アドホックネットワークにおけるデータ差替え攻撃を考慮したTop-k検索手法および攻撃端末特定手法

津田 琢士[1]　駒井 友香[1]　佐々木 勇和[1]　原 隆浩[1]　西尾 章治郎[1]

概要：アドホックネットワークでは，通信帯域が限られているため，端末が何らかの値（スコア）によって順序付けられたデータの上位 k 個のものを検索する Top-k 検索を用いることが有効である．しかし，外部からの攻撃により，ネットワーク内の端末が攻撃端末となった場合，正確な検索結果の取得を妨害される恐れがある．本稿では，攻撃端末が受信したデータを自身が保持するスコアが低いデータに差し替えて返信（データ差替え攻撃とよぶ）を行うことを想定し，アドホックネットワークにおけるデータ差替え攻撃を考慮した Top-k 検索手法および攻撃端末特定手法を提案する．提案手法では，クエリ応答を複数の経路によって送信することで，取得精度を維持する．また，クエリ応答の転送経路をメッセージに添付して送信することで，クエリ発行端末は，データ差替え攻撃を検知することができる．さらに，この情報を用いて，攻撃端末候補を絞り込み，攻撃端末候補のデータの送信先端末にデータの受信状況を問い合わせることで，攻撃端末を特定する．シミュレーション実験の結果から，提案する Top-k 検索手法は，複数経路でクエリ応答を送信することで取得精度を維持でき，さらに攻撃端末特定手法により，攻撃端末を高い確率で特定できることを確認した．

Methods for Processing a Top-k Query Against a Data Replacement Attack and Identifying a Malicious Node in MANETs

Takuji Tsuda[1]　Yuka Komai[1]　Yuya Sasaki[1]　Takahiro Hara[1]　Shojiro Nishio[1]

1. はじめに

近年，ルータ機能をもつ移動端末のみで一時的な無線ネットワークを形成するアドホックネットワークへの関心が高まっている．アドホックネットワークでは，通信帯域が限られているため，膨大なデータの中から必要なデータのみを効率的に検索を行う必要がある．そのため，何らかの値（スコア）によって順位づけられたデータの上位 k 個のものを検索する Top-k 検索に関する研究が頻繁に行われている．

既存の Top-k 検索手法は，全ての端末が正常に動作することを前提としている．しかし，ネットワーク内の端末が外部からの攻撃により，攻撃端末となることが考えられる．このような状況で Top-k 検索を行うと，攻撃端末によって，他の端末から受信したデータを自身が保有するスコアの低いデータに差し替えて返信する攻撃（データ差替え攻撃とよぶ）により，クエリ発行端末は正しい検索結果を取得できない可能性がある．さらに，スコア上位 k 個のデータは中継端末で置き換えられ，その状況をクエリ発行端末が把握できないため，攻撃を受けたかどうか判断することができない．そのため，クエリ発行端末は必要なデータを取得したと勘違いしてしまう．

そこで本稿では，アドホックネットワークにおけるデータ差替え攻撃を考慮した Top-k 検索手法および攻撃端末特定手法を提案する．提案手法では，クエリ応答を複数の経路によって送信することで，取得精度を維持する．また，クエリ応答の転送経路をメッセージに添付して送信することで，クエリ発行端末は，データ差替え攻撃を検知することができる．さらに，この情報を用いて，攻撃端末候補を絞り込み，攻撃端末候補のデータの送信先端末にデータの受信状況を問い合わせることで，攻撃端末を特定する．シミュレーション実験の結果から，提案する Top-k 検索手法

[1] 大阪大学 情報科学研究科 マルチメディア工学専攻
Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University

は，複数経路でクエリ応答を送信することで取得精度を維持でき，さらに攻撃端末特定手法により，攻撃端末を高い確率で特定できることを確認した．

以下では，2 章で関連研究を紹介し，3 章で想定環境について説明する．その後，4 章で提案手法について説明し，5 章でシミュレーション実験の結果を示す最後に，6 章で本稿のまとめと今後の課題について述べる．

2. 関連研究

2.1 セキュアルーティング手法

センサネットワークやアドホックネットワークの分野では，攻撃端末による送信データの改ざんや，通信の妨害によってサービスの提供を妨げる DoS（Denial of Service）攻撃 [9] に対する研究が盛んに行われている．そのような攻撃に対するルーティング手法（セキュアルーティング手法）として，送信元端末から目的端末まで複数経路を用いてデータを送信する手法，および送信するデータを共通鍵や公開鍵を用いて暗号化して送信する手法が提案されている．文献 [7] の手法では，データ送信端末から目的端末まで複数の経路でデータを送信する手法を提案している．この手法は，データを送信する前に安全に送信できる経路を発見することを目的としており，さらに，データ送信時に，検索結果に入るデータが差し替えられることを想定していない．文献 [3] では，ネットワーク内の各センサ端末がシンク端末との暗号鍵を事前に所有しておき，送信するデータを暗号鍵で暗号化してシンク端末に送信する手法を提案している．この手法では，暗号化されたデータを復号する鍵をもっていないセンサ端末は暗号化されたデータを解読することができないため，安全にデータを送信できる．この研究では，データの検索ではなく，特定の端末への安全なメッセージ送信を目指しており，Top-k 検索は考慮していない．

