情報処理学会論文誌 Vol.57 No.10 2236–2249 (Oct. 2016)

クラスタツリー型センサネットワークに適した秘密分散法とそれを用いた鍵共有方式

岩村 恵市[1,a)] 須賀 祐治[2] 後藤 慎一[1] 金田 北洋[3]

受付日 2015年12月11日, 採録日 2016年7月5日

概要：センサネットワークの構成の 1 つに複数のノードを集めてクラスタ化を行い，ネットワークを階層的に構成するクラスタツリー型がある．このネットワーク構成で暗号通信を行う場合，既存の暗号鍵共有法では，クラスタ内のノードを管理し基地局と通信を行うクラスタヘッド（以下，CH）を解析すればクラスタ内の全ノードの鍵が漏えいする問題がある．さらに，全ノードが CH になる可能性がある場合に，クラスタ内の全ノードの鍵を保有する場合に備えて全ノードに比較的大きな記憶容量を持たせなければならないという問題が生じる．そこで本論文では秘密分散法を用いて，まずクラスタツリー型センサネットワークにおいて CH がクラスタ内の全子ノードと個別に暗号鍵を共有でき，かつ暗号通信時以外のときに CH を解析されても暗号鍵がまったく漏えいしない情報理論的安全性を持つ鍵共有方式を提案する．さらに，CH がクラスタ内の各ノードとの鍵を保存するための記憶容量を必要とせず自分の鍵のみの管理で済む鍵共有方式を提案する．そのために，秘密分散法を改良し，その安全性を評価する．最後に，暗号通信時以外で CH またはすべての子ノードが解析されてもまったく鍵が漏洩しない鍵共有法を示す．

キーワード：センサネットワーク，鍵共有方式，秘密分散，情報理論的安全性，記憶容量削減

Secret Sharing Scheme and Key Sharing Scheme Suitable for Clustered Sensor Networks

Keiichi Iwamura[1,a)] Yuji Suga[2] Shinichi Goto[1] Kitahiro Kaneda[3]

Received: December 11, 2015, Accepted: July 5, 2016

Abstract: Wireless sensor networks include clustered sensor network, which collects two or more nodes and makes a cluster. This network has a problem in which encryption keys on all the nodes in a cluster reveals, if the cluster head (CH) which manages all nodes in a cluster and communicates with a base station is analyzed. On the other hand, when all the nodes may be set to CH, the problem that comparatively big storage capacity must be given to all the nodes arises to hold the key of all the nodes in a cluster. In this paper, we propose the first key sharing scheme which carries out key sharing with CH and all the nodes in a cluster, and realize the information theoretical security using secret sharing scheme, even if CHs are analyzed. In addition, we propose the second key sharing scheme in which storage capacity for CH to save the key on each node in a cluster is not needed. In order to realize it, secret sharing scheme is improved and the security is evaluated. In the last, we show the third key sharing scheme which reveals no key, even if CH or all nodes except CH are analyzed.

Keywords: Sensor network, Key sharing, Secret sharing scheme, Information theoretical security, Storage capacity reduction.

[1] 東京理科大学
Tokyo University of Science, Katsushika, Tokyo 125–0051, Japan
[2] 株式会社インターネットイニシアティブ
Internet Initiative Japan Inc., Chiyoda, Tokyo 102–0071, Japan
[3] 大阪府立大学
Osaka Prefecture University, Sakai, Osaka 599–8531, Japan
[a)] iwamura@ee.kagu.tus.ac.jp

1. はじめに

無線で接続できる端末のみで構成されるアドホックネットワークの一種として，センサネットワークがある．センサネットワークは一般にセンサノードと基地局（Base Station：以降，BS）で構成されていて，その目的は種々

の情報の収集である．手順としては，まずセンサノードが温度や電力，湿度などの様々な情報を観測・収集し，それを直接，もしくは他のノードを用いたマルチホップ通信を行うことで BS へデータを送信する．用途としては，軍事目的から一般消費者をターゲットとした民生用まで多岐にわたり，新たな通信インフラとして期待されている．

センサノードは以下のような特徴を持つ．センサノードはコスト削減の必要性から簡易な構造をしている．よって耐タンパ性に乏しく，高性能な CPU も使われていない．よって，公開鍵暗号方式のような高度な計算処理を行うことに適していない．加えて，メモリも小さいものであるため，記憶する情報を少なくする必要がある．さらに，電池の利用などを想定するため，効率的なエネルギーの消費が求められる．また，センサノードは野外に放置されて情報収集することも多い．このことにより攻撃者が放置されたノードを盗んで解析することで設定された秘密鍵などの情報を容易に得ることができる．それに対して，BS は一般に十分な電力や計算資源を持ち，安全に管理されるとする．

一方，センサノードのエネルギー消費を効率的にするために，いろいろな手法が研究されてきた．まず，各ノードが BS と直接通信を行うユニキャストの場合においては，遠くのノードが電力を大きく消費してしまうため，他のノードよりも早く電池切れになる問題がある．それに対して，ノードが観測したデータを BS により近いノードに送信しそれを転送していく手法がある．送信電力は通信相手との距離が短いほど小さくなるため，近くのノードへの通信は消費エネルギーを小さくできる．しかしこの場合，BS に近いノードほど転送するデータが集中し，通信量が増えるため，BS に近いノードが先に電池切れになってしまうという問題がある．一方，データを転送する接続トポロジとして，複数のノードを集めて，接続を階層化するクラスタツリー型トポロジと呼ばれる手法が提案されている．この手法はノードをいくつかのグループ（クラスタ）にわけて，そのクラスタ内でノードのリーダ（クラスタヘッド，以降 CH）を決定する．各クラスタ内のノードはデータを近くの CH に転送し，CH はそのデータを BS へ転送する．しかしこの方法も CH となったノードは転送量が増えるため他のノードより早く電池切れになるという問題を持つ．そこで直接 BS と通信を行う CH を順番に変えていく手法，LEACH [1], [2] が考案された．

LEACH（Low-Energy-Adaptive-Clustering-Hierarchy）は 1 つのノードが CH となっている期間をラウンドと呼び，各ノードが順番に CH になることを宣言し，BS の命令なしに自律的にクラスタを編成する．最大の特徴は一定の周期ごとにすべてのノードが CH になるようにする点である．これにより全ノードのエネルギー消費の偏りを平均化し，ネットワークの寿命を長くすることができる．

しかしこのプロトコルの原案 [1] はセキュリティを考慮していない．そこで，LEACH に対するセキュリティ対策に関する研究として，暗号鍵を共有して共通鍵暗号による通信を行う SecLEACH [3] や MS-LEACH [4] が発案された．SecLEACH ではランダム鍵配布方式 [5] を適用して鍵共有を実現する．しかし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ためエネルギー消費が大きなものとなるという点で問題がある．また，ノードが盗難・解析された場合に他のリンク間の鍵も漏洩する危険性がある．また，MS-LEACH では LEAP [6] と呼ばれる方式を用いて暗号鍵共有を行う．しかしこの方式では，もし初期鍵が解析できた場合に，公開情報であるノード ID から全暗号鍵を生成できるため全ノードの鍵が漏洩するという欠点をかかえている．

さらに，LEACH をはじめとするクラスタツリー型センサネットワークにおける暗号鍵共有方式に共通する問題としては，一般に CH はクラスタ内の全ノードと暗号通信を行うため全ノードの鍵を保持しており，CH を解析すればクラスタ内の全ノードの鍵が漏えいするということがある．また，CH となるノードはクラスタ内の全ノードの鍵を保有する必要があるため，比較的大きな記憶容量が必要である．よって，LEACH のように全ノードが CH になる可能性がある場合，全ノードに比較的大きな記憶容量を持たせなければならないという問題も発生する．

一方，Shamir の (k, n) 秘密分散法 [7] を用いて暗号通信用の鍵共有を行う方式 [8], [9] が提案されている．これらの方式では，BS-CH 間の暗号鍵と CH-子ノード間の暗号鍵を得るための情報を各々秘密情報として n 個のノードに分散する．ただし鍵を復元する際，各ノードに保存されている分散情報をそのまま通信路に出すと安全性が保てないため，公開鍵暗号的な手法を用いて分散情報の秘匿通信を行っている．すなわち，この論文はクラスタツリー型センサネットワークに秘密分散を用いるという点においては新しいが，公開鍵暗号的な手法を用いていかに分散情報を安全に共有するかが中心となっており，秘密分散法が持つ情報理論的な安全性や処理が軽いという効率性をまったく生かしていない．さらに，この手法は閾値未満の数のノードが解析されても暗号通信用の鍵は漏洩しないが，すべての子ノードが解析されると鍵が漏えいする．また，この手法はメモリを削減するために BS-CH 間および CH-子ノード間の鍵は各々共通の鍵を設定している．そのため，鍵を復元した CH を解析できれば，そのクラスタ内の通信はすべて盗聴可能である．すなわち，この手法は他の手法と同様，CH を解析すればそのクラスタ内で用いられる鍵が漏洩するという問題を解決していない．

そこで本論文では秘密分散法を用いて，クラスタツリー型センサネットワークにおいて CH と子ノードが必ず個別に鍵共有でき，かつ以下を個々に実現する方式を提案する．(1) 暗号通信時以外に CH が解析されても鍵が漏えいしな

い情報理論的安全性を持つ鍵共有方式を提案する．特に，CH を含むどれだけのノードが解析されてもそれ以外の子ノードと CH 間の鍵に関する情報がまったく漏洩しないという従来にない利点を持つ．ただし，子ノードが解析されるとその子ノードの鍵は漏洩する．

