# 無線センサネットワークにおけるFalse Eventの検知

清 雄一[†1] 本位田 真一[†1,†2]

大規模なセンサネットワークでは，個々のセンサがセキュリティ侵害を受けやすい．セキュリティ侵害を受けたセンサは，嘘のイベント（False Event）を発生させるのに利用される．この攻撃は，イベントの受け取り手を混乱させるだけではなく，個々のセンサの有限のエネルギーを消費させる．False Event 攻撃に対する既存のセキュリティデデザインの共通目標は，小さな閾値を設定し，閾値未満の数のセンサがセキュリティ侵害を受けた場合の False Event を，In-network で検知することである．だが既存研究は，実際に発生したイベントが失われたり，独自のルーティングアルゴリズムを利用する必要があったりするという課題や，閾値以上のセンサがセキュリティ侵害を受けた場合にネットワーク全体のセキュリティが崩壊してしまうという課題を持つ．本論文では，ルーティングアルゴリズムの指定を行わず，閾値以上のセンサがセキュリティ侵害を受けた場合でも，ネットワーク全体のセキュリティをある程度保つことができる新しい機構を提案する．ルーティングアルゴリズムの指定を行わない既存研究と比較し，少ないホップ数で False Event の検知ができることを，数学的な解析やシミュレーションによって示す．

# False Event Detection in Wireless Sensor Networks

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

In a large scale sensor network, sensor nodes have a high risk of being captured and compromised. A compromised node can be used to generate false events. Such false events can deceive the user into wrong decisions. They can also waste a significant amount of network resources. Related works have problems; some works lose correct events stochastically and/or need to use their original routing protocols only for their methods. We propose a new method for detecting false events, which does not lose any correct events and does not specify any routing algorithms. Moreover, many security designs can address only a small threshold number of compromised nodes; the security protection completely breaks down when the threshold is exceeded. Our proposed method can achieve resiliency against an increasing number of compromised nodes. When we set the probability of losing correct events to 1%, our proposed method can detect more false events than related works do. We show this by mathematical analysis and simulations.

## 1. はじめに

無線センサネットワークは，軍事目的の偵察・森林火災の監視等のアプリケーションにおいて，環境をモニタリングすることができる．このようなネットワークでは，多数のセンサが広範囲に設置される．センサに検知されたイベントは，センサ群を管理するベースステーションまでマルチホップ無線ネットワークを通じて伝達される．

しかし，センサはセキュリティ侵害を受ける高いリスクを負っている．このようなアプリケーションでは，センサは物理的に入手可能な場所に配置されるため，攻撃者がセンサを取得，鍵を抽出し，その鍵を用いて嘘のイベント（False Event）を発生させることができる．

False Event の検知を目的とするいくつかの既存研究[2),11)]は，外部からの攻撃には対応できるが，センサの鍵を奪われることによる内部攻撃には対応していない．

内部攻撃に対応した既存研究では，ランダム鍵配備方式[17)]，地理制約鍵配備方式[8),15),21)]に大きく分類することができる．

これら既存研究に共通する目標は，False Event をできるだけ In-network で検知することである．False Event をセンサからベースステーションまで複数ホップを通じて転送することは，センサの有限の資源を無

†1 東京大学
The University of Tokyo
†2 独立行政法人国立情報学研究所
National Institute of Informatics

駄に消費することにつながるため，できるだけ早い段階で検知することが必要である．だが，ランダム鍵配備方式や地理制約鍵配備方式には以下のような問題がある．

ランダム鍵配備方式を用いた文献 17) では，各センサノードに重複を許して 1 個ずつ鍵を与えておく．手法の詳細は 3.2 節において詳述するが，イベントが発生した際，与えられている鍵を用いて，異なる $T$ 個の MAC（Message Authentication Code）を生成する．この手法により，$T$ 個以下のノードが物理的に取得され，鍵を抽出された場合でも，不正に生成されたイベントは In-network において False Event であると検知することができる．

だが本当にイベントが起こった場合でも，異なる $T$ 個の MAC を生成できない場合，ベースステーションまでイベントを通知する際に，中継ノードにおいて False Event として処理されイベントが消失してしまう．ランダム鍵配備方式では，$T$ 個以上のノードがイベントを検知しても，それぞれのノードが異なる鍵を保持しているとは限らないため，異なる $T$ 個の MAC を生成できる保証はない．高確率で異なる MAC を生成するためには，ノードに配備する鍵の重複度を下げる必要がある．だが鍵の重複度を下げると，In-network において False Event を検知できる確率が減少してしまう．

ランダム鍵配備方式では，実際に発生したイベントがつねに一定の確率で消失するという前提で評価が行われている．仮に実際に発生したイベントの消失率を 1%に抑えると，False Event を検知できるまでのホップ数は急激に増加する．また，$T$ 個以上の鍵が流出するとセキュリティが崩壊し，ベースステーションにおいても False Event を検知することはできなくなる．

一方，地理制約鍵配備方式を用いた文献 8)，15)，21) では，イベントのルーティング方法に大きな制約を課すことで，False Event を検知するまでのホップ数を抑えている．だが，センサネットワークにおけるルーティング方法は数多く提案されており[6),19)]，False Event 検知のためだけに大きな制約を課すことは問題である．文献 8)，21) では，イベント発生地点からベースステーションまで単一の固定された経路を用いるのみで，中継ノードの故障やバッテリ切れ等による経路変更に対応できない．ノードの故障やバッテリ残量を考慮しないこのルーティング方法は，センサネットワークで利用するのは難しいと考えられる．文献 15) では，イベント発生地点からベースステーションまでほぼ最短経路でイベントを転送するという制約を設けることにより，False Event を In-network で検知する確率を高めている．センサノードは，経路を外れたイベントメッセージを無条件に False Event と見なして破棄する．Geographic ルーティング[5),7)]を用いると，最短に近い経路でメッセージをルーティングすることができるが完全ではないため，10–20%の確率で実際に発生したイベントが False Event であると見なされて破棄されると論文中に表記されている．

