SMGSAF: A Secure Multi-Geocasting based Routing Scheme for Opportunistic Networks

Jagdeep Singh · Sanjay K. Dhurandher · Raghav Kumar · Geetanshu Gupta

Received: date / Accepted: date

Abstract This paper proposes a novel Secure Multi-Geocasting Spray And Flood technique for opportunistic networks called SMGSAF, which uses secret sharing and disjoint path routing to secure the privacy of messages. In multi-geocasting, routing aims to successfully deliver a given geomessage to all the nodes, or to as many as possible, located inside defined geographic areas within a given time interval. It is desired that as long as a message is outside its destination casts it cannot be read by intermediate nodes. Encryption is a proven way to ensure this but classical encryption techniques are not well suited for the opportunistic paradigm. Key distribution and scarcity of resources are the major challenges in this regard. Therefore we have used secret sharing and disjoint path routing to protect the privacy of messages. Simulation results show that the proposed SMGSAF protocol provides the intended security but at the expense of performance, that is within acceptable limits. Notably, the SMGSAF protocol outperforms unitary geocasting in terms of delivery probability. The proposed protocol is evaluated in terms of delivery probability, network overhead, and latency. Its performance has been compared to MGSAF and GSAF.

Keywords · Opportunistic Networks · Multi-Geocasting · Disjoint Path · Security · ONE Simulator

1 Introduction

Opportunistic network [1] is a class of networks that is composed of mobile nodes that communicate with each other intermittently. Messages are carried by the nodes using a hop-carry-forward routing mechanism in which nodes store messages in their buffers and carry them around in the network before forwarding them to the next node which may be the destination of the message or which may be better suited to deliver the message than itself. Nodes carry messages around the network exchanging them among one another and nodes continually drop copies of old messages to make room for new ones. Thus, multiple copies of the same message exist in the network at any time. This redundancy ensures that a message remains in circulation long enough so that it can be received by its destination node. Routing [2] in OppNets is a probabilistic process, and every encounter between nodes must be used effectively to ensure maximum chances of delivery as well as minimal redundancy in the network.

To make sure of this at every communication opportunity a node evaluates if the encountered node is suitable to forward a message or not. Mostly such encounters are scarce and nodes cannot be very conservative in making the decisions. Also, nodes have very little information regarding other nodes due to unstable network topology, absence of infrastructure, and privacy concerns of other nodes. This provides an opportunity for malicious parties to pose as suitable carriers of the message and read messages in transit. One way to preserve the confidentiality of messages is to encrypt them so that they can only be read by the intended parties. Encryption [3] in OppNets is a challenging oper-
ation. Conventional encryption schemes [4] both symmetric and asymmetric techniques are resource-intensive and nodes in OppNet are usually resource-constrained. Being mobile they have limited energy and memory. Key distribution is also a major challenge in opportunistic networks due to the absence of infrastructure and stable network topology. To overcome these challenges a key-less mechanism is required which is based on simple operations that can be carried out using fewer resources. Preserving confidentiality is one of the key concerns in the transmission of information. To make sure that only the intended parties can see the information contained in a message, mechanisms are required to encode the information in a way that renders it unintelligible to all the parties except the intended recipient. This procedure is called encryption of message and there are numerous algorithms in practice to carry it out. These algorithms make use of shared knowledge among the sender and the receiver, usually in the form of keys. AES, DES, RSA are some examples [5]. The problem is that most of these encryption techniques have intensive processing requirement and they also require a key distribution mechanism.

However, nodes in OppNets [6] are mobile devices that are usually handheld, wearable, or mounted on a carrier. Smartphones and military communication equipment are some examples. Such devices rely on a battery to power them and there are limitations to the size and weight of the battery to ensure their ease of use. Consequently, the capacity of nodes in terms of power backup as well as information processing is limited. Also, there is no infrastructure present to perform key distribution. Therefore the usual encryption algorithms [7] cannot be used for OppNets. Another way of ensuring confidentiality is secret sharing i.e. to break the information into parts and encode them such that unless and until a party has all the encoded parts, it cannot recover the whole or even part of the original information. In this work, we propose such a mechanism and have shown how it performs in terms of delivering messages. We have implemented it over a geo-casting addressing scheme, to show how it protects the confidentiality of information outside its destination area. The geographic routing protocol extended is GSAF [8] and we use it to route messages to multiple geo-casts instead of one as it originally did.

In this paper, we propose a secret sharing scheme that is used with disjoint path routing to securely route messages in a multi-geocasting addressing scheme. Geocasting [9] is a special case of anycasting in which messages are addressed to all the nodes present in a geographic location. To geo-cast messages, we extend the GSAF protocol such that messages can be addressed to multiple geographic casts and as long as they remain outside the destination cast, they are indecipherable.

1.1 Contributions

The major contributions of this work are-

1. We introduce a multi-path routing mechanism for OppNets that secure messages using secret sharing.
2. Extend the Geocasting Spray And Flood (GSAF) protocol to support multiple geocasting.
3. Propose a novel Secure Multi-Geocasting Spray And Flood (SMGSAF) protocol to securely route messages to multiple destination geo-casts.

The rest of the paper is organized as follows. In Section II, some related work is presented. In Section III, the proposed protocol is described. In Section IV, simulation results and justifications are presented. Finally, Section V concludes the paper.

2 Related Work

To the best of our knowledge, there is no work in the literature on secure multi geocasting in OppNets [10–14]. Therefore, the work described in this paper is based on unitary geocasting and multi-geocasting in oppnets and other network environments.

