東海大学紀要情報通信学部
Vol.8,No2,2015,pp.9-15

論文

災害時の MANET における秘密分散法を用いた秘匿通信手法の検討

小林 桂[*1]，戸谷 洋介[*2]，宇津 圭祐[*3]，石井 啓之[*4]

A Study on Secure Communication Method using Secret Sharing Scheme over MANET in case of Disaster

by

Kei KOBAYASHI[*1], Yosuke TOTANI[*2], Keisuke UTSU[*3] and Hiroshi ISHII[*4]

(received on Sep.25, 2015 & accepted on Jan.14, 2016)

あらまし

災害時に利用可能な臨時通信ネットワーク構築技術として，MANET(Mobile Ad-hoc NETwork)が研究されている．しかしながら災害時における MANET には多くの課題が残されており，情報セキュリティに関する課題もあげられる．不特定多数のモバイル端末による無線通信手法であるというMANETの特性上，通信経路上での盗聴やデータの改ざんと行った脅威に晒されることは避けられず，また災害時を想定すると，信頼できる第三者がネットワーク内に必ずしも存在しないため，公開鍵暗号方式などの既存のセキュリティ技術も適用することができない．本稿では，経路上のノードが信頼できない状況において通信情報を第三者に奪取されるリスクを低減する通信手法の提案を行い，ネットワークシミュレータによって評価を行うことでその有効性を示している．

Abstract

A Mobile Ad-hoc Network (MANET) has been studied in order to realize a network capability in case of disaster. However, MANET has many issues to be tackled with in case of disaster. One of the issues is security. For example, MANET is vulnerable to network sniffing because it is using a wireless communication and also existing security approaches have much difficulty to be applied to MANET due to absence of trusted third party considering emergency in disaster situation. In this paper, we propose a communication method that may decrease the risk of restoration of secret information network trough sniffing by the third party. We evaluate the proposed method using network simulator and show its effectiveness.

キーワード：アドホックネットワーク、秘密分散法、秘匿通信
Keywords: *Ad-hoc network, Secret sharing schemes, Secure communication*

1. はじめに

MANET(Mobile Ad-hoc NETwork[1])は，スマートフォンやラップトップPCといった携帯可能な端末(＝ノード)間の無線通信により自律分散型のネットワークを構築し，通信インフラに依存しないネットワークの構築を実現する技術である．MANETは，災害発生時など，通信インフラが利用不可となった際の情報伝達手段としても研究されている．しかしながら，災害時におけるMANETの利用には数多くの課題が残されており，情報セキュリティの課題もそのひとつとなっている．災害時におけるMANETでは，緊急避難的に急ごしらえのネットワークを構成するため，事前に周到な準備を行うことは難しい．また既存インフラとの接続も不可能である最悪の事態を考慮しなくてはならない．よって必ずしも信頼できる第三者がネットワーク内に存在するとは限らず，そういった条件では，認証機関やPKI(Public Key Infrastructure)などの既存のセキュリティ基盤に依存した技術の適用は困難となっている．そこで，これらを利用せずともセキュアな通信を確保するためにいくつかのアプローチが検討されてきた．例えば，ネットワーク内のノードが独自の判断でノードの信頼性を評価する方式[2]や，既存のインフラ通信網と連携することで，既存のセキュリティインフラを適用する手法[3]といった提案がある．しかしながら，自律分散型ネットワークにおいて，ノードの信頼性は限界があり，上記のように既存インフラとの連携も，災害時などのインフラが必ずしも利用できない可能性があるため，災害時という状況下においてはこれらの既存手法の適用は難しいと考えられる．このように，提案されている様々なアプローチは，災害時における個々の可能性について考慮し，対応しているが，災害時のMANETの利用において，全てに対応できるセキュアな通信手法の実現は困難となっている．

