Vol.2010-DPS-142 No.27
Vol.2010-CSEC-48 No.27
2010/3/5

ワイヤレスセンサネットワークにおける グループ鍵分配プロトコルの考察

村 上 大 樹[†1]　双 紙 正 和[†1]

ワイヤレスセンサネットワークにおけるグループ通信においては省電力技術が重要である．しかし，一般のネットワークで用いられる手法を用いグループ鍵を分配するのでは，効率的とは言えない．これに対し，本論文ではワイヤレスセンサネットワークで用いられる事前鍵分配プロトコルを前提としたグループ鍵分配手法を提案し，考察する．これにより，グループ鍵分配のコストを大幅に抑えつつ，効率的にグループ鍵分配を行うことが出来るようになった．また，ノード捕縛攻撃を受けた場合でも，効率的に再分配を行えることを確認した．

Consideration of a Group Key Distribution Protocol in Wireless Sensor Networks

DAIKI MURAKAMI[†1] and MASAKAZU SOSHI[†1]

The power saving technique is critical for group communication in wireless sensor networks. However, it is not efficient to use the technique for general computer networks and to distribute the group key. In this paper, we propose the group key distribution scheme based on a key pre-distribution scheme used in wireless sensor networks. We show that the proposed group key distribution is efficient and robust against node capture attacks.

1. はじめに

近年ワイヤレスセンサネットワーク（Wireless Sensor Networks，以後 WSNs）が注目されている．WSNs とは，ある種のセンサ（温度センサ，光センサなど）と無線通信機能を持つ安価な機器（以後ノード）を多数用いることによって，広範囲の情報収集などを行う無線ネットワークである．ほとんどの場合，ノードはバッテリで駆動し，自律的にネットワークを構築する．

WSNs もネットワークであるためセキュリティが重要な課題となっている．しかし，WSNs ではノードはリソース（計算能力，電力，メモリ等）が制限されているため，公開暗号方式を用いることが難しい．上記の問題に対する鍵共有方式として，Eschenauer 等が鍵を事前に分配する方法 (以後 EG プロトコル)[3] を提案した．この方法はあらかじめ鍵を大量に生成し，各ノードにその部分集合を割り当てることによって，ノード同士が確率的に鍵を共有できるようにしたものである．しかし，ノードを物理的に見つけ解析することで鍵を取得し，元の鍵の集合を復元しようとするノード捕縛攻撃に比較的弱い．

WSNs の使用用途として，環境や自然動物の調査等に用いられ，そのため長時間の動作を期待されることが多い．しかし，バッテリ駆動のノードの電力は有限であるため，稼動時間を延ばすための省電力技術が研究されている．WSNs は無線ネットワークであるため，全ての通信がブロードキャストになる．そのため 1 対 1 の通信を N 回繰り返すよりも，1 対 N の通信を 1 回行うほうが省電力である．セキュリティを保ちつつ 1 対 N 通信を行う手法に，グループで鍵を共有しグループ内でブロードキャストを行う方法がある．しかし，グループ鍵を共有するのに 1 対 1 通信を N 回行う必要がある．これに対し一度のブロードキャストを行うことで，メンバーがお互いの秘密鍵を秘密にしつつ，一度の通信で同じ鍵を共有することのできる方式がある[6),7)]．しかし WSNs に特化された手法とはいえないため，効率的とは言い難い．

本論文では，EG プロトコルを前提としたグループ鍵分配プロトコルを提案し考察する．EG プロトコルを前提としたグループ鍵分配プロトコルでは，従来手法に比べ少ないデータ送信量でグループ鍵を共有することができ，ノードのエネルギー消費を抑えることができるようになった．

本論文は以下の用に構成される．2 章は，関連研究の紹介を行う．3 章は，本研究で用いる仮定の説明を行う．4 章は，提案手法の紹介を行う．5 章は，提案手法に対し，評価実験を行う．6 章は，考察を行う．7 章は，まとめとなる．

2. 関連研究

一般のネットワークで鍵共有を行う場合，公開鍵暗号方式が用いられる．公開鍵暗号方式とは，ある種の数学的問題の困難性に基づく暗号方式であり，暗号化する鍵（公開鍵）と復

