Towards Address Privacy in Mobile Ad Hoc Networks

Yanchao Zhang
Dept. of Electrical and Computer Engineering
New Jersey Institute of Technology
yczhang@njit.edu

Kui Ren
Dept. of Electrical and Computer Engineering
Illinois Institute of Technology
kren@ece.iit.edu

ABSTRACT

Security concerns are an impediment to deploying mobile ad hoc networks (MANETs) in hostile environments. This paper proposes and investigates solutions to a new security requirement called *address privacy* to prevent attackers from ascertaining network addresses of MANET principals. Lack of address privacy is devastating to critical MANET operations. For example, if knowing the network address of a target principal, attackers can easily locate the target by passively monitoring the open wireless channel and then launch a pinpoint attack. We present *Swarms*, the first solution to address privacy in MANETs. Swarms eliminates the conventionally explicit one-on-one mappings between MANET principals and network addresses and allows any two principals to communicate while blind to each other's address. We quantitatively measure the address privacy offered by Swarms via an entropy-based information-theoretic metric.

Categories and Subject Descriptors

C.2.0 [**Computer Communication Networks**]: General—*Security and protection*; C.2.1 [**Computer Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Design, Security, Performance

Keywords

Mobile ad hoc networks, security, privacy, routing

1. INTRODUCTION

A mobile ad hoc network (MANET) consists of mobile nodes communicating via multi-hop wireless links. Each node is affiliated with a principal (a person or a piece of equipment) that interacts with others over the MANET substrate. Security designs have been considered indispensable for both military and civilian MANETs. Conventional MANET security research focuses on securing node-to-node communications while largely overlooking the role of principals. Partially filling this gap, this paper studies *address privacy* of MANET principals to prevent attackers from ascertaining their network addresses.

Lack of address privacy is highly undesirable in both military and civilian MANETs. Note that the open nature of wireless channels couples the address privacy of MANET principals tightly with their location privacy. In particular, if knowing the address of a target principal, attackers can locate him by overhearing and analyzing radio messages, thus breaching his location privacy. This is unacceptable in both military and civilian settings. For example, attackers in a military MANET can locate and launch pinpoint attacks on VIP principals after obtaining their addresses. Attackers may also profile the movement of principals to infer secret tactical information such as a forthcoming action. Principals in a civilian MANET often have similar requirements for address and location privacy, as many do not want to expose their whereabouts. This situation highlights the necessity of address privacy in MANETs. As far as we know, address privacy in MANETs has not received any attention so far. Instead, it is a long-held implicit assumption that any two communicating principals in a MANET know each other's address. Therefore, if either is compromised, attackers will immediately know the other's address, which is a clear violation of address privacy.

Our contributions are mainly threefold. First, we identify address privacy as a security requirement for MANETs, which is likely to inspire new research ideas. Second, we present a solution to address privacy in MANETs, named *Swarms*. Third, we quantitatively analyze the address privacy offered by Swarm via an information-theoretic metric. Swarms allows any two principals to communicate without knowing each other's address, thus improving the other's address privacy when either is compromised. This is achieved by hiding each principal's address within a set of addresses (called a *swarm*) and routing a packet via a sequence of swarms from the source to the destination.

Swarms is motivated by a common scene in spy movies. For example, suppose that Alice need deliver a message to Bob, but she only knows Bob's pseudonym, say Alex, instead of his address; there is a middleman Tom knowing the address of Bob, but he associates it with Alex. In this scenario, Alice can address the message to Alex and then deliver it to Tom who subsequently forwards the message to Alex (who is actually Bob). The whereabout of Bob can thus be protected from both Alice and Tom. Swarms extends this familiar

Qshine '08 July 28-31, 2008, Hong Kong, China

DOI 10.4108/ICST.QSHINE2008.3839

scene to enable both efficient communications and adequate address privacy via a de Bruijn graph [5]. Built atop the MANET substrate, Swarms can be easily integrated with any underlying routing protocol. This nice feature would greatly enhance the feasibility and applicability of Swarms.

The rest of this paper is organized as follows. Section 2 presents network and threat models. Section 3 details the Swarms protocol, followed by quantitative analysis in Section 4.2. This paper is then concluded in Section 5.

2. PRELIMINARIES

2.1 Network Model

We consider single-authority MANETs, of which typical examples are those deployed in military, counter-terrorist, and law enforcement actions. The network consists of N principals whose IDs compose the set $\mathcal{U} = \{u_i\}_{i=1}^N$. Hereafter we will refer to the principal with ID u_i as principal u_i for convenience. Each u_i is equipped with a mobile node (radio device) that has a layer-2 MAC address and a layer-3 network address denoted by IP_i, which are also referred to as u_i's MAC and network addresses. This means that we have to provide both MAC-address privacy and network-address privacy so that attackers cannot link an interested MAC or network address to a certain principal.

MAC addresses are only used in local communications, and their privacy can be easily achieved. For example, if each node has a unique fixed MAC address, the mapping between a MAC address and the corresponding principal should be kept confidential to the trusted authority (not in the MANET). Another way of dissociating a principal from his nodal MAC address is to let his node use a dynamic MAC address [4]. Both approaches can effectively prevent attackers from obtaining the MAC address of a target principal.

Assuming that MAC-address privacy can be preserved, we only focus on network-address privacy in this paper and may use the term "address privacy" for short. More specifically, we want to keep attackers from knowing the address-ID mappings $\{u_i \leftrightarrow IP_i\}_{i=1}^N$. We will refer to the node with address IP_i as node IP_i henceforth for simplicity. In addition, when we say that u_i sends a message, it should be understood that the message is actually and appears to be sent from node IP_i as viewed by others.

2.2 Threat Model

The goal of the adversary is to break network-address privacy, i.e., obtaining the one-on-one address-ID mappings. In particular, the adversary attempts to ascertain the corresponding address (or principal) of any given principal (or address). It is beyond the scope of this paper to mitigate other important attacks on MANETs such as physical-layer jamming, MAC-layer misbehavior, or routing disruption. The adversary has a number of agents in the target MANET from which to collect information for analyzing address-ID mappings. Some agents are *external attackers* that do not belong to the MANET and only passively eavesdrop on radio transmissions. Other agents are *internal attackers* that are legitimate MANET principals compromised and fully controlled by the adversary. We also assume that compromising a principal amounts to compromising his mobile node, so we will not differentiate compromised principals and nodes hereafter. To design a feasible solution, we follow the conventional assumption that non-compromised nodes are always the majority. The capabilities and strategies of the adversary will be further illustrated when appropriate.

