Secure and Privacy-Preserving Communication in Hybrid Ad Hoc Networks

Srdjan Čapkun[1], Jean-Pierre Hubaux[2] and Markus Jakobsson[3]

[1,2]Laboratory for Computer
Communications and Applications (LCA)
Swiss Federal Institute of
Technology Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
srdan.capkun@epfl.ch, jean-pierre.hubaux@epfl.ch

[3]RSA Laboratories
174 Middlesex Turnpike
Bedford, MA 01730
USA
mjakobsson@rsasecurity.com

ABSTRACT

We present a scheme for secure and privacy-preserving communication in hybrid ad hoc networks. Our scheme enables users to secure communication and to protect their anonymity and location privacy. Our approach is based on frequently changing node pseudonyms and cryptographic keys, which enable users to avoid being identified by the locations they visit, or by the type of traffic they generate. We show how our scheme can be effectively used for secure and private routing in hybrid ad hoc networks. We study the robustness of the proposed solution with respect to various attacks. We further show that the proposed solution introduces a very moderate overhead to the network operation.[1]

Categories and Subject Descriptors

C.0 [**Computer-Communication Networks**]: [Security and protection]

General Terms

Security, Anonymity, Location, Privacy

Keywords

Anonymity, Privacy, Mobile Networks, Hybrid Networks

1. INTRODUCTION

Ad hoc networks are appealing for a number of reasons, including their flexibility of deployment. However, these networks are known to have limited scalability, and do not provide, as such, an access to large scale networks such as the Internet. For this reason, several researchers have recently studied the possible interconnection of ad hoc networks to a backbone by means of one or several access points [25]. In this way, a mobile station can communicate with an access point over several other mobile stations in a multi-hop fashion. It is reasonable to assume that one or several *network operators* are in charge of the proper operation of such networks, and receive an appropriate remuneration from their subscribers.

This approach seems to be promising, as it combines the best of both worlds: the extended reach and scalability of classical, large-scale wired networks with the flexibility of ad hoc networks; in this paper, we call a network resulting from this combination a “hybrid ad hoc network”. A possible incarnation of such a network is a multi-hop Wi-Fi network. In this case, the network operator would typically be a Wireless Internet Service Provider. Other examples of hybrid ad hoc networks include multi-hop cellular networks [37].

In order to gain acceptance from the users, hybrid ad hoc networks must provide an appropriate level of security. Indeed in general, a user trusts his network operator, but he does not trust the other users; he may also distrust the operators of the networks in which he roams.

The paper addresses both routing security *and* privacy preservation; we will show that the two mechanisms can be embedded in the same protocols.

A number of papers [16, 28, 32] have recently addressed the problem of secure routing in ad hoc networks. We base our threat analysis on the extensive description of attacks provided by these research efforts.

As for privacy, we show how to provide the two following features: anonymity and location privacy. Privacy International [38] defines four categories of privacy: information privacy, bodily privacy, communication privacy, and territorial privacy. Location privacy is a particular case of information privacy and can be defined as the *ability to prevent other parties from learning one's current and past locations* [3].

[1] The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322 (`http://www.terminodes.org`).

EPFL-IC Technical report no. IC/2004/10

Anonymity can be defined as *the state of being not identifiable within a set of subjects called the anonymity set* [30].

We propose a set of protocols that protects users' anonymity and location privacy. More precisely, our scheme enables each mobile node to keep its location and its identifier hidden from other network nodes. We assume, however, that the network operator has access to the locations and the identifiers (and potentially also to the user identities) of the registered mobile nodes. Our approach is based on the frequently changing of node pseudonyms, so that the users avoid being identified by the locations they visit, or by the type of traffic they generate. We do not restrict our investigation to privacy, but we also investigate how privacy can be achieved and user accountability enforced. In our scheme, accountability is enforced through dynamic, but verifiable, cryptographic keys used by the nodes to secure their communication.

Although originally designed for data-centric hybrid ad hoc networks, our scheme can be, with some minor modifications, used to enable secure and privacy-preserving communication in hybrid voice-centric networks.

In this paper we specifically: (i) present an overview of privacy threats, (ii) propose a scheme for secure and privacy-preserving communication, and (iii) present a quantitative analysis of privacy.

The rest of the paper is organized as follows. We introduce our system and security model in Section 2. In Section 3, we describe privacy challenges and goals. In Section 4, we describe the basic mechanisms of our scheme. In Section 5, we provide a detailed description of our privacy preserving routing protocol. In Sections 6 we quantify the level of achieved privacy, analyze security and the performance of our scheme. We describe the related work in Section 7. Finally, we present our conclusions in Section 8.

2. SYSTEM MODEL

In this section we describe our network model along with the security and trust assumptions.

2.1 Network model

Our system consists of a set of *access points*, mutually connected via a high-speed backbone, and a set of *mobile nodes*. Each access point controls a bounded geographic area called a *control area*. All communications between nodes and between a node and a access point are wireless. We assume that the access points and mobile nodes have the same power range, and that the latter is smaller than the size of the control area. We further assume that all links are bi-directional, meaning that the link between two nodes (or between a node and a access point) exists only if both mobile nodes are in each others' power range. This means that the majority of nodes in the control area will not be able to communicate directly with an access point, but will use other mobile nodes as relays to reach the bases stations.

In order to make our study as generic as possible, we will consider that both hosts involved in the communication access the backbone in a multi-hop fashion.

When a source node S wishes to send a message m to a destination D, it transmits a packet p containing m to a first access point BS_S (which is typically the geographically closest access point to S) using an *uplink protocol*. If the destination node is mobile, the packet is then sent to a second access point BS_D, which is typically the closest station to the destination D. Then, BS_D transmits a packet containing m to the intended destination D using a *downlink protocol*. If the destination node is a fixed (internet) host, the packet is routed through the fixed network using traditional network protocols. Both the uplink and downlink protocols are multi-hop, i.e., they require the participation of nodes on the route. These nodes are typically peers of the source and destination. Later we detail the two protocols.

We assume that all nodes in the control area are loosely synchronized; we later describe how to avoid clock skew.

2.2 Security and Trust

We initially assume that all access points are run by the same operator (we describe later how to loosen this assumption). Each mobile node has a unique *identifier*, and carries a *secret key*, both of which are also known by the network operator[2]. Neither the identifier nor the secret key is disclosed to other parties. Like in cellular networks, there is a contractual agreement between a user of the mobile node and the network operator, which defines mutual rights and obligations. This contract defines policies that are enforced by the operator, charging, privacy protection and tariff. Notably, the network central authority (through its access points) acts as a network supervisor, and protects the network from user misbehavior. The access points therefore monitor node behavior, maintain node reputations and detect node misbehavior. As a reaction to node misbehavior, central authority limits node's access to a subset of services or excludes the node from the network, temporarily, or permanently, by revoking its membership certificate.

The network membership is controlled by the network central authority and each node carries a certificate of membership that it uses to prove to other nodes that it is a part of the network. Furthermore, each node is able to uniquely sign a message such that other users can verify that the message originated from the legitimate network node, but no one, except the central authority can identify which node signed the message. This is important for ensuring that the execution of network protocols is both secure and anonymous, and that the users are accountable for their behavior by the uniqueness of their signatures.

However, the users do not have mutual contractual agreements, and are not willing to trust each other with their identities and locations, nor do they want to trust each others' nodes to correctly execute networking functions (forwards packets, or provide accurate routing information).

To summarize, we assume that each node carries a secret key shared with the network central authority and a set of

[2] For simplicity, we assume that the network operator distributes secret keys and node identifiers to the access points under its control. In reality, nodes would establish shared secret keys with the access points, using a certificate issued to them by the central authority.

keys, certified by the operator, which it uses to secure the operation network protocols.

3. PRIVACY GOALS AND CHALLENGES

In this section we outlay our main design goals and privacy challenges for hybrid ad hoc networks.

3.1 Design goals

We want to design a system that enables the users anonymous and location private communication. Location privacy is the ability to prevent other parties from learning one's current and past locations, and anonymity is the state of being not identifiable within a set of subjects called the anonymity set. Here, we define these terms more precisely in the context of hybrid ad hoc networks.

One of the main goals of our scheme is to enable source and destination anonymity. Source anonymity is defined as the property that a particular message is not linkable to any source, and vice-versa. A similar definition applies to destination anonymity. Unlinkability in this context means that the probability that a particular message was sent by a given source and/or received by the same destination is the same as imposed by the a priori knowledge. This means that the process of sending and/or receiving messages does not reveal any additional information about the identities of the source and/or destination that was not already known to the attacker prior to the message transmission.

In our scenario, we want to achieve the following. Clearly, the source S needs to know the identity D of the destination, but not its location. The access points need to know who the source and the destination are, and where they are located, so as to route the messages accordingly. The access points need to know the identities of the source and the destination, in order to check if they are registered network nodes. However, nobody else, including the nodes on the route of the packet, should be able to infer either the source of any packet that they observe, or its destination. Furthermore, nobody should be able to infer where the nodes are located. This means that neither a node not registered to the network, nor a registered network node, could infer the identities of the communicating parties or their locations by observing network traffic. Here, we consider that a location of the node is compromised if an attacker can infer the distance in terms of number of hops (relay nodes) from the node to the access point(s), or its exact physical location. Note that we do not assume any sophisticated mechanisms for node positioning, such as the use of GPS.