(2) CH はクラスタ内の各子ノードの鍵または分散情報を保存する必要がなく，自分の鍵のみを管理すればよい鍵共有方式を提案する．これによって，全ノードは CH になった場合に備えた比較的大きな記憶容量の準備を不要にする．ただしこの方式において，通信が盗聴された場合の安全性が情報理論的安全性から計算量的安全性に落ちる場合がある．また，この方式も子ノードが解析されるとその子ノードの鍵は漏洩する．

(3) 暗号通信時以外にクラスタ内の CH またはすべての子ノードが解析されても鍵が漏洩しない鍵共有法を提案する．この手法は情報理論的な安全性と計算量的安全性のどちらも選択することができる．

情報理論的安全性とは，攻撃者が無限の計算能力を持っていても鍵を解析できない安全性を指し，計算量的安全性とは攻撃者が多項式時間の計算量では鍵を解析できない安全性を指す．以上より，必要に応じて最適な手法を選ぶことができる．すなわち，CH 解析に対する安全性を重視し，クラスタ内の鍵を保有する記憶容量を許容する場合は (1) の方式を選択でき，計算量的安全性になる場合を許容しても，記憶容量を削減したい場合は，(2) の方式を選択すればよい．さらに，高い安全性を求めて CH または全子ノードが解析されても鍵がまったく漏洩しないようにしたい場合，(3) で情報理論的安全性を実現する方式を選択すればよい．特に，(2) の方式は今後の発展が見込まれる IoT デバイスなどで鍵共有を実現する際重要である．

ただし，(2) の方式は従来の秘密分散法では実現できないため改良を行い，高速処理が可能で分散情報を保存する必要がないという新たな特徴を持つ秘密分散法を提案し，それを用いる．また，その安全性などを詳細に評価する．

本論文の構成は，まず 2 章において今回対象とするクラスタツリー型センサネットワークである LEACH プロトコルの詳細について述べ，関連研究として LEACH に対してランダム鍵配布方式を適用した SecLEACH と LEAP を適用した MS-LEACH と秘密分散法を用いて鍵共有を行う手法についてその概要を説明する．次に，3 章において提案方式 1 を説明する．これによって上記 (1) を実現できることを示す．そして 4 章において新たな秘密分散法を示し，上記 (2) を実現する提案方式 2 を説明する．さらに，5 章において (3) を実現する手法を示す．最後に従来方式と各提案方式に対する効率や拡張性に関する評価を行う．

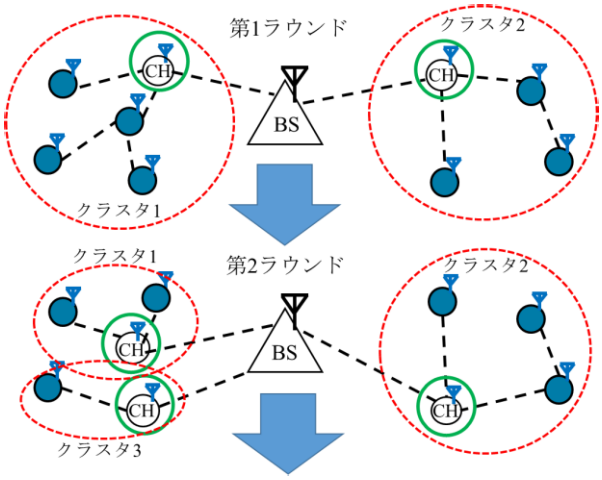

図 1 LEACH の概要
Fig. 1 LEACH.

2. 従来研究

2.1 LEACH

LEACH（Low Energy Adaptive Clustering Hierarchy）とはクラスタとその CH を周期的に変え，ノードが消費するエネルギーを平均化し，システム全体の寿命を伸ばす方式である．**図 1** は第 1 ラウンドにおいて 2 つのクラスタから構成されたネットワークが，第 2 ラウンドにおいて CH が変わることによって，クラスタ構成も変わり 3 つのクラスタから構成される様子を示す（青丸は子ノードを表す）．

LEACH は setup phase と steady-state phase の 2 つのフェーズを繰り返す．setup phase では，各ノードが乱数を用いて CH になるノードをランダムに決定する．CH になったノードは自分が CH であることを伝えるメッセージを各ノードにブロードキャストする．それを受信した各ノードは，信号の強さから最も近い CH を選択し，その子ノードになることを CH へ伝達する．この過程の終了後，CH は子ノードに対して後の steady-state phase における通信のための TDMA スケジュール [10] を子ノードに送信する．steady-state phase は割り当てた TDMA スケジュールに基づいて各子ノードが観測データを自分が所属する CH に送信するフェーズである．CH は複数のメンバから受信したデータを結合することで送信情報を圧縮する．CH は結合したデータを BS へ送信する．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．

2.2 SecLEACH

SecLEACH [3] では LEACH の setup phase において CH が定まった後，ランダム鍵配布方式を用いて CH-子ノード間の暗号鍵共有を行い，steady-state phase における通信の秘匿化を実現したものである．

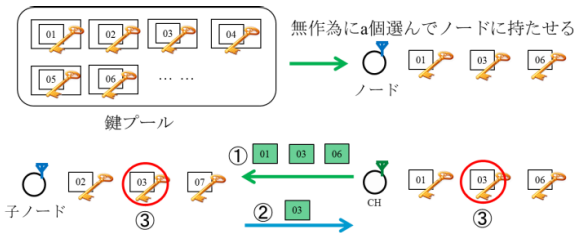

図 2 SecLEACH の概要
Fig. 2 SecLEACH.

ランダム鍵配布方式とは，鍵の集合である鍵プールから鍵（要素鍵）を各ノードにランダムに持たせる方式で，要素鍵には鍵 ID が付けられている．**図 2** の上部に多くの鍵を持つ鍵プールから各ノードが a 個ずつランダムに鍵を設定される様子が示される．CH-子ノード間の暗号鍵を共有する手順は以下のようになる．

① CH が自分の持つ鍵の鍵 ID を子ノードに知らせる．
②子ノードはその中から自分も共通に持つ鍵の鍵 ID を CH に知らせる．
③その後子ノードと CH は共通した鍵のハッシュ値をとるなどして共通鍵を生成し保存する．

図 2 の下部に上記手順の例が示される．① では CH は自分が持つ要素鍵の ID01，03，06 を子ノードに知らせ，② では子ノードがその中から自分が共通に持つ要素鍵の ID03 を知らせることによって，03 の ID を持つ要素鍵が共有され，それから共通鍵が生成される．

このように SecLEACH では要素鍵の ID のみを相手に知らせることによって共通鍵を決めることができる．しかしながら，SecLEACH にはいくつかの問題点がある．まず 1 つ目に，CH と子ノードの鍵共有は確率的なものであることがあげられる．これによって，子ノードは最近隣の CH と暗号通信を行えない可能性がある．さらには，子ノードがどの CH とも鍵を共有できない場合が確率的に起こり，その場合子ノードはどの CH とも暗号通信を行うことができない．2 つ目の問題として，いくつかのノードの鍵が漏洩することによって，他のノードの鍵が解析できる可能性がある．すなわち，あるノードを a とし，別のノードを b としたときに，a と b で同じ要素鍵を保存している場合，a の要素鍵の漏洩は b の要素鍵の漏洩も意味するため，b の共通鍵が解析できる可能性がある．この問題に対しては一般的に，保存する要素鍵数を多くすると解析が困難になるが，その分要素鍵の記憶量が多くなるという問題が発生する．ランダム鍵配布方式を改良し，記憶量を削減できる手法 [18] もあるが，鍵共有が確率的である，または他ノード解析に弱いという問題は解決されていない．

2.3 MS-LEACH

MS-LEACH [4] では LEAP（Localized Encryption and Authentication Protocol）と呼ばれる鍵共有方式を用いてノード間の通信の秘匿化を実現する．

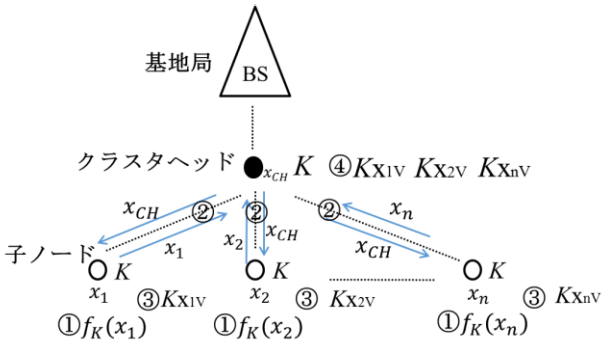

図 3 MS-LEACH の概要
Fig. 3 MS-LEACH.

