IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727
PP 79-85
www.iosrjournals.org

Identification of Malicious Node in Top-K Query Processing In Manet

Nayana Santhosh[1], Arun R [2], Ancy A[3]
[1]*Department of CSE, SNGCE, Kadayiruppu, Kerala, India* nayanasanthosh12@gmail.com
[2]*Department of CSE, SNGCE, Kadayiruppu, Kerala, India* csarunr@gmail.com
[3]*Department of CSE, SNGCE, Kadayiruppu, Kerala, India,* anilan.ancy@gmail.com

Abstract: *It is effective to retrieve only the most important data from the large amount of information available in the MANET. Top-K query processing can be used to retrieve the most important data, in which the data items are assigned with a particular score based on their attribute values and up on issuing a query the data items with highest scores are retrieved. If malicious nodes are present, it may replace the high score data with lower ones which affect the accuracy of the query result. So, in this paper it is discussing a method to identify the malicious nodes that perform data replacement and identity replacement attacks. The query result is send through multiple routes along with the path information. The nodes in the attack path are considered as the candidate malicious node, from which the malicious node is identified by enquiring about information on data items send from candidate nodes. And also, a unique value is assigned to each score in a node to identify the malicious node that replaces the identity of the node possessing higher scores. So by this method we can identify the malicious nodes and maintain accuracy of the query result.*

Keywords – *Top-K query processing, Malicious node, Data replacement attack, Identity replacement attack.*

I. Introduction

Mobile Adhoc Network (MANET) is attaining greater momentum in present scenario. In MANET, since the mobile nodes are moving randomly they change their topology dynamically, so they do not require pre-existing base stations. Because of this nature they are required in wide variety of applications. Query processing is a major concern in the area of MANET. Due to limitation in the network bandwidth it is essential to done the query processing efficiently. So to perform query processing efficiently, we should retrieve only the most important data items, which will result in traffic reduction. To achieve this, we can use top-k query processing in which the data items in the network are assigned with a score and upon issuing a query it will retrieve only the data items with highest score.

Normally top-k query processing is done as follows: a node issues a query. This query is flood on the network and up on getting the query the nodes in the network will respond with the top k results in them to the query issuing node. This result in great traffic and the query result may also include data those which are not exactly the top ones. So it is necessary to perform the top-k query processing efficiently in MANET since it is a crucial requirement in many interactive environments that contain massive data.

Another factor that affects the efficiency of the top-k query processing is the presence of malicious nodes. If malicious nodes are present in the network, it may replace the top scored data items with low scored ones which decreases the accuracy of the query result. So a method to identify the malicious node in top-k query processing is required.

In this paper, it is considering two attacks, the data replacement attack and identity replacement attack. In data replacement attack the malicious node present in the network replaces the high score data with low score data items. Where as in identity replacement attack the malicious node replaces its identity with the identity of a node having higher score. So here to maintain the accuracy of the query result, the reply of the nodes are send through multiple routes instead of sending through a single route. Also each score in a node is assigned with a unique id for the detection of identity replacement attack. When a node sends the reply this id is also attached with it. This id is verified by the query node to check whether any malpractice is occurred or not. To identify the malicious node that performs data replacement attack, the query issuing node first finds the path in which the attack is occurred and the nodes in that path is considered as candidate malicious nodes. From these candidate nodes the malicious node is identified using received message information and then requests information on the data item send by these candidates. In these ways the malicious nodes in the network can be identified and the accuracy of the query result can be maintained.

II. Related Work

There are different methods for top-k query processing. In [1], [2], the authors have proposed methods to reduce traffic in unstructured P2P networks, by reducing the number of messages by nodes. The benefits of best match/top-k [1] queries s discussed in this incase of distributed peer-to-peer information infrastructures and also considers a method to extend the limited query processing in existing peer-to-peer networks by allowing the distributed processing of top-k queries, and also maintaining a minimum data traffic. There is also a technique [2] which uses the fuzzy nature to avoid flooding the network with messages. Several optimization techniques for single and multiple-attribute queries are also considered in this.

