Private and Mobile Inter-Network Routing for Wireless Sensor Networks and Internet of Things

Mohsen Saberi

Department of Computer Engineering, Bozorgmehr University of Qaenat, Iran

***Abstract*:** In the last few years, using the Internet of Things has been expanded in many areas, such as environmental monitoring, industries, and smart home. Since the Internet of Things has a direct relation to human life, its security is of paramount importance. Therefore, the communication between the nodes should be secured and the valuable private information should be kept private so that the attacker cannot detect the network structure. This article provides a protocol that can handle routing privately. To do this, we use the data structure called Spatial Bloom Filter (SBF). In addition, the proposed protocol uses random identifiers instead of IP addresses, so that an attacker cannot collect network structure information and location of nodes from IP addresses. Using a homomorphic encryption scheme, the protocol prevent attackers from retrieving valuable network information, if they can infiltrate to one or more network nodes. Also, since almost all nodes in the internet of things are mobile, the structure of networks and subnets is constantly changing. The proposed protocol has the ability to manage to route in networks with a dynamic structure.

***Keywords*:** Private routing; Mobile Routing; Internet of Things; privacy-preserving protocols; Spatial Bloom Filters; Reliable Mobile Management

1. Introduction

In the last decade, the Internet of Things has expanded a lot in our lives, which is one of the capabilities brought by wireless networks [1, 2, 3, 4]. IoT has entered in many areas and has developed its own technologies, services, and standards [5, 6, 7]. From a logical point of view, an IoT system can be considered as a set of intelligent tools that are put together to achieve a specific goal. From a technical point of view, the use of IoT, based on its purpose, involve communication architectures, technologies, design techniques, and various processes. For example, an IoT system can use a wireless sensor network (WSN) to collect environmental information and using an application running on mobile devices, analyze the information. There are also middlewares for quick, abstract, and easy access to resources and services. The middlewares can also be constructed using various technologies such as cloud or peer-to-peer systems [8].

Undoubtedly, the use of this heterogeneous structure for IoT systems increases the likelihood of harm and security threats. Because IoT is very close to human life and currently exists in cars and electronic devices, IoT security attacks have a far greater impact than other technologies. In addition, due to the power and memory limitations of the devices, the usual methods of controlling the security and privacy that are used on the Internet, cannot be directly applied to the IoT technology. Also, for IoT to be used by individuals, it must raise the security, privacy, and trust to a credible level [2, 9, 10, 11, 12]. Another point that matters in the IoT is to be able to adapt the infrastructure used based on changes occurring in the system or environment. As a result, a security or privacy method should be flexible.

In this paper, we will use the Spatial Bloom Filter and homomorphic encryption data structures to design a protocol for anonymous routing. The purpose of the protocol is to preserve the privacy of the network structure which is constructed in a combination of several subnets that are connected to each other. The protocol encrypts the communication between the nodes, hides the structure and topology of the network, hides the address and location of the transmitter and receiver nodes from the middle nodes and the routing layer, and by changing the structure and displacement of nodes from a subnet to another subnet will update routing information. With these features, privacy and security for network nodes are provided while the movement ability of nodes will not be eliminated. Therefore, the network will not allow attackers to control it.

2. Related Work

As the approach proposed in this paper is based on two core technological areas, private and mobile routing, and reliable mobility management, this section surveys the main concepts involved as well as existing research.

2.1 Private and Mobile Routing

Privacy preservation for IoT devices and users is a key issue in IoT. Even with the existing authentication approaches and cryptographic mechanisms in place to safe guard users' privacy in IoT networks, issues like heterogeneity of IoT networks, limited battery capacity of devices and the devices' resource constraints in terms of available memory cripple the communication. As a result, multiple devices in IoT network end up not utilizing in an optimal manner the available authentication and cryptographic mechanisms. This clearly shows the need for better secure systems for the IoT networks. The US Federal State Commission (FTC) identified this and have announced the need to secure the IoT ecosystem after security violation was reported for the TRENDNet IP camera in 2012 where live footage from thousands of TRENDNet security cameras have been penetrated, permitting web users to access live video footage without requiring any password. Similarly, such security requirements have also been reported by the European Union Data Protection WP29 committee.