LEAP では，各ノードに 4 つの鍵を持たせることを目的としている．1 つ目の鍵は BS と共有する Individual Key，2 つ目はネットワークのすべてのノードと共有する Group Key，3 つ目は近隣ノード全体で共有する Cluster Key，最後は近隣ノードと 1 対 1 で共有する Pairwise Key である．

4 つの鍵のうち，MS-LEACH は Pairwise Key の共有によって暗号通信を行う．その共有の手順は以下のようになる．

①初期鍵 K を管理者が作成し，全ノードに設定する．子ノード u は自分の ID を用いてマスタ鍵 $K_u = f_K(u)$ を作成する．ここで，$f_K(a)$ とは K と a を入力とした擬似乱数生成関数である．
② CH と子ノードは自身の ID を互いに送信する．
③子ノード u は受信した CH のノード ID と自身のマスタ鍵を用いて擬似乱数生成を行い pairwise key：$K_{UV} = f_{Ku}(ch)$ を生成する．
④ CH は受信した子ノードの ID とあらかじめ所持している初期鍵を用いてその子ノードに対応するマスタ鍵 Ku を生成し，マスタ鍵 Ku と自身の ID を用いて pairwise key：$K_{UV} = f_{Ku}(ch)$ を生成する．
⑤ pairwise key を生成した後に，すべてのノードは初期鍵 K を消去する．

図 3 以降において，CH を●，子ノードを○，BS を△で表す．図 3 では ① において子ノード x_i が自分のマスタ鍵 $f_K(x_i)$ を生成し，② において CH と各子ノードは互いの ID を交換し，③ において子ノードが CH との pairwise key：$K_{xiV} = f_{Ki}(ch)$ を生成し，④ において CH が全子ノードとの pairwise key：$K_{xiV} = f_{Ki}(ch)$ を生成する様子を示す．

SecLEACH と MS-LEACH を比較すると，SecLEACH では鍵共有が確率的なものになっているのに対して，MS-LEACH では CH は全ノードと確実に鍵共有を行うことができる．また，SecLEACH ではある複数のノードで同じ要素鍵を持つ可能性があるため，あるノードの要素鍵が漏えいすると，他のノードにも影響を及ぼす可能性があ

る．それに対して，MS-LEACH では CH と子ノードの ID（すべて異なる）を用いて鍵共有を行っているため，生成される共通鍵はすべて異なる．よって，初期鍵を消去後にあるノードが解析されたとしても，他の CH-子ノード間の暗号鍵の秘匿性は保たれる．しかしながら MS-LEACH は初期鍵 K があれば公開情報であるノード ID を用いてすべての鍵を生成できることから，初期鍵 K を消去する前に子ノードを 1 つでも盗まれ解析されると，すべての鍵が漏えいしてしまうという問題をかかえている．LEAP では，初期鍵の消去はノードに内蔵されるタイマによって管理されると想定される．また，ノードは耐タンパ性を持たないことからタイマを止めることは容易であり，これによって初期鍵消去が防止され，鍵解析できることが指摘されている [11]．また，pairwise key は擬似乱数生成器によって生成されるため，情報理論的な安全性を MS-LEACH は実現しない．

2.4 秘密分散を用いた鍵共有方式

この方式は Shamir の (k, n) 秘密分散法を用いる．Shamir の (k, n) 秘密分散法は情報を $k-1$ 次の多項式を用いて以下のように n 個に分散し，生成した情報をそれぞれ n 個のサーバに分散管理する．例として，s を秘密情報とし記述する．以下の演算は素数 p を法として行われる．

[分散]

1. $s < p$ かつ $n < p$ である素数 p を選ぶ．
2. GF(p) から n 個の x_i（$i = 1, \ldots, n$）を選択する．
3. GF(p) から $k-1$ 個の乱数 a_l（$l = 1, \ldots, n$）を選択し，以下の式を生成する．

$$W_i = s + a_1 x_i + a_2 x_i^2 + \cdots + a_{k-1} x_i^{k-1} \qquad (1)$$

4. 式 (1) に各 x_i を代入し，分散情報 W_i を計算する．そして各サーバに x_i と W_i を配布する．

[復元]

- k 個のサーバから分散情報 W_i と x_i を集め，k 個の連立方程式を作る．連立方程式を解き，s を復元する．

続いて，Shamir の秘密分散法を用いた鍵共有方式 [8] について記述する．この方式では BS-CH 間の暗号鍵（ネットワーク鍵）を共有した後に，CH-子ノード間の暗号鍵（クラスタ鍵）を共有する．まず，BS-CH 間の鍵共有の手順について述べる，以下において，ネットワーク鍵を K_{CH} とし，S_{CH} および n より大きな素数 p や十分大きな素数 N，およびその原始元 g は共有されているとする．また，各 CH_i（$i = 1, \cdots, n$）は固有の鍵を所持していて，BS はすべての鍵を知っているものとする．また，② ⑥ は N を法として行われ，① ④ ⑦ は p を法として行われる．③ ⑤ の暗号化は安全であるとする．

① BS は Shamir の秘密分散を用いて，秘密情報 S_{CH} に対して $k-1$ 次の多項式を定める．また，$Z_{CH} = K_{CH} + S_{CH}$ を計算しブロードキャストする．

② BS はランダムに秘密鍵 x_0 を，各 CH_i は秘密鍵 x_i を選択し，原始元 g に対して $y_i = g^{xi}$ を計算する．

③ 各 CH_i は x_i を暗号化し (y_i, ID_{CHi}) とともに BS に送る．

④ BS は各分散値 $f(ID_{CHi})$ を生成する．

⑤ BS は全 x_i を暗号化して各 CH_i に送り，$(y_0, ID_{CHj}, f(ID_{CHj})(y_i)^{x0})$ をブロードキャストする．

⑥ 各 CH_i は k 個の $f(ID_{CHj})(y_i)^{x0}$ から $f(ID_{CHj})(y_i)^{x0}/(y_0)^{xi}$ を計算して k 個の $f(ID_{CHj})$ を取り出す．

⑦ 各 CH_i は，秘密情報 S_{CH} を復元してネットワーク鍵 $K_{CH} = Z_{CH} - S_{CH}$ を復元する．

CH-子ノード間の暗号鍵であるクラスタ鍵の共有についても上記のネットワーク鍵とほぼ同様に行う．

この手法は，クラスタツリー型センサネットワークの鍵共有に秘密分散を用いるという点では新しいが，多くの無駄を含む．まず，子ノードは y_i を計算するためにべき乗演算を行う必要があり，計算資源が乏しいセンサネットワークには不向きである．また，各ノードが復元した鍵を保存せず，毎回暗号化された分散値を集めて鍵を復元する場合，煩雑な計算と多くの通信が通信のたびに要求される．よって，上記処理は 1 度だけ行い復元したネットワーク鍵またはクラスタ鍵を各ノードが保存すると考えられるが，その場合 CH を解析すればすべての鍵が漏洩するという問題が解決されない．

また，この方式ではクラスタ内で共通のクラスタ鍵を用いているが，これは各ノードの記憶容量を少なくするためと考えられる．CH-子ノードごとに違う鍵を設定する場合，CH は少なくとも $n-1$ 個の鍵を記憶する必要がある．

また，秘密分散法と組み合わせず，公開鍵暗号を用いる鍵共有法も存在する [19] が，公開鍵暗号は前述のように非常に大きな計算量と通信量が必要であり，省電力を考える場合，センサノードには負荷が大きい．

3. 提案方式 1

3.1 XOR を用いた閾値秘密分散法

文献 [8], [9] の手法は Shamir の秘密分散法を用いているが，センサネットワークは計算資源が少ないため，多項式演算が必要なく，XOR 処理のみで分散・復元処理を高速に行うことができる秘密分散法 [12] が適している．本節では，この秘密分散法のアルゴリズムについて記述する．

まず，本節で用いる記号と演算子を以下に定義する．

$\oplus$：ビット単位の XOR 演算

n_p：$n_p > n$ かつ $n_p > Si$ を満たす素数

$\|$：ビット列の結合

n：分散値（ここでは $n_p = n$ として説明する）

k：閾値

i：ユーザ番号

j：部分分散情報の番号 $0 \leq j \leq n-2$

d：各処理におけるデータのビット長
$\{0,1\}^d$：0 と 1 から構成される d ビットのデータ
S：秘密情報（$S \in \{0,1\}^{(n-1)d}$）
Sx：部分秘密情報（$1 \leq x \leq n-1, S_0 \in \{0\}^d$）
$r_\propto^\beta$：乱数 $r_\propto^\beta \in \{0,1\}^d$, $0 \leq \alpha \leq k-2$, $0 \leq \beta \leq n_p - 1$
W_i：ユーザに配布される分散情報（$W_i \in \{0,1\}^{(n-1)d}$）
$W_{(i,j)}$：ユーザに配布される部分分散情報 $W_{(i,j)} \in \{0,1\}^d$

以下に分散，復元それぞれのアルゴリズムを示す．

［分散］

①ディーラは秘密情報（$S \in \{0,1\}^{(n-1)d}$）を $n-1$ 個の部分秘密情報（$S_1, S_2, \ldots, S_{n-1}$）に分割し，さらに $S_0 \in \{0\}^d$ を生成する．

②ディーラは $(k-1)n-1$ 個の d ビットの乱数 $r_\propto^\beta$ を独立に生成する．

$$r_0^0, r_1^0, \cdots, r_{n-2}^0, r_0^1, \cdots, r_{n-2}^1, r_{n-1}^1, \cdots, r_0^{k-2}, \cdots, r_{n-1}^{k-2}$$

③ディーラは部分分散情報 $W_{(i,j)}$ を $0 \leq i \leq n-1$, $0 \leq j \leq n-2$ において以下のようにそれぞれ生成する．

$$W_{(i,j)} = S_{j-i} \oplus \{\oplus_{h=0}^{k-2} r_{h \cdot i+j}^h\} \in \{0,1\}^d$$
$$(0 \leq i \leq n-1,\ 0 \leq j \leq n-2)$$

④ディーラは $0 \leq i \leq n-1$ において各部分分散情報を連結して分散情報 W_i を生成する．

［復元］

①復元者は k 個の分散情報（$W_{t0}, W_{t1}, \ldots, W_{tk-1}$）を集め，それをすべて以下のように部分分散情報に分割する．

$$(W_{(t0,0)}, W_{(t_0,1)}, \ldots, W_{(t_0,n-2)}), (W_{(t1,0)}, W_{(t1,1)}, \cdots,$$
$$W_{(t1,n-2)}), \ldots (W_{(tk-1,0)}, W_{(tk-1,1)}, \ldots, W_{(tk-1,n-2)})$$

