ALARM: Anonymous Location-Aided Routing in Suspicious MANETs

Karim El Defrawy, *Member,* Gene Tsudik, *Senior Member*
Computer Science Department
University of California, Irvine, CA, USA

Abstract— **In most common mobile ad-hoc networking (MANET) scenarios, nodes establish communication based on long-lasting public identities. However, in some hostile and suspicious settings, node identities must not be exposed and node movements should be untraceable. Instead, nodes need to communicate on the basis of their current locations. While such MANET settings are not very common, they do occur in military and law enforcement domains and require high security and privacy guarantees. In this paper, we address a number of issues arising in suspicious location-based MANET settings by designing and analyzing a privacy-preserving and secure link-state based routing protocol (ALARM). ALARM uses nodes' current locations to securely disseminate and construct topology snapshots and forward data. With the aid of advanced cryptographic techniques (e.g, group signatures), ALARM provides both security and privacy features, including: node authentication, data integrity, anonymity and untraceability (tracking-resistance). It also offers protection against passive and active insider and outsider attacks. To the best of our knowledge, this work represents the first comprehensive study of security, privacy and performance trade-offs in the context of link-state MANET routing.**

Index Terms— **Privacy, Communication system security, Communication system routing, Mobile communication, Location-based communication, Military communication.**

I. Introduction

During the last two decades, research in various aspects of mobile ad-hoc networks (MANETs) has been very active, motivated mainly by military, disaster relief and law enforcement scenarios. More recently, location information has become increasingly available through small and inexpensive GPS receivers, partially prompted by the trend of introducing location-sensing capabilities into personal handheld devices [36]. A natural evolutionary step is to adopt such location-based operation to MANETS. This results in what we term *location-based MANETS*. In such a MANET, devices rely on location information in their operation. The main distinguishing feature of the envisaged location-based MANET environment is the communication paradigm based not on permanent or semi-permanent identities, addresses or pseudonyms, but on *instantaneous node location*. In other words, a node (A) decides to communicate to another node (B), depending on exactly where (B) is located at present. If node location information is sufficiently granular, a physical MANET map can be constructed and node locations – instead of persistent node identities – can be used in place of network addresses. In some applications, such as military, law enforcement and search-and-rescue, node identities are not nearly as useful as node locations. Such critical settings have certain characteristics in common. First, node location is very important – knowledge of the physical as opposed to logical or relative topology enables avoiding wasteful communication and focusing on nodes located within a specific area. Second, critical settings must contend with security and privacy attacks. Security attacks might attempt to distribute false – or impede propagation of genuine – routing information. Whereas, privacy attacks aim to track nodes as they move.

When the operating environment is *hostile*, as is the case in military and law enforcement settings, node identities must not be revealed. We use the term "hostile" to mean that communication is being monitored by adversarial entities that are not part of the MANET. If we further assume that genuine MANET nodes do not even trust each other (perhaps because of possible node compromise, i.e., the environment is "suspicious"), the need to hide node identities becomes more pressing. Also, in this setting, it is natural for node movements to be obscured, thus making it impossible (or, at least, very difficult) to track a node, even without knowing its identity. While such suspicious and hostile MANET environments might not be very common, they do occur in military and law enforcement domains and require high security and privacy guarantees.

In this paper, we consider what it takes to provide privacy-preserving secure communication in hostile and suspicious MANETS. We construct a protocol for Anonymous Location-Aided Routing in MANETS (ALARM) that demonstrates the feasibility of simultaneously obtaining, strong privacy and security properties, with reasonable efficiency. In this context, privacy means node anonymity and resistance to tracking. Whereas, security includes node/origin authentication and location integrity. Although it might seem that our security and privacy properties contradict each other, we show that some advanced cryptographic techniques can be used to reconcile them.

The rest of this paper is organized as follows: We discuss design choices and assumptions in Sections II and III, followed by description of the adversarial model in Section IV. The ALARM protocol is presented in Section V and its security is analyzed in Section VI. Performance analysis and simulation results are discussed in Sections VII and VIII, followed by an overview of related work in Section IX. The paper concludes with a summary in Section X.

II. Design Choices

We begin by justifying our design choices, in particular, the use of link-state routing. We then overview the cryptographic construct of *group signatures* – one of the principal building blocks in our protocol.

A. Routing Protocol Choices

MANET routing protocols can be roughly partitioned into two groups: *reactive* (or on-demand) and *proactive*. The latter can be further broken down into *link-state* and *distance-vector* (including path-vector) protocols. Reactive protocols typically use route discovery to identify a route to a given destination. The notion of discovering the destination is premised upon the source *knowing* the persistent identity or address of the destination. This assumption is invalid in our MANET scenario, since the destination is selected based on its current location, which is not known to the source *a priori*. Consequently, we claim that reactive routing protocols are unsuitable for the problem at hand.

Distance vector (DV) protocols [32] inherently offer relatively weak levels of security. A single compromised node can easily create any number of phantom node-location entries and propagate them to the entire MANET, thus “poisoning” everyone’s DV tables. This issue can be addressed, in principle, by using a path vector protocol (e.g., BGP [5]) along with some security enhancements (e.g., BGP-SEC [21]) where each Source-Destination path component is signed. However, verifying $O(n*r)$ signatures, where n is the number of nodes an r is the network diameter, would be very expensive. Also, as is well-known, DV protocols exhibit slow convergence, which can be problematic in highly-mobile MANETs.

The alternative is link-state (LS) routing protocols, such as OLSR [26]. One advantage of LS protocols is that, unlike their reactive counterparts, they obviate the need for route discovery. This makes LS protocols suitable for real-time applications that impose strict delay constraints. On the other hand, LS protocols do not scale well due to excessive broadcasting – n updates flooded throughout the MANET for each update period. However, this has been mitigated in OLSR by reducing the number of nodes that forward routing control messages to a subset of the first hop neighbors of any node, called multipoint relays (MPRs). In addition, since our goal is to accommodate relatively modest-sized MANETs (on the order of tens or few hundreds of nodes), scalability can be easily achieved. (This is discussed further in Section VII). Furthermore, LS allows us to achieve stronger security, since origin authentication and integrity of LS updates can be easily supported. There are a number of well-known techniques that achieve this, e.g., [38] and [3], [35].

The main challenge arises from the need to reconcile security and privacy (anonymity and untraceability) requirements that we address below. Based on the above discussion, we consider link-state to be best-suited for supporting location-based routing with the privacy and security features described earlier. In the rest of this paper, we use a simple flooding-based scheme to illustrate the operation of ALARM. However, we note that any optimization for reducing LS flooding overhead (e.g., MPR-based flooding in OLSR), can be easily integrated into ALARM.

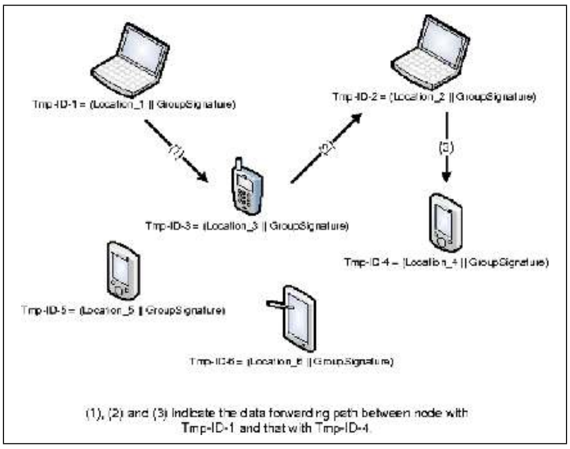

Tmp-ID-1 = (Location_1 || GroupSignature)
Tmp-ID-2 = (Location_2 || GroupSignature)
(1)
(2)
(3)
Tmp-ID-3 = (Location_3 || GroupSignature)
Tmp-ID-4 = (Location_4 || GroupSignature)
Tmp-ID-5 = (Location_5 || GroupSignature)
Tmp-ID-6 = (Location_6 || GroupSignature)
(1), (2) and (3) indicate the data forwarding path between node with Tmp-ID-1 and that with Tmp-ID-4.

Fig. 1. MANET Topology Snapshot in ALARM

B. Group Signatures

Group signatures can be viewed as traditional public key signatures but with additional privacy features. In a group signature scheme, any member of a large and dynamic group can sign a message, thereby producing a *group signature*. (However, each member has its own unique private key, as described in the Appendix). A group signature can be verified by anyone who has a copy of a constant-size group public key. A valid group signature implies that the signer is a *genuine* group member. At the same time, given two valid group signatures, it is computationally infeasible to decide whether they are generated by the same (or different) group members. Furthermore, in case of a dispute over a group signature, a special entity called a Group Manager (GM) can open a group signature and identify the actual signer. This important feature is called *Escrowed Anonymity*. Based on the above, it seems that group signatures are a perfect fit for our envisaged MANET setting. A mobile node can periodically sign its current location (link-state) information without fear of being tracked, since multiple group signatures are not linkable. At the same time, anyone can verify a group signature and be assured that the signer is a legitimate MANET node. (A more detailed description of group signatures can be found in the Appendix).

Table I shows timings for group signature generation and verification, compared to standard Elliptic Curve DSA (EC-DSA) measured using OpenSSL [2][1]. Measurements are reported as in [10]. They were obtained on a 1.5GHz Centrino processor. The processing power used is a close approximation of the European Union Cooperative Vehicle-Infrastructure System (EU-CVIS) vehicle PC, a platform adopted for future development of vehicular ad-hoc networks (VANET) applications [1].

[1] Note that security levels on elliptic curves correspond to 1024-bit security in RSA-like settings.