There are methods to reduce energy consumption and traffic in wireless sensor networks, in which the sensor nodes are capable to filter unnecessary data items. However, these methods do not protect against DRA, and are not suitable for use in MANETs, because they are not adapted to node mobility. One among them is a history-based approach [3] that optimizes top-k query processing in sensor network and use a Threshold-Estimate-Prune-Query algorithm. In this, the consumption of energy is reduced by pruning the unwanted sub-queries and using the cached data the query messages handled properly. Subset of the sensor network will respond the query. When the environment is changed, to get accurate data a Local-Expand-Query method is used.

Another method [4] considers the exact top-k query problem in wireless sensor networks, in which a query asks for the top reported values and the nodes which reported them. The primary contribution in this is EXTOK, which is a topology-independent new filtering-based algorithm for processing exact top-k queries. There is a query reevaluation algorithm [5] that can handle concurrent sensor updates. To reduce the probing cost it also includes optimization techniques. It consists of the design of a skewed filter setting scheme, which aims to balance energy consumption and prolong network lifetime. Two filter update strategies, namely, eager and lazy, are also proposed to favor different application scenarios.

Top-k query processing methods for MANETs, adapted to the node mobility, maintaining high accuracy of the top-k result and reducing traffic are also there.

There is a routing method [6] for top-k query processing in MANETs. In this method, each mobile node has a routing table consisting of scores of the data items in the network and the corresponding node identifiers, so upon issuing a query the query node refers its routing table and forward the query to the corresponding nodes to get the required data item. Also, each node updates its routing table and scores of data items according to the changing environment. However, these methods are not designed for environments in which malicious nodes exist, for example, the data items in the top-k result are sent back along a single route, and thus are vulnerable to DRA.

Another method for top-k query processing that reduce traffic and maintain accuracy is there. In this method [7] the candidate of data items included in the top-k result are reduced to avoid unnecessary data transmission. If a disconnection of a radio link occurs, it searches for an alternative path to transmit the reply message to the query-issuing node. In two-phase top-k query processing method [8] a threshold score is estimated and acquires data items with score equal or greater than the threshold. There is an alternate two-phase query processing method [9] in which the threshold score is estimated based on the scores of data items in the network which is collected beforehand. Then the query along with this threshold score is transmitted and the odes having score higher than or equal to this threshold value sends the reply. In this way, the proposed method can further reduce the traffic and also keep high accuracy of the query result.

Location based service is very important in MANET. There are two beacon-less k NN [10] query processing methods for reducing traffic and maintaining high accuracy of the query result in MANETs. In these methods, the query-issuing node first forwards a k NN query to the nearest node from the query point. Then, the nearest node from the query point forwards the query to other nodes close to the query point, and each node receiving the query replies with the information on itself. In this process, there are two different approaches: the Explosion (EXP) method and the Spiral (SPI) method. In the EXP method, the nearest node from the query point floods the query to nodes within a specific circular region, and each node receiving the query replies with information on itself. In the SPI method, the nearest node from the query point forwards the query to other nodes in a spiral manner, and the node that collects a satisfactory k NN result transmits the result to the query-issuing node. In the field of sensor network and MANET design, secure routing protocols protect against falsifications of data and DoS attacks, which disable services by disrupting communication. Secure routing protocols commonly employ data transmission along multiple routes, and data encryption using symmetric or public keys. In these methods, before sending data items, the source node determines multiple routes by which to safely send the data items.

A secure on-demand ad hoc network routing protocol, called Ariadne [11] prevents DoS and attackers that tampers the data. In [12] nine black hole detection methods within the scope of AODV routing protocol is considered. AODV is an on-demand routing protocol that discovers a route based on demand. Different black hole detection methods such as, Detection based on Path based method, based on learning automata, based on collaborative Bayesian Watchdogs, using fuzzy logic, using anomaly detection, using promiscuous mode are discussed in this.