2.2 Top-k 検索手法

Top-k 検索手法は，様々な分野で研究が行われている．非構造型 P2P ネットワークにおける Top-k 検索 [5] や無線センサネットワークにおける Top-k 検索 [6], [10] では，各端末が送信するメッセージ数を削減することで，無駄なトラヒックを削減している．しかし，これらの手法はデータ差替え攻撃を考慮していないことに加えて，端末の移動にも対応しておらず，アドホックネットワークに適応することは難しい．アドホックネットワークにおける Top-k 検索において，検索クエリおよびデータ返信におけるトラヒックを削減しつつ，取得精度の維持を実現する Top-k 検索手法 [1], [4], [8] が提案されている．文献 [4] の手法では，検索クエリにデータのスコアの一部を添付することにより，検索結果に含まれないデータの返信を抑制し，トラヒックを削減する．また，文献 [8] の手法では，ネットワーク内の上位 k 個のデータのスコアを収集し，収集したデータのスコアの k 番目の値を閾値に設定した後，閾値以上のスコアをもつデータを取得する 2 フェーズ検索手法を提案している．さらに，文献 [1] の手法では，検索結果を取得するために必要な端末のみで Top-k 検索を行うことで，トラヒック削減を実現している．しかし，これらの研究は攻撃端末が存在する環境を想定しておらず，データ差替え攻撃に対応できない．

3. 想定環境

本稿では，アドホックネットワークを構成する各端末が，自身と他の端末が持つデータに対して Top-k 検索を行う環境を想定する．システムモデルおよび攻撃モデルについて詳細を以下に示す．

3.1 システムモデル

ネットワーク内には，同等の性能を持つ m 台の端末（識別子：M_1，M_2，...，M_m）が存在し，各々が自由に移動する．また，ネットワーク内の端末は，自身の通信範囲およびネットワークの領域サイズを把握しているものとする．さらに，ネットワーク内の端末は，データ（クエリ応答）を送信する際，送信先の端末と 1 対 1 の暗号化通信を行う．一方，検索クエリを送信する際は，隣接端末を事前に把握していないため，暗号化を行わず，メッセージをブロードキャストする．データのスコアは，検索条件から何らかのスコアリング関数を用いて算出されるものとする．また，各端末の所有するデータは事前に暗号化されており，クエリ発行端末のみ解読することが可能であるものとする．ただし，データのスコアはクエリメッセージに添付されるため，把握することができるものとする．

3.2 攻撃モデル

攻撃端末は，クエリ発行端末に，攻撃されていることを検知，および自身が攻撃端末であることを特定されないように攻撃を行うものとする．データに添付されているスコアを改ざんした場合，データのスコアとの整合性を確認することにより，攻撃端末の存在が検知されてしまうため，スコアの改ざんは行わない．攻撃端末は，隣接端末から受信したクエリ応答に含まれるデータおよび自身の所有するデータの上位 k 個のうち，$\lceil h \cdot k \rceil$ 個（h：データ差替え割合）のデータを自身の所有するデータに差し替えて送信する．また，データ差替え攻撃以外の攻撃は行わないものとする．攻撃端末は，1 台のみ存在する．この想定は，実環境では必ずしも正しくはないが簡単化のため 1 台とする．

4. 提案手法

4.1 手法概要

提案する Top-k 検索手法では，検索結果の取得精度を維持しつつ，データ差替え攻撃の検知を行うため，2 本の経

路によってデータを返信する．まず，クエリ発行端末は，フラッディングにより検索クエリを送信し，受信した端末はクエリ発行端末までの複数の経路を把握する．クエリ応答では，隣接端末を 2 台選び，クエリ応答を送信する．クエリ応答にメッセージの送信元および送信先端末の情報を格納した応答転送経路リストを添付することにより，クエリ発行端末では応答転送経路リストを用いて，攻撃の検知を行う．また，アドホックネットワークでは端末の移動より，端末間の無線リンクが切断する．この場合，2 本の経路によってメッセージの返信ができなくなるため，リンク切断を検出した端末は別の隣接端末に送信する．

提案する攻撃端末特定手法では，攻撃を検知したクエリ発行端末が，まず応答転送経路リストより攻撃端末の候補を絞り込む．その後，攻撃端末の候補が送信したクエリ応答の受信端末に対して，差し替えられたデータがクエリ応答に含まれているか確認することで，攻撃端末を特定する．

4.2 Top-k 検索手法

4.2.1 メッセージ処理

検索クエリ転送

フラッディングにより検索クエリを転送する．検索クエリは，検索クエリの発行端末の識別子 *Query-issuing nodeID*，検索クエリの識別子 *Query ID*，要求データ数，k，検索条件 *Query condition*，およびクエリ経路リスト（クエリ発行端末から検索クエリ送信元端末までの端末識別子のリスト）*Query path* が含まれる．Top-k 検索を行う端末 M_p は，検索条件および要求データ数 k を指定し，クエリ経路リストに自身の識別子を格納した検索クエリを隣接端末に送信する．検索クエリを受信した端末 M_q は，Algorithm 1 に従って検索クエリを転送する．ここで，Algorithm 1 における，*hop counts* は *Query path* に含まれる端末数より算出したクエリ発行端末からのホップ数を表す．また，*RD* はクエリ応答を送信するまでの返信待ち時間を表し，次式のように設定する．

$$RD = (hop_{max} - hop_{cnt}) \cdot T_{wait} \quad (1)$$

hop_{cnt} はクエリ発行端末からのホップ数，hop_{max} は領域サイズおよび通信半径から推測される最大のホップ数，T_{wait} は事前に設定される正の定数を示す．この式を用いることで，ホップ数が大きい端末ほど，小さい *RD* が設定される．M_q が再度検索クエリを受信した場合に，メッセージの送信元端末を隣接端末として記録し，メッセージに含まれるクエリ経路リストおよびそのホップ数を記録する（Algorithm 1 第 10・11 行目）．