上記の二つのアプローチの難点を踏まえると，外

*1 情報通信学研究科 情報通信学専攻 修士課程
Graduate School of Information and Telecommunication Engineering, Course of Information and Telecommunication Engineering, Master's Program
*2 情報通信学部 通信ネットワーク工学科
School of Information and Telecommunication Engineering, Department of Communication and Network Engineering
*3 情報通信学部 通信ネットワーク工学科 講師
School of Information and Telecommunication Engineering, Department of Communication and Network Engineering, Junior Associate Professor
*4 情報通信学部 通信ネットワーク工学科 教授
School of Information and Telecommunication Engineering, Department of Communication and Network Engineering, Professor

部のネットワークを必要とせず，また，十分に信頼できるノードの利用を前提としない手法を考慮する必要がある．

そこで，本稿では，秘密分散法[4]を適用することで経路上のノードが信頼できない場合でも，ネットワーク上の特定のノード間で，秘密情報の解読のリスクを低減する秘匿通信を実現する手法の適用性を検討する．

本論文は以下のように構成される．2章は本研究で利用する手法について述べ，3章では提案手法の詳細を述べ，4章では提案手法を評価してその有効性を述べ，5章で本手法についての考察を述べ，6章で本稿をまとめる．

2. 本研究で利用する手法

この章では，本研究で提案している手法で利用する手法として，MANET における情報伝達手法のひとつである単純フラッディングと，秘密情報を分散管理する暗号手法である(k,n)しきい値法の二つの手法についての概要を説明する．

2.1 単純フラッディング（Simple flooding）

MANET で用いられる情報伝達手法の中で，最もシンプルな情報伝達手法の一つである．送信元ノードが，データパケットを近傍ノードに対してブロードキャストし，そのパケットを受信したノードは，自分が過去に同一パケットを一度も受信したことが無かった場合，同じパケットを自身の近傍ノードに対してブロードキャストする．このようにブロードキャストの連鎖（＝フラッディング）を発生させ，ネットワークエリア内に情報を拡散させることで情報伝達を行う．ネットワーク内のノードに対して，網羅的に情報を発信するというフラッディングを利用するという特性上，通信のオーバーヘッドが大きいというデメリットもある．

2.2 (k,n)しきい値法[4]

(k,n)しきい値法は，秘密分散法と呼ばれる，秘密情報を分散管理するための暗号手法の一種である．(k,n)しきい値法では，秘密情報をシェアと呼ばれる n 個のデータ群に暗号的に分散し，それぞれをシェアのうち k 個(ただし k,n は整数で，k≤n)を集めることで元の秘密情報を復元することができる．シェアのひとつひとつは元の秘密情報とは相関が無いため，k-1 以下のシェアから元の情報を復元するのは不可能となっている．

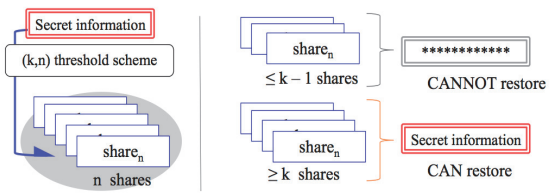

Fig.1 (k,n) threshold scheme

(k,n)しきい値法は，固定網においてセキュアな情報伝達手法にも応用されている[5]．特定のノード間で秘密情報を共有するとき，送信元ノード側で秘密情報を(k,n)しきい値法を用いていくつかのシェアに分散し，それらのシェアを別々の経路，あるいは別々の中継者（＝メンバ）を経由して宛先ノードに送信する．そうすることで，経路上の不正なノードにより，秘密情報が解読されるリスクや，なりすまし攻撃に対するリスクを低減し，且つ宛先ノード側に k 個のシェアが到達すれば秘密情報を復元できるため，経路上での情報の損失にも耐性を持つというメリットがある．しかし，MANET 環境での実施は，事前に暗号方式と鍵を共有しておくことが難しいためシェアの暗号化が困難であり，結果として経路を分散させたとしてもシェアの盗聴が容易であるという問題がある．従って，シェアを如何にして特定メンバ以外に対して秘匿するかが課題となる．

