったのが原因でノード $\hat{n}$ が LVN となったことがないことを意味する．したがって，ノード $\hat{n}$ 自身のみが L_{n_i} 回以上不正を行ったのが原因でノード $\hat{n}$ が L_{n_i} 回 LVN になったということである．一方，$I(L_{\hat{n}})$ は，ノード $\hat{n}$ 以外のノードのみが不正を行った影響により，ノード $\hat{n}$ が $I(L_{\hat{n}})$ 回 LVN となる確率を表す．したがって，ノード $\hat{n}$ は不正を行っていないということであり，ノード $\hat{n}$ は不正ノードではないことになる．

式 (5) の分子は，ノード $\hat{n}$ 以外のノードのみが不正を行った影響により，ノード $\hat{n}$ が 0 回から $I(L_{\hat{n}}) - 1$ 回だけ LVN になる確率の合計を表す．つまり，少なくとも 1 回は，ノード $\hat{n}$ 自身の影響により LVN になるということである．したがって，ノード $\hat{n}$ が不正ノードである確率（1 回以上不正メッセージを作成した確率）と等しい．

シンクは $I(z)$ を下記のように計算を行う．ノード n に注目し，L'_n（ノード n が自らの影響で LVN となる回数）を c' とし，W_n（ノード n が不正を行った回数）を r とする．ベイズの定理より，ノード n が自身の影響により c' 回 LVN となったとき，ノード n が r 回不正を行った条件付き確率 $P_n(W_n = r|L'_n = c')$ は，

$$P_n(W_n = r|L'_n = c') = \frac{P(W_n = r) \cdot P(L'_n = c'|W_n = r)}{\sum_{i=0}^{\infty} P(W_n = i) \cdot P(L'_n = c'|W_n = i)} \tag{6}$$

である．ここで，$P_n(L'_n = c'|W_n = r)$ は，ノード n が r 回不正を行ったとき，それが原因で c' 回 LVN となる確率を表す．

ノード n が不正メッセージを作成し，シンクがそのメッセージを不正メッセージであると検知したとする．シンクにおける LVN の検証プロセスにおいて，ノード n の次に検証を行うノードの検証が失敗した場合，ノード n が LVN となる．この確率は，$1 - 2^{-k}$ である．ノード n の次に検証を行うノードの検証が成功した場合，ノード n は LVN とはならない．この確率は 2^{-k} である．したがって，

$$P_n(L'_n = c'|W_n = r) = {}_rC_{c'}(1 - 2^{-k})^{c'}(2^{-k})^{r-c'} \tag{7}$$

である．

$P(W_n = r)$ はノード n が r 回不正メッセージを作成する確率である．ノード n 自身の影響で LVN となった回数が c' であるから，ノード n は c' 回以上不正メッセージを作成しているはずである．したがって，$r < c'$ のとき $P(W_n = r) = 0$ である．また，$r \geq c'$ のときは，すべての r について $P(W_n = r)$ は等しい値を持つと考えられる．なぜなら，不正ノードが何度不正メッセージを作成するかは任意であるからである．したがって，

$$\begin{aligned} P_n(W_n = r|L'_n = c') &= \frac{{}_rC_{c'}(1 - 2^{-k})^{c'}(2^{-k})^{r-c'}}{\sum_{i=c'}^{\infty}\left[{}_iC_{c'}(1 - 2^{-k})^{c'}(2^{-k})^{i-c'}\right]} \\ &= (2^{-k})^{1-c'-r}(1 - 2^{-k})^{c'}(-1 + 2^k){}_rC_{c'} \end{aligned} \tag{8}$$

ノード n について，L'_n（ノード n が自ら不正メッセージを作成した影響により，ノード n が LVN となった回数）が c' であり，W_n（ノード n が不正メッセージを作成した回数）が r であるとき，この影響により，パス集合 $P_{\hat{n}}$ において，ノード n から h ホップ離れたところにあるノード $\hat{n}$ が，$D_n = q$ 回だけ LVN となる条件付き確率は，

$$\begin{aligned} P_n(D_n = q|W_n = r\&L'_n = c') &= \frac{P(W_n = r) \cdot P(D_n = q\&L'_n = c'|W_n = r)}{P(W_n = r) \cdot P(L'_n = c'|W_n = r)} \\ &= 2^{-k(c'+q+hq-r)}(2^k - 1)^q \times \{2^{-(1+h)k}(1 - 2^k + 2^{hk})\}^{r-c'-q}{}_{r-c'}C_q \end{aligned} \tag{9}$$