In [15], the authors have proposed the GEOOPP geocasting protocol for OppNets. The routing procedure has been broken into two separate phases:

1. Forwarding messages to the destination cast,
2. Flooding it inside the region using Epidemic [18] routing.

Their major focus has been on the first phase which is concerned with delivering messages to a node inside the addressed geographic cast, and the second phase has been ignored in their evaluation. Authors have tried measuring the probability of geographic progress made by a node to carry a message to its destination (using Chebyshev’s inequality) and it has taken the probability into account whenever any message has to be forwarded to its next hop. In GEOOPP the network is divided into small non-overlapping cells, it calculates the probability of reaching the destination cell via every other cell and forwards the message to a neighboring node if the future destination of the neighboring node has a higher chance of delivering the message (i.e. higher calculated rate) as compared to the present location.

In the work, presented by Y. Ma and A. Jamalipour [16], similar to GEOOPP [15], the authors have only focused on routing messages to a node located inside the location followed by a flooding based approach. The destination cast is defined as a circle based on the parameters and the coordinates of the center, and the radius of the circle respectively. The definition of casts as circles suffers from the limitations
discussed before. Routing decisions are based upon the Expected Visiting Rate (EVR), which is the probability of each user visiting the destination cast to deliver messages to their destinations. EVR is calculated based upon the history of visited regions in the network. The exchange of location history raises a lot of privacy concerns as discussed before.

In [17], the author has proposed a geocast routing protocol for mobile partitioned networks. It has been assumed that nodes in the partitioned subregions are constantly connected, and each node belonging to a said subregion will remain inside that particular subregion; i.e. nodes cannot move between subregions. This assumption is at odds with the conventional scheme of geocasting in OppNets as conventionally nodes can move freely on their own will. In this approach, nodes start exchanging hello messages, until they get a complete picture of the network and have inferred the mobility pattern of other users. This causes higher energy consumption and high network overhead. Also, the mobility pattern of nodes may change even before other nodes are informed about the previous pattern of movement because of the complexity of OppNets. Following that, when the mobility pattern has been extracted, messages are transmitted towards the regions which have more stable mobility patterns to ensure higher chances of delivery to their destination.

In [8], the author has proposed a geocasting scheme that is efficient and effective. The work has also laid down a cast definition scheme using which an arbitrary n-polygon cast can be defined as the destination of messages. Messages are transmitted to a node located inside the geographic cast by using the spray and wait protocol followed by intelligent flooding once the message has reached inside the geographic cast. This report draws inspiration from the GSAF protocol and builds upon the proposed scheme to describe a multi-geocasting protocol. In GSAF messages can be destined for single geographic casts only. In [8], the author has proposed another protocol called DAGSAF or Direction Aware GSAF.

In [19], the authors have proposed two multi-geocast protocols for ad-hoc wireless sensor networks. These protocols guarantee delivery of packets and low transmission cost irrespective of the density of the network. Both the protocols are based on path discovery using a short initial message and differ in terms of the nodes that they look for. In the first protocol shortest paths from the source node to the nodes just outside of the destination casts are discovered, whereas in the latter protocol paths to any one node inside each cast are discovered.

In [20], the authors present Recursive Multiregion Geocasting (RMG), a multi-region geocast routing protocol that addresses the problem of delivering data from a source to multiple remote geocast regions in large-scale wireless sensor networks. The key idea behind this protocol is to treat a group of geocast regions as a point destination and forward data packets along a straight line towards the group, until a division point at which the group is divided, and the packet is forwarded towards the sub-groups in the same fashion. Simulations have shown that it incurred very low computation overhead.

In [21], the authors have proposed a multi-geocasting scheme called Smart Geocast for VANETs. The scheme is composed of two procedures viz. geocasting initialization and geocasting maintenance. It aims at disseminating information about abnormal traffic conditions, along routes that will be affected due to them. In the first procedure, the information about traffic is delivered to multiple regions quickly and efficiently by using path sharing and path splitting schemes. New vehicles may continuously drive into the target regions, thus the traffic information has to be kept floating or visible to them until the effect of the event disappears. In the second procedure, a small area in each target region is divided to disseminate information to recently arrived vehicles continuously, thus further reducing message redundancy and maintenance costs.

Multi-geocasting is also being studied as a use-case in mainstream networks such as 5G. This proves the growing imminence of multi-geocasting as a major routing scheme. Technologies such as Non-Orthogonal Multiple Access (NOMA) for 5G systems are being leveraged to define schemes for realizing the simultaneous delivery of different messages to different user groups characterized by different geographical locations [22].

The aforementioned protocols either have their limitations and shortcomings are simply not aligned with the Opp-Net framework to implement multi-geocasting in opportunistic scenarios. As discussed in the above research, related studies have ignored the fact that the definition of geocast membership is different from the usual unicasting and multicasting approaches in OppNets.

2.1 Motivation

This work proposes a mechanism to define a secure routing protocol using which messages can be securely routed to multiple geographic casts at the same time. It should be noted that in geocasting, a message that is destined for a geocast, is disseminated minimally as long as it is outside the cast. Till the time it reaches the destination cast(s), the replication of the message is highly controlled to keep network overhead down. The contribution of this work is the mechanism to securely route a message from source to multiple destination geographic casts.

3 System Model

Geographic Spray and Flood (GSAF) [8] is a simple yet efficient and flexible geo-casting protocol for oppnets. It fol-
follows a simple approach in which messages take random walks towards destination cast. Destination cast is defined by specifying the coordinates of the end points of the boundaries. Multi GSAF is a simple extension of GSAF in which a message can contain descriptions of multiple geographic casts as destination instead of one as was the case in GSAF [8]. Then while delivering a message a node checks if it is present in any one of the message’s destination casts. The process of routing a message is divided into two phases:

1. The message is being transmitted outside the geographic cast
2. The message is being transmitted inside one of its destination casts

In this work we, propose a secure routing scheme SMGSASF which ensures privacy of message as long as it is in phase 1 of transmission i.e. outside the destination cast. SMGSASF makes use of secret sharing and disjoint path routing to achieve this goal, which are described in the following subsections. Table 1 on page 4 refers to the the notations and their descriptions.