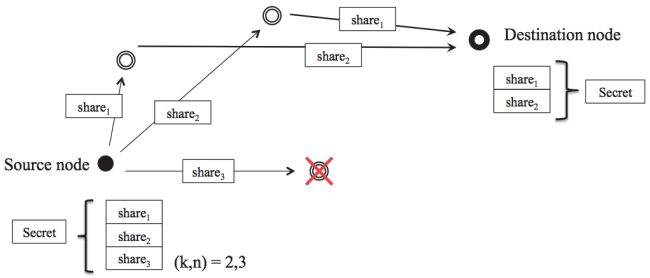

Fig.2 Secure communication method using (k,n) threshold scheme

3. 提案手法

MANET を災害時に利用するにあたっては，必ずしも信頼できる第三者がネットワーク内に存在するとは限らないため，PKI のような既存のセキュリティ技術を適用できるとは限らず，通信される情報の解読や改ざんへの対処が難しいという問題あった．本章では，この問題に対して我々が提案した手法の詳細を説明する．

3.1 前提条件

- 各ノードは，ネットワーク内をランダムウェイポイントモデル[6]に基づいて移動する．
- 各ノードにとっての，信頼できる第三者はネットワーク内に存在しないものとする．
- 各ノードは GPS 等により，自分の位置情報を把握できる．
- 秘匿通信を行う当事者間(送信元ノードにとっての宛先ノードのノード識別子（＝ノード ID）と，宛先ノードにとっての送信元ノードのノード ID)は既知であるとする．
- 通信相手を明示的に指定しないために，パケットの送信には全て，宛先アドレスにブロードキャストアドレスを指定した単純フラッディング

を用いるもとのする.

3.2 事前動作

提案手法において，ネットワーク内のノードは，以下に説明する動作を常時行う.

- HELLO パケットを用いて自らの情報(ノード ID，位置情報，送信時のタイムスタンプ）を近傍ノードに送信する.
- 自身が送信した HELLO パケットの情報を有限バッファ内（=送信 HELLO バッファ）に保存する.
- 自身が受信した HELLO パケットの情報を有限バッファ内（=受信 HELLO バッファ）に保存する.

IP header (dst IP = broadcast)	packet_type (= 0)	nodeID	location_info	timestamp

Fig.3 Format of HELLO packet

3.3 不正なノードの仮定

本手法では，秘密情報の復元を行うノードの内，当事者が想定していないノードを不正なノードとし，3.1 の前提，3.2 の事前動作に加えて以下のように仮定する.

- 不正なノードはネットワーク空間内であればあらゆる場所に存在し得る.
- 不正なノードは，自らの近傍ノードの HELLO パケットおよびデータパケットを傍受できる.
- 不正なノードは送信元ノードおよび宛先ノードのノード ID を知り得る.
- 不正なノードが知り得る他のノードの位置情報およびタイムスタンプは，自身が受信した HELLO パケットに記載されたもののみであり，受信していない HELLO パケットの情報を総当たりによっても特定する能力を持たないものとする.

3.4 秘匿通信動作

特定ノードと秘匿通信を行う際は,上記事前動作に加えて以下の(1)～(6)に示す動作を行う.

(1) 送信元ノードは，宛先ノードと共有したい秘密情報を，(k,n)しきい値法を用いて n 個のシェアに分散する．次に，自身の受信 HELLO バッファの中に格納された HELLO パケットの送信元ノードの中から，シェアの数である n と対応するように，2.2 で示したメンバに相当するノード(＝メンバノード）を決定する．Fig.4 に示す例では，3 つのシェア(share1,share2,share3)に対して，3 つのメンバノード(m1,m2,m3)を選択している.

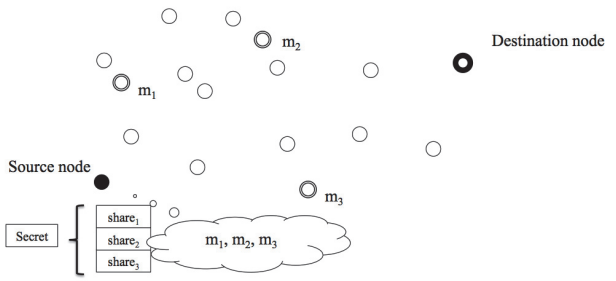

Fig.4 Share production and selection of member nodes

(2) 送信元ノードは，n 個のシェアをそれぞれ別のパケットのペイロード部に格納し，それぞれのシェアにメンバノードの HELLO 情報を用いて難読化を行う．この難読化処理は，本提案手法の最重要項目であり，正当なメンバノードと不正なノード間で，シェアの復元にかかる処理量に大きな差異を作り出し,不正なノードによりシェアを解読されるリスクを低減することを目的としている．この難読化のアルゴリズムの詳細を以下に示す.