Scheme	Security Level (bits)	Sign (sec)	Verify (sec)	Signature Size (bytes)
ECDSA-192	80	$5 \cdot 10^{-4}$	$3 \cdot 10^{-3}$	48
ECDSA-256	120	$8 \cdot 10^{-4}$	$4.2 \cdot 10^{-3}$	64
GSIG	80	$1.7 \cdot 10^{-2}$	$1.56 \cdot 10^{-2}$	151
GSIG	128	$5.37 \cdot 10^{-2}$	$4.93 \cdot 10^{-2}$	225

TABLE I

COMPUTATION COSTS, SIGNATURE AND KEY SIZE FOR A GROUP SIGNATURE (GSIG) [7] AND EC-DSA (OBTAINED FROM [10]).

PK_X, SK_X	Public and corresponding secret key of node X
K_X^Y	Symmetric key shared by nodes X and Y
$E_K(m)$	Encryption of m with key K, where K is either: (1) symmetric key shared between two nodes, or (2) node X's public key PK_X
TS	Time-stamp of current time slot
$H()$	Cryptographic hash function (e.g. SHA-2)
$M_1 \| \| M_2$	Concatination of M_1 and M_2
$GSig(M_1 \| \| M_2)$	Group signature on concatenation of M_1 and M_2
σ	A normal public key signature (e.g. RSA signature)

TABLE II

NOTATION SUMMARY

III. ASSUMPTIONS AND GOALS

The following assumptions are necessary in ALARM:
- [LOCATION] Universal availability of location information: each node is equipped with a device that provides accurate positioning information, e.g., GPS.
- [MOBILITY] Sufficiently high mobility: a certain minimum fraction (or number) of nodes move periodically, such that tracking a given mobile node from one topology snapshot to the next requires distinguishing it among all nodes that have moved in the interim.
- [TIME] All nodes maintain loosely synchronized clocks. This is easily obtainable with GPS.
- [RANGE] Nodes have uniform transmission range. Once a node knows the current MANET map, it can determine node connectivity (i.e., transform a map into a graph) [2].

ALARM has the following goals:
- [PRIVACY] There are no public node identities or addresses. Each node is anonymous and its occurrences at different locations (movement patterns) cannot be linked; we elaborate on this later.
- [SECURITY] The network must be resistant to passive and active attacks stemming from both outsiders and malicious (e.g., compromised) insiders.
- [PERFORMANCE] Security and privacy goals must be achieved without undue sacrifices in performance (i.e., without requiring excessive computations and/or high delay).

IV. ADVERSARIAL MODEL

As stated earlier, we are concerned with both outsider and insider adversaries and attacks. However, our adversarial model does not take into account adversaries that physically track nodes, e.g., visually or using physical-layer signal fingerprinting. Furthermore, we **do not consider** adversaries that mount denial-of-service (DoS) attacks by creating sinkholes, wormholes and other topological abnormalities.

A. Outsiders

An outsider can be *passive* or *active*. It does not have any keys used for encryption or authentication. Its goal is to violate privacy, security or both. A *passive outsider* eavesdrops on all communication and aims to compromise privacy, i.e., track nodes. It does not engage in any active attacks (i.e., does not inject, modify and replay any messages). By definition, a passive outsider can not be stronger than a passive insider that has encryption and authentication keys. By providing protection against passive insiders (see below), protection against passive outsiders is obtained for free. An *active outsider* can inject, modify and replay messages. Its goals can include: disruption of routing, node impersonation and creation of phantom nodes, e.g., via Sybil attacks. An active outsider does not know any keys and is not stronger than an active insider.

B. Passive (Honest-but-Curious) Insiders

A *passive* insider possesses all cryptographic keys used for network-wide encryption/authentication. It can eavesdrop on all exchanged messages and outwardly behaves correctly by following all rules and protocols. In other words, it sends no fraudulent messages, does not attempt to impersonate other nodes and does not delete or modify other nodes' traffic. Behaving otherwise would attract attention and could result in eventual detection and exposure. However, a passive insider is not assumed to be *silent*, i.e., its communication patterns are not different from those of non-malicious nodes. A passive insider can also attempt to track other nodes' movements by linking different location announcement messages or using trajectory information.

C. Active Insiders

An active insider is the most powerful adversary type. It can modify, inject and replay "genuine" messages. In more traditional MANET settings, the identity of each node is known and the power of the active insider is constrained, since its activity can be detected and/or traced. However, since privacy is one of our main goals, nodes have no persistent identities. Therefore, an active insider can easily modify or inject seemingly genuine routing messages, thus masquerading as other nodes. Concretely, we consider two kinds of active insider attacks:

- *Sybil attack:* adversary creates one or more phantom nodes by generating fake routing control messages ostensibly from these nodes' locations. Even though these routing messages contain valid authentication information (e.g., signatures), other nodes cannot link them to the originating malicious node.

[2] If transmission range is not uniform, each node should include its transmission range in its location announcement message.

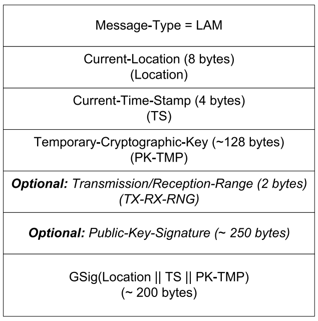

Message-Type = LAM
Current-Location (8 bytes)
(Location)
Current-Time-Stamp (4 bytes)
(TS)
Temporary-Cryptographic-Key (~128 bytes)
(PK-TMP)
Optional: Transmission/Reception-Range (2 bytes)
(TX-RX-RNG)
Optional: Public-Key-Signature (~ 250 bytes)
GSig(Location || TS || PK-TMP)
(~ 200 bytes)

Fig. 2. ALARM LAM Message Format

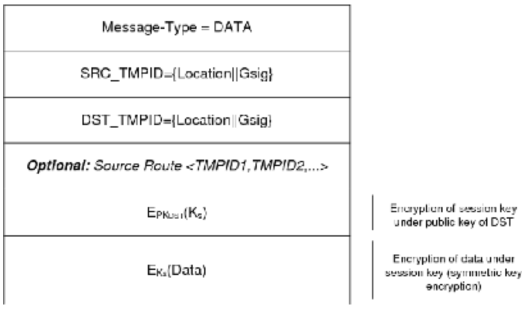

Message-Type = DATA
SRC_TMPID={Location||Gsig}
DST_TMPID={Location||Gsig}
Optional: Source Route <TMPID1,TMPID2,...>
E_PKDST(Ks)
Encryption of session key under public key of DST
E_Ks(Data)
Encryption of data under session key (symmetric key encryption)

Fig. 3. ALARM Data Message Format

- *Location fraud*: adversary lies about its own location. This can be harmful in situations where node communication is location-centric. For example, a malicious insider claiming a certain fake location can result in attracting (or repelling) traffic.

We note that the insider adversary is clearly not restricted to either attack type, i.e., it is free to blend them.

V. ALARM Protocol

This section describes basic operation of ALARM and its limitations. It then outlines several extensions that mitigate such limitations.

A. Basic Operation

The basic steps in ALARM's operation are as follows:

1- Initialization (Off-line)

a) The group manager (GM) initializes the underlying group signature scheme and enrolls all legitimate MANET nodes as group members. During this phase, each member (node) creates a unique private key (SK_{member}), that is not revealed to anyone. This key is needed to produce valid group signatures. It also creates a corresponding public key (PK_{member}), that is revealed only to the GM. In addition, each member learns the common group public key (PK_{GM}) that is subsequently used to verify group signatures. In case of a dispute and for *off-line* forensics, GM is responsible for opening any contested group signatures and determining actual signers.

b) Depending on the specific group signature scheme, GM might also handle future joins for new members as well as revocation of existing members. However, in most envisaged MANET scenarios, membership is likely to be fixed, i.e., all joins can be done in bulk, before deployment. Also, revocation might not be feasible or desired, since it would require propagating – in real-time – updated revocation information to all legitimate nodes. However, if dynamic membership is necessary, ALARM can support it, with minor additional assumptions.

2- Operation (On-line)

a) Time is divided into equal slots of duration T. At the beginning of each slot, each node s generates a temporary public-private key-pair: *PK-TMP*$_s$ and *SK-TMP*$_s$, respectively. *PK-TMP*$_s$ is subsequently used by other nodes to encrypt session keys to establish secure channels with s. Note that these keys can be generated off-line.

b) Each node broadcasts a Location Announcement Message (LAM), containing: its location (GPS coordinates), time-stamp, temporary public key (*PK-TMP*$_s$) and a group signature computed over these fields. Each LAM is flooded throughout the MANET (more on the overhead and scalability of the flooding process in Section VII). Figure 2 shows the LAM format used to construct the network topology snapshot in Figure 1. The sequence of steps required for sending a LAM is shown in the flow chart in Figure 5.

c) Upon receipt of a new LAM, a node first checks that it has not received the same LAM before, it then verifies the time-stamp and group signature. If both are valid, the node re-broadcasts the LAM to its neighbors. Having collected all current LAMs, each node constructs a geographical map of the MANET and a corresponding node connectivity graph. A flowchart describing this sequence of steps is shown in Figure 6.

Between successive LAMs, a node can be reached (addressed) using a temporary pseudonym formed as: current location concatenated with the group signature in the last LAM ($TmpID = \{Location||GSig\}$). Note that the pseudonym represents a valid address even if the actual node moves in the interim. The location is included in the pseudonym in order to minimize required state and assist in the forwarding process [3]. If the location is not part of the pseudonym, a node forwarding a message to a pseudonym would have to look up the associated location and decide how to forward to that location. (See below for more details on the forwarding process). Including location in the pseudonym speeds up the forwarding process and requires fewer look-ups.

d) Whenever a node desires to communicate with a certain location, it checks to see if any node currently exists at (or near) that location. If so, it sends a message to the destination's current pseudonym ($TmpID$). This message is encrypted with a session key using a symmetric cipher. The session key is, in turn, encrypted under the current public key (*PK-TMP*) included in the destination's latest LAM.