クエリ応答転送

返信待ち時間 *RD* 経過後，各端末はクエリ応答を返信する．クエリ応答には，クエリ応答の送信端末の識別子 *Sender nodeID*，クエリ応答の送信先端末の識別子 *Dest nodeID*，返信データとそのスコアおよび保有端末の識別子 *Data list*，および応答返信経路リスト（クエリ応答の送信元端末識別子と送信先端末識別子のペアのリスト）*Forwarding Route* が含まれる．

Algorithm 1 検索クエリの転送

```
1: /* 検索クエリの受信 */
2: if M_q receives the query for the first time then
3:     Store Query path and hop count as its Parent Query path
4:     Store the nodeID at the end of Query path as its parent
5:     Set RD for replying data items
6:     /* 隣接端末に検索クエリを送信 */
7:     Add M_q's nodeID to the end of Query path
8:     Send the query to neighbor nodes
9: else
10:    Store Query path and hop counts as its Neighbor Query path
11:    Store the nodeID at the end of Query path as its neighbor
12: end if
```

表 1 各端末の持つデータのスコア

Table 1 Scores of data items held by each node

端末	スコア
M_1	79，72，69，56，55，47，32，29
M_2	72，65，62，59，51，49，40，22
M_3	95，76，75，61，53，46，37，35
M_4	84，81，79，71，66，60，58，27
M_5	91，80，77，54，44，36，25，19
M_6	98，86，78，67，58，42，38，30

端末 M_r が，クエリ応答を送信するアルゴリズムを Algorithm 2 に示す．ここで，Algorithm 2 において，*REP* は，クエリ応答を表し，*REP.FR(Sender nodeID, Dest nodeID)* は，クエリ応答に添付された応答転送経路リストを表す．また，R はクエリ応答の再送回数を表す．受信した *Query path* を用いて，ホップ数が最も短く，かつ重複の少ない経路をもつ隣接端末および自身の親端末を送信先端末とする（Algorithm 2 第 3〜10 行目）．

リンク切断時の対応

アドホックネットワークでは，端末の移動によりネットワークトポロジが動的に変化する．ここで，クエリ応答送信時に，親端末，または隣接端末とのリンクが切断した場合，その送信先端末へクエリ応答を送信できないため，検索結果の取得精度が低下する可能性がある．そのため，R 回クエリ応答を送信したにも関わらず，送信先端末から ACK を受信できなかった端末は，送信先端末とリンク切断が発生したと判断する．リンク切断を検出した端末は，自身の隣接端末のうち，すでに送信先とした端末以外で，クエリ発行端末までの経路上に送信先の端末が含まれる隣接端末が存在する場合，その隣接端末（複数台の場合はそのうちの一つ）にクエリ応答を送信する（Algorithm 2 第 36 行目）．これに該当する隣接端末が存在しない場合，すでに送信先とした端末以外で Algorithm 2 第 9 行目に示した条件に該当する隣接端末にクエリ応答を送信する（Algorithm 2 第 38・39 行目）．

Algorithm 2 クエリ応答の返信

```
/* RD 経過後，クエリ応答を送信 */
/* クエリ応答の送信先を選択 */
for each Neighbor do
    if Neighbor's hopCount is the minimum then
        Insert Neighbor into DestNode
    end if
end for
if |DestNode| > 1 then
    Select a Neighbor whose Neighbor Query path least overlaps with the parent Query path as a DestNode
end if
Add the local top-k result to REP
for i = 0 to 1 do do
    if i = 0 then
        Add (M_r, parent node) to received REP.FR and send REP to parent node
    else if i = 1 then
        Add (M_r, DestNode) to received REP.FR and send REP to DestNode
    end if
end for
/* クエリ応答を受信 */
Send ACK to the sender node of REP
if before RD then
    Store REP
else if after RD and M_r receives a data item with higher score than with the kth-highest score among data items already sent then
    Send REP including new local top-k result to parent node and DestNode
end if
/* クエリ応答を再送 */
if M_r does not receive ACK from its parent by waiting time for retransmission and the number of retransmissions < R then
    Resend REP to parent
else if M_r does not receive ACK from DestNode by waiting time for retransmission and the number of retransmissions < R then
    Resend REP to DestNode
else if the number of retransmissions > R then
    /* M_r がリンク切断を検知 */
    if M_r has sent REP to all Neighbor then
        Discard REP
    else if M_r knows a Neighbor whose Neighbor Query path includes DestNode then
        Send REP to the Neighbor
    else
        Select randomly a Neighbor among Neighbors which have not been selected yet
        Send REP to the Neighbor
    end if
end if
```

