

**CERIAS Tech Report 2004-60**

**JANUS: TOWARDS ROBUST AND MALICIOUS RESILIENT ROUTING IN HYBRID WIRELESS NETWORKS**

by B. Carbunar, I. Ioannidis, C. Nita-Rotaru

Center for Education and Research in  
Information Assurance and Security,  
Purdue University, West Lafayette, IN 47907-2086

# JANUS: Towards Robust and Malicious Resilient Routing in Hybrid Wireless Networks

Bogdan Carbutar, Ioannis Ioannidis and Cristina Nita-Rotaru  
Department of Computer Sciences  
Purdue University  
250 N. University Street  
West Lafayette, IN 47907-2066  
{carbunar, ioannis, crsn}@cs.purdue.edu  
FAX: (765) 496-3181

## ABSTRACT

In this paper we investigate and provide solutions for security threats in the context of hybrid networks consisting of a cellular base station and mobile devices equipped with dual cellular and ad-hoc (802.11b) cards. The cellular connection is used for receiving services (i.e. Internet access) from the base station, while the ad-hoc links are used to improve the quality of the connection. We provide detailed descriptions of several attacks that arbitrarily powerful adversaries, whether outsiders or insiders, can mount against well-behaved members of the network. We introduce a secure routing protocol called JANUS, that focuses on the establishment of secure routes between the base station and mobile devices, and the secure routing of the data. We show that our protocol is secure against the attacks described and experimentally compare the message overhead introduced by JANUS and UCAN.

## Categories and Subject Descriptors

C.2.0 [General]: Security and protection; C.2.1 [Network Architecture and Design]: Wireless communication; C.2.2 [Network Protocols]: Routing protocols

## General Terms

Algorithms, Design, Reliability, Security, Theory

## 1. INTRODUCTION

The generous promise of the 3G cellular networks is to provide a unified framework where users can move seamlessly between cellular networks anywhere in the world. 3G services have been provided in Japan by the wireless company NTT DoCoMo for more than two years. More recently, similar services started being offered in the United Kingdom and there are indications that 3G services will be also offered in US by the end of 2004. 3G services are designed to offer broadband cellular access at speeds of 2Mbps (as

opposed to the 9.6Kbps rate currently provided by 2G networks), which allow mobile multimedia services to become possible. The Internet becomes thus available to numerous clients using laptops or PDAs, via special network cards, or mobile phones enabled to act as wireless modems.

In spite of its convenience, this service has the drawback that the rate decreases dramatically as destination clients move towards the outskirts of the base station's coverage area. A solution proposed to overcome the problem is to use a network consisting of both cellular and WLAN devices. Laptops located in the proximity of the base station can act as relayers for less fortunate devices, allowing them to obtain an increased bandwidth. All devices are equipped with two wireless interfaces. The routing protocols are aware of both interfaces and use them as appropriate to improve the performance of data delivery to client devices.

We refer to such a network consisting of both cellular and WLAN components as a *hybrid wireless network*. We note that the new proposed framework determines a significant change in the trust and communication model previously existing in the network. In a regular cellular network, the base station requires authentication of all clients. Once authenticated the clients trust the base station and the traffic flows directly between the base station and the client host without the help of any intermediary.

In the case of the hybrid network, the trust is no longer centralized. A client uses several, possibly unknown, devices as relayers. In turn, the relayers have to trust each other to forward the actual data. A client can benefit from the use of the hybrid network only if the last relayer in the path, before the base station, has a higher cellular rate, and all the relayers behave correctly. Misbehaving devices forwarding data can act either in a selfish or malicious manner. The intentions of a "selfish" device [22] are usually to obtain service without reciprocating. In contrast, the goal of a malicious device is to disrupt the communication of other devices in the network, without regard for its own resource consumption. Such behavior can waste the time and resources of multiple participants, making them reluctant to use the service and choose the direct cellular link instead.

Prior work in security aspects of hybrid wireless networks focused mostly on the selfish node problem [23, 31] or anonymity and privacy preserving [8]. No work, to the best of our knowledge, was conducted in addressing the arbitrary malicious node problem in hybrid wireless networks. However, the effect of attacks coming from nodes that refuse to act according to the protocol can be devastating, particularly when attackers are strategically placed.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WiSe'04, October 1, 2004, Philadelphia, Pennsylvania, USA.  
Copyright 2004 ACM 1-58113-925-X/04/0010 ...\$5.00.

## 1.1 Our Focus

The goals of this paper are to identify the types of attacks that malicious hosts can perform in a hybrid wireless network and provide efficient detection and further avoidance of the malicious parties, as well as an evaluation of the associated costs. Some of the security aspects that we will describe were addressed in the context of ad hoc networks [3, 28, 15, 16, 32, 25]. However, such schemes focus many times on achieving security goals for specific routing protocols, or have significant overhead because of the completely decentralized nature of the ad hoc network. We also note that such protocols often use a routing selection criterion that is different from the one appropriate in hybrid wireless networks (i.e. finding the highest throughput path), making unsuitable their direct application.

The existence of channels with different levels of trust and performance, and of a point of total trust (which is natural to the system and not an unwarranted assumption) can lead to improved schemes. Our approach exploits the hybrid nature of the network to minimize the cost of the security mechanisms, while making very few assumptions about the behavior of participants. More specifically, our contributions are:

- We identify attacks possible in a hybrid wireless network. We make no assumption about the attacker, but rather assume arbitrary behavior.
- We propose JANUS<sup>1</sup>, a new routing algorithm robust to malicious attacks. Our solution focuses on the protection of the routing mechanism itself, as well as the routing of data, and exploits the hybrid trust model. When appropriate, we take advantage of the low-bandwidth direct communication link that each host has with the base station to transmit critical information and identify problematic links.
- We provide a security analysis of the proposed scheme. We experimentally compare our protocol with UCAN, in terms of the cellular and ad-hoc traffic generated. Our simulations show that JANUS generates up to 4 times less cellular traffic and up to 60% less ad-hoc traffic than UCAN, in order to maintain up to 30 concurrent flows for up to 500 mobile hosts.