[3] An earlier version of ALARM [18] had the pseudonym consisting only of the group signature.

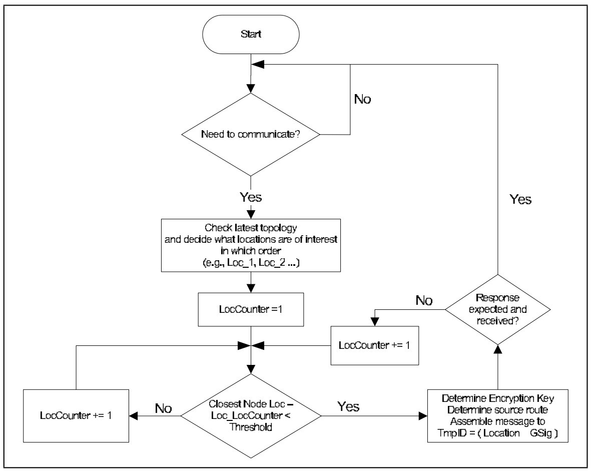


Fig. 4. ALARM Communication Decision Flow-Chart

When the destination receives the message, it first recovers the session key and uses it to decrypt the rest.

ALARM is not restricted to any specific public key technique. One obvious choice is Diffie-Hellman (DH) [16], whereby each LAM includes an ephemeral (period-specific) DH half-key. The sender then simply generates its own DH half-key, computes a shared key and encrypts the session key with it. Clearly, the sender's half-key must be included in the clear-text part of the message. Other key agreement schemes can also be used. The sequence of steps involved in determining a destination node is shown in Figure 4.

e) *Forwarding:* As described above, nodes disseminate current topology by periodically flooding LAMs. Once each node has the entire topology view, it decides whether to communicate with a certain location (node). Message forwarding is independent of topology dissemination. One option is for a node to create a source route, explicitly encoding locations of nodes on the path to the destination. The actual path can be computed using the shortest path algorithm or any other location-aided routing algorithm, such as [33], [25] or [29]. For example, consider the simple topology of Figure 1. Assume that the node at location1 ($TmpID1 = \{Location1||GSig1\}$) requires sending a message to another node at location4 ($TmpID4 = \{Location4||GSig4\}$). The sender calculates the route to location4 and determines that it has to pass through location2 and location3. It then generates a session key (K_s) and encrypts data with that key using a symmetric cipher (e.g., AES). It then uses the public key in the last LAM of location4 to encrypt K_s and assembles a data message with the destination set to ($TmpID4$) and source – to ($TmpID1$). It finally composes a source route: $< TMPID2, TMPID3 >$.

3- Forensics (Optional, off-line) Each node logs all sent and received LAMs (except duplicates). Collectively, this information constitutes an operational log that is, after each field deployment, transferred to an off-line server, e.g, GM. All LAMs collected by all nodes are then reconciled and, in the process, all group signatures are verified and opened by GM. Each group signature's originator is thus identified. This process allows most insider misbehavior, such as Sybil attacks, to be detected post factum. The only insider attacks that might not be identifiable using logs is location fraud. (This is discussed in Section VI).

In general, operational logs are used for accountability purposes by allowing GM to reconstruct the exact sequence of node movements and topology snapshots. We stress that this is an optional procedure that does not incur any additional overhead (beyond storage) during on-line operation of ALARM. Assuming LAM size of 350 bytes (8 for location, 4 for time-stamp, 128 for temporary key, and 200 for short group signature [6]), a network of 100 nodes deployed for a week and topology update frequency of 10 LAMs per minute, combined storage for *all* operational logs would amount to around 3.5 GBytes.

4- ALARM Limitations The main advantage of the basic ALARM protocol is its simplicity and effectiveness. However, it has two notable drawbacks: (1) since flooding is used to disseminate LAMs, scalability becomes problematic for large MANETS (thousands of nodes); (2) any node can lie about its location or generate multiple LAMs as part of a Sybil attack.

B. Extensions

We now describe some extensions to the basic ALARM protocol that address scalability and insider threat issues.

1) Scalability: If a MANET is sufficiently large for flooding to cause significant overhead, a hierarchical approach can be used to limit its scope. Similar ideas have been explored in GeoGRID [33] and OLSR [26]. In GeoGRID, the network is partitioned into logical grids, with a single elected node acting as a gateway for each partition. Only gateways forward packets to other gateways, which limits the scope of flooding. In OLSR, each node selects only a subset of its immediate neighbors – each called a multi-point relay (MPR) – that forwards its routing control messages. MPRs are selected such that there is a route to every second-hop neighbor through one MPR. MPR selection was shown to significantly reduce routing overhead without worsening routing performance. In Section VII, we explore routing control overhead in ALARM and show how it affects scalability.

2) Group Signatures with Self-Distinction: As discussed above, ALARM takes advantage of group signatures to simultaneously obtain node anonymity and authentication. Any group signature scheme can be used with ALARM to protect against attacks by outsiders and passive (honest-but-curious) insiders. However, if resistance to Sybil attacks is needed, the underlying group signature scheme must offer the additional *self-distinction* feature.

Self-distinction is an optional feature that is offered by (or that can be added to) some group signature schemes, such as [4] and [48]. It prevents attacks involving a genuine group member who signs multiple messages all purpotedly originated by distinct signers. In our suspicious MANET context, this feature can precisely address Sybil attacks, where a legitimate node assumes several pseudonyms and pretends to be at

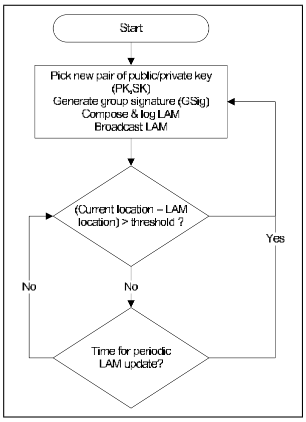

Start
Pick new pair of public/private key (PK,SK)
Generate group signature (GSig)
Compose & log LAM
Broadcast LAM
(Current location – LAM location) > threshold ?
Yes
No
No
Time for periodic LAM update?

Fig. 5. ALARM Sender Process

several locations at once. Self-distinction seems to contradict what group signatures try to achieve, i.e., anonymity and unlinkability. However, in our context, self-distinction implies that each node can have at most *one* identity within a given LAM interval. Thus, node privacy *across* time slots is still preserved.

Two examples of group signatures with self-distinction are [48] and [4]. The intuition behind these constructs is that a signer (group member) proves its distinction from others while signing a message. This is achieved by having nodes first agree on some common parameter, e.g., a common random number. This parameter varies for each round of signing. If a node uses the same parameter to sign twice within the same round, the two group signatures would have matching components that would immediately signify misbehavior. The challenge with adopting such schemes in ALARM is in generation of this common parameter. One straightforward – but inefficient – approach is to run a group key agreement protocol at the beginning of every time-slot and use the resulting group key as the common parameter. This is clearly unscalable. An alternative and more efficient approach is is to use a group key agreement protocol just once, in order to agree on the initial common parameter. Another possibility is for GM to generate and distribute this starting value.

3) One-Time Certificates: Group signatures offer a number of benefits. Any node receiving a LAM can verify that it was produced by a legitimate peer. At the same time, node pseudonyms are unlinkable, which inhibits tracking. Also, no two nodes have the same pseudonym, even if they are at the same exact location, at the same time. Despite their advantages, group signatures are expensive in terms of generation and verification costs as well as size (as shown in Table I). There is still an order of magnitude difference in both computational and storage/bandwidth cost between group signatures and their plain counterparts.

An alternative approach that emulates the functionality of group signatures is using one-time certificates. Initially, an off-line Certification Authority (CA) issues to each node (N_i) a number of public key certificates: $C_i^1, ..., C_i^m$ where m is the maximum number of time-slots for a given MANET deployment. Each certificate (C_i^j), includes the following fields:

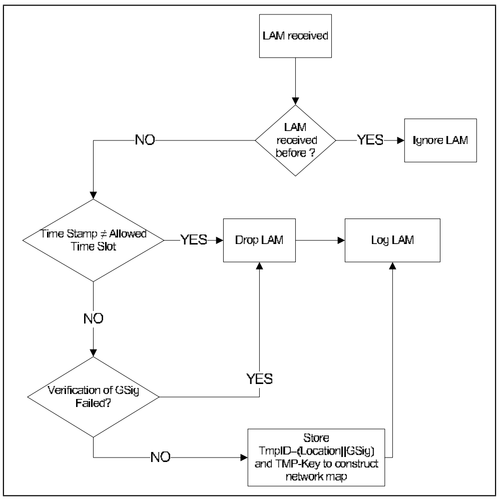

LAM received
LAM received before ?
YES
Ignore LAM
NO
Time Stamp ≠ Allowed Time Slot
YES
Drop LAM
Log LAM
NO
YES
Verification of GSig Failed?
NO
Store TmpID–(Location||GSig) and TMP-Key to construct network map

Fig. 6. ALARM LAM Receiver Process

1) Unique public key (PK_i^j) for a plain (non-group) signature scheme, e.g., RSA or DSA. We assume that the specific signature scheme is global and fixed beforehand.
2) Time-stamp indicating the future (j-th) time-slot when this certificate can be used.
3) CA's signature of the certificate: σ_i^j

The public-private key-pair for each certificate can be either generated by CA or by each node independently. In the latter case, CA has to make sure that all PK_i^j-s are unique across all nodes. For each C_i^j, a node is assumed to know the corresponding private key (SK_i^j).