Encryption protocols can be used to maintain privacy in communications. But these protocols should be selected based on limited power and memory of the devices. Also, since determining the location of nodes in the network can impact on privacy, a strategy should be developed to prevent access to nodes' locations. An IP address has the location information of a node within itself. Therefore in order to

keep the location of nodes private, the address of the nodes should be kept hidden. The result of hiding the addresses is maintaining the structure and topology of the network and subnets hidden [13]. This is necessary to prevent attacks that target the base station. The base stations are the node that collects data from nodes. In fact, the failure of a base station means that the subnet that is connected to it will not be in access. There are several ways to make a base station hidden from the adversaries. A basic strategy to achieve context privacy is to use flooding and transmissions of fake or dummy packets, which make network traffic observation more difficult [14, 15]. However, this solution introduces significant overheads in the communication, and can reduce the efficiency of the IoT network. More complex strategies are normally based on some flavor of anonymity, including the use of random walks to route packets anonymously [16, 17]. This strategy can be used to transfer packets in the network anonymously. The random walks method is designed in a variety of ways [18, 19, 13]. while GROW (Greedy Random Walk) [20] introduced a two-way random walk, from both source and destination, that can reduce the chance of an eavesdropper being able to collect location information. Finally, layers of encryption can be used to protect the information at each hop in the walk. In addition, more advanced technologies have been created to protect information. Palmieri et al. use Spatial Bloom Filter in [21] to preserve privacy. But in their protocol nodes cannot move between subnets. Additionally, [21] assumed that the nodes start their job by having an ID and the network starts with a basic configuration.

More recently, more advanced anonymity techniques have been applied to the IoT. Black routing and node obscuring for IoT have been proposed by Chakrabarty et al. in [3]. Their strategy hides the source of network traffic via a token-based routing approach, while the destination is obscured by forwarding the packet beyond the final destination. However, to achieve anonymity of source-destination pairs, a minimum of 40% of the total IoT nodes in the path is needed, thus restricting application of this technique to more complex settings, where different IoT networks are interconnected. An onion routing protocol derived from Tor has also been designed for the Internet of Things scenario [15]. This strategy, however, requires IoT nodes to be able to perform complex computations, which may not always be possible in power and resource constrained scenarios.

2.2 Reliable Mobility Management

Lee et al. [22] had investigated a simulation research on analytical comparison of IPv6 mobility management protocols handover scheme. The researchers compared the Host-Based mobility management protocols and Network-Based mobility management protocols to identify the optimized routing protocol for mobile network. Vasu et al. [23], had investigated a survey and comparative analysis for MIPv6 protocols. The researchers had performed various mobility management protocols in terms of handover latency and the number of hops is needed to evaluate these protocols. Sun et al. [24], had investigated the mobility management techniques for next generation wireless networks. The researcher had performed macro and micro mobility protocols in terms of handover performance. The macro and micro mobility protocols such as Mobile Internet Protocol version 6 (MIPv6), Fast Handover Mobile Internet Protocol version 6 (FMIPv6), Hierarchical Mobile Internet Protocol version 6 (HMIPv6) and Fast Handover for Hierarchical Internet Protocol version 6 (FHMIPv6) and Proxy Mobile Internet Protocol version 6 (PMIPv6).

3. Preliminaries

Before presenting the proposed routing protocol, it is necessary to provide explanations about the concepts and mechanisms used. First, Spatial Bloom Filter [25, 26] is explained. Then Homomorphic Encryption methods are presented.

3.1 Spatial Bloom Filters

A Bloom Filter (BF) is formed for a set of elements and can respond to requests for the membership of an element in the set without knowing about the elements for the set [27, 17]. If an element belongs to a set, its BF will respond positively to its membership queries. In the absence of an element in a set, the BF usually returns a negative response. There is a possibility of a false positive error for non-existence elements.

The SBF data structure is a modified version of BF which is used to store and control location information. Similar to BF, The data structure can also be used in privacy-preserving applications [25, 26]. Since in this article the data structure is intended to be used along the network and to check the membership of a node in a subnet, we define it by network terminologies. SBF examines the membership of a network node in several sets (subnets) Δ1, Δ2, ..., Δs, but it does not need to know the nodes of these subnets for this investigation. SBF is defined as follows: assume that Ɛ is the set of possible IDs for nodes which can be present in one of the subnets and S = {Δ1, Δ2, ..., Δs} is the set of all subnets under management. Each member within Δi is a member of Ɛ. Also, none of the two subnets Δi and Δj have a common member. Suppose O is a strict total order on the set S such that for i < j there is an inequality Δi < Δj. In addition, suppose H = {h1, h2, ..., hk} is a set of hash functions, each applies on a node ID of the network and its output is one of the values of the set {1,2, ..., m }. In this case, the Spatial Bloom Filter on (S, O) is a set of ordered pairs that are obtained as follows:

$$B\#(S,O) = \bigcup_{e\in\mathcal{E},h\in H}(h(e),i) \quad (1)$$

In such a way that for j > i there is no member e*ϵΔj which h(e*) = h(e).