3.1 Secret Sharing

Secret sharing is a process in which a piece of information is split into ‘n’ parts and unless a party holds at least ‘k’ of those parts it cannot construct either any part or whole of the original information. Although secret sharing guarantees that at least k parts are required to decrypt the entire message, it cannot solely solve the challenges we face. Many secret sharing algorithms such as shamirs secret sharing scheme [23] and others are based on modular arithmetic, which is resource intensive and hence not well suited for OppNet paradigm.

We therefore define a secret sharing scheme that makes use of the xor (⊕) [24] operation, which is very convenient to carry out in hardware as well as software. Though there is a trade-off that while using this scheme if a malicious node comes in possession of less than ‘k’ parts, it may figure out a part of the message. Hence to ensure complete privacy we need to pair up this scheme with disjoint path routing that is explained in the next subsection. For our purpose we selected n = k = 4 i.e. the message is broken into 4 parts and unless a node is in possession of all the 4 parts it cannot construct the whole message. Let ‘P’ be the original information, then it is broken into 4 equal parts. If the length of the information is not a multiple of 4 then it can be padded to make its length a multiple of 4. Let the parts be P1, P2, P3 and P4. Then to make them unreadable we encode them as follows –

\[ C_1 = P_1 \oplus P_2 \]

\[ C_i = P_i \oplus P_{i-1} \text{ for } i = 2, 3, 4 \]

Now instead of transmitting Pi, C_i is transmitted thus hiding the original information. The original information can be decoded as follows –

\[ P_i = C_i \oplus C_{i-1} \text{ for } i = 2, 3, 4 \]

\[ P_1 = C_1 \oplus P_4 \]

The decoding can be verified as by using equation (2) –

\[ C_i \oplus C_{i-1} = P_i \oplus C_{i-1} \oplus C_{i-2} = P_i \]

and, by using equation (1), the decoding can be verified as -

\[ C_1 \oplus P_2 = P_1 \oplus P_4 \oplus P_4 = P_1 \]

The original information m can then be constructed by concatenating P1, P2, P3, P4 thus obtained.

3.2 Multi-path routing

From Equation 3, we can infer that if a node is in possession of parts C_i and C_{i-1} it can decode P_i, therefore it is necessary that no node except the ones in the destination cast ever have consecutive parts. However, it can be verified that even if a node has alternate parts i.e. C_1 and C_3 or C_2 and C_4, it cannot infer any part of the original information.

Therefore, to transmit data P, it is broken into 4 parts and encoded as described in the last section. Then the four encoded parts are set as the payload of geomessages gm1, gm2, gm3 and gm4. The geomessage has an attribute partag which can take values from 0 to 4. If it is set to zero it implies that its payload is a complete message otherwise it indicates the part number of the encoded message it is carrying. Then the geomessages are transmitted along disjoint paths i.e. no two

| Notation | Description |
|----------|-------------|
| m, m_i | Message |
| m | Encoded message |
| H_s | Source node |
| R | Receiver node |
| g_i | Geographic cast |
| G | Set of geographic casts |
| SV | Summary Vector |
| gm, gm_i | Geomessage |
| gm | Encoded Geomessage |
| H_i, where i = 2, 3, ....7 | Intermediate node |
| H_i, where j = a, b, ....d | Geocast region nodes |
geomessages carrying consecutive parts of the same data are present on the same node unless that node is present inside the destination cast. It must also be noted that to keep the scheme simple for implementation, geomessages \( gm_1 \) and \( gm_2 \) are also routed along disjoint path.

Thus, geomessages \( gm_1 \) and \( gm_2 \) can be transmitted together to a nearby host or geomessages \( gm_3 \) and \( gm_4 \) can be transmitted together, but no other combination is allowed. Thus by combining secret sharing and disjoint path routing, we were able to preserve the privacy of the original information being carried by geomessages.

Let’s assume that the data to be transmitted is ‘P’ (‘P’ for plaintext) which can be of arbitrary size. To send the data securely it is broken into 4 blocks of equal size, viz. \( P_1 \), \( P_2 \), \( P_3 \) and \( P_4 \), which are then encoded, using the formulas 1 and 2, into data blocks \( C_1 \), \( C_2 \), \( C_3 \) and \( C_4 \) (‘C’ for ciphertext). If the size of data ‘P’ is not divisible by 4, dummy data can be appended to it.

The data blocks \( C_1 \) to \( C_4 \) are carried around by geomessages \( gm_{i} \), \( i = 1 \) to 4. Along with the data \( C_i \), \( gm_i \) also contains metadata which was described in the last subsection.

A geomessage can be visualized as a packet that contains the data as its payload and metadata as its header. Nodes in the onpet route geomessages in the network as packets are routed in packet switched network while enforcing additional rules that ensure that the data inside the geomessages cannot be read by any node other than the destination node.

Eg. – Node \( S \) has to send data \( P \) to node \( D \). \( S \) appends dummy data to \( P \) so that it can be broken into 4 equal parts, if necessary. \( P \) generates a unique id for the data \( P \), lets call it ‘ID’, then \( P \) is broken into \( P_1 \), \( P_2 \), \( P_3 \) and \( P_4 \) i.e \( P = P_1 * P_2 * P_3 * P_4 \), where ‘*’ is the concatenation operation. Then \( S \) encodes \( P_1 \) into \( C_1 \). The encoded blocks \( C_i \) are set as the payload of geomessages \( gm_{id} \), \( i = 1 \) to 4, i.e –

\[
\begin{align*}
\text{Set } gm_{id}.id & = id, \\
\text{Set } gm_{id}.from & = S, \\
\text{Set } gm_{id}.to & = D, \\
\text{Set } gm_{id}.payload & = C_i, \\
\text{Set } gm_{id}.partid & = i, \\
\text{Set } gm_{id}.ttl & = m.ttl, \\
\text{Set } gm_{id}.ktk & = 3
\end{align*}
\]

Then \( gm_{id} \), \( i = 1 \) to 4 are routed in accordance with algorithm 2. When all these geomessages are received by node \( D \), then it extracts the payload of the geomessages viz. \( C_1 \), \( C_2 \), \( C_3 \) and \( C_4 \) and decodes using Equations 3 and 4.