$$h(nodeID_{mi}:location\_data_{mi}:hello\_time_{mi}) \quad \cdots (A)$$
$$\oplus\{h(nodeID_{mi}):share\_type:nodeID_s:nodeID_d:share\} \quad \cdots (B)$$

上記アルゴリズムの(B)部は，シェア(*share*)と宛先ノードの ID($nodeID_d$)，送信元ノードの ID($nodeID_s$)，シェアの最終的な宛先を判断するためのフィールド(*share_type*),そして難読化の解除の正否を判断するために任意のメンバノード(m_i)の ID($nodeID_{mi}$)が記載される．このときノード ID をハッシュ化することで，同じハッシュ関数でハッシュ化されている鍵情報(後述の(A)部)よりも bit 長を長くし，(B)部全てを難読化するまで，鍵情報を繰り返して XOR を計算することであらゆるハッシュ関数を想定した難読化処理を行うことができる.本稿では，簡単のためハッシュ関数には SHA1 を使用した．これらの値が当該メンバノード以外に秘匿される情報である.

上記の(A)部は，当該メンバノードの HELLO 情報であるメンバノードの ID($nodeID_{mi}$)，位置情報($location\_data_{mi}$)，その HELLO パケットの送信時のタイムスタンプ($hello\_time_{mi}$)を連結したもののハッシュ値からなっており，この情報は難読化の際の鍵として用いられる．HELLO パケットは，近傍ノード間でのみやり取りされる情報であるため，ネットワーク内で同一の位置情報，タイムスタンプを含む HELLO パケットを持つノードは限られる．この(A)部と(B)部の XOR 演算を行った結果をパケットに格納することで，シェアを解読されるリスクを低減している.

(3) 送信元ノードは，難読化が完了したシェアを格

納したパケットを，それぞれ単純フラッディングによってネットワーク内に送信する．

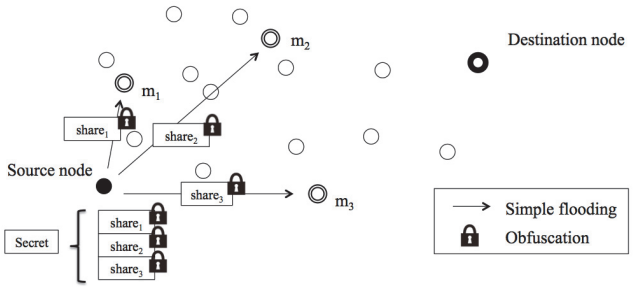

Fig.5 Transmission of obfuscated share data

IP header (dst IP = broadcast)	packet_type (= 1)	payload (= obfuscated (B))

Fig.6 Format of data packet

(4) 難読化されたパケットを受信したノードは，パケットのヘッダ部分の *packet_type* フィールドを確認し，*packet_type* フィールドが 1 だった場合，このパケットが難読化されたシェアを格納したパケットであると認識し，難読化の解除を試みる．難読化の解除は，パケットを受信したノード自身の送信 HELLO バッファ内の HELLO パケットの情報（ノード ID，位置情報，送信時のタイムスタンプを連結しハッシュ化したもの）を使い，総当たり的に EX-OR を計算することで試みられる．3.4 の(2)で説明した通り，難読化されたペイロード部は，送信元ノードによって選択されたメンバノードが送信した HELLO パケットの情報を鍵として難読化されている．したがって，自身がメンバノードであった場合は，難読化された際に鍵として用いられたものと同じ HELLO 情報が，送信 HELLO バッファ内に(FIFO による消失が生じない限り）残っているため，自身の送信 HELLO バッファ内の HELLO 情報をひとつずつ取り出し，3.4 の(2)で示した難読化アルゴリズムの(A)部のフォーマットに従って鍵を生成し，XOR を順次計算していくことで，元の情報が復元され，難読化を解除することができる．一方で，通信される情報の奪取を試みるノード(以下，不正なノード)にとっては，シェアの復元のためには，どのノードが何時どこで発信した HELLO 情報でシェアが難読化されているかを特定する必要があるため，自身の知り得る全ての HELLO 情報(=自身の送信 HELLO バッファと受信 HELLO バッファの HELLO 情報）で総当たりする必要がある．したがって，メンバノードと不正なノード間での復元時の処理の量に大きな差異を作り出すことができる．この仕組みが，本提案手法の安全性を高める要因のひとつとなっている．