An SBF can be represented as a vector b# which is formulated as follows.

$$b\#[i] = \begin{cases} 1 \ if \ (i,1)\in B\#(S,O) \\ 0 \ if \ (i,1)\notin B\#(S,O) \end{cases} \quad (2)$$

To build an SBF, all cells of b# are first set to 0. Assuming that the subnets Δi are arranged in accordance with the specified order O, we assign the elements belonging to Δ1 as inputs to each hash functions in H. Suppose that for the hash function h and the element e of Δ1, h(e) = i. In this case, the

value 1 will be set in cell i, which is the same as the index of Δ1. Upon completion of the elements Δ1, the computation will be done on the elements Δ2 and the value 2 is set in cells whose indexes are equal to the output of H functions. These steps will continue for all Δi subnets. As expected, a collision may occur, and subnet number with a higher index substitutes for a subnet with a lower index based on the processing order of the nodes. The algorithm is depicted in Figure 1.

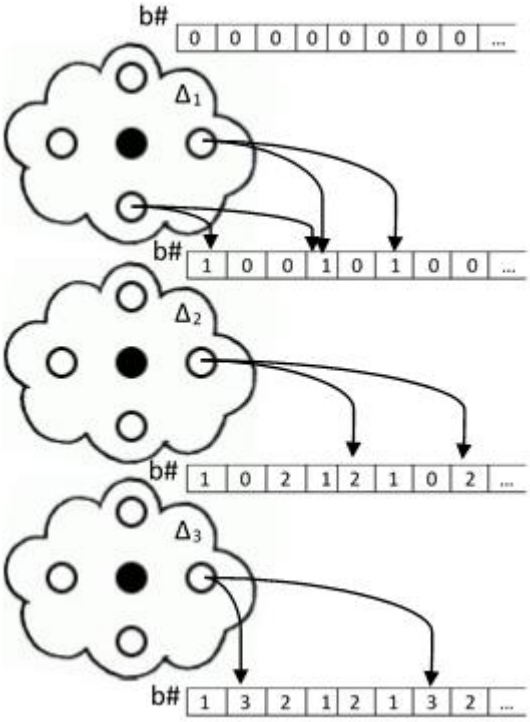

Figure 1. Subnets Δ1, Δ2 and Δ3 along with node IDs within them are used in order to construct a network b#. In this figure, two hash functions are used to transform the node IDs into index of b# vector in which index number of subnet is to be set. In the first step, the algorithm set value 1 in cells where those indexes are output of hash functions applied on node IDs of subnet Δ1.

Now, to confirm the existence of the value e in Δi, there must be two conditions:

$$\exists h \in H: b\#[h(e)] = i \text{ and } \forall h' \in H, h' \neq h, b\#[h'(e)] \geq i \quad (3)$$

Of course, the possibility of false positive should be considered.

Also, to confirm that a node e belong to none of the members of S, the output of one of the functions h for e must be equal to 0.

3.2 Homomorphic Encryption

We require the proposed protocol to use the homomorphic properties of encryption. A cipher has a homomorphic property when a computation can be done on the cipher-text without having to know the key, and thus without having the plaintext. An encryption scheme has a homomorphic additive property if the operator ⊕ applies to Enc(p1) and Enc(p2) and obtains the following result:

$$Dec(Enc(p1) \oplus Enc(p2)) = p1 + p2 \quad (4)$$

An encryption scheme also has a multiplicative homomorphic property if the operator • acts on Enc (p1) and p2 and obtains the following result:

$$Dec(Enc(p1) \bullet p2) = p1 * p2 \quad (5)$$

Paillier cryptosystem [28] is an encryption scheme that has both features of a homomorphic system. This article uses this cryptosystem. But it's possible to use a different cryptosystem as well, such as [29].

4. Architecture of Secure and Private Routing

Assume that we want to establish a connection between heterogeneous networks and create a larger network. These subnets are connected by some routers using a routing layer, and the task of the layer is to transfer packets between the networks. Each subnet is made up of several nodes connected through a gateway to the routing platform. Gateways can also be part of the routing platform. The gateways can be the same as collect stations. Our goal is to communicate between nodes in different subnets. We also want to prevent the attacker from accessing the information and position of any node. Specifically, we want an attacker not to detect the number of subnets and also cannot figure out in which subnet each node is located. So, we want to reach this level of security that each node needs only the IDs of those nodes that it what to communicate with. Since the ID is a random number, no information can be obtained from the ID of the destination node.

Each network node has an ID that is randomly generated. Therefore, unlike an IP address that contains information about the location of a node in the network, an ID does not show any information. The nodes of the network communicate with each other based on their ID, and similar to privacy-preserving protocols, like onion routing [30], they use encapsulation and tunneling in their different layers. The tunneling connection is established between the sending node and the gateway at the sending subnet, the sending gateway, and the routing layer, the routing layer and the gateway of the destination subnet, and ultimately between the end gateway and the destination node. Each node refuses to give additional information to the next node. For example, the gateway node of the source subnet does not send the source node ID to the routing layer. So the nodes after the source gateway will not have the source address information. Also, since the target gateway also does not know which nodes are the destination node, it is necessary to broadcast the packet in the subnet so that the destination node receives it. Since the communication is done securely, only the destination will be able to decrypt the packet information received with its secret key.