Another method [13] designs a threat model for ad hoc routing and considers several attacks. It discusses the secure mechanism in five categories: solutions based on asymmetric cryptography; solutions based on symmetric cryptography; hybrid solutions; reputation-based solutions; and a category of add-on mechanisms that satisfy specific security requirements. A robust and energy efficient multipath routing protocol REER [14] that uses the residual energy, node available buffer size, and Signal-to- Noise Ratio (SNR) to predict the best next hop through the paths construction phase is discussed which examines two methods of traffic allocation; the first method uses a single path among the discovered paths to transfer the data message, when this path cost falls below a certain threshold, it then switches to the next alternative path. The second method is to split up the transmitted message into number of segments of equal size, add XOR-based error correction codes, and then transmit it across multiple paths simultaneously to increase the probability that an essential portion of the packet is received at the destination without incurring excessive delay.

III. Proposed System

In the proposed top-k query processing method, the query-issuing node first floods a query over the entire network, and each node receiving the query stores information. Then, each receiving node replies with data items with the k highest scores along with the unique id assigned to two neighbor nodes. In addition, each node includes, in its reply message, information on the reply message forwarding routes which consist of pairs of sender node and next node IDs. Based on this attached information, the query-issuing node can detect an attack occurring along a reply message route.

In the proposed method for identifying a malicious node, a query-issuing node that detects an attack narrows down the malicious node candidates based on information on the reply message routes included in its received reply messages. Then, the query-issuing node determines whether a given reply message sent back by a malicious node candidate includes replaced data items or not, by sending inquiries to nodes receiving reply messages from this candidate. In this way, the query-issuing node can identify the malicious node.

3.1. Top-k Query processing

The query-issuing node forwards the query by flooding over the entire network. The query consists of the node identifier of the query-issuing node (Query node ID), the query identifier of the query (Query ID), the number of requested data items (k), the query condition, and a list of the node identifiers of nodes on the path along which the query message is to be transmitted (Query path). The, query issuing node Mp transmits a query message whose Query path includes its identifier, Mp, to its neighbor nodes. A node, Mq, which receives the query, transmits a query according to the following steps:

```
1. If Mq receives the query for the first time then
2. . Store Query path and hop count as its Parent Query path
3.  Store the nodeID at the end of Query path as its parent
4.  Set RD for replying data items
5.  Add Mq's nodeID to the end of Query path and send the query to neighbor nodes
6.  Else
7.  Store Query path and hop counts as its Neighbor Query path
8. Store the nodeID at the end of Query path as its neighbor
9.  end if
```

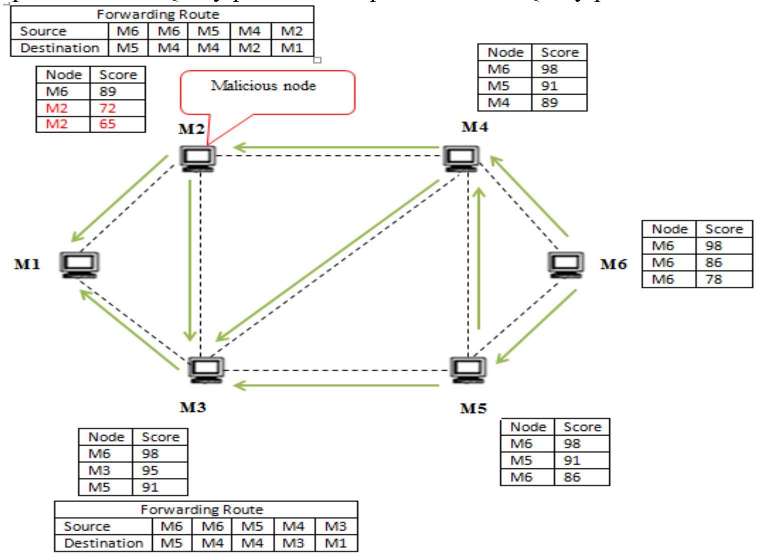
Here, hop count denotes the number of hops to the query-issuing node. Then, Mq sets a waiting time for reply (RD). As hop count increases, RD decreases. When Mq receives the query later again, it stores the ID of the query sender node as its neighbor node, as well as, the Query path and the number of hops.