†1 広島市立大学
Hiroshima City University

Vol.2010-DPS-142 No.27
Vol.2010-CSEC-48 No.27
2010/3/5

号する鍵（秘密鍵）が異なる[2]．公開鍵暗号方式を用いる利点は，公開鍵と秘密鍵が異なるため，公開鍵を公開しても安全性に問題が無い点である．このため秘密鍵暗号方式と違い，相手が変わるごとに異なる鍵を事前に用意しなくて良い．しかし，秘密鍵暗号方式と違い，暗号化及び復号するための計算コストが非常に大きい．そのため，WSNs のノードの制限されたリソースでは公開鍵暗号方式を用いることが難しく，多くの場合共通鍵暗号方式が用いられている．共通鍵暗号方式を用いる場合事前に鍵を共有する必要がある．その方法として，安直な方法としては次のような方法が考えられる．1 つ目は全体で同じ鍵を用いる方法である．このようにすることにより，通信情報を外部から見えないようにしつつ効率的に通信を行うことができる．しかし，何らかの理由で鍵が攻撃者に知られてしまうと，攻撃者が全ての通信を盗聴することができてしまうという問題点がある．この場合は，ノードを回収しない限り，新しい鍵を攻撃者に秘密にしたまま鍵を変更することができない．2 つ目は，n 個のノードで完全グラフを作り全ての辺にユニークな鍵を割り当て，各ノードにそのノードと関係する $n-1$ 個の辺それぞれに割り当てられている鍵全てを持たせる方式である．こうすることにより，ノード同士の物理的な位置関係に影響されずに，すべてのノード同士が秘密の 1 対 1 通信を行うことができる．また，何らかの理由により攻撃者に鍵を知られてしまった場合でも，その鍵が割り当てられている辺の両端のノード以外に影響は一切ない．問題点は，ノードの数が増えるとノードの持つ鍵の数が膨大になる点である．WSNs ではノードの数が数百から数十万まで想定されている．従って，各ノードが必要な鍵全てを保持することができないと，実現不可能である．しかし，現在の WSNs のリソースでは実現することは難しい[5]．

これらに対し，Eschenauer と D. Gligor が WSNs 向けの鍵共有プロトコルを提案した[3]．このプロトコルは，WSNs において多くの鍵共有プロトコルの基礎となっている．

EG プロトコルは事前鍵分配段階，直接鍵設立段階，経路鍵設立段階の 3 つに分かれている．事前鍵分配段階は，ノードが目的のエリア（森や被災地など）に配置される前に行い，直接鍵設立段階と経路鍵設立段階は，ノードが目的のエリアに配置された後ノード自身が自律的に行う．事前鍵分配段階では初めに無作為に鍵を K_{pool} 個作り，これらの鍵の集合を鍵プールとする．次に，各ノードは鍵プールから大きさが K_{ring} となる部分集合を無作為に選び，自身のメモリに保存する．直接鍵設立段階では直接暗号通信が可能かどうかを確認するために行う．初めに各ノードは無線通信ができる範囲に他のノードがいるかどうかを確認する．無線通信ができる範囲に他のノードがいた場合，お互いが同じ鍵を持っているかどうかを確認する（確認方法はチャレンジレスポンス方式[1] など）．確認した結果，お互いのノードの持っている鍵の同じものが 1 個以上あった場合，その中から鍵を 1 つ選び暗号通信を開始する．そうでない場合，経路鍵設立段階へ進む．経路鍵設立段階は，直接鍵設立段階で同じ鍵が 1 個もなかったノード同士が行う．ノード $A(Node_A)$ とノード $B(Node_B)$ が鍵を共有できなかったとする．初めに，$Node_A$ は $Node_B$ と直接鍵を共有しているかを，直接通信可能な他のノード全てに対して問い合わせる．問い合わせを受けたノードは，$Node_B$ と直接鍵を共有していた場合経路の情報を返す．問い合わせを受けたノードが $Node_B$ と直接鍵を共有していなかった場合，$Node_B$ と直接鍵を共有しているかを直接通信可能な他のノード全てに対して問い合わせる．これを繰り返した結果迂回路が存在した場合，その経路を用いて新しい鍵を無作為に生成し暗号通信を開始する．迂回路が存在しなかった場合，$Node_A$ と $Node_B$ とは暗号通信を行わない．

一方，WSNs においてデータの送信に必要なコストを少なくするために，グループ鍵を共有するという手法がとられている．特に WSNs においては実質的に全ての通信がブロードキャストになるため，用いられることが望ましい．グループ鍵を共有しグループ通信を行う利点は，グループ内であれば全ての送信をブロードキャストで行える点である．これにより，個別通信を複数回行うより効率的にデータの送受信を行うことができる．

前に述べたとおり，WSNs ではグループ鍵を個別通信で配るのではなく，一度の通信で全てのメンバーへ鍵を送信することができるのが望ましい．そこで，グループ鍵を一度に分配する手法がいくつか提案されている．

Zheng 等が中国の剰余定理を用いたグループ鍵共有方法を提案した[7]．この手法は，中国の剰余定理を用いることによってグループ鍵を一度の通信で共有しようという手法である．これにより，送信ノードは 1 つの値を送るだけで他のノードの秘密鍵は判らないまま複数のノードに同じ鍵を配ることができる．しかし中国の剰余定理の性質より，メンバーの数が多いほど受信者の行う処理が複雑になってしまう．