図 1 を用いて，クエリ発行端末 M_1 が検索クエリを送信した後，各端末がクエリ応答を送信する動作について説明する．この図では，$k = 3$ の場合の Top-k 検索の例を示す．なお，各端末の持つデータのスコアは表 1 で表されている．まず，M_1 から最もホップ数の大きい M_6 が最も小さい RD を設定するため，最初にクエリ応答を送信する．M_6 は，自身のもつデータのうちスコアの大きい 3 個のデータ（スコア：98，86，78）を返信データとし，親端末 M_4，および隣接端末 M_5 にクエリ応答を送信する．このとき，M_4 へ送信したクエリ応答の応答転送経路リストは $\{(M_6,M_4)\}$，M_5 へ送信したクエリ応答の応答転送経路リストは $\{(M_6,M_5)\}$ となる．M_2〜M_5 も同様の動作を行い，M_2 および M_3 からクエリ応答を受信した M_1 は，M_2 による攻撃に関わらず，スコアの高い上位 3 個のデータ（スコア：98，95，91）を取得できる．

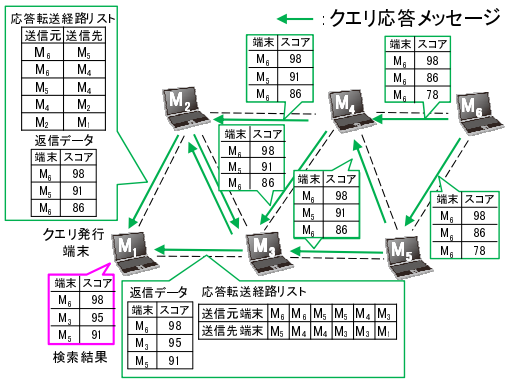


図 1 攻撃端末が存在する環境下でのクエリ応答の返信

Fig. 1 Example of transmission of *REP*

Algorithm 3 攻撃の検知

```
/* クエリ応答をすべて受信後 */
INPUT: Top-k Result, REPs
OUTPUT: SendRoute
SendRoute ← ∅
for each REP do
    for each Top-k Result do
        if REP.FR includes the nodeID of a node processing a data item in Top-k Result and REP.Data does not include the data item then
            Insert a route from the node with the missing data item to the query-issuing node into SendRoute
        end if
    end for
end for
if SendRoute ≠ ∅ then
    Detect Attack
end if
```

4.2.2 攻撃の検知

全てのクエリ応答を受信したクエリ発行端末 M_p は，Algorithm 3 に従って，攻撃の検知を行う．ここで，Algorithm 3 における *Top-k Result* は，クエリ発行端末で取得した検索結果を表す．また，*REP.Data* および *REP.FR* は，それぞれクエリ応答の返信データおよび転送経路リストを表す．さらに，*SendRoute* は，差し替えられたデータの保有端末から，クエリ発行端末までの経路上の端末識別子リストを表す．クエリ発行端末では，受信したクエリ応答から，検索結果に含まれるデータの保有端末が転送経路リストに含まれているのに，検索結果に含まれるデータを受信していない場合，攻撃を検知し（Algorithm 3 第 13 行

Algorithm 4 攻撃端末の候補の絞り込み

```
1: INPUT: SendRoute
2: OUTPUT: Candidate
3: /* クエリ発行端末で，データが差し替えられたか確認 */
4: for each nodeID in SendRoutes do
5:    if nodeID is included in SendRoute then
6:       Insert nodeID into Candidate
7:    end if
8: end for
9: if |Candidate| = 1 then
10:   return Candidate as a Malicious Node
11: else if |Candidate| > 1 then
12:   /* 攻撃端末の候補に問合せ */
13:   Perform the procedure of Algorithm 5
14: end if
```

目），4.3 節に示す攻撃端末の特定を行う．一方，攻撃を検知しなかった場合，正しい検索結果を得られたと判断し，検索を終了する．

図 1 を用いて，クエリ発行端末 M_1 が攻撃の検知を行う例について説明する．図 1 において，攻撃端末 M_2 は，M_3 および M_4 から受信したクエリ応答に含まれる返信データのうち，上位 3 個のスコアをもつデータ（スコア：98，91，86）の上位 1 番目と 2 番目のスコアをもつデータを差し替えたデータを返信データ（スコア：86，72，65）として送信している．このとき，M_1 は，M_2 からの応答転送経路リストには検索結果に入るデータの所有端末 M_5 および M_6 が送信元端末に含まれていることがわかる．しかし，M_2 からの返信データには M_5 および M_6 が所有する検索結果に入るデータ（スコア：98，91）が含まれておらず，このメッセージの返信データは，データが差し替えられた（攻撃を受けた）ことを検知できる．

4.3 攻撃端末特定手法

クエリ発行端末は，攻撃を検知した場合，攻撃端末の特定を行う．攻撃端末の候補を絞り込み，問合せにより攻撃端末を特定するアルゴリズムを Algorithm 4 および 5 に示す．この手法では，まずクエリ発行端末が，4.2.2 項で攻撃を検知した時（Algorithm 3）の，*SendRoute* を用い，Algorithm 4 に従って攻撃端末の候補を絞り込む．ここで，*Candidate* は，攻撃端末の候補の識別子のリストで，クエリ発行端末からのホップ数による昇順に格納している．差し替えられたデータを含むクエリ応答の全ての経路上に存在する端末は，攻撃の機会があると考えられるため，これらを攻撃端末の候補とする（Algorithm 4 第 6 行目）．このとき，攻撃端末の候補が 1 台となった場合，この端末を攻撃端末と特定し，終了する（Algorithm 4 第 10 行目）．