### 3.3 Proposed Secure Multi-Geocasting based Routing Scheme for Opportunistic Networks

In this protocol, the term ‘geomessage’ is used to denote messages that are addressed to a geographic cast whereas the term ‘message’ refers to the information that a geomessage is carrying as its payload. A geomessage has following attributes:

1. \( m_{id} \) : a message id, which is same for all geomessages carrying parts or whole of the same message \( m \)
2. from : id of the host that created it
3. to : a list of description of destination casts
4. payload : the message it is carrying
5. \( ttl \) : timestamp till which it is valid
6. partid : the index of the part of encoded message e.g. geomessage \( gm_1 \) contains the first encoded part of message \( m \) therefore \( gm_1.partid = 1 \). If a geomessage \( gm \) contains a complete message \( m \), then \( gm.partid = 0 \).

7. tkt (read ticket) : number of copies of this geomessage that can be transmitted to intermediate nodes. Eg. if node \( H \) contains geomessage \( gm \) such that \( gm.tkt = 2 \) and \( H \) is not inside any cast described in \( gm.to \), then \( H \) can forward copies of \( gm \) to only two intermediate nodes and \( gm.tkt \) will be decremented by 1 before each forward. If \( gm.tkt = 0 \) then \( H \) can forward \( gm \) only if it is inside one of the casts defined in \( gm.to \).

A node \( H \) has to transmit a message \( m \) to a set of geographic casts \( G = ( g_1, g_2, g_3, ... \) where \( g_i \) defines the \( i^n \) destination cast). It divides \( m \) into 4 parts and encodes them as described by eqn. 1 and 2. The four encoded parts are then set as the payload of four geomessages with partid corresponding to the index of the encoded part set as its payload. All the other attributes of the four geomessages are set the same. The \( m_{id} \) of the geomessages is set to MID, from is set to \( H \), to is set to \( G \), tkt is defined as per the requirement and tkt is set to 3. These geomessages are then routed in the network while following a set of rules defined in the following paragraphs. The process of creating a message is shown in Algorithm 1.

It is assumed that hosts automatically drop geomessages whose \( ttl \) has expired. Let \( H_i \) and \( H_j \) be two nodes in the network. When these nodes encounter each other they exchange a summary vector (SV) that is a set of tuples \((m_{id}, partid)\) of geomessages it is carrying. \( H_i \) sends \( SV_i \) to \( H_j \) and \( H_j \) sends \( SV_j \) to \( H_i \). Then \( H_i \) takes the role of the sender and \( H_j \) takes the role of receiver. After \( H_i \) is done sending messages to \( H_j \), the roles are reversed. The sender forwards the geomessages in its buffer according to the rules described below.

The sender \( S \) repeats the process for every geomessage \( gm \) it is carrying. It’s summary map is denoted by \( SV_k \) and the summary map of receiver \( R \) is denoted by \( SV_R \). For every geomessage \( gm \) in its buffer it checks if \((gm.m_{id}, gm.partid)\) is not present in \( SV_k \), if it is then \( S \) moves on to the next geomessage in buffer. If the present geomessage is not found in \( SV_k \), then \( S \) checks \( gm.tkt \). If \( gm.tkt \) is greater than zero it means that \( S \) is allowed to forward copies of \( gm \) outside the destination cast. If \( gm.tkt \) is zero, then \( S \) can forward \( gm \) if
and only if it is present inside one of the geographic casts defined in \(gm.to\). It should be mentioned here that when a geomessage is received by a host present inside its destination cast its \(kt\) is set to 0, which we will see in the following paragraphs. If \(gm.to\) \(>0\) and \(S\) is not inside any casts in \(gm.to\) it refers to the summary vector \(SV_R\) to check if \((gm.m_id, \text{gm.part}_id + 1)\) exists in \(SV_R\). In this exchange the receiver \(R\) keeps track of geomessages that it receives from \(S\). For every geomessage \(gm\) it receives it performs the following checks. If \(R\) is present inside \(gm.to\) then it sets \(gm.to\) to 0. Then it checks \(gm.part_id\). If \(gm.part_id\) is 0 it implies that \(gm\) contains a complete message \(m\) as its payload and \(R\) deletes all the geomessages it has with \(m_id\) equal to \(gm.m_id\) and \(part_id\) not equal to 0 and puts \(gm\) in its buffer. Thus \(R\) only stores a copy of the complete message \(m\) and not the parts of the message. If on the other hand \(gm.part_id\) is not zero then \(R\) checks if other geomessages with \(m_id\) equal to \(gm.m_id\) are present in \(R\)’s buffer or not. If 3 other geomessages with mid equal to \(gm.m_id\) and \(part_id\) not equal to \(gm.part_id\) are present in the buffer of \(R\), then it constructs the original message \(m\) by combining the payloads of the four geomessages according to equations 3 and 4. It then creates a new geomessage \(gm\) with \(gm.payload\) set as \(m\), \(gm.part_id\) set as 0 and all the other attributes set same as \(gm\), and puts it in its buffer. All the geomessages in \(R\)’s buffer corresponding to \((gm.m_id, i)\) are deleted from the buffer and received geomessage \(gm\) is also cleared. Thus \(R\) ends up storing only the geomessage with complete message \(m\) as its payload. If on the other hand \(R\) is not present inside \(gm.to\), it just puts \(gm\) in its buffer, making space if necessary by dropping existing geomessages from buffer, and wait for the next geomessage. The receiver also updates local copy of its summary vector to include the geomessage that it has put in its buffer.