Anonymity can be measured with various metrics, among which two are the most common: one based on anonymity set and other based on entropy. In our system, if the attacker holds the list of registered network nodes, the maximum degree of anonymity that the system can provide is proportional to the size of the list; in this case, the list corresponds to the anonymity set of the network. We will assume that the network has a sufficiently large anonymity set, so that it thus provides a reasonable anonymity to the users. The second metric is more sophisticated and is computed based on probabilities assigned to each identity (e.g., the probability that a given user is the originator of a message). In our analysis, we will reason in terms of both of these metrics.

3.2 Privacy challenges

In hybrid ad hoc networks, users' privacy is at risk from various threats. Here we overview the most important ones.

Malicious/Compromised users
In hybrid ad hoc networks, nodes need to maintain their neighborhood information for various reasons, mainly for routing and packet forwarding and as a prerequisite to running distributed network algorithms (e.g., positioning). Thus, to enable proper network operation, the nodes would need to disseminate to other nodes their identifiers, topology information and/or locations. This information would then be freely available to any registered network node, or even to a passive attacker that observes network communication. It is thus clear that even a single node, by logging the identifiers of its neighbor nodes, can gather much information about other users' behavior (e.g., if the attacker stays at the same location for a longer time, it can observe the frequency at which other nodes visit this location).

Active attacks can be far more sophisticated. A simple example of such an attack is when an attacker periodically explores network topology by asking for routes to other nodes. Another possible attack is if an attacker positions itself close to the access point and thus gathers the information about the communication between nodes. If the routing protocol is not secure, the attacker could even, regardless of its location, advertise the shortest route to the access point, collect the the traffic, and analyze it to observe which pairs of nodes currently communicate, and what are the nodes' distances to the access point(s). Another important set of attacks are those launched by communication parties against each other, meaning that a source wants to discover a location of the destination, or vice versa.

Untrusted network operators
Nodes roaming in untrusted networks are susceptible to attacks from malicious network operators. Unless the user does not protect its privacy, an untrusted operator could easily trace users and/or reveal their true identity. This is especially important as the number of network operators in hybrid ad hoc and multi-hop WiFi networks might be significantly larger than the number of operators of today's cellular networks. It will be thus hard to regulate all operators and risks of privacy violation would be higher.

Unique network addresses, interface addresses and cryptographic keys
One of the main potential causes of users' privacy vulnerabilities in hybrid ad hoc networks could be the uniqueness of the identifiers and keys that the nodes use to communicate and to secure their communication. Typically, each node uses a static and unique network address to perform network layer operations, and static and unique interface addresses for media access control (MAC) operations. Furthermore, to secure their mutual communication, nodes need to establish keys between each other and with the access points. These keys enable nodes to authenticate the source of the messages that they receive and to protect the confidentiality and integrity of the messages that they send. The fact that the keys are unique either to the node (public key cryptogra-

phy), or to the pair of nodes (symmetric key cryptography) enables a malicious user to track other users.

Radio fingerprinting
Besides unique identifiers, the nodes are equipped with radio transceivers whose emitted signals contain unique fingerprints [34]. An attacker can, therefore, identify a mobile device by the unique "fingerprint" that characterizes its signal transmission. This process is normally used by cellular network operators to prevent cloning fraud; namely, a cloned phone does not have the same fingerprint as the legal phone with the same electronic identification numbers. This technique can be used by the attacker to track mobile nodes, given that the attacker remains in the power range of the node. A similar technique is that an attacker observes the signal to noise (S/N) ratio of nodes' signals. If a node is static for a given period of time and uses pseudonyms to protect its identity, the attacker can infer that the pseudonyms are generated by the same node since the S/N ratio did not change. Recent measurements in WiFi networks [13] show that S/N based attack can significantly reduce the capability of the nodes to prevent tracking.

4. OVERVIEW OF THE SOLUTION

In this section, we introduce two important ingredients of our privacy preserving scheme: node pseudonyms, and dynamic public keys. At the end of the section, we define the notation that we use in the paper.

4.1 Node pseudonyms

As we mentioned earlier, each node shares a secret key with the access point. This key and the node's true identity are known only to the node and to the authority that controls the access point. A node pseudonym changes over time, according to the following equation:

$$P_S(t) = HMAC_{K_S}(ID_S, t)$$

where P_S is the node pseudonym at time step t, HMAC is a keyed hashing function, ID_S is the true node identity, and K_S is the secret key that the node shares with the central authority. HMAC can be implemented by any iterative cryptographic hash function, such as, MD5 or SHA-1. The cryptographic strength of HMAC depends on the properties of the underlying hash function. Thus, the pseudonyms of the node will be of the same size as the output of the hash function that is used for keyed hashing; if used with SHA-1, the HMAC output is 160 bits long. Depending on the size of each control area, and the number of nodes in the control area, the pseudonym length can be reduced, without significantly increasing the probability of identifier collision. For this, the pseudonyms obtained from the original equation can be truncated or hashed into shorter bit strings. To guarantee that the access point and the node generate the same node pseudonyms at the same time, the time is slotted and t as an input to the HMAC function represents a time step, and not the actual timestamp provided by the devices. We later discuss how to choose the time granularity. Alternative ways to generate node pseudonyms can be envisioned. One solution consist in generating node pseudonyms by making use of pseudo-random number generators.

4.2 Dynamic keys

To ensure that the communication between nodes and the access points is performed through legitimate nodes, the nodes that belong to the same control area need to be able to verify that the messages they receive come from the registered network nodes. At the same time, we need to ensure that the central authority can track the actions of each individual node and that the misbehavior of nodes can thus be detected and sanctioned.

Securing ad hoc routing protocols in multi-hop wireless networks is notoriously hard. Researchers have identified a number of attacks that can be mounted equally against pure ad hoc and hybrid ad hoc networks. These attacks include: excessive route requests, blackhole attack [16], rushing attack [18], wormhole attack [15], etc. In hybrid ad hoc networks, due to the presence of access points, routing will be easier to secure, as access points can control node behavior, act as on-line central authorities and sanction misbehavior. Nevertheless, like in ad hoc networks, to protect the network against these attacks, nodes need to be able to authenticate each others' messages and even to keep their communication confidential. For this, nodes need to share secret keys or hold authentic public-keys. We propose here, a privacy-preserving key management scheme for hybrid ad hoc networks.

A naive solution would be to make use of **control area-wide secret keys**. This scheme assumes that the nodes that belong to the same control area share a common secret key. This key is generated and in an authentic and confidential manner distributed to all the nodes in the control area by the access point that controls the control area. The key is periodically updated by the access point. The introduction of the control area-wide key enables each node to verify if other nodes belong to the network, without revealing any information about its identity or the identity of the other nodes. However, if any of the control area nodes is compromised, the attacker can use the control area-wide key to attack the network without being detected. Furthermore, the access point will not be able to detect node misbehavior as each node uses the same control area key for authentication to other nodes.

Given the drawbacks of the control area-wide key scheme, we propose a different scheme that we call a **dynamic public key scheme.** In this scheme, along with its changing pseudonym, each node A holds a set of public/private key pairs $(PK_A^1/PrK_A^1, ..., PK_A^n/PrK_A^n)$ and certificates signed by the central authority, certifying these keys. This set can be generated either by the node or by the central authority. If it generates the key pairs, the node sends the public keys from the pairs to the central authority, which certifies them and sends the certificates back to the node. If the public/private key pairs are generated by the central authority, they are sent to the node along with the certificates containing the public keys. The certificates of the public keys signed by the central authority certify only that the holder of the private key corresponding to the public key in the certificate is a registered network node. The certificates have the following format:

$$Cert_A^k = [PK_A^k, SIG_{PrK_{Auth}}(PK_A^k)]$$

where k denotes the position of the key in the key set of A, and PrK_{Auth} is the private key of the central authority[3]. We note here that the exchange of public/private key pairs between a node and the central authority is protected by the key shared between the node and the central authority.

A node uses its public/private key pairs to establish symmetric secret keys with its neighbors. Each time that a node changes its pseudonym, it changes the public/private key pair and establishes new symmetric keys with its neighbors. If a node encounters a new neighbor, it establishes a secret key with the new neighbor using the same public/private key that it uses for this period. Before a node runs out of public/private key pairs, it generates some new pairs and sends them to the central authority for certification. The central authority replies with the certified keys. This update of public keys and certificates can be performed in a number of ways, through periodic updates, or by piggybacking the certificates on uplink and downlink traffic. Moreover, as we discuss in Section 6, the nodes can reuse the private/public key pairs after some time, without compromising their location and identity privacy.