そこで本論文では，実際に発生したイベントの消失を起こさず，ルーティング方法も指定しない，False Event 検知の新しい手法を提案する．そのために，イベント検知ノード集合という新しい概念を導入し，イベント検知ノード集合のセンサにそれぞれ別の鍵を割り当てる．このときの鍵の配備方法として，すべてのイベント検知円内の $T$ 個の鍵の組合せに重複が起きないようにする．また，$T$ 個以上のセンサの鍵が流出してもある程度セキュリティを保つことができる．実際に発生したイベントの消失率を 1%に抑えたとき，本論文の提案手法が既存手法よりも大幅に少ないホップ数で False Event の検知ができることを，数学的な解析やシミュレーションによって示す．

## 2. 背　　景

本章では，本研究において想定するセンサネットワークのモデルや，False Event 攻撃のモデルを定義する．

### 2.1 センサネットワークのモデル

多くの小さなセンサノードから成り立つセンサネットワークを考える．また，センサは高密度に配置されており，1 つのイベントを複数のセンサで検知できるものとする．これは，複数センサが協調することで高精度な検知を可能とするためや，ノードの故障に対応するためには必要である．検知したノードの中から 1 つのノードがイベントリーダノード（EN）に選択される．EN が周りのノードの検知結果を収集し，要約してベースステーションへ通知する．これを**イベントメッセージ**と呼ぶことにする．このとき多くの中継ノードを媒介し，マルチホップで通知される．イベント情報を収集するベースステーションは，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．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．また，センサノードは 1 度配置されたら移動しないものとする．

### 2.2 False Event 攻撃

攻撃者は 1 つまたは複数のノードのセキュリティを侵害することができる．ノードがセキュリティ侵害を

受けると，ノードに含まれている秘密鍵，データ，コード等すべての情報が漏洩するものとする．攻撃者はそれらを用いて本来は発生していない False Event を発生させることができる．セキュリティ侵害を受けたノードは，近くのイベントを「検知」したり，任意の場所のイベントを「転送」したりすることができる．このような嘘の通知は，ユーザに誤った判断をさせるだけでなく，ネットワークの貴重なリソース（エネルギーやネットワーク帯域）を消費させる．多くの False Event が発生した場合，実際に発生したイベントの通知が妨げられることになる．

セキュリティ侵害を受けたノードによる脅威は，イベントメッセージの破棄や DoS 攻撃等も存在する．だがこれらの脅威は他の研究[14),18)] 等で対策がとられており，本論文中では焦点を当てない．

## 3. 関 連 研 究

センサネットワークにおいて，各ノードに安全に鍵を配備する手法はさまざまに研究されているが，これら[4),10)] は，ノードがセキュリティ侵害を受け，ノード内の情報が抽出される状況は想定していない．

また，類似のセキュリティ課題として，セキュアなデータ集約に関する研究がある[3),16)]．これらの手法を工夫してイベント検知に利用することは可能であると考えられるが，In-network で嘘の通知を検知できないという問題がある．

False Event 検知に関する研究は，主にセンサへの鍵の配備方法に工夫がなされており，“ランダム鍵配備方式” と “地理制約鍵配備方式” の 2 手法に分類される．いずれの手法においても，共通の方針は次のとおりである．

### 3.1 False Event 検知の共通手法

まずセンサノードに重複を許して鍵を配備する．鍵にはそれぞれ鍵 ID が振られているものとする．イベント発生時には，ある小さな閾値 $T$ 個以上のノードがそのイベントを検知する必要がある．各ノードは，配備されていた鍵を用いて MAC（Message Authentication Code）を作成し，その鍵の ID とともにイベントリーダノード（EN）に通知する．EN は，イベント情報と $T$ 個の MAC，鍵 ID をあわせて 1 つのメッセージとし，ベースステーションへ通知する．中継ノードは，MAC 作成に使われた鍵 ID と，自分が持っている鍵 ID が一致している場合，その MAC の真正性を確認する．メッセージに $T$ 個の MAC が付加されていない場合や，真正性の確認に失敗した場合，このイベントを False Event と見なして破棄する．

したがって，あるイベントを $T$ 個以上のノードが同時に検知する必要がある．これは，イベントメッセージに $T$ 個の MAC を付加する手法に共通の制約であり，本提案手法でもこの制約を受ける．またセンサノードは，多くの場合に屋外に設置されるため物理的に故障することも考慮する必要がある．したがって，いくつかのセンサノードが故障した場合においても，できるだけ $T$ 個の異なる MAC を生成できるような耐故障性の要素も求められる．

また，$T$ 個以上の鍵を攻撃者に取得された場合，ベースステーションにおいても検知ができない False Event の発生が可能となる．これはすべての既存手法における共通の限界である．だが，検知ができない False Event を発生させることができる領域を限定できるかどうかは手法によって異なる．ランダム鍵配備方式では，攻撃者は，任意の $T$ 個の鍵を取得すると，全領域からこのような False Event を発生させることが可能となる．一方，地理制約鍵配備方式では，検知できない False Event の発生領域を，鍵を取得されたノードの領域にほぼ限定することができる．本提案手法においても，検知できない False Event の発生領域をある程度限定することができる．

### 3.2 ランダム鍵配備方式とその課題

文献 17) においては，センサノード全体に重複を許して鍵をランダムに配備する．ノードの位置情報が不必要であり，ノードが移動してもかまわないといった利点がある．だが，あるイベントが発生した際，そのイベントを $T$ 個以上のセンサが検知したとしても異なる $T$ 個の MAC が集まらない可能性がある．これは，鍵をランダムに配備しているためである．極端な例では，イベントを検知したすべてのセンサに同じ鍵が割り当てられていた場合，1 個の MAC しか集まらない．このような場合，ベースステーションまでイベントを通知する際，中継ノードによって False Event として破棄されてしまう．高い確率で異なる $T$ 個の MAC を集めるためには，ネットワーク全体で用いる鍵の種類を増やす必要がある．だが鍵の種類を増やすと，In-network で False Event であると検知できる確率が減少してしまう．