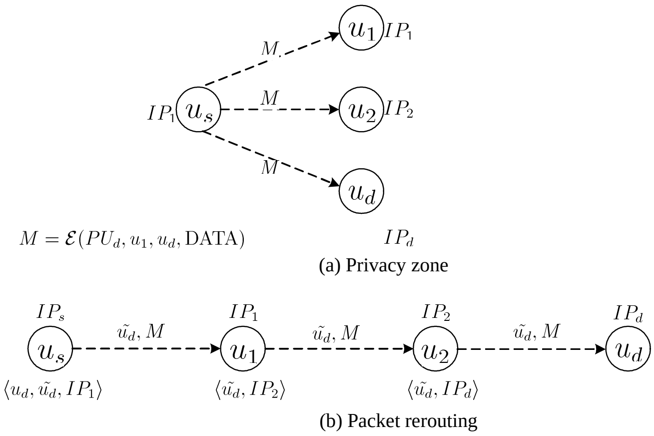

Figure 1: Illustration of the basic idea ($\tilde{u_d}$ is the pseudonym of principal u_d).

3. SWARMS

In this section, we first introduce two basic solutions that motivate the design of Swarms. Then we illustrate Swarms and defer its security analysis to Section 4. In what follows, we assume that each principal u_i has a public/private key pair PU_i/PR_i. Efficient public-key management in MANETs can be realized via many schemes such as [9]. Also let $\mathcal{E}(PU_i, X)$ denote asymmetric encryption of plaintext X using PU_i and $\mathcal{D}(PR_i, Y)$ denote asymmetric decryption of ciphertext Y using PR_i.

3.1 Basic Solutions

Network addresses of MANET principals were seldom considered necessary to be kept secret, and there is often a known unique mapping between a network address and the corresponding principal. Consider an example in which principal u_s need send messages to u_d. He has to know u_d's address IP_d before sending packets via the underlying routing protocol. There are various ways for u_s to acquire IP_d. For example, he can know $u_d \leftrightarrow IP_d$ from the authority before network deployment. This conventional method is very straightforward and efficient, but it may cause severe security issues. For instance, if managing to compromise u_s and thus know $u_d \leftrightarrow IP_d$, attackers can precisely locate u_d. This is extremely dangerous if u_d happens to have a critical role (e.g., a commander). We can improve u_d's address privacy through the following two solutions.

3.1.1 Privacy zone

In this solution, the address set $\{IP_i\}_{i=1}^N$ is partitioned into mutually disjoint subsets, each called a *privacy zone*. The address of each principal is thus unidentifiable within a privacy zone that is known to all his potential correspondents (i.e., principals allowed to directly send messages to him).

Continue with the previous example. Assume that IP_d belongs to privacy zone $\mathsf{ZONE}_d := \{IP_{i_1}, IP_{i_2}, ..., IP_{i_k}, IP_d\}$, where k is a design parameter. This time u_s only knows ZONE_d and $IP_d \in \mathsf{ZONE}_d$, but he cannot single IP_d out.

To send a message DATA to u_d, principal u_s transmits $\mathcal{E}(PU_d, u_s, u_d, \text{DATA})$ individually to each node in ZONE_d, where PU_d is the public key of u_d. Upon receipt of the packet, each node in ZONE_d tries to decrypt it using the private key of its affiliated principal, and only IP_d can succeed in finding u_d in the decryption result which is then passed to u_d. Fig. 1(a) shows an example where $\mathsf{ZONE}_d := \{IP_1, IP_2, IP_d\}$ and each virtual link may correspond to single/multi-hop physical links in the MANET substrate.

Let us briefly analyze the security of this technique. u_s only knows that u_d's address is in ZONE_d, but he cannot associate IP_d with u_d. As a result, even after compromising u_s and knowing ZONE_d, the adversary cannot easily establish the mapping $u_d \leftrightarrow IP_d$. Instead, he must attempt to compromise nodes in ZONE_d one by one to find principal u_d and may have to try on the average $\lceil \frac{|\mathsf{ZONE}_d|-1}{2} \rceil = \lceil \frac{k}{2} \rceil$ times.

3.1.2 Packet rerouting

Motivated by the spy-movie scene in Section 1, we can also improve u_d's address privacy by interposing k *middlemen* $\{u_{i_j}\}_{j=1}^{k}$ between u_s and u_d, through which packets from u_s to u_d will be rerouted. u_d also has a pseudonym $\tilde{u_d}$. Now u_s knows $\langle u_d, \tilde{u}_d, IP_{i_1}\rangle$, u_{i_j} $(1 \le j \le k-1)$ knows $\langle \tilde{u_d}, IP_{j+1}\rangle$, and u_{i_k} knows $\langle \tilde{u_d}, IP_d\rangle$. To send a message DATA to u_d, principal u_s transmits $\langle \tilde{u}_d, \mathcal{E}(PU_d, u_s, u_d, \text{DATA})\rangle$ to node IP_{i_1}. Upon receipt of it, node IP_{i_1} forwards it to node IP_{i_2} based on the embedded $\tilde{u_d}$. This process continues until node IP_d receives and terminates the forwarding of $\langle \tilde{u}_d, \mathcal{E}(PK_d, u_s, u_d, \text{DATA})\rangle$. Fig. 1(b) shows an example with two middlemen, where each virtual link again may correspond to single/multi-hop physical links.

We now briefly analyze the security of this technique. Assume that effective countermeasures (such as [8]) are in place to prevent the adversary from identifying that the forwarding of packet $\langle u_d, \mathcal{E}(PK_d, u_s, u_d, \text{DATA})\rangle$ is terminated at node IP_d. All the middlemen only know that they are forwarding information for principal $\tilde{u_d}$, and even u_{i_k} cannot link IP_d to $\tilde{u_d}$ or u_d. Suppose that u_s is compromised. The adversary only knows the address of the next middleman towards u_d. To locate u_d, the adversary would have to sequentially compromise nodes $IP_{i_1}, IP_{i_2}, ..., IP_{i_k}, IP_d$. To put it differently, before compromising node IP_d, the adversary cannot distinguish u_d's address from the addresses of all the other non-compromised nodes.