The nodes do not have to change their public keys (and pseudonyms) at a given frequency, but this change can be event driven (e.g. when a node starts a new session). For efficiency, the speed of key changes can be limited by the access points, but each user can determine if it wants its pseudonyms and keys being changed with a lower frequency. To enable this, the nodes need to change their keys and pseudonyms at frequencies which are related to the given basic frequency[4].

We will comment briefly on other alternatives to this proposal in Section 7.

In our proposal, mutual node authentication is performed automatically, so that routing is secured permanently. However, the system can be also designed in such way that nodes perform mutual authentication only if they are requested to do so by the access point; notably, if the access point detects misbehavior within a given region of its control area, it can enforce mutual node authentication in that region to prevent future misbehavior. This optimization can reduce the communication overhead introduced by the dynamic key scheme.

Update frequency
The frequency of the key and pseudonym change can be chosen arbitrarily by the base stations and mobile nodes. To increase the level of anonymity and location privacy, nodes should increase their update frequencies. However, as we show in Section 6.3, pseudonym/key change frequency is only one of the factors that determines the degree of privacy that can be achieved, whereas others are related to node mobility and attacker strength. We further conclude that in our scenario the sufficient frequency of pseudonym change is in the order of $1/min$.

4.3 Further notation

By BS_S we denote the access point that controls the control area in which node S is located. By $MAC_K(m)$ we denote the message authentication code of message m with the key K. A one-way hash function on message m we denote by $H(m)$. By $E_K(m)$, we denote a message encrypted with the key K, and by $SIG_{PrK}(m)$ we denote the signature over message m with the private key PrK.

5. PRIVACY PRESERVING ROUTING

In this section, we introduce the **P**rivacy **P**reserving **R**outing protocol (PPR). This protocol consists of four sub-protocols that we describe in brief as follows. We first describe the *downlink protocol* that is used for routing from the access point BS_D to the destination node D; we then describe the *uplink protocol* that is used to route packets from the source node S to the first access point BS_S; next we describe the *inter-station protocol* used to communicate messages between access points; finally, we describe the *bookkeeping protocol* that is used by the access points to keep track of node locations, pseudonyms and network topology.

5.1 Protocol overview

In this section we briefly overview PPR.

Downlink. As we will show later in more detail, the downlink protocol is a source routing protocol and it is a rather straightforward. The access point BS_D first determines the route to the destination D, given the information that it has about the location and topology of the nodes in its control area. It computes the current pseudonyms of the nodes on the route, and includes them in the packet, after which it sends the packet to the first node on the route. The first node, addressed by its pseudonym, removes its pseudonym descriptor from the packet and transmits the resulting packet to the next node on the route. Eventually, and assuming the route is not broken, the packet is received by the intended destination D.

Uplink. It is evident that a simple modification of the downlink protocol cannot be used for uplink communication. The reason is that the packet originator S does not know the pseudonyms of the nodes on the path to the access point BS_S, nor does it have access to the database specifying the location of various nodes. For this reason, we designed the downlink protocol as a distance vector protocol. In fact, given the requirements on privacy, nodes will not build routing tables, as these are irrelevant in that they are constantly changing with the rapid updates of pseudonyms. Instead, each node will keep track of its distance (in terms of number of hops) to the closest access point, along with the time at which this distance was known to be correct. When a packet is sent from one node to another, the sending node will announce its believed distance to the access point, and its neighbors will determine whether to route the packet based on this information. Nodes will update the distance information over time to reflect the changes of the topology.

Inter-station. Here we briefly describe the inter-station protocol used for the communication between two access points, one controlling the control area of the source, and the other controlling the control area of the destination. If both access points are under the control of the same authority,

[3] Note that neither the identity of the node nor the position of the key in its key set are disclosed to other nodes.
[4] This basic frequency is a system parameter and is controlled by the access point.

the inter-station protocol is straightforward. Each uplink packet that arrives to the source access point BS_S is simply forwarded to the destination access point BS_D, where the message is verified. Subsequently, a corresponding downlink packet is created, encrypted with the keys that the access point shares with nodes on the path, and sent to the destination. If the node does not trust the access point controlling its control area, the inter-station protocol is somewhat more complex. We elaborate this further in Section 5.4.

Book-keeping. The access points keep records of the identities, pseudonyms, keys and distances of the nodes in their cells. Based on trust and pre-established policies, the access points also exchange this information. The databases are indexed by all five of these types of information, and are updated by piggybacking information on uplink messages and by periodic updates. Not only access points, but also mobile nodes keep information about their location and distance to the access points. This information is updated either from the downlink traffic or upon the execution of the periodic updates.

Discussion. We note here that the PPR is a proactive routing protocol, and that other, potentially fairly different solutions for routing in hybrid ad hoc networks can be proposed. Nevertheless, as we will show, the way that secure and privacy-preserving communication is implemented in PPR can be a useful inspiration for making other (possibly reactive, or proactive-reactive) protocols secure and privacy-preserving.

5.2 Uplink

In this section we describe the uplink protocol in more detail. As we already described, nodes establish symmetric secret keys with their neighbors. As we show, we use these keys to secure network protocols and privacy of users.

The uplink protocol works as follows. Each node regularly updates the list of its neighbors (their pseudonyms) and their distances to the access point. Each neighbor and distance update are properly secured with the keys that the node shares with its neighbors. When the node S needs to send a packet, it chooses the next hop node from the list of its neighbors and forwards the packet to the chosen node. The choice of the next hop node is based on the distance information provided by the neighbors; typically, the source forwards the message to the node that is the closest to the access point. Each consecutive node repeats this procedure until the message reaches the access point. The nodes maintain their distances to the access point with a book-keeping protocol, as described in Section 5.5.

An example of the uplink protocol execution is shown on Figure 1. The figure shows the message evolution from the message source node S, through nodes A and B, to the access point BS_S.

Here, Up means that the message is an uplink message, t_S, t_A and t_B are timestamps that guarantee message freshness, BS_S is the identifier of the access point to which the message is sent, $E_{K_S}(D, m)$ is the encrypted identifer of the message and of the destination identifier (it can only be decrypted by the access point), and m is the message from S intended for the destination D. In each packet, the first pseudonym in the message represents the message source and the second pseudonym represents the intended message destination.

`UPLINK PROTOCOL`

$$
\begin{array}{rl}
S: & MHead = [P_S(t), P_A(t), t_S, Up, BS_S] \\
: & E_S = E_{K_S}(D, m) \\
: & M_{SA} = MAC_{K_{SA}}(MHead, E_S) \\
S \rightarrow A: & [\underline{P_S(t)}, \underline{P_A(t)}, Up, \underline{t_S}, BS_S] \mid \underline{E_S} \mid \underline{M_{SA}} \\
 & \\
A: & \text{check the validity of } M_{SA} \\
: & MHead = [P_A(t), P_B(t), t_A, up, BS_S] \\
: & E_A = E_{K_A}(P_S(t), E_S) \\
: & M_{AB} = MAC_{K_{AB}}(MHead, E_A) \\
A \rightarrow B: & [\underline{P_A(t)}, \underline{P_B(t)}, Up, \underline{t_A}, BS_S] \mid \underline{E_A} \mid \underline{M_{AB}} \\
 & \\
B: & \text{check the validity of } M_{AB} \\
: & MHead = [P_B(t), BS_S, t_B, Up, BS_S] \\
: & E_B = E_{K_B}(P_A(t), E_A) \\
: & M_B = MAC_{K_B}(MHead, E_B) \\
B \rightarrow BS_S: & [\underline{P_B(t)}, \underline{BS_S}, Up, \underline{t_B}, BS_S] \mid \underline{E_B} \mid \underline{M_B} \\
 & \\
BS_S: & \text{decrypt } E_B, E_A, \text{ and } E_S, \text{ check} \\
 & \text{the validity of } M_B; \\
: & \text{update the distances of } S, A \\
 & \text{and } B \text{ in the distance database}
\end{array}
$$

Figure 1: An example of the run of the uplink protocol. The fields changing from the previous hop are underlined.

M_{SA} is the MAC of the packet content, computed by the source S, to prove to the forwarding node (A) that S is indeed a registered network node and that the packet content was not modified. The secret key K_{SA} is a key shared between A and S and it is established through the dynamic key scheme prior to the protocol execution. When the neighbor A of the source S receives the packet, it checks the validity of M_{SA}, and encrypts the source node pseudonym concatenated to the received encrypted message hash E_S with the key K_A that it shares with the access point. It then replaces E_S with a newly created E_A, replaces the distance information in MHead with its own distance, and replaces the MAC received from S by a new MAC computed with the key that it (A) shares with the next forwarding node (B). Then, A forwards the message to B. A similar operation over the message is performed at each node that forwards the message. Per-hop re-encryption of the message m is important so that the access point can verify the hop count and the identities of the nodes on the routes.

Encryption of the message and of the destination by the source guarantees that no one but BS_S can infer the identity of the destination D. The re-encryption of message m by all the nodes on the route guarantees that the message cannot be tracked by an attacker. Specifically, at each node, the message is re-encrypted and therefore altered, so that an attacker will not be able to track the messages in the network. This is especially important if the attacker controls several nodes. The node's location privacy is protected by its pseudonym and by the fact that its distance to the access point does not propagate any further than its one-hop neighborhood, so that most of the nodes will not be able to link not even the node pseudonyms with their distances to the access point.