To estimate storage requirements, consider MANET deployment of one week with 10 LAM updates per minute. A total of $7 \cdot 24 \cdot 60 \cdot 10 = 100080$ one-time certificates will be required. Assuming standard X.509-type format [50] with a certificate size of 1KB, each node requires 100MB of storage. This is reasonable for modern PDA-class MANET nodes.

The operation of ALARM with one-time certificates is slightly different from the description in Section V-A:

- When constructing a LAM for current time-slot (j), each node (i) includes the entire certificate (C_i^j) in its LAM, instead of *PK-TMP* only.
- Each LAM contains a signature (σ) with SK_i^j, corresponding to PK_i^j included in C_i^j. Recall that C_i^j can only be used in the current time-slot.
- Upon receipt of a LAM, each node checks if the time-stamp and the certificate in the LAM match the current time-slot. It then validates the certificate C_i^j by checking CA's signature. Finally, it verifies LAM signature (σ) using PK_i^j extracted from C_i^j. If verification succeeds, it logs and re-broadcasts the LAM.

It is easy to see that, as long as all PK_i^j values are independent, linking multiple LAM-s originating from the same node is infeasible. Moreover, one-time certificates offer effective and inexpensive mitigation of most insider attacks. This is because each node only knows its own sequence of one-time certificates and corresponding secret keys. Sybil attacks are prevented by tying each certificate to a fixed time-slot and only allowing (via controls by the issuing off-line CA) the use of one certificate per node, per time-slot. The only insider attack not addressed here is insider location fraud.

The main drawback of one-time certificates is the requirement to pre-determine maximum duration of MANET deployment. Another issue is additional storage for certificates. On the other hand, both generation and verification of LAM signatures is much faster than with group signatures.

4) Sequential Aggregate Signatures (SAS) : This extension leverages the fact that each node already includes a temporary public key in its LAM. A node first sends its own LAM before forwarding LAMs of other nodes. A node can use its private key to sign other forwarded LAMs. Such signatures can be aggregated (e.g., Sequential Aggregate Signatures) to maintain a constant size LAM. An adversary launching an active attack (by generating phantom nodes, impersonating other nodes and/or lying about its location) will be detected due to mismatching signatures in received LAMs. Note that these are *not* group signatures, but sequential aggregate signatures (SAS) that are constant in size.

A similar approach has been used to secure route discovery in the DSR routing protocol in [28]. One such SAS construct is based on RSA [34] and its signature generation cost is equivalent to a plain RSA signature. Verification cost, on the other hand, increases linearly with number of signers (nodes) on the path. However this cost can be minimized by using small public exponents (e.g., 3 or 17). Such small exponents speed up verification by a factor of ten [28]. We demonstrate how this extension would operate with an example based on the SAS scheme from [34]:

1) Assume that a node's *i*-th private key is $SK_i = x_i$ and its public key PK_i consists of the pair (n_i,y_i), where $x_i y_i = 1(mod \phi(n_i))$. This is a typical RSA [41] setting.
2) The only requirement for the RSA-based SAS scheme is for all modulii to be of roughly the same length. The signature expands by t bits $b_1, b_2, ...b_t$ where t is the number of signers in the aggregate signature.
3) During operation, if the *i*-th signature $\sigma_i \geq n_{i+1}$ then b_i is set to 1; otherwise, it is set to 0. During verification phase, if $b_i = 1$ then n_{i+1} is added to σ_i before proceeding with the verification of σ_i.
4) Consider the following example: assume that node A sends a LAM though nodes B and C to reach D, the signing procedure is as follows:
   a) A: computes $h_A = H(LAM, (n_A, y_A))$ and $\sigma_A = (h_A)^{x_A}(mod\ n_A)$. σ_A is then added to the LAM.
   b) B: If $\sigma_A \geq n_B$ set $\sigma_A = \sigma_A - n_B$ and $b_1 = 1$, else $b_1 = 0$ compute $h_B = H(LAM, (n_B, y_B))$ and $\sigma_{AB} = (\sigma_A + h_B)^{x_B}(mod\ n_B)$. σ_{AB} is then added to the LAM instead of σ_A.
   c) C: If $\sigma_{AB} \geq n_C$ set $\sigma_{AB} = \sigma_{AB} - n_C$ and $b_2 = 1$, else $b_2 = 0$ compute $h_C = H(LAM, (n_C, y_C))$ and $\sigma_{ABC} = (\sigma_{AB} + h_C)^{x_C}(mod\ n_C)$. σ_{ABC} is then added to the LAM instead of σ_{AB}.
   d) D: the computes $h_C = H(LAM, (n_C, y_C))$,
   $\sigma'_{AB} = \sigma_{ABC}^{y_C} - h_C(mod\ n_C)$,
   $\sigma_{AB} = \sigma'_{AB} + b_2 n_C$, $h_B = H(LAM, (n_B, y_B))$,
   $\sigma'_A = \sigma_{AB}^{y_B} - h_B(mod\ n_B)$,
   $\sigma_A = \sigma'_A + b_1 n_B$, $h_A = H(LAM, (n_A, y_A))$,
   and finally check if $\sigma_A^{y_A}(mod\ n_A)$ equals h_A
   e) Signature verification fails if a LAM does not travel the same route as it should.

ALARM Extension	Sybil Attack	Location-fraud Attack
Group Signatures with Self-Distinction	Prevent	Fail
One-Time Certificates	Prevent	Fail
Sequential Aggregate Signatures (SAS)	Prevent	Prevent
Secure Hardware	Prevent	Prevent

TABLE III
SECURITY OF EXTENSIONS AGAINST ACTIVE INSIDER ATTACKS

5) Secure Hardware : Recent advances in group signature research have yielded efficient schemes with constant-size signatures and public keys. There have also been proposals to implement group signatures using tamper-resistant hardware. For example, [12] shows how to implement group signature functionality on smartcards. If a similar implementation is coupled with a tamper-resistant GPS device, all insider attacks in ALARM can be virtually eliminated. Specifically, an insider would be unable to lie about its current location or to mount a Sybil attack. With tamper-resistant hardware, group signature schemes with self-distinction are no longer needed, since a node would be prevented from generating more than one signed LAM within a given time-slot.

VI. SECURITY ANALYSIS

Recall that our adversary model of Section IV does not consider physical-layer jamming and denial-of-service (DoS) attacks on message transmission.

A. Outsider Attacks

A passive outsider eavesdropping on all LAMs can, at most, obtain exactly the same information available to any legitimate MANET node (i.e., the current topology snapshot). This would only happen if keys used to encrypt all communication in the MANET are leaked. Thus a passive outsider is at most as powerful as a passive insider and thus protection against it is guaranteed as a side effect of thwarting passive insider attacks.

Since group signatures attached to each LAM are untraceable and unlinkable, the only way to track nodes is by guessing possible trajectories. However, as discussed in Section III, our MOBILITY assumption involves a minimum number of nodes (k out of n) moving within each time-slot. Thus, tracking movements of a given node translates into k-anonymity [45], i.e., the problem of identifying one out of k possible nodes. However, we note that, if LAM-s are encrypted using a group-wide key, topology information would become completely "invisible" to eavedroppers. An outsider would only be able to determine node presence at certain locations. Also, physical-layer techniques, such as CDMA, can be used to hide transmission from unintended receivers.

Active outsider attacks are addressed in ALARM through the use of LAM time-stamps and group signatures. An active outsider cannot inject new LAMs or modify any existing LAMs, since it has no group signature capability. Replays are trivially prevented by LAM time-stamps.

B. Passive Insider Attacks

A passive insider (legitimate MANET node) can, by design, obtain all LAMs and determine their authenticity by verifying corresponding group signatures. But, also by design, it can neither identify nor link nodes that generated these LAMs, since group signatures are untraceable. A passive insider with other means of collecting mobility information, e.g., by visual monitoring, can determine that a certain node remains stationary. This might happen if, in two consecutive time-slots, the insider physically (i.e, visually) observes lack of mobility and also receives two LAMs referring to the same location. Clearly, there is no protection against such attacks, since they involve adversary's physical presence.

A passive insider can attempt to track a node's movements by using viable trajectory information [24]. This attack is possible if the adversary knows the MANET topology as well as approximate node speed and trajectory and direction of movement of a given node. If nodes do not move along straight lines and their direction is randomized, or, if a group of nodes move closely together or intersect paths, such attacks fail or degenerate to k-anonymity. We use simulations to evaluate the loss of privacy due to such attacks; see Section VII for details.

C. Active Insider Attacks

The basic incarnation of ALARM is not secure against active insider attacks in *real time*. Section V-B presented extensions that mitigate such attacks (see Table III):

- As discussed in Section V-B.2, group signature schemes with self-distinction can be used to prevent Sybil attacks, albeit, at extra computation and communication cost.
- If each node has a secure hardware component (Section V-B.5) housing group signature generation, Sybil attacks can be prevented without requiring self-distinction from the underlying group signature scheme. If secure hardware also encompasses a GPS receiver, location-fraud is easily prevented. However, ubiquitous secure hardware is clearly an expensive option.
- Through the use of one-time certificates (Section V-B.3) ALARM can prevent Sybil attacks, but not location-fraud.
- The use of sequential aggregate signatures (Section V-B.4) can help prevent Sybil and location-fraud attacks.

In addition, Sybil attacks can be easily detected *off-line*, if the optional forensics feature is enabled and operational logs are later off-loaded to GM for analysis.

VII. Performance Analysis

We now analyze ALARM's routing overhead and compare its scalability to other link-state routing protocols. We then consider the delay caused by periodic flooding of LAMs. Finally, we discuss the effect of node mobility on route availability. The goal of this section is to demonstrate that security and privacy features of ALARM do not introduce high overhead that hurts scalability and performance.