Each message sent in the network consists of a header and a payload. Within the header part, the routing information is placed and the destination encrypted message is laid inside the payload part.

Each node has its private and public keys. The private key is only available to the node. But the public key could be used by the network nodes using the IMCS presented in Section 4. If the public and private keys of the recipient are Pk(r) and Sk(r), respectively, the transmitter encrypts the message

using Pk(r) and sends it toward the receiver. The receiver also decrypts the message using Sk(r), which is only available to it. Because the receiver may need to send the transmitter's response, it would need the ID of the transmitter. The ID is sent the receiver within the encrypted message. Therefore, the nodes-in-the-middle will not be able to view the transmitter ID. The typical structure of the inter-network is presented in Figure 2.

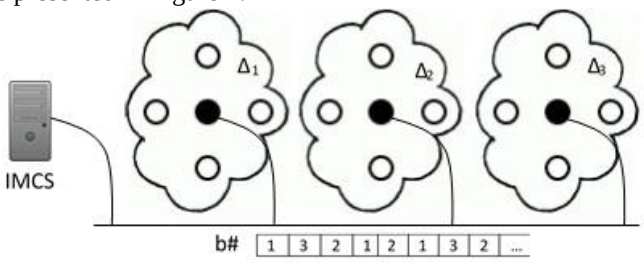

Figure 2. A sample of network which is constructed from 3 subnetworks. Each subnetwork has its own nodes. Transmitting between nodes of separate subnetworks is achieved by the routing layer. The routing is done anonymously

Using random IDs eliminates the transmitter's need to know the destination IP address. It also does not require the source IP address of the packets. But the source needs to obtain the destination ID. Managing and responding the ID requests is the responsibility of the structure described in Section 4. Using the structure, the source can obtain the ID of each node that it targets.

For anonymous routing to take place, routing information is set in the packet header in the form of homomorphically encrypted SBF, rather than origin and destination IDs. The elements on which SBF is constructed are the IDs of the nodes and subnets. A sample of the SBF construction is shown in Figure 1. Figure 1 uses the name b# for the SBF. After encrypting b# using the homomorphic encryption scheme which is described in Section 2.1, it is distributed between nodes. A b# can be changed due to the mobility property of the nodes. If a b# is changed, the new b# is constructed and, after encryption, redistributed between subnets. So each time a node has a packet to be sent, it can send the packet without delay. The private key for b# is only available on the routing layer. But all nodes have the public key associated with the b#, the encrypted filter EncPk(r)(b#) and the hash functions set.

Table 1 shows the information that each part of the network has.

Table 1. Information in access of each node

Ordinary Node i	**Encpk(r)(b#)** **Hash Functions** **Pki, Ski** **PkIMCS, IDIMCS**
Routing Layer	Sk for encrypting b#

Suppose that node s is going to send a packet to node r. To determine a comprehensive approach, we assume that these two nodes are located in two different subnets with the names Δi and Δj. To transfer a packet from s to r, the following steps will be happen. Figure 3 presents the algorithm schematically.

1. First, the node s look for the ID and the public key of r in its cache memory. If these values are not found, it would request them from DNS. The procedure for sending requests to DNS is similar to the procedure for sending requests to other nodes. When the ID of node r is specified, it uses the hash function set H to construct a b#, just from the ID of r. The number of ones in b# is stored in variable z. Then, according to the multiplicative properties of the cryptosystem, this filter is multiplied in EncPk(r)(b#). This result is named e#. The result is shuffled and placed inside the packet header with z. Also, the message which is going to be sent to r, plus the ID of s are encrypted by the public key of r and placed in the body part of the packet. The created packet will be sent to the gateway side of the transmitter. Figure 3 presents the step.

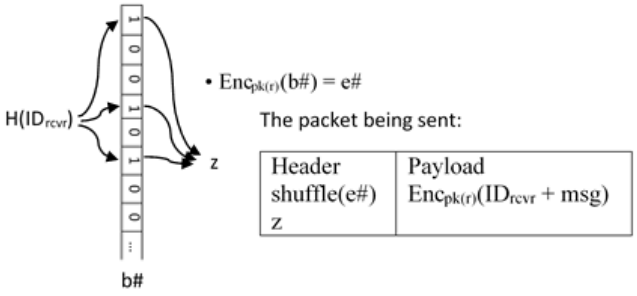

Figure 3. Node s is going to send a packet to node rcvr which belongs to other subnetwork. Node s have only the ID of the receiver. Node s creates a SBF related to ID of the receiver and count number of none zero cells of the SBF. It puts the SBF, multiplied by encrypted filter b#, and z values in the header of the packet and send it to the routing layer.