When its RD has passed, each node sends back a reply message, which includes its own node identifier (Sender nodeID), the identifier of the next node along the reply route (Dest nodeID), an encrypted id (Enc ID), a list of the data items (including their scores) and the node identifiers of the nodes possessing them (Data list), and a list summarizing the reply message routes, i.e., a list of the pairs of sender and next node identifiers

(Forwarding Route). Following are the steps to send the reply to a message. The reply will be send after the RD has elapsed.

1. Select the Neighbor with minimum hopCount as the DestNode
2. If more than one node with same minimum hopCount select the one with least overlap with parent Query path
3. Add the local top-k result to REP and send the reply REP to the parent and the selected neighbor node
4. When a node receives the REP, send ACK to the sender node of REP
5. If RD is not elapsed, store the REP
6. Else if after RD and Mr receives a data item with higher score than with the kth-highest score among data items already sent then
7. Send REP including new local top-k result to parent node and DestNode
8. Otherwise discard it
9. if Mr does not receive ACK from its parent or DestNode by waiting time for retransmission and the number of retransmissions < R then
10. Resend REP to parent or DestNode correspondingly
11. else if the number of retransmissions > R then
12. if Mr has sent REP to all Neighbor then discard REP
13. else if send REP to the Neighbor whose Neighbor Query path includes DestNode then
14. End

Here, node Mr sends a reply message when its RD has passed. Here, REP denotes a reply message and REP.FR denotes the forwarding route list and R denotes the maximum number of reply messages to be re-sent. Mr selects from the stored Query path, the next nodes, as its parent node and a neighbor node with the least hop count and least overlap between its Query path and the parent node's Query path.



Forwarding Route					
Source	M6	M6	M5	M4	M2
Destination	M5	M4	M4	M2	M1

Node	Score
M6	89
M2	72
M2	65

Node	Score
M6	98
M5	91
M4	89

Node	Score
M6	98
M6	86
M6	78

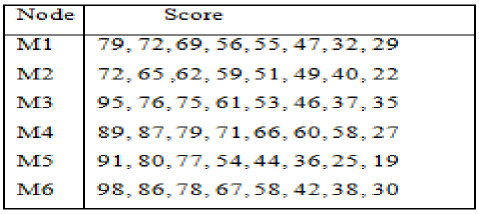
Node	Score
M6	98
M3	95
M5	91

Node	Score
M6	98
M5	91
M6	86

Forwarding Route					
Source	M6	M6	M5	M4	M3
Destination	M5	M4	M4	M3	M1

fig 1: Example of reply forwarding

An example of reply forwarding is shown in (fig 1) where k = 3 and M1 is the query-issuing node. Table1 shows the scores of data items retained by each node.



Node	Score
M1	79, 72, 69, 56, 55, 47, 32, 29
M2	72, 65 ,62, 59, 51, 49, 40, 22
M3	95, 76, 75, 61, 53, 46, 37, 35
M4	89, 87, 79, 71, 66, 60, 58, 27
M5	91, 80, 77, 54, 44, 36, 25, 19
M6	98, 86, 78, 67, 58, 42, 38, 30

Table 1: score of data items

Since M1 floods the query first, each node knows its neighbors and hop counts from M1. M6 having the largest hop count sends the reply first. M6 incorporates the local top-k result, which consists of its data items with the three highest scores (score: 98, 86, 78), in a reply message, and sends the reply message to its parent node, M4, and neighbor node, M5. In this case, the list of sender and next node pairs included in the reply message sent to M4 is {(M6, M4)}, and the list of similar pairs in the reply message sent to M5 is {(M6, M5)}. The other nodes perform the same procedure. Finally, the query-issuing node, M1, having received reply messages from M2 and M3 acquires the global top-k result (score: 98, 95, 91), even though M2 attacks.