The remainder of the paper is organized as follows. We provide an overview of related work in Section 2. We define the network and security models we assume in Section 3. Our robust algorithm is presented in Section 4 and its security is analyzed in Section 5. We present the experimental comparison of JANUS and UCAN in Section 6. Finally, conclusions and future work are presented in Section 7.

## 2. RELATED WORK

Although new, the hybrid wireless network field has been quite active, particularly in designing efficient routing mechanisms. In this section we overview previous research conducted in two areas related with our work: routing and addressing security concerns.

### 2.1 Routing in Hybrid Wireless Networks

<sup>1</sup>Janus, the god who was guardian of portals and patron of beginnings and endings had his head always shown with two faces: one in the front and one at the back. As Janus, all the hosts in the network have two "faces", a cellular communication link and an 802.11b communication link.

UCAN [23] provides two routing algorithms that allow client hosts to find hosts that are willing to forward their traffic. The first algorithm greedily propagates the routing request to a single neighbor, based on the best rate advertised by its neighbors. The second algorithm is based on a restricted depth flooding to find the best proxy host. However, since both algorithms ignore the finite capacities of the ad-hoc links, they do not find the highest throughput path.

Fujiwara, Iida and Watanabe [13] provide a unicast routing algorithm that allows mobile hosts to find an alternate path to the base station, when the cellular connection fails. The focus of the work is on reestablishing connectivity to the base station via short paths, in response to emergency situations where direct connections to the base station may become sparse. This is a different goal from JANUS, which attempts to optimize throughput under normal conditions, where all hosts have a direct connection to the base station via the cellular interface.

### 2.2 Security in Hybrid Wireless Networks

Securing hybrid wireless networks is an area that was not intensively explored to date. By using a mechanism based on incentives, UCAN [23] provides only fragile protection against selfish nodes. UCAN's focus is on preventing individual hosts from deleting legitimate hosts or adding non-authorized hosts to the set of relayers that receive credit for forwarding data. The mechanism is entirely defenseless when faced with collusion. The work in [31] provides incentives for collaboration in a network where mobile hosts are covered by several access points. The main assumption of the paper is that the hosts are selfish, therefore only trying to gain undesired advantages. Moreover, the solution provided does not describe the underlying routing algorithm, but only assumes that there is one and that it is secure. Finally, the work in [8] focuses on preserving the anonymity and privacy of mobile hosts in a network covered by several access points.

The problem of defining compelling methods to make nodes participate in forwarding data was also addressed in the context of traditional ad hoc wireless networks [22] and resulted in designing protocols that provide fair access to the medium. Also the peer-to-peer (P2P) community conducted research for a similar problem, most of the solutions focusing on designing incentive mechanisms [34, 24, 2, 9] to motivate peers to participate in providing storage resources or in data forwarding.

In the context of ad hoc wireless network, research was conducted mainly on securing routing protocols and designing key management protocols. Problems that were addressed include traditional attacks such as impersonation and replay relying on secure association, symmetric-key based cryptographic mechanisms, or digital signatures [28, 15, 16]. More sophisticated attacks such as wormhole [17], flood-rushing [18], or arbitrary Byzantine behavior [3] were also considered. We note that in most of the cases, the work focused on securing specific protocols, such as DSDV [29], DSR [21] or AODV [30].

The cellular network protocols and standards also include a security component. The core services provided by any of the cellular communication protocols (such as GSM [6], GPRS [1] or UMTS [27]) are authentication of subscribers (clients), providing subscriber identity confidentially, and confidentiality and integrity of both data communication and radio signaling. Clients are authenticated using a challenge-response protocol, based on a key and using algorithms stored on the Subscriber Identity Module (SIM) smart card, obtained when the client subscribes to the service.

### 3. SYSTEM MODEL

In this section we present the model we assume for this paper. We first describe the network model, and then show what set of actions defines an adversarial behavior.

#### 3.1 Network Model

The network consists of a cellular base station (BS) and several mobile hosts (MHs). All the mobile hosts are inside the cellular coverage range of the base station, and each host has an account established with BS.

The base station has access to the wired Internet, and the mobile hosts download information via their cellular connection. Thus, most of the traffic is issued by the base station. In addition, mobile hosts can communicate with other hosts in range, using 802.11b wireless cards. We use a multi-channel MAC scheme [33] requiring only one half-duplex transceiver per host. We assume bi-directional communication on all the 802.11b links in the network. This is also required by most wireless MAC protocols, including 802.11, to operate correctly. The following table provides the terminology used in this paper.

Name	Definition
forward link	cellular link from BS to a MH
reverse link	cellular link from a MH to BS
ad-hoc link	802.11b link between two MHs
bandwidth	capacity of any wireless link
throughput	capacity of a path from BS to an MH

We assume that each mobile host is equipped with a GPS receiver that provides reliable positioning. In addition, we assume that the base station and the mobile hosts have synchronized clocks. Since GPS already provides a clock synchronization method with an error of up to 340 ns [10], we could use it as a synchronization mechanism between the base station and the mobile hosts.

#### 3.2 Security Assumptions

The base station is trusted by each host. The base station authenticates every client and establishes a secure authenticated channel with the client. As shown in Section 2.2 most of the cellular communication protocols provide such a service. Attacks at the physical layer are not considered in this paper.

Hosts that can not be authenticated by the base station do not participate in the protocol and are not trusted. Any intermediate host on the path between the base station and the destination client can be authenticated, but may exhibit arbitrary (Byzantine) behavior. Attackers will try to inflict as much harm as possible on the other hosts in the network, without consideration for their own resources. We assume that an intermediate host can exhibit such behavior either alone or in collusion with other hosts.