3.1.3 Comparison

Now we briefly compare these two techniques. The privacy-zone technique involves $(k+1)$ unicast transmissions via the MANET substrate, so does the packet-rerouting technique. For simplicity, we assume the same communication cost to unicast a packet between any two nodes, so both techniques have the same communication overhead under the chosen parameter k. The end-to-end delay of the privacy-zone technique is, however, k times less than that of the packet-rerouting technique, as packets can be almost simultaneously sent to all the nodes in the privacy zone.

In contrast, the packet-rerouting technique outperforms the privacy-zone technique in protecting u_d's address privacy. In particular, assume that the adversary only compromises u_s. If the packet-rerouting technique is used, the adversary only knows that u_d's address is one of the $(N-1)$ non-compromised addresses; if the privacy-zone technique is used, the adversary knows that u_d's address must be one of the k addresses in ZONE_d.

The Swarms scheme we will illustrate next is built upon the privacy-zone and packet-rerouting techniques to strike a balance between end-to-end delays and address privacy. The essential idea is to hide each principal's address within an address block, called a *Swarm*, and route a packet between any two principals via a sequence of intermediate swarms. In what follows, we will detail the Swarms design. Section 3.2 presents the formation of swarms, followed by the routing process among swarms in Section 3.3. Then we illustrate the complete process of routing a packet from a source principal to a destination principal in Section 3.4. Finally, we discuss how to improve the routing efficiency.

3.2 Swarms Formation

We assume that the MANET is bootstrapped by a trusted authority (TA) not in the resulting network. Prior to network deployment, the TA generates an arbitrary partition $\{\mathcal{U}_a\}_{a=0}^{2^\beta-1}$ of the principals such that

$$\begin{cases} \mathcal{U}_b \cap \mathcal{U}_c = \emptyset, \forall b, c \in \{0, \cdots, 2^\beta - 1\}, b \neq c \\ \bigcup_{a=0}^{2^\beta-1} \mathcal{U}_a = \mathcal{U} = \{u_i\}_{i=1}^{N} \quad , \end{cases}$$

where β is a system parameter determining the maximum number of allowable swarms. $\mathcal{U}_a$ is called the a^{th} swarm with an address block $\mathcal{IP}_a$ comprising the network addresses of the principals in $\mathcal{U}_a$, namely, $\mathcal{IP}_a := \{IP_i\}_{u_i \in \mathcal{U}_a}$. Apparently, we also have

$$\begin{cases} \mathcal{IP}_b \cap \mathcal{IP}_c = \emptyset, \forall b, c \in \{0, \cdots, 2^\beta - 1\}, b \neq c \\ \bigcup_{a=0}^{2^\beta-1} \mathcal{IP}_a = \mathcal{IP} = \{IP_i\}_{i=1}^{N} \quad . \end{cases}$$

It is worth noting that swarms are virtual: principals of the same swarm may be physically apart.

Each principal knows which swarm he belongs to and the corresponding address block, but he is blind to other principals in the same swarm, called *swarm peers*. Consider principal u_d as an example who belongs to $\mathcal{U}_{a_l}$. He only knows his affiliation with $\mathcal{U}_{a_l}$ as well as $\mathcal{IP}_{a_l}$, but without any information about $\mathcal{U}_{a_l} \setminus \{u_d\}$ (he does not know the IDs of swarm peers). In this way, each address cannot be linked to the corresponding principal, so each principal can protect his address privacy from swarm peers.

3.3 Routing among Swarms

In Swarms, the address block $\mathcal{IP}_a$ of each $\mathcal{U}_a$ is only known to members of selected swarms called *landmarks* of $\mathcal{U}_a$. More specifically, suppose that swarm $\mathcal{U}_b$ is a landmark of $\mathcal{U}_a$. Each principal $u_v \in \mathcal{U}_b$ knows a tuple $\langle a, \Psi_v^a\rangle$, where Ψ_v^a is a random λ-subset of $\mathcal{IP}_a$. The impact of the system-wide parameter λ on the address privacy will be discussed in Section 4.2.2. Packets for principals in $\mathcal{U}_a$ need be routed through a node in one of $\mathcal{U}_a$'s landmarks. This mimics the spy-movie scenes described earlier.

The issues we need further address include (1) how to select landmarks for each swarm $\mathcal{U}_a$ and (2) how to efficiently route a packet from any source swarm to any destination swarm. Regarding the first issue, it is desirable for $\mathcal{U}_a$ to have as few landmarks as possible so as to minimize the exposure of its address block $\mathcal{IP}_a$. The second issue requires a small number of intermediate swarms to avoid high communication overhead. Consider a simple case that swarm $\mathcal{U}_{(a-1) \bmod 2^\beta}$ is the single landmark of $\mathcal{U}_a$ for $0 \le a \le 2^\beta - 1$.

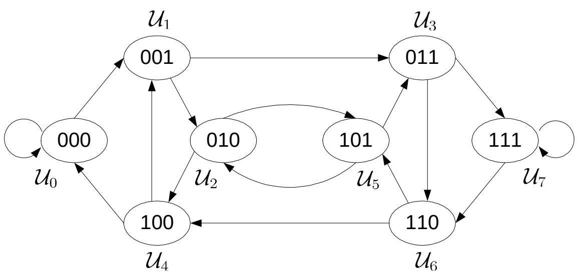

Figure 2: A de Bruijn graph of swarms for $\beta = 3$.

Then each swarm has the smallest possible number of landmarks, but the number of intermediate swarms between any two swarms is $O(2^\beta)$, the largest possible value. To balance these two factors, we view each swarm as a "virtual node" over the MANET substrate and its address block as its "network address." The landmarks of each swarm are like maintaining "routing information" about that swarm. Then it is easy to draw the analogy between routing among swarms and that in P2P networks [6]. This motivates us to apply many elegant results from P2P networks research.