Wu, K.P 等が各ノードと共有している鍵を分配したいグループ鍵をマスクするために用いる手法が提案された[6]．これにより，メンバーが増えた場合においても受信者の行う処理が Zheng 等の方法よりも少なくて良い．

3. 仮　　定

ここでは本論文で用いる用語の定義と，仮定する WSNs のモデルを示す．

3.1 WSNs のモデル

ここでは本論文において前提とする WSNs のモデルを定義する．

Vol.2010-DPS-142 No.27
Vol.2010-CSEC-48 No.27
2010/3/5

・基地局

ノードよりはるかに高い性能を持つとし，消費電力を無視してよいほど大きな電源を持つとする．基地局は外部の有線ネットワークに繋がっており，ユーザが直接操作できる．各ノードの持つ情報（鍵など）を全て持っており，ノード捕縛攻撃の対象にならないものとする．

・ノードノードはバッテリで駆動し，代替電源（太陽電池など）は持たないものとする．全てのノードは自力で物理的な移動が出来ないとし，外的要因によって移動することもないとする．ノードはユニークな ID を割り当てられる．

3.2 用語の定義

本論文で用いる各種用語とその意味を表 3.2 に示す．

表 1 用語の定義

鍵プール	EG プロトコルにおいて生成される鍵全体の集合
鍵リング	各ノードの持つ鍵の集合，鍵プールの部分集合
K_{pool}	鍵プールの大きさ
K_{ring}	各鍵リングの大きさ
$SKey$	あるノード同士が共有している鍵
基地局	1 つの WSNs に少数存在する高性能な端末
ノード	1 つの WSNs に多数存在する性能の限られた端末
$Node_A$	WSNs に存在する A という名前のノード
隣接ノード	あるノードと直接無線通信可能な他のノード

4. 提 案 手 法

この章では，EG プロトコルを用いた WSNs を仮定し，その上で使うことのできるグループ鍵分配プロトコルを提案する．初めに，提案手法で用いる考え方を説明する．その後，提案手法の説明を行う．

4.1 基本的なアイデア

ここでは，提案手法で用いる基本的なアイデアを説明する．

4.1.1 仮　　定

初めに，基本的なアイデアで用いる仮定を述べる．

WSNs ではなく，少数のセンターと多数の端末が存在する無線ネットワークを仮定する．センターと各端末は電力制限が無いものとする．p を暗号に使うことのできる大きな素数とする．全体で N 個の端末が存在し，センター D は他の端末 N_i との共通鍵 K_i を全て知っているとする（$i = 1, 2, \cdots, N, K_i \in Z_p^*$）．また，$K_i \neq K_j (i \neq j)$ とする（$i, j = 1, 2, \cdots, N$）．

4.1.2 基本的なアイデアのプロトコル

センター D は N 個の端末のうち m 個（$N_1, N_2, \cdots, N_m$ とする）にグループ鍵の分配を行いたいとする．

初めに，センター D はグループ鍵 α を無作為に生成する（$\alpha \in Z_p^*$）．次に，以下の多項式 $g(x) \in Z_p[x]$ を計算する．

$$g(x) = f(x) + \alpha \tag{1}$$

ただし，

$$f(x) = (x - K_1)(x - K_2) \cdots (x - K_m) \tag{2}$$

である．ここで，式 1 を展開すると

$$g(x) = x^m + \sum_{i=0}^{m-1} a_i x^i \tag{3}$$

と書き直すことができる．特に，

$$a_0 = (-1)^m \prod_{i=1}^{m} K_i + \alpha \tag{4}$$

である．

これらを元に，センター D は式 3 で得られた値 $a_0, a_1, \cdots, a_{m-1}$ をブロードキャストする（$g(x)$ はモニックであるので，$a_m (= 1)$ は省略できる）．各端末 N_i は，値 $a_0, a_1, \ldots a_{m-1}$ を受信した後，それらから $g(x)$ を作り $g(K_i) = \alpha$ を計算しグループ鍵を復元する $(i = 1, 2, \cdots, m)$.

以上が基本的なアイデアを用いた場合のグループ鍵分配方式である．

4.1.3 基本的なアイデアのプロトコルの考察

基本的なアイデアのプロトコルの安全性と効率について考察する．

・安全性の評価

次の 2 つの場合を考える．

- グループ内の複数の端末が共謀した場合
- グループ外の攻撃者が $a_0, a_1, \cdots, a_{m-1}$ を傍受した場合

・グループ内の複数の端末が共謀した場合

グループ内の複数の端末が共謀し，特定のノードの持つ秘密鍵を暴露しようとする攻撃を

考える．

グループ内からの攻撃なので，攻撃者に $f(x)$ は知られている．グループ全体で m 個の端末があるとする．その中で i 個の端末が共謀した場合，$f(x)$ の次数を i 個減らすことができる．その結果，ある程度の人数が集まれば残りの共通鍵が暴露されてしまう（簡単に連立方程式を立てることができるので，$m-i \leq 2$ である場合は明らかに可能）．そのため，ある程度の人数が集まることができれば，特定のノードの持つ秘密鍵を計算できる．これにより，共謀攻撃に弱いといえる．