However, in the proposed protocol, some information within the messages is still sent in the clear. Specifically, node pseudonyms are sent in the clear to enable nodes to efficiently verify if the messages are intended for them. Node pseudonyms could also be hidden from passive attackers by encrypting them with the keys shared between nodes. However, this would require each node to try to decrypt the destination addresses of all messages that it collects on its interface. Furthermore, even if node pseudonyms are encrypted, the same problem would appear with interface identifiers and interface pseudonyms. As with node pseudonyms, interface pseudonyms can be hidden, but also at the expense of efficiency.

5.3 Downlink

In this section, we describe the downlink protocol for the communication between the access point BS_D and the mobile node D. We assume that the message m, sent by node S reached BS_D, and needs now to be routed to node D. We also assume that the access point has information about the topology of its control area, which it obtained by means of the book-keeping protocol. Thus, the access point knows the optimal route to the node D and sends the message to D via that route. An example of the downlink protocol execution is shown on Figure 2. In this example the path from the access point to node D contains two nodes C and D.

Here, $Down$ field denotes that the message is sent on a downlink, t_{BS}, t_C and t_E are the timestamps that guarantee that the messages are fresh, BS_D is the identifier of the access point that sent the message, $E_{K_D}(S, m)$ is the identifier of the source S and the message m encrypted by the key shared between the access point and the destination, $P_D(t), P_C(t)$ and $P_E(t)$ are the pseudonym of the nodes D, C and E at time t, and m is the message sent by S for D. K_CE and K_ED are the symmetric keys that C and E, E and D pairwise share. The integrity, authenticity and confidentiality of the message and of the source identifier are protected by the encryption of the message by a secret key K_D that the access point shares with the destination D. After creating the downlink packet, the access point sends it to the first node on the route to D (in this case node C, addressed by its pseudonym $P_C(t)$). Node C then verifies if the message truly originated from the access point by decrypting E_{K_C} and verifying the packet content MAC M_{BS}. If the message is valid, C forwards the message to the next node on the list (node $P_E(t)$), but decrypts E_{K_C}. Node E then performs the same decryption, verification and reduction, and forwards the message to the next node on the list. This is repeated until the message reaches its destination. At the destination, node D verifies the packet authenticity and integrity by checking M_{ED}, and decrypts the message and the identity of the message source.

Here, like in the uplink protocol, the per-hop changing packet content prevents attackers from tracking the message through the network and the node pseudonyms are not forwarded further then the nodes' neighborhoods.

DOWNLINK PROTOCOL

$$\begin{array}{rl}
BS_D: & MHead = [BS_R, P_C(t), t_{BS}, Down, BS_D] \\
: & E_{BS} = E_{K_C}(P_E(t), E_{K_E}(P_D(t), E_{K_D}(S, m))) \\
: & M_{BS} = MAC_{K_C}(E_{BS}, MHead) \\
BS_D \rightarrow C: & [\underline{BS_R}, \underline{P_C(t)}, \underline{t_{BS}}, Down, BS_D] \mid \underline{E_{BS}} \mid \underline{M_{BS}} \\
 & \\
C: & \text{check the validity of } M_{BS}\text{, decrypt } E_{K_C} \\
: & MHead = [P_C(t), P_E(t), t_C, Down, BS_D] \\
: & E_C = E_{K_E}(P_D(t), E_{K_D}(S, m)) \\
: & M_{CE} = MAC_{K_{CE}}(E_C, MHead) \\
C \rightarrow E: & [\underline{P_C(t)}, \underline{P_E(t)}, \underline{t_C}, Down, BS_D] \mid \underline{E_C} \mid \underline{M_{CE}} \\
 & \\
E: & \text{check the validity of } M_{CE}\text{, decrypt } E_{K_E} \\
: & MHead = [P_E(t), P_D(t), t_E, Down, BS_D] \\
: & E_E = E_{K_D}(S, m) \\
: & M_{ED} = MAC_{K_{ED}}(E_E, MHead) \\
E \rightarrow D: & [\underline{P_E(t)}, \underline{P_D(t)}, \underline{t_E}, Down, BS_D] \mid \underline{E_E} \mid \underline{M_{ED}} \\
 & \\
D: & \text{check the validity of } MAC_{K_{DR}}\text{, decrypt } E_E
\end{array}$$

Figure 2: An example of the run of the downlink protocol. The fields changing from the previous hop are underlined.

If the route is broken and the message delivery fails, the node that is not able to forward the message, reports the broken link to the access point, that updates the route and re-sends the message.

5.4 Inter-station protocol

As we already described, if all access points are controlled by the same authority, the inter-station protocol is relatively straightforward. Each uplink packet that arrives to the source access point BS_S is simply forwarded to the destination access point BS_D, where the message is decrypted and MACs are verified. Subsequently, a corresponding downlink message is created, encrypted with the keys that the access point shares with nodes on the path, and sent to the destination. The inter-station protocol is somewhat different if the node (source/destination) does not trust the access point controlling its control area.

We consider the following scenario: a node S is situated in the control area controlled by an untrusted access point BS_U, which is controlled by an untrusted network operator NU. However, S does not trust BS_U either with its identity or with its keys, but wants, nevertheless, to establish communication through it. Note here that we assume that BS_U is at all times connected to the home network HN_S of S through a high speed link. Here, there are two issues related to trust. Node S does not trust NU either with its identifier or with the identifier of the destination node, but only shows to NU its pseudonyms. However, in order to route packets to S, BS_U needs to know its distance in hops to S, and the location of S in the topology of its control area. This already reveals some information about S to BS_U, but hiding this information from BS_U would be difficult to achieve efficient routing to S.

To allow a node to communicate through an untrusted network, we propose the following. The node pseudonyms are still computed only by the node itself and by its home network. If another node A needs to communicate to node S, A's messages will be first sent to HN_S. HN_S then computes S's pseudonym and sends the packet to the untrusted access point, with this pseudonym as the destination address. BS_U then sends this packet to S, along the path that it determines from its control area topology, with the S's pseudonym as a destination address.

If the guest node S wants to send a message through the untrusted network, it needs to prove to the BS_U and to other nodes in the control area that it is a legitimate node. For this, S still uses the same dynamic public keys as previously described, and its public keys are still certified by its home network HN_S. However, HN_S's public key is now certified by the untrusted network NU. By checking both these certificates, the nodes that belong to the untrusted network NU can verify if S is a node that legitimately participates in network operations. This allows to S to communicate with the nodes in the untrusted control area, while still protecting its anonymity. Furthermore, BS_U can charge HN_S (and implicitly S) for the service, as it can register the packets passing through its access points. An alternative solution is if NU issues short term certificates directly to S, so that the nodes that belong to the untrusted network do not have to check two certificates upon authentication. For this, we could use an efficient certificate revocation system proposed by Micali [27] which alleviates the need for certificate revocation lists.

5.5 Book-keeping

As we briefly described earlier, the access points keep records of the time, distances, identities, and pseudonyms of the nodes in their control area.

Secure and Private Topology Discovery.
Topology discovery is initiated by the access point to discover network topology that it uses for determining optimal routing paths. Researchers have already proposed several topology discovery algorithms in wireless networks [9]. To the best of our knowledge, all topology discovery proposals have considered a non-adversarial setting. Here, we propose a simple topology discovery mechanism that is both secure and privacy-preserving.

The scheme works as follows. First, the access point sends a topology discovery request, authenticated with its public key, with the following message format:

$$BS \rightarrow * : TREQ, rid, BS, t \mid SIG_{PrK_{BS}}(TREQ, BS, t)$$

where $TREQ$ is a field that denotes that the message is the topology discovery request, BS is the identifier of the access point that initiates the request (so that only the nodes controlled by that access point reply), and t is a timestamp that guarantees the freshness of the message.

Each node receiving this request forwards it to its neighbors if it has not seen the same request before; otherwise, it drops the packet. If it accepts the topology request, the node performs neighborhood discovery/update, authenticates its neighbors (with the keys that it shares with them), encrypts the neighbor list with the key that it shares with the access point and sends this encrypted message back to the node from which it received the request. This neighbor list includes both the neighbor pseudonyms and their public keys that they used to establish shared secret keys. The destination node merges the received information with its own neighborhood information, and forwards it further. Each node will repeat the procedure, until the certificates reach the access point. The access point will verify the signatures of the nodes, match the public keys to users' real identities, and reconstruct network topology. It is important to note that each node encrypts its neighborhood information, and that only the access point can decrypt the content of the message, whereas other nodes cannot modify or observe this topology information, and thus cannot reconstruct network topology by simply observing nodes' replies to topology discovery.

The frequency of the topology discovery is determined by the access point, and can be either fixed or can depend on the speed of the topology change, which can be measured by the access point based on the estimated optimality of the routes.