In a P2P network with N' nodes, it is well-known that any two nodes can communicate in $O(\log N')$ hops (the *network diameter*) with each node maintaining the IP addresses about κ other nodes, where κ (the *node degree*) depends on the specific P2P architecture [6]. P2P networks based on de Bruijn graphs [5] can achieve diameter $O(\log N')$ with a constant and smallest possible $\kappa = 2$ [6]. Therefore, we decide to perform swarm routing using de Bruijn graphs [5], as doing so allows each swarm to have a small number of landmarks and communicate with any other swarm via $O(\log N')$ intermediate swarms.

Recall that each swarm index is $\{0, \cdots, 2^\beta\}$ and can be converted into a β-bit binary index. The de Bruijn graph in Swarms is a directed graph with 2^β nodes, each corresponding to a swarm. There are 2 outgoing and 2 incoming edges at each swarm $\mathcal{U}_a$. In particular, if $\mathcal{U}_a$'s binary index is $a_1 \cdots a_\beta$, then $\mathcal{U}_a$ has an outgoing edge to swarms with binary indices $a_2 \cdots a_\beta 0$ and $a_2 \cdots a_\beta 1$. In addition, swarms with binary indices $0a_1 \cdots a_{\beta-1}$ and $1a_1 \cdots a_{\beta-1}$ each have an outgoing edge to swarm $\mathcal{U}_a$ and are its only 2 landmarks. Fig. 2 shows a de Bruijn graph of swarms for $\beta = 3$, where swarms' binary indices are depicted in the eclipses. As we can see, $\mathcal{U}_1$'s landmarks are $\mathcal{U}_0$ and $\mathcal{U}_4$, while $\mathcal{U}_0$ only has one landmark $\mathcal{U}_4$ (excluding itself).

Without considering the MANET substrate, shortest-path routing between any two swarms follows the approach given in [5, 6]. Assume that a packet need be routed from the source swarm $\mathcal{U}_x$ with binary index $x_1 \cdots x_\beta$ to the destination swarm $\mathcal{U}_y$ with binary index $y_1 \cdots y_\beta$. First $\mathcal{U}_x$ finds the longest match between the suffix of $x_1 \cdots x_\beta$ and the prefix of $y_1 \cdots y_\beta$. If there is no match, the packet is routed along the path $x_1 \cdots x_\beta \rightarrow x_2 \cdots x_\beta y_1 \rightarrow \cdots \rightarrow x_\beta y_1 \cdots y_{\beta-1} \rightarrow y_1, \cdots y_\beta$. If $\mathcal{U}_x$ finds a match of length l, which is denoted by $x_{\beta-l+1} \cdots x_\beta$ for $1 \leq l \leq \beta - 1$, the packet can be routed following $x_1 \cdots x_\beta \rightarrow x_2 \cdots x_\beta y_{l+1} \rightarrow \cdots \rightarrow x_{\beta-l+1} \cdots x_\beta y_{l+1} \cdots y_\beta = y_1 \cdots y_\beta$. If we combine both cases, the route from $\mathcal{U}_x$ to $\mathcal{U}_y$ consists of $\beta - l - 1$ intermediate swarms $(0 \leq l \leq \beta - 1)$ whose binary indices are β-bit substrings (read from the left) of $x_2 \cdots x_\beta y_{l+1} \cdots y_{\beta-1}$. As an example, a packet from swarm $\mathcal{U}_0$ to $\mathcal{U}_3$ in Fig. 2 follows the swarm path $000 \rightarrow 001 \rightarrow 011$, while a packet from $\mathcal{U}_3$ to $\mathcal{U}_1$ follows $011 \rightarrow 110 \rightarrow 100 \rightarrow 001$.

3.4 Packet Routing and Forwarding

Our previous description about routing among swarms ignores the underlying MANET routing operations. In this section, we illustrate the compete process of routing a packet from any source principal to any destination principal. Without loss of generality, we assume that principal u_s in swarm $\mathcal{U}_{a_0}$ intends to send some packets to principal u_d in swarm $\mathcal{U}_{a_l}$. In addition, the shortest path from $\mathcal{U}_{a_0}$ to $\mathcal{U}_{a_l}$ consists of l $(0 \leq l \leq \beta)$ hops on the de Bruijn graph, denoted by $\mathcal{U}_{a_0} \rightarrow \mathcal{U}_{a_1} \rightarrow \cdots \rightarrow \mathcal{U}_{a_l}$. Source u_s does not know the exact address IP_d of u_d, but he knows that u_d is in the a_l^{th} swarm. The Swarms scheme can be built upon any MANET routing protocol, be it proactive (like OLSR [2]) or reactive (like AODV [7]). We will explain the slightly different operations needed to be taken when either approach is used. We consider the following two cases.

3.4.1 Case 1: $l = 0$

This means that u_s and u_d are in the same swarm. Since u_s knows $\mathcal{IP}_{a_0}$ and $IP_d \in \mathcal{IP}_{a_0}$, he multicasts to nodes in $\mathcal{IP}_{a_0} \setminus \{IP_s\}$ the following information:

$$\mathsf{Payload} := \langle a_l, \mathcal{E}(PU_d, u_s, u_d, K_{s,d}), \mathcal{E}(K_{s,d}, \mathrm{DATA}) \rangle, ^1$$

where $K_{s,d}$ is a session key picked by u_s and DATA is the information for u_d. We want to emphasize here that u_s does not know the corresponding principals $\mathcal{U}_{a_0} \setminus \{u_s\}$.

After finding a_l (i.e., a_0) in the received Payload, each principal in $\mathcal{U}_{a_l} \setminus \{u_s\}$ knows that he is a potential receiver and then attempts to decrypt $\mathcal{E}(PU_d, u_s, u_d, K_{s,d})$ using his private key. Only u_d can get a meaningful decryption result in which his ID u_d appears and then use the embedded $K_{s,d}$ to decrypt $\mathcal{E}(K_{s,d}, \mathrm{DATA})$. Other principals simply dump Payload. To minimize public-key decryptions, all the principals in $\mathcal{U}_{a_l} \setminus \{u_s\}$ buffers $(u_s, u_d, K_{s,d})_{K_d}$. If seeing it in a later Payload, they can immediately decide whether that Payload is intended for them or not without having to do a public-key decryption. In this way, each involved principal in $\mathcal{U}_{a_l}$ merely performs one public-key decryption per communication session between u_s and u_d.