A. Control Traffic Overhead

In any MANET link-state routing protocol, the number of hops between any random source-destination pair increases when neighborhood size decreases, thus influencing control traffic overhead [9]. We examine this overhead in ALARM by analyzing the maximum manageable neighborhood size using the model proposed in [9]. We compare ALARM's neighborhood size to that of OSPF [37] and OLSR [26]. We show that, in a 2-D network model without fading, maximum neighborhood size is limited to 16 nodes in the basic OSPF protocol (42 for a modified version), whereas it is 45 in the basic un-optimized ALARM and 62 in OLSR. This shows that the overhead of the basic ALARM protocol is close to that of OLSR, which is honed to minimize control traffic overhead and does not provide any privacy features. ALARM can be optimized (similar to OLSR) by restricting the number of nodes that forward LAMs. ALARM's lower overhead is because it omits OLSR neighbor sensing phase, due to the use of locations for addressing. If further optimized, ALARM would outperform OLSR.

Neighbor and Network Topology Models

The model in [9] assumes a network with N transmitters distributed according to a Poisson process with a rate parameter (λ). Density of transmitters per time slot and per square area unit is $\lambda = fN/A$, where f is packet transmission rate per slot, per node, and A is the area. A node is considered a neighbor of another node if probability of receiving HELLO messages from each other is greater than a certain threshold p_0 (typically $p_0 = 1/3$). A packet can be decoded if its signal-to-noise ratio exceeds a given threshold K (typically $K = 10$).

A node is a neighbor of another node if the distance between them (r) is such that the probability of receiving a certain signal intensity is greater than the threshold p_0. Specifically this probability is defined as: $P(W < r^{-\alpha}/K) > p_0$, where $r < r(\lambda)$. $r(\lambda)$ is the critical radius such that $\int_0^{r(\lambda)} w(x)dx = p_0$. If W is the signal intensity received by node X at a random slot then W is a random variable with $w(x)$ as its density function [9]. By integration, $r(\lambda) = \lambda^{1/2} r(1)$ and the surface covered by radius $r(\lambda)$ is the neighborhood area $\sigma(\lambda) = \sigma(1)/\lambda$. The constant $\sigma(1)$ for different values of α and λ can be computed as in [9]. Specifically, for $\alpha = 2.5$ and $\lambda = 1$,$P(W < x)$ reaches $p_0 = 1/3$ close to $x = x_0 = 20$. Therefore, $r(1) = (x_0 K)^{-1/\alpha} \approx 0.12$ and $\sigma(1) \approx 0.045$.

This model assumes that the total number of nodes is $N = \nu A$ where ν is node density per unit area. If λ represents network traffic density, the average number of neighbors per node is [9]:

$$M = \sigma(\lambda)\nu = \sigma(1)\nu/\lambda \qquad (1)$$

Link-State Overhead

Our goal is to derive traffic density caused by ALARM control packets. There are two sources of control traffic in link-state protocols: (1) neighborhood sensing (e.g., HELLO messages), and (2) topology discovery via link-state announcements (LAMs in ALARM).

Parameter	Value
Simulation Area	$1000m$ X $1000m$
Simulation Time	100000 sec
Simulation Repetition	1000 runs
Inter-LAM interval	Varied from 5sec to 30sec
Node Speed	Varied from 5m/sec to 100m/sec
Number of Nodes	100
Mobility Models	– Random Walk and Random Waypoint Mobility [11] – Reference Point Group Mobility (RPGM) [22] (5 groups with 20 nodes per group) – Time-variant User Mobility (TVUM) [27] (4 communities)

TABLE IV
SIMULATION PARAMETERS

Neighborhood sensing is the same for most link-state protocols: each node periodically broadcasts a HELLO containing the list of neighbors heard by it. By comparing their lists nodes determine the set of neighbors for which they have symmetric links. This is not the case in ALARM: because each node is aware of its own location, mere knowledge of another's location is sufficient to determine whether that node is a neighbor.

Assume h is the neighborhood information refresh rate and let B be the maximum number of node identifiers within a slot. We assume that each identifier (a group signature and a location) is about 250 bits; see LAM format in Figure 2. For a MANET with a capacity of 100Mbps, there are 1000 slots per second, assuming a slot can carry 100Kb, i.e., 1 msec. Thus $B = 100Kb/250b = 400$. If the neighbor list exceeds B, several HELLOs are generated per update period. A node must generate $\lceil \frac{M}{B} \rceil$ HELLOs per period. This leads to traffic density of $h\nu\lceil \frac{M}{B} \rceil$. Omitting fractional parts, we have [9]:

$$\lambda = h\nu\frac{M}{B} \tag{2}$$

If HELLOs are the only source of control traffic, since $M = \sigma(1)\nu/\lambda$, we get:

$$\frac{\sigma(1)}{M} = h\frac{M}{B} \tag{3}$$

This is only an upper bound because the network may be smaller than $\sigma(1)$. In OLSR, a node generates HELLOs every 2 seconds, i.e., $h = 1/2000$. Therefore, the maximum manageable neighbor size with only the HELLO control traffic is: $\sqrt{B\sigma(1)/h} \approx 190$. The basic ALARM protocol does not have HELLO messages; so, the previous upper bound does not apply.

We now express λ only in terms of ALARM protocol overhead (similar derivation for OLSR and OSPF can be found in the Appendix). We assume that, in all protocols, the topology discovery and control (TC) update period are the same. For the standardized OLSR [26], TC rate per node is $\tau = 1/5000$ (i.e., every 5 seconds, which we also use as a LAM flooding period in ALARM and also in OSPF).

ALARM Model: A node periodically: (1) transmits its LAMs with rate h, and (2) re-transmits received LAMs with some delay (one copy to all M neighbors). Thus, ALARM traffic density satisfies[4]:

$$\lambda = \tau\nu N\lceil\frac{M}{B}\rceil \tag{4}$$

From equations (1) and (4), we get:

$$\sigma(1)\frac{\nu}{M} = \tau\nu N\lceil\frac{M}{B}\rceil \tag{5}$$

Dropping the ceiling results in:

$$M = \sqrt{\frac{\sigma(1)B}{\tau N}} \tag{6}$$

This represents the relationship between network size N and average neighborhood size M. The minimum neighborhood size M is 1, below which the network no longer has any significant connected components.

The maximum size of the network N is obtained when $M = 1$, then: $N_{max} = \frac{\sigma(1)B}{\tau} \approx 90000$ for $B = 400$ with $\sigma(1) = 0.045$ and $\tau = 1/5000$. For the special case of $N = M$ (i.e., a single-hop network), we get: $M = \sqrt[3]{\frac{\sigma(1)B}{\tau}}$ which gives $N_{ALARM} = 45$ for $B = 400$.

To summarize, the basic ALARM incarnation can achieve 0.73 (45/62) of maximum neighborhood size, compared to OLSR. A modified OSPF (to improve performance) under assumptions given above can only achieve 0.677 (42/62) of maximum neighborhood size, compared to OLSR. Because routing overhead is inversely proportional to neighborhood size, ALARM would incur slightly higher overhead than OLSR, which is the price for its simplicity and its privacy features. We note that a simple modification to ALARM that makes nodes selectively forward LAMs (similar to MPR selection in OLSR) would result in significantly lower overhead.

B. Time to Construct Network Topology

Recall that LAMs are periodically flooded to facilitate timely update of topology information. This requires that cumulative LAM propagation delay (T_{prop}) coupled with group signatures verification delay (T_{ver}) be smaller than LAM flooding period. We now assess the feasibility of this constraint and analyze the relationship between number of nodes and area size for which it can be satisfied. Time to construct topology (T_{top}) is:

$$T_{top} = T_{prop} + T_{ver} \tag{7}$$

where $T_{ver} = N{\cdot}T^{ver}_{gsig}$ is time to verify all N group signatures. Time to verify a single group signature T^{ver}_{gsig} depends on the specific group signature scheme.

For example, using the group signature scheme of Table I, a node can verify about 60 group signatures in less than a second. For small- to medium-size networks (of 10-s or 100-s of nodes) such performance is reasonable. Faster group signature schemes exist, however, they feature longer signature

[4] We neglect the term of sending a node's own LAM with rate h because it is one message of constant size independent of the number of neighbors. Taking it into account would only slightly affect neighborhood size.

and key sizes. T_{prop} is the total time to transmit *all* (N^2) LAMs to *all* nodes:

$$T_{prop} = \frac{N^2 \cdot LAM_{size}}{MaxNumTx \cdot BW} \tag{8}$$

where LAM_{size} is LAM size, BW is the bandwidth of the underlying wireless channel (e.g., 10Mbps) and $MaxNumTx$ is maximum number of simultaneous transmissions. We now estimate the latter using a medium access protocol based on the DCF function (as in the IEEE 802.11 MAC). The analysis is based on the model in [53]. In general, for node j to correctly receive a signal from node i, the signal to noise ratio has to exceed a certain threshold (capture threshold, z_0):

$$SIR = \frac{P_i \gamma_{ij}}{N_0 + \sum_{k \neq i} P_k \gamma_{kj}} > z_0 \tag{9}$$

where P_i is transmission power of node i, $\gamma_{ij} \propto d^{-\alpha}$ is channel gain between nodes i and j (with d being distance between i and j and power loss exponent α assumes values between 2 and 4), N_0 is background noise power and z_0 ranges from 1 (perfect capture) to ∞ (no capture). We assume that N_0 is small and the transmit power is constant. In the general case with multiple interferes, the number of simultaneous senders is maximized when they are located as close as possible. In this setting, each transmission does not interfere with the rest of the senders. [53] shows such an arrangement and only considers the first-tier (one hop away) interferes, since their interference is much stronger than that of second-tier (two hops away). The worst-case interference with respect to communication from i to j occurs when distances from j to the six interferes are $(D-d)$, $(D-d)$, $(D-d/2)$, D, $(D+d/2)$, $(D+d)$, respectively. Thus, SIR becomes [53]:

$$SIR = \frac{d^{-\alpha}}{2(D-d)^{-\alpha} + (D-\frac{d}{2})^{-\alpha} + D^{-\alpha} + (D+\frac{d}{2})^{-\alpha} + (D+d)} \tag{10}$$

where d and D denote sender-to-receiver (i-j) and interferer-to-receiver (k-j) distances, respectively. Let D_{min} be minimum distance satisfying SIR. Maximum number of concurrent transmissions in area L^2 then becomes:

$$MaxNumTx = \frac{L}{D_{min}} \cdot \frac{L}{\frac{\sqrt{3}}{2} D_{min}} = \frac{2L^2}{\sqrt{3} D_{min}^2} \tag{11}$$