3.4 Illustrative example

Let’s take an example to illustrate the SMGSAF protocol. In Figure 1, an example network has been shown. Only one destination geographic cast has been shown to keep the depiction simple. But this same procedure can be replicated for multiple geographic casts as we will explain ahead.

For a geomessage its path is the set of nodes that have carried the geomessage in their buffers. A node \(H_i\) (shown in green) has a message \(m\) identified by \(MID\) to deliver to geographic cast \(g\) that is bounded by the green lines. It creates four geomessages \(gm_1, gm_2, gm_3\) and \(gm_4\) containing the encoded parts of message \(m\) according to Algorithm 1. Let’s call the geomessages \(gm_1\) and \(gm_3\) odd geomessages and \(gm_2\) and \(gm_4\) even geomessages. Both the odd geomessages can be routed over the same path because even if a node contains \(gm_1\) and \(gm_3\), it cannot obtain the message \(m\) or any part of it. Looking at step 5 in Algorithm 1., it can

---

**Algorithm 1: Proposed SMGSAF Protocol**

1. Node \(H_i\) take the role of sender \(S\) and \(H_j\) takes the role of receiver \(R\).
2. Once \(S\) has looped over all the geomessages in its buffer, the algorithm is repeated only once more with \(H_j\) as sender \(S\) and \(H_i\) as receiver \(R\).
3. \(R\) transmits its summary vector \(SV_R\) to \(S\).
4. if the node is sender \(S\) then
5. for all geomessage \(gm\) in the buffer do
6. if \((gm.m_id, gm.part_id) \in SV_R\) then
7. skip to next geomessage in buffer and go to step 5
8. end if
9. if \(gm.to > 0\) then
10. if \((gm.m_id, gm.part_id + 1)\%4\) and \((gm.m_id, gm.part_id + 3)\%4\) not in \(SV_R\) then
11. Forward \(gm\) to \(R\)
12. if \(gm.part_id = 0\) then
13. Set \(gm.m_id = gm.m_id\), \(gm.part_id = 0\)
14. Forward \(gm\) to \(R\)
15. end if
16. else if \(gm.to = 0\) then
17. if \(S\) is present in any cast in \(gm.to\) then
18. Forward \(gm\) to \(R\)
19. Update \(SV_R\)
20. else
21. skip to next message in buffer and goto step 5
22. end if
23. end if
24. end for
25. else if the node is receiver \(R\) then
26. for every geomessage \(gm\) received from \(S\) do
27. if \(R\) is present in any cast in \(gm.to\) then
28. Set \(gm.to = 0\)
29. if \(gm.part_id = 0\) then
30. delete all messages from buffer with \(m_id = gm.m_id\)
31. put \(gm\) in buffer
32. else
33. if all message with \(m_id = gm.m_id\) and \(part_id\) belonging to \{1,3\} - \(gm.part_id\) are present in buffer then
34. Create geomessage \(gm\)
35. Set \(gm.m_id = gm.m_id\)
36. Set \(gm.payload\) from \(gm\), from
37. Set \(gm.to = gm.to\),
38. Set \(gm.ttl = gm.ttl\),
39. Set \(gm.part_id = 0\)
40. for \(i = 1,2,3,4\) do
41. if \(i = 1,2,3,4\) do
42. set \(m[i] = gm.m_id\), \(i\), \(m[i]\) payload where \((gm.m_id, i)\) is the geomessage in buffer with \(mid\) = \(gm.m_id\) and \(part_id = i\)
43. end for
44. if \(i = 4,3,2\) do
45. Set \(m[i] = m[i] + m[i-1]\),
46. end for
47. Set \(m[1] = m[1] \oplus m[4]\),
48. Set \(gm.payload = m[1] + m[2] + m[3] + m[4]\), where \(\oplus\) is concatenation
49. for \(i = 1,2,3,4\) do
50. Remove \((gm.m_id, i)\) from buffer where \((gm.m_id, i)\) is the geomessage in buffer with \(mid\) = \(gm.m_id\) and \(part_id = i\)
51. end for
52. put \(gm\) in buffer
53. else
54. put \(gm\) in buffer
55. end if
56. end if
57. else if \(R\) is not present in any cast in \(gm.to\) then
58. put \(gm\) in buffer
59. end if
60. end for
61. end if
be verified that the protocol does not prevent \((\text{MID}, 1)\) and \((\text{MID}, 3)\) from being present on the same node. These arguments also hold for the even geомessages. Both the odd (or even) geомessages may be routed through different paths, not necessarily disjoint, but an odd geомessage and an even geомessage are always routed over disjoint paths.

To keep the example simple, we have assumed that the odd geомessages both are forwarded along the same path and both the even geомessages are routed along the same path. Nodes shown in blue \((H_2, H_3, H_6\text{ and }H_7)\) form the path taken by odd geомessages and nodes shown in black \((H_3, H_5, H_8)\) form the path taken by the even geомessages. The orange nodes are those nodes that either created the geомessage \(g_m\) containing the original message \(m\) (node \(H_0\)) or that received the geомessage \(g_m(H_0, H_1, H_2)\). At the end of the routing process nodes \(H_7\) and \(H_8\) will also have received \(g_m\). The gray node \(H_9\) is one that has not received any geомessage out of \(g_{m1}, g_{m2}, g_{m3}\) and \(g_{m4}\). The solid arrows define the direction of geомessage transfer and are labeled with the payload of the transferred geомessages, the dashed arrows denote a future transmission of geомessages, carrying complete message as payload, along with their directions and the dashed line shows the connections that were present but over which no geомessage was transmitted.