2. If the gateway does not participate in the routing layer, it sends the message without any change to the routing layer. It is necessary to point out that the transmitter node ID is encoded only in the body part of the packet, and thus the nodes-in-the-middle cannot view the ID by the way.
3. The network layer decrypts the e# contained in the packet header with its private key. The resulting value contains a number of zeros plus the number of j greater than zero. If the number of j is equal to z, then the destination node is in the network and is located in the subnet of j. If the non-zero values are variables, the smallest value would be considered as the subnet's identifier. This is explained in Section 2. Therefore, the routing layer delivers the body of the packet to the subnet gateway j. Note that only the encrypted message will be sent to the node r. Figure 4 presents the step.
4. The receiver subnet gateway broadcasts the received message in its subnet.
5. The node receives the message and can decrypt it using its private key.

The SBF property may cause a packet destined to the subnet Δj to be sent, unintentionally, to the subnet Δi. In other words, the destination routing node may be mistakenly identified within a subnet that is opposed to the main subnet. But, on the other hand, based on [26], the possibility of the

mistake can go down if some parameters such as the length of the filter and the number of hash functions is justified at the time of constructing the SBF.

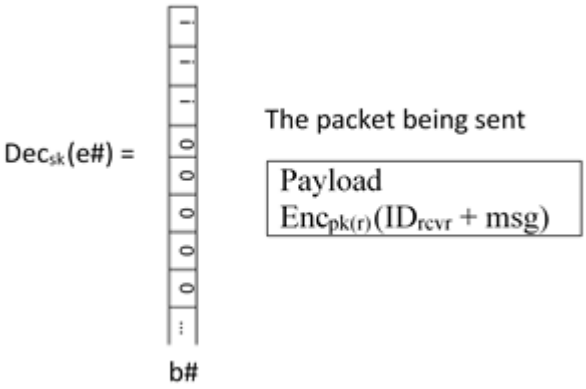

Figure 4. The routing layer recieves the packet, decrypts its header and sends the payload to subnetwork acheived by decryption

5. Reliable Mobility Management

Here, we presented an appropriate architecture for mobile management. In this architecture, the IoT Mobility Control Server (IMCS), (Figure 2), is responsible for maintaining the Mobility Control Table (MCT). Using this table, the server determines which node is going out of which subnet and what subnet it enters. This server also registers the new node that is being added to one of the subnets.

It is assumed that, like normal servers on the Internet, each node has a predefined domain name. It is also assumed that each node has the public key and ID of IMCS. Since the subnet gateways periodically publish their identifiers, the new node can also be informed it by entering a subnet. First, each node generates a public and private key. Then, it puts its public key, domain name, and the subnet ID in an ID request message and sends it to IMCS. After receiving the request, the IMCS adds a record in the MCT, generates and stores a random ID for that node and update the record with the subnet ID and node-specific domain name. Since a new ID has been added to the network, so b# must be updated and notified. IMCS is responsible for managing b#. Therefore, IMCS calculates the new b# filter based on existing IDs and the new ID and sends it to the routing layer. The routing layer also updates the filter on all subnets. After that, IMCS creates an answer packet containing the ID, hash functions, and the new b#, and encrypts it with the public key of the new node and sends it to the node. In this way, the new node is added to the subnet. Now assume that a node from its current subnet Δj is supposed to travel to the subnet with the identifier Δi. When a node travels from a subnet to another subnet, b# needs to be updated. The node examines the current Radio Signal Strength (RSS) for the Access Point to determine the time that the handover must be done. If this value is less than a threshold, the node sends the request for a new subnet to the IMCS. The IMCS also replaces the Δj identifier with the previous value in the MCT-related node record. Then calculates the new b# and sends it to the routing layer. As it is seen, no nodes is informed from the new position of the node.

6. Evaluation Results and Discussion

6.1 Performance Evaluation

In order to evaluate the proposed protocol, we have created five subnetworks with different number of nodes. Number of initial nodes in each of these subnetworks spans the range 15 to 30. Also there are different number of nodes sources and destinations which have been selected randomly. The speed of nodes changes from 0 to maximum of 10 meters/sec. We have run 20 simulations as the final result presented in all the figures of this section. Following we discuss these experimental results.