Here, we note that nodes can be tricked by an attacker into believing that they are each others' neighbors, which in return can introduce errors into topology discovery. This attack is similar to the Mafia fraud attack [10] and can be resolved by distance bounding techniques [6]. We do not see this attack as a major threat to the topology discovery and maintenance mechanisms, given that the access point collects information from all network nodes and can detect inconsistencies in the topology information, even if the nodes do not use distance bounding.

Topology update.
Maintenance. The nodes maintain their distance to the access point by collecting distance information from their neighbors. This information is timestamped and encrypted by the nodes, which guarantees that malicious outside nodes cannot insert false distance information and that the insertion of false distance information by compromised nodes can be detected by the access point. The timestamp in each distance update guarantees that the nodes accept only recent distance information. Nodes do not propagate their neighbors' pseudonyms to other nodes they just send to their neighbors their own believed distance to the access point at a given time. This prevents compromised nodes from gathering information about network topology; topology gathering by an attacker can be dangerous, even if the topology information contains node pseudonyms and not node identifiers.

Uplink. When a node forwards a packet in the uplink, it re-encrypts it with the keys that it shares with the access point. Thus, as the packet moves towards the access point, it stores the identifiers of the nodes on the path. When the access point receives the packet, it determines what nodes were on the route, and updates their topology information accordingly.

Downlink. When the access point transmits a packet in the downlink, it can piggyback the believed distances to the nodes on the route onto the packets it sends. Thus, each node on the route can decrypt its distance to the access point, and if the distance information is more recent than the information that it holds, it will update its distance.

Secure time synchronization

In our protocols we assume only loose time synchronization. We assume that the reference time is given by the access points (which are mutually synchronized). Whenever a node gets in the power range of a access point, it performs clock synchronization by measuring the time of flight to the access point and by taking into account the difference between its local time and the global network time provided by the access point. This protocol is executed in the following manner: the node first sends the challenge to the access point, encrypted with the key shared between the access point and the node. The access point immediately replies with the same challenge concatenated with its current time and encrypted with the same key. In this message the access point can also include its processing time. The node measures the time that was necessary for the challenge to reach the access point and to return. The node then deduces the access point processing time and half of this roundtrip time from the reference time that it received from the access point and sets its clock to that value.

Time synchronization can also be performed by leveraging on neighboring nodes. For example, each node could ask the nodes in its neighborhood for the time reference, and perform the same time synchronization as with the access point. A node disseminating false time information could be easily detected if the majority of neighboring nodes provide true time information. The node would eventually synchronize with the access point and could also detect which nodes were disseminating false time reference. Upon the detection of false time information, each node can make a complaint to the access point and after some number of complaints, malicious nodes can be excluded from the network. To avoid malicious complaints against honest nodes, reputation systems can be used. The privacy of this scheme is guaranteed by the dynamic public key scheme described in the previous subsection.

If the nodes are equipped with GPS receivers (which provide accurate time reference) secure time synchronization mechanism can be avoided altogether. In this paper, we do not make such assumptions.

Recently, more sophisticated and very precise (μ second precision) time synchronization techniques have been proposed for wireless networks [11]. However, in our scheme, we do not need such a level of precision as we assume only loose time synchronization. This is because even to guarantee a high level of node privacy, node pseudonyms and public-keys do not need to change very frequently. Thus, the difference between node clocks can then be as high as several seconds. Our simple and secure time synchronization technique offers therefore a sufficient accuracy for our protocols as it can achieve microsecond precision with “off the shelf” components.

6. SECURITY AND PERFORMANCE

Having presented our privacy-preserving scheme, we now analyze its performance and resistance to various attacks.

6.1 Attacker model

We call a node *malicious* if it is controlled by a malicious adversary and cannot authenticate itself to the access points. We call a node *compromised* if it can authenticate itself to the access points (it is a registered network node), but is controlled by a malicious adversary. We assume that when a node is compromised, its secret keys and the other secrets that it shares with other nodes and the access points become known to the attacker. Thus, a compromised node is, for the access point and for other nodes, undistinguishable from an honest node. We further assume that when a node is compromised, this is not detected by other network nodes, nor by the central authority (at least for some time). A central authority (in this case network operator) can detect a malicious behavior, but it cannot detect if the node has been captured or compromised. We distinguish attackers according to the number of malicious and compromised nodes that they control. By Attacker-C-M we denote the attacker that controls C compromised and M malicious nodes [16].

Clearly, with the proposed privacy-preserving scheme and PPR protocol, we limit the information that attackers can obtain by observing network traffic and the actions that they can perform to track the nodes and infer users' real identities.

All links in the considered network are wireless; hence an Attacker-0-1 (a single malicious node) can:

- observe if nodes (pseudonyms) in its neighborhood send/receive messages
- observe which nodes (pseudonyms) in its neighborhood are neighbors to each other
- observe signal-to-noise (S/N) ratios of the devices in its neighborhood and try to link each S/N ratio with a given node pseudonym
- detect signal watermarks of the devices in its neighborhood and link them with node pseudonyms
- estimate how distant nodes in its neighborhood are from the access point (in term of number of hops), based on its physical distance to the access point.

A stronger attacker, one that controls a single compromised node (Attacker-1-0), can observe, in addition to the previously listed observations, accurate pseudonym distances to the access point of the nodes (pseudonyms) in its neighborhood and modify network traffic or generate traffic to infer nodes' locations or real identities. Thus, an Attacker-0-1 can only passively observe the traffic, whereas Attacker-1-0 can actively, by generating new traffic, try to infer more information about other nodes.

If an attacker controls several malicious or compromised nodes, it can observe the traffic generated from more pseudonyms and on a wider network area, and can, by combining the collected information, try to infer users' real identities and locations. The attacker can even observe the times at which packets are sent and by this detect which packets

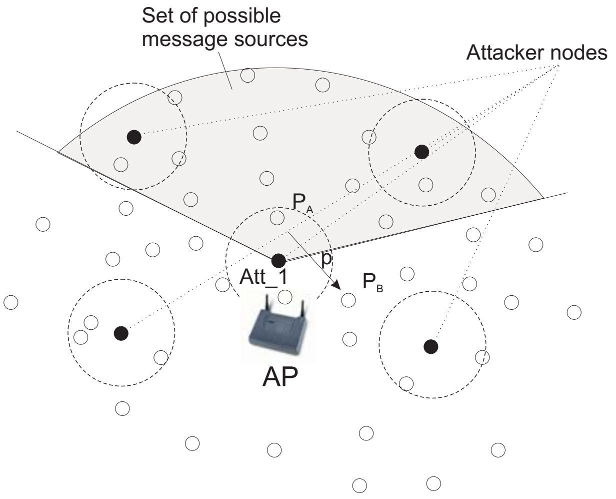

Figure 3: An example of a scenario in which an attacker node Att_1 observes a packet p being sent from P_A to P_B. The shaded area represents the attacker's estimation of the region from which the packet p could have originated.

carry the same message. Another simple attack is if an attacker sends a message to some destination D and tries to track the message to establish where D is located. This attack is, however, very ineffective as the messages go through a set of nodes typically not controlled by the attacker and through access points which act as message mixes thus by minimize the chance of D's location being detected.

6.2 Anonymity

Here, we analyze the level of source and destination anonymity achieved by our scheme. For this, we will use an anonymity metric based on entropy proposed in [33] and [21]. Let X be a discrete random variable with probability function $p_i = Pr(X = i)$, where i represents each possible value that X may take. In our case, each i corresponds to an element of the anonymity set (a node). We denote by $H(X)$ the entropy of the system after the attack has taken place. For each node belonging to the node set of size N, an attacker assigns a probability p_i. $H(X)$ can be calculated as:

$$H(X) = -\sum_{i=1}^{N} p_i \log_2 p_i$$

Thus the maximum entropy of the system equals to:

$$H_{max} = \log_2 N$$

where N is the size of the anonymity set. Based on this, we compute the *degree of anonymity* d provided by the system as:

$$d = \frac{H(X)}{H_{max}}$$

The degree of anonymity quantifies the amount of information the system is leaking. A system in which a user or a small group of users appear to be message originators or destinations, does not provide a high degree of anonymity.

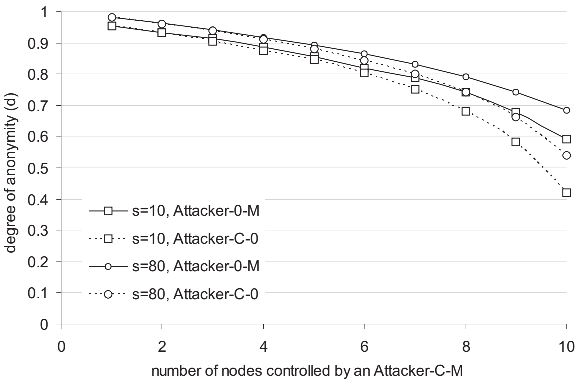

Figure 4: Pseudonym anonymity degree with Attacker-0-M and Attacker-C-0, for a control area with 80 nodes and for two sizes of the set of possible sources ($s = 10$ and $s = 80$).

Previously, we defined source/destination anonymity as the property that a particular message is not linkable to any source/destination, and vice-versa.