3.2. Detection of Attack

After the query-issuing node, Mp receives all the reply messages, it detects a DRA according, to the following steps.

```
Initialize the SendRoute to zero
for each Top-k Result in each REP
If a node ID of a node processing a data item in Top-k Result is included in REP.FR but not in REP.Data then
Insert a route from the node with the missing data item to the query-issuing node into SendRoute
if SendRoute is not null then an attack is detected
End
```

Here, Top-k Result denotes the data items with the k highest scores, acquired by the query issuing node, REP.Data and REP.FR respectively denote the data list and forwarding route included in the reply message, REP, and SendRoute denotes the set of node identifiers along the route from the node possessing a given data item to the query-issuing node. If the nodes which have data items in the top-k result are included in SendRoute , but the data items in the top-k result are not included in REP.Data, the query-issuing node detects an attack, and initiates the malicious node identification process. If the query-issuing node does not detect an attack, it completes the top-k query processing.

An example in which the query-issuing node, M1, detects an attack is shown in (fig 1). Here, the malicious node, M2, replaces data items having the highest and second highest scores among received data items (score: 98, 91), with data items it possesses, whose scores are lower than the third highest data item it received, and sends the now corrupted data items (score: 89, 72, 65). Receiving the reply message from M2, M1 can know that M5 and M6, which have data items included in the top-k result, are on the forwarding route included in the reply message from M2. However, the data items received from M2 do not include data items of the top-k result which M5 and M6 possess (score: 98, 91). Therefore, the query issuing node detects that the data items it received from M2 have been corrupted (attacked), and knows precisely which have been replaced.

3.3. Identification of Malicious Node

After detecting an attack, the query-issuing node identifies the malicious node. First, the query-issuing node narrows down the candidates for the malicious node, and identify the malicious node by making respective inquiries. The query-issuing node narrows down the malicious node candidates by using SendRoute. Consider the node ID's in the SendRoutes and if this node ID is also included in the other SendRoute also the insert that node to the list of Candidate nodes. The nodes included in SendRoute, whose data items are corrupted (by the malicious node), are all possible attackers. Therefore, the query-issuing node recognizes these nodes as malicious node candidates. If there is only one node in the candidate list, then return it as the malicious node else inquire about the candidate malicious nodes. The procedure for inquiring about information on data items sent from malicious node candidates is as follows:

```
for each node in Candidate
Send MNI-INQ to MDest to ask data items that Candidate[i] sent
if MDest receives MNI-INQ then
Send MNI-IREP including scores of data items sent by Candidate[i] to Mp
end if
 if Mp receives MNI-IREP then
if scores includes the score of the missing data items in global Top-k result then
 return Candidate[i − 1]
end if
end for
```

Here, MNI-INQ denotes an inquiry message, which contains the query-issuing node identifier, the node identifier of the destination node for the inquiry message (MDest), the set of malicious node candidate identifiers (Candidate), and the forwarding route of the inquiry message from the query-issuing node to the destination node (InqRoute). MDest denotes the destination node to which Candidate[i] (ith candidate) has sent a reply message. MNI-IREP denotes a message sent in reply to the inquiry message, which contains the scores of the data items, and the identifiers of nodes possessing these data items, which are included in the reply message received from the Candidate node. The query-issuing node, Mp, finds InqRoute which does not include the malicious node candidates, and sends an inquiry message to nodes at the top of the InqRoute, in ascending hop count order of the malicious node candidates. A MNI-inquiry message is not sent to nodes whose hop count is one, because the query-issuing node receives reply messages directly from such nodes . After Mp has received a reply message to its MNI-inquiry, it identifies the malicious node. Specifically, if the data items sent by Candidate[i] do not include the replaced data items, Mp identifies the candidate with a hop count of one less than that of Candidate[i] (i.e., Candidate[i − 1]), as the malicious node , and completes the procedure.