We focus on providing a secure routing protocol, which specifically addresses threats to the network layer in the ISO/OSI model. We do not address attacks against lower layers in the ISO/OSI model. We note that the physical layer can also be disrupted by jamming, and MAC protocols such as 802.11 can be disrupted by attacks that deny access to clients and allow a potential misbehaving node to take over the channel [22, 7]. We do not address such attacks in this work.

A public-key infrastructure is required for operations such as signature generation and verification, and shared key establishment. This infrastructure can be based on certificates, where the base station plays the role of Certificate Authority (CA), and each host knows the base station's public key.

### 4. OUR SOLUTION

In this section we present JANUS, our routing algorithm, that identifies links over which attackers operate. The algorithm also provides information about the detected link to the parties affected, allowing them to locally take the necessary countermeasures. We first provide a high-level description of an un-secure routing algorithm that uses as routing criterion the highest throughput to the base station. We then identify the type of attacks that can be mounted in the different phases of the protocol and describe in details our security mechanisms.

#### 4.1 Algorithm Overview

The core mechanism of our routing algorithm is to select for each host, a path providing the highest throughput from the base station. This is achieved by each host periodically probing its neighbors for their current throughput and selecting the one providing the highest value. Such a host is called the *parent*. The period of the neighbor probing is called *refresh rate*. The overall information maintained by the hosts can be viewed as a tree encoding the best throughput from the base station to any host. Note that in reality, this is an approximation of the best throughput since hosts move, thus changing the throughput. A small period refresh of the tree will provide a good approximation, but might incur higher cost. We call this tree the *routing tree*.

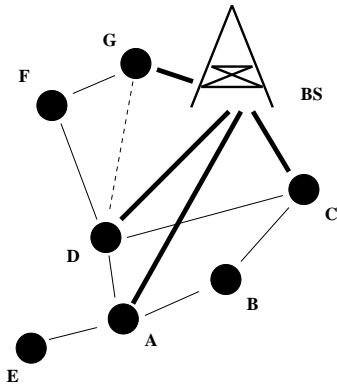
In order to compute its current throughput, a host may need to traverse  $O(n)$  hosts up to the base station,  $n$  being the total number of hosts. For small  $n$ , this is not a problem, but for dense, metropolitan areas, the number of messages produced can aggravate congestion. To maintain the routing tree with  $O(\log n)$  number of messages per update operation, we assume that a supporting *topology tree* [12] is used. The constructions and operations associated with the topology tree are described in Section 4.3

The routing and topology tree construction represents the proactive part of the protocol. Due to host mobility, the trees require permanent maintenance. The protocol has also a reactive part, initiated when a host needs to download information. The client host then contacts its parent in the routing tree, that in turn will contact its parent and so on, until the message reaches the base station. This phase, called *path reservation*, requires each host contacted to locally verify the availability of the resources requested by the client. Such a host then appends its identity to the message received from its child and forwards it to its parent. At the completion of this phase, the base station knows the client host name, the information requested by it, the path to that client, and the bandwidth available on that path. It then sends the information requested along the path, at the available rate. The protocol can be roughly broken into the following components.

- constructing the routing tree
- constructing the topology tree
- periodically refreshing the routing tree and topology tree
- performing the path reservation protocol
- forwarding the data

#### 4.2 Security Goals

In this section we present details of several types of attacks that organized malicious hosts can perform against well behaved hosts.



**Figure 1: Rate Inflater Attack illustration.** The thicker lines represent cellular links and the plain lines ad-hoc links. The dashed line represents an inexistent link, called a tunnel. The first number represents the advertised link bandwidth and the second one represents the correct value.

*Impersonation.* A host can try to impersonate other hosts in order to unfairly obtain free services or place blame for malicious actions on other hosts. Also a node can try to impersonate the base station to create havoc in the network.

*Rate Inflation.* A host can advertise a larger throughput than it can provide. This can be done by inflating the bandwidth of its forward link or of its ad-hoc links. A host M that lies about its throughput will be able to negatively affect not only adjacent hosts, but also hosts that have M as an ancestor in the routing tree.

For example, in Figure 1, host A should choose B as parent, since it provides the largest throughput to BS. However, if host D advertises an increased forward link bandwidth or an inflated bandwidth for its ad-hoc link to C, A will end up choosing D as parent instead, effectively achieving a lower throughput. This attack will negatively affect host E as well, since A is its parent.

*Tunneling.* Two non-adjacent hosts can collude by advertising an excellent bandwidth for the link between them. Their communication can be encrypted and sent through a path of mobile hosts established ad hoc, or simply through the base station. This attack can be viewed as a particular case of the rate inflation attack, but also as an instance of a *wormhole* [17] attack.

For example, in Figure 1, host D could collude with host G, and advertise a high rate for an inexistent link between D and G. Their agreement could be perfected either through F or through BS. Note that F does not need to cooperate, since the agreement can be encrypted.

*Denial of Service.* Since each host has a direct cellular connection with the base station, any number of malicious colluding devices cannot prevent other hosts from receiving cellular service. Malicious hosts could however generate arbitrary amounts of useless traffic in the network, by generating requests for data that they do not intend to use, wasting network resources.

*Path Scrambling.* The last phase of JANUS requires the base station to know the entire routing tree path between a client host and the base station. Since that knowledge is obtained by requiring intermediate hosts on that path to append their names to a message, malicious intermediate hosts can not only add and remove hosts

from that path, but also change the order of hosts. This attack will cause packets to be lost, with the additional problem of allowing the attacker to frame other hosts. In addition, by manipulating the path, a host can make sure he is selected on a particular path and later on use that to his advantage.

*Black Hole.* A malicious relay can selectively drop packets received, instead of forwarding them towards their intended destination. This attack can potentially drop the throughput of a host to zero.

### 4.3 Secure Tree Construction and Maintenance

Our routing algorithm distributively builds a spanning tree, called *routing tree*, to provide hosts with the highest throughput from the base station. The mobility and intermittent connectivity of the hosts impose adaptive requirements on any routing algorithm. In this section we define the operations used for maintaining the routing tree and discuss their secure implementation.