難読化の解除の正否の判定は，EX-OR を計算した際に出力された文字列(3.4 の(2)の難読化アルゴリズムの(B)部に当たる部分)の先頭（本稿においては，ハッシュ関数に SHA1 を用いているため先頭 20byte）が，自身のノード ID のハッシュ値と一致した場合は，難読化解除成功，一致しなかった場合は難読化解除失敗として判定する．

難読化を解除できたノードは，ペイロード部の *share_type* フィールドを読み取り，*share_type* フィールドが 0 だった場合，自身がメンバノードであると判断し，(5)の処理へ移行する．また，*share_type* フィールドが 1 のときは(6)の処理となる．難読化を解除できなかった場合は，自身はメンバノードや宛先ノードでは無かったと判断し，当該パケットを破棄するか TTL に従ってそのままブロードキャストによる転送を行う．

(5) 上記(4)で難読化を解除したメンバノードは，取り出したシェアを再度難読化する．難読化アルゴリズムは 3.4 の(2)で用いたものと同様であるが，鍵の元となる HELLO 情報は宛先ノードのものを用いる．難読化したシェアは，単純フラッディングによりネットワークに送信される．

この際，宛先ノードになりすましている不正なノードが存在する場合，正規のノードの HELLO 情報で難読化されるシェアと，不正なノードの HELLO 情報で難読化されるシェアに分散されるため，k 個以上のシェアに不正なノード HELLO 情報による難読化がなされない限り，なりすましノードによる秘匿情報の奪取を防ぐ働きもある．

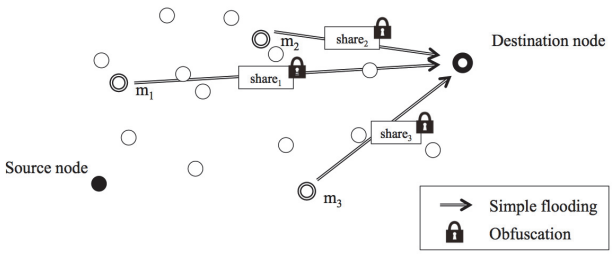

Fig.7 Obfuscated share data submission to destination by simple flooding

(6) 上記の(5)で送信されたパケットを受信したノードは，3.4 の(4)で説明したものと同様の手順で難読化の解除を行う．難読化の解除に成功した場合は，*share_type* フィールドを参照し，*share_type* フィールドが 1 であった場合，自身は宛先ノードであると判断し，シェアを取り出して保存する．そして，3.4 の(4)で，それぞれのメンバノードによって送信された他の難読化されたシェアの到達を待ち受け，k 個分のシェアの復元に成功した時点で秘密情報を復元する．

この際，パケットの損失や，メンバノードや中継ノードの不正により，いくつかのシェアが宛先ノードに届かなかった場合も，(k,n)閾値法

の特性上，(n-k)個までの損失であれば，問題なく秘密情報を復元することができる．

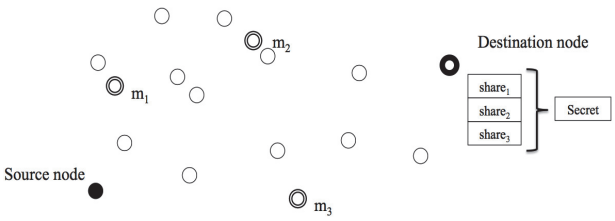

Fig.8 Share decoding and secret restoration

3.5 提案手法の拡張：データパケット削減手法

3.4 で説明した提案手法は，データ送信に単純フラッディングを用いるため，通信に際してのオーバーヘッドが大きいという欠点がある．そこで，データ送信時のデータパケットを削減する手法として，別々のノードの HELLO 情報によって難読化された複数のシェアをひとつのパケットに格納することで，シェアを送信するための単純フラッディングの回数を削減する．