At \(H_1\) all the geомessages have a \(tkt = 3\), thus \(H_1\) can forward at most 3 copies of each geомessage. It forwards \(g_{m1}\) and \(g_{m3}\), with \(tkt = 2(=3-1)\), to \(H_2\) and \(g_{m2}\) and \(g_{m4}\), with \(tkt 2(=3-1)\), to \(H_3\). Afterwards, \(H_2\) and \(H_3\) come within each other’s range, as shown by the dotted line between them, but they cannot exchange geомessages as the condition on line 5 in algorithm 2 prevents it. This condition ensures that the even and odd geомessages take disjoint paths. \(H_2\) then forwards \(g_{m1}\) and \(g_{m1}\), with \(tkt = 1(=2-1)\), to \(H_4\) and \(H_5\) forwards \(g_{m2}\) an \(g_{m4}\), with \(tkt = 1(=2-1)\), to \(H_8\). A connection is eventually established between \(H_2\) and \(H_3\) but they do not exchange any geомessages to ensure disjoint path routing. Afterwards \(H_2\) forwards \(g_{m1}\) and \(g_{m3}\), with \(tkt = 0(=1-1)\) to \(H_6\). It must be noted here that since the \(tkt\) value of the copies of geомessages carried by \(H_6\) is zero and \(H_0\) is not present inside the destination cast \(g\), it cannot forward the geомessages to node \(H_0\).

Nodes \(H_3\) and \(H_5\) then forward the copies of geомessages that they are carrying, with \(tkt\) set as \(0(=1-1)\), to nodes \(H_7\) and \(H_8\) respectively. Both these nodes are present inside the destination cast \(g\). \(H_7\) forwards \(g_{m1}\) and \(g_{m3}\) to \(H_a\). \(H_8\) forwards \(g_{m2}\) and \(g_{m4}\) to \(H_a\), irrespective of the fact that \(H_a\) already contains \(g_{m1}\) and \(g_{m3}\) because \(H_8\) is present inside the destination cast \(g\). \(H_a\) on receiving all the encoded parts of the message \(m\) constructs it following the step on line yyy in Algorithm 2. The original message is placed in an new geомessage \(g_m\) with id \((\text{MID}, 0)\) and \(tkt\) set to 0. \(H_a\) then forwards \(g_m\) to \(H_6\) and \(H_7\), even though \(g_m.tkt\) is zero because \(H_a\) is present inside \(g\). \(H_6\) and \(H_7\), being present inside

![Fig. 1: Proposed SMGSAF Protocol Example with Single Geocast Region](image1)

![Fig. 2: Proposed SMGSAF Protocol Example with Multigeocast Region](image2)
4 Evaluation

In this section, the performance of the proposed SMGSAF is simulated and compared against that of two benchmark routing protocols: multi-GSAF (called MGSAF) and GSAF [8], using the Opportunistic Network Environment (ONE) simulator [25]. ONE is one of the efficient and best simulation tool opportunistic networks that implement several mainstream routing protocols and provide the functionality of generating the mobility traces with the help of different mobility movement models.

Routing schemes in OppNets are evaluated in terms of the number of messages that were correctly delivered to their destination and the amount of redundancy that had to be introduced, which is measured in terms of the overhead ratio which we will be defined shortly. What has to be noted here that since SMGSAF splits a message into multiple parts and parts are routed along disjoint paths we need to redefine the notion of the number of deliveries and redundancy.

To keep the description clear we use the term message to denote the original information that was intended to be transmitted and the term geomessage is used to denote the unit of information that is exchanged between nodes. For GSAF and multi-GSAF message and geomessage correspond to the same information i.e. every geomessage corresponds to a message, but this is not the case for SMGSAF. In SMGSAF a message is broken into parts, that are encoded, and are then carried by geomessages. A geomessage may as well be carrying a complete message if its carrier node had received all the geomessages that were carrying the different parts of the message.

Therefore the question is whether to define the number of deliveries and the number of copies created in terms of messages or geomessages. To test the efficacy of the system it is important to ensure that complete messages reach their destinations and hence it becomes important to look at the fraction of nodes that received the message out of all the nodes that were present inside the destination casts of the message during its lifetime. Therefore for every message, we found the number of nodes that received a message and divided it by the number of nodes that were present in any one of its destination geographic casts during its lifetime. The delivery probability of the network is given by averaging the pmdr of all the messages created in the network.

\[
delivery probability = \frac{\sum i=1^n pmdr_i}{\text{number of messages created}}
\]  

It should be noted that here we look at the number of messages created and not geomessages.

4.1.2 Overhead Ratio

the ratio of the total number of messages relayed over the total number of unique messages delivered. It can be interpreted as the number of geomessages that had to be relayed for each relay that resulted in delivery.

4.1.3 Average Latency

The average time that is used to deliver messages to corresponding destination. We consider the delay time of undelivered messages as the time that they have been staying in the network by the time our simulation stops.
4.1.4 Number of Packets Dropped

In attacks based environment, the malicious nodes drop packets instead of relaying them.