The set of maintenance operations consists of *Cut*, *Link*, *Update* and *Mincost*. *Cut* splits a tree into two subtrees by removing an edge between two vertices. *Link* joins two subtrees by adding an edge between two vertices, each in a different subtree. *Mincost* returns the weight of the minimum weight edge on the path from a vertex to the root of the tree, which in our case is the base station. Finally, *Update* adds or removes  $w$  from the weight of each edge on the path from a vertex to the root of the tree. A simple, low overhead implementation of these operations with a  $O(n)$  time, where  $n$  is the number of vertices, or mobile hosts, per update operation may be preferable when the size of the network is small. In this section, we discuss how we can implement the above operations using topology trees, with a  $O(\log n)$  time per operation.

Topology trees are an instance of link-cut trees, which support a superset of the above four operations for maintaining rooted trees in  $O(\log n)$  time per update operation. The size of the topology tree is  $O(n)$  and its height  $O(n)$ . We assume that the base station stores and maintains the topology tree. Since cellular base stations have to keep information for every host that is logged in their cell, storing the topology tree does not impose an unrealistic overhead. Effectively, the base station acts as an oracle that answers queries and update requests from the hosts. The use of topology trees guarantees that such an oracle is efficient, but in principle any implementation of the four operations can serve as oracle. We note that even though the base station is a single point of failure, the inherent loss of the topology tree due to the base station's failure is not an additional problem. This is because the topology tree is used to provide hosts with easy to maintain, high throughput paths *from* the base station.

*Link Weight Assignment.* The routing tree is weighted, and the weight of each edge is computed as a function of the bandwidth of the physical link between the hosts that are the edge's endpoints. There are two types of edges in the routing tree, cellular forward links and symmetric ad-hoc links. The bandwidth evaluation of forward links can be done by the base station, using a technique similar to the one presented in [11]. More specifically, the base station periodically sends link probes of fixed size and the end-host counts the probes received in a certain time interval. The link's delivery ratio is then computed, allowing the host to measure the link's bandwidth. We assume that on each mobile host, the MAC layer can provide the routing layer with information about the bandwidth of ad-hoc links.

In order to be able to find for each host the highest throughput path from the base station, we need to be able to compute the aggregated throughput of a path. As shown in [4, 11], the knowledge

of the bandwidth of all the links along a 802.11b multi-hop path, is not enough to compute the aggregate throughput of the path. Due to interference between successive hops, the classic method of wired networks of evaluating the throughput of a path as the bandwidth of the path’s bottleneck link no longer works. We solve this problem by using a multi-channel MAC protocol [33, 26, 35, 19]. We use the scheme presented in [33], since it does not require additional hardware, needing only one half-duplex transceiver. This allows hosts that are 2-hops away on the same path, to transmit simultaneously, by choosing different channels for transmission. Since the half-duplex transceiver does not allow a host to receive and send simultaneously, we use the aggregate throughput of 2-hop paths presented in Table IV in [4] to evaluate path throughputs. Thus, our solution consist in breaking a path into 2-hop subpaths, such as the aggregate throughput of the whole path is the throughput of the path’s bottleneck 2-hop subpath.

In the following, we describe in more detail each of the topology tree operations, in the context of the base station acting as an oracle, and discuss their secure deployment for JANUS.

*Cut.* The  $\text{cut}(v)$  operation splits a routing tree by removing the edge between host  $v$  and its parent. Whenever a host needs to change the parent, either due to the parent’s failure or the discovery of a better placed neighbor, the host contacts the base station to request it to update the routing tree by cutting the edge to its parent. The request is done via a message encrypted using the key shared with the base station. The purpose of encryption is to prevent other hosts from cutting edges arbitrarily in the routing tree. The  $\text{cut}$  operation is not otherwise a hazardous operation from a security standpoint, since a host has the liberty of choosing any parent.

*Mincost.* The  $\text{mincost}(v)$  operation returns the minimum weight of an edge in the routing tree, on the path between host  $v$  and the base station. Whenever a mobile host needs to choose a parent in the routing tree, it queries its ad-hoc neighbors. Since the oracle is assumed to reside in the base station, we have two choices for the deployment of  $\text{mincost}$ . In the first solution, whenever a host needs to find a new parent it contacts the base station, providing the list of neighbors and the throughput of the corresponding edges. The base station retrieves the  $\text{mincost}$  of each neighbor and returns the identity of the one providing the client host with the highest throughput. In the second solution, the base station periodically sends each host a signed and timestamped certificate containing the host’s  $\text{mincost}$  value. Whenever a host needs a new parent, it collects the certificates of all its neighbors, checks their validity and freshness and makes its own local decision.

*Link.* The  $\text{link}(u, v, w)$  operation merges the routing tree rooted at host  $u$  with the routing tree of host  $v$ , by making  $v$  the parent of  $u$ . The bandwidth of the added edge is  $w$ . This operation is the reverse of  $\text{cut}$ , and it is used by a host to complete the procedure of changing the parent. We can now summarize the parent change operation to be a succession of  $\text{cut}$ ,  $\text{mincost}$  and  $\text{link}$  operations.

There are however security issues that need to be addressed. Malicious hosts can invent links that do not exist or provide overestimates of the bandwidth of existing links. To prevent this, the bandwidth evaluation of forward links can be secured by the base station, by requiring the mobile endpoint of the link to send back a message digest of the probes received in a time interval. The base station can then compute the delivery ratio and verify the link’s bandwidth.

The secure evaluation of the ad-hoc links is more difficult, since both endpoints can be malicious and colluding. If the base sta-

tion relies only on the bandwidth evaluated and provided by the *remote* mobile hosts, there is no way of deciding if the information is correct. To solve this issue, we use the verifiable position information [14], that mobile hosts can provide, see Section 3.1. For this, the endpoints of an ad-hoc link first locally evaluate the bandwidth of the link, in a way similar to the base station evaluation of forward links. Then, along with the `link` parameters, each of the endpoint hosts sends to the base station its position. Using a technique similar to `Sat-Range` [14], the base station verifies the accuracy of the position reported by the endpoints. If the positions are accurate, the base station evaluates the bandwidth of the ad-hoc link, based on distance, and performs the `link` operation using the minimum between the evaluated value and the value reported by the endpoints.

