Vol.2013-CSEC-61 No.16
Vol.2013-IOT-21 No.16
2013/5/10

低消費電力と高安全性を両立する新しい LEACH プロトコル

齊藤壮馬[†1]　　岩村惠一[†1]

ワイヤレスセンサネットワークを構成するセンサノードは小さい電池容量であり、リソースも少ないため効率的な利用が求められてくる．LEACH(Low Energy Adaptive Clustering Hierarchy)はクラスタヘッドを周期的に選択することで、各ノードの消費電力を平均化するプロトコルである．このプロトコルに対するセキュリティ対策はいくつか提案されているが、電力消費が大きく、クラスタヘッドと通信できない孤立ノードを発生される可能性がある．そこで、階層型鍵共有方式を適用することで消費電力が小さく孤立ノードが発生しない方式を提案し、消費電力などをシミュレーションにより評価する．

The new LEACH protocol balancing low power consumption and high security

SOMA SAITO[†1]　　KEIICHI IWAMURA[†1]

Sensor nodes costructing wireless sensor network have small electrical power and low resource so we need to use them efficiently in terms of energy consumption. LEACH(Low Energy Adaptive Clustering Hierarchy), the protocol that changes cluster head periodically leads to average each nodes' power consumption. At this time, some researchers have suggested security systems of LEACH but they induce high energy consumption and have a possibility that some nodes don't connect with others in the network. In this paper, we propose the new LEACH protocol that improves these problems using hierarchical-structured preshared key scheme. And we estimate its necessary energy and so on.

1. はじめに

センサネットワークとは、情報収集を目的としたアドホックネットワークである．センサネットワークはセンサノード（ノード）と基地局で構成され、センサノードは温度や湿度、加速度といった環境情報収集し、それをダイレクトに、あるいは複数のノードにマルチホップさせ基地局へデータを転送する．使用目的は軍事的利用から環境保護まで多岐にわたる．

センサノードは PC のような高機能デバイスとは異なり簡易的で小型なものである故、各ノードが保有するリソースに制約がある．まず簡易的であるため耐タンパー性に乏しい．そのため不正ユーザがノードを分析して内部情報を知ることが可能である．また比較的安価であるため CPU も低性能である．このため高度な計算処理をセンサノードで行わせることができない．よって、セキュリティに関しては、公開鍵暗号を使用することは適当ではない．またメモリも小さいものであるため、記憶させる情報量もできる限り小さくする必要がある．そしてセンサノードは電池を搭載して駆動させるため､使える電力に制限がある．従って、センサネットワークでは効率的にエネルギーを利用できる方式を研究することが重要である．

このエネルギーを効率的に利用することを目的として、様々な手法が研究されてきた．まず、ノードが観測データを他のノードに送信しそれを他のノードが転送する手法が考えられた．送信電力は通信相手の距離が短いほど小さくなるので、近隣とのノードの通信では消費エネルギーを小さくできる．しかし基地局に近いノードほど転送量が増えるため、基地局に近いノードが遠いノードより先に電池切れになってしまう．そして基地局と各ノードが直接通信を行う手法も考えられた．だがノードが基地局より遠方になればなるほど消費電力は大きくなるので、基地局に近いノードより遠方のノードの方が先に電力を使い果たしてしまう．そこでクラスタ化を行い、直接 BS と通信を行うクラスタヘッドを順番に変えていく手法が考案された．この手法はノードを数個のグループにわけて、そのグループ内のノード１つにリーダー（クラスタヘッド、CH）を定める．各クラスタのノードはデータをクラスタヘッドに転送し、クラスタヘッドはそのデータを基地局へ転送する．基地局から遠方に位置するノードはより近いクラスタヘッドにデータを転送すればよいので、必要な消費エネルギーを抑えることができる．しかしクラスタヘッドは基地局と受信と送信を行わなければならず、エネルギーの消費が全体としてクラスタヘッドに集中してしまう．そこでこのクラスタヘッドを動的に変えていく手法が LEACH[1]である．LEACH(Low-Energy-Adaptive-Clustering-Hierarchy)は１つのノードがクラスタヘッドとなっている期間をラウンドと呼び､各ノードが順番にクラスタヘッドになることを申告し、基地局の命令なしに自立的にクラスタを編成する．最大の特徴はある周期ごとに全てのノードがクラスタヘッドになるようにする点である．このことによりクラスタヘッドによる消費エネルギーの偏りを全てのノードに平均化し、ネ

†1 東京理科大学
Tokyo University of Science

ットワークの寿命を長くすることができる.

しかしそのプロトコルはセキュリティを考慮していない. 例えば盗聴やメッセージが改竄されたことを検知または防御する機能が無い. また、外部から追加された不正ノードがクラスタヘッドになることを防止できない. このLEACHに対するセキュリティ対策に関する研究として、SLEACH[2]やSecLEACH[3]がある. SLEACHはメッセージ認証子（MAC）を利用することで通常ノード、クラスタヘッドが正当なクラスタメンバであることを保証する. しかし各ノード間に共通鍵が無いため、暗号通信が不可能であり、機密性に欠ける. 一方、SecLEACHではLEACHに対してランダム鍵配布方式を用いることでセキュリティ対策を付加している. しかしランダム鍵配布方式の欠点である、鍵共有ができないノードが存在してしまう点と、多くの鍵IDを知らせる必要がある点によりエネルギー消費が大きなものとなっている. それに対して本稿では階層型鍵共有方式を利用するLEACHを提案する. この鍵共有方式を利用することで、必ずどのノード間でも鍵共有することができ、鍵IDを送信する必要がなくなるため、エネルギーをより抑えることができる.