Payload appears to be sent by node IP_s from the viewpoint of principals $\mathcal{U}_{a_l} \setminus \{u_s\}$ because they cannot link IP_s to u_s. We also assume that effective countermeasures (such as [8]) are in place such that when observing a packet output from a node, the adversary cannot differentiate whether the observed packet was initiated or just forwarded by that node. As a result, node IP_s cannot be pinpointed as the packet initiator. This is very important because otherwise u_d (if malicious) may be able to link IP_s to principal u_s, thus breaking u_s's address privacy.

3.4.2 Case 2: $1 \leq 1 \leq \beta$

Recall that u_s knows a λ-subset $\Psi_s^{a_1}$ of $\mathcal{IP}_{a_1}$, the address block of swarm $\mathcal{U}_{a_1}$. What u_s need do is to unicast Payload to the closest node in $\Psi_s^{a_1}$.

If the underlying routing protocol is proactive, node IP_s always maintains a route to each node in $\Psi_s^{a_1}$. Assume that

[1] A keyed-hash message authentication code should be attached to each packet to ensure its integrity, but we ignore it for brevity.

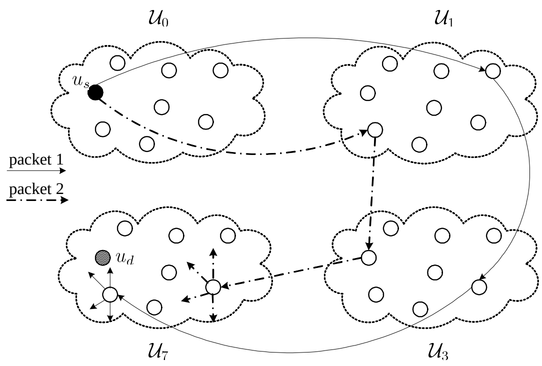

Figure 3: An exemplary routing process.

the route to node $IP_{r_{a_1}} \in \Psi_s^{a_1}$ is the shortest among all such routes. Then u_s transmits $\mathsf{Payload}$ to node $IP_{r_{a_1}}$ while blind to the corresponding principal $u_{r_{a_1}}$.

If the underlying routing protocol is reactive, node IP_s first searches its routing table for the shortest route to nodes $\Psi_s^{a_1}$ and then delivers $\mathsf{Payload}$ along that route. If such a route does not exist, node IP_s follows the route discovery process of the routing protocol to send a special route request for nodes $\Psi_s^{a_1}$. Any node receiving this request sends a route reply to node IP_s if knowing a valid route to any node in $\Psi_s^{a_1}$. Then IP_s chooses the shortest one among all route replies along which $\mathsf{Payload}$ is delivered.

Assume that in both cases node $IP_{r_{a_1}}$ is the one receiving $\mathsf{Payload}$ from IP_s. It first checks the embedded a_l to determine whether he is in the destination swarm. If so (i.e., $a_1 = a_l$), it passes $\mathsf{Payload}$ to its host principal $u_{r_{a_1}}$ who will process $\mathsf{Payload}$ as described in Case 1. In addition, node $IP_{r_{a_1}}$ need multicast $\mathsf{Payload}$ to nodes $\mathcal{IP}_{a_1} \setminus \{IP_{r_{a_1}}\}$, no matter whether $u_{r_{a_1}}$ is the intended destination principal u_d or not. In this way, attackers cannot guess whether $IP_{r_{a_1}} = IP_d$ based on whether $IP_{r_{a_1}}$ further multicasts $\mathsf{Payload}$.

If not in the destination swarm (i.e., $a_1 \neq a_l$), node $IP_{r_{a_1}}$ need further forward $\mathsf{Payload}$ to the next swarm $\mathcal{U}_{a_2}$. Since it knows a λ-subset $\Psi_{r_{a_1}}^{a_2} \subseteq \mathcal{IP}_{a_2}$, it follows what node IP_s did to unicast $\mathsf{Payload}$ to the closest node in $\Psi_{r_{a_1}}^{a_2}$. This process continues until $\mathsf{Payload}$ reaches some node in $\mathcal{IP}_{a_l}$ which in turn multicasts $\mathsf{Payload}$ to all the other nodes in $\mathcal{IP}_{a_l}$.

Fig. 3 gives an example, where $u_s \in \mathcal{U}_0, u_d \in \mathcal{U}_7$, and the shortest path is $\mathcal{U}_0 \to \mathcal{U}_1 \to \mathcal{U}_3 \to \mathcal{U}_7$. Note that each link there is virtual and may be a multi-hop link in the MANET substrate. As we can see, node mobility may result in a different relaying point in each involved swarm, which means that packets from u_s to u_d may traverse physically totally different routes in the MANET substrate. Destination u_d can either directly receive each packet for him or get it from some node in $\mathcal{U}_7$.

The packet forwarding process bears some similarity to the packet-rerouting technique presented earlier but with better resilience to sporadic network partitions. In particular, if any middleman in the packet-rerouting technique is unreachable, packets from u_s cannot be successfully delivered to u_d. In the Swarms scheme, this is unlikely to occur as long as at least one node in each intermediate swarm is reachable.

3.5 Enhancing Routing Efficiency

The routing process in Section 3.4 can be further improved for better routing efficiency. As an example, node IP_s delivers a packet via a chosen route to $IP_{r_{a_1}} \in \Psi_s^{a_1}$ which it believes to be the best route to reach nodes $\Psi_s^{a_1}$. However, some intermediate nodes on the chosen route may have better routes to other nodes in $\Psi_s^{a_1}$ than the chose route or even to a subsequent swarm $\mathcal{U}_{a_i}$ $(2 \leq i \leq l)$. It is better to allow these nodes to reroute the packet to improve the routing efficiency. To make this possible, IP_s has to append $\Psi_s^{a_1}$ to each packet. If each address is δ bits long, the resulting packet overhead is then $|\Psi_s^{a_1}|\delta$ bits long. This possibly large overhead can be reduced by using hash functions or a Bloom filter [1]. Due to the space limitation, we only present the former technique here and will report the latter in a full version of this paper. For ease of illustration, we let $IP_{r_{a_0}} = IP_s$ and consider the general case that node $IP_{r_{a_i}} \in \mathcal{IP}_{a_i}$ need deliver $\mathsf{Payload}$ to the next swarm $\mathcal{U}_{a_{i+1}}$ $(0 \leq i \leq l-1)$.