The proposed protocol allows nodes to move. The movement of nodes causes changes in the structure of subnets. Whenever a node transfers from a subnet to another subnet, it must notify the IMCS of its new location. Informing and updating the IMCS table, and then making the new b# and broadcasting it will cause delays in communicating and transferring information. Increasing the speed causes further changes in subnets, and as a result, increase in transmission delay. By increasing the number of nodes in subnetworks, the number of b# update operations increases, and therefore, the delay will increase. The transmission delay is shown in Figure 5. As shown in the figure, there is also a delay for the time that nodes are not moving (nodes have speed of 0). The delay is due to the distance between the source and destination nodes as well as the routing operations.

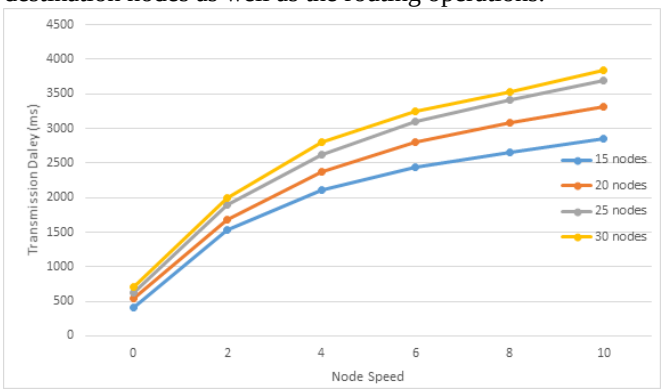

Figure 5. Average transmission delay vs. node speed. Initial number of nodes in each subnetworks varies between 15 to 30

In order to route packets correctly, b# should be created based on the current structure of subnets. Created b# should be encrypted and distributed between network nodes. If the speed of network nodes increases, the changes inside b# would also increase, which should be sent to network nodes. But since the number of subnets is constant, increasing the number of nodes in subnetworks does not greatly affect the increase in b# changes. Therefore, the overhead due to broadcasting the encrypted b# is affected more from the speed parameter than the total number of nodes in the network. Figure 6 shows the case.

Figure 7 shows the load of IMCS, based on the total number of nodes in the network and the average node speed. IMCS, as the first task, must act in the same way as DNS and respond to requests for IDs of destination nodes. In addition, the second task of the IMCS is to calculate b# based on changes in the location of nodes in subnets. Therefore, with

the increase in the speed of node movements, the number of network changes and after that, the amount of computing and workload of IMCS increases. As the figure shows, if the nodes are stationary, the IMCS workload will be subtle. This workload is due to the response to the first task IMCS should do. But with starting nodes' movements and changing the location of nodes from a subnet to another subnet, the workload of IMCS would increase.

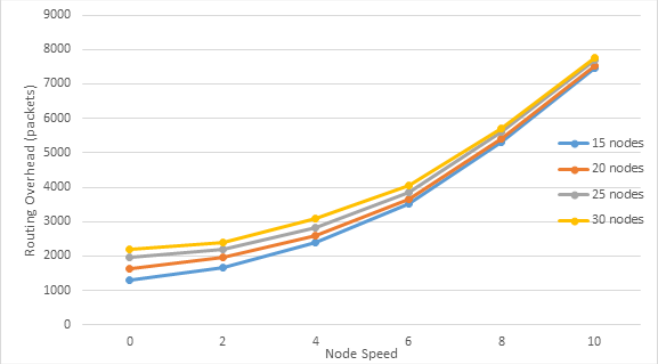

Figure 6. Routing overhead vs. average number of nodes in subnetworks

This workload increasing is due to the second task of IMCS which is added to its first task. This change in workload does not increase very fast by nodes' speedup until threshold 8m/s. After that, changing the node speed to more than 8m/s causes a relatively significant change in the workload of IMCS. The reason for the significant change is due to additive changes in b#.