4. 評価

3 章で説明した提案手法をシミュレーションにより評価した．評価条件，評価項目，評価結果を以下に示す．

4.1 評価条件

シミュレータで用いた評価条件を以下の Table1 に示す．

Table 1 The simulation conditions

Total number of nodes	100
Network size	500×500 [m]
Radio coverage	250 [m]
Simulated time	60[sec]
Moving velocity	1~10[m/s]
Hash algorism	SHA1
Buffer size	50~200
HELLO TTL	1~2
Data Packet TTL	7
(k,n)	(3,5) or (4,5)

4.2 評価項目

(1) 秘匿通信の成功率

ネットワーク内にランダム配置した任意の 2 つのノード間で，本提案手法を用いた秘匿通信を行い，宛先ノードが秘密情報の取得に成功する確率を評価する．シミュレーション時間である 60 秒のうち，最初の 30 秒間を HELLO パケットのみの交換時間とし，30 秒を経過した時点で秘匿通信を開始する．

(2) 秘密情報の解読に成功し得るノード数

本手法は，秘密情報の復元に必要となる全ての HELLO パケットを収集することの困難性を有することにより，秘密情報の解読リスクを低減する要素のひとつとしている．本項では，ネットワーク内に存在する秘密情報を復元し得る不正ノード数を調査している．調査は，シミュレーション終了時にネットワーク内に存在している全ノードの受信 HELLO バッファを参照し，秘密情報の復元に必要な全ての HELLO パケットを持つノードを，秘密情報の解読に成功し得るノードと定義し，これを探索している．

(3) 正当なノードと不正なノードとの計算量の差異

本手法は，送信元ノードによって決定されたメンバノードや宛先ノードといった正当なノードと，不正なノードとの間に処理量の差を作り出すことが，安全性を高める一要因となっている．ここでは，正当なノードと不正なノード間で，秘密情報を復元するための処理量の差を評価する．

(4) パケット削減率

3.4 で説明した拡張手法である，データパケットの削減機能について，その削減率を評価する．本稿では，ひとつのパケットに二つのシェアを格納し，HELLO_TTL が 1 のときと 2 のときで評価を行った．

4.3 評価結果

(1) 秘匿通信の成功率

Fig.9 は提案手法の秘匿通信の成功率を，k=3 のときと k=4 のときでそれぞれ評価した結果である．

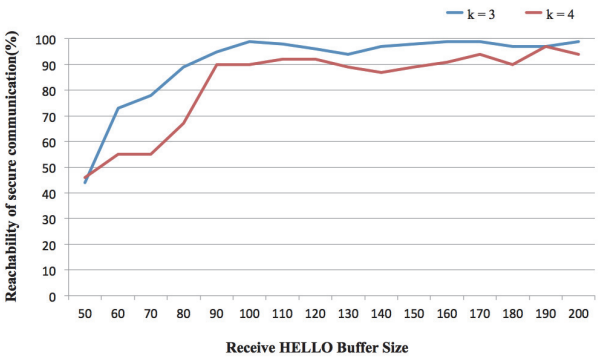

Fig.9 Reachability of secure communication

受信 HELLO バッファサイズが小さい場合を除き，概ね 9 割近い成功率を維持できることを示している．受信 HELLO バッファサイズが小さいときに到達率が低下する原因は，メンバノードが難読化をする際に，宛先ノードの HELLO パケットを保存している確率が低下するため，メンバノードが難読化の処理に失敗するためであると考えられる．