*Update.* The  $\text{update}(v, w)$  operation adds the value  $w$  to the bandwidth of all the links on the routing tree path from host  $v$  to the base station. This operation is used whenever a host needs to reserve or release a path of relayers used for downloading information from the base station.

Using the `update` operation, a malicious host could try to add arbitrary values to the bandwidth of the links of its ancestors in the routing tree. However, this operation can be easily supervised by the base station, in the following way. When a host needs to download information, the base station first retrieves the minimum bandwidth of a link on the path from the host to the base station, see Section 4.4. The BS then performs the `update` operation with the negative of that minimum bandwidth value as  $w$ , in order to reserve resources along that path. When a download is completed, the base station performs an `update` operation using the flow’s transmission rate, as the positive  $w$  value, in order to release resources on the path from the client host to the base station. The operation can therefore produce no harm, since the base station decides the  $w$  value of each  $\text{update}(v, w)$  operation. Moreover, an `update` with a positive  $w$  value can only be performed after an `update` with a negative  $w$  value, and the absolute  $w$  values need to coincide.

## 4.4 Secure Path Reservation

The reactive part of JANUS takes place when a mobile host,  $A$ , needs to download information from the Internet, via the base station. In such a situation, all the intermediate hosts between  $A$ ’s parent and the base station need to be notified of the decision. The intermediaries are given the opportunity to refuse participation or to confirm their available resources. We call this process the *path reservation* phase. The throughput value maintained by  $A$ ,  $\text{rate}_A$ , can be inaccurate, due to large values of the refresh rate of the topology tree and high network mobility. During the path reservation phase, intermediate hosts have the ability to adjust the inaccurate throughput values of their routing tree edges.

The path reservation phase has the additional purpose of providing the base station with the identities of all the intermediate hosts to  $A$ . A simple way to achieve this would be to require each notified host to contact the base station on the reverse cellular link and reveal its identity. We avoid this solution, due to the potentially large number of such contacts on a low bandwidth link.

Algorithms 1 and 2 present the pseudocode of our solution using an Orca [5] like syntax. Orca is a parallel programming language for distributed systems, that provides elegant constructions for expressing reactive behavior, such as *guards*. Operations can consist of one or more guards with syntax

**guard expression do statementSeq od**<sup>2</sup>.

<sup>2</sup>expression is a boolean expression and statementSeq is a se-

**Algorithm 1** The generic host's view of the path reservation and data forwarding phases. We use  $\text{pkt}[i]$  to extract the  $i$ th field of  $\text{pkt}$ .

---

```

1. Object implementation MobileHost;
2. BS : BaseStation;
3. inQ : InputQueue;
4. Id, ssn : integer;
5. rate, throlld : real;
6.  $K_{\text{shr}}$  : string; #key shared with BS
7.  $K_{\text{pub}}^{\text{BS}}$  : string; #BS's public key
8. Operation pathReserve()
9.   p := new String(INIT, Id, ssn ++, rate, throlld, fn);
10.  pkt := new Packet(INIT,  $E_{K_{\text{shr}}}(\text{p})$ );
11.  sendToBS(pkt);
12.  guard inQ.first.type = SGN do
13.    verify( $K_{\text{pub}}^{\text{BS}}$ , sgn := inQ.first[2]);
14.    p := new String(ADDF, Id, ssn, sgn, hmac( $K_{\text{shr}}$ , Id));
15.    pkt := new Packet(ADDF, p);
16.    sendToPar(pkt);
17.  od
18. end
19. Operation main()
20.  guard inQ.first.type = ADDF do
21.    pkt := inQ.first;
22.    sgn := verify( $K_{\text{pub}}^{\text{BS}}$ , pkt[4]);
23.    if correct(sgn, pkt) = false then
24.      p := new String(ERR, Id, Idk);
25.      sendToBS(new Packet(ERR, p));
26.    fi
27.    store(pkt);
28.    if cap(parent) < sgn[3] then
29.      p := new String(LOW, Id, parent, cap(parent));
30.      sendToBS(new Packet(LOW, p));
31.      guard inQ.first.type = SGN do
32.        pkt[4] := inQ.first[2];
33.      od
34.    fi
35.    append(pkt, Id);
36.    pkt[5] := hmac( $K_{\text{shr}}$ , pkt[5], Id);
37.    sendToPar(pkt);
38.  od
39.  guard inQ.first.type = FLOW do
40.    pkt := inQ.first;
41.    if pkt[2] = Id then
42.      if checkHmac(pkt) = true then
43.        h := hmac( $K_{\text{shr}}$ , ACK, Id, pkt[3] + 1, pkt[4]);
44.        sendToBS(new Packet(ACK, Id, h));
45.      else sendToHost(next(pkt[2]), pkt);
46.    fi
47. end

```

---

Host A initiates the path reservation phase with operation `pathReserve`, by contacting the base station through its reverse link, (lines 9-11), with a message of type INIT, containing A's identity, a session identifier  $\text{ssn}_A$ ,  $\text{rate}_A$ , a threshold value, and an identifier of the information needed from the base station,  $\text{fn}$ , all encrypted with the secret session key shared by A with the base station. The message structure is the following

$$\text{INIT}, \text{Id}_A, E_{K_{\text{shr}}}(\text{INIT}, \text{Id}_A, \text{ssn}_A, \text{rate}_A, \text{thr}_A, \text{fn}).$$

The values contained in the packet are the ones described in line 9, instantiated for host A. The threshold value  $\text{thr}_A$  is the smallest throughput that A considers useful, and must be therefore larger than A's forward link bandwidth. When the base station receives an

quence of statements. The operation containing guards blocks until one or more guards are true. Then one of those guards is randomly chosen and its statements are executed atomically.