To simplify, we approximate the distance between node j and all interferes as D. In this case, from the SIR equation (equation 10), we have:

$$D_{min} = \sqrt[\alpha]{6 z_0 d} \tag{12}$$

using this D_{min} to calculate the $MaxNumTx$ and substituting with typical values for the attenuation exponent ($\alpha = 2$) and the capture threshold ($z_0 = 10$), the propagation time (T_{prop} in equation 8) becomes:

$$T_{prop} = \frac{60d \cdot LAM_{size} \cdot N^2 \sqrt{3}}{2BW \cdot L^2} \tag{13}$$

Assuming that uniform node distribution (according to a Poisson process with λ nodes per unit area) average distance between nodes becomes $d = \frac{128}{45\pi}\sqrt{\frac{N}{\lambda\pi}}$ [47]. T_{prop} can be expressed as:

$$T_{prop} = \frac{N^{5/2} \cdot LAM_{size} \cdot 256}{BW \cdot L^2 \pi^{3/2} \sqrt{3\lambda}} \tag{14}$$

We assume that time available for cumulative LAMs propagation is a fraction (f_{prd}) of the LAM flooding period (LAM_{prd}). Then, the relationship between maximum number of nodes (N) and area size (L^2) becomes:

$$N = L^{4/5} \sqrt{\frac{LAM_{prd} \cdot f_{prd} \cdot bw \cdot \pi^{3/2} \sqrt{3\lambda}}{LAM_{size} \cdot 256}} \tag{15}$$

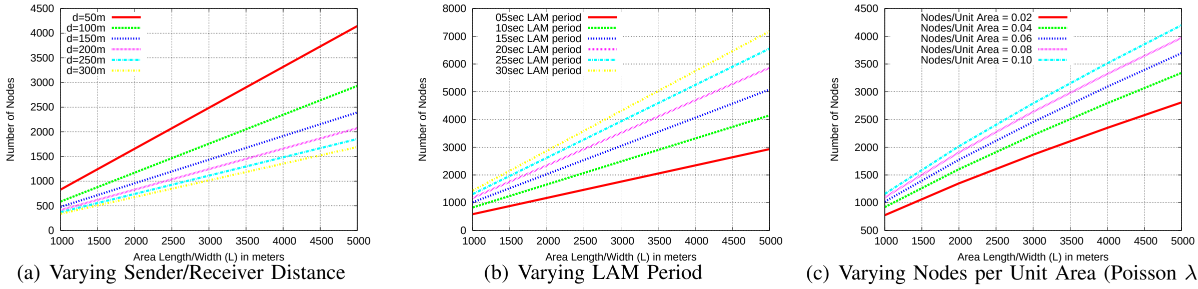
Figure 7 shows maximum number of nodes that satisfies different LAM flooding periods for various area sizes. Network parameters used are: $LAM_{size} = 350$ bytes, BW=10Mbps, $f_{prd} = 0.1$, $LAM_{prd} = 5sec$ (in Figures 7(a) and 7(c)). Graphs in Figures 7(a) and 7(b) show maximum number of nodes satisfying Equation 13 for $T_{prop} = LAM_{prd} \cdot f_{prd}$ with $f_{prd} = 0.1$. Graphs in the Figure 7(c) are based on Equation 15. Number of nodes (y-axis) is plotted for various area Length/Width (x-axis) for different values of Poisson parameter for node density per unit area (λ, varied between 0.02 and 0.1).

C. Effect of Node Mobility on Route Availability

Node mobility affects availability of wireless links, which, in turn, influences routes over these links. An important question is: how long do routes persist under different mobility models? An exhaustive study [20] of effects of mobility on MANET routing protocols has shown that, in a MANET of 40 nodes in a 1000m-X-1000m area, moving according to the reference point group mobility (RPGM) model (consisting of one big group), average lifetime of a link is around 900sec for speeds less than 30m/sec. For a setting with 4 groups (of 10 nodes each), link lifetime drops significantly, but exceeds 240sec for speeds up to 50m/sec. Link lifetime is around 60sec under the Freeway and Manhattan mobility models [20]. The same study analyzed path lifetime and showed that similar durations are observed for path availability (i.e., 100-s of seconds for RPGM and 10-s of seconds for RWM, Manhattan and Freeway Mobility). [20] also reports that the path availability[5] for RPGM (single and multiple groups), RWM, Freeway and Manhattan was found to be 100%, 92%, 97%, 99% and 95%, respectively.

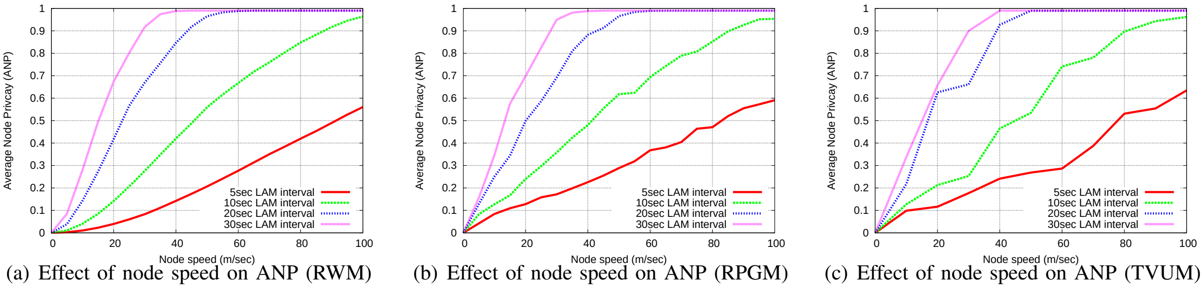
Recall that ALARM periodically (on the order of seconds) floods topology updates (LAMs). Between topology updates, routes would remain stable and available based on results from [20] showing that routes remain available for several minutes in RPGM, and for around one minute under other models (RWM or VANET models, e.g., Manhattan and Freeway). If traffic patterns are bursty and data sessions are short-lived (lasting on the order of seconds) then mobility would not affect ALARM operation.

[5] fraction of time for which a path between any two nodes was available



(a) Varying Sender/Receiver Distance (b) Varying LAM Period (c) Varying Nodes per Unit Area (Poisson λ)

Fig. 7. Maximum number of nodes satisfying different LAM flooding periods for various area sizes. ($LAM_{size} = 350$ bytes, BW=10Mbps, $f_{prd} = 0.1$, $LAM_{prd} = 5sec$ if not varied)



(a) Effect of node speed on ANP (RWM) (b) Effect of node speed on ANP (RPGM) (c) Effect of node speed on ANP (TVUM)

Fig. 8. Effect of node speed on ANP: Random Walk (RWM), Reference Point Group Mobility (RPGM) and Time-Varying User Mobility (TVUM)

VIII. Simulation Results

We first introduce a new privacy metric to measure ALARM's effectiveness in combating node tracking. We then simulate ALARM with several mobility models to show its resistance to insider attacks.

A. Privacy Metric

Recall that ALARM provides node privacy by preventing tracking by both insider and outsider adversaries. To illustrate its effectiveness, we define a new privacy metric called *Average Node Privacy* **(ANP)**. Basically, ANP is a cumulative version of k-anonymity [45] over time and averaged over the entire network. Given the successive topology snapshots during the operation of the network (T snapshots), ANP represents the average fraction of nodes that a given node can be equally likely mapped to. This is similar to the k-anonymity concept where a node's privacy is preserved by making it indistinguishable from a set of k other nodes. ANP is computed as follows:

$$ANP = \sum_{t=1}^{t=T} \sum_{i=1}^{i=K} \frac{K - K_i^t}{T \cdot K^2} \tag{16}$$

where K is the total number of nodes in the MANET. T is the number of snapshots of the network over time. K_i^t is the number of nodes from snapshot t to which node i can not be mapped to, assuming that the adversary knows where i was at snapshot $t-1$. The $T \cdot K^2$ term in the denominator normalizes the metric so that it has a maximum value of 1. K_i^t depends on the underlying mobility pattern (i.e. direction and speed of movement), time between successive topology snapshots (i.e. time between two LAM-s) and size of the area within which the nodes move. Between two successive snapshots of the topology, K_i^t will include nodes outside a circle defined by r (r=*node speed* $\cdot$ *LAM period*) as its radius and the location of node i in the first snapshot as the center.

ANP is highest when the best mapping an adversary can construct is one where a node from snapshot $t-1$ is equally likely to be mapped to *any* of the K nodes in snapshot t. In this case, r is the longest possible traveling distance in the area of movement (e.g., the diagonal in the case of a square) and ANP will be 1. When each node can only be mapped to one other node, then nodes become completely traceable and node privacy is violated. In this case, an adversary can look at two subsequent snapshots of the network topology and deterministically map nodes from the first snapshots to nodes in the second snapshot.