②復元者はすべての部分分散情報を以下のように表し，$kn-2$ 元の 2 進数ベクトル $V_{ti,j}$ を生成する．
部分分散情報 $W_{(t_i,j)}$ の場合

$$W_{(t_i,j)} = V_{(ti,j)} \cdot R_{(k,n)}$$
$$R_{(k,n)}$$
$$= (S_1, \ldots, S_{n-1}, r_0^0, \ldots, r_{n-2}^0, r_0^1, \ldots, r_{n-1}^1, \ldots,$$
$$r_0^{k-2}, \ldots, r_{n-1}^{k-2})^T$$

③復元者は ② で定まった $V_{(t0,0)}, \ldots, V_{(tk-1,n-2)}$ のベクトルから以下の 2 進数の行列を生成する．

$$M_{(t_0,\ldots,t_{k-1})}^{(k,n)}$$
$$= (V_{(t0,0)}, \ldots, V_{(t0,n-1)}, \ldots, V_{(tk-1,0)}, \ldots, V_{(tk-1,n-1)})^T$$

④復元者はすべての部分分散情報を表す $k(n-1)$ 元ベクトル $W_{(t0,\ldots,t_{k-1})}$ を以下のように表す．

$$W_{(t0,\ldots,t_{k-1})} = (W_{(t0,0)}, W_{(t_0,1)}, \ldots,$$
$$W_{(t_0,n-2)}, \ldots, W_{(tk-1,0)}, W_{(tk-1,1)}, \ldots, W_{(tk-1,n-2)})^T$$
$$W_{(t0,\ldots,t_{k-1})} = M_{(t_0,\ldots,tk-1)}^{(k,n)} \cdot R_{(k,n)}$$

⑤復元者は ④ の式に Gauss-Jordan の掃き出し法を用いて対角化処理を行う．これによって，すべての部分秘密情報に該当する部分を求める．

⑥復元者は部分秘密情報を連結して秘密情報 S を復元する．

3.2 LEACH への適用

従来方式としてクラスタツリー型センサネットワーク以外のネットワーク構成に対して，秘密分散法を適用する [15], [16], [17] のような手法が存在する．これらは鍵を近隣の n 個のノードに分散し，その中の k（$k \leqq n$）個の分散情報を集めることによって鍵を復元する．よって，k 本の通信路を盗聴すれば鍵が解析される．そのため，2.4 節に示したように分散情報を秘匿しながら分散情報を集める非効率な仕組みが導入されている．それに対して，提案方式の基本原理は閾値 k を対象とする子ノードの数より大きな値に設定し，鍵の復元を許可されたノードだけがあらかじめ復元に必要な複数の分散情報を持つことである．よって，以下において $(k,n) = (2,3)$ とし，子ノードが W_{i1}, W_{i2} を持ってもよいが，鍵を直接持つ後述の手法の方が簡単である．これによって，子ノードからの通信が全部盗聴されても鍵は漏洩しない．さらに，その閾値からの差分にあたる分散情報または鍵自体を，暗号通信を行う当該ノードが保有することによって鍵復元を実現する．

はじめに以下の前提をおく．クラスタには $m+1$ 個のノードが含まれ，その ID を ID_1, ID_2, $ID_3, \ldots, ID_m$, ID_{m+1} とし（子ノードの数が m 個で CH が 1 個），ID_i（$i = 1, \cdots, m+1$）のノードには自らの ID とノードごとに異なる固有鍵 K_i と CH-子ノード間の暗号通信に用いられるリンク鍵 L_i（L_i は $n_p - 1$ 以下の一様分布する真性乱数から独立に選択される）が事前に初期格納されているとする．また，BS はすべてのノードについて上記情報を知っているとする．さらに，以下では CH が 1 つの場合について説明するが，クラスタおよび CH が複数あっても以下の処理を独立に実行すればよい．

全提案方式における鍵共有は，他方式と同様に CH が定まった後の setup phase において実行される．提案方式 1 は，各子ノードが 1 対 1 で CH と鍵共有を行うため，対象とする子ノードの数は 1 であるので $k = 2$ として，CH-子ノード間のリンク鍵共有を以下のように行う．以下において説明を簡単にするため，BS は子ノードに直接暗号化した分散情報を送るが，CH は各子ノードの固有鍵を知らないので，CH を介して送信しても送信段階での鍵漏洩は発生せず問題ない．

［分散フェーズ］

①CH は自分のクラスタ内の子ノードの ID を BS へ知ら

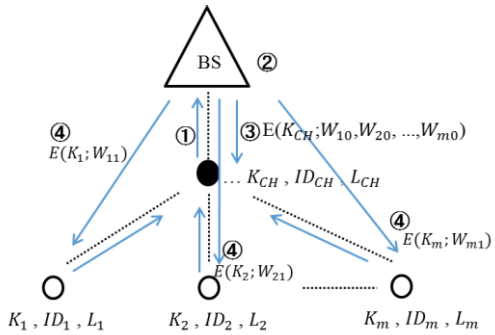

図 4 提案方式 1 の鍵共有方式

Fig. 4 Key sharing scheme on proposed method 1.

せる．

② BS はディーラとして，秘密情報である子ノード ID_i のリンク鍵 L_i に対して，3.1 節を用いて $(2,2)$ の秘密分散を行い，2 つの分散情報 W_{i0} と W_{i1} を生成する．この処理を m 個の子ノードに対して独立に実行する．

③ BS は生成した W_{i0}（$i=1,\cdots,m$）を CH の固有鍵で暗号化して CH に送り，CH は W_{i0}（$i=1,\cdots,m$）を復元して，ID_i に対応付けて保存する．

④ BS は各 W_{i1}（$i=1,\cdots,m$）を ID_i の子ノードの固有鍵で暗号化して送り，各子ノードは W_{i1} を復号して保存する．

［鍵共有・暗号通信フェーズ］

①子ノード ID_i はセンシングした情報 f_i を自分のリンク鍵 L_i で暗号化した $E_{-Li}(f_i)$ とともに以下を CH に送る．

$$(ID_i, W_{i1}, E_{-Li}(f_i))$$

② CH は復元者として，ID_i に対応付けて保存している W_{i0} と受信した W_{i1} から，秘密情報であるリンク鍵 L_i を復元する．

③ CH は L_i を用いて $\mathrm{E}_{-Li}(f_i)$ を復号し，f_i を取り出す．

④ CH は受信した W_{i1} とリンク鍵 L_i を消去する．

図 4 は上記手順を示す．図 4 では ① において CH が自分のクラスタ内の子ノードの ID を BS へ知らせ，② において BS が各ノードに関する 2 つの分散情報 W_{i0} と W_{i1} を生成し，③ において BS は CH に W_{i0}（$i=1,\cdots,m$）を暗号化して送信し，④ において BS は各子ノード ID_i に W_{i1} を暗号化して送る様子を示す．ただし，各子ノード ID_i は固有鍵 K_i，自身の ID である ID_i，リンク鍵 L_i を保持しているとする．

3.3 安全性

提案方式 1 の安全性について評価する．ただし，3.1 節に示される秘密分散法は情報理論的安全性を持つことが証明されている [12]．

(1) CH 解析に関する安全性

提案方式 1 では，CH は片方の分散情報しか保持していないため，暗号通信時以外で CH が解析されてもリンク鍵が漏洩することはない．特に，3.1 節の秘密分散法は情報理論的な安全性を持つため，全子ノードのリンク鍵はまったく漏洩しない（CH が複数あっても各クラスタの処理は独立に行われるため同様）．ただし，暗号通信時に CH は通信対象の子ノードのリンク鍵を生成するので鍵漏洩の恐れがあるが，他ノードの鍵には影響しない．また，CH は暗号通信終了時にリンク鍵などを消去するので，それ以外のタイミングで解析されても暗号通信時以外では鍵は漏洩しない．それに対して，他の方式は CH が setup phase において全子ノードの鍵を保有するため，どのタイミングでも CH が解析されるとすべての子ノードの鍵が漏洩する．

(2) 子ノード解析に関する安全性

提案方式 1 では，子ノードは自分のリンク鍵 L_i を所持している．よって，子ノードが盗難・解析された場合，リンク鍵 L_i が漏洩する．ただし，子ノードのリンク鍵 L_i は独立に設定されているので，他のノードのリンク鍵にはまったく影響せず，情報理論的安全性を持つ．それに対して，前述したように SecLEACH は多くの子ノードが解析された場合，MS-LEACH は初期鍵を消去する前に子ノードが盗難された場合に他のリンク鍵も漏洩する．2.4 節の方式は鍵復元後であれば，子ノードを 1 つ盗み・解析するだけでクラスタ鍵が漏洩する．

(3) CH と子ノード解析に関する安全性

提案方式 1 では CH と子ノードが盗難・解析されても子ノードが解析された場合と同様の安全性を持つ．すなわち，盗難・解析された子ノードのリンク鍵は漏洩するが，各リンク鍵は独立に設定されており，CH が持つ分散情報を解析しても盗難されていない他ノードは情報理論的に安全であるため，盗難・解析された子ノード以外のリンク鍵はまったく漏洩しない．

(4) 通信路の盗聴に対する安全性

分散フェーズ ③ ④ で用いられる暗号化は安全であると仮定すると，各ノードへの通信は暗号化されているため分散情報は漏洩しない．鍵共有フェーズにおいては 1 つの分散情報である W_{i1} がそのまま通信されるが，W_{i0} が通信路に出ることはないので，情報理論的安全性が実現される．一方，SecLEACH では通信路に出る情報は鍵 ID だけであり，MS-LEACH ではノード ID のみであるので安全である．2.4 節の方式も用いる暗号が安全であれば盗聴によって鍵は漏洩しない．

通信量や計算量，記憶量などに関する評価は 5 章で行う．

4. 提案方式 2

提案方式 1 は上記のように高い安全性を持つが，CH は管理する子ノード m 個の分散情報を保存する必要があった．よって，m が比較的大きいときそれにともなう記憶容量が必要である．そこで，CH が管理する子ノードに関す