本論文の構成は、まずLEACHプロトコルの詳細について述べ、関連研究としてLEACHに対するセキュリティ付加方式のSLEACH、SecLEACHの概要を説明する. 次に提案方式の特徴である階層型鍵共有方式について説明をし、提案方式のプロトコルを示す. そして提案方式に対する安全に関する評価を行う. 最後にLEACH、既存研究と提案方式の比較のためのシミュレーションを示す.

2. 従来研究

2.1 LEACH

LEACH(Low Energy Adaptive Clustering Hierarchy)とは自律的にクラスタを構成し､クラスタヘッドを周期的に変え、全ノードの消費エネルギーの平均化を図る方式である. クラスタヘッドは通常の環境情報の観測とデータの送信に加え、近隣ノードのデータの受信、集約を行うため電力消費が集中する. LEACHではクラスタヘッドは周期的に全ノードがなるため、電力消費の集中による電池切れを防ぐ役割を果たす. なお、基本的なLEACHによる方式では、基地局への送信は２ホップになる. そして、基地局は十分なリソースを有しており、センサノードはリソースが小さいことを前提としている. またセンサノードは最大出力で基地局と直接通信が可能であると仮定する.

次にLEACHのプロトコルのアルゴリズムを説明する. LEACHはsetup phaseとsteady-state phaseの2段階が存在する. setup phaseではクラスタヘッドの決定や通常ノードの送信先クラスタヘッドの設定を行い、steady-state phaseでは観測データの転送を行う. にLEACHプロトコルを示す. なお、setup phase、steady-state phaseを合わせたものを1ラウンドと定義する.

setup phaseではまず各ノードがクラスタヘッドになるかどうかを判断する. 各ノードは0から1までの乱数を生成する. その乱数が下記の数式で与えられる閾値未満の際に、ノードはそのラウンドの間クラスタヘッドとなる.

$$T(n)=\begin{cases}\dfrac{P}{1-P\times(r \bmod \frac{1}{P})} & if \quad n\in G \\ 0 & otherwise\end{cases} \quad \cdots (1)$$

ここで、nはノードの番号、Pは全ノードにおけるクラスタヘッドの割合、rはラウンド数、Gは過去1/Pラウンドにおいてクラスタヘッドにならなかったノード集合を示す. この閾値を使うことで各ノードは1/Pラウンドにおいて1回だけクラスタヘッドになることが保証される. 1/P - 1ラウンド後、閾値は1となり全てのGに含まれるノードがクラスタヘッドとなる. そし1/P後、すべてのノードがGに含まれることになり再び1/Pラウンド間、クラスタヘッドになる可能性を持つこととなる. この閾値に基づいてクラスタヘッドになったノードは他のノードに対してメッセージadvと自分のノードID id_{Ai}をブロードキャストする. この時クラスタヘッドはCSMA MAC protocolを使用する. 全てのクラスタヘッドが送信を終えると、通常ノードは信号の強さに基づいて近隣のクラスタヘッドに対してメッセージjoin_req"、クラスタヘッドID id_Hと自分のノードID id_{Ai}送る. これによって、通常ノードはそのクラスタヘッドのクラスタメンバになることを申請する. 全ての通常ノードがこれを終えた後、クラスタヘッドはメッセージを受け取った通常ノードに対してTDMAスケジュール t_{Ai}を割り当て、割り当てたノードID id_{Ai}とセットにしてブロードキャストする.

TDMAスケジュールの割り当てが終わると、steady-state phaseに移行する. 各ノードはそのスケジュールに従ってデータをクラスタヘッドに転送を開始する. 通常ノードはノードIDと、送信先クラスタヘッドID、実際のデータを送る. クラスタヘッドは受信データを結合し、圧縮する. 例えばセンサノードを用いて温度の平均値を測定する場合、クラスタヘッドには各ノードにおける温度データが転送されてくる. クラスタヘッドはそれらを全て加算しておく. 最終的に基地局へ転送された時に全ノード数で割ることで、平均値が得られる. クラスタヘッドの受信データを圧縮後、基地局へデータを転送する. ここではデータ結合関数($F()$)としてデータの結合、圧縮を表現している. 基地局へデータを転送後、再びsetup phaseに戻り各ノードがクラスタヘッドになるか否かを決定する段階へ移行する.

ところで､同一クラスタ内の通信にTDMAが使われていても、複数のクラスタを考慮すると信号が衝突する可能性がある. これを防ぐため各ノードはCDMAを利用する. クラスタヘッドとなった際ノードはCDMAに使われるコー

ドリストから使用するコード(spreading code)を無作為に選ぶ．このコードを最初の、通常ノードにブロードキャストする際に知らせておく．

Setup phase
1. $H \Rightarrow G$: id_H, adv
2. $A_i \rightarrow H$: id_{A_i}, id_H, $join\_req$
3. $H \Rightarrow G$: id_H, (..., <id_{A_i}, t_{A_i}>,...), $sched$

Stead-state phase
4. $A_i \rightarrow H$: id_{A_i}, id_H, d_{A_i}
5. $H \rightarrow BS$: id_H, id_{BS}, $F(..., d_{A_i}, ...)$

記号の意味

A_i, H, BS	:	ノード、クラスタヘッド、基地局
G	:	基地局含むノード全体
⇒、→	:	ブロードキャスト、ユニキャスト
id_X	:	ノード X の ID
d_x	:	観測データ
<id_X, t_x>	:	ノード X の ID とタイムスロットスケジュール
adv, $join\_req$, $sched$	:	メッセージ識別用コード
$F()$	:	データ結合関数