・グループ外の攻撃者が $a_0, a_1, \cdots, a_{m-1}$ を傍受した場合

悪意のある攻撃者が $a_0, a_1, \cdots, a_{m-1}$ を傍受し，グループ鍵 α の復元を試みるとする．攻撃者が $a_0, a_1, \cdots, a_{m-1}$ から α を復元する方法として，次の方法が考えられる．

1 つ目は，a_0 から α を復元する方法である．この方法の場合，a_0 から α を推測することが困難なので，仮の α' を決めその上で $g(x)-\alpha'$ を因数分解する必要がある．ここで，有限体上の多項式の因数分解はいろいろな方法がある[4]．しかし，因数分解を行わなければ α' が正しいかどうかは判別できないため，素数 p が十分に大きければ困難であると考えられる．

2 つ目として，攻撃者が $f(x)$ 自体を推測する攻撃が考えられる．$f(x)$ と $g(x)$ の次数が同じであるため，$f(x)$ は m 個に因数分解できることはすぐに分かる．また，因数定理より $f(x)$ は 0 を m 回通るので，$g(x)$ は α を m 回返すことになる．これにより，m 回出現した値を α とし，盗聴を試みることができる．しかし，この方法を行うためには $g(x)$ に対し総当りで検索を行う必要があり，α を総当りする場合と比べて計算量が減ったとは言い難い．そのため，p が十分に大きい場合においては非現実的な時間が必要になる．

上記より，グループ内の攻撃には多少弱いが，外からの攻撃には十分な耐性を持つと考えられる．

・効率の評価

m 個の端末と 1 度の通信で同じグループ鍵を共有するためには，m 個の値 $a_0, a_1, \ldots, a_{m-1}$ をブロードキャストする必要がある（a_m は常に 1 なのでここでは省略）．このため，通信コストは $m\log_2(p)$ となる．しかし，このコストは 1 対 1 で鍵を暗号化して m 回送る場合とほぼ同じであり，あまり効率が良いとは言えない．

4.2 提案するグループ鍵分配プロトコル

ここでは基本的なアイデアを元に，EG プロトコルで鍵分配が行われているという限定的な環境下でのグループ分配プロトコルを説明する．

基本的なアイディアでは，$K_i \neq K_j\ (i \neq j)$ という仮定を置いた．この時，一部の鍵が同じであるという仮定をおくことができるのであれば，生成する多項式 $f(x)$ の次数を減らすことができる．この仮定が成り立つと，グループ鍵を分配するために必要な通信コストを抑えることができる．しかし，この仮定は複数のユーザが同じ鍵を持っていることを意味するので，望ましい状況とは言えない．

これに対し，EG プロトコルにおいてはプロトコルの性質から上記の仮定が容易に成り立つ．従って，EG プロトコルを用いている環境であれば鍵の重複が多く存在しているため，グループ鍵を分配するための通信コストを減らすことができると考える．

4.2.1 仮　　定

ここでは，提案するグループ鍵分配プロトコルで用いる仮定を説明する．

WSNs を想定し，各ノードは EG プロトコルによって鍵の分配されているとする．基地局は各ノードの鍵リングにある鍵を全て知っているものとする．

4.2.2 提案プロトコル

p を暗号に使うことのできる大きな素数とし，ネットワークに合計で N 個のノードがいるとする．基地局は m 個のノードとグループ鍵を共有したいとする．

初めに，基地局はグループ鍵 α を適当に選び，乱数 r を生成する（$r, \alpha \in Z_p^*$）．次に，基地局は m 個のノード全てを網羅するできるだけ少ない鍵の組み合わせ $K_1, K_2, \ldots, K_l$ を探す（$K_i \in Z_p^*, i = 1, 2, \cdots, l$）．そして，基地局 $K_1, K_2, \ldots, K_l$ を元に

$$f(x) = (x - h(r, K_1))(x - h(r, K_2)) \cdots (x - h(r, K_l)) \tag{5}$$

と置く（$h(a, b)$ は a と b のを引数とするハッシュ関数．$h(a, b) \in z_p^*$）．その後，基本的なアイデアと同様に

$$g(x) = f(x) + \alpha \tag{6}$$

とし，式 6 を展開しまとめて

$$g(x) = x^m + \sum_{i=0}^{m-1} a_i x^i \tag{7}$$

とする（基本的なアイデアのままでは脆弱であるためと，乱数とハッシュ関数を用いている[8]）．そして基地局は，値 $r, a_0, a_1, \ldots, a_{l-1}, E_\alpha(M)$ をブロードキャストする（$E_\alpha(M)$ は，M を値 α を用いて暗号化したデータ．M はあらかじめ決められたメッセージ，hello など）．