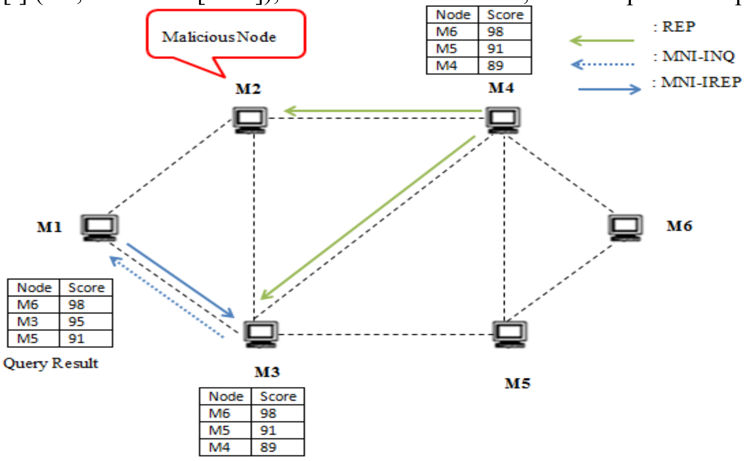


fig 2: Identification of Malicious Node

An example of how the query-issuing node identifies the malicious node after detecting an attack is shown in (fig 2), in the cases where the data items included in the top-k result possessed by M5 and M6, are not included among the data items in the reply message from M2. The query-issuing node, M1, determines each route along which a reply message is transmitted from M5 or M6 to M1 via M2 using the reply message received from M2. M1 designates M2 and M4 as the malicious node candidates, since they are included in the SendRoute. Therefore, M1 sends a MNI-inquiry message to M3, since malicious node candidate, M4, has sent a reply message to M3, and M3 is not itself a malicious node candidate. Since the hop count of M2 is one (i.e., M1 receives replies directly from M2), M1 does not send a MNI-inquiry to M2. When M3 receives the MNI-inquiry message, it sends the scores of the data items (score: 98, 91, 86) sent to it by M4, to the query-issuing node, M1. When M1 receives this reply message, it confirms whether M4 has sent data items included in the top-k result (score: 98, 91). Finally, since M4 has sent the correct data items and M2 has not, M1 identifies M2 as the malicious node.

3.4. Identification of Identity Replacement Attack

Highly secure communications and data exchanges are essential in top-k query processing. Therefore, each node has the public key of all nodes in the network. When a node replies with data items (i.e., sends a reply message), it encrypts [15] [16] the data items using the public key of the destination node to avoid intermediate nodes modifying and reading the data items. In addition, the node sends the reply message (including the encrypted data items) to some of its neighbors after encrypting the message using the neighbors' public key. This is to ensure a secure communication with neighbors and avoid others overhearing the message. On the other hand, when a node sends a query, it broadcasts the query without encryption since a query is not aimed to send to specific nodes but should be sent to all neighbors.

If the malicious node replaces the identity of a node having higher score with its identity, it can be found out by providing a unique identity for each score in a node. When a node replies to the query issuing node a unique id is also attached along with it. This unique id is the encrypted value of combination of node ID and corresponding score. If a malicious node is present in the network and replaces its identity with the identity of a node having higher score, an identity replacement attack is occurred. When a reply reaches at the query issuing node, the encrypted value is decrypted and verified with the identity of the sender node. If there is a mismatch in the node ID's then an attack is detected. So even if the malicious node replaces the identity, it can be found out. From the reply itself we can find out the malicious node that replaces the ID of the node with higher score.

IV. Conclusion and Future Work

In this paper, we present a method for the identification of malicious nodes in top-k query processing in MANET. Here, two types of attack are discussed, data replacement attack and identity replacement attack. The data replacement attack is the one in which the malicious node replaces the data items having higher score with its data item that having a lower score. In identity replacement attack instead of replacing the data the identity of the node containing higher score is replaced.

In the top-k query processing method, the query results are sending through multiple routes to avoid attack. In the malicious node identification method, the query-issuing node first finds the attack path and then marks the node in that path as candidate nodes. From these candidate nodes the malicious node is found using received message information and then requests information on the data item send by these candidates. In these ways the malicious nodes in the network can be identified and the accuracy of the query result can be maintained. And also, the score and identity of the nodes are encrypted and send along with the reply. This is to found out if the malicious node replaces the identity of a node which having higher score with its own identity. In this way we can identify whether there occurs an identity replacement attack.

The proposed methods the existence of only one malicious node in the entire network is considered. But, in real environments there may be multiple malicious nodes. In this case, the number routes along which the reply is send will increase which result in an increased traffic. So a mechanism to overcome this can be considered as a future work.