である．したがって，L'_n が c' であるとき，この影響により，パス集合 $P_{\hat{n}}$ において，ノード n から h ホップ離れたところにあるノード $\hat{n}$ が，$D_n = q$ 回だけ LVN となる条件付き確率は，

$$\begin{aligned} P_n(D_n = q|L'_n = c') &= \sum_{r=c'+q}^{\infty} P(W_n = r|L'_n = c') \cdot P(D_n = q|W_n = r\&L'_n = c') \\ &= \frac{2^{-hkq}(1 + 2^{-hk})^{-1-c'-q}(c' + q)!}{c'!\, q!} \end{aligned} \tag{10}$$

である．これらから，ノード $\hat{n}$ の論理ノード $\tilde{n}_1, \ldots, \tilde{n}_{N_{\hat{n}}^{max}}$ が不正メッセージを作成した影響により，ノード $\hat{n}$ が z 回だけ LVN となる条件付き確率は，

$$I(z) = \sum_{q_1=0}^{z} \sum_{q_2=0}^{z-q_1} \cdots \sum_{q_{N-1}=0}^{z-\sum_{i=1}^{N-2} q_i} \left[P_{n_N}\left(D_N = z - \sum_{i=1}^{N-1} q_i | L'_{n_N} = c'_N \right) \prod_{i=1}^{N-1} P_{n_i}\left(D_{n_i} = q_i | L'_{n_i} = c'_i \right) \right] \quad (11)$$

である．ここで，$N = N_{\hat{n}}^{max}$ である．

不正ノードである確率が th を超えるとき，そのノードを不正ノードであると判断する．th はシステム管理者が決定する値である．式 (5)，(11) より

$$Q(\hat{n}) > th \quad (12)$$

を満たすとき，ノード $\hat{n}$ が不正ノードである（不正メッセージを 1 回以上作成した）確率が th を超える．

4.3　不正ノードと判断した後の処理

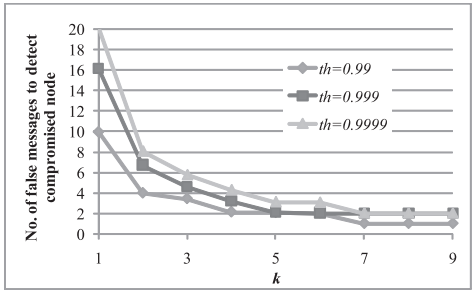
本手法により不正ノードであると判断されたノードが，本当に不正ノードであるかどうかの判定をするためには，3 章で紹介した，コードの完全性検証の手法等を利用することができる．本当に不正ノードであった場合は，物理的に該当ノードを取り除く手法や，ネットワーク全体に，該当ノードとのコミュニケーションを禁止するメッセージを通達する等の手法をとることができる．本論文においては，どの手法を用いるべきかについては言及しない．

一方，本手法により不正ノードであると判断されたノード $\hat{n}$ が，実際には不正ノードではなかった場合，ノード $\hat{n}$ が LVN となった回数を 0 にリセットする．

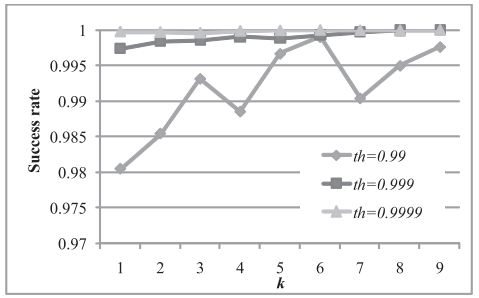
5. 評　　価

提案手法の有効性をシミュレーションによって評価する．ノード ID を 10 ビット，既存研究において用いられる MAC の長さを 64 ビットに設定した．

最初のシミュレーションでは，ノードを直線上に 30 体用意し，1 番目のノードをメッセージ発生ノードとした．15 番目のノードを不正ノードとし，30 番目のノードをシンクにメッセージを転送する役のノードとした．15 番目のノードは，受け取ったメッセージをつねに改ざんするように設定した．メッセージ発生ノードはシンクが不正ノードを検知するまで繰り返しメッセージを発生させた．この試行を 10,000 回繰り返した．



(a) No. of false messages sink received until it detected compromised node



(b) Success rate

図 4　不正ノードを検知するまでに必要な不正メッセージ数と，検知の成功率
Fig. 4　No. of false messages and success rate for detecting a compromised node.