データを受信したノードは，値 $r, a_0, a_1, \ldots, a_{l-1}, E_\alpha(M)$ から $g(x)$ を構築する．次に，自分の持っている鍵 $K_{i,1}, K_{i,2}, \ldots, K_{i,K_{ring}}$ と $g(x)$ を用い，値 $\alpha_1, \alpha_2, \ldots, \alpha_{K_{ring}}$ を得

る．そして，α_i で $E_\alpha(M)$ に対し順次復号を試み，M が得られたとき $\alpha_i = \alpha$ とする（$i = 1, 2, \cdots, K_{ring}$）．

4.2.3 理論的評価

実験を行う前に，提案するグループ鍵分配プロトコルの安全性の評価を行う．また，効率についても簡易的な評価を行う．

・安全性

次の 2 つの場合を考える．

- グループ内の複数のメンバーが共謀した場合
- グループ外の攻撃者が $r, a_0, a_1, \cdots, a_{m-1}$ を傍受した場合

・グループ内のノードが共謀した場合

基本的なアイデアのプロトコルの場合と同様に，グループ内の複数の端末が共謀し，特定のノードの持つ秘密鍵を暴露しようとする攻撃を考える．

グループ全体で m 個のノードがあるとする．グループ内からの攻撃なので，攻撃者に $f(x)$ は知られている．そのため，各ノードは，それぞれがグループ鍵 α を復元することができた鍵を持ち寄ることによって，$f(x)$ の次数を減らすことができる．ただし，持ち寄った鍵の一部が重複する可能性があるため，必ずしも次数が減るとは限らない．しかし，ある程度の人数が集まれば残りの共通鍵が暴露されてしまう可能性がある（簡単に連立方程式を立てることができるので，$m - i \leq 2$ である場合は明らかに可能）．これにより，ある程度の人数が集まることができれば，特定のノードの持つ秘密鍵を計算できる．ただし，どの程度のノードが集まる必要があるかについては不確定である（ただし，$m - 1$ 個のノードが集まれば確実に可能）．従って，グループ内のノードが共謀した攻撃については弱いといえる．

・WSNs 外の攻撃者が $r, a_0, a_1, \cdots, a_{m-1}$ を傍受した場合

グループ外の攻撃者が，$r, a_0, a_1, \cdots, a_{m-1}$ からグループ鍵 α を復元を試みるとする．ここで，攻撃者がノード捕縛攻撃を行っている場合と行っていない場合が考える．

初めに，攻撃者がノード捕縛攻撃を行っている場合を考える．この場合ノードはいくつかの鍵を得ているため，攻撃者は $r, a_0, a_1, \cdots, a_{m-1}$ から α を復元できる可能性がある．このため，攻撃者はノード捕縛攻撃を続けるほど α を得る可能性が高くなる．この時，センターがどのノードが捕縛攻撃を受けたかの情報を得ていれば，捕縛されたノード群の持つ鍵全てを用いずに多項式を生成することができる．このように行うことによって攻撃者に読まれること無くグループ鍵を分配することができる．ただし，多項式の次数が大きくなる可能性がある．従って，センターが適切な鍵の選び方を行えば，安全であると考えられる．

次に，攻撃者がノード捕縛攻撃を行っていない場合を考える．この場合，攻撃者は $r, a_0, a_1, \cdots, a_{m-1}, E_\alpha(M)$ から α を復元する必要がある．しかし基本的なアイデアのプロトコルの場合と同様に，p が非常に大きい場合 a_0 から α を求めるのは困難である．また，$E_\alpha(M)$ から α を求めようとすることもできる．これについては，適切な暗号アルゴリズムを用いていれば，鍵を総当りするコストとほぼ等価になるので安全であると考えられる．

5. 評 価 実 験

このセクションでは，提案したグループ鍵分配プロトコルの評価を行うために各種シミュレーションを行う．

5.1 実験の目的

提案したグループ鍵分配プロトコルは EG プロトコルを前提としている．そのため，乱数要素が多く定量的に評価することが難しい．これに対し，シミュレーションを行うことによって，提案したグループ鍵共有方法の有効性を評価することを目的とする．

5.2 実 験 環 境

ここではシミュレーションを行う機器の詳細等を示す．

今回シミュレータで用いるパラメータの意味を表 2 に示す．

表 2 シミュレーションパラメータの説明

$NumberOfNodes$	シミュレーション環境に存在するノードの総数
$KeyPoolSize$	EG プロトコルにおいて，作成する鍵の種類の総数
$KeyRingSize$	EG プロトコルにおいて，各ノードが持つ鍵の数
$Repeat$	試行回数，実験結果はこの回数行った平均を出力
$NumberOfTargetNodes$	グループ鍵を送りたいノードの数