図 1 LEACH protocol

2.2 SLEACH

SLEACH は SPINS[4]というセキュリティを付加したセンサネットワークプロトコルの構造を利用した LEACH である．特徴としてメッセージ認証子（MAC）を利用している．MAC とは入力値を鍵とメッセージとして生成される値である．一方向性関数を使用しているため得られた値から鍵とメッセージは求めることができないという性質を持つ．メッセージは公開情報でも構わないが、鍵は秘密情報であるため、正当な鍵を持つノードでないと正しい MAC を生成することができない．また入力値にカウンタを利用することで、再送攻撃を防ぐ．カウンタはラウンドごとに 1 つずつ増加する．これは SPINS の SNTP という構造を元にしている．

また SLEACH では μ TESLA という方式を利用して基地局から送られる正当なクラスタヘッドのリスト V を各ノードに知らせる．このことにより、正当な基地局からのリストであることが保証される．まず通常ノードと基地局は時刻同期が可能であると前提条件をおく．基地局は事前にハッシュ関数に通した数列を準備しておき、最後のハッシュに通して得られた値 k^j をノードと共有しておく．そして V と、k^{j-1} を鍵、入力値をリスト V とした MAC：MAC$k^{j-1}(V)$ をブロードキャストする．送信後 k^{j-1} を各ノードにブロードキャストする．なおハッシュ関数を $f()$ とすると $f(k^{j-1}) = k^j$ である．ノードは受信した鍵をハッシュ関数にかけ、持っていた鍵 k^j と値が一致するかどうか確認する．これが一致するならば正当な基地局からのメッセージであることがわかり、MAC の確認が行うことができ V に誤りが無いこと（完全性）も保証される．次のラウンドでは基地局は k^{j-2} を使って MAC を使い μ TESLA を利用する．この通信は予めスケジュールが決められており、もし決められていなければ通常ノードが MAC を受信した時刻以前に、μ TESLA の鍵が公開されていないことを保証できない．つまり外部攻撃者がリストと MAC を受信しそれを保持した状態で鍵を受信して、改変したリストと MAC、鍵を送る man in the middle 攻撃が可能になる．これを防ぐため、各ノードと基地局と時刻同期が不可欠であり、通信スケジュールを定めている．以上のようにハッシュ連鎖を利用して基地局からの通信であることを証明する．

SLEACH では MAC を用いることで不正なノードがクラスタヘッドになることを防ぎ、不正なノードがクラスタヘッドになりすましてデータを送信したことを発見することができる．しかし各ノードが持つのは固有に割り当てた鍵と μ TESLA 用の鍵であるため、クラスタヘッドとノードのリンク鍵が無い．そのため暗号通信は行えず機密性は無い．

2.3 SecLEACH

SecLEACH はランダム鍵配布方式を用いてノードとクラスタヘッド間の機密性を実現した方式である．クラスタヘッドにデータを転送する際に各ノードは共通鍵を用いてデータの暗号化を行う．クラスタヘッドがそれを受信後、データの復号を行いデータの結合を行う．そしてデータを再び暗号化し、基地局へ送る．SecLEACH protocol は SLEACH protocol を元に作られた形をしており、メッセージの完全性を保証するために MAC や μ TESLA を利用している．

SecLEACH で用いられているランダム鍵配布方式について説明する．ランダム鍵配布方式では、まず鍵の集合である鍵プールから鍵を各ノードに無作為に持たせる．鍵プールに入っている鍵には鍵 ID が付けられている．実際に通信を開始する際は、まずクラスタヘッドが自分の持つ鍵 ID を通常ノードに知らせる．通常ノードはその中から共通した鍵の ID をクラスタヘッドに知らせる．その後通常ノードはその共通鍵を用いて暗号化してデータを送信する．このように鍵自身を知らせることなく、相手と共通鍵を決めることができる．

SecLEACH は SLEACH protocol を元に、ランダム鍵配布方式を用いることでクラスタと通常ノード間の通信路暗号化を実現した．各ノードにランダムに鍵を割り当てることで、クラスタヘッドと共通鍵を確率的に持たせることができるため機密性を得ることができ、MAC を利用することでノードの正当性、データの完全性を保証できる．しかしクラスタヘッドと通常ノード間の鍵共有は確率的なものであり、鍵が共有できなかった場合、最も近いクラスタヘッドとリンクを生成できるとは限らない．そのため、遠くのノードと通信しなければならず消費エネルギーが増加してしまう．また、クラスタヘッドの数や持たせる鍵の個数によってすべてのクラスタヘッドと共通鍵が無いノードが出てくる．SecLEACH の論文ではこのノードを orphan と定義している．orphan の数はクラスタヘッドの個数、鍵プール

に含まれる鍵の個数、割り当てた鍵の個数に依存する．そして各ノードに割り当てる鍵数が多くなればクラスタヘッドが通常ノードに知らせる鍵 ID が多くなり送信電力が増える．さらに、無作為に割り当てた鍵からリンク鍵を構成するため攻撃者によってノードが解析された場合、一部のリンクの盗聴が可能となりうる．

3. 提案方式(Modified LEACH)

3.1 階層型鍵共有方式

階層型鍵共有方式について説明する．この方式は岩村[5]によって考案された鍵共有方式であり、階層構造を用いることで保持する鍵数の削減とどのエンティティ間でも共通鍵を得られる仕組みを実現している．まず階層には階層数 h、分岐数 i に対応した鍵配列 G_{h1}、G_{h2}、…、G_{hi} を準備する．G_{hi} には G_{h1}, G_{h2}, G_{h3}、…、G_{hi} が含まれ、これらを要素鍵と呼ぶ．h, i は全エンティティ数に応じて設定する．各エンティティは階層構造の終点に割り当てられ自分自身が対応している分木の鍵配列が格納される．従って各エンティティは階層数分の鍵配列を持つことになる．所持する要素鍵の個数は $h \times i$ となる．なお、鍵配列を構成する要素鍵には $g_{hij} = g_{hji}$ という特徴をもたせる．鍵共有する場合、ノード ID のみを交換する．すべてのノードは相手のノード ID から鍵配列中の要素鍵を特定できる．