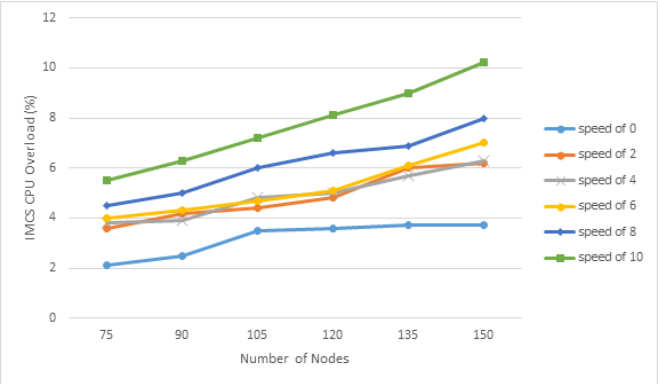

Figure 7. IMCS overload vs. number of nodes in all subnetworks

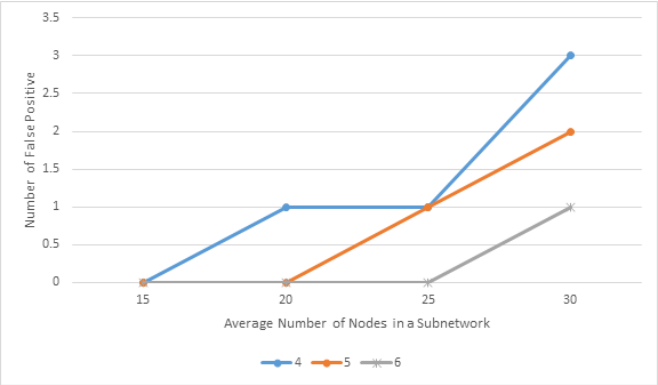

Figure 8. Number of false positive vs. average number of nodes in subnetworks

A set of hash functions H is used in constructing b#. These functions are applied to each node in a subnet and return a value. The value is used as the index of the cell of b# filter in which the network identifier of the node has to be set. The more number of hash functions, the more cells in the b# filter will represent the network identifier for a node. Therefore, by increasing the number of hash functions, the interference caused by the hash of all nodes in the network will have less effect on the loss of the correct location of a node in the network. This case is shown in Figure 8.

6.2 Security Evaluation

To check the security of the architecture, firstly, assume that the attacker can be as an instance node of a subnet, in the second case it can infiltrate one of the gateways and, in the third mode, it can view the information in the routing layer. Of course, assuming the attacker only seeks to collect information about the subnet structure and it acts in passive mode. In other words, we assume that the attacker has a semi-honest behavior. The scenario for each of these three modes will indicate that the attacker will not be able to obtain information from the network structure and thus the security is maintained.

In the first case, the attacker can see all information that the attacked node receives or sends. Therefore, the attacker will be able to see the ID of the nodes with which the attacked node is communicating. But an ID does not show any information from a node. On the other hand, the attacker will access encpk(r)(b#). But because it does not have the private key of the network layer, it cannot decrypt the filter. Also, since routing is performed as anonymous, the attacker cannot access the network structure and determines which node is in the subnet.

If the attacker can access a subnet gateway that is not working as a routing layer component, it will be able to view all packets that are exchanged by this gateway. But since the body information of these packets is encrypted and can only be decrypted by the private key of the destination node, the attacker will not be able to get notified of the content of the message. On the other hand, the information inside the packet also can only be accessed by the private key of the routing layer. And thus the attacker will not be able to view e#. Therefore, the attacker will not be able to obtain the network structure based on the information.

In the third case, that the attacker has accessed the routing layer, it is able to view packets that are transmitted between subnets. But the body of these packets are not understood by the routing layer, and thus the attacker cannot access the messages. On the other hand, even if it is assumed that the attacker can access e# by decrypting the header, it cannot identify the recipient node ID. Because the transmitter ID does not exist within the packet, the attacker cannot identify the transmitter. Therefore, the attacker will not be able to detect the network structure.

7. Conclusions

In this article, we proposed a protocol for private routing that can be used in the Internet of things. When network nodes are moving and traveling from one subnet to another, the network structure changes. It is possible to use the architecture when networks and subnets are changing.

The protocol uses homomorphic encryption, tunneling, and Spatial Bloom Filters to reach the privacy goal so that only the destination node will be able to determine which node has sent the message and what its content is. No node can understand the network structure or identify the subnet which a particular node is an element of. The routing layer is not able to determine which transmitter is sending the packet and the packet is destined for which receiver. Therefore, if the attacker is able to control one or more nodes, it cannot detect and control the network structure by obtaining network packets.

In addition, the proposed protocol is capable of working on a network structure that is constantly changing due to traveling nodes from subnets to subnets.

References

[1] L. Atzori, A. Iera and G. Morabito, "The internet of things: a survey," Computer Networks, vol. 54, no. 15, p. 2787–2805, 2010.

[2] D. Miorandi, S. Sicari, F. D. Pellegrini and I. Chlamtac, "Survey internet of things: vision, applications and research challenges," Ad Hoc Networks, vol. 10, no. 7, p. 1497–1516, 2012.

[3] M. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. Grieco, G. Boggia and M. Dohler, "Standardized protocol stack for the internet of (important) things," IEEE Communications Surveys & Tutorials, vol. 15, no. 3, p. 1389–1406, 2013.

[4] J. Kaur and K. Kaur, "Internet of Things: A Review on Technologies, Architecture, Challenges, Applications, Future Trends," International Journal of Computer Network and Information Security(IJCNIS), vol. 9, no. 4, pp. 57-70, 2017.

[5] D. Boswarthick, O. Elloumi and O. Hersent, M2M Communications: A Systems Approach, Wiley Publishing, 2012.

[6] B. Emmerson, "M2M: the internet of 50 billion devices," Huawei Win–Win Mag. J., no. 4, pp. 19-22, 2010.