5.3 実験 1：鍵プールを固定した場合における多項式への影響の評価

ここでは，EG プロトコルにおける鍵プールの大きさを固定した上で，各ノードの持つ鍵リングの大きさを変化させる．これにより，提案グループ鍵分配プロトコルおいて生成する次数がどの程度減少するかを実験，評価する．

5.3.1 実 験 手 法

実験手順は次の通りとなる．手順 1）ノードを $NumberOfNodes$ 個生成し，鍵を $KeyPoolSize$ 個生成する．手順 2）EG プロトコルと同様に鍵を各ノードに配布する．手順 3）$NumberOfTargetNodes$ 個のノードを選び，網羅するために必要な鍵の数を求める．

Vol.2010-DPS-142 No.27
Vol.2010-CSEC-48 No.27
2010/3/5

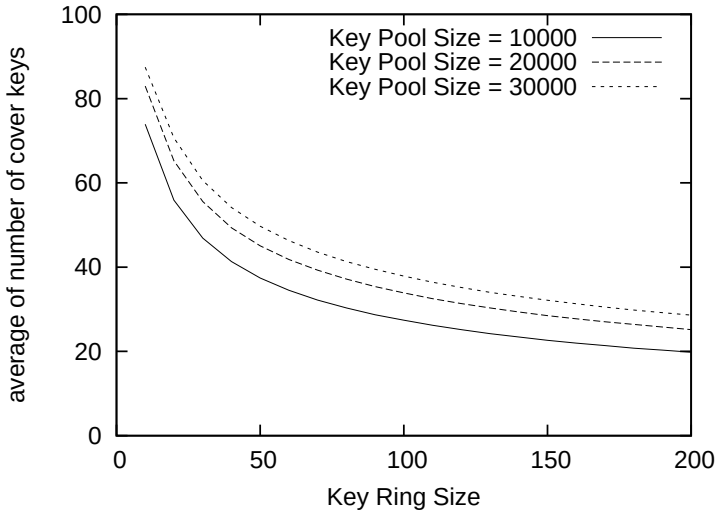

図 1 実験 1 のシミュレーション結果

手順 4）手順 1〜3 を $Repeat$ 回繰り返し，その平均を出力する．

また，パラメータは表 3 に示す．

表 3 実験 1 のパラメータ

$NumberOfNodes$	1000
$KeyPoolSize$	10000〜30000
$KeyRingSize$	10〜200
$Repeat$	2000
$NumberOfTargetNodes$	100

5.3.2 実 験 結 果

シミュレーションの結果を図 1 に示す．

5.3.3 実験 1 の考察

実験 1 に対する考察を行う．

図 1 を見ると，次の事が分かる．

- 鍵リングが大きくなるほど，網羅するのに必要な鍵の数が減少している．
- 鍵プールが多いほど，網羅するのに必要な鍵の数が多い．

鍵プールの大きさによらず，鍵リングが大きいほど次数の大きさが小さくなっている．これは，ノード同士の接続確率が高いほど次数が小さくなっていると考えることができる．また，鍵リングがある程度小さい場合においても，次数の大きさを半分程度に抑えることができている．WSNs においては，演算コストより通信コストのほうが非常に大きい[5]．このため，必要な次数を抑えることができることはエネルギー消費を抑えるのに非常に有効であると考えることができる．鍵リングが小さくノード同士の接続確率が小さい場合においても本プロトコルは有用であると考えることができる．また，鍵リングを大きくするほど効果を高めることができるため，よりエネルギーの消費を抑えることができると考えられる．

5.4 実験２：ノードの取り消しが及ぼす影響の評価

ノードの取り消し（Revocation）とは，EG プロトコルにおいて，あるノードが持つ鍵をネットワーク全体から削除する操作である．これにより，そのノードが他のノードと鍵共有できなくなり，ネットワークから除外されることになる．この操作によってノードを除外することは，ノード捕縛攻撃に対する対策である．

本プロトコルにおいても，攻撃者がノード捕縛攻撃によって得ることのできた可能性のある鍵を除外することにより，攻撃者がグループ鍵を求めることができなくするように多項式を生成することができる．そこで，本プロトコルに対するノードの取り消しが及ぼす影響を実験を行い評価する．

実験は，2 つ行う．1 つ目は，ノード同士の接続確率を固定し鍵リング（及び鍵プール）を変化させた場合，一定数のノードの取り消しが及ぼす影響の評価実験．2 つ目は，取り消すノードを変化させた場合の本プロトコルへの影響の評価実験．