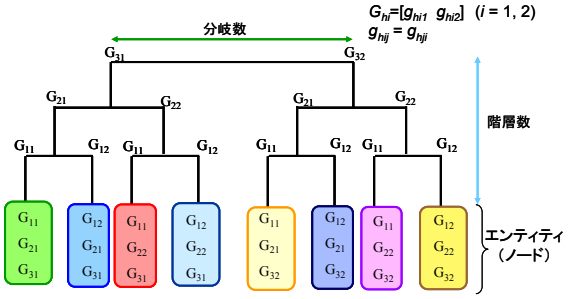


図 2 階層型鍵共有方式

図 2 に $h = 3, i = 2$ の時の階層型鍵共有方式の概要図を示す．このとき、$2^3 = 8$ 個のノードに対して鍵を割り当てることになる．まず全ノードの半分にそれぞれ G_{31}, G_{32} の鍵配列を割り当てる．次にそれらを割り当てたグループそれぞれを半分に分け, G_{21}, G_{22} を各々割り当てる．次の階層にも同様の過程を踏む．この時各エンティティが所持する鍵配列は 3 個であり、要素鍵の数は 6 個である．

この階層型鍵共有方式で得られる要素鍵を k_1, k_2, k_3…,k_n とすると、それらを並べて結合したものを入力値としてハッシュ関数に通す．得られた数列を鍵として利用する．ゆえにリンク間における共通鍵 K は

$$K = k_1 \oplus k_2 \oplus k_3 \oplus \ldots \oplus k_n$$

となる．この階層下鍵共有方式によって得られる共通鍵を階層鍵と定義する．

3.2 鍵の個別化

3.1 の階層型鍵共有方式により各ノードはノード ID を交換するだけで、相手との共通鍵を生成することができる．しかしノードが乗っ取られた場合、要素鍵漏洩問題が発生する．あるリンクに使われる要素鍵は他のノードも持っており、数個のノードを乗っ取られるだけでそのリンクに使われる要素鍵が完全に漏洩する．そこで各リンクに乱数を割り振り、その乱数を階層鍵と排他的論理和をとったものを鍵として使うことで、たとえノードの盗難により要素鍵が漏洩しても他ノード間のリンクで使われる階層鍵がわからないような仕組みを実現する．鍵の個別化は 3.1 の階層の下に更に階層を設ける形となる．鍵の個別化の概要図を図 3 に示す．

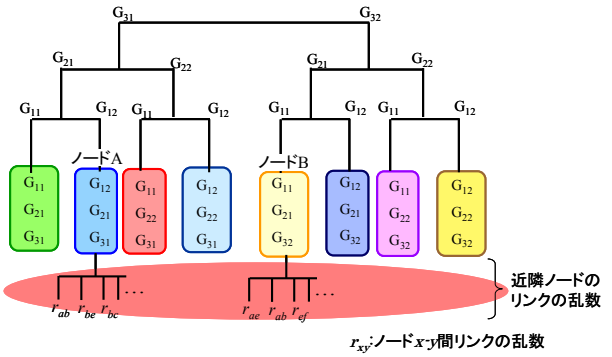


図 3 鍵の個別化

ここで、ノード A と B ついて階層型鍵共有方式によって階層鍵を k_{AB} を共有したとする．その後、A が乱数 r_B を生成し暗号化して B に伝え、B は乱数 r_A を生成し暗号化して A に伝える．この時暗号鍵は $k_{AB} \oplus \sum r_{AB}$ である $\left(r_{AB} = r_A + r_B\right)$. $\sum r_{AB}$ は過去に A,B 間で通信して共有した乱数和を意味する．これを setup phase が行われる毎に繰り返すことで、鍵を更新している．階層鍵も要素鍵の和によって計算されるため、個別化した階層の乱数を含むすべての階層の要素鍵を加算するという簡単な処理によって、更新階層鍵は実現される．

階層鍵は他のノードを乗っ取り要素鍵を得れば知ることができるが、乱数和はその通信までの全ての通信を知っていなければわからない．従って他のノードが乗っ取られた場合でもリンク鍵が漏洩することはないのである．

一度リンク間で乱数を割り当てたらその乱数は記憶する．全てのノードがクラスタヘッドになるのは $1/P$ ラウンド（P:全ノードにおけるクラスタヘッドの割合）である．従って平均して $1/P$ 個のノードとなる．故に極端に小さい P では記録する乱数の数が増大する可能性がある．しかし LEACH ではエネルギー消費を最小にするクラスタヘッドの個数が存在する．クラスタヘッドを最適な値にすることで、保持する乱数の個数も小さくなり問題にならない．例

えば P=0.2 のとき、1/P=5 となり各ノードが保持する乱数は平均して 5 個となる．

3.3 提案方式の LEACH プロトコル

階層型鍵共有方式と鍵の個別化を利用した提案方式の LEACH protocol について説明する．まず各ノードに階層型鍵共有方式を用いて鍵を割り当てる．また各ノードの固有鍵、μ TESLA に用いるハッシュ連鎖の最後の鍵を持たせる．従って各ノードは 3 種類の鍵を持つ．基地局は割り当てた鍵とノードは全て知っており、かつハッシュ関数を通したすべての連鎖鍵 k^0, k^1, k^2....., k^j を知っているものとする．