攻撃端末の候補に問合せを行うアルゴリズムを Algorithm 5 に示す．ここで，*INQ* は，問合せメッセージを表し，クエリ発行端末の識別子 M_p，問合せ先の端末の識別子 M_{dest_i}，攻撃端末の候補の識別子 *Candidate*，およびクエリ発行端末 M_p から問合せ先の端末 M_{dest_i} までの送信経路

Algorithm 5 攻撃端末の特定

```
1: INPUT: Candidate
2: OUTPUT: MaliciousNode
3: /* M_p が問合せを開始 */
4: for each i in Candidate.size do
5:    if InqRoute include other candidates in Candidate then
6:       /* 問合せをせずに終了 */
7:       break
8:    else if hop count to Candidate > 1 then
9:       /* 問合せメッセージを送信 */
10:      Send INQ to M_{dest_i} to ask data items that Candidate[i]
         sent
11:   end if
12:   /* M_v が，問合せメッセージを送信 */
13:   if M_v receives INQ then
14:      Send INQ to the next node of M_v in InqRoute
15:   end if
16:   /* 攻撃端末の候補が，問合せメッセージを受信 */
17:   if M_{dest_i} receives INQ then
18:      Send IREP including scores of data items sent by Can-
         didate[i] to M_p
19:   end if
20:   /* M_u が，問合せ返信メッセージを受信 */
21:   if M_u receives IREP then
22:      Send IREP to sender INQ
23:   end if
24:   /* M_p が，問合せ返信メッセージを受信 */
25:   if M_p receives IREP then
26:      /* クエリ発行端末が，攻撃端末を特定 */
27:      if scores includes the score of the missing data items in
         global Top-k result then
28:         return Candidate[i − 1]
29:      end if
30:   end if
31: end for
```

InqRoute が含まれる．また，M_{dest_i} は，*Candidate[i]* のクエリ応答の送信先端末を表す．*IREP* は，問合せ返信メッセージを表し，*missing Top-k result* は，差し替えられたデータを表す．攻撃端末の候補 *Candidate* から受信したクエリ応答の返信データのスコアとそのデータの所有端末の識別子が含まれる．クエリ発行端末は，攻撃端末の候補がクエリ応答を送信した端末のうち，いずれの端末も転送経路上に攻撃端末の候補が含まれない端末に，ホップ数が小さい攻撃端末の候補から順に問合せメッセージを送信する．問合せメッセージの送信先端末が，攻撃端末の候補を経由しなければいけない場合は，問合せメッセージを送信せず，攻撃端末の特定を終了する．ここで，ホップ数が 1 の端末は，クエリ発行端末が直接返信データを受信しているため，問合せを行わない．M_p が問合せ返信メッセージを受信した後，攻撃端末の特定を行う．問い合わせを行った攻撃端末の候補が，送信したデータに差し替えられたデータが含まれる場合，M_p に 1 ホップ近い候補 (*Candidate[i − 1]*) を攻撃端末として特定する（Algorithm 5 第 28 行目）．

図 1 において，攻撃を検知した後，攻撃端末を特定する動作について図 2 を用いて説明する．クエリ発行端末 M_1

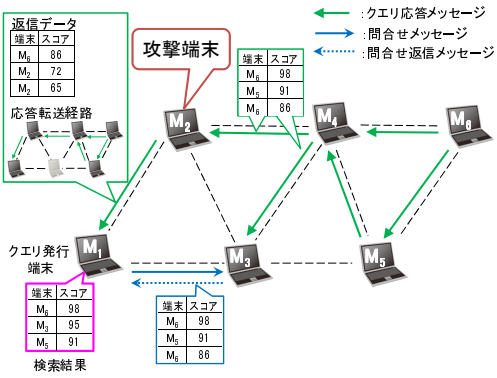


図 2 攻撃端末の特定

Fig. 2 Identification of the malicious node

は，M_2 から受信したクエリ応答の応答転送経路リストより，検索結果に入るデータを所有する M_5 および M_6 から，M_2 を経由してクエリ発行端末 M_1 までクエリ応答が送信された経路をそれぞれ算出する．全ての経路の共通部分である M_2，および M_4 を攻撃端末の候補とする．図 2 では，クエリ発行端末 M_1 は，攻撃端末の候補である端末 M_4 のクエリ応答の送信先であり，攻撃端末の候補でない端末 M_3 に問合せメッセージを送信する（M_2 は，ホップ数が 1 なので，問合せを行わない）．問合せメッセージを受信した M_3 は，攻撃端末の候補 M_4 からの返信データのスコア情報（スコア：98，91，86）を M_1 に返信する．M_3 から問合せ返信メッセージを受信した M_1 は，受信したスコア情報（スコア：98，91，86）より，M_4 は検索結果に含まれるデータを返信していることを確認する．これにより，M_4 は正しく返信を行っているが，M_2 より受信したクエリ応答に必要なデータが含まれなかったため，攻撃端末は M_2 であると特定する．