る分散情報を保存する必要がない手法を提案方式 2 として以下に示す．ただし，提案方式 2 は文献 [12] の秘密分散法をそのまま適用しても実現できないので，文献 [13] に示される非対称化を行う．非対称化とは通常 n 個のサーバに均等に保存される分散情報を，ランプ型 [14] のように均一に削減するのではなく，$k-1$ 個までのサーバが持つ分散情報を 0（鍵のみを管理）にする，すなわち非均一に分散情報を削減することを指す．文献 [13] に示される非対称秘密分散法は Shamir の秘密分散法に適用され，用いる暗号方式に依存した計算量的安全性を持つことが証明されている．そこで，本論文ではまず 3.1 節に示した XOR を用いる秘密分散法の非対称化を行う．すなわち，分散情報を 0 にする子ノードは鍵と擬似乱数生成器を持ち，生成した擬似乱数を分散情報として扱う．

4.1 XOR を用いる非対称秘密分散法

［分散］

①ディーラは $i_0, i_1, \ldots, i_{n-1}$ の n 台のサーバから任意に t 台（$1 \leq t \leq k-1$）を選択し，そのサーバ番号を i とする（$1 \leq i \leq t$）．これを分散情報 0 のサーバと呼ぶ．

②ディーラは選択したサーバ i に鍵 key_i を設定する．

③ディーラはサーバ i の鍵 key_i を用いて各々 $(n-1)d$ ビットの擬似乱数 q_i を生成する．

④ディーラはサーバ i において擬似乱数 q_i をそれぞれ $n-1$ 個の d ビットの部分擬似乱数 $q_{(i,j)}$（$0 \leq j \leq n-2$）に分割する．また，秘密情報 S を $n-1$ 個の d ビットの部分秘密情報（$S_1, S_2, \ldots, S_{n-1}$）に分割し，$S_0 = 0$ とする．

$$q_i = q_{(i,0)} \,\|\, q_{(i,1)} \,\|\, \ldots \,\|\, q_{(i,n-2)}$$
$$S = S_1 \,\|\, S_2 \,\|\, \ldots \,\|\, S_{n-1},\ S_0 \in \{0\}^d$$

⑤ディーラは $(k-1)n-(n-1)t-1$ 個の乱数を生成し，任意に $\alpha^h_{h\cdot i+j}$ に割り当てる．

⑥ディーラは $(n-1)t$ 個の部分擬似乱数 $q_{(0,0)}, \ldots, q_{(0,n-2)}, \ldots, q_{(f,n-2)}$ から以下の式が成り立つように残り $(n-1)t$ 個の $\alpha^h_{h\cdot i+j}$ を定める．

$$q_{(i,j)} = S_{i-j} \oplus \{\oplus_{h=0}^{k-2} \alpha^h_{h\cdot i+j}\}$$
$$(1 \leq i \leq t,\ 0 \leq j \leq n-2)$$

⑦ディーラは ⑤ ⑥ で得られた $\alpha^h_{h\cdot i+j}$ を用いて部分分散情報を以下の範囲で生成する．

$$W_{(i,j)} = S_{i-j} \oplus \{\oplus_{h=0}^{k-2} \alpha^h_{h\cdot i+j}\}$$
$$(t+1 \leq i \leq n-1,\ 0 \leq j \leq n-2)$$

⑧ディーラは各部分分散情報を連結して分散情報 W_i を生成する．これをサーバ i（$1 \leq i \leq t$）以外に配布する．

$$W_i = W_{(i,0)} \,\|\, W_{(i,1)} \,\|\, \ldots \,\|\, W_{(i,n-2)}$$
$$(W_i \in \{0,1\}^{(n-1)d})$$

［復号］

復元者は分散情報 0 のサーバ t 台と分散情報を持つサーバ $n-t$ 台のうちから任意に k 台選択し，分散情報を k 個集めて秘密情報を復元する．ただし，分散情報 0 のサーバは鍵 key_i を用いて擬似乱数 q_i を生成し，これを復元者へ送る．一方，分散情報を持つサーバは分散情報 W_i を送信する．以降では q_i と W_i を合わせて k 個の分散情報として，$(W_{t0}, W_{t1}, \ldots, W_{tk-1})$ で表す．以降は，3.1 節に示した［復元］と同じである．

4.2 XOR を用いる非対称秘密分散法の安全性

文献 [12] に示される 3.1 節の秘密分散方式の安全性は以下の 2 つの定理によって証明される．

● 定理 1

(k,n) 閾値秘密分散法のアクセス構造 Γ は $\Gamma = \{A \in 2^P \,|\, |A| \geq k\}$ と定義されている．ここで P は $P = \{P_i \,|\, 0 \leq i \leq n-1\}$ を満たし，A は $A \subset P$ となる部分集合である．

このとき，ユーザ数が $|A| \leqq k-1$ を満たすような任意の集合 A を用意する．A はアクセス構造 Γ には含まれていないため，以下を満たす．

$$H(S \,|\, V_A) = H(S)$$

ここで V_A は A 内の各ユーザに与えられている分散時の確率変数を表す．

● 定理 2

$|A| \geq k$ を満たす参加者の集合 A を用意する．3.1 節の復元アルゴリズムを用いることで集合 A の各参加者に与えられた分散情報から秘密情報が復元できる．

3.1 節と 4.1 節に示す手法において，秘密情報と分散情報の関係は同じであるので，定理 2 と同様に，4.1 節の手法においても $|A| \geq k$ を満たす参加者の集合 A の参加者は秘密情報を復元できることがいえる．

一方，3.1 節と 4.1 節の違いは 3.1 節で用いる乱数が一様ランダムな真性乱数であるのに対して，4.1 節の手法で用いる乱数は擬似乱数生成器によって生成される擬似乱数である点である．よって，4.1 節は定理 1 に示されるように $H(S \,|\, V_A) = H(S)$ ではなく，$H(S) = H(S \,|\, V_A) + \varepsilon$ となる（ε は擬似乱数生成器の安全性に依存して漏洩する S に関する情報量）．よって，4.1 節に示す非対称秘密分散法は用いる擬似乱数生成器が持つ計算量的な安全性に依存した安全性を実現するといえる．

4.3 LEACH への適用

各ノードが保持する情報は提案方式 1 と同様である．

［分散フェーズ］

①CH は自分のクラスタ内の子ノードの ID を BS へ知らせる．

②BS はディーラとして，秘密情報である子ノード ID_i の

リンク鍵 L_i に対して，4.1 節を用いて秘密分散を行う．ただし，$n=k=2$，$t=1$ とする．すなわち，分散情報を持たないサーバ i を CH とし，CH が持つ固有鍵 K_{CH} を key として ID_i を暗号化したものを擬似乱数 q_i として 4.1 節の分散処理を実行し，子ノードに送る分散情報 W_{i1} を生成する．BS はこの処理を m 個の子ノードに対して独立に実行する．

③ BS は W_{i1}（$i=1,\cdots,m$）を各子ノードの固有鍵で暗号化して送り，各子ノードはそれを復号して保存する．

［鍵共有・暗号通信フェーズ］

①子ノード ID_i はセンシングした情報 f_i を自分のリンク鍵 L_i で暗号化した $E_{\text{-}Li}(f_i)$ とともに以下を CH に送る．

$$(ID_i, W_{i1}, E_{\text{-}Li}(f_i))$$

② CH は各子ノードから送られてきたノード ID_i を自身の持つ固有鍵 K_{CH} で暗号化して擬似乱数 q_i を生成し，分散情報 W_{i0} とする．この W_{i0} と送られてきたもう 1 つの分散情報 W_{i1} から，秘密情報であるリンク鍵 L_i を復元する．

③ CH は L_i を用いて $E_{\text{-}Li}(f_i)$ を復号し，f_i を取り出す．

④ CH は受信した W_{i1} とリンク鍵 L_i を消去する．

4.4 安全性

4.2 節において提案方式 2 で用いる秘密分散法は計算量的安全性を持つことが示された．ここでは，以下の各場合についてその安全性を詳細に検討する．

(1) CH 解析に関する安全性

CH が解析されると片側の分散情報を生成するための鍵が漏洩する．しかし，これは提案方式 1 において片側の分散情報がすべて漏洩した状態と等価である．提案方式 2 においても，提案方式 1 と同様に子ノードのリンク鍵は独立に設定されているため，片側の分散情報だけからリンク鍵に関する情報を得ることはできない．すなわち，CH 解析に関しては提案方式 1 と同様に情報理論的安全性を持つ．

(2) 子ノード解析に関する安全性

提案方式 2 も提案方式 1 と同様に，子ノードが持つリンク鍵が漏洩する．たとえば，$(2,2)$ の非対称秘密分散における分散情報を，CH が持つ $W_{i0}=R_i$，子ノードが持つ $W_{i1}=L_i \oplus R_i$ とする（L_i はリンク鍵，R_i は擬似乱数生成器から生成される乱数）．いくつかの子ノードの解析から W_{i1} と L_i が漏洩し，擬似乱数生成器の脆弱性により，R_i から他の R_j が分かったとする．しかしその場合でも，①と同様に R_j が分かっても W_{j1} が分からなければ L_j は分からない（L_j は一様ランダムに選択されている）．すなわち，子ノードの解析によって，その子ノードの鍵は漏洩するが，他のノードに影響しない．すなわち，情報理論的安全性を持つ．