ノードを配置後、準備通信として各ノードは ID 順にメッセージを周辺のノードが受信できる信号の強さでブロードキャストする．近隣ノードはそれを受信した後乱数を生成して送信することで、ブロードキャストしたノードとのリンクにおける乱数を共有することができる．この時ノードは安全に管理されており盗聴されないものとする．一度乱数を共有したノードに関してはメッセージを受信後、乱数を生成・送信はしない

全ノードがメッセージをブロードキャストし終わったら、setup phase に移る．図 4 に提案方式の LEACH protocol を示す．通常の LEACH と同様に各ノードがクラスタヘッドになるかどうかを判定する．クラスタヘッドになったノードはメッセージ adv とともに自分のノード ID id_H, ノンス nonce、MAC を送信する．SecLEACH とは異なり adv には鍵 ID は含まれておらず、単に識別用メッセージとなっている．そのメッセージを通常ノードが受信するとクラスタヘッド ID id_H,を一時保存する．一方基地局は MAC の検証を行い、正当なクラスタヘッドのリストを作成する．全クラスタヘッドがメッセージをブロードキャストすると、基地局が μ TESLA を用いて正当なクラスタヘッドのリストを通常ノードに伝える．そのリストを使って通常ノードは正当な、最近隣のクラスタヘッドを選択する．そしてクラスタヘッド ID と過去の乱数和からリンク鍵 $link\_k_{n-1}$ を生成し乱数を暗号化する．また新たな乱数も加えた新しいリンク鍵 $link\_k_n$ を用いて MAC を生成しクラスタヘッドへ送信する. 全てのクラスタメンバからメッセージを受け取ると、暗号化された乱数を $link\_k_{n-1}$ で復号し、新たな乱数を加えたリンク鍵 $link\_k_n$ を作り、MAC の検証を行う．全てのクラスタメンバについて乱数の割り当てと MAC の検証が終了すると、TDMA スケジュールを作りクラスタメンバにブロードキャストする．

そして steady-state phase になると通常ノードは観測データの転送を開始する．この時データは $link\_k_n$ で暗号化し、MAC もこの鍵を使って生成する．なお、提案 LEACH でも steady-state phase を複数回繰り返して運用することを想定しており MAC の入力値に、現在の繰り返し回数 l だけ足したノンス $nonce + l$ を用いる．クラスタヘッドは受信後、MAC の検証と共にデータを復号し結合・圧縮を行う．そして結合したデータを固有鍵で暗号化、基地局へ転送する．このフェイズを終えると steady-state phase または setup phase に移行する．

Setup phase

1.1	$H \Rightarrow G$	:id_H,$nonce$, MAC$k_H(id_H\|c_H\|adv)$, adv
	A_i	:store(id_H)
	BS	: if MAC$k_H(id_H\|c_H\|adv)$ is valid, add(id_H, V)
1.2	$BS \Rightarrow G$	:V, MAC$k^{j-1}(V)$
1.3	$BS \Rightarrow G$	:k^{j-1}
	A_i	:if ($f(k^{j-1}) = k^j$) and ($id_H \in V$) H is authentic.
2.	$A_i \Rightarrow H$	idA_i, id_H, $rand$, $join\_req$, MAC$link_k_n(idA_i\|id_H\|rand\|nonce)$
3.	$H \Rightarrow G$	:id_H, (..., <id_{Ai}, t_{Ai}>,...), $sched$

Stead-state phase

4.	$A_i \rightarrow H$	:id_{Ai}, id_H, d_{Ai}, MAC$link_k_n(idA_i\|id_H\|dA_i\|nonce + l)$
5.	$H \rightarrow BS$	:id_H, id_{BS}, $F(..., d_{Ai}, ...)$, MAC$k_H(F(..., dA_i, ...)\|c_H)$

記号の意味

k_X	:	X の固有鍵
$nonce$	:	ノンス
c_X	:	基地局とノード X と共有するカウンタ
MAC$_k$(msg)	:	鍵 k を使って生成した MAC
store(id_H)	:	後に行う認証のため id_H を記録
add(id_H, V)	:	Vに id_H を追加
k^j	:	j番目のハッシュ連鎖の鍵
$f()$	:	一方向性関数
V	:	正当な id_H リスト
$link\_k_n$	:	n ラウンド目のリンク鍵(階層鍵⊕過去の乱数和)
$rand$	:	個別化のための乱数を $link\_k_n$ で暗号化したもの(MAC の中の $rand$ は暗号化していなくても良い)
l	:	stead-state phase の反復回数
adv, $join\_req$, $sched$	:	メッセージ識別用コード

図 4　提案 LEACH(Modified LEACH) protocol

3.4 SecLEACH との比較

提案方式の LEACH で利用した階層型鍵共有方式によって SecLEACH の問題点を解決することができる．SecLEACH ではノードに対してランダムに鍵を割り当てるため、鍵共有は確率的なものになる．そのためデータ送信先クラスタヘッドに制約が生じ、遠方クラスタヘッドへデータを転送しざるを得ないノードがでてきてしまう．最悪の場合、どのクラスタヘッドとも鍵共有が行うことができず孤立する orphan が生じることがある．orphan が生じる場合は、そのノードをそのラウンドだけ沈黙させたり、基地局に直接通信させたり、付加的にプロトコルを入れたりすることで解決しなければならない．一方階層型鍵共有方式では必ず鍵共有をすることができる．よって LEACH と同じように最近隣ノードと通信することができる．遠方クラスタヘッドに通信することが無いので、電力消費を小さくすることが可能である．また orphan となるノードも生じないため、鍵共有不可時の対策をする必要がなくなり、ノード全てを必ず正常動作させることができる．