If the underlying routing protocol is reactive, the route request from node $IP_{r_{a_i}}$ contains a list $\{\mathcal{H}(IP_c)|IP_c \in \Psi_{r_{a_i}}^{a_{i+1}}\}$ of hash values instead of $\Psi_{r_{a_i}}^{a_{i+1}}$, where $\mathcal{H}(*)$ denotes an arbitrary good hash function. Any node receiving this request will send a route reply to node $IP_{r_{a_i}}$ if it knows a route to any address whose hash value appears in $\{\mathcal{H}(IP_c)|IP_c \in \Psi_{r_{a_i}}^{a_{i+1}}\}$.

Assume that node $IP_{r_{a_{i+1}}} \in \Psi_{r_{a_i}}^{a_{i+1}}$ is the destination of the best route chosen by $IP_{r_{a_i}}$ under either a proactive or reactive routing protocol. Node $IP_{r_{a_i}}$ delivers $\langle\{\mathcal{H}(IP_c)|IP_c \in \Psi_{r_{a_i}}^{a_{i+1}}\}, \mathsf{Payload}\rangle$ along the chosen route. Each intermediate node, say IP_e, attempts to reroute the packet as follows.

- Node IP_e first searches its routing table for the best route to each swarm $\mathcal{U}_{a_j}$ $(i+1 < j \leq l)$. If multiple routes are available, the one to the largest j value is chosen to bypass as many intermediate swarms as possible. This is only possible when node IP_e is in one landmark of $\mathcal{U}_{a_j}$ and thus knows a λ-subset $\Psi_e^{a_j}$ of $\mathcal{IP}_{a_j}$. Then IP_e sends $\langle\{\mathcal{H}(IP_c)|IP_c \in \Psi_e^{a_j}\}, \mathsf{Payload}\rangle$ along the chosen route.

- Otherwise, IP_e checks whether it has a better route to another address whose hash value is in $\{\mathcal{H}(IP_c)|IP_c \in \Psi_{r_{a_i}}^{a_{i+1}}\}$. If multiple such routes are found, the best one is chosen to reroute $\langle\{\mathcal{H}(IP_c)|IP_c \in \Psi_{r_{a_i}}^{a_{i+1}}\}, \mathsf{Payload}\rangle$.

- If no new route is found in either case, node IP_e continues forwarding $\langle\{\mathcal{H}(IP_c)|IP_c \in \Psi_{r_{a_i}}^{a_{i+1}}\}, \mathsf{Payload}\rangle$ along the current route to node $IP_{r_{a_{i+1}}}$.

Using hash functions may unfortunately introduce *false positives*, which occur when an address, say IP_o, is erroneously considered an element of $\Psi_{r_{a_i}}^{a_{i+1}}$ because $\mathcal{H}(IP_o)$ appears in $\{\mathcal{H}(IP_c)|IP_c \in \Psi_{r_{a_i}}^{a_{i+1}}\}$. False positives may occur in both route discovery and packet forwarding. Node $IP_{r_{a_i}}$ can detect the false positives in route discovery by checking whether the destinations of reported routes indeed belong to $\Psi_{r_{a_i}}^{a_{i+1}}$. False positives in packet forwarding may cause $\langle\{\mathcal{H}(IP_c)|IP_c \in \Psi_{r_{a_i}}^{a_{i+1}}\}, \mathsf{Payload}\rangle$ to be rerouted to a node not in $\Psi_{r_{a_i}}^{a_{i+1}}$.

Now let us analyze the false positive probability. To simplify the analysis, we assume that $\{\mathcal{H}(IP_c)|IP_c \in \Psi_{r_{a_i}}^{a_{i+1}}\}$

contains $\lambda = |\Psi_{r_{a_i}}^{a_{i+1}}|$ hash values, each of μ bits. Then the saving in packet overhead is $\frac{\delta-\mu}{\delta}$, and the false positive probability is given by $prob_{FP}^{\text{hash}} = 1 - (1 - \frac{1}{2^\mu})^\lambda$. Note that a single false positive occurring in an intermediate node does not necessarily mean that Payload will not reach the next swarm. The reason is that nodes on the wrong route will also attempt to reroute Payload and may possibly "correct" the false positive.

4. SECURITY ANALYSIS

Here we first define an address-privacy metric and then use it to quantitatively measure the address privacy offered by Swarms.

4.1 Address-Privacy Metric

We use an entropy-based metric to quantify the address privacy offered by Swarms. Assume that attackers intend to find out the network address IP_d of a target principal u_d. Let $\mathcal{IP}$ denote $\{IP_i\}_{i=1}^N$. After obtaining some information from the network, attackers assign each address $IP_w \in \mathcal{IP}$ a probability p_w as being IP_d. This a-posteriori probability distribution can be described by a discrete random variable Z_d with probability mass function $p_w = \Pr\{Z = IP_w\}$ s.t. $\sum_{IP_w \in \mathcal{IP}} p_w = 1$.

Definition 1. *The **address-privacy degree** of principal* u_d

$$AP_d = H(Z_d) = -\sum\nolimits_{IP_w \in \mathcal{IP}} p_w \log_2(p_w) \tag{1}$$

AP_d measures the uncertainty that attackers have about which address in $\mathcal{IP}$ is IP_d. One can also interpret AP_d as the number of bits of additional information that attackers need in order to precisely identify IP_d within $\mathcal{IP}$. It follows that $0 \le AP_d \le \log_2(|\mathcal{IP}|) = \log_2(N)$ [3]. The lower bound is achieved when IP_d is assigned a probability of one and each $IP_w \in \mathcal{IP} \setminus \{IP_d\}$ is assigned a probability of zero; the upper bound is attained when each $IP_w \in \mathcal{IP}$ is assigned an equal probability of $1/N$, meaning that all the addresses are equally likely to be IP_d as viewed by attackers (the ideal case).