4.2 Simulation Scenario

SMGSAF was simulated in Helsinki city center (an area of 4500 m x 3400 m). We have experimented with 6 different node densities in our simulations (130, 195, 260, 325, 390, and 455 nodes in the given area). The nodes are of the three types: pedestrians, cars, and buses. Table 3 shows the number of nodes for each model. Table 4 and Table 2 show the usability models of all three simulated forms of users and default simulation settings respectively. The default total number of nodes in our simulations, unless otherwise specified, is 195. Also, we have experimented with different sizes of system buffers (5, 10, 15, 20, 25 and 30 MB; the default being 10 MB) and various lifespans of messages (15, 30, 45, 60, 75, 90, 105, 120 minutes; the default being 120 minutes). All simulations were running for 8 hours; for each simulation, the warm-up and cool-down periods were 1 hour each. Messages are scheduled as follows: a sender and multiple destination casts are selected at random from the set of devices in the networks and the set of predefined casts, respectively; message weight is fixed (500 KB and thus the size of each element is 125 KB) and a new message is scheduled every 25 to 35 seconds. The buffer scheduling policy is random, which means that messages are picked randomly from the system buffer when a computer meets another system in the network. The attack model implemented for these simulations is the message fabrication attack, where some of the nodes in each simulation behave maliciously. Finally, our secure multi-geocasting protocol (SMGSAF) is compared with GSAF and MGSAF.

4.3 Simulation Results

The effect of node density on the metrics described in the preceding section can be inferred from Figure 3, Figure 4, and Figure 5. It can be observed that higher node density is conducive to message delivery. With more nodes to route messages, delivery probability increases and then plateaus at around 77%. The value plateaus because more nodes do not translate to more effective message exchange. After all, the number of copies is limited by ticket value. The delivery probability of SMGSAF falls between GSAF and MGSAF. A multi geocasting framework will have better delivery probability because there are more destination nodes than unitary geocasting. Therefore both MGSAF and SMGSAF outperform GSAF in terms of delivery probability. It was mentioned in a previous section that the delivery probability for SMGSAF is calculated based on messages and not geomessages, which is what is observed for MGSAF and GSAF. In SMGSAF as every message depends on multiple geomessages for delivery, every message has a lower prob-
ability of being delivered than its carrier geomessage. Thus we can see SMGSAF underperforming MGSAF. What was encouraging to observe was that SMGSAF does not underperform MGSAF by a huge margin and the tradeoff between security and delivery probability seems acceptable.

The overhead ratio can be seen to decrease with increasing node density. This can be explained by the fact that the number of the delivered message increases faster with node density than the number of relayed messages. This happens because the number of times a geomessage is copied depends more on ticket value than node density. As a result, we see decreasing overhead ratio with increasing node density. When SMGSAF is compared to MGSAF and GSAF, it has large overhead as compared to the latter two. This can be explained by the fact that in SMGSAF every message leads to the creation of multiple geomessages. Therefore, SMGSAF has huge overhead when compared with other protocols. Notably, MGSAF has a lower overhead than GSAF and this might be because of more successful delivery in MGSAF than GSAF.

Average latency also has an inverse relationship with node density. With more nodes, more messages are delivered more quickly and the average latency decreases. The decrease in latency though tapers when more and more nodes are added to the network. This happens for the same reason for which delivery probability saturates. More number of nodes does not necessarily mean more relays in the network as the number of copies is limited by ticket value. SMGSAF does outperform GSAF in this respect but not by a large margin. This can be attributed to the increase in the number of recipients which brings the average value of delay down. MGSAF outperforms both of them.

Nodes in OppNet follow a store-carry-forward system and hence the memory capacity of the nodes matters. Buffer capacity is used by nodes to carry geomessages around the network. When they receive new geomessages and do not have enough space in the buffer, they drop old ones. The dropping of geomessages impedes their delivery, though not necessarily due to redundancy. Yet more buffer capacity allows more geomessages to survive for a long time in the network before being dropped. As a result delivery probability increases which is evident from Figure 6. As more geomessages are successfully delivered, and the number of copies is limited by ticket value overhead ratio also decreases as can be seen in figure Figure 7. It can be argued that as fewer geomessages are dropped, more of their copies should be relayed in the network causing the number relayed messages to increase. This is true, but the rate of increase in the number of delivered messages prevails the increase in the number of relayed messages. Average latency, on the other hand, can be seen to increase with buffer size in Figure 8. The relative performance of SMGSAF, MGSAF, and GSAF have shown a similar trend as they had shown against node density, thus the arguments for it remain the same.

It must be pointed out here that large buffer capacity is not a silver bullet for any OppNet’s performance. Large buffers are not only costly but the duration for which nodes in an OppNet interact is low and all the contents of a large buffer may not be successfully exchanged during all the encounters. To reap the benefits of the larger buffer, network interfaces that have high throughput are required.

Every geomessage is valid only for a specific period after its creation. This is so because most information is useful only for a small period, for example, a weather update is only good for a day. If a node keeps carrying information that has expired, it simply wastes precious buffer space and network bandwidth. Therefore Time To Live is a metric that
plays a key role in keeping a network running efficiently. As can be seen in Figure 9, Figure 10, and Figure 11, a larger TTL allows geomessages to live longer in the network and thus lead to more successful delivery. The increase in delivery also causes the overhead to decrease. Message latency increases with the duration for which the messages live.

It has frequently come up before that ticket value is the main mechanism to control the redundancy in the network. A lower ticket value will not allow enough copies of a geomessage to be created in the network so that it can be delivered efficiently. When routing decisions are random a few stray geomessages may here and there reach their destination casts, but this assumption cannot be made for the ample number of geomessages. On the other hand, a large ticket value will lead to a lot of copies of geomessage being generated which will lead to filling up the buffer space in nodes. This, in turn, will cause a lot of geomessages to be dropped and from the observations, it can be inferred that the redundancy resulting from a large number of copies, does not offset the effects of nodes dropping geomessages more frequently.

From Figure 12, it can be seen that low initial ticket value results in low delivery probability as was discussed earlier. For larger ticket values, again the delivery probability is less with the maxima occurring at 3, which we have used as default ticket value. It is imperative to choose an optimal ticket value such that the network performs efficiently.
The average latency is seen to decrease with ticket value in Figure 13. Our understanding is that as more and more geomessages are created in the system, they are delivered more frequently and as the number of successful deliveries increases the average also comes down.