また提案方式では、鍵共有する場合に必要な情報は相手のノード ID のみで済む．SecLEACH ではクラスタヘッドが持つ鍵 ID 全てを、通常ノードにブロードキャストしな

ければならない．どの鍵 ID と共通鍵であるかどうか通常ノードに確認させる必要がある．それゆえ割り当てる鍵 ID が大きくなると消費電力も増大する．しかし提案方式では、ノード ID さえ分かれば各ノードは対称マトリクスを用いて共通鍵を得ることができる．そのためメッセージ長を短くすることができ、結果としてエネルギー消費を小さくすることができる．

そしてノードが乗っ取られた場合、鍵漏洩が生じたとしても各ノード間に個別の乱数が鍵として足し込まれているので、他ノード間のリンク鍵は漏洩しない．SecLEACH ではノードが盗難にあって鍵が漏洩した場合、他のリンクで使用される鍵が確率的に含まれる．十分なノード数が盗難にあえば全てのリンクの暗号鍵が漏洩することになる．この場合暗号通信は意味をなさなくなる．一方提案方式では共通鍵に更に乱数の排他的論理和をした鍵を利用する．このことにより、他のノードが乗っ取られ鍵が漏洩したとしてもリンク鍵は漏洩しない．

4. 安全分析

4.1 想定する攻撃者

提案方式が想定する攻撃者はネットワーク内を流れる情報の盗聴や改竄を目的、すなわち、機密性と完全性の破壊を目的としジャミングなど大規模電波設備を利用した攻撃、すなわち可用性に関する攻撃は行わないとする．特に、攻撃者が正当なノードを盗難・解析してネットワーク内を流れる情報の盗聴や改竄を行う場合を含む．ただし、ノード投入時には不正ノードは存在せず攻撃者は部分的な通信の傍受は可能であるが､全ネットワークの傍受は困難とする．

4.2 機密性

リンク間の機密性について分析する．ノード-クラスタヘッド間は鍵の個別化を含む階層型鍵共有方式によるリンク鍵を用いてデータを暗号化する．暗号化方式は安全であると仮定すると、ネットワークを流れる情報の機密性は、リンク鍵に依存する．リンク鍵は階層鍵と過去の乱数和の排他的論理和演算によって作られる．階層鍵はノードの盗難・要素鍵の解析によって漏えいする可能性があるが、乱数和はそのノード間の通信を最初から傍受していなければ解析不能である．前述のように、攻撃者は全ネットワークの傍受は困難としているので、攻撃者が意図したノード間の鍵しか漏えいしない．よって、一部のノードの盗難・解析によって全ネットワークの情報が漏えいすることはない．

4.3 完全性

基地局を含む各ノードからのメッセージには MAC がつけられており、送信者が正当とするならばメッセージ認証が可能である．よって、メッセージの完全性は MAC が安全であれば保証される．

4.4 メッセージ送信者の正当性

メッセージ送信者の正当性は固有鍵を用いた MAC によって保証される．まずノードがクラスタヘッドとなった際にブロードキャストするメッセージに MAC が用いられる．入力値はクラスタヘッド ID と adv とカウンタである．また鍵として固有鍵を使う．固有鍵は秘密情報であり、ノードを解析する以外に知る方法はない．従って正当な MAC を作れるのは正当なノード（クラスタヘッド）のみである．そして基地局が MAC の検証を行う．ID は送られてきたメッセージで、adv は定形である．またカウンタは全ノードと基地局間で同期させているため MAC の入力値を全て基地局は知ることができる．また基地局はすべてのノードに割り当てた固有鍵を知っているため同じ MAC を生成でき、検証ができる．ゆえにクラスタヘッドになると宣言したノードの正当性を検証することができ、正当なクラスタヘッドリスト V に加えるか否かの判断できる．

次に各ノードからクラスタヘッドにクラスタメンバになる段階で送信されるメッセージ送信者の正当性に関して、各ノードは自分とクラスタヘッドの ID,新たに知らせた乱数、ノンスを入力値とし、鍵を階層型鍵共有方式によるリンク鍵で MAC を生成する．先述と同様に、MAC の入力鍵がそのノードとクラスタヘッド間のみの秘密情報であるため、不正な攻撃者が正しい MAC を作成できない．受信したクラスタヘッドは秘密情報を知っており、入力値のいずれも送信メッセージから知ることができる．ノンスは最初にクラスタヘッドが送信しているため既知である．よってクラスタヘッドは MAC を検証でき、不正なノードを排除可能となる．

steady-state phase におけるノード-クラスタヘッド間及びクラスタヘッド-基地局間でも同様に検証することができ、不正なメッセージを除外することができる．

ところで、TDMA スケジュールを送信するクラスタヘッドは MAC を送信しておらず正当性が保証されない．攻撃者がクラスタヘッドと偽り不正なスケジュールを送信し、ネットワークを妨害する攻撃が可能である．しかしこの攻撃は非常に高度な操作が必要であり、同様の目的を達するならば容易なジャミングを利用すると考えられるため、MAC を利用していない．

4.5 再送攻撃耐性

再送攻撃の耐性について分析する．再送攻撃による対策としてカウンタとノンスがある．カウンタは各ノードが持っており、おのおのと同期しているカウンタを基地局が保有している．そのため、ノードがカウンタ値を送信する必要はない．カウンタはノードがクラスタヘッドになったときに、MAC の入力値として利用する．またクラスタヘッドが基地局にメッセージを送信するたびに、カウンタの値が＋１させる．カウンタそのものを送信することはなく、ノードの内部情報であるためノードを乗っ取る以外で攻撃者が正しいカウンタの値を知ることができない．そのため正しい MAC を生成することができず、再送攻撃を防ぐこ