4.2 Analysis

Now we use the address-privacy degree to measure the address privacy provided by Swarms. For ease of illustration, we still use the previous example in which principal $u_s \in \mathcal{U}_{a_0}$ is allowed to send packets to $u_d \in \mathcal{U}_{a_l}$. Also recall that the shortest path from $\mathcal{U}_{a_0}$ to $\mathcal{U}_{a_l}$ on the de Bruijn graph is $\mathcal{U}_{a_0} \to \mathcal{U}_{a_1} \to \cdots \to \mathcal{U}_{a_l}$. Assume that principal u_s is compromised so that the adversary obtains the information that principal u_d belongs to swarm $\mathcal{U}_{a_l}$. Then attackers attempt to find out the network address IP_d of u_d who happens to have a critical role in the MANET. We assume that attackers are smart in the sense that they will additionally compromise only nodes in the swarms along the shortest path from $\mathcal{U}_{a_0}$ to $\mathcal{U}_{a_l}$. In doing so, they can more quickly and surreptitiously narrow down the search of IP_d to $\mathcal{IP}_{a_l}$. For simplicity, we have the following assumptions:

- Swarms $\{\mathcal{U}_a\}_{a=0}^{2^\beta-1}$ are of equal cardinality L, i.e., $|\mathcal{U}_a| = |\mathcal{IP}_a| = L$ for $0 \le a \le 2^\beta - 1$, where $\beta \ge 1$;
- $N = 2^\beta * L$ (the network size);
- Principals in any landmark swarm of any swarm $\mathcal{U}_a$ each know a random λ-subset of $\mathcal{IP}_a$ (cf. Section 3.3), where $1 \le \lambda \le |\mathcal{IP}_a| = L$.

We assume that the adversary also knows the above system parameters. Let C denote the number of nodes (including node IP_s) the adversary has compromised before locating IP_d, so we have $1 \le C \le L + l - 1$. There are two cases to be considered.

4.2.1 *Case 1:* $l = 0$

This means that u_s and u_d are in the same swarm. From u_s, attackers know the address block $\mathcal{IP}_{a_l}$.

Let $\Upsilon \subset \mathcal{IP}_{a_l}$ ($|\Upsilon| = L - C$) be the set of non-promised nodes, each of which is equally likely to be IP_d as viewed by attackers. All the other nodes in $\mathcal{IP} \setminus \Upsilon$ are impossible to be IP_d. Therefore, attackers assign the following probabilities:

$$p_w = \begin{cases} \frac{1}{|\Upsilon|} = \frac{1}{|\mathcal{IP}_{a_l}| - C} = \frac{1}{L-C} & IP_w \in \Upsilon \\ 0 & IP_w \in \mathcal{IP} \setminus \Upsilon \ . \end{cases}$$

We thus have

$$AP_d^s = \log_2(L - C), \tag{2}$$

where the superscript s indicates u_d's correspondent u_s with regard to which the address-privacy degree is analyzed.

4.2.2 *Case 2:* $1 \le l \le \beta$

This means that u_s and u_d are in different swarms and separated by $(l-1)$ intermediate swarms. After compromising u_s, the adversary sequentially compromises one node in each intermediate swarm until compromising one in $\mathcal{U}_{a_l}$. Then he focuses on compromising nodes in $\mathcal{U}_{a_l}$.

In particular, recall that u_s knows a λ-subset $\Psi_s^{a_1}$ of $\mathcal{IP}_{a_1}$. Attackers then compromise a random node $IP_i \in \Psi_s^{a_1}$ from which they know a subset $\Psi_i^{a_2}$ of $\mathcal{IP}_{a_2}$. They proceed to compromise a random node $IP_j \in \Psi_i^{a_2}$ to obtain a subset $\Psi_j^{a_3}$ of $\mathcal{IP}_{a_3}$. This process continues until attackers compromise one node in $\mathcal{IP}_{a_l}$ (or $\mathcal{IP}_{a_{l-1}}$ when $\lambda = L$) from which they know the whole address block $\mathcal{IP}_{a_l}$. From then on, attackers focus on compromising the rest nodes in $\mathcal{IP}_{a_l}$ to locate IP_d.

If $1 \le C \le l - 1$, then attackers have compromised one node in each of $\{\mathcal{IP}_{a_i}\}_{i=0}^{C-1}$ and thus known $\{\mathcal{IP}_{a_i}\}_{i=0}^{C-1}$. None of the nodes in $\bigcup_{i=0}^{C-1} \mathcal{IP}_{a_i}$ are likely to be IP_d, while all the other nodes in $\mathcal{IP}$ are equally likely to be IP_d. Therefore, attackers assign the following probabilities:

$$p_w = \begin{cases} 0 & IP_w \in \bigcup_{i=0}^{C-1} \mathcal{IP}_{a_i} \\ \frac{1}{N - \sum_{i=0}^{C-1} |\mathcal{IP}_{a_i}|} = \frac{1}{(2^\beta - C)L} & \text{o.w.} \end{cases}$$

It follows that

$$AP_d^s = \log_2(N - CL) = \log_2(L) + \log_2(2^\beta - C). \tag{3}$$

If $C = l$, attackers have compromised exactly one node in $\mathcal{IP}_{a_{l-1}}$, say IP_t, from which they know $\mathcal{IP}_{a_{l-1}}$ and a λ-subset $\Psi_t^{a_l}$ of $\mathcal{IP}_{a_l}$. From the attackers' point of view, all the nodes in $\Psi_t^{a_l}$ are equally likely to be IP_d with probability $\frac{1}{|\mathcal{IP}_{a_l}|} = \frac{1}{L}$, all the nodes in $\bigcup_{i=0}^{C-1} \mathcal{IP}_{a_i}$ are unlikely to be IP_d, and the rest nodes are equally likely to be IP_d with probability $\frac{1 - |\Psi_t^{a_l}|/L}{X} = \frac{1-\lambda/L}{X}$, where $X = N - \sum_{i=0}^{C-1} |\mathcal{IP}_{a_i}| - |\Psi_t^{a_l}| = (2^\beta - l)L - \lambda$. That is, attackers assign the following probabilities:

$$p_w = \begin{cases} 0 & IP_w \in \bigcup_{i=0}^{C-1} \mathcal{IP}_{a_i} \\ \frac{1}{L} & IP_w \in \Psi_t^{a_l} \\ \frac{1-\lambda/L}{(2^\beta - l)*L-\lambda} & \text{o.w.} \end{cases}$$