(3) CH と子ノード解析に関する安全性

提案方式 2 も提案方式 1 と同様に，CH と子ノードが盗難・解析されても子ノードが解析された場合と同様の安全性を持つ．

(4) 通信路の盗聴に関する安全性

提案方式 2 では，提案方式 1 と同様の仮定をおくと分散フェーズにおける各ノードへの通信は暗号化されているため分散情報は漏洩しない．鍵共有フェーズにおいては子ノードから分散情報 W_{i1} と自身のノード ID がそのまま通信される．前記と同様に擬似乱数生成器の脆弱性から R_i が分かったとすると，W_{i1} にはリンク鍵が含まれているためリンク鍵が漏洩する可能性がある．すなわち，この場合においてのみ 4.2 節に示したように擬似乱数生成器の安全に依存する．

5. 提案方式 3 およびその変形

提案方式 1，2 および従来方式における子ノードは，CH との暗号鍵を保持しているため子ノードを解析すれば少なくともその子ノードの鍵は漏洩する．そこで，暗号通信時以外ですべての子ノードを解析しても，そのノードの暗号鍵も含めてまったく鍵漏洩が発生しない方式を提案方式 3 として以下に示す．提案方式 3 はすべての子ノードを対象とするため，閾値 k はそれを超える $k=m+1$ となる．ここでは 3.1 節で説明した情報理論的安全性を持つ XOR による秘密分散法を用いた手法を説明する．

5.1 LEACH への適用

ノード ID_i があらかじめ保持する情報は自らの ID と固有鍵のみであり，リンク鍵は保持しない．また，子ノードの数 m は $m>1$ とする．

［分散フェーズ］

① CH は自分のクラスタ内の子ノードの ID を BS へ知らせる．

② BS は子ノード ID_i のリンク鍵 L_i を定めてそれを秘密情報とし，$n=m+2, k=m+1$ として 3.1 節を用いて秘密分散を行い，n 個の分散情報 W_{ij} $(j=1,\cdots,m+2)$ を生成する．BS はこの処理を m 個の子ノード ID_i $(i=1,\cdots,m)$ に対して独立に実行する．

③ BS は W_{ij}（$i=1,\cdots,m$）を子ノード ID_i にその固有鍵で暗号化して送り，各ノードはそれを復号して保存する．CH には $W_{i,m+1}$ と $W_{i,m+2}$（$i=1,\cdots,m$）を CH の固有鍵で暗号化して送る．

［鍵共有・暗号通信フェーズ］

①子ノード ID_i はセンシングした情報 f_i を CH に送るとき，全ノードに自分の ID_i をブロードキャストする．

② ID_i を除く全ノード（CH も含む）は保存している ID_i に対応する分散情報をブロードキャストする．ただし，CH は $W_{i,m+2}$ を送信しない．

③ ID_i は送信された m 個の分散情報に自分だけが持つ分散情報を加えた $k=m+1$ 個の分散情報から自らのリンク鍵 L_i を復元し，復元したリンク鍵 L_i で f_i を暗号化した $E_{-Li}(f_i)$ を CH に送る．

④ CH は m 個の分散情報に自分が持つ未公開の $W_{i,m+2}$ を加えた $k=m+1$ 個の分散情報から，ID_i のリンク鍵 L_i を復元する．

⑤ CH は L_i を用いて $E_{-Li}(f_i)$ を復号し，f_i を取り出す．

⑥ CH と子ノード ID_i は受信した分散値とリンク鍵 L_i を消去する．

4.1 節に示す非対称秘密分散法を用いる場合を提案方式 3' とする．CH に分散情報が集中しないよう，提案方式 3' は［分散フェーズ］における ② ③ を以下のように変形する．

② BS は子ノード ID_i のリンク鍵 L_i を定めてそれを秘密情報とし，$n=m+2$, $k=m+1$, $t=m$ として 4.1 節を用いて秘密分散を行う．ただし，ID_i を除く $m=k-1$ 個のノード（CH 含む）を分散情報 0 のサーバとする．すなわち，m 個のノードが持つ固有鍵を key_i（$i=1,\cdots,m$）として ID_i を暗号化して擬似乱数 q_i を生成して 4.1 節の分散処理を実行し，保存すべき 2 個の分散情報 $W_{i,m+1}$ と $W_{i,m+2}$ を生成する．BS はこの処理を m 個の子ノードに対して独立に実行する．

③ BS は ID_i（$i=1,\cdots,m$）に $W_{i,m+1}$ と $W_{i,m+2}$ を ID_i の固有鍵で暗号化して送る．

また，［鍵共有・暗号通信フェーズ］の ② ④ を以下のように変形する．

② CH を除く全ノードは ID_i に対応して生成または保存した分散情報をブロードキャストする．ただし，ID_i は $W_{i,m+2}$ を送信しない．

④ CH は送信された m 個の分散情報と自分が生成した分散情報から $k=m+1$ 個の分散情報得て，ID_i のリンク鍵 L_i を復元する．

これによって，各ノードは 2 個の分散情報を保存するだけでよく，大きな記憶量の削減ができる．

ただし，提案方式 3 では 1 つでも子ノードが故障または通信不能になると，鍵が復元できない．そこで，u 個（$0 \leqq u \leqq m-2$）の子ノードが通信不能でも復元可能とする場合は，［分散フェーズ］② において $n=m+u+2$ として分散情報を生成し，③ で CH に $W_{i,m+1}$，$W_{i,m+2}$ に加えて u 個の分散情報 $W_{i,m+3}$〜$W_{i,m+u+2}$ を暗号化して送る．また，［鍵共有・暗号通信フェーズ］では ② は以下となる．

② ID_i を除く通信可能なノード（CH も含む）は保存している ID_i に対応する分散情報を 1 個ブロードキャストする．ただし，CH は送信された分散情報が m 未満のとき，自分が保有する $W_{i,m+3}$〜$W_{i,m+u+2}$ の分散情報の中から，m になるまで分散情報を送信する．

すなわち，子ノードからの分散情報が $m-v$ 個（$0<v \leqq u$）であった場合，CH は追加された u 個の分散情報の中から v 個の分散情報を送信するが，$W_{i,m+2}$ は未公開のままであるので，鍵の復元が可能である．これを提案方式 4 とする．また，4.1 節に示す非対称秘密分散法を用いる場合は，$m=k-1$ 個の子ノードを分散情報 0 のサーバとして BS は保存すべき分散情報 $W_{i,m+1}$〜$W_{i,m+u+2}$ を作成し，CH に暗号化して送る．これを提案方式 4' とする．ただし，提案方式 3' のように CH を分散情報 0 のサーバとし，$W_{i,m+1}$〜$W_{i,m+u+2}$ を CH ではなく ID_i に送り，CH に分散情報を集中させないようにすることもできる．

最後に，BS がすべての L_i を共通に設定すればクラスタ共通のクラスタ鍵となる．この場合，［分散フェーズ］の ② ③ を以下のように変形する．これを提案方式 5 とする．

② BS はクラスタ鍵 L を定めてそれを秘密情報とし，$n=m+u+2$, $k=m+2$（$0 \leqq u \leqq m-2$）として 3.1 節を用いて秘密分散し，n 個の分散情報 W_j（$j=1,\cdots,m+u+2$）を生成する．

③ BS は CH（CH は ID_{m+1} とする）を含む全ノード ID_i に分散情報 W_i と W_{m+2} を暗号化して送り，CH に追加した u 個の分散情報 W_{m+3}〜W_{m+u+2} も暗号化して送り，各ノードはそれを復号して保存する．

［鍵共有フェーズ］は以下のようになる．

①クラスタ鍵 L の復元を要求する信号を送る．

② CH を含む通信可能なノードは W_i をブロードキャストする（CH は W_{m+1} をブロードキャスト）．

③ CH は送信された分散情報が $m+1$ 未満のとき，自分が保有する W_{m+3}〜W_{m+u+2} の分散情報の中から，$m+1$ になるまで分散情報を送信する．

④全ノードはブロードキャストされた $m+1$ 個の分散情報に，自分が持つ W_{m+2} を加えた $k=m+2$ 個の分散情報からクラスタ鍵を復元する．

また，4.1 節の非対称秘密分散を用いる場合，全子ノードを分散情報 0 のサーバとして保存すべき分散情報を計算し，CH には W_{m+1}〜W_{m+u+3} を暗号化して送る．これを提案方式 5' とする．

5.2 安全性

(1) CH 解析に関する安全性

提案方式 3 では，CH は ID_i のリンク鍵に関する 2 つの分散情報を持つだけ（$k=m+1$，$m \geqq 2$）であり，提案方式 4，5 では $u+2$ 個（$0 \leqq u \leqq m-2$）の分散情報を持つだけであるので，リンク鍵を復元できず情報理論的安全性が実現される．提案方式 3'，4'，5' では 4.1 節に示される非対称秘密分散法を用いるので，擬似乱数生成器の安全性に依存した計算量的安全性を持つ．

(2) 子ノード解析に関する安全性

提案方式 3，4 では子ノードの数は m であり，$k=m+1$

であることから，すべての子ノードを解析しても ID_i に関するリンク鍵 L_i およびクラスタ鍵は漏洩せず，情報理論的安全性を持つ．また，提案方式 5 では子ノードは共通の W_{m+2} を持つが，$k = m + 2$ であるので同様に情報理論的安全性を持つ．提案方式 3' は ID_i が 2 つの分散情報を持つため，$m - 1$ 個までの子ノード解析に耐性を持つ．ただし，提案方式 3'，4'，5' は 4.1 節に示される非対称秘密分散法を用いるので計算量的安全性となる．

(3) CH と子ノード解析に関する安全性

提案方式 3 では CH は 2 個の分散情報を持つため，CH と $m - 2$ 個の子ノード解析まで情報理論的安全性を持つ．提案方式 4，5 では CH と $m - u - 2$ 個までの子ノード解析に情報理論的安全性を持つ．提案方式 3'，4'，5' は計算量的安全性となる．