またランダム鍵配備方式においては，$T$ 個以上の鍵が漏洩した場合，センサネットワークの全領域から，検知することができない False Event を発生させられてしまう．

### 3.3 地理制約鍵配備方式とその課題

文献 8)，15)，21) においては，センサノードの位置情報に基づいて鍵を配備する．ある領域の $T$ 個のノー

ドの鍵が流出しても，検知することができない False Event の発生をその領域にほぼ限定することができる．

だが，前述したように，EN からベースステーションまでのイベント転送時に，独自のルーティングアルゴリズムを用いなければならないという制約がある．無線センサネットワークにおけるルーティング方法は数多く研究されており[6),19)]，センサの密度やエネルギー消費率等さまざまな指標によって最適なものが選択されるべきである．したがって，False Event 検知のためだけのルーティングを指定することは問題である．

## 4. 提 案 手 法

### 4.1 新しい概念の定義

まず本手法に必要な概念である，イベント検知ノード集合（SE），イベント検知円，イベント検知ノード最小集合（MSE）を定義する．

#### 4.1.1 イベント検知ノード集合（SE）とイベント検知円

ここではイベント検知ノード集合（SE）とイベント検知円という概念を導入する．本論文中では，センサが検知できるイベントまでの距離はすべてのセンサについて共通であり，$R_d$ であるとする．SE の定義は，

(1) $R_d$ を半径とする円に含まれる $T$ 個以上のセンサの組合せ

(2) 他の SE の部分集合ではない

をいずれも満たすセンサの集合である．$T = 3$ であるときの SE の例を**図 1** に示す．図 1 では，センサを六角形で表現している．また，各センサには ID を振ってある．図中に描かれている円は，半径 $R_d$ の円である．各円に含まれているセンサの集合が，それぞれ SE となる．

また，もしノード 1 が存在しない場合，円 $A$ に含まれる 3 つのノードは，円 $B$ にすべて含まれる．この場合，円 $A$ に含まれている 3 つのノード（ID2，ID3，ID4 のノード）の組合せは，SE の定義 (2) を満たさなくなるため，SE とはならない．

また，それぞれの SE を完全に含む半径 $R_d$ の円をイベント検知円と定義する．図 1 では，4 つの円すべてがイベント検知円となる．また，もしノード 1 が存在しない場合，円 $A$ に含まれている 3 つのノードの組合せは SE とならないため，円 $A$ はイベント検知円とはならない．

#### 4.1.2 イベント検知ノード最小集合（MSE）

各イベント検知ノード集合（SE）内において，$T$ 個のノードの組合せを，それぞれイベント検知ノード最小集合（MSE）と呼ぶ．図 1 では，円 $A$ については，

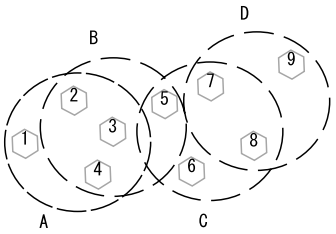



**図 1** イベント検知ノード集合（SE）の例（$T = 3$）
Fig. 1 Example of sets of event detection nodes ($T = 3$).

{1, 2, 3}，{1, 2, 4}，{1, 3, 4}，{2, 3, 4} の 4 つの MSE が存在する．他の円についてもそれぞれ MSE が存在する．

また，まったく同じノード群を含む MSE は同一視する．図の例では，円 $A$ と円 $B$ にそれぞれ {2, 3, 4} の組合せがあるが，これは 1 つの MSE と見なす．

### 4.2 提案手法の概要

本手法では既存研究と同じように，鍵を重複を許してセンサノードに配備する．重複を許すことで，In-network における False Event の検知が可能となる．我々独自の手法として，鍵配備は次の条件を満たすようにする．

(1) MSE 内のそれぞれのノードには，別々の鍵を配備する．

(2) それぞれの MSE には，別々の $T$ 個の鍵の組合せを配備する．

(1) により，$T$ 個のノードがイベントを検知した場合，確実に $T$ 個の MAC を収集することができる．

(2) により，ある $T$ 個のノードの鍵情報が流出したとしても，検知されない False Event を発生させることができるのは，その $T$ 個の鍵の組合せを持つ MSE に限定される．また，攻撃者が $T$ 個以上の鍵を取得して，広範囲の領域において False Event を発生させる状況を考える．ベースステーションは，全 MSE の $T$ 個の鍵の組合せを把握している．攻撃者が任意に $T$ 個の鍵を選択しても，その鍵の組合せを持つ MSE が存在しない可能性もあり，その場合はベースステーションにおいて False Event であると判定することができる．

以降で，本手法の詳細を述べる．

### 4.3 False Event の発生と False Event の検知

本節では，ノードを配置と鍵の配備，イベントが発生してからベースステーションへ通知されるまでの手順の詳細を述べる．

#### 4.3.1 センサノードへの準備

各ノード $n_i$ に，共通鍵 $\hat{K}_i$ を配備する．この鍵は前節において記述した False Event の検知のために必

要な鍵とは別の鍵である．False Event の検知に必要な鍵は，MSE の導出後にベースステーションから設定される．この鍵の配備は，ノードを対象領域に配置する前に行う．$i \neq j$ であるとき，$\hat{K}_i \neq \hat{K}_j$ であるとする．ベースステーションはすべてのノード ID とノードに与えた鍵を保持しておく．センサノードを対象領域に配置する際には，飛行機等を用いて空から散布する手法が考えられる．本提案手法においては，配置方法や配置条件に制限を設けない．センサノードを配置した後，各ノードは位置推定手法[1),13)] を用いることにより，自分の位置を把握することができるものとする．自分の位置 $L_i$ を把握したノード $n_i$ は，$L_i$ を $\hat{K}_i$ で暗号化したものをそれぞれベースステーションまで報告する．暗号化することにより，盗聴や改ざんの防止を行うことができる．ここでは，すべてのノードにそれぞれ異なる共通鍵を配備しているため，いずれかのノードが物理的に取得されたとしても，その他のノードに影響が及ぶことはない．