To analyze our privacy-preserving scheme, we observe two important aspects: (i) anonymity of node pseudonyms (i.e., linkability of the messages to node pseudonyms and their respective public keys) (ii) mutual linkability of node pseudonyms. These two aspects combined will give us a true degree of identifier anonymity.

We first observe the degree of pseudonym anonymity that our system provides. An Attacker-0-1 can observe the behavior of its neighboring nodes and try to link the messages that they transmit to their pseudonyms. Thus, if Attacker-0-1 observes that a node $P_A(t)$ sent a message to $P_B(t)$, it can conclude the following: (i) $P_A(t)$ is either the message source, or a forwarding node (ii) $P_B(t)$ is not the message source (iii) other nodes in the attacker neighborhood $P_1(t), ..., P_{k-1}(t)$ which have not sent any messages in the recent past are probably not sources of the message. For simplicity, we denote $p(X = P_A(t))$ by p_A.

The attacker assigns to each pseudonym that it observes a probability that this pseudonym is the source of the message:

$$p_A = \frac{1}{s}$$
$$p_B = 0$$
$$p_1 = ... = p_{k-1} = 0$$
$$p_i = \frac{1 - p_A}{N - k - 1}$$

where $s \leq N$ is the number of sources (including $P_A(t)$) that might have sent the message given that it is forwarded by $P_A(t)$; k is the number of neighbors of the attacker (including $P_B(t)$). The attacker can estimate this number based on the expected node (or even traffic) density in a given region. If $P_A(t)$ is located close to the access point, then almost any node in the control area is a potential source. If $P_A(t)$ is located on the edge of the control area, then only a few nodes can be potential message sources. However, Attacker-

0-1 does not know which pseudonyms are potential sources, it can only estimate their number. Thus, the entropy of this system can be computed as

$$H(X) = \frac{1}{s}\log_2(s) + (1 - \frac{1}{s})\log_2\frac{s(N-k-1)}{s-1}$$

If an attacker controls several malicious nodes (Attacker-0-M), each of the nodes it controls will provide information if the nodes in its neighborhood could be potential sources of the message. The entropy in this case is then

$$H(X) = \frac{1}{s'}\log_2(s') + (1 - \frac{1}{s'})\log_2\frac{s'(N-M'-1)}{s'-1}$$

where M' is the number of neighbors of malicious nodes (typically $M' = M \times k$), and $s' \leq s$ is the number of nodes in the set of possible sources that are not neighbors of the nodes controlled by the attacker.

In the case of Attacker-C-0, where the attacker controls a set of compromised nodes, the attacker can also exclude the nodes that it controls from the set of potential sources. Thus, for all C nodes under the attacker's control, $p(X = P_{Att}(t)) = 0$, and the entropy can be computed as

$$H(X) = \frac{1}{s'}\log_2(s') + (1 - \frac{1}{s'})\log_2\frac{s'(N-C'-C-1)}{s'-1}$$

where C' is the number of neighbors of compromised nodes (typically $C' = C \times k$). The maximum entropy for Attacker-0-M and Attacker-C-0 differs, as the size of the anonymity set for the first attacker is N (as the attacker does not control any registered network nodes), whereas it is $N - C$ for the second attacker (given that the attacker controls C network nodes). Thus, for Attacker-0-M, $H_{max}(X) = \log_2(N)$ and for Attacker-C-0 $H_{max}(X) = \log_2(N - C)$. The size of the anonymity set in the case of Attacker-0-M remains N even if the attacker knows that the set of possible sources s is smaller then N. This is because even if the attacker knows the size of the set of possible sources, it does not know their pseudonyms, and thus any of the pseudonyms can be within the set of possible sources. The same is valid in the case of Attacker-C-0, except that here, the size of the anonymity set is diminished for the number of nodes that the Attacker controls, and for which it knows the pseudonyms and knows that they are not the message sources.

On Figure 4 we show the pseudonym anonymity degree with Attacker-0-M and Attacker-C-0, for a control area with 80 nodes and for two sizes of the set of possible sources ($s = 10$ and $s = 80$), in the case where the attacker distributes the nodes uniformly over the network, and where each attacker node has the same number of neighbors ($k = 6$). As expected, as the set of possible sources gets smaller, and the number of attacker nodes increases, the pseudonym anonymity degree decreases. It is interesting to observe that the anonymity of the system does not decrease significantly with the reduction in size s of the set of possible sources. This is because even if the attacker knows the size of this set, it does not know which pseudonyms belong to the set, and thus any pseudonym has an equal chance of being in the set. Only if the attacker controls a larger number of nodes, pseudonym anonymity decreases significantly.

These results demonstrate the effectiveness of our scheme, as they show that our scheme prevents an attacker that controls a smaller number of nodes to jeopardize node anonymity. To link node pseudonyms to messages, an attacker thus needs to control a large number of nodes, and furthermore needs to be able to distinguish which packets carry the same messages.

If we would assume that an attacker has information about network topology, the pseudonym anonymity would be significantly smaller. This is because the size of the anonymity set would be reduced from N to s (for Attacker-0-M) and to $s - C''$ (for Attacker-C-0). Here, C'' is the number of attacker nodes that are located within the region of possible sources.

Pseudonym anonymity is not sufficient to quantify the degree of node/user anonymity in the system. Even if an attacker can link messages to node pseudonyms, as the pseudonyms are changing, it is difficult for the attacker to distinguish which messages were generated by the same entity, or better, which pseudonyms belong to the same node. Attackers can try to mutually link node pseudonyms in two obvious ways: by observing the S/N ratio of the devices and by observing their signal watermarks.

The first attack is simple to mount, but effective only in some situations. It consists in the following: an attacker observes S/N ratio of the signals of the nodes in its neighborhood. If the attacker detects the same the S/N ratio for two pseudonyms which are used one after another, the attacker can conclude that the two pseudonyms are used by the same node. However, this attack can be performed by the attacker only during a short period of time, as long as the node does not move. If the observed node moves, the attacker cannot correlate anymore the pseudonyms that it used while it was still with any other pseudonyms that it generates in the future.

The second attack is more powerful, but requires higher attacker sophistication. This attack is performed such that the attacker detects signal watermarks of a network node, and links them with node pseudonyms. Whenever a node changes its pseudonym, and the signal has the same fingerprint, the attacker can assume that the previously observed and the current pseudonyms are used by the same device.

6.3 Location Privacy

Both S/N and fingerprinting attacks aim at linking node pseudonyms, but also at tracking node locations. This is because the attacker nodes know their own current locations and know which pseudonyms appear in their neighborhood. What they need to complete the picture about node movements is to link node pseudonyms. The fingerprinting attack aims at doing exactly that. However, the fingerprinting attack has its limitations as well. One of the most obvious is that in order to track nodes effectively, the attacker needs to have many nodes installed all over the control area. However, this attack can be effectively countered by implementing radio transmitters that can randomize fingerprints. Nevertheless, if appropriate protection against this attack is not applied, this attack can be very effective.

If we exclude the possibility of fingerprinting attacks, node tracking becomes very difficult for the attacker, but not im-

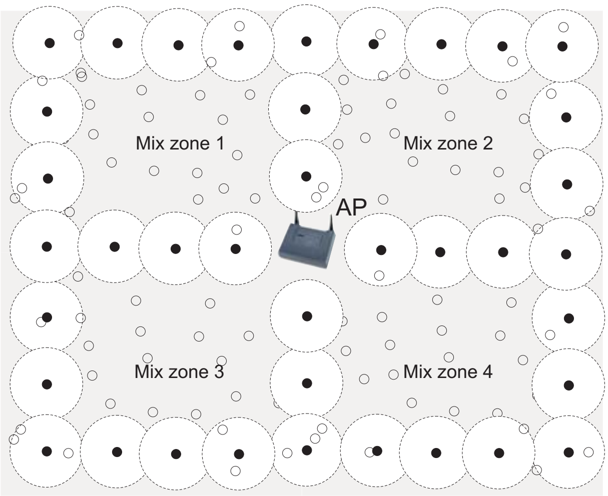

Figure 5: An example of a scenario in which the attacker divides the access point control area into four mix zones of equal size.