To achieve an ANP of 1 for nodes moving inside an area (LxL), the time between snapshots (LAM period) has to be long enough for the slowest node to travel a distance equal to ($\sqrt{2 * L^2} \approx 1.4 * L$). In this case, a node at a location L_1 in the first snapshot is equally likely to be at any other location L_2 in the second snapshot. An adversary that compares these two snapshots and aims to track a certain node's movement will at most be able to determine the mapping between the first snapshot and the second correctly with probability ($1/K$) (because of random guessing). If the adversary wants to track more nodes the probability of success decreases rapidly. If the adversary wants to track all (K) nodes, the probability of success will be $\frac{1}{K!}$. The probability of tracking (i) out of the (K) nodes is: $\frac{(K-i)!}{K!}$.

B. Effects of Node Mobility on Privacy

We simulated a MANET with nodes moving in a square area with $1000m$ side length. Simulations were performed using the SimPY [43] discrete-event simulation framework.

We used four mobility models. Two are entity-based: (1) random walk and (2) random waypoint [11], and the other two are group-based: (3) reference point group mobility model (RPGM) [22] and (4) time-variant user mobility model (TVUM) [27]. TVUM was developed based on behavior found in wireless network traces obtained from university networks and is the closest approximation of real-life mobility patterns [27]. We summarize simulation parameters in Table IV.

Random Walk Mobility (RWM): In this model, a node chooses a random destination within the area and moves towards it. Once a node reaches its destination, it randomly chooses a new one and starts moving towards it. Random waypoint and RWM have been criticized to be unrealistic [20], however, we use RWM as a base-case to show that completely random movements might not yield the highest level of privacy. Also, RWM could be a reasonable approximation of mobility in military (e.g., battlefield) settings, for which no traces are available, for obvious reasons. The results for RWM are shown in Figure 8(a). Very similar results were also obtained for the random waypoint model [11]. Figure 8(a) shows that, when the inter-LAM interval is 5 seconds, each node can be mapped to less than 10% of other nodes (i.e., ANP=0.1) at speeds below 32 m/sec (about 100 Km/h). If node speed exceeds that, privacy increases. We note that this ANP of 0.1 means that each node cannot be distinguished from 10 other nodes in this setting. Increasing the inter-LAM interval to 10 sec results in significant gain in privacy – ANP of 0.3. This goes up to 0.7 for a 20 sec inter-LAM interval.

RPGM: Figure 8(b) shows simulation results for the RPGM model. In it, nodes are pre-divided into equally sized groups. Each group has a logical center which defines movement patterns for the entire group, i.e., speed, acceleration and direction. Each group member is placed randomly in the vicinity of its reference point, relative to the group center. This ensures that relative positions of nodes inside the group change over time.

When nodes move according to the RPGM model with low speeds and with small inter-LAM intervals, ANP is higher than when all nodes move independently. Figure 8(a) shows the result of simulating 100 nodes divided into 5 equal-sized groups (20 nodes each). ANP in RPGM is 0.4 at 32 m/sec (instead of 0.3 in RWM). This is because the mobility pattern guarantees that at least nodes within the same group remain in each other's vicinity. The difference between RPGM and RWM for larger inter-LAM intervals (20 and 30 sec) is small, (about 0.05), especially, at high speeds, because the area of possible coverage is large and includes most of the nodes, regardless of the mobility model.

Figure 9 shows the effect of the number of groups on ANP under the RPGM model. It is easy to see that, due to the construction of the model, smaller number of groups implies better privacy. If we double the number of groups (assuming constant network size), the number of nodes in each group is halved and a linear drop in ANP occurs. This is because nodes in the same group moving more-or-less together are indistinguishable. We claim that RPGM may be common in mission-critical settings and its relatively high privacy illustrates ALARM's suitability in such settings.

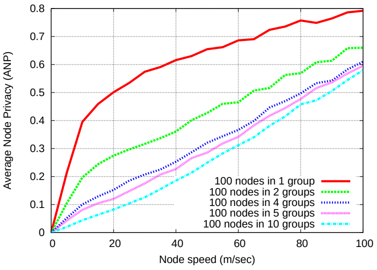


Fig. 9. Effect of number of groups on ANP (RPGM)

TVUM: this model was motivated by two observations typical in traces of mobile wireless networks: *skewed location visiting preference* and *periodic re-appearance*. The distinctive feature of TVUM is in defining often-visited communities (areas) so as to capture skewed location visiting preferences and the use of time periods with different mobility parameters to create periodic re-appearance. Each node is randomly assigned to a community. TVUM defines two time periods: *normal movement period (NMP)* and *concentration movement period (CMP)*. Within a CMP, a node visits its community with high probability. A node has two different modes of movement: *local epoch* and *roaming epoch*. In a local epoch, node's mobility is confined within its community. In a roaming epoch, a node is free to move within the whole simulation area. A node switches between epochs based on a two-state Markov chain model.

We use the following values in our simulations: 4 communities, defined as an area covered by a circle with 100m radius and center selected at random. NMP is 200sec and CMP is 400sec. The probability of switching from local to roaming epoch is $p_r = 0.4$, and, from roaming to local – $p_l = 0.7$. Local epoch is set to 200sec and roaming – 100sec.

Figure 8(c) shows the simulation results. ANP is, on average, lower than that under RPGM mainly because each node moves independently from others. However, ANP is higher (by about $0.05 - 0.1$) than in RWM. Nodes belonging to the same community are more likely to select destinations that are closer and are more likely to intersect.

IX. Related Work

Secure MANET routing has been extensively studied in both security and networking research communities. A comprehensive survey of this work can be found in [23]. All secure MANET routing protocols focus on securing route discovery, route maintenance and defending against modification and fabrication of routing information. Privacy – especially, tracking-resistance – is not one of the goals of these protocols.

A more relevant body of research focused on proactive anonymous MANET routing protocols, such as SPM [40]. SPM is a modified link-state protocol that requires nodes joining (and leaving) the MANET to report such events to "super" nodes. Super nodes collect and distribute topology information and also handle communication between different "local" MANETS. SPM assumes that nodes periodically change their pseudonyms and that they communicate based on instantaneous pseudonyms. SPM is thus identity-based and requires nodes to be able to retrieve each other's public keys.

Another related research direction tackles anonymous on-demand MANET routing, e.g., SPAAR [13], AO2P [49], ASR [55], MASK [54], ANODR [30], D-ANODR [52], ARM [42], ASRP [15] and ODAR [46]. A brief survey comparing ANODR, ASR and discussing general anonymity and security issues in MANET routing protocols can be found in [31]. Of the anonymous on-demand protocols, SPAAR [13] and AO2P [49] require on-line location servers. ASR [55] and ARM [42] assume that each authorized source-destination pair pre-shares a unique symmetric key. AnonDSR [44], ASRP [15], EARP [51] and ARMR [17] assume that each source-destination pair shares some secret information, which could be the public key of the destination or a symmetric key. ANODR [30] assumes that the source shares some secret with the destination for the construction of a trapdoor, for example the destination's TESLA [39] secret key. SDAR [8] assumes that the source knows the public key of the destination, obtained from a certification authority (CA), and ODAR [46] requires an on-line public key distribution server. MASK [54] and D-ANODR [52] contain the final destination in the clear in each RREQ message. AMRSS [14] and ARMR [17] utilizes multiple paths for routing. AMRSS [14] assumes that the entire network shares a pair of public-private keys and that the destination ID will be encrypted using such a key. AMRSS also includes the entire path encrypted under the network key in each data message. In addition all aforementioned on-demand anonymous routing protocols assume that nodes know the long term identities of the other nodes they will communicate with, i.e., the communication paradigm is identity centric.

Table I in the Appendix compares these schemes with ALARM in more detail. The fundamental difference between ALARM and above protocols is that ALARM is geared for location-centric communication and does not assume any knowledge or existence of persistent node addresses or ID-s. ALARM also does not require any online trusted parties or any pre-shared secret keys among MANET nodes.

PRISM [19] is another recent on-demand anonymous MANET routing protocol. Despite their common use of group signatures, ALARM differs markedly from PRISM. Since ALARM is a link-state protocol, before attempting to communicate, nodes know the entire MANET topology; therefore, precise destination addressing is used. In contrast, in PRISM, a node has no *a priori* topology knowledge; it has to first determine its geographical area of interest and probe it with a route-request message (RREQ). Global knowledge of current topology in ALARM makes it easier to contend with active insider attacks.

X. Conclusions

This paper presented the ALARM protocol which supports anonymous location-based routing in suspicious MANETS. ALARM relies on group signatures to construct one-time pseudonyms used to identify nodes at their present locations. The protocol works with any group signature scheme and any location-based forwarding mechanism. We evaluated the overhead and scalability of ALARM and showed that it performs close to other protocols (e.g., OLSR) optimized to reduce control traffic. We also evaluated ALARM's tracking-resistance with different mobility models via simulations. ALARM is a viable and practical approach to routing in mission-critical location-based MANETS where security and privacy requirements must be reconciled and resistance to both outsider and insider attacks is needed.

References

[1] EU Cooperative Vehicle-Infrastructure System Project. http://www.cvisproject.org/.

[2] OpenSSL: The Open Source toolkit for SSL/TLS. http://www.openssl.org/.

[3] Ospf with digital signatures (rfc 2154). http://www.ietf.org/rfc/rfc2154.txt.

[4] Giuseppe Ateniese and Gene Tsudik. Some open issues and new directions in group signatures. In *FC '99: Proceedings of the Third International Conference on Financial Cryptography*, pages 196–211, London, UK, 1999. Springer-Verlag.

[5] A border gateway protocol 4 (bgp-4) (rfc 1771). http://www.ietf.org/rfc/rfc1771.txt.

[6] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *In proceedings of CRYPTO 04, LNCS series*, pages 41–55. Springer-Verlag, 2004.

[7] Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In *Proceedings of CCS 2004*, pages 168–177. ACM Press, 2004.