**Algorithm 2** The base station's view of the path reservation and data forwarding phases.

---

```

48. Object implementation BaseStation;
49. Operation main()
50.  guard inQ.first.type = INIT do
51.    intId := inQ.first[2];
52.    pkt := decrypt(inQ.first[3],  $K_{\text{Id}}$ );
53.    store(pkt);
54.    pkt := sign(Id, pkt[2], pkt[3],  $K_{\text{priv}}^{\text{BS}}$ );
55.    sendToHost(Id, new Packet(SGN, pkt));
56.  od
57.  guard inQ.first.type = LOW do
58.    pkt := inQ.first;
59.    p := signNewRate(pkt);
60.    sendToHost(pkt[2], new Packet(SGN, p));
61.  od
62.  guard inQ.first.type = ADDF do
63.    pkt := inQ.first;
64.    if checkHmac(pkt) = true then
65.      dest := pkt[pkt.size];
66.      Id := pkt[2]; #pkt[2] is client
67.      String info :=  $E_{K_{\text{shr}}}^{\text{Id}_A}(\text{retrieve}(\text{fn}_{\text{Id}_A}))$ ;
68.      break(info, n);
69.      for i := 1 to n do
70.        h := hmac( $K_{\text{shr}}^{\text{Id}}$ , FLOW, Id, i, info[i]);
71.        p := new String(FLOW, Id, i, info[i], h);
72.        sendToHost(dest, new Packet(FLOW, p));
73.      od
74.    else detectFaultyLink
75.    fi
76.  od
77. end

```

---

INIT message (line 50), it responds by sending A a signed message  $\text{SGN}, E_{K_{\text{priv}}^{\text{BS}}}(\text{Id}_A, \text{ssn}_A, \text{rate}_A)$ . When A receives the message (line 12), it verifies the signature and contacts its parent, with a message of type ADDF with the following structure

$$\text{ADDF}, \text{Id}_A, \text{ssn}_A, E_{K_{\text{priv}}^{\text{BS}}}(\text{ADDF}, \text{Id}_A, \text{ssn}_A, \text{rate}_A), \text{HMAC}_{K_A}(\text{Id}_A).$$

When a host N receives an ADDF message (line 20),  $\text{ADDF}, \text{Id}_A, \text{ssn}_A, \text{HMAC}_k, E_{K_{\text{priv}}^{\text{BS}}}(\text{ADDF}, \text{Id}_A, \text{ssn}_A, \text{rate}_A)$ ,  $\text{Id}_1, \dots, \text{Id}_k$ , where  $\text{Id}_A, \text{Id}_1, \dots, \text{Id}_k$  are the identities of all the intermediate hosts from A to N and  $\text{HMAC}_k$  is an onion HMAC of all these hosts as reported by  $\text{Id}_k$ , it first checks the signature of BS (line 22). N then checks that the last host in the path received in the ADDF message,  $\text{Id}_k$ , is its neighbor and child in the spanning tree (line 23). If the check fails, the host contacts the base station with an ERR message containing its identity and the identity of  $\text{Id}_k$ . Otherwise, if a message from A was never received by N, N stores a record for A (line 27), with the following format

$$[\text{Id}_A, \text{ssn}_A, \text{HMAC}_k, \text{Id}_k],$$

where  $\text{HMAC}_k$  is the HMAC received.

If both checks succeed, N checks the capacity of its own link to its parent,  $\text{cap}(N, P)$ , (line 28). If  $\text{cap}(N, P) < \text{rate}_A$ , it contacts BS, with a message of type LOW, containing its identity, the identity of its parent, and the value  $\text{cap}(N, P)$  (lines 29-30). The base station (lines 57-61) compares  $\text{cap}(N, P)$  with  $\text{thr}_A$ . If it is smaller, it contacts A that will look for an alternate path, or choose a different threshold value. If the value is larger than the threshold, it replaces  $\text{rate}_A$  with  $\text{cap}(N, P)$ . It then returns to N the SGN message  $\text{SGN}, E_{K_{\text{priv}}^{\text{BS}}}(\text{Id}_A, \text{ssn}_A, \text{cap}(N, P))$ . When N receives the SGN message (line 31), it replaces the signature in the ADDF packet with the one received in the SGN message (line 32). N then appends its identifier

to the packet (line 35), computes a new HMAC incorporating the received  $\text{HMAC}_k$  and its identity and key shared with the base station (line 36) and sends the new ADDF message to its parent (line 37). The sent message has the following format

$$\text{ADDF, Id}_A, \text{ssn}_A, E_{K_{\text{priv}}^{\text{BS}}}(\text{ADDF, Id}_A, \text{ssn}_A, \text{rate}_A), \\ \text{HMAC}_{K_N}(\text{HMAC}_k, \text{Id}_N), \text{Id}_{1..k}, \text{Id}_N$$

The path reservation process ends when the base station receives the ADDF message initiated by A (line 62). In the following section we describe the data forwarding phase.

## 4.5 Black Hole Detection

When the base station receives an ADDF message (line 62), BS checks the validity of the HMAC, against the identities and shared keys with the hosts in the path received (line 64). If the HMAC is valid, the base station retrieves the information requested in the INIT message and encrypts it with the symmetric key shared by BS with A (line 67). The encrypted information is broken into packets (line 68) and forwarded to the host whose identifier is the last in the ADDF message received by BS (lines 69-73), in messages of type FLOW

$$\text{FLOW, Id}_A, \text{PID, PKT}_i, \text{HMAC}_{K_A}(\text{FLOW, Id}_A, \text{PID, PKT}_i),$$

where PID is the packet identifier and  $\text{PKT}_i$  is the  $i^{\text{th}}$  packet of the flow.  $\text{PKT}_i$  is part of the encrypted requested information, thus maintaining the confidentiality of the information.