References

[1] W.-T. Balke, W. Nejdl, W. Siberski, and U. Thaden. "Progressive Distributed Top-k Retrieval in Peer-to-Peer Networks," Proc. Int'l Conf. on Data Engineering, pp.174–185, 2005.

[2] P. Kalnis, W.S. Ng, B.C. Ooi, and K.-L. Tan, "Answering Similarity Queries in Peer-to-Peer Networks," Information Systems, vol.31, no.1, pp.57–72, 2006.

[3] Qunhua Pan', Shuwang Li2 ,Minglu Li', Min-You Wu', "Energy-Efficient Top-k Query Processing in Dynamic Sensor Networks," Conf. on Information and Knowledge Management, pp.329-338, 2009.

[4] B. Malhotra, M.A. Nascimento, and I. Nikoladis, "Exact Top-k Queries in Wireless Sensor Networks," IEEE Trans. Knowledge and Data Engineering, vol.23, no.10, pp.1513–1525, 2011.

[5] M. Wu, J. Xu, X. Tang, and W.-C. Lee, "Top-k Monitoring in Wireless Sensor Networks," IEEE Trans. Knowledge and Data Engineering, vol.19, no.7, pp.962–976, 2007.

[6] D. Amagata, Y. Sasaki, T. Hara, and S. Nishio, "A Routing Method for Top-k Query Processing in Mobile Ad Hoc Networks," Proc. IEEE Int'l Conf. on Advanced Information Networking and Applications, 2013.

[7] R. Hagihara, M. Shinohara, T. Hara, and S. Nishio, "A Message Processing Method for Top-k Query for Traffic Reduction in Ad Hoc Networks," Proc. Int'l Conf. on Mobile Data Management, pp.11–20, 2009.

[8] Y. Sasaki, R. Hagihara, T. Hara, and S. Nishio, "A Top-k Query Method by Estimating Score Distribution in Mobile Ad Hoc Networks," Proc. Int'l Workshop on Data Management for Wireless and Pervasive Communications, pp.944-949, 2010.

[9] Y. Sasaki, T. Hara, and S. Nishio, "Two-Phase Top-k Query Processing in Mobile Ad Hoc Networks," Proc. Int'l Conf. on Network-Based Information System, pp.42–49, 2011.

[10] Yuka Komai, Yuya Sasaki, Takahiro Hara and Shojiro Nishio," K NN Query Processing Methods in Mobile Ad Hoc Networks", IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 13, NO. 5, pp.1090-1103 MAY 2014

[11] Y.-C. Hu, A. Perrig, and D.B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," Proc. Int'l Conf. on Mobile Computing and Networking, pp.23–26, 2002.

[12] S. Kurosawa, H. Nakayama, N. Kato, A. Jamalipour, and Y. Nemoto, "Detecting Blackhole Attack on AODV-Based Mobile Ad Hoc Networks by Dynamic Learning Method" Int'l Journal of Network Security, vol.5, no.3, pp.338–346, 2007.

[13] P. Papadimitratos and Z.J. Haas, "Secure Routing for Mobile Ad Hoc Networks," Proc. SCS Communication Networks and Distributed Systems Modeling and Simulations Conf., pp.193–204, 2002.

[14] B. Yahya and J. Ben-Othman, "REER: Robust and Energy Efficient Multipath Routing Protocol for Wireless Sensor Networks," Proc. IEEE Global Telecommunications Conf. , pp.1–7, 2009.

[15] R.J. D'Souza and G. Varaprasad, "Digital Signature-Based Secure Node Disjoint Multipath Routing Protocol for Wireless Sensor Networks," IEEE Sensors Journal, vol.12, no.10, pp.2941–2949, 2012.

[16] W. Galuba, P. Papadimitratos, M. Poturalski, K. Aberer, Z. Despotovic, and W. Kellerer, "Castor: Scalable Secure Routing for Ad Hoc Networks," Proc. IEEE Int'l Conf. on Computer Communications, pp.1–9, 2010.