図 4 は，このシミュレーションの結果を表している．利用する MAC のビット数 k を 1 から 9 まで変化させ，閾値 th を 0.99 から 0.9999 まで変化させた．シンクが不正ノードを検知するまでに必要としたメッセージ数を図 4 (a) に示す．k の値を大きく，th の値を小さくしたほうが，不正ノードをより早く検知できることが分かる．

シンクが不正ノードを正しく検知する成功率を図 4 (b) に示す．ここで成功率を，不正ノードの数（ここでは 1）を，シンクがあるノードを不正ノードであると断定した回数で割ったものであると定義している．図 4 (b) はこの平均値を表している．ルーティングパスが不変である単純なシミュレーションにおける成功率は，おおよそ閾値 th と一致していることが分かる．また，シンクが正しいノードを不正ノードであると判断してしまった場合，4.3 節の手順に従うことで，最終的に不正ノードを検知することができた．

次に，既存研究である PNM [20] との比較を行った．結果を**図 5** に示す．X 軸は 1 つの

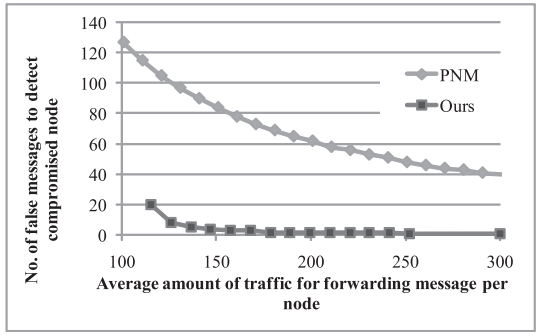


図 5　シンクが不正ノードを検知するまでに必要とした不正メッセージ数
Fig. 5　No. of false messages sink received until it detected compromised node.

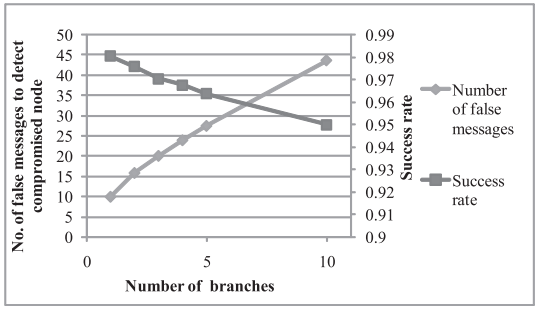


図 6　共謀攻撃への耐性
Fig. 6　Resilience to collusion attacks.

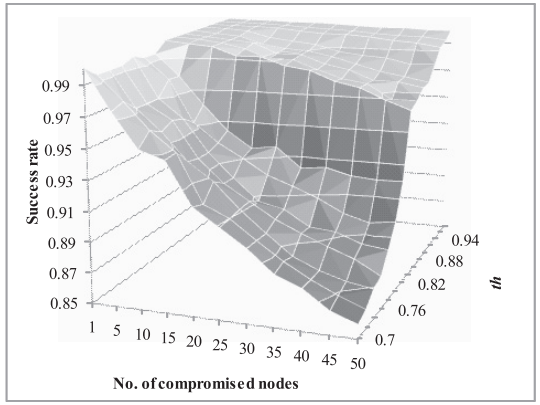
ノードがメッセージを転送するのに必要なトラフィック量の平均値を表している．この値は，メッセージ発生元ノードからシンクまでメッセージを転送するのに必要なトラフィック量を，ノード数で割ったものである．提案手法においては，MAC のビット数である k を調節することでこのトラフィック量を調節した．提案手法においては，すべてのメッセージ中継ノードがノード ID と k ビットの MAC を付加するため，トラフィック量をある一定量より小さく設定することはできない．PNM においては，PNM で利用するパラメータを調節することで，メッセージ転送に必要なトラフィック量を調節することができる．また，PNM では，成功率を 0.99 に設定したが，提案手法においては，th を 0.999 に設定した（図 4 (b) より，th を 0.999 に設定したときの成功率の最小値が 0.99 を超えている）．結果を図 5 に示す．提案手法が PNM よりも少ない不正メッセージ数から，早く不正ノードを検知できたことが分かる．たとえば，メッセージ通信に必要な 1 ノードあたりの平均トラフィック量を 125 ビットに設定した場合，提案手法は PNM の 8%の数の不正メッセージから，不正ノードを検知できる．