Each host that receives a packet of type FLOW, retrieves from its local database the record corresponding to  $\text{Id}_A$ , and forwards the packet on the link to the next hop associated with  $\text{Id}_A$  (line 45). For each packet received, A checks the HMAC and sends to BS on the reverse cellular link, an ACK packet (lines 42-44)

$$\text{ACK, Id}_A, \text{HMAC}_{K_A}(\text{ACK, Id}_A, \text{PID} + 1, \text{PKT}_i)$$

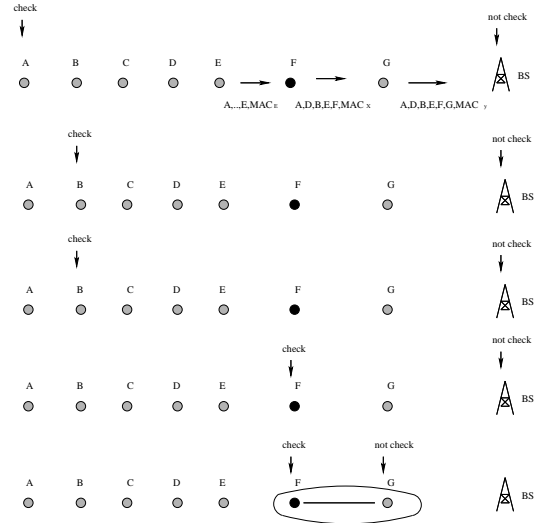
The purpose of the HMAC with an incremented PID is to authenticate A for BS.

During this phase, packets sent from the base station to a host can be dropped by malicious hosts trying to interrupt the data flow. Our defense against these attacks is based on acknowledgments and the insertion of probes [3]. Similar to [3], we use a threshold on the number of tolerable packet losses, and define a fault to be a packet loss higher than the threshold. Initially, as seen above, only the client host A needs to send an acknowledgment to the base station. The base station keeps track of the number of packet losses, as the number of acknowledgments not received during a certain window of packets. When the number of packets not acknowledged is higher than the acceptable threshold, the base station detects a fault, and initiates a faulty link discovery protocol.

The faulty link discovery protocol consists of selecting a number of intermediate hosts on the path from the base station to the client host, A, and requesting them to acknowledge future forwarded packets, sent by the base station to A. The hosts selected are called *probes*. The acknowledgment format is

$$\text{ACK, Id}_p, \text{HMAC}_{K_p}(\text{ACK, Id}_p, \text{PID} + 1, \text{PKT}_i),$$

where  $\text{Id}_p$  is the identity of the probe, and  $\text{PKT}_i$  and PID represent the packet acknowledged and its identifier. The selection of the probes models a binary search of the faulty link. The binary search views the path between the base station and the client A as an interval whose endpoints are BS and A. An interval whose right endpoint does not acknowledge a packet but whose left endpoint does, is said to be a faulty interval. When a faulty interval is detected, initially the BS to A interval, the interval is divided by selecting as a



**Figure 2: Path scrambler attack performed by host F against host A. The ADDF message received by F contains the identities of A,...,E, but the ADDF message sent by F to G has a scrambled list of identities, A, D, B, E, F.**

new probe, the host that is equidistant, in number of hops, to the endpoints of the interval. The faulty interval division process continues until the faulty interval is a link.

JANUS eliminates several difficulties of [3]. First, as a side effect of the path reservation phase, the sender, BS, knows the identities of all the hosts on its path to A. Unlike [3], where the discovery of the intermediaries is broadcast-based, our path reservation protocol requires only unicast on the routing tree path between BS and A. Second, in [3], the probes are selected by the originator of the traffic, by sending messages on the multihop ad-hoc path from itself to each probe. The acknowledgments sent by each probe, are also sent on the multihop ad-hoc path between the probe and the originator. In contrast, JANUS profits from the dual nature of hybrid networks, and transfers all the communication with the probes on the secure cellular links, thus greatly reducing the traffic. By making this distinction of the links in the hybrid network model, we can view the ad-hoc links as the data path, and the cellular links as the control path.

## 5. ANALYSIS

### 5.1 Impersonation

All the communication between the base station and mobile hosts is authenticated using pairwise secret keys. Moreover, when the authenticity of packets needs to be verified by third party hosts, we use digital signatures, created with the base station's private key. Note that each host knows the base station's public key, (Section 3.2). The use of digital signatures is restricted due to their high computation cost. We note however that mobile hosts only perform signature verification, which is much faster than signature generation, performed only by the base station.

In this section we present the defenses provided by JANUS against the attacks described in Section 4.2.

### 5.2 Rate Inflation and Tunneling

Rate inflation occurs when a host advertises a maliciously increased bandwidth value on its forward link or one of its adjacent



ad-hoc links. Tunneling occurs when two non-adjacent colluding hosts create a path between them, either through the base station or other mobile hosts, and then advertise each other as neighbors. Since the base station verifies the accuracy of the advertised link bandwidths using verifiable location information (Section 4.3) such attacks can be easily detected. In the case of tunneling, due to the use of verifiable position information [14], the actual bandwidth will be detected to be 0.

### 5.3 Denial of Service

In this hybrid setting, the only denial of service attacks can be performed by hosts that repeatedly initiate a path reservation protocol, without intending to use the established path, wasting the resources of the intermediate hosts.

Our defense relies on the INIT/SGN step that each host has to perform with the base station at the beginning of the path reservation phase. The base station authenticates the subsequent ADDF messages, by sending the client host a signed message (Section 4.4), that the client host has to use when contacting its parent with an ADDF message. The impact of a denial of service attack is localized by our protocol to the ad-hoc neighbors of a malicious host. The processing of ADDF packets still consumes resources, needed to verify the base station’s signature, although, hosts can decide to ignore ADDF packets received from repeatedly misbehaving neighbors.