#### 4.3.2 SE と MSE の導出

すべてのノードの位置を把握したベースステーションは，イベント検知ノード集合（SE）とイベント検知ノード最小集合（MSE）を導出し，各 SE・MSE とノード ID のテーブルを作成・保持する．まず SE の導出アルゴリズムは次のようになる．

(1) 対象領域を細かく格子状に分割する．
(2) 1 つの格子点において，半径 $R_d$ の円を考える．
(3) 半径 $R_d$ の円内において，$T$ 個以上のノードがある場合，そのノードの組合せ $SE_i$ を，ノードの組合せの集合 $S_{SE}$ に追加する．
(4) 上記 (1)–(3) の処理をすべての格子点において行う．

この処理を行った結果，$S_{SE}$ にはすべての SE が含まれることになる．MSE の導出アルゴリズムは次のようになる．

(1) $S_{SE}$ から 1 つの $SE_i$ を取り出す．
(2) $SE_i$ の要素から，$T$ 個のノードの組合せをすべて抽出し，$MSE_1, ..., MSE_j$ とする．
(3) 抽出した $MSE_1, ..., MSE_j$ を，MSE の集合 $S_{MSE}$ に追加する．
(4) 上記 (1)–(3) の処理を $S_{SE}$ すべての要素に対して行う．

この処理を行った結果，$S_{MSE}$ にはすべての MSE が含まれることになる．

このようなアルゴリズムにより，与えられたノードの配置に対し，SE や MSE の集合は一意に定まる．また，ノードの密度に対して $T$ の値が大きすぎる場合は，SE や MSE が生成されない状況も考えられる．この問題は，イベントメッセージに $T$ 個の異なる MAC を必要とするすべての手法に共通の問題であり，1 つのイベントに対し高い確率で $T$ 個以上のノードが検知できるような $T$ の値の設定が必要である．

次に，ベースステーションは鍵配備条件 (1)，(2) を満たすような鍵を $M$ 個作成し，ノード ID と鍵 ID のテーブルを作成，保持する．各ノード $n_i$ に，それぞれ $\hat{K}_i$ で暗号化した鍵 $K_i$ と鍵 ID $i$ を 1 つずつ送信する．

#### 4.3.3 イベントの発生

イベントが発生したら，周りのノードがそれを検知し，その中から 1 つイベントリーダノード（EN）を決定する．EN の決定手法は本論文の範囲外であるが，たとえば文献 18) で用いられている手法を使うことができる．EN は，イベントを検知したノードの中から $T$ 個のノードを選択し，$\{L_E, t, E\}$ の情報を通知する．ここで，$L_E$ はイベントが発生した場所，$t$ はイベントが発生した時刻，$E$ はイベントの型を表す．イベントには他の情報が含まれることもあるが，ここでは文献 17) にならってこの情報のみを保持することにする．次に各ノードは，

$$M_i = \overline{MAC}(K_i, L_E||E) \quad (1)$$

を作成する．ここで，|| はストリームの結合を表し，$\overline{MAC}(a, b)$ は鍵 $a$ を用いて作成されたメッセージ $b$ の MAC を意味する．また，$K_i$ はノードに配備されている鍵である．ノードは，$\{i, M_i\}$ を EN に通知する．$i$ は MAC の作成に使用した鍵の ID である．EN からベースステーションへ通知されるメッセージの内容は，$\{L_E, E, M_{i_1}, i_1, ..., M_{i_T}, i_T\}$ となる．

また，イベントリーダノードが選択したノードは，必ず同一のイベント検知ノード最小集合に含まれる．これは，同一のイベントを検知できる任意の $T$ 個のノードの組合せを，同一のイベント検知ノード最小集合と定義しているからである．

#### 4.3.4 中継ノードの処理

中継ノードがイベントメッセージを受け取ったとき，メッセージの中に $T$ 個の鍵 ID と MAC が含まれているかを調べる．鍵 ID や MAC が $T$ 個より少ない場合はイベントメッセージを破棄する．

また，各中継ノードは，イベントメッセージ内の MAC 作成に使われた鍵を保持していた場合，その鍵で MAC を作成し，イベントメッセージに含まれる MAC と比較する．同じ MAC が作成されなかった場合はイベントメッセージを破棄する．

破棄しないイベントメッセージは，そのままベース

ステーションまで転送を続ける．

#### 4.3.5 ベースステーションの処理

ベースステーションがイベントメッセージを受け取ったとき，$T$ 個すべての MAC の正しさを調べることができる．なぜならベースステーションはすべての鍵を保持しているからである．$T$ 個の MAC の中で 1 つでも捏造されたものがあれば，ベースステーションで False Event であると確実に判断することができる．False Event であると判断されたイベントメッセージは破棄される．

次に，鍵 ID の $T$ 個の組合せが，イベント検知ノード最小集合（MSE）のいずれかと一致しているかを調べる（ベースステーションは，すべての MSE を把握している）．すべての MSE と一致しない場合は，False Event であると判断し，このイベントメッセージを破棄する．

### 4.4 鍵の数の決定

本節では鍵の数 $M$ の決定方法を述べる．前述した鍵配備の条件を満たすように配備する．まず，用いる鍵の数 $M$ を導出する．

以降では，イベント検知ノード集合（SE）の総数を $N_{SE}$，イベント検知ノード最小集合（MSE）の総数を $N_{MSE}$ とおく．また，議論を簡潔に行うため，これ以降は各 SE に含まれるノードの数を $N_a$ で一定であるとする．