次に，提案手法が共謀攻撃に対して耐性を持っているかどうかを調べるシミュレーションを行った．このシミュレーションでは，図 2 のようにノードを配置した．ノード n_2 は隣接ノードの 1 つをランダムに選択しメッセージを転送する．ノード n_2 からメッセージを受け取ったノードは，つねにメッセージを改ざんするように設定した．結果を**図 6** に示す．X 軸は，分岐の数，つまり，ノード n_2 の隣接ノードである不正ノードの数を表している．たとえば，図 2 においては，分岐の数は 3 である．シミュレーションでは，th を 0.99 に，k を 1 に設定した．分岐の数が多い場合，不正ノード検知の成功率は減少する．だが，減少の程度は 0.98 から 0.95 というように緩やかであると考えられる．

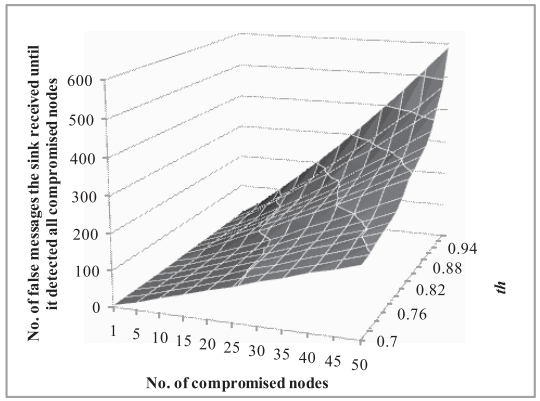
最後に，ルーティングパスの変化に提案手法が対応できるかどうかを調べるシミュレーションを行った．センサノードの数を 1,000 に設定し，その中の 1 つのノードが繰り返しメッセージを発生させるように設定した．また，不正ノード（メッセージを改ざんするノード）数を 1 から 50 に変化させながら実験を行った．また，シンクが不正メッセージを受け取るたびに，すべてのノードの場所を変更させた．したがって，毎回，ルーティングパスが大規模に変化することになる．**図 7** が結果を表している．th を 0.7 から 0.98 に変化させながら評価した．

図 7 (a) から，各成功率は th よりも高いことが分かる．ルーティングパスを固定した実験（図 4 (b)）と比較すると，ルーティングパスが変化するほうが，高い成功率で不正ノードを検知できるといえる．この理由は以下のように考えられる．たとえば，図 1 を用いて例を述べる．ノード n_1 とノード n_2 があるメッセージの中継ノードであり，ノード n_2 がメッセージを改ざんした場合，ノード n_2 の隣接ノードであるノード n_1 がある一定の確率で LVN となる．だが，ルーティングパスが変化する場合，ノード n_2 の隣接ノードも変化するため，LVN となるノードが変化する．したがって，不正ノード検知を失敗する（不正ノードでないのに不正ノードであると判断してしまう）確率が減少する．

また，図 7 (b) から，シンクがすべての不正ノードを検知するために必要な不正メッセージ数が増加していることが分かる．だが，我々はこの増加の割合はスケーラブルであると考える．たとえば，th を 0.98 に（このときの検知成功率は 0.999 以上であることが図 7 (a)



(a) Success rate for detecting compromised nodes



(b) No. of false messages sink received until it detected compromised node

図 7　不正ノード検知までに必要な不正メッセージ数と成功率
Fig. 7　The number of false messages and success rate.

よりいえる），不正ノードの数を 50 に設定した場合，提案手法は 1 つの不正ノードを検知するために平均 12 の不正メッセージを必要としている．これは，PNM がただ 1 つだけ存在する不正ノードを検知するのに必要な不正メッセージ数よりも少ない値である．