とができる．

一方ノンスはワンタイムトークンと呼ばれるものであり、この通信を使った通信は 1 度しか行わない．ノンスはクラスタヘッドになった際にブロードキャストするときに送信される．そして通常ノードがクラスタメンバになることを伝えるときに MAC の入力値としてノンスを使う．ノンスを入力値とした MAC はこのラウンドのみ、かつ 1 だけ有効である．そのため攻撃者が不正にメッセージを傍受し、以降のラウンドでメッセージをそのまま送信したとしても、無効になりネットワークの混乱を引き起こすことができない．また steady-state phase においてもノンスを用いる．たとえ steady-state phase が複数回繰り返された場合でも、ノンスの値に加算したものを MAC の入力値として用いることで、同様の理由により再送攻撃を防ぐことができる．

5. 動作分析：LEACH シミュレーション

提案 LEACH と SecLEACH, LEACH の消費エネルギー比較を行うために MATLAB を用いてシミュレーションを行った．setup phase, steady-state phase を一度ずつ行った 1 ラウンドにおける全ノードの消費エネルギー合計を、クラスタヘッドを変えて計算する．このシミュレーションでは LEACH の論文に記載されているモデルを用いた．

シミュレーションの流れとして、あるクラスタヘッド数において 10 回シミュレーションして平均をとった結果をそのクラスタヘッド数における平均消費エネルギー合計とする．なお、消費エネルギーは(3)のように通信距離とメッセージによるものだけであり、暗号化や復号、MAC 生成に必要とされる消費エネルギー、データ結合による消費エネルギー、待機電力などその他の要因による消費エネルギーは含まれない．

5.1 シミュレーション条件

シミュレーションを実行するにあたって、各通信のメッセージ長を適宜定めた．ノードは 100 個とし、10^4[m^2]の領域に無作為に配置したと想定する．基地局はこの領域の中心に位置する．ノードの配置図を図 5 に示す．

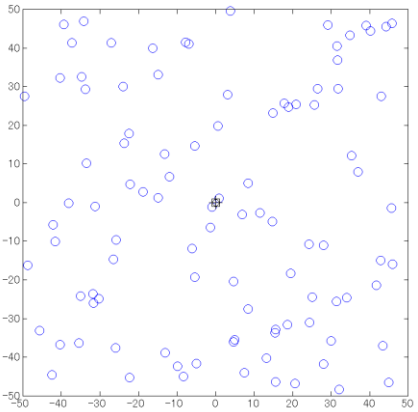


図 5 ノード配置図

また SecLEACH では、どのクラスタヘッドとも通信できない orphan が存在することがある．このノードに関しては、そのラウンドでは全く動作せず通信しないとした．SecLEACH は鍵プールに含まれる個数 S と各ノードに割り当てる鍵数 m に依存して特性が異なる．ここでは(S, m) = (50, 5000), (100, 10000), (150, 15000)の 3 種類に関してシミュレーションを行った．

5.2 シミュレーション結果

シミュレーション結果を図 6 に示す．(a)は setup phase の結果を、(b)は steady-state phase のエネルギー消費合計を示す．これらを足したものが 1 ラウンドにおける消費エネルギーの合計となり、(c) のようになる．また(d)に SecLEACH の orphan 数についての特性グラフを示す．setup phase では SecLEACH の消費エネルギーが提案 LEACH のそれと比べ非常に大きくなっている．SecLEACH ではクラスタヘッドの持つ鍵 ID をブロードキャストしなければならない．また鍵 ID が増えるほどメッセージ長は長くなり、クラスタヘッドの数が増えるほど全体として消費エネルギーは大きくなる．一方、提案 LEACH では鍵 ID は知らせないためメッセージ長がその分短い．そのため消費エネルギーが抑えられている．そして LEACH と比較すると提案 LEACH は大きな消費エネルギーとなっている．これは LEACH にはないメッセージや通信があるため、それだけ電力消費が発生しているためである．また、クラスタヘッドが 1, 2 個といった非常に少ない領域では SecLEACH の消費エネルギーが、LEACH のそれよりも下回る結果となっている．クラスタヘッドが少ない場合 orphan が存在する．orphan は動作を停止する振る舞いをさせているので、orphan の多い領域すなわちクラスタヘッドの少ない領域では、SecLEACH の消費エネルギー合計が小さくなったと考えられる．