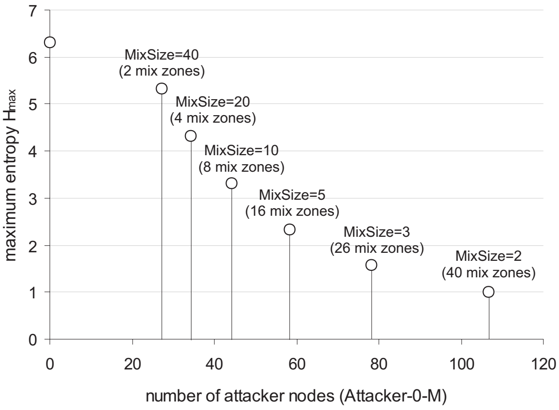

Figure 6: Maximum entropy as a function of the number of attacker nodes. The attacker divides the access point control area into sub-areas (smaller mix zones). The size of the mix zones determine the level of the system entropy.

possible. By installing a large number of nodes accross the access point control area, the attacker can track pseudonyms and correlate them by their locations. To illustrate this, it is sufficient to assume that the attacker has all the information about the past locations of node pseudonyms. Then, given that the users do not move in an unpredictable manner, an attacker can manage to correlate node pseudonyms and thus jeopardize users' location privacy. Thus, even if nodes change their pseudonyms very frequently, it does not increase the level of their privacy because they cannot change their locations at the same pace as they change their pseudonyms. As the nodes are being tracked, their locations will be sufficient for the attacker to conclude which pseudonyms are used by the same node. This means that an attacker that can monitor pseudonym locations at all times, has a high chance of tracking all network nodes. Note that to perform efficient pseudonym tracking, the attacker needs to precisely locate network nodes. For this, the attacker can use triangulization-based methods (for which at least three attacker nodes are needed to locate the pseudonym), or location fingerprinting (for which the attacker needs to perform detailed measurements of the access point control area). Besides the imprecisions introduced by the location algorithms, the attacker also needs to cope with an incomplete information about pseudonym locations. Therefore, it would be too pessimistic to assume that the attacker has a complete information on pseudonym locations. Instead, we will assume a more realistic scenario in which the attacker has only partial knowledge on pseudonym locations; namely we assume that the attacker can observes pseudonym locations only within some specific regions controlled by the attacker.

This problem was already observed in [3] by Beresford and Stajano. In their work, these authors define a term *mix zone*. A mix zone for a group of users is a connected spatial region of maximum size in which none of the users has registered any application callback. This means that a node, while in a mix zone, does not register its location with any authority. Thus, while in its mix zone, a node can change its identifier, so that it appears as a different entity when it leaves the zone. This scheme works only if there is a sufficient number of other users in the same mix zone (anonymity set) which guarantees a low probability of node's pseudonyms (the pseudonym that it uses before it enters a zone and the one that it uses when it leaves the zone) being correlated. In our scenario, mix zones can be defined in the following way: a mix zone for a set of nodes is a connected spatial region of a maximum size in which none of the nodes is in the power range of any of the nodes controlled by the attacker. This is illustrated in Figure 5. In this example, the attacker divided the access point control area into four mix zones thus by reducing the entropy of the system. Namely, if the attacker subdivides the control area, it will reduce the number of pseudonyms that can be correlated, as it knows which pseudonyms entered which zone, and which pseudonyms have left the zone. Thus, for each pair of pseudonyms $P_X(t_1)$ and $P_Y(t_2)$, the probability that $X = Y$ (that both pseudonyms originated from the same node) can be computed as:

$$Pr(P_X(t_1), P_Y(t_2) : X = Y) = 1/MixSize$$

where $P_X(t_1)$ is the pseudonym observed by the attacker entering a mix zone at time t_1, and $P_Y(t_2)$ is the pseudonym observed by the attacker leaving the zone at time $t_2 > t_1$, and $MixSize$ is the number of nodes in the mix zone. The maximum entropy of each of the mix zones is determined by the number of nodes in that zone. Thus, if the attacker divides the access point control area into smaller mix zones (Figure 5), the entropy drops and it is easier for the attacker to correlate pseudonyms. On Figure 6 we show the maximum entropy as a function of the number of attacker nodes. Here, the more nodes that the attacker controls, the sizes of the mix zones become smaller and their number increases; hence, the entropy of the system decreases.

Here we note that in the observed scenario, the attacker does not hold any additional information about nodes' possible behavior (e.g., probable node trajectories). As shown by Beresford and Stajano, given that the user motion can be predictable within some geographical location, an attacker can leverage on that data in order to better correlate nodes' pseudonyms. The attacker thus can create a matrix $M[i, j]$ of frequencies with which nodes go from one zone i to another zone j through a mix zone z, where zones i and j are controlled by an attacker. This data can then be further used by the attacker to compute the probabilities that the pseudonyms belong to the same node, and thus by to reduce the entropy.

The success of pseudonym correlation clearly depends on the number of nodes that the attacker controls and not so much on the frequency of pseudonym change. We estimate thus, that to enable location private communication, it is sufficient that the frequency of pseudonym change needs to be only two times higher then the average frequency at which the nodes displace from the zone controlled by an attacker to the mix zone. We can roughly estimate that this frequency is around $\frac{1}{t(r)}$, where $t(r)$ is the average time that it takes a user to cross the distance equivalent to the power range.

6.4 Security of routing

As already elaborated in our routing protocol description, we use a number of cryptographic primitives to ensure the secure and correct operation of our PPR protocol. Here we analyze how resistant is the protocol to various attacks.

False distance information dissemination attack can be mounted by an attacker that tries to insert false distance information into the network, by claiming that it is closer or further from the access point than it really is. Due to the mutual network membership verification between nodes, this attack cannot be performed by an Attacker-0-M. If the attack is mounted by an Attacker-1-M, the misbehaving node will be easily detected, as its distance information will not match the distance information that the attacked nodes receive from their other neighbors. This attack can be successful only if the fraction of compromised nodes in the network is sufficiently large to fake the whole network topology without being detected by the access point.

A similar attack is the *black hole attack.* In this attack an attacker advertises a close distance to the access point, gathers the traffic from other nodes and drops the packets. This attack is similar to the false distance dissemination attack, and can be detected in a similar way, and can also be prevented if the nodes randomize their choice of the next hop node in the uplink. Unlike in mobile ad hoc networks, in hybrid ad hoc networks the black hole attack cannot paralyze the whole network, as the attacker can affect only a fraction of its neighboring nodes, until it is detected.

Another attack related to network topology is the *wormhole attack*, in which an attacker tunnels the received packets and retransmits them in the remote part of the network. This attack can threaten the safety of routing and can disturb nodes' distances to the access point. Similarly to the black hole attack, the wormhole attack can be resolved by access point topology control, or by temporal packet leashes proposed against wormholes in ad hoc networks [15].

Besides topology-related attacks, attackers can try a number of other attacks. An attacker can try to insert random packets into the network to *drain out nodes' battery power.* This attack could be mounted, provided that the attacker controls at least one compromised node (Attack-1-M). However, because the traffic in hybrid ad hoc networks is controlled by the access point, the access point can limit the use of network resources by each node, thus limiting its capability to drain the power of network nodes. Furthermore, unlike in ad hoc networks, in hybrid ad hoc networks, nodes do not need and are thus not allowed to issue any broadcast traffic, which prevents an attacker from creating *broadcast storms* to halt the network.

Other attacks can be envisioned against hybrid ad hoc networks; they are not addressed by this work. However, from a simple analysis it is clear that secure routing in hybrid ad hoc networks is much more resistant to attacks then routing in pure ad hoc networks. The main reason for this is the existence of an on-line authority (i.e., access points) capable of controlling traffic and monitoring node behavior.

6.5 Performance analysis

In this section, we briefly analyze the costs associated with our scheme. We first overview the cryptographic and then the communication costs.

In terms of cryptographic operations, to secure routing, the nodes use symmetric-key cryptography, whereas public key cryptography is only used for the dynamic key establishment. To maintain a dynamic change of keys, nodes periodically run authenticated key establishment with their neighbors. Therefore, the establishment of the secret key between two nodes typically requires the execution of one public key signature and one signature verification per node. An additional signature verification is required from each node to verify the certificate issued by the authority. As key updates between the nodes is performed rarely (every minute, or even every several minutes) the cryptographic cost of the dynamic key scheme is not high. This cost is fixed and does not change with changes in network traffic.

For each packet that it forwards, a node performs three symmetric-key operations: it computes a message MAC with a key that it shares with the preceding node, it re-encrypts the message and it computes a new message MAC with a key that it shares with the next node on the route.

If it would be necessary to perform key updates with symmetric-key cryptography only, public-keys could be replaced with TESLA [29] keys. However, this approach would require a slight modification of our protocols and we leave it for future work.

The communication cost of our scheme is, similarly to the cryptographic cost, comprised of two parts: the cost of the dynamic key update scheme and the cost of adding security and privacy to routing. The cost of the dynamic key update scheme is the cost of the update of the public keys that the nodes use to establish their shared secret keys. This cost depends on the update frequency, and for our application, this frequency is not high. It is thus sufficient that the access point sends one certificate to each node at the same frequency at which the keys and the pseudonyms are updated. This certificate update can be performed in various ways, and one possible optimization is if the access point sends the certificates to a node each time that it gets in the direct neighborhood of the access point. The communication cost of adding security and privacy to message forwarding is small; only a single MAC is added to each message on its way to and from the access point.

7. RELATED WORK

In this section, we discuss some existing research efforts related to hybrid ad hoc networks, secure routing, anonymity and location privacy.

Hybrid ad hoc networks: Bejerano [2] and Wu et al. [37] study the benefits of installing relaying stations (also called access points) to provide the nodes with an access to a backbone. In [26], Luo et al. present a unified cellular and ad-hoc network architecture that enhances control area throughput while maintaining fairness. In [25] Liu, Liu and Towsley presented a study of the capacity of hybrid wireless networks. Much work on hybrid ad hoc network has been also reported in the framework of TErrestrial Trunked RAdio (TETRA) project [39].