(2) 秘匿情報の復元に成功し得るノード数

Fig.10 は，秘密情報の復元に必要となる全ての HELLO パケットの収集に成功したノード数を，送信元のパケット送信間隔(ti)ごとに示している．

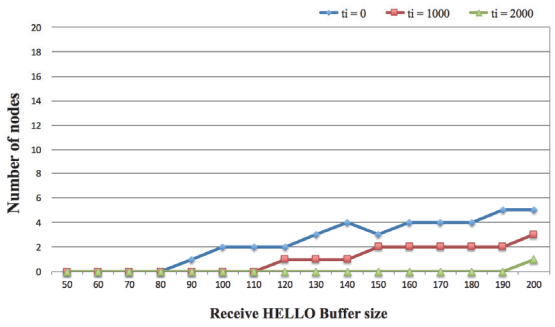

Fig.10 Number of inappropriate nodes that capable of restoring a secret information

結果から,秘密情報の復元に必要な HELLO 情報を全て収集できる可能性が低く,パケット送信間隔を大きくすることで,さらに秘匿情報の復元に成功し得るノード数が低下することがうかがえる．これは,不正なノードの受信 HELLO バッファを,送信間隔を大きくすることにより溢れさせていると考えられる．しかしながら,送信間隔の増加は,データ転送遅延の増加を招く問題がある.さらに,不正なノードが巨大なバッファを用意することで,バッファが溢れることを避け得る問題もある．

(3) メンバノードと不正なノードとの計算量の差

各ノードの受信 HELLO バッファのサイズを m，ネットワーク内のノード数を N とすると，正当なメンバノードがシェアの復元にかかる計算量は平均 m/2 となる．不正なノードがシェアの復元を試みる場合は，ノードを特定するために平均 N/2 倍の計算量がかかる．また，メンバノードの場合は，自身が担当するシェアのみ復元すれば良いが，不正ノードが秘密情報そのものを解読する場合は，残りの k 個のシェアも同様に復元する必要があり，k 倍の計算量が生じる．よって，メンバノードと不正なノードの計算量の比率（C_{co}）は以下の数式(1)に示す通りになる．

$$C_{co} = \frac{k \cdot \frac{N}{2} \cdot \frac{m}{2}}{\frac{m}{2}} = k \cdot \frac{N}{2} \qquad \cdots(1)$$

Fig.11 は，メンバノードや宛先ノードといった正当なノードにおける秘密情報の復元に必要な計算量に対して,不正なノードが秘密情報の復元に掛かる処理の比率を，数式(1)に基づいて示したものである．本研究では，簡単のため(k,n)の値を小さいものとしており，計算量の際は大きくないが，評価の結果，k=2 から k=5 の範囲内においては，正当なノードと不正なノードとの間で約 200～500 倍の計算量の差異が生じることがわかる．

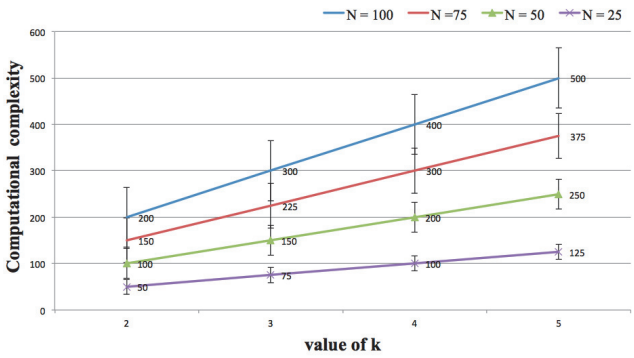

Fig.11 Differences of computational complexity between legitimate nodes and inappropriate nodes

(4) パケット削減率

Fig.12 は 3.4 で説明したデータパケット削減機能の効果を示している．赤いバーが，データパケットの削減手法を用いなかった場合で，青いバーがデータパケットの削減手法を用いた場合である．結果から，一度の秘匿通信でのデータパケットを，HELLO_TTL が 1 のときに約 47%，HELL_TTL が 2 のときに約 35%削減できることがわかる．