5. シミュレーション評価

本章では，提案手法の性能評価のために行ったシミュレーション実験の結果を示す．本実験では，ネットワークシミュレータ Qualnet5.2[*1]を用いた．

5.1 シミュレーション環境

500[m] × 500[m] の 2 次元平面状の領域に 50 台の端末（M_1,M_2, ⋯;M_{50}）が存在する．そのうち，1 台の端末を攻撃端末とする．各端末はランダムウェイポイント [2] に従い，0.5[m/秒] の速度で移動し，停止時間は 30[秒] とした．各端末は，IEEE802.11b を使用し，伝送速度 11[Mbps]，通信伝搬距離が 100[m] 程度となる送信電力でデータを送信する．各端末は，128[B] のサイズのデータをそれぞれ 50 個もつものとし，ネットワーク内には 2,500 個のデータが存在するものとした．攻撃端末は，受信したデータのうち $\lceil h \cdot k \rceil$ 個のデータを任意に選んで，自身の所有するデータに差し替える攻撃を行うものとした．攻撃端末は，クエリの送信 10 回毎にランダムに決定した．比較手法として，クエリ発行端末が検索クエリをネットワーク全体にフラッディングし，検索クエリを受信した端末が，自身の持つデータおよび受信したデータの中からスコアの高い k 個のデータを，親端末にのみ返信（単一経路）する単純手法を用いた．表 2 に本実験で用いたパラメータを示す．各パラメータは基本的に定数値をとるが，そのパラメータの影響を調査する際には括弧内の範囲で値を変化させた．また，再送回数 R は 3 とした．

表 2 パラメータ設定

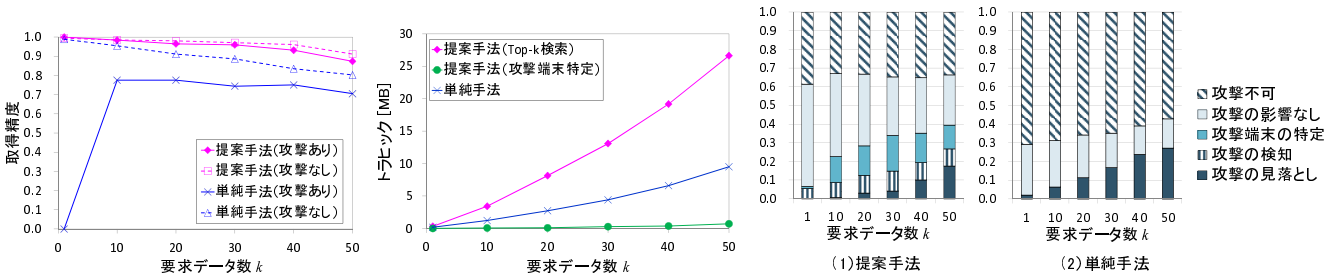
Table 2 Parameter Configuration

パラメータ	意味	値
k	要求データ数	30 (1〜50)
h	差替え割合	0.5 (0.2〜1.0)

以上のシミュレーション環境において，各端末の初期位置をランダムに決定して配置し，30 秒ごとにランダムに選択された端末がクエリを発行するという処理を 1,000 回繰り返した際の，以下の評価値を調べる．

- **取得精度**：ネットワーク内のスコアの高い上位 k 個のデータのうち，クエリ発行端末が取得できたデータ数の割合（1 クエリ当たりの平均）．
- **トラヒック**：検索クエリを処理するために送信された全てのメッセージの平均バイト数（1 クエリ当たりの平均），および攻撃端末を特定するために送信された全てのメッセージの平均バイト数（1 問合せ当たりの平均）．
- **攻撃端末の特定割合**：以下の 5 つに分類し，それぞれの割合を示す．
  - **攻撃不可**：発行したクエリのうち，攻撃端末が他の端末からのデータを受信せず，攻撃ができなかった割合．
  - **攻撃の影響なし**：発行したクエリのうち，攻撃端末により攻撃が行われたが，中継端末によってそのデータが正しいデータに置き換えられたため，クエリ発行端末が正しい検索結果を取得できた割合．（つまり，攻撃されたことがクエリ発行端末で認識できない）
  - **攻撃端末の特定**：発行したクエリのうち，クエリ発行端末で攻撃を検知し，さらに攻撃端末を特定できた割合．（単純手法では 0）
  - **攻撃の検知**：発行したクエリのうち，クエリ発行端末で攻撃を検知したが，攻撃端末を特定できなかった割合．（単純手法では 0）
  - **攻撃の見落とし**：発行したクエリのうち，攻撃端末による攻撃により，クエリ発行端末で正しい検索結果を得ることができず，かつ，攻撃を検知できなかった割合．

[*1] Scalable Network Technologies: Creators of Qualnet Network Simulator Software,
< http://www.scalable-networks.com >