Secure Routing: So far, the problem of routing in ad hoc networks has been mainly studied in a non-adversarial setting, and only recently has the focus of research shifted to the design of secure routing protocols; researchers have already devised a number of proposals to secure both reactive (on-demand) and proactive routing protocols and identified a number of attacks [16, 19, 17, 15, 28, 32, 14], which we detail in Section 6.4.

Anonymity and Location privacy: A seminal work in the domain of anonymity was notably reported by Chaum in [8]. In [31], Reiter and Rubin present Crowds, a scheme that enables anonymity of web transactions. In [4], Berthold, Federrath and Kopsell propose Web MIXes, a system for anonymous and unobservable internet access. In [3], Beresford and Stajano address the problem of location privacy in pervasive computing. They propose a new construction, called a mix zone, a spatial regions in which a user is not registered for callback (i.e., does not report its location to other entities). The users then change their identities within these zones, so that when they leave the zone, their identity becomes different from the one they had when entering the zone. In [13], Gruteser and Grunwald propose an approach to enhance location privacy in wireless LANs based on disposable interface (MAC) identifiers. The Mist routing project [1] addresses the problem of routing a message to the user while keeping its location private. Mist operates by making use of a set of mist routers organized in a hierarchical structure that provides location privacy. In [35], Smailagic et al. present two location sensing systems and compare them to the existing location sensing proposals. They further perform a user privacy study and show that users expect two unique behaviors from the system: an introvert model, where privacy is preferred, and an extrovert model where availability is preferred. In [20], Jackson proposes a system that allows user control of the location information disclosure in systems like Active Badge [36]. An important work on IP private roaming has been reported in the framework of the Freedom Network [40, 5]. Several researches have also addressed the problem of preventing foreign operators from obtaining subscribers identities in GSM networks [22]. Several researchers designed privacy-enhancing location servers, for scenarios in which location data needs to be revealed to external users [12, 24]. Recently, Kong and Hong have proposed a protocol for anonymous communication in mobile ad hoc networks [23].

Anonymous credentials: As we already indicated in Section 4, other schemes can be used to fulfill a similar purpose as our dynamic key scheme. One of the best examples of such scheme are anonymous credentials proposed by Camenisch and Lysyanskaya [7].

8. CONCLUSION

In this paper we have proposed a scheme to secure and protect the privacy of communication in hybrid ad hoc networks. We have shown that both security and privacy preservation can be easily integrated in the same protocol; we have explained how we managed to reach a high level of privacy preservation by using pseudonyms and dynamic keys renewal. We have provided a detailed description of the Privacy Preserving Routing protocol, as well as an evaluation of the overhead and of the robustness of our scheme.

To the best of our knowledge, this is the first paper addressing these issues.

In terms of future work, we intend to refine the quantification of the degree of anonymity and location privacy in various scenarios and to perform a detailed simulation study of our solution.

9. ACKNOWLEDGMENTS

The authors are thankful to Ari Juels for useful comments and discussions.

10. REFERENCES

[1] J. Al-Muhtadi, R. Campbell, A. Kapadia, D. Mickunas, and S. Yi. Routing Through the Mist: Privacy Preserving Communication in Ubiquitous Computing Environments. In *Proceedings of the International Conference of Distributed Computing Systems (ICDCS)*, 2002.

[2] Y. Bejerano. Efficient Integration of Multi-Hop Wireless and Wired Networks with QoS Constraints. In *Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking*, pages 215–226. ACM Press, 2002.

[3] A. R. Beresford and F. Stajano. Location Privacy in Pervasive Computing. *Pervasive Computing*, January-March 2003.

[4] O. Berthold, H. Federrath, and S. Köpsell. Web MIXes: A system for anonymous and unobservable Internet access. In H. Federrath, editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, pages 115–129. Springer-Verlag, LNCS 2009, July 2000.

[5] P. Boucher, I. Goldberg, and A. Shostack. Freedom System 2.0 Architecture. *Zero-Knowledge Systems Inc. white paper*, December 2000.

[6] S. Brands and D. Chaum. Distance-bounding protocols (extended abstract). In *Theory and Application of Cryptographic Techniques*, pages 344–359, 1993.

[7] J. Camenisch and A. Lysyanskaya. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In *Proceedings of Crypto*, 2002.

[8] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2), February 1981.

[9] B. Deb, S. Bhatnagar, and B. Nath. A topology discovery algorithm for sensor networks with applications to network management. In *Proceedings of the IEEE CAS Workshop*, September 2002.

[10] Y. Desmedt. Major security problems with the 'unforgeable' (feige)-fiat-shamir proofs of identity and how to overcome them. In *SecuriCom'88*, 1988.

[11] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. In *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002)*, 2002.

[12] M. Gruteser and D. Grunwald. A Methodological Assessment of Location Privacy Risks in Wireless Hotspot Networks. In *Proceedings of the First International Conference on Security in Pervasive Computing*, 2002.

[13] M. Gruteser and D. Grunwald. Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis. In *Proceedings of WMASH*, 2003.

[14] M. Guerrero Zapata and N. Asokan. Securing Ad Hoc Routing Protocols. In *Proceedings of the ACM Workshop on Wireless Security (WiSe)* , 2002.

[15] Y.-C. Hu, A. Perrig, and D.B. Johnson. Packet leashes: A defense against wormhole attacks in wireless networks. In *Proceedings of IEEE Infocom*, April 2003.

[16] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. In *Proceedings of MobiCom*, September 2002.

[17] Y.-C. Hu, A. Perrig, and D. B. Johnson. Efficient Security Mechanisms for Routing Protocols. In *Proceedings of NDSS*, February 2003.

[18] Y.-C. Hu, A. Perrig, and D.B. Johnson. Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols. In *Proceedings of the ACM Workshop on Wireless Security (WiSe)*, September 2003.

[19] Y.-C. Hu, D. B. Johnson, and A. Perrig. Secure efficient distance vector routing in mobile wireless ad hoc networks. In *Proceedings of (WMCSA)*, June 2002.

[20] I. W. Jackson. Anonymous Addresses and Confidentiality of Location. In *Proceedings of International Workshop on Information Hiding*, 1996.

[21] D. Kesdogan, J. Egner, and R. Buschkes. Stop-and-go-mixes providing probabilistic anonymity in an open system. In *Proceedings of the International Information Hiding Workshop*, March 1998.

[22] D Kesdogan, H Federrath, A Jerichow, and A Pfitzmann. Location management strategies increasing privacy in mobile communication. In *12th International Information Security Conference*, pages 39–48, Samos, Greece, 21–24 1996. Chapman & Hall.

[23] J. Kong and X. Hong. ANODR: ANonymous On Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks. In *Proceedings of MobiHoc*, 2003.

[24] U. Leonhardt and J. Magee. Security Considerations for a Distributed Location Service. *Journal of Network and System Management*, 6(1), March 1998.

[25] B. Liu, Z. Liu, and D.F. Towsley:. On the Capacity of Hybrid Wireless Networks. In *Proceedings of Infocom*, 2003.

[26] H. Luo, R. Ramjee, P. Sinha, L. Li, and S. Lu. UCAN: A Unified Cellular and Ad-Hoc Network Architecture. In *Proceedings of MobiHoc*, 2003.

[27] S. Micali. Efficient certificate revocation. Technical Report MIT/LCS/TM-542b, 1996.

[28] P. Papadimitratos and Z.J. Haas. Secure Routing for Mobile Ad Hoc Networks. In *Proceedings of CNDS*, January 2002.

[29] A. Perrig, R. Canetti, J.D. Tygar, and D. Song. The TESLA Broadcast Authentication Protocol. *RSA CryptoBytes*, 5(Summer), 2002.

[30] A. Pfitzmann and M. Kohntopp. Anonymity, unobservability, and pseudonymity – a proposal for terminology. In *Proceedings of International Workshop on Design Issues in Anonymity and Unobservability*, 2000.

[31] M. Reiter and A. Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1), June 1998.

[32] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer. A Secure Routing Protocol for Ad hoc Networks. In *International Conference on Network Protocols (ICNP)*, 2002.

[33] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *Privacy Enhancing Technologies (PET)*, 2002.

[34] D. Shaw and W. Kinsner. Multifractal modeling of radio transmitter transients for clasification. In *Proceedings of the Conference on Communications, Power and Computing*, 1997.

[35] A. Smailagic, D. P. Siewiorek, J. Anhalt, D. Kogan, and Y. Wang. Location Sensing and Privacy in a Context Aware Computing Environment. *Pervasive Computing*, 2001.

[36] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, January 1992.

[37] H. Wu, C. Qios, S. De, and O. Tonguz. Integrated Cellular and Ad Hoc Relaying Systems: iCAR. *IEEE Journal on Selected Areas in Communications*, 19(10), October 2001.

[38] http://www.privacyinternational.org.

[39] http://www.tetramou.com.

[40] http://www.zeroknowledge.com.