In Figure 14, we see that the overhead ratio increases with an increase in ticket value. For low ticket values, the delivery rate is so low that the overhead is high compared to the optimal ticket value of 3. Increasing ticket value anymore simply results in more copies of geomessages being created and relayed in the network causing the overhead to increase in an almost exponential fashion.

In Figure 15 to Figure 18, we have conducted simulations in attack based environment. For these simulations, we have considered only multi-GSAF and proposed SMGSAF scheme. The number of malicious nodes [5, 10, 15, 20, 25, and 30] are present in the existing network as mentioned in above scenario as number of nodes. Hence, we have shown the evaluation of our proposed protocol and compared it with existing ones. We saw that SMGSAF performs better than MGSAF in some extent.

5 Conclusion

In this work, the author presented a novel secure multi-geocasting based routing scheme for opportunistic networks (called SMGSAF), which a secret sharing scheme that can be coupled with disjoint path routing to preserve the privacy of messages in an opportunistic network. Simulation results have shown that SMGSAF outperforms MGSAF and GSAF, chosen as benchmark routing protocols for OppNets, in terms of average latency, overhead ratio, and delivery ratio. We studied the effect of node density, buffer size, message lifetime, and initial ticket value of geomessages on the network working under the SMGSAF protocol. We discovered that SMGSAF provides the security described above, at a slight cost of network performance. It underperformed MGSAF, but the difference is within acceptable limits considering the increased security. SMGSAF did outperform unitary GSAF and it bolsters the effectiveness of GSAF as a geocasting mechanism. Also, from a practical perspective, SMGSAF does not require high resources, high computation, or any form of key exchange. Further, the authors try to enhance the performance of SMGSAF on real mobility traces such as the ones provided in as future work.

Declarations

Funding: Not Applicable
Conflicts of interest/Competing interests: The authors declare no conflict of interest.
Availability of data and material: Not Applicable
Code availability: Not Applicable
Author’s contributions: Design of the protocol and methodology, J.S. and S.K.D; algorithm implementation, R.K. and J.S.; simulations and validation, G.G.; analysis, J.S. and S.K.D; writing-review and editing, all the authors. All authors have read and agreed to the published version of the manuscript.

References

1. Lilien L, Kamal Z H, Bhuse V, Gupta A. (2006) Opportunistic Networks: The Concept and Research Challenges in Privacy and Security. In: International Workshop on Research Challenges in Security and Privacy for Mobile and Wireless Networks (WSPWN), Miami, FL, USA, pp. 1 – 14.

2. Khan R, Kumar N, Obaidat MS (2018) LA-MHR: Learning automata based multilevel heterogeneous routing for opportunistic shared spectrum access to enhance lifetime of WSN. IEEE Systems Journal, vol. 13 issue 1 pp. 313 – 323.

3. Alajeely, M., Doss, R., & Ahmad, A. A. (2016) Security and trust in opportunistic networks – a survey. IETE Technical Review, 33(3), pp. 256 – 268.

4. Khan, R., Kumar, P., Jayakody, D. N. K., & Liyanage, M. (2019) A survey on security and privacy of 5G technologies: Potential solutions, recent advancements and future directions. IEEE Communications Surveys & Tutorials, pp. 196 – 248.

5. Tourani, R., Misra, S., Mick, T., & Panwar, G. (2017) Security, privacy, and access control in information-centric networking: A survey. IEEE communications surveys & tutorials, 20(1), pp. 566 – 600.

6. Dhurandher SK, Kumar A, Obaidat MS (2017) Cryptography-based misbehavior detection and trust control mechanism for opportunistic network systems. IEEE Systems Journal, 12(4), pp. 3191-3202.

7. Dhurandher SK, Mehra V (2009) Multi-path and message trust-based secure routing in ad hoc networks. In: International Conference on Advances in Computing, Control, and Telecommunication Technologies. IEEE, pp. 189 – 194.

8. Rajaee, A., Chalmers, D., Wakeman, I., & Parisis, G. (2018) Efficient geocasting in opportunistic networks. Computer Communications, 127, pp. 105 – 121.

9. Maihofer, C. (2004) A survey of geocast routing protocols. IEEE Communications Surveys & Tutorials, 6(2), pp. 32 – 42.

10. N. Derakhshanfard, M. Sabaei, & A. M. Rahmani (2017) CPTR: conditional probability tree based routing in opportunistic networks. Wireless Networks, 23, pp. 43 – 50.

11. J. Wu, Z. Chen, M. Zhao (2020) An efficient data packet iteration and transmission algorithm in opportunistic social networks. Journal of Ambient Intelligence and Humanized Computing, volume 11, pp. 3141 — 3153.

12. M. Cuka, D. Elmazi, K. Bylykbashi, E. Spaho, M. Ikeda, L. Barolli (2019) Implementation and performance evaluation of two fuzzy-based systems for selection of IoT devices in opportunistic networks. Journal of Ambient Intelligence and Humanized Computing, volume 10, pp. 519 – 529.

13. A. Ghaffari (2020) Hybrid opportunistic and position-based routing protocol in vehicular ad hoc networks. Journal of Ambient Intelligence and Humanized Computing, volume 11, pp. 1593 – 1603.

14. S. Sankar Ganesh & G. Ravi (2020) Real time link quality based route selection and transmission in industrial Manet for improved QoS. Journal of Ambient Intelligence and Humanized Computing, https://doi.org/10.1007/s12652-020-02331-1.

15. Shanshan Lu and Yonghe Liu, (2014) Geoopp: Geocasting for opportunistic networks. In: Wireless Communications and Networking Conference (WCNC), IEEE, pp. 2582 — 2587.