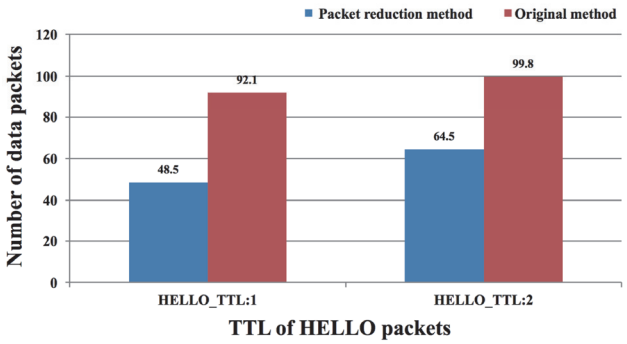

Fig.12 Packet reduction effect

5. 考察

以上より，提案方式は，不正ノードによる秘密情報の解読のリスクを低減できる可能性を示した．しかしながら，課題も残されている．

本研究の課題は，大きく方式そのものについての課題と，評価における課題がある．

方式面での課題は,その安全性があくまでリスク低減の域を出ないことである．本手法の安全性は,秘密情報の復元に必要な HELLO パケットの情報を,不正ノードが入手することが難しいことと，当該 HELLO パケットを特定することの困難性からなる．難

読化アルゴリズムや，鍵情報の強化により，これらの安全性を向上させる必要がある．また，本稿における評価では，全てのノードが移動しているため，不正なノードが特定の位置で HELLO 情報を収集する行為を想定しておらず，それを考慮する必要がある．

評価における課題は，本稿におけるシミュレーションについてである．本稿におけるシミュレーションは，提案した手法に秘匿情報を解読されるリスクを削減効果が存在することを調査するために行ったものであり，一部のパラメータや，利用手法は，簡単のため，安全性や妥当性の観点から最適とは言えないものとなっている．例えば，本手法で利用している，(k,n)しきい値法の，k と n の値は極めて小さいものとなっており，また (k,n)しきい値法自体にも，メモリを大量に消費するというデメリットがあるため，コストが大きい．検討課題としては，これを(k,L,n)ランプ型しきい値法へ変更し，(k,L,n)の値を最適化し，コストと安全性の評価を行う．

また，難読化アルゴリズムにおいて利用しているハッシュ関数は，簡単のため SHA1 を用いていたが，脆弱であるため，これを SHA2 に変更する．

6. まとめ

本稿では，MANET 上での秘匿通信が困難であり，経路上のノードによる通信の盗聴や改ざんが容易であるという問題の解決手法として，秘密分散法を用いて経路上のノードの信頼性に依存せずに通信経路上での盗聴リスクを低減する手法を提案し，評価を行った．評価の結果，概ね 9 割近い秘匿通信の成功率を示し，秘匿情報の復元に必要な HELLO パケットの入手の困難性や，同パケットを入手した上での計算量の差異により，正当なノードと不正なノードとの間の秘匿情報復元の際の困難性の差があることを示した．これにより提案手法が，経路上での秘密情報の解読リスクを低減できる可能性があるという結果を示した．しかしながら,安全性は充分とは言えず，手法の改善を検討する必要がある．

謝辞

本研究の一部は，日本学術振興会科研費26420372の支援を受けたものである．

参考文献

1) 間瀬憲一，阪田史郎，アドホック・メッシュネットワーク ユビキタスネットワーク社会の実現に向けて”，コロナ社，Vol.1 2006
2) Srdjan Capkun, Levente Buttyan, jean-Pierre Hubaux, “Self-Organized Public-Key Management for Mobile Ad Hoc Networks”, IEEE Transctions on Mobile Computing, vol.2, No.2, pp-52-64, Jan-Mar.2003
3) Takeshi Kubo, Hidetoshi Yokota, Akira Idoue, “A Proposal of Secure Ad hoc Routing Protocol with an Infrastructural Support” 2005
4) A Shamir, “How to Share a Secret,” Communications of the ACM, Vol.22, No.12, pp.612-613, Nov.1979
5) 山中仁昭，宮本伸一，三瓶政一:秘密分散法に基づくセキュアな無線通信リンクの形成に関する一検討，信学ソ大会，B-5-127，Sept.2010
6) Camp, T., Boleng, J. and Davies, V.: A Survey of Mobility Models for Ad Hoc Network Research, WCMC：Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, Vol.2, pp.483-502 (2002)