(a) 取得精度　(b) トラヒック　(c) 攻撃端末の特定割合

図 3　要求データ数 k の影響

Fig. 3　Effect of k

5.2　要求データ数 k の影響

要求データ数 k を変化させたときの結果を図 3 に示す．これらの図において，横軸は要求データ数 k を表し，縦軸は，図 3(a) は取得精度，図 3(b) はトラヒック，図 3(c) は，攻撃端末の特定割合を表す．図 3(a) 中の凡例では，それぞれ攻撃端末に攻撃を受けた場合を「攻撃あり」(攻撃端末の特定，攻撃の検知，および攻撃の見落としの場合)，攻撃を受けなかった場合を「攻撃なし」(攻撃不可，および攻撃の影響なしの場合) と表記し，以降，取得精度の図中では同様の表記とする．

図 3(a) の結果より，どちらの手法においても要求データ数が大きくなるにつれて，基本的に取得精度が低下している．これは，返信するデータサイズが大きくなり，パケットロスが起こる機会が増えるためである．単純手法では，「攻撃あり」の場合，k が 1 で取得精度が 0 になる．これは，全ての場合でスコアの一番高いデータを攻撃端末に差し替えられ，取得できなかったためである．攻撃の有無にかかわらず，提案手法は，単純手法に比べて取得精度が高い．これは，提案手法では，複数経路でクエリ応答を返信しているため，攻撃に強いのに加えて，片方の経路でパケットロスによりクエリ発行端末まで送信できない場合でも，もう片方の経路によりデータを送信できるためである．

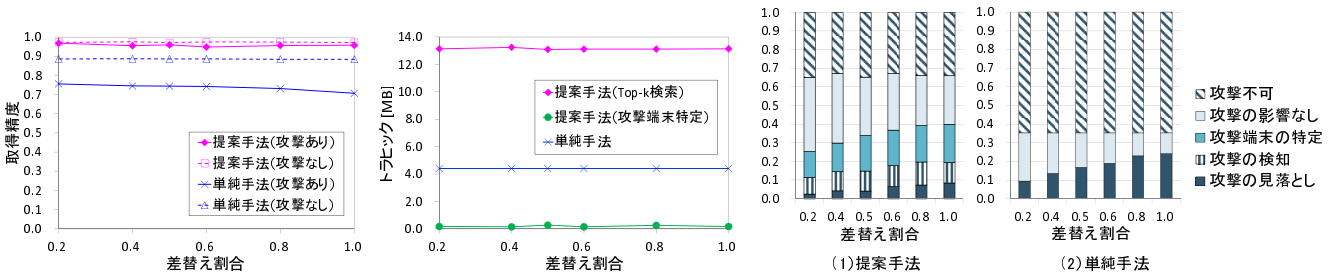
図 3(b) の結果より，提案手法および単純手法において，要求データ数が大きくなるとトラヒックが大きくなる．これは，要求データ数が大きくなると，返信するクエリ応答のサイズが大きくなるためである．提案手法は，単純手法と比べて，単純手法と比べて 2 倍以上のトラヒックが生じる．これは，単純手法では，クエリ応答を親端末のみに送信するのに対して，提案手法では，クエリ応答を親端末および隣接端末に送信しており，衝突が多く発生することにより，再送回数が多くなるためである．一方，提案手法において，攻撃端末特定手法により発生するトラヒックは非常に小さいことが分かる．これは，攻撃端末特定手法で送信するメッセージはデータを含んでおらず，データの送信を行う Top-k 検索手法に比べて非常に小さなトラヒックで実現できるためである．また，要求データ数が大きくなると，攻撃端末特定手法におけるトラヒックが若干大きくなる．これは，要求データ数が大きくなることで，図 3(c) に示すように，攻撃を検知できる場合が増加するためである．なお，攻撃を受けた場合でも，送信するデータ数やデータサイズは同じであるため，Top-k 検索時のトラヒックは提案手法，単純手法ともに攻撃の影響はない．

図 3(c) の結果より，どちらの手法も要求データ数が大きくなると，攻撃あり (「攻撃端末の特定」，「攻撃の検知」，および「攻撃の見落とし」) の割合が大きくなる．これは，要求データ数が大きくなると攻撃端末が差し替えるデータ数 (つまり，$\lceil 0.5 \cdot k \rceil$ 個) が増加し，クエリ発行端末で検索結果に含まれるデータを受信できない可能性が増加するためである．ここで，提案手法では，攻撃を検知できたが，攻撃端末を特定できない場合 (凡例における，攻撃の検知) が存在する．これには，問合せメッセージを送信する際にリンク切断が発生した場合や，問合せメッセージの送信経路上に問合せ先以外の攻撃端末の候補の端末が存在し，問い合わせできなかった場合である．さらに，要求データ数が大きくなると，提案手法において，攻撃の見落とし割合が増加する．これは，要求データ数の増加により，パケットロスが頻繁に発生し，データを複数の隣接端末から受信できず，攻撃を検知できないためである．なお，攻撃不可の割合は攻撃端末が他の端末からデータを受信できる場合が多いほど低くなる (攻撃の機会が増える) ため，複数端末へ返信を行う提案手法の方が，単純手法に比べて割合が小さい．

5.3　差替え割合 h の影響

差替え割合 h を変化させたときの結果を図 4 に示す．これらの図において，横軸は差替え割合 h を表し，縦軸は，図 4(a) は取得精度，図 4(b) はトラヒック，図 4(c) は，攻撃端末の特定割合を表す．