(4) 通信路の盗聴に関する安全性

鍵共有フェーズに限れば，提案方式 3，4，5 では通信路に出る分散情報は $k - 1$ 個以下であるため盗聴に対して情報理論的安全性を持つ．提案方式 3'，4'，5' は計算量的安全性となる．

6. 評価

6.1 通信量・記憶量・計算量に関する評価

従来方式と提案方式 1〜3 に関して通信量，記憶量，計算量について比較を行う．提案方式 4 以降は提案方式 3 の変形であり，類推できるので省略する．

前提として，クラスタ内で CH が 1 個，子ノードが m 個とする．また，全子ノードが CH との鍵を共有し，1 つの子ノードが CH と暗号通信する場合を想定する．簡単のために ID と鍵の長さは同じとして L で表し，センシング情報の長さを H で表す．また，Setup phase において CH は全子ノードの ID を，子ノードは CH の ID をすでに知っているとする．また，1 ラウンド中 1 ノードは c 回通信を行うとする．ただし，BS は十分な電力や計算資源を持つため評価対象とせず，CH と子ノードのみ評価する．また，2.4 節に示した秘密分散法を用いる手法はべき乗演算や多項式計算などが必要で，べき乗演算に関する N は十分大きいため他と比べて明らかに非効率であるので，比較対象とはしない．よって，センサネットワークにおける代表的な鍵共有法である SecLEACH および MS-LEACH を対象として比較する．

(1) 通信量

鍵共有時（提案方式では分散フェーズ）と暗号通信時（提案方式では鍵共有・暗号通信フェーズ）における通信量を示す．ただし，提案方式 3，3' は暗号通信時に通信を行う当該子ノードと他ノードの動作が異なることから，全子ノードの通信の総和を示す（提案方式 3，3' 以外では，当該ノード以外はスリープしているため通信和で計算しても同じ）．

まず，SecLEACH では CH を含む各ノードが持つ鍵数を a，その中で CH と子ノードで一致する鍵数の平均を b とすると，鍵共有時では CH が a 個の鍵 ID を送信し，子ノードが b 個の鍵 ID を送信する．MS-LEACH では ID を既知とするため鍵共有時に何もしない．暗号通信時には両方式とも子ノードが ID とセンシング情報の暗号化情報を送信する．

表 1 通信量
Table 1 comparison to data traffic.

	鍵共有時		暗号通信時
	CH	子ノード	子ノード和
SecLEACH	aL	bL	$c(L+H)$
MS-LEACH	0	0	$c(L+H)$
提案方式 1	mL	0	$c(2L+H)$
提案方式 2	mL	0	$c(2L+H)$
提案方式 3	mL	0	$c\{(m+1)L+H\}$
提案方式 3'	mL	0	$c\{(m+1)L+H\}$

また，提案方式 1，2 では分散時に CH が BS に m 個の子ノード ID を送信する．暗号通信時は子ノードが ID，分散情報，暗号化情報を送信する．提案方式 3，3' では暗号通信時に当該ノードが ID を，他ノードが分散情報を順にブロードキャストした後，当該ノードが暗号化情報を送信する．

以上の結果を**表 1** に示す．通信量では MS-LEACH が最も良いが，$a > m$，かつ $b > c$ であれば提案方式 1，2 は SecLEACH より通信量が少ない．提案方式 3，3' は高い安全性と引換えに通信量は多いが，$H \gg L$ であれば H が支配的になり，その差分は小さくなっていく．

以上により，提案方式 1，2 は従来方式に比べ大きな消費電力の増加は生じないと考えられる．提案方式 3 以降の消費電力は全子ノードの解析に対する安全性を実現する分他方式に比べ大きいが，H が mL よりも十分大きければその差分は小さくなる．ただし，表 1 において鍵共有時における SecLEACH の aL および暗号通信時の提案方式 3，3' の $c(m+1)L$ はブロードキャストである．ブロードキャストがユニキャストより消費電力が大きい場合，SecLEACH と提案方式 3，3' の消費電力は増加する可能性がある．

(2) 記憶量

CH と子ノード 1 個あたりの記憶量を示す．ただし，LEACH では全ノードが CH と子ノードになるため，その大きい方が全ノードの記憶量となる．

SecLEACH は各ノードが a 個の鍵とその鍵 ID を記憶し，CH となるノードはさらにクラスタ内のノード ID とその鍵の記憶が必要である．また，MS-LEACH において各ノードは基本的に初期鍵と自身の ID のみを記憶すればよく，CH はクラスタ内のノード ID とその鍵の記憶が追加される．

提案方式 1 において各ノードは自身の ID，固有鍵，リン

表 2 記憶量
Table 2 comparison to amount of memory.

	子ノード	CH
SecLEACH	2*aL*	2*aL*+2*mL*
MS-LEACH	2*L*	2*L*+2*mL*
提案方式 1	4*L*	3*L*+2*mL*
提案方式 2	4*L*	3*L*
提案方式 3	2(1+*m*)*L*	(2+3*m*)*L*
提案方式 3’	4*L*	2*L*

表 3 計算量
Table 3 comparison to calculation.

	鍵共有時		暗号通信時	
	CH	子ノード	CH	子ノード和
SecLEACH	0	0	*cH*	*cH*
MS-LEACH	2*mL*	2*L*	*cH*	*cH*
提案方式 1	*mL*	*L*	*cH*	*cH*
提案方式 2	0	*L*	c(*L*+*H*)	*cH*
提案方式 3	2*mL*	*mL*	*cH*	*cH*
提案方式 3’	0	2*L*	*c*(*L*+*H*)	*c*(*mL*+*H*)

ク鍵，分散情報を保持し，CH は加えてクラスタ内のノード ID とその分散情報を記憶する．提案方式 2 の子ノードは提案方式 1 と同様であるが，CH は分散情報を持たない．提案方式 3 では子ノードは自身の ID，固有鍵に加えて m 個の子ノードの ID と分散情報を持ち，CH は全子ノードに対して ID と 2 個の分散情報を持つ．提案方式 3’ では子ノードは自分のリンク鍵に対する 2 つの分散情報を持ち，CH は分散情報を保存しない．

以上の結果を**表 2** に示す．4.1 節に示す非対称秘密分散法により提案方式 2，3’ は CH が全子ノードの分散情報を保存する必要がなく，大きな記憶量削減ができる．すなわち，記憶量では提案方式 2，3’ が最も少ない．

(3) 計算量

計算量として最も処理量が多い暗号化・復号を対象とする．ただし，提案方式 3’ は暗号通信時において当該ノードとそれ以外の子ノードで暗号化処理が異なるため，通信量のときと同様，子ノードの暗号化の総和で示す．

SecLEACH は鍵 ID を比較するのみであるので，その計算量は暗号化・復号に比べて無視できるため 0 とする．

MS-LEACH は，鍵共有のために 2 回の暗号化を必要とする．暗号通信時はセンシング情報の暗号化・復号を行う．

提案方式 1 は分散時に CH が BS から送られた m 個の分散情報を復号し，子ノードが自分の分散情報の復号を行う．提案方式 2 では分散時に CH は BS からの分散情報を復号しないが，暗号通信時に CH は通信のたびに ID を暗号する必要がある．提案方式 3 では分散時に子ノードは m 個の子ノードに対する分散情報を復号し，CH は m 個の子ノードに対して 2 つの分散情報の復号を行う．提案方式 3’ では分散時に各子ノードは 2 個の分散情報を復号するが，暗号通信時に当該ノード以外の子ノードは通信回数 c に応じた ID の暗号化処理を必要とする．ただし，全提案方式では Shamir の秘密分散法ではなく，XOR のみを用いる秘密分散法を用いているため，XOR の計算量は暗号化・復号に比べて無視できるので計算量を 0 とする．

以上を**表 3** に示す．計算量は SecLEACH が最も小さい．$c < m$ であれば提案方式 2 が，$m < c < 2m$ であれば提案方式 1 が次いで小さい．MS-LEACH の計算量は提案方式 1 より大きく，$c > 2m$ であれば提案方式 2 より小さい．次いで，提案方式 3 となる．

以上により，提案方式 1，2 は従来方式に比べ大きな消費電力の増加は生じないと考えられる．提案方式 3 以降の消費電力は全子ノードの解析に対する安全性を実現する分他方式に比べ大きいが，H が mL よりも十分大きければその差分は小さくなる．

6.2 拡張性

まず，新たな子ノードがクラスタに加わる場合を考える．この場合，子ノードはそのクラスタに属することを CH に伝え，CH はその子ノードの ID を BS に伝える．提案方式 1 では BS は CH とその子ノードに各分散情報を暗号化して送り，提案方式 2 では BS は子ノードにその分散情報を送る．提案方式 3 以降は子ノードの数が変わると，k も変わるので全分散情報を計算し直して，全ノードに再配布する．ただし，提案方式 3’，4’，5’ では分散情報 0 のノードの分散情報は同じとできるので，その子ノードにのみ再計算した分散情報を再配布すればよい．一方，子ノードがクラスタから抜ける場合，提案方式 1 では CH はその子ノードの分散情報を破棄し，提案方式 2 では何もしない．提案方式 3 以降では追加の場合と同様，分散情報の再計算と再配布が必要である．よって，提案方式 3 以降は子ノードの追加・破棄に鍵共有と同様の付加がかかる．そこで，提案方式 3 以降では子ノードが追加・破棄されたラウンドでは何もせず，次のラウンドでその子ノードを含んで対応することができれば，特別な処理を追加せずに対応できる．

また，鍵更新に関しては提案方式 1，2 において各ノードがリンク鍵を保持せず，BS が各ノードのリンク鍵を定めて，分散時に W_{i1} と L_i を子ノード ID_i に暗号化して送れば，ラウンドごとに鍵更新が可能である．提案方式 3 以降はラウンドごとに鍵更新が実行される．