5.4.1 実 験 手 法

実験手順は次の通りとなる．手順 1）ノードを $NumberOfNodes$ 個生成し，鍵を $KeyPoolSize$ 個生成する．手順 2）EG プロトコルと同様に鍵を各ノードに配布する．手順 3）$NumberOfTargetNodes$ 個のノードを選び網羅するために必要な鍵の数を求め，これを $Before$ の値とする．手順 4）全ノードの内，無作為に $NumberOfRevocationNodes$ 個選び，それらのノードが 1 個以上持つ鍵を全てのノードから削除する．手順 5）$NumberOfTargetNodes$ 個のノードを選び網羅するために必要な鍵の数を求め，これを $After$ の値とする．手順 6）手順 1〜5 を $Repeat$ 回繰り返し，その平均を出力する．

パラメータは表 4 と表 5 に示す．

Vol.2010-DPS-142 No.27
Vol.2010-CSEC-48 No.27
2010/3/5

表 4 実験 3-1 のパラメータ

$NumberOfNodes$	1000
$KeyPoolSize$	$KeyRingSize$ に対応する値
$KeyRingSize$	25〜500
$Repeat$	2000
$NumberOfTargetNodes$	100
$NumberOfRevocationNodes$	100

表 5 実験 3-2 のパラメータ

$NumberOfNodes$	1000
$KeyPoolSize$	$KeyRingSize$ に対応する値
$KeyRingSize$	125
$Repeat$	2000
$NumberOfTargetNodes$	100
$NumberOfRevocationNodes$	20〜400

5.4.2 実 験 結 果

実験 2-1 のシミュレーションの結果を図 2 に示す．実験 2-2 のシミュレーションの結果を図 3 に示す.

5.4.3 実験２の考察

・実験 2-1 の考察

実験結果を見ると，鍵リングが小さい場合ノード捕縛攻撃の影響が非常に大きく出ている．そして，鍵リングが大きくなるにつれて影響が小さくなっている．これは，ノード同士の接続確率が一定であれば，鍵プールが小さいほど攻撃者はノード捕縛攻撃によって鍵プールを推測しやすいことを示している．ただし，鍵プールが小さいほど本プロトコルの効果が高くなっている．これらより，ノード捕縛攻撃を仮定する状況では，ある程度の鍵プールの大きさが無ければ本プロトコルの効果は薄くなってしまう．しかし鍵プールを大きくしても，緩やかではあるが本プロトコルの効果が薄くなっている．変わりに，ノード捕縛攻撃の影響が徐々に小さくなっている．従って，どの程度のノード捕縛攻撃を想定するかによって，本プロトコルでどの程度の効果を得ることができるかが大きく変わってしまう．そのため，ノード捕縛攻撃がどの程度行われるかを適切に判断するかが重要であると考えられる．

・実験 2-2 の考察

実験結果より，ノード同士の接続確率が高いほどノード捕縛攻撃の影響が強く出ている．しかし，ノード同士の接続確率をある程度まで下げると，ノード捕縛攻撃の影響が少なくなっている．これらより，ノード捕縛攻撃が少ないと考えられる場合は，ノード同士の接続