[7] O. Hersent, D. Boswarthick and O. Elloumi, The Internet of Things: Key Applications and Protocols, Wiley Publishing, 2012.

[8] L. Grieco, M. Alaya, T. Monteil and K. Drira, "Architecting information centric ETSI-M2M systems," in IEEE PerCom, 2014.

[9] R. Weber, "Internet of things - new security and privacy challenges," Computer Law & Security Review, vol. 26, no. 1, pp. 23-30, 2010.

[10] H. Feng and W. Fu, "Study of recent development about privacy and security of the internet of things," in 2010 International Conference on Web Information Systems and Mining (WISM), 2010.

[11] R. Roman, J. Zhou and J. Lopez, "On the features and challenges of security and privacy in distributed internet of things," Computer Networks, vol. 57, no. 10, p. 2266–2279, 2013.

[12] J. Anderson and L. Rainie, "The Internet of Things will Thrive by 2025," 14 5 2014. [Online]. Available: http://www.pewinternet.org/2014/05/14/internet-of-things/. [Accessed 5 10 2017].

[13] M. Conti, J. Willemsen and B. Crispo, "Providing Source Location Privacy in Wireless Sensor Networks: A Survey," IEEE Communications Surveys and Tutorials, vol. 15, no. 3, p. 1238–1280, 2013.

[14] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar and G. Cao, "Towards event source unobservability with minimum network traffic in sensor networks," in The First ACM Conference on Wireless Network Security WISEC 2008, 2008.

[15] M. G. Solomon, V. S. Sunderam, L. Xiong and M. Li, "Enabling mutually private location proximity services in smart cities: A comparative assessment," in IEEE International Smart Cities Conference, ISC2 2016, Trento, Italy, 2016.

[16] P. Kamat, Y. Zhang, W. Trappe and C. Ozturk, "Enhancing Source-Location Privacy in Sensor Network Routing," in 25th International Conference on Distributed Computing Systems (ICDCS 2005), 2005.

[17] S. Geravand and M. Ahmadi, "Bloom Filter applications in network security: A state-of-the-art survey," Computer Networks, vol. 57, no. 18, p. 4047–4064, 2013.

[18] L. Zhang, "A self-adjusting directed random walk approach for enhancing source-location privacy in sensor network routing," in Proceedings of the International Conference on Wireless Communications and Mobile Computing, Vancouver, British Columbia, Canada, 2006.

[19] Y. Xi, L. Schwiebert and W. Shi, "Preserving source location privacy in monitoring-based wireless sensor networks," in 20th International Parallel and Distributed Processing Symposium (IPDPS 2006), 2006.

[20] P. Palmieri and J. A. Pouwelse, "Key management for onion routing in a true peer to peer setting," in 9th International Workshop on Security, IWSEC 2014, 2014.

[21] P. Palmieri, L. Calderoni and D. Maio, "Private inter-network routing for Wireless Sensor Networks and the Internet of Things," in CF'17 Proceedings of the Computing Frontiers Conference, Siena, Italy, 2017.

[22] J.-H. Lee, J.-M. Bonnin, I. You and T.-M. Chung, "Comparative handover performance analysis of IPv6 mobility management protocols," IEEE Transactions on Industrial Electronics, vol. 3, no. 60, 2013.

[23] K. Vasu, S. Mahapatra and C. S. Kumar, "MIPv6 protocols: A survey and comparative analysis," Computer Science & Information Technology (CS & IT), vol. 07, pp. 73-93, 2012.

[24] J. H. Sun, D. Howie and J. Sauvola, "Mobility management techniques for next generation wireless networks," in Proceedings of SPIE. Wireless and Mobile Communications, 2012.

[25] L. Calderoni, P. Palmieri and D. Maio, "Location privacy without mutual trust: the Spatial Bloom

Filter," Computer Communications, vol. 68, no. 1, pp. 4-16, 2015.

[26] L. C. a. D. M. Paolo Palmieri, "Spatial Bloom Filters: Enabling Privacy in Location-Aware Applications," in Information Security and Cryptology - 10th International Conference, Beijing, China, 2014.

[27] B. H. Bloom., "Space/Time Trade-offs in Hash Coding with Allowable Errors," Commun. ACM, vol. 13, no. 7, p. 422–426, 1970.

[28] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," in International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, 1999.

[29] A. Jegede, N. I. Udzir, A. Abdullah and R. Mahmod, "State of the Art in Biometric Key Binding and Key Generation Schemes," International Journal of Communication Networks and Information Security (IJCNIS), vol. 9, no. 3, pp. 333-344, 2017.

[30] R. Dingledine, N. Mathewson and P. F. Syverson., "Tor: the Second-Generation Onion Router," in the 13th USENIX Security Symposium, 2004.