鍵 $M$ 個から $T$ 個選ぶ組合せの数は ${}_MC_T$ 通りである．すべての MSE に別の組合せの鍵を配備するためには少なくとも，

$$ {}_MC_T \geq N_{MSE} \tag{2}$$

を満たす $M$ 個の鍵が必要である．式 (2) を満たす最小の $M$ を $M_{min}$ とする．ただし鍵配備条件を満たすためには，ノード配置の地理的な制約上，$M_{min}$ 以上の鍵が必要な場合もある．

また，利用する鍵の数が少ないほど，False Event を In-network で検知できる確率が向上する．これは，各ノードが持つ鍵の重複率が向上するためである．しかし，攻撃者が $T$ 個以上の鍵を取得したとき，多くの MSE における鍵の組合せを復元することができてしまう．したがって，$T$ 個以上の鍵が流出した場合を考えると，利用する鍵は多いほうが良い．

そこで新たに**重複パラメータ**$a$（$a > 0$）を導入し，実際に利用する鍵の数を

$$ M = a \times M_{min} \tag{3}$$

とおく．$a$ が大きいほど，鍵の重複度が小さくなり，In-network 内で False Event を検知できる確率が減少する．だが，$T$ 個以上の鍵が漏洩したときの被害を抑えることができる．

## 5. 数学的な解析

まず $N_c$ 個（$N_c < T$）のノードの情報が流出した場合，平均何ホップで False Event を検知できるかを求める．次に，$N_c$ 個（$N_c > T$）のノードの情報が流出した場合，検知できない False Event の領域の，全センサネットワークの領域に対する割合 $\Re$ を求める．

これらを求めるために，$N_c$ 個のノードの情報が流出した場合に何個の鍵 $N_k$ が流出することになるかを考える[⋆1]．

センサノード数が十分に多い場合，$N_c$ 個のノードをセキュリティ侵害してちょうど $N_k$ の鍵を取得できる確率を $P(M, N_c, N_k)$ とする．このとき，$N_k$ の期待値は，

$$ N_k = \sum_{i=1}^{N_c} i \cdot P(M, N_c, i) \tag{4}$$

となる．

また，

$$ P(M, N_c, N_k) = \frac{{}_MC_{N_k} \cdot \left( N_k^{N_c} + \sum_{i=1}^{N_k} [(-1)^i \cdot {}_{N_k}C_{N_k - i} \cdot (N_k - i)^{N_c}] \right)}{M^{N_c}} \tag{5}$$

と表すことができる．

次に，$T$ 個未満の鍵が漏洩しているときに，False Event を 1 ホップで検知できる確率 $p_1$ を求める．攻撃者は $N_k$ 個のみの鍵を取得している場合，$T - N_k$ 個の MAC を捏造しなければならない．鍵全体の数が $M$ であり，1 ホップ目のノードが特定の $T - N_k$ の鍵のいずれかを保持している確率は，

$$ p_1 = \frac{T - N_k}{M} \tag{6}$$

である．

また，検知できるまでの平均ホップ数 $p_h$ は，

$$ p_h = \sum_{i=1}^{H} i \cdot (1 - p_1)^{i-1} \cdot p_1 = \frac{1 - (1 - p_1)^H}{p_1} \tag{7}$$

と表現される．ここで，$H$ はベースステーションまでの最大ホップ数を表す．

次に，攻撃者が $N_c$ 個のノードの情報を取得した場合に，検知されない False Event を発生させることが

⋆1 鍵は重複を許して配備されているため，$N_c$ 個のノードがセキュリティ侵害を受けても，実際に漏洩する鍵は $N_c$ 個以下である．

できる領域の割合 $\Re$ を求める．イベント検知円内の $T$ 個のノードの鍵が流出した時点で，そのイベント検知円内のいずれの場所の False Event も検知することができなくなる．ここでは，各イベント検知円に含まれるノード数を，一定値 $N_a$ であるとする．また，$N_c$ 個のノードの情報から得られる鍵の数 $N_k$ を式 (4) から求める．漏洩した $N_k$ 個のうち $i$ 個の鍵を配備されうるイベント検知円の総数は，$S(i) = {}_{N_k}C_i \cdot {}_{M-N_k}C_{N_a-i}$ である（$M$ は，ノードに配備された鍵の総数である）．

また，イベント検知円内のノードに配備する鍵はすべて違うものであるから，鍵 $M$ 個を使って配備できるイベント検知円の最大数は，$DC_{max} = {}_MC_{N_a}$ 個である．実際には，このうち $N_R$ 個だけのイベント検知円が存在するものとする．したがって，漏洩した $N_k$ 個のうち $i$ 個の鍵を配備されたイベント検知円の数の期待値は $S(i) \cdot N_R/DC_{max}$ となる．

$\Re$ は，検知されない False Event を発生させることができるイベント検知円の数の期待値を，イベント検知円の総数（$N_R$）で割ることによって導かれる．したがって，

$$\Re = \sum_{i=T}^{N_a} S(i) \cdot \frac{N_R}{DC_{max}}/N_R = \sum_{i=T}^{N_a} \frac{{}_{N_k}C_i \times {}_{M-N_k}C_{N_a-i}}{{}_MC_{N_a}} \quad (8)$$

となる．

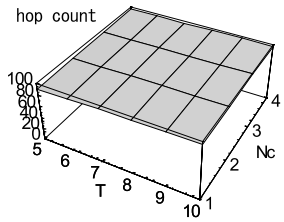
## 6. 評　　価

### 6.1 数学的な解析による評価

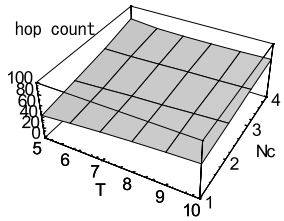
本節では，提案手法を数学的な解析により評価する．また，ランダム鍵配備方式の SEF [17] との比較も行う．多くの既存研究[8),9),15),17)] の設定に従い，1 つのベースステーションを配置し，検知されなければ，ベースステーションまで 100 ホップ程度かかる距離から False Event を発生させる状況を考える．