[8] A. Boukerche and K. El-Khatib et al. An efficient secure distributed anonymous routing protocol for mobile and wireless ad hoc networks. *Elsevier Computer Communications*, 2005.

[9] E. Bacelli C. Adjih and P. Jacquet. Link state routing in wireless ad-hoc networks. *MILCOM 2003*.

[10] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy. Efficient and Robust Pseudonymous Authentication in VANET. In *Proceedings of the ACM International Workshop on Vehicular Ad hoc Networks (VANET)*, pages 19–28, September 2007.

[11] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2:483–502, 2002.

[12] Sébastien Canard and Marc Girault. Implementing group signature schemes with smart cards. In *CARDIS'02: Proceedings of the 5th conference on Smart Card Research and Advanced Application Conference*, pages 1–1, Berkeley, CA, USA, 2002. USENIX Association.

[13] S. Carter and A. Yasinsac. Secure position aided ad hoc routing. *Proceedings of the IASTED International Conference on Communications and Computer Networks (CCN02)*, pages 329–334, 2002.

[14] Siguang Chen and Meng Wu. Anonymous multipath routing protocol based on secret sharing in mobile ad hoc networks. In *Measuring Technology and Mechatronics Automation (ICMTMA), 2010 International Conference on*, volume 1, pages 582–585, 13-14 2010.

[15] Y. Cheng and D. Agrawal. Distributed anonymous secure routing protocol in wireless mobile ad hoc networks. *OPNETWORK*, 2005.

[16] W. Diffie and M. Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644–654, Nov 1976.

[17] Ying Dong, Tat Wing Chim, Victor O. K. Li, S. M. Yiu, and C. K. Hui. Armr: Anonymous routing protocol with multiple routes for communications in mobile ad hoc networks. *Ad Hoc Netw.*, 7(8):1536–1550, 2009.

[18] K. El Defrawy and G. Tsudik. Alarm: Anonymous location-aided routing in suspicious manets. *Network Protocols, 2007. ICNP 2007. IEEE International Conference on*, pages 304–313, Oct. 2007.

[19] Karim El Defrawy and Gene Tsudik. Prism: Privacy-friendly routing in suspicious manets (and vanets). *Network Protocols, 2008. ICNP 2008. IEEE International Conference on*, pages 258–267, Oct. 2008.

[20] Narayanan Sadagopan Fan Bai and Ahmed Helmy. Important: A framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks. In *INFOCOM*, 2003.

[21] Toni Farley, Patrick Mcdaniel, and Kevin Butler. A survey of bgp security issues and solutions. Technical report, ATT Labs - Research, Florham Park, NJ, 2004.

[22] X. Hong, M. Gerla, G. Pei, and C. Chinag. A group mobility model for ad hoc wireless networks. *ACM/IEEE MSWiM*, 1999.

[23] Yih-Chun Hu and Adrian Perrig. A survey of secure wireless ad hoc routing. *IEEE Security and Privacy*, 2(3):28–39, 2004.

[24] Leping Huang, K. Matsuura, H. Yamane, and K. Sezaki. Enhancing wireless location privacy using silent period. *Wireless Communications and Networking Conference, 2005 IEEE*, 2:1187–1192 Vol. 2, March 2005.

[25] A. Ruhil I. Stojmenovic and D. Lobiyal. Voronoi diagram and convex hull based geocasting and routing in wireless networks. *Proceedings of Eighth IEEE International Symposium on Computers and Communication (ISCC 2003)*, 1:51–56, 2003.

[26] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. pages 62–68, 2001.

[27] Wei jen Hsu, T. Spyropoulos, K. Psounis, and A. Helmy. Modeling time-variant user mobility in wireless mobile networks. pages 758–766, May 2007.

[28] Jihye Kim and Gene Tsudik. Srdp: Securing route discovery in dsr. In *Mobiquitous '05*, 2005.

[29] Young-Bae Ko and Nitin H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Wirel. Netw.*, 6(4):307–321, 2000.

[30] Jiejun Kong and Xiaoyan Hong. Anodr: anonymous on demand routing with untraceable routes for mobile ad-hoc networks. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 291–302, New York, NY, USA, 2003. ACM.

[31] E.H.J. Kumari and A. Kannammal. Privacy and security on anonymous routing protocols in manet. In *Computer and Electrical Engineering, 2009. ICCEE '09. Second International Conference on*, volume 2, pages 431–435, 28-30 2009.

[32] J. Kurose and K. Ross. Computer networks: A top down approach featuring the internet. *Pearson Addison Wesley*, 2005.

[33] W. Liao and Y. Tseng at al. Geogrid: A geocasting protocol for mobile ad hoc networks based on grid. *Journal of Internet Technology*, 1(2), 2000.

[34] Anna Lysyanskaya, Silvio Micali, Leonid Reyzin, and Hovav Shacham. Sequential aggregate signatures from trapdoor permutations. In *Advances in Cryptology – EUROCRYPT 2004*, pages 74–90. Springer-Verlag, 2004.

[35] S. L. Murphy and M. R. Badger. Digital signature protection of the ospf routing protocol. In *SNDSS '96: Proceedings of the 1996 Symposium on Network and Distributed System Security (SNDSS '96)*, page 93, Washington, DC, USA, 1996. IEEE Computer Society.

[36] Nokia 6110 navigator. `http://europe.nokia.com/A4344146`.

[37] Ietf rfc5614 - mobile ad hoc network (manet) extension of ospf. `http://www.ietf.org/rfc/rfc5614.txt`.

[38] R. Perlman. Network layer protocols with byzantine robustness, ph.d. dissertation, mit lcs tr-429. `http://www.vendian.org/mncharity/dir3/perlman_thesis/`.

[39] Adrian Perrig, Ran Canetti, J. D. Tygar, and Dawn Song. The tesla broadcast authentication protocol. *RSA CryptoBytes*, 5:2002, 2002.

[40] Jian Ren, Yun Li, and Tongtong Li. Spm: source privacy for mobile ad hoc networks. *EURASIP J. Wirel. Commun. Netw.*, 2010:5–5, 2010.

[41] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[42] Stefaan Seys and Bart Preneel. Arm: anonymous routing protocol for mobile ad hoc networks. *Int. J. Wire. Mob. Comput.*, 3(3):145–155, 2009.

[43] Simpy simulator. `http://simpy.sourceforge.net/`.

[44] Ronggong Song, Larry Korba, and George Yee. Anondsr: efficient anonymous dynamic source routing for mobile ad-hoc networks. In *SASN '05: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, pages 33–42, New York, NY, USA, 2005. ACM.

[45] Latanya Sweeney. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, October 2002.

[46] Denh Sy, Rex Chen, and Lichun Bao. Odar: On-demand anonymous routing in ad hoc networks. *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, pages 267–276, Oct. 2006.

[47] Hideaki Takagi and Leonard Kleinrock. Optimal transmission ranges for.randomly distributed packet radio terminals, 1984.

[48] Gene Tsudik and Shouhuai Xu. A flexible framework for secret handshakes. In *In Proc. of PET06 (a one-page abstract appeared in ACM PODC05*. ACM Press, 2005.

[49] Xiaoxin Wu and B. Bhargava. Ao2p: ad hoc on-demand position-based private routing protocol. *Mobile Computing, IEEE Transactions on*, 4(4):335–348, July-Aug. 2005.

[50] X509 certificate (rfc 2459). `http://www.ietf.org/rfc/rfc2459.txt`.

[51] Hui Li Jianfeng Ma Xiaoqing Li and Weidong Zhang. An efficient anonymous routing protocol for mobile ad hoc networks. In *IAS*, pages 287–290, 2009.

[52] Liu Yang, Markus Jakobsson, and Susanne Wetzel. Discount anonymous on demand routing for mobile ad hoc networks. *Securecomm and Workshops, 2006*, pages 1–10, 28 2006-Sept. 1 2006.

[53] Chansu Yu, Kang G. Shin, and Lubo Song. Link-layer salvaging for making routing progress in mobile ad hoc networks. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 242–254, New York, NY, USA, 2005. ACM.

[54] Yanchao Zhang, Wei Liu, Wenjing Lou, and Yuguang Fang. Mask: anonymous on-demand routing in mobile ad hoc networks. *Wireless Communications, IEEE Transactions on*, 5(9):2376–2385, September 2006.

[55] Bo Zhu, Zhiguo Wan, M.S. Kankanhalli, Feng Bao, and R.H. Deng. Anonymous secure routing in mobile ad-hoc networks. *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 102–108, Nov. 2004.

Karim El Defrawy obtained a Ph.D. in Networked Systems from the Bren School of Information and Computer Science (ICS) at the University of California in Irvine (UCI) in 2010. He holds an M.Sc. in Networked Systems from UCI (2008), an M.Sc. and B.Sc. in Electrical Engineering from Cairo University in Egypt (2006 and 2003). His research interests include: security and privacy in wireless networks, in peer-to-peer networks, mitigating large-scale attacks on the Internet and applied cryptography.

Gene Tsudik is a "Lois and Peter Griffin" Professor of Computer Science at the University of California, Irvine (UCI). He obtained his PhD in Computer Science from USC in 1991 for research on firewalls and Internet access control. Before coming to UCI in 2000, he was a Project Leader at IBM Zurich Research Laboratory (1991-1996) and USC Information Science Institute (1996-2000). Over the years, his research interests included: routing, firewalls, authentication, mobile networks, secure e-commerce, anonymity ad privacy, group communication, digital signatures, key management, mobile ad hoc networks, as well as database privacy and secure storage. He currently serves as Director of Secure Computing and Networking Center (SCONCE) and Vice-Chair of the Computer Science Department. In 2007, he was on sabbatical at the University of Rome as a Fulbright Senior Scholar. Since 2009, he is the Editor-in-Chief of ACM Transactions on Information and Systems Security (TISSEC).