6. 考　　察

本章では提案手法についての考察を行う．

k の値．提案手法では，MAC のビット長 k を自由に設定することができる．もし k の値が小さければ，1 つのメッセージをシンクまで転送するのに必要なトラフィック量は減少する．だが，シンクが不正ノードを検知するまでにより多くの不正メッセージが必要となる．結果として不正メッセージを転送するために必要なトラフィック量が増大する．最適な k を求めるためには，不正メッセージと正しいメッセージの発生割合を考慮する必要がある．仮に不正メッセージの発生率が高い場合，できるだけ早く不正ノードを検知する必要があるため，k の値を大きく設定しておくべきだと考えられる．一方，正しいメッセージが多い場合は，k の値は小さく設定すべきである．なぜなら，k ビットの MAC は正しいメッセージと不正メッセージの区別なく付加されるからである．正しいメッセージに付加される MAC は，シンクにおいて，そのメッセージが改ざんされていないことを確認できる指標になるが，不正ノードの検知という観点からは無駄なものである．

異なる ID を付加する攻撃．不正ノードは，メッセージに付加する自分のノード ID を偽ることができる．だが，もしノードが移動できないのであれば，不正ノードの 1 ホップ隣接ノードがつねに LVN となるため，不正ノードの 1 ホップ隣接ノードまではシンクからトレースバックが可能である（文献 20) において議論がされている）．また，文献 4) 等によって提案されているような，隣接ノードの認証手法を用いることにより，ノード ID を偽ることができないようにすることで，提案手法を改良することができると考えられる．

ノードの移動への対応．本論文においては，ノードは移動できないことを想定している．だが，ノードが移動する場合においても，ノード ID を偽ることができない想定の下であれば，提案手法はある程度有効であると考えている（前章における最後のシミュレーションでは，すべてのノードが移動している）．だが，本論文では，不正ノードはつねに不正メッセージに対して正しい MAC を付加すると想定している．ノードが移動しないのであれば，この想定が誤っている場合でも，不正ノードの 1 ホップ隣接ノードを検知することができる．もしノードが移動する場合，隣接ノード以外のノードを不正ノードであると判断してしまうことが考えられる．このことにどう対応するかは将来の課題である．

7. おわりに

不正ノードの問題は，無線センサネットワークにおける大きな課題となっている．我々は，

シンクに通知される不正メッセージから，不正ノードを検知する手法を提案した．既存研究は共謀攻撃に対して脆弱性を持っていたり，メッセージ発生元ノードからシンクまでのルーティングパスが不変であったりするような環境にしか利用できないといった課題があった．本論文では，メッセージに付加する MAC のビット数を削減する手法を提案し，また，ルーティングパスの変更に対応するために論理ノードの導入を行った．シミュレーション結果より，既存研究よりも少ない不正メッセージ数で不正ノードを検知することが可能であることが分かった．また，ルーティングパスの変更に対応できるだけでなく，その変更の度合いが高いほど高い性能が得られることが分かった．将来研究として，本提案手法を実機のセンサに実装し，そのパフォーマンスを評価することを考えている．また，6 章において考察したことに取り組む必要がある．