まず既存研究である SEF [17] について，論文中に示されている数式に基づいて評価を行った．結果を**図 2** に示す．図における縦軸の hop count は，False Event であると検知するまでに必要なホップ数の期待値である．ノード総数は 10,000 に設定し，1 つのイベントを検知できるノード数は $T + 1$ に設定した．

SEF では，全体の鍵の数をシステムのパラメータとして設定できる．全体の鍵の数を少なくすると，False Event であると検知できるまでのホップ数は減少する．だが実際に起きたイベントを $T$ 個以上のノードが検知した場合でも，異なる $T$ 個の MAC を集められる確率も減少してしまう．したがって，実際にイベントが起きた際にも，異なる $T$ 個の MAC を付加したイベントメッセージを作成することができない．




(a) 実際に発生したイベントの消失率=1%

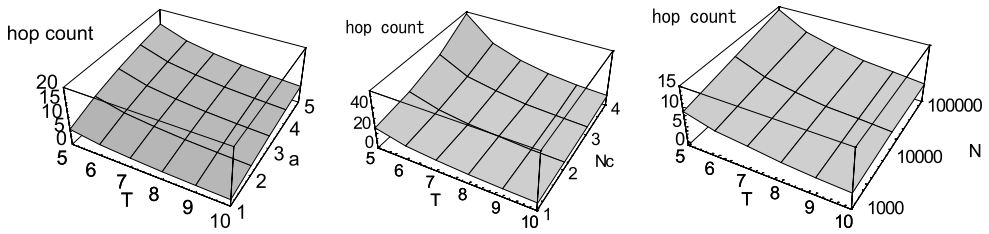



(b) 実際に発生したイベントの消失率=10%

**図 2** SEF における False Event 検知までの平均ホップ数
Fig. 2 Number of hops traveled in SEF.

評価では，センサネットワークの耐故障性を考慮し，イベント検知ノード集合のノードのうち 1 つのノードが故障しても，イベントリーダノードが 90%または 99%の確率で異なる $T$ 個の MAC を集めることができる値にパラメータを設定した．つまり残りの 10%または 1%の確率において，実際にイベントが発生しても異なる $T$ 個の MAC を集めることができず，イベントメッセージは False Event として処理され消失する．一方，本論文の提案手法では，1 つのノードが故障しても理論上は 100%の確率で異なる $T$ 個の MAC を集めることができる．またその他の SEF 特有のパラメータは，文献 17) のデフォルトの値に従った．図 2 (a) より，イベント消失率を 1%程度に抑えた場合，SEF の手法では False Event の検出までに非常に多くのホップ数を要するようになることが分かる．イベント消失率を 10%程度に抑えた場合は，図 2 (b) より，約 30 から 80 ホップ程度において，False Event の検知に成功していることが分かる．だが依然として，False Event を検知するまでに必要なホップ数は多い．

次に本論文の提案手法を用いて解析を行った．**図 3** に解析結果を示す．$N$ はノード総数，$N_c$ はセキュリティ侵害を受けたノードの数，$a$ は重複パラメータ，$T$ は 1 つのイベントメッセージに付加する MAC の数である．また，イベント検知ノード集合は，各実験において $T + 1$ に設定した．




(a) $N = 10^4, N_c = 1$ (b) $N = 10^4, a = 3$ (c) $a = 3, N_c = 1$

**図 3** False Event 検知までの平均ホップ数
Fig. 3 Number of hops traveled.

図 3 (a) では，イベントメッセージに付加する MAC の数 $T$ と，鍵の重複パラメータ $a$ を可変パラメータに設定して解析を行った．$T$ の値が大きいほど，少ないホップ数で False Event を検知できていることが分かる．攻撃者が $N_c$ 個の鍵を取得している場合，攻撃者は $T - N_c$ 個の架空の MAC をイベントメッセージに付加しなければならない．このとき中継ノードが，イベントメッセージを False Event であると検知するためには，架空の MAC 作成に用いられた $T - N_c$ 個のいずれかの鍵を保持していればよい．したがって，$T$ の値が大きいほど少ないホップ数で False Event を検知できる．また，重複パラメータ $a$ が大きいほど，False Event の検出までに多くのホップ数を要する．重複パラメータが大きいということは，ネットワーク全体で使用する鍵の数が多いということである．ネットワーク全体で使用する鍵の数が多いと，イベントメッセージに付加されている MAC に使用された鍵と同じ鍵を持つノードの数が減るため，False Event を検知できるまでのホップ数は増加する．

図 3 (b) では，重複パラメータ $a$ を 3 に固定し，$T$ と $N_c$ を可変パラメータに設定している．$T$=5 かつ $N_c$=4 である状況は，イベントメッセージに付加された 5 個の MAC のうち 4 個の MAC が正しいものであるという状況である．このように，$T$ 個の MAC のうち 1 つの MAC だけが正しくない場合にでも，少ないホップ数で False Event の検知ができることが分かる．

図 3 (c) では，センサネットワークのノード数 $N$ を 1,000 から 100,000 に変化させて解析を行った．ノード総数の増加にともなって False Event を検知するまでのホップ数が増大していることが分かる．これは，ノード数が増加するとネットワーク全体で使用する鍵の数が増加するため，イベントメッセージにおける MAC の作成に用いられた鍵を持つノードの数が減少するからである．だが図より，ノード数の増加に対し

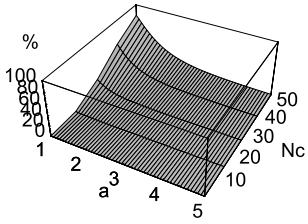



**図 4** False Event 検知不可領域の割合
Fig. 4 Portion of region where attackers can generate an undetected false event.

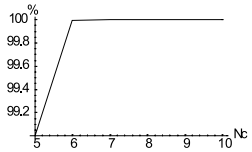



**図 5** SEF における False Event 検知不可領域の割合
Fig. 5 Portion of region where attackers can generate an undetected false event in SEF.