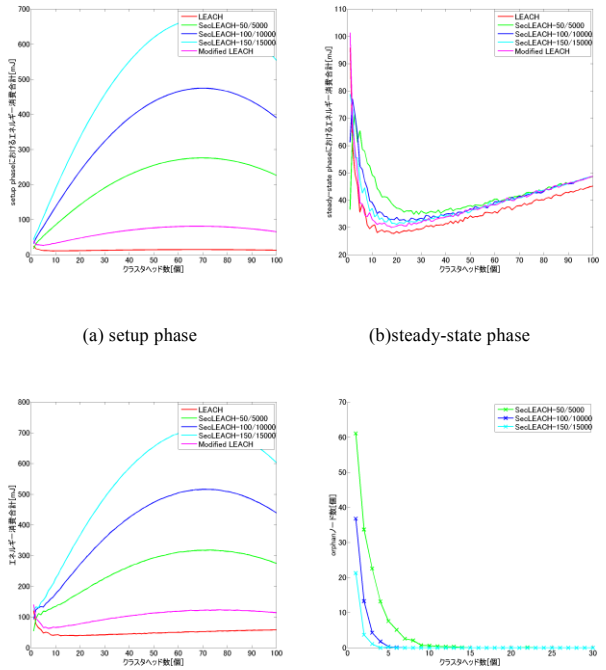
steady-state phase における結果では、提案 LEACH が SecLEACH の消費エネルギーを下回る最小値である．しかし SecLEACH の、鍵プール 15000 個から各ノードに 150 個割り当てたもののグラフと比較するとあまり差が無いようにみえる．SecLEACH ではランダムに鍵を割り当てているため、最近隣クラスタヘッドと鍵共有できるかは確率的なものとなる．その確率を高確率にするには少ない鍵プールの数で多くの鍵を各ノードに割り当てればよいが、ノードが盗まれ場合リンク鍵として使われている鍵が漏洩する可能性が上昇する．つまりリンク鍵の漏洩確率と消費エネルギーはトレードオフの関係にある．提案方式ではリンク鍵が漏洩せず、かつ最近隣クラスタヘッドと通信できることを実現している．また steady-state phase を複数回繰り返すこと場合、消費エネルギー量の差は大きくなる．ここで steady-state phase において、あるクラスタヘッド数の提案 LEACH と SecLEACH それぞれの消費エネルギー量を W_{mod}, W_{sec} [mJ]とし、その差を ΔW とすると

$$\Delta W = W_{sec} - W_{mod}$$

となる．そして複数回 steady-state phase を繰り返すことを考える．繰り返し回数を n 回とおくと

$$nW_{sec} - nW_{mod} = n(W_{sec} - W_{mod}) = n\Delta W$$

となる．つまり 1 回の steady-state phase で生じる差の n 倍が消費エネルギー差となることがわかる．ゆえに繰り返し回数が多くなると比例的に消費エネルギー差も大きくなる．



(a) setup phase　　(b)steady-state phase

(c)1 ラウンド　　(d)orphan 特性

図 6　シミュレーション結果

setup phase と steady-state phase を合わせた 1 ラウンドの消費エネルギーの特性を見ると、提案提案 LEACH の消費エネルギーが SecLEACH のそれを下回っている．従って提案 LEACH は SecLEACH より消費電力の小さいプロトコルであると言える．また 1 ラウンドにおけるエネルギー消費の特性が setup phase のそれに似ている．これは setup phase の方が steady-state phase の消費エネルギーが大きいため、加算された結果が前者の特性と形が似ている．steady-state phase の繰り返し回数が多くなれば setup phase と消費エネルギー量が逆転し、十分大きな回数となると 1 ラウンドにおける消費エネルギーの特性グラフは、steady-state phase の形に近いものとなる．

6. まとめ

センサネットワークプロトコルである LEACH プロトコルにセキュリティを加えた新しい LEACH プロトコルを提案した．既存研究のプロトコルを引用し、各ノードと必ず鍵共有ができる階層型鍵共有方式とリンク鍵に乱数を加算する鍵の個別化によって外部攻撃者に対してセキュリティを付加することを実現した．このことから、目標である LEACH にセキュリティを付加することができたと言える．また各ノードと必ず鍵共有ができることによってノードの通信先クラスタヘッドに制限を加えることがなく、メッセージ長を短くすることで低消費電力化が可能になった．このことに関して実際シミュレーションを行い、評価を行った．

今後の課題として、ラウンド方向に関しての評価シミュレーションと内部攻撃耐性がある．LEACH はネットワーク寿命を伸ばす特徴がある．本方式がラウンド数が増えるに連れてどのような特性となるか、またノードがエネルギーを完全消費しネットワークメンバが減少した場合でも安全性に問題ないかを検証する必要がある．そして本提案方式は決してネットワークメンバに悪意の持つノードが存在しないことを前提にしていたため、攻撃者がノードを乗っ取り悪用し始めた場合の耐性が皆無である．ネットワークメンバに悪意を持つノードが存在しても安全性を維持できる仕組みが必要と考えられる．

謝辞　本研究を行うにあたりご指導いただきました岩村惠市教授、並びに柿崎淑郎助教、稲村勝樹様に心より感謝いたします。また、研究を進めるにあたり多くのアドバイスを頂きました研究室の皆様に心より感謝いたします。

参考文献

1)　Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. "Energy-Efficient Communication Protocol forWireless Microsensor Networks". In IEEE Hawaii International Conference on System Sciences, January 4-7, 2000.
2)　Adrian Carlos Ferreira, Marcos Aurelio Vilaca, Leonardo B. Oliveira, Eduardo Habib, Hao Chi Wong, Antonio Alfredo Ferreira Loureiro. “On the Security of Cluster-Based Communication Protocols for Wireless Sensor Networks”.　Networking - ICN 2005, Lecture Notes in Computer Science Volume 3420, pp.449-458, 2005
3)　Leonardo B. Oliveira, Adrian Ferreira, Marco A. Vilaca, Hao Chi Wong, Marshall Bern, Ricardo Dahab, Antonio A.F. Loureiro. "SecLEACH - On the security of clustered sensor networks".　Signal Processing Volume 87, Issue 12, pp.2882-2895, 2007.
4)　Perrig, A., Szewczyk, R., Wen, V., Culler, D., Tygar, J.D."SPINS: Security protocols for sensor networks". Wireless Networks Volume 8, Issue8, pp.521 - 534, 2002
5)　特許番号 特許第 3548215 号, 登録日平成 16 年 4 月 23 日

●正誤表

誤：岩村惠一

正：岩村惠市

以上