参 考 文 献

1) Dong, Q., Banerjee, S., Adler, M. and Hirata, K.: Efficient probabilistic packet marking, *IEEE ICNP*, pp.368–377 (2005).
2) Du, W., Deng, J., Han, Y.S. and Varshney, P.K.: A pairwise key pre-distribution scheme for wireless sensor networks, *ACM CCS*, pp.42–51 (2003).
3) Ganesan, P., Venugopalan, R., Peddabachagari, P., Dean, A., Mueller, F. and Sichitiu, M.: Analyzing and modeling encryption overhead for sensor network nodes, *ACM WSNA*, pp.151–159 (2003).
4) Gu, W., Bai, X., Chellappan, S., Xuan, D. and Jia, W.: Network decoupling: A methodology for secure communications in wireless sensor networks, *IEEE Trans. Parallel Distrib. Syst.*, Vol.18, No.12, pp.1784–1796 (2007).
5) Krauß, C., Schneider, M. and Eckert, C.: Defending against false-endorsement-based dos attacks in wireless sensor networks, *ACM WiSec*, pp.13–23 (2008).
6) Krawczyk, H., Bellare, M. and Canetti, R.: HMAC: Keyed-hashing for message authentication, IETF – Network Working Group, RFC2104 (Feb. 1997).
7) Li, F. and Wu, J.: A probabilistic voting-based filtering scheme in wireless sensor networks, *ACM MOBICOM*, pp.27–32 (2006).
8) Marti, S., Giuli, T.J., Lai, K. and Baker, M.: Mitigating routing misbehavior in mobile ad hoc networks, *ACM MOBICOM*, pp.255–265 (2000).
9) Park, T. and Shin, K.G.: Soft tamper-proofing via program integrity verification in wireless sensor networks, *IEEE Trans. Mobile Computing*, Vol.4, No.3, pp.297–309 (2005).
10) Sei, Y. and Honiden, S.: Resilient security for false event detection without loss of legitimate events in wireless sensor networks, *IEEE DOA*, pp.454–470 (2007).
11) Seshadri, A., Perrig, A., van Doorn, L. and Khosla, P.: SWATT: Software-based attestation for embedded devices, *IEEE S&P*, pp.272–282 (2004).
12) Shaneck, M., Mahadevan, K., Kher, V. and Kim, Y.: Remote software-based attestation for wireless sensors, *ESAS*, pp.27–41 (2005).
13) Song, D.X. and Perrig, A.: Advanced and authenticated marking schemes for IP traceback, *IEEE INFOCOM* (2001).
14) Wang, G., Zhang, W., Cao, G. and Porta, T.: On supporting distributed collaboration in sensor networks, *IEEE MILCOM* (2003).
15) Wood, A.D. and Stankovic, J.A.: Denial of service in sensor networks, *IEEE Computer*, Vol.35, No.10, pp.54–62 (2002).
16) Yang, H., Shu, J., Meng, X. and Lu, S.: SCAN: Self-organized network layer security in mobile ad hoc networks, *IEEE Journal on Selected Areas in Communications*, Vol.24, No.2, pp.261–273 (2006).
17) Yang, H., Ye, F., Yuan, Y., Lu, S. and Arbaugh, W.: Toward resilient security in wireless sensor networks, *ACM MOBIHOC*, pp.34–45 (2005).
18) Yang, Y., Wang, X., Zhu, S. and Cao, G.: Distributed software-based attestation for node compromise detection in sensor networks, *IEEE SRDS*, pp.219–230 (2007).
19) Ye, F., Luo, H., Lu, S. and Zhang, L.: Statistical en-route filtering of injected false data in sensor networks, *IEEE Journal on Selected Areas in Communications*, Vol.23, No.4, pp.839–850 (2005).
20) Ye, F., Yang, H. and Liu, Z.: Catching “moles” in sensor networks, *IEEE ICDCS* (2007).
21) Ye, F., Zhong, G., Lu, S. and Zhang, L.: Gradient broadcast: A robust data delivery protocol for large scale sensor networks, *Wirel. Netw.*, Vol.11, No.3, pp.285–298 (2005).
22) Yu, Z. and Guan, Y.: A dynamic en-route scheme for filtering false data injection in wireless sensor networks, *IEEE INFOCOM*, pp.1–12 (2006).
23) Zhang, Q., Zhou, X., Yang, F. and Li, X.: Contact-based traceback in wireless sensor networks, *IEEE WiCom*, pp.2487–2490 (2007).
24) Zhang, Y., Yang, J., Jin, L. and Li, W.: Locating compromised sensor nodes through incremental hashing authentication, *DCOSS*, pp.321–337 (2006).
25) Zhu, S., Setia, S., Jajodia, S. and Ning, P.: An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks, *IEEE S&P*, pp.259–271 (2004).

(平成 20 年 5 月 13 日受付)
(平成 20 年 9 月 10 日採録)

清　　雄一（学生会員）

2004 年東京大学工学部システム創成学科卒業．2006 年東京大学大学院情報理工学研究科コンピュータ科学専攻修士課程修了，同年同博士課程進学，文部科学省国立情報学研究所アーキテクチャ研究系リサーチアシスタント．エージェント技術，センサネットワークの研究に従事．現在に至る．

本位田真一（正会員）

1978 年早稲田大学大学院理工学研究科修士課程修了．（株）東芝を経て 2000 年より国立情報学研究所教授，2004 年より同研究所アーキテクチャ科学研究系研究主幹を併任，現在に至る．2001 年より東京大学大学院情報理工学系研究科教授を兼任，現在に至る．2002 年 5 月〜2003 年 1 月英国 UCL ならびに Imperial College 客員研究員．2005 年度パリ第 6 大学招聘教授．早稲田大学客員教授．工学博士（早稲田大学）．1986 年度情報処理学会論文賞受賞．ソフトウェア工学，エージェント技術，ユビキタスコンピューティングの研究に従事．IEEE，ACM 等各会員，日本ソフトウェア科学会理事，情報処理学会理事を歴任．日本学術会議連携会員．