また，一階層のクラスタツリーを例に説明したが，クラスタツリーが多階層になっていても，階層ごとに提案方式を繰り返せば対応できることは明らかである．特に，BS と直接通信できず他の CH などを介して通信を行うクラスタがあったとしても，BS から送られる分散情報は当該ノー

ドの固有鍵で暗号化されているため，複数のノードを介して転送されても問題ない．

7. まとめ

クラスタツリー型センサネットワークに秘密分散法を適用し5つの新しい鍵共有方式を提案した．用途に応じた使い分けが望まれる．各提案方式の特徴を，使い分けに関する例も加えて以下にまとめる．

提案方式1：子ノードごとに独立に $k = n = 2$ の秘密分散を適用して鍵を分散することにより，どれだけの子ノードが解析されても他の子ノードの鍵にまったく影響を与えず，情報理論的安全性を実現する．また，6章の評価にあるように，既存の鍵共有方式に比べ特に大きな負荷を必要とせずに情報理論的安全性が実現できる．また，既存のセンサノードはある程度の記憶容量を保持することから，提案方式1に必要な全子ノードの分散情報に対する記憶容量は許容範囲内と考えられる．よって，既存のセンサネットワークにセキュリティを導入しようとする場合，提案方式1が最も推奨される．

提案方式2：秘密分散法を非対称化し，提案方式1と同様の設定で秘密分散を適用する．これによって，保持する分散情報を0にするノードを作ることができ，記憶量などを大幅に削減できる．現在，IoTへのセキュリティ導入が1つの大きな課題となっており，IoTで用いられるIoTデバイスは一般に今までネットワークにつながれていなかった「モノ」に計算能力と通信能力を与えて，ネットワークの一部としたものであるため，その「モノ」に大きな計算資源や記憶資源を与えることは望まれない．IoTにセキュリティを導入するため，IoTデバイスが共通鍵暗号による暗号化機能を有するとするならば，それを有効利用するための鍵共有法として提案方式2はその鍵保存とXOR機能のみを持てばよく（擬似乱数生成は暗号化機能によって実現できる），ほぼ最小限の記憶量・計算量の追加で安全な暗号通信が実現できる．よって，提案方式2は今後開発されるIoTデバイスに対して推奨される．

提案方式3：$n = m+2$，$k = m+1$ として m 個の子ノードの各鍵をネットワーク全体に秘密分散する．これにより，暗号通信時以外ですべての子ノードが解析されてもまったく鍵が漏洩しない情報理論的安全性を実現する．よって，たとえば建物の各部分にセンサノードを配置しており，ある位置のセンサノードからの情報が他ノードの情報より重要である場合（センサノードが温度をセンシングする場合燃料庫の温度は他より重要であり，会話などをセンシングする場合重役室の会話は他より重要と考えられる），提案方式3はその位置にあるノードのみを解析しても鍵は漏洩せず，建物全部のノードを解析しなければならない．よって，提案方式3は子ノードからの情報の重要性にばらつきがある場合などに有効である．

提案方式4：$n = m+u+2$，$k = m+1$ $(0 \leqq u \leqq m-2)$ として，提案方式3において u 個のノードが欠損しても鍵共有を可能にする．用途は提案方式3と同様である．

提案方式5：$n = m+u+2$，$k = m+1$ $(0 \leqq u \leqq m-2)$ として，ネットワーク全体に共通の1つの鍵を分散する．1回の復元処理で全ノードが鍵を共有できる．よって，クラスタ内で共通の鍵（クラスタ鍵）が必要な場合に有効である．たとえば，子ノードとの通信が離散的でなく連続的に行われる場合，CHは1度生成したクラスタ鍵をその期間は保持して複数の子ノードと連続的に暗号通信した方が負荷が小さい．このような場合，提案方式5は1度［鍵共有フェーズ］を実行するだけでクラスタ内の全ノードがクラスタ鍵を共有できるため効率的である．

謝辞 本研究はJSPS科研費26420373の助成を受けたものです．

参考文献

[1] Heinzelman, W.R. et al.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks, *Proc. 33rd Hawaii International Conference on System Sciences*, Vol.8, p.8020 (2000).
[2] Heinzelman, W.B. et al.: An application-specific protocol architecture for wireless microsensor networks, *IEEE Trans. Wireless Communications*, Vol.1, No.4, pp.660–670 (2002).
[3] Oliveira, L.B. et al.: SecLEACH—On the Security of Clustered Sensor Networks, *Signal Processing*, Vol.87, No.12, pp.2882–2895 (2007).
[4] Qiang, T. et al.: MS-LEACH: A Routing Protocol Combining Multi-hop Transmissions and Single-hop Transmissions, *Pacific-Asia Conference on Circuits, Communications and Systems*, pp.107–110 (2009).
[5] Chan, H. et al.: Random Key Predistribution Schemes for Sensor Networks, *Proc. 2003 IEEE Symposium on Security and Privacy*, pp.197–213 (2003).
[6] Sencun Z. et al.: LEAP+: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks, *ACM Trans. Sensor Networks* (*TOSN*), Vol.2, pp.500–528 (2006).
[7] Shamir, A.: How to share a secret, *Comm. ACM*, Vol.22, pp.612–613 (1979).
[8] Yiying, Z. et al.: A Secret Sharing-Based Key Management in Hierarchical Wireless Sensor Network, *International Journal of Distributed Sensor Networks* (2013).
[9] Chunying, W. et al.: Key Management scheme based on secret sharing for Wireless Sensor Network, *2013 4th International Conference on Emerging Intelligent Data and Web Technologies*, Vol.7, pp.574–578 (2013).
[10] Ephremides, A. and Truong, T.V.: Scheduling broadcasts in multihop radio networks, *IEEE Trans. comm.*, Vol.38, pp.456–460 (1990).
[11] 金子 良，岩村惠市，金田北洋：大規模ネットワークに適したワイヤレスセンサネットワーク向け鍵共有方式の提案，電子情報通信学会論文誌 D，Vol.J98-D, pp.418–427 (2015).
[12] Kurihara, J. et al.: On a fast (k,n)-threshold secret sharing scheme, *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, Vol.E91-A, pp.2365–2378 (2008).

[13] 高橋 慧，小林史郎，岩村惠市：記憶容量削減と計算量的安全性および復元の独立性を実現するクラウドに適した秘密分散法，情報処理学会論文誌，Vol.54, pp.2146–2155 (2013).
[14] 山本博資：(k,L,n) しきい値秘密分散システム，電子通信学会論文誌，Vol.J-68-A, pp.945–952 (1986).
[15] Bertier, M. et al.: Low-cost secret sharing in sensor networks, *Proc. IEEE 12th International Symposium on High Assurance Systems Engineering* (*HASE'10*), pp.1–9 (2010).
[16] Claveirole, T. et al.: Securing wireless sensor networks against aggregator compromises, *IEEE Communications Magazine*, Vol.46, pp.134–141 (2008).
[17] Diop, A. et al.: Efficient Group Key Management using Symmetric Key and Threshold Cryptography for Cluster based Wireless Sensor Networks, *I.J. Computer Network and Information Security*, pp.9–18 (2014).
[18] Ruj, S. et al.: Pairwise and Triple Key Distribution in Wireless Sensor Networks with Applications, *IEEE Trans. Comput.*, Vol.62, pp.2224–2237 (2013).
[19] Seo, S.H. et al.: Effective Key Management in Dynamic Wireless Sensor Networks, *IEEE Trans. Information Forensics and Security*, Vol.10, pp.371–383 (2015).

岩村 惠市 （正会員）

1958 年生．1980 年九州大学工学部情報工学科卒業．1982 年同大大学院情報工学研究科修士課程修了．同年キヤノン（株）入社．1994 年東京大学博士（工学）．現在，東京理科大学工学部電気工学科教授．主に符号理論，並列処理，情報セキュリティ，電子透かしの研究に従事．IEEE，電子情報通信学会，情報ハイディングおよびその評価基準（IHC）研究会委員長．本会フェロー．

須賀 祐治

九州大学大学院数理学研究科修士課程修了．（財）九州システム情報技術研究所，電気機器メーカを経て 2008 年より株式会社インターネットイニシアティブにて季刊技術レポート IIR の執筆等，暗号と情報セキュリティ全般に関わる調査・研究活動に従事．筑波大学大学院システム情報工学研究科後期課程在籍．IPSJ CSEC 研究会幹事．IPSJ 論文誌ジャーナル/JIP 編集委員．CELLOS（暗号プロトコル評価技術コンソーシアム）幹事．IPSJ 2004 年度山下記念研究賞．CSS×2.0 in CSS2008/2010/2012 キャンドルスター 1 等星．AsiaJCIS 2012 Best Paper Award．IWSEC 2012 Best Poster Award．第 6 回インターネットと運用技術シンポジウム優秀プレゼンテーション賞．第 76 回全国大会大会優秀賞．IWSEC2015 Program co-chair．SSR2015 Genaral co-chair．

後藤 慎一

2015 年東京理科大学工学部電気工学科卒業．2015 年同大学大学院工学研究科電気工学専攻在学中．

金田 北洋 （正会員）

1984 年早稲田大学理工学部機械工学科卒業．1986 年同大学大学院修士課程修了．同年キヤノン株式会社入社．1995 年米国デューク大学大学院電気工学科修士課程修了．2010 年東京理科大学大学院理工学研究科博士課程修了，博士（工学）．現在，キヤノン株式会社アドバンスト IRT 開発センター，および大阪府立大学連携大学院客員教授，同文書解析・知識科学研究所客員研究員．主に文書画像解析/認識，言語処理，ビッグデータ解析の研究・製品開発に従事．画像電子学会会員．