It follows that

$$\begin{aligned} AP_d^s &= \tfrac{\lambda}{L}\log_2(L) + (1-\tfrac{\lambda}{L})\log_2(\tfrac{(2^\beta-l)L-\lambda}{1-\lambda/L}) \\ &= \log_2(L) + (1-\tfrac{\lambda}{L})\log_2(\tfrac{(2^\beta-l)L-\lambda}{L-\lambda}). \end{aligned} \quad (4)$$

If $l+1 \leq C \leq L+l-1$, attackers have compromised at least one node in $\mathcal{IP}_{a_l}$ to know $\mathcal{IP}_{a_l}$ and started to focus on compromising nodes in $\mathcal{IP}_{a_l}$. Let $\Upsilon \subset \mathcal{IP}_{a_l}$ be the set of non-promised nodes, where $|\Upsilon| = |\mathcal{IP}_{a_l}| - (C-l) = L-C+l$. From the viewpoint of attackers, each node in Υ are equally likely to be IP_d, while all the other nodes in $\mathcal{IP} \setminus \Upsilon$ are impossible. Therefore, attackers assign the following probabilities:

$$p_w = \begin{cases} \frac{1}{|\Upsilon|} = \frac{1}{L-C+l} & IP_w \in \Upsilon \\ 0 & IP_w \in \mathcal{IP} \setminus \Upsilon\ . \end{cases}$$

We thus have

$$AP_d^s = \log_2(L-C+l). \quad (5)$$

Note that $AP_d^s = 0$ when $C = L+l-1$. This means that the adversary is pretty sure that the only non-compromised node in $\mathcal{IP}_{a_l}$ is the target IP_d.

4.3 Discussion

To analyze the above results, we use $AP_d^{s(2)}$, $AP_d^{s(3)}$, $AP_d^{s(4)}$, and $AP_d^{s(5)}$ to denote the address-privacy degree derived in Eqs. (2), (3), (4), and (5), respectively.

We first discuss the impact of λ. It can can easily shown that $AP_d^{s(4)}$ monotonically decreases with λ if β is sufficiently large (e.g., $\beta \geq 3$). Since $0 < \lambda \leq L$, we have $\log_2(L) \leq AP_d^{s(4)} < \log_2(L) + \log_2(2^\beta - l)$. It is thus wise to choose a smaller λ to achieve better address privacy. On the other hand, a larger λ is preferable for better routing reliability and efficiency because more candidates routes towards next swarm will be available. It is necessary to strike a good balance between them in practice.

Now we check the impact of the number C of compromised nodes. Obviously, $AP_d^{s(2)}$ monotonically decreases with C. So do $AP_d^{s(3)}$ and $AP_d^{s(5)}$. In particular, $\log_2(L)+\log_2(2^\beta - l+1) \leq AP_d^{s(3)} \leq \log_2(L)+\log_2(2^\beta-1)$ and $0 \leq AP_d^{s(5)} \leq \log_2(L-1)$. Therefore, we have $AP_d^{s(5)} < AP_d^{s(4)} < AP_d^{s(3)}$ and can conclude that the address privacy of u_d decreases as C grows, which complies with the intuition.

Now let us discuss the impact of l, the distance in hops between $\mathcal{U}_{a_0}$ and $\mathcal{U}_{a_l}$ on the de Bruijn graph. Let l_1, l_2 be two integers satisfying $0 \leq l_1 < l_2 \leq \beta$. We can easily verify that $AP_d^s(l=l_1) \leq AP_d^s(l=l_1)$ for any given $C \in \{1, ..., L+l_2-1\}^2$. Therefore, the address privacy of $u_d \in \mathcal{U}_{a_l}$ with regard to $u_s \in \mathcal{U}_{a_0}$ is in direct ratio to l. Let $\Omega_d \subset \mathcal{U}$ be the set of principals allowed to send messages directly to u_d. The address privacy of u_d is determined by the nearest correspondent $u_t \in \Omega_d$. That is, $AP_d = \min_{u_x \in \Omega_d} AP_d^x = AP_d^t$. Therefore, it is necessary to put Ω_d in swarms as distant from $\mathcal{U}_{a_l}$ as possible. How to allocate principals to different swarms to satisfy diverse address-privacy requirements is an open problem worthy of further study.

[2] $AP_d^s(l=l_1)$ is defined to be zero for $C \geq L+l_1$.

5. CONCLUSION

This paper introduced address privacy as a new security requirement for MANETs. We presented the Swarms scheme to prevent attackers from establishing the one-on-one mappings between network addresses and MANET principals. The security of Swarms was quantitatively evaluated using an entropy-based information-theoretic metric. As the future work, we plan to investigate tradeoffs between address privacy and communication efficiency as well as strategies to satisfy diverse address-privacy requirements of MANET principals.

6. ACKNOWLEDGMENTS

This work was supported in part by ERIF (IIT) and the US National Science Foundation under grant CNS-0716302.

7. REFERENCES

[1] B. Bloom. Space/time trade-offs in hash coding with allowable errors. *Comm. ACM*, 13(7):422–426, July 1970.

[2] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). RFC 3626, Oct. 2003.

[3] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2 edition, July 2006.

[4] M. Gruteser and D. Grunwald. Enhancing location privacy in wireless LAN through disposable interface identifiers. *ACM Mobile Networks and Applications*, 10(3):315–325, June 2005.

[5] M. Imase and M. Itoh. Design to minimize diameter on building-block networks. C-30(6):439–442, June 1981.

[6] D. Loguinov, A. Kumar, V. Rai, and S. Ganesh. Graph-theoretic analysis of structured peer-to-peer systems: Routing distances and fault resilience. In *ACM SIGCOMM'03*, pages 395–406, Karlsruhe, Germany, Aug. 2003.

[7] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing. RFC 3561, July 2003.

[8] Y. Zhang, W. Liu, W. Lou, and Y. Fang. MASK: anonymous on-demand routing in mobile ad hoc networks. 5(9):2376–2385, Sep. 2006.

[9] Y. Zhang, W. Liu, W. Lou, and Y. Fang. Securing mobile ad hoc networks with certificateless public keys. *IEEE Transactions on Dependable and Secure Computing*, 3(4):386–399, Oct.-Dec. 2006.