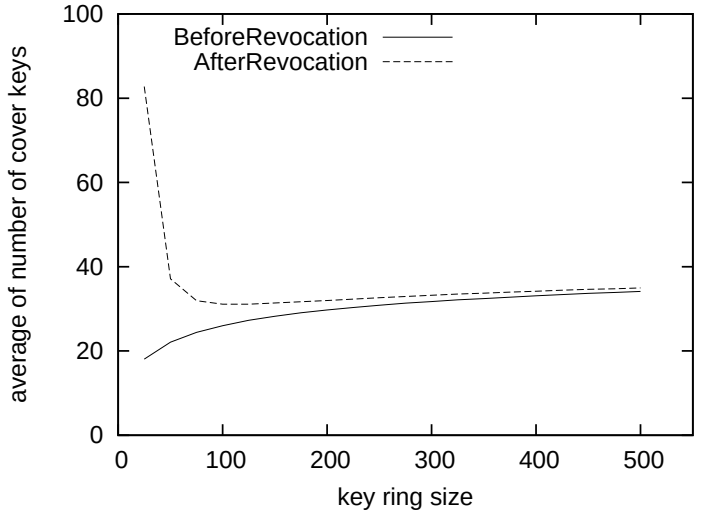

図 2 $p_{connect} = 0.69$ の場合における実験 3-1 の結果

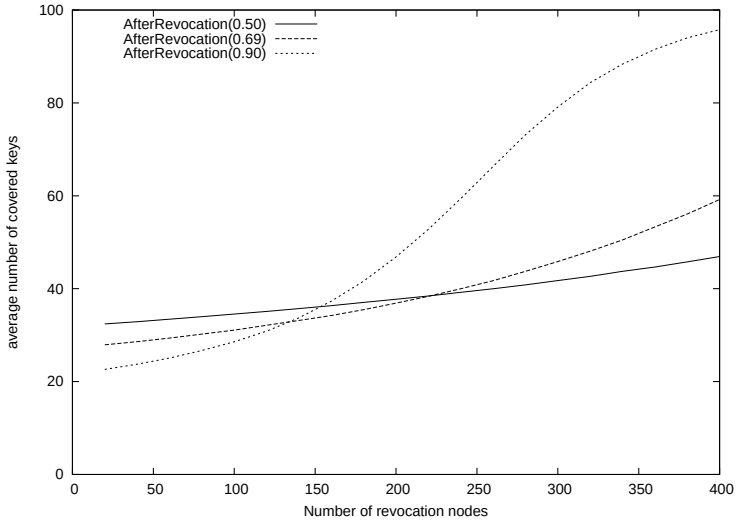

図 3 実験 3-2 のシミュレーション結果

Vol.2010-DPS-142 No.27
Vol.2010-CSEC-48 No.27
2010/3/5

確率を高くするほど本プロトコルの効果が高いと考えられる．またノード捕縛攻撃が多いと考えられる場合は，ノード同士の接続確率が低いほど良いと考えられる．しかし，ノード同士の接続確率を下げるとEGプロトコルで鍵共有ができなくなる可能性が高くなる．そのため，ノード捕縛攻撃の多さを適切に予測することが重要であると考えられる．

6. 比　　較

ここでは，本プロトコルと中国の剰余定理を用いたグループ鍵分配プロトコル[7]の比較を行い考察する．

中国の剰余定理を用いたグループ鍵分配プロトコルとは，各ノードと共有している鍵を整数とみなし，中国の剰余定理（Chinese remainder theorem）という数学的性質を用いる事で，1度の送信で複数の相手とグループ鍵共有を行おうという手法である．このプロトコルの利点は，グループ鍵共有を行うために送信する値が1個でよいという点である．欠点は，送信する値のビット長が，各ノードの秘密鍵のビット長の総和にほぼ等しくなるという点である（例えば，10個のノードと一度に共有したい場合は，値のビット長がそれぞれの約10倍になる）．

コンピュータ上で大きな値を計算する場合，値のビット長が大きくなるほど計算コストが増大する．また，中国の剰余定理を用いる場合有限体上の剰余を求める必要があるため，値のビット長が大きくなると計算コストが非常に大きくなる．そのため，WSNsのノードで計算を行うのは困難であると考えられる．これに対し，本プロトコルでは値の数が多いものの，それぞれの値のビット長は秘密鍵と同等である．メンバーが増加した場合においても，送信される値の数が増大するだけであり，それぞれのビット長に変化は無い．従って，本プロトコルのほうが計算コストが少ないと考えられる．

7. ま と め

EGプロトコルによって鍵共有がなされているWSNsという限定環境化に特化した，グループ鍵分配プロトコルの提案を行った．EGプロトコルでは鍵の重複が起こりやすいという性質を利用し，データの送信量を大幅に抑えることに成功した．また，ノード捕縛攻撃を適切に仮定することができれば，終始データの送信量を抑えたままグループ鍵分配を続けることができることを確認した．

提案手法で網羅する鍵を検索する場合，近似は容易に求まるが最適解を求めるのは困難である．このため，より効率的に送信を行うために，網羅する鍵を効率的に検索することが今後の課題となる．

Acknowledgment

本研究は科研費（21240001，20500075，20300003）の助成を受けたものである．

参 考 文 献

1) D.Chaum and H.VanAntwerpen. Undeniable signatures. In *Crypto*, Vol.89, pp. 286–299. Springer, 1989.
2) W.Diffie and M.Hellman. New directions in cryptography. *IEEE Transactions on information Theory*, Vol.22, No.6, pp. 644–654, 1976.
3) L.Eschenauer and V.D. Gligor. A key-management scheme for distributed sensor networks. *Proceedings of the 9th ACM conference on Computer and communications security*, pp. 41–47, 2002.
4) Joachim VonZur Gathen and Jurgen Gerhard. *Modern Computer Algebra*. Cambridge University Press, New York, NY, USA, 2003.
5) A.S. Wander, N.Gura, H.Eberle, V.Gupta, and S.C. Shantz. Energy analysis of public-key cryptography for wireless sensor networks. In *Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications*, pp. 324–328. Citeseer, 2005.
6) K.P. Wu, S.J. Ruan, F.Lai, and C.K. Tseng. On key distribution in secure multicasting. In *Proceedings of the 25th Annual IEEE Conference on Local Computer Networks*, p. 208. IEEE Computer Society, 2000.
7) X.Zheng, C.T. Huang, and M.Matthews. Chinese remainder theorem based group key management. In *Proceedings of the 45th annual southeast regional conference*, pp. 266–271. ACM New York, NY, USA, 2007.
8) W.T. Zhu. Cryptanalysis of two group key management protocols for secure multicast. *Lecture notes in computer science*, Vol. 3810, p.35, 2005.