False Event を検知できるまでのホップ数の増加は比較的限定されていると考えられる．

次に，攻撃者が $N_c$ 個のノードから情報を抽出したとき，検知されない False Event を発生させることができる領域の割合を算出した．$T$=5 に設定した．結果を**図 4** に示す．$a = 3$ 以上であるとき，ノード 50 個から情報が漏洩していても，検知できない False Event が発生できる領域はごくわずかに限られていることが分かる．一方ランダム鍵配備方式では，ノード 5 個から情報が漏洩した時点で，検知できない False Event が発生できる領域はほぼ 100%となる（**図 5**）．

### 6.2 シミュレーションによる評価

より正確な解析を行うため，シミュレーションによる評価も行った．$50 \times 200\,\mathrm{m}^2$ のフィールドに 10,000 ノードをグリッド上に配置した．この際，すべてのイベント検知ノード集合（SE）に 6 個のノードが含まれるようにした．1 つのベースステーションを配置し，

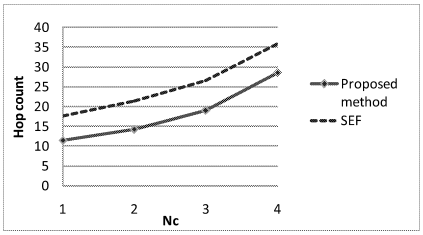



**図 6** False Event 検知までの平均ホップ数（SEF におけるイベント消失率 = 10%）
Fig. 6 Number of hops traveled (disappearance rate of legitimate events of SEF is 10%).

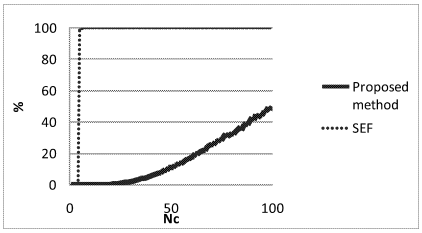



**図 7** False Event 検知不可領域の割合
Fig. 7 Portion of region where attackers can generate an undetected false event.

ランダムにノードを選択して False Event を発生させた．ベースステーションから最も遠いノードにおいて False Event を発生させた際，ベースステーションまでおおむね 100 ホップを必要とした．実験では $a = 3$, $T = 5$ に設定した．

1 つ目の実験では，$N_c$ 個のノードの鍵が流出しているとき，平均何ホップで False Event を検知できるかを測定した．手順は次のとおりである．まず全体の $M$ 個の鍵の中からランダムに $N_c$ 個の鍵を選択する．また，任意に $T - N_c$ 個の鍵を捏造する．これらの鍵を用いて False Event を発生させ，検知できるまでのホップ数を測定する．この試行を各 $N_c$ の値に対して 1,000 回ずつ行った．**図 6** に，SEF の結果と提案手法の結果を載せている．SEF においては，実際にイベントが発生した際にそのイベントが False Event であると誤判断される確率（イベントの消失率）を 10%に設定した．図 6 より，提案手法では，$N_c$ の増加にともなって，False Event を検知するまでのホップ数は増加しているが，SEF の手法よりも少ないホップ数で検知できていることが分かる．

次に，$N_c$ 個のノードの鍵が流出しているとき，False Event を検知することができない領域が全体の何%になるか測定をした．手順は次のとおりである．提案手法においては，最初の実験と同様に，全体の $M$ 個の鍵の中からランダムに $N_c$ 個の鍵を選択する．この中のいずれかの鍵を割り当てられているノードにフラグを立てる．構成要素のノードすべてにフラグが立っている MSE を抽出する．抽出された MSE を構成要素とする SE の数を数え（重複は除く），全体の SE の数で割る．このように導出された値を測定した．この試行を各 $N_c$ に対して 1,000 回ずつ行った．SEF においては，ノードをランダムに $N_c$ 個選択した際に，何%の確率で $T$ 種類以上の鍵を取得できたかを測定した．結果を**図 7** に示す．SEF においては，ノードを 5 個程度攻撃された時点で，攻撃者はほぼ 100%の領域から，検知することのできない False Event を発生させることができる．一方，提案手法においては，ノードを 10 個攻撃された時点では，そのような領域は全体のわずか 2%である．100 個のノードを取得された状況においても，検知することのできない False Event を攻撃者が発生させることができる領域は，全体の約半分程度である．

## 7. 考　　察

### 重複パラメータについて

前章の評価で解析したとおり，本論文で提案する手法は，重複パラメータ $a$ の設定が結果に大きな影響を与える．$a$ の値が小さい場合，ネットワーク全体で使用する鍵の数が少ないため，同じ鍵を違うノードが保持している確率が高くなり，False Event を少ないホップ数で検知することが可能となる．一方で $T$ 個の鍵の組合せの種類数も同様に少なくなるため，攻撃者が $T$ 個以上の鍵を取得した際，より広範囲の領域から，検知されることのない False Event を発生させることが可能となってしまう．$a$ の値は，アプリケーションの要求と，鍵の更新手法[9),20)] を利用するかどうかを考慮して設定すべきであると考えられる．

アプリケーションが軍事目的の使用である等，ベースステーションにおいても検知できない False Event の発生をできるだけ防ぎたい場合には，$a$ の値を大きく設定すべきであると考えられる．一方，イベントが発生した場合，対応する場所に出向くことでそのイベントが本当に発生したものなのか False Event なのかを簡単に判定することができるようなアプリケーションであれば，$a$ の値を小さく設定することができる．

また，本論文では鍵の更新については言及していないが，鍵の更新を行う研究もいくつか存在する[9),20)]．鍵の更新を行うことで，攻撃者に鍵が流出していたとしても，その鍵の効果を失わせることができる．鍵の更新はトラフィック量や計算量のコストが大きいため頻繁に行うことはできないが，攻撃者が数多くのノー