図 4(a) の結果より，単純手法では，差替え割合が大きくなると，攻撃の影響を受け取得精度が低下している．一方，提案手法は差替え割合が大きくなっても高い取得精度を維持できている．これは，単一経路で送信している単純手法では，差替え割合が大きくなると，攻撃の影響を受けやすいためである．一方，提案手法では複数経路でクエリ応答を返信しており，差替え割合に関わらず，必要なデー



(a) 取得精度　(b) トラヒック　(c) 攻撃端末の特定割合

図 4　差替え割合 h の影響

Fig. 4　Effect of h

タを取得できている．

図 4(b) の結果より，提案手法，単純手法ともに，差替え割合が大きくなってもトラヒックはほとんど変化しない．これは，データの差替えによる攻撃では，送信するデータ数やデータサイズが変わらないためである．

図 4(c) の結果より，データ差替え割合が大きくなると，提案手法において，攻撃端末の特定割合が大きくなる．一方，単純手法では差替え割合が大きくなると，攻撃の見落とし割合が大きくなる．これらは，差替え割合が大きくなることで，攻撃端末によって差し替えられるデータ数が多くなり，クエリ発行端末で検索結果に含まれるデータを取得できない場合が増加したためである．特に，提案手法では，攻撃を受けたクエリ応答を受信するほど攻撃を検知しやすく，それに伴って，攻撃端末の特定割合も増加する．

6. おわりに

本稿では，アドホックネットワークにおけるデータ差替え攻撃を考慮した Top-k 検索手法および攻撃端末特定手法を提案した．提案手法では，クエリ応答を複数の経路によって送信することで，取得精度を維持する．また，クエリ応答の転送経路をメッセージに添付して送信することで，クエリ発行端末は，データ差替え攻撃を検知することができる．さらに，この情報を用いて，攻撃端末候補を絞り込み，攻撃端末候補のデータの送信先端末にデータの受信状況を問い合わせることで，攻撃端末を特定する．シミュレーション実験の結果から，提案する Top-k 検索手法は，複数経路でクエリ応答を送信することで，取得精度を維持でき，さらに攻撃端末特定手法により，攻撃端末を高い確率で特定できることを確認した．

今後の課題として，本稿では，データ差替え攻撃が検索結果にどの程度影響を与えるのか，および一度のクエリでどの程度攻撃端末を特定できるのかを検証するために，攻撃端末を 1 台と想定した．しかし，実環境では複数台の攻撃端末が存在することも考えられる．単純には，返信データの転送経路を増やせばいいが，トラヒックが増大してしまう．そのため，トラヒックを抑えつつ，データを転送する手法が必要となる．さらに，攻撃端末が複数存在する場合，攻撃端末を特定することも難しくなる．そのため，一度の Top-k 検索だけではなく，複数のクエリを考慮して，攻撃端末を特定する手法について検討する予定である．

謝辞　本研究の一部は，文部科学省研究費補助金・基盤研究 S(21220002)，基盤研究 B（24300037），および特別研究員奨励費（24-293）の研究助成によるものである．ここに記して謝意を表す．

参考文献

[1] Amagata, D., Sasaki, Y., Hara, T. and Nishio, S.: A Routing Method for Top-k Query Processing in Mobile Ad Hoc Networks (2013).

[2] Camp, T., Boleng, J. and Davies, V.: A survey of mobility models for ad hoc network research, *Wireless communications and mobile computing*, Vol. 2, No. 5, pp. 483–502 (2002).

[3] D'Souza, R. and Varaprasad, G.: Digital Signature-Based Secure Node Disjoint Multipath Routing Protocol for Wireless Sensor Networks, *IEEE Sensors Journal*, Vol. 12, No. 10, pp. 2941–2949 (2012).

[4] Hagihara, R., Shinohara, M., Hara, T. and S.Nishio: A message processing method for top-k query for traffic reduction in ad hoc networks, *Proc. Int. Conf. on Mobile Data Management*, pp. 11–20 (2009).

[5] Kalnis, P., Ng, W., Ooi, B. and Tan, K.-L.: Answering similarity queries in peer-to-peer networks, *Information Systems*, Vol. 31, No. 1, pp. 57–72 (2006).

[6] Malhotra, B., Nascimento, M. A. and Nikoladis, I.: Exact top-k queries in wireless sensor networks, *IEEE Trans. Knowledge and Dara Engineering*, Vol. 23, No. 10, pp. 1513–1525 (2011).

[7] Papadimitratos, P. and Haas, Z.: Secure routing for mobile ad hoc networks, *Proc. of the SCS Commnication Networks and Distributed Systems Modeling and Simulation Conference*, pp. 193–204 (2002).

[8] Sasaki, Y., Hara, H. and Nishio, S.: Two-phase top-k query processing in mobile ad hoc networks, *Proc. Int. Conf. on Network-Based Information System*, pp. 42–49 (2011).

[9] Wood, A. and Stankovic, J.: Denial of Service in Sensor Networks, *IEEE Computer*, Vol. 35, No. 20, pp. 54–62 (2002).

[10] Wu, M., Xu, J., Tang, X. and Lee, W.: Top-k monitoring in wireless sensor networks, *IEEE Trans Knowledge and Data Engineering*, Vol. 19, No. 7, pp. 962–976 (2007).