ドから鍵を抽出する前に鍵の更新を行うことができるのであれば，重複パラメータ $a$ の値を小さく設定しておくことが可能である．

**ノードのエネルギー消費について**

False Event を In-network において検知するのは，ノードのエネルギー消費量を節約するためである．少ないホップ数で False Event を検知し破棄することで，ノードのエネルギー消費を抑えることができる．だが，各ノードが False Event を検知するために行う MAC の計算にもエネルギーを要する．文献 12) によると，1 ビットの転送に要するエネルギーは 1,000 回の計算サイクルに要するエネルギーに相当する．したがって，MAC の計算に必要なエネルギーよりも，False Event を転送するために必要なエネルギーのほうが問題になると考えられるため，本研究においては False Event を In-network において検知することを目標の 1 つとした．だが，正しいイベントの数よりも False Event の数が十分に小さい場合，False Event を In-network において検知しないほうがエネルギー消費量が少なくなる可能性がある．将来課題として，ノードのエネルギー消費の定量的評価を行うべきであると考えている．

## 8. おわりに

本研究では，独自のルーティング方法を指定せず，In-network における False Event の高確率の検知を目標とした．独自のルーティング方法を指定しない既存研究では，実際に発生したイベントが確率的に消失するという課題があった．消失する確率を 1%に抑えたとき，本研究が提案する手法が，この既存手法よりも大幅に少ないホップ数で False Event の検知を行うことができた．

また，多くのノードの情報が流出した際にも，検知できない False Event を発生させることができる領域を限定することが可能となった．

将来課題としては，ノードが移動したときの処理を考える必要がある．また，実機のセンサへの実装を行うことを考えている．

## 参考文献


1) Bruck, J., Gao, J. and Jiang, A.A.: Localization and routing in sensor networks by local angle information, *ACM MOBIHOC*, pp.181–192 (2005).
2) Chan, H., Perrig, A. and Song, D.: Random Key Predistribution Schemes for Sensor Networks, *IEEE Symposium on Security and Privacy*, pp.197–213 (2003).
3) Chan, H., Perrig, A. and Song, D.: Secure hierarchical in-network aggregation in sensor networks, *IEEE CCS*, pp.278–287 (2006).
4) Du, W., Deng, J., Han, Y.S., Chen, S. and Varshney, P.K.: A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge, *IEEE INFOCOM* (2004).
5) Fang, Q., Gao, J. and Guibas, L.J.: Locating and bypassing holes in sensor networks, *Mob. Netw. Appl.*, Vol.11, No.2, pp.187–200 (2006).
6) He, T., Krishnamurthy, S., Luo, L., Yan, T., Gu, L., Stoleru, R., Zhou, G., Cao, Q., Vicaire, P., Stankovic, J.A., Abdelzaher, T.F., Hui, J. and Krogh, B.: VigilNet: An integrated sensor network system for energy-efficient surveillance, *ACM Trans. Sen. Netw.*, Vol.2, No.1, pp.1–38 (2006).
7) Karp, B. and Kung, H.T.: GPSR: Greedy perimeter stateless routing for wireless networks, *ACM MOBICOM*, pp.243–254 (2000).
8) Kraub, C., Schneider, M., Bayarou, K. and Eckert, C.: STEF: A Secure Ticket-Based Enroute Filtering Scheme for Wireless Sensor Networks, *IEEE ARES*, pp.310–317 (2007).
9) Li, W., Zhang, Y. and Yang, J.: Dynamic Authentication-Key Re-assignment for Reliable Report Delivery, *IEEE MASS*, pp.467–476 (2006).
10) Liu, D. and Ning, P.: Establishing pairwise keys in distributed sensor networks, *IEEE CCS*, pp.52–61 (2003).
11) Perrig, A., Szewczyk, R., Wen, V., Culler, D.E. and Tygar, J.D.: SPINS: Security protocols for sensor networks, *ACM MOBICOM*, pp.189–199 (2001).
12) Raghunathan, V., Schurgers, C., Park, S. and Srivastava, M.: Energy aware wireless microsensor networks, *IEEE Signal Processing Magazine*, Vol.19, No.2 (2002).
13) Stoleru, R., Vicaire, P., He, T. and Stankovic, J.A.: StarDust: A flexible architecture for passive localization in wireless sensor networks, *ACM SENSYS*, pp.57–70 (2006).
14) Wood, A.D. and Stankovic, J.A.: Denial of Service in Sensor Networks, *IEEE Computer*, Vol.35, No.10, pp.54–62 (2002).
15) Yang, H., Ye, F., Yuan, Y., Lu, S. and Arbaugh, W.: Toward resilient security in wireless sensor networks, *ACM MOBIHOC*, pp.34–45 (2005).
16) Yang, Y., Wang, X., Zhu, S. and Cao, G.: SDAP: A secure hop-by-Hop data aggregation protocol for sensor networks, *ACM MOBIHOC*, pp.356–367 (2006).

17) 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, Special Issue on Self-organizing Distributed Collaborative Sensor Networks*, Vol.23, No.4, pp.839–850 (2005).
18) 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).
19) Zhang, H., Arora, A. and Sinha, P.: Learn on the Fly: Data-driven Link Estimation and Routing in Sensor Network Backbones, *IEEE INFOCOM* (2006).
20) Zhang, W. and Cao, G.: Group rekeying for filtering false data in sensor networks: A predistribution and local collaboration-based approach., *IEEE INFOCOM*, pp.503–514 (2005).
21) Zhu, S., Setia, S., Jajodia, S. and Ning, P.: An Interleaved Hop-by-Hop Authentication Scheme for Filtering False Data Injection in Sensor Networks, *IEEE Symposium on Security and Privacy*, pp.259–271 (2004).

(平成 19 年 5 月 18 日受付)
(平成 19 年 11 月 6 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．