### 5.4 Black Hole

Hosts that ignore data packets after correctly participating in the path reservation protocol, generate black holes. We detect such misbehaving hosts, with a link granularity, using  $O(\log n)$  probes, (Section 4.5). The feedback about malicious links is indirectly provided by the base station to the hosts affected by them, through the use of the topology tree, see Section 4.3. More specifically, when the base station detects a link  $(u, v)$ , where  $v$  is the parent of  $u$ , responsible for dropping more packets than advertised, it updates the link’s bandwidth in the topology tree. This is done using first a  $\text{cut}(u)$  operation to remove the existing link, followed by a  $\text{link}(u, v, w)$  to add a link between  $u$  and  $v$  of bandwidth  $w$ , where  $w$  is the bandwidth detected during the probing protocol, (Section 4.5). Subsequently, when calling  $\text{mincost}$ , hosts that use that link will be able to choose a better neighbor as parent.

### 5.5 Path Scrambler

At the conclusion of the path reservation protocol, the base station uses the identities of the hosts retrieved in the ADDF packet, and their keys shared with the base station, to verify the correctness of the HMAC received in the same packet. The two values do not coincide only in the case of an intermediate host having tampered with the ADDF message. The base station then performs a binary search, similar to the one used for the black hole attack, in order to retrieve a link that has a malicious host as an endpoint. The only difference from the black hole countermeasure, is that the malicious link is found using HMAC verification instead of packet loss detection. Let the path received by BS be  $P_1, \dots, P_k$ . The base station finds the median of the path,  $P_{(k+1)/2}$ , and probes it, through the forward link. The host has to reply with the entire path and the  $\text{HMAC}_{m-1}$  value received from its child. The base station then checks  $\text{HMAC}_{m-1}$  against the host’s path. The base station can perform this check since it knows the identities of the hosts on the path and the keys used to compute the HMAC.

The search continues on the interval whose left endpoint has a correct HMAC and whose right endpoint has an incorrect value. The search ends when two consecutive hosts on the path received by BS give different results on the HMAC check. If the two hosts are neigh-

bors, the link is considered malicious. Otherwise, the host whose HMAC does not check is malicious. It is easy to see why this is true if the two hosts are not neighbors, since each host has to check the parent-child relation during the path reservation phase, see Section 4.4. A host that does not report a message received from a host that is not its child, must be malicious. For the case where the hosts are neighbors, let the two consecutive hosts be T and U. The HMAC of T checks and the HMAC of U does not. Let S be the host preceding T and V the host succeeding U, in the path received by BS. The HMAC of S checks and the HMAC of V does not. If S would be a malicious host, then the HMAC of T would not check, since S would change the path that T receives. If V would be malicious and U honest, then the HMAC of U would check, since V cannot maliciously change the path that reaches one of its predecessors in the routing tree, before it reaches itself.

## 6. SIMULATION RESULTS

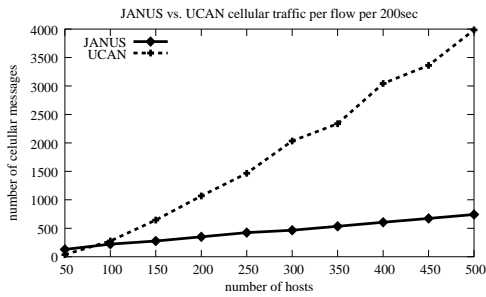
In this section we compare the performance of JANUS with UCAN [23]. UCAN proposed the dual interface model and introduced an on-demand routing algorithm. We compare the two algorithms in terms of the cellular and ad-hoc message overhead introduced.

We perform the experiments by placing mobile hosts in a square of size  $886 \times 886\text{m}^2$ , having the base station, with a transmission range of 600m, placed at its center. Each host is therefore covered by the base station’s transmission range. We use the dependency between a host’s forward link bandwidth and its distance to the base station, from [23], with the highest bandwidth value of 2.4Mbps. We use the random waypoint mobility model [20, 36], without a pause time, to simulate the behavior of mobile hosts. We assume that the ad-hoc transmission range of hosts is 115m, with a link bandwidth of 11Mbps at less than 50m distance and 1Mbps at 115m.

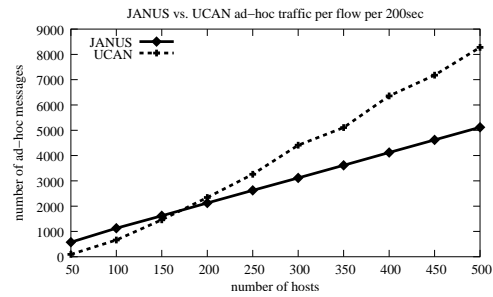
We place a number of hosts (between 50 and 500) in the square of size  $886 \times 886\text{m}^2$ , 20 of which have a flow from the base station. For JANUS, we assume a refresh rate of  $O(\log n)$ , where  $n$  is the number of hosts, and for UCAN we fix a TTL value of 3. For each network size, we choose 10 random initial configurations, and let the hosts move at a maximum speed of 10m/s for 200 seconds. Figure 3 shows the total number of cellular and ad-hoc messages required by JANUS and UCAN for 200 seconds of host movements, averaged over the 10 initial configurations.

Figure 3(a) shows that the overhead of JANUS in terms of cellular messages grows much slower than in UCAN. In JANUS, the number of cellular messages is dominated by the proactive part, where at each refresh period, all the hosts contact the base station to update the topology tree. Even though reactive, UCAN needs to maintain the paths used by the flows, that often break due to host mobility. Every host that receives the flooding message from a host with a lower forward link bandwidth, contacts the base station. The overhead in terms of the number of ad-hoc messages, shown in Figure 3(b) is higher for JANUS for networks with less than 150 hosts, but smaller for larger networks. This is because in JANUS, the number of ad-hoc messages is dominated by the periodic beaconing of the parent host in the routing tree. On the other hand, for dense networks, UCAN needs to contact many hosts in order to update broken paths (up to 200 hosts per broken path for a network of 500 hosts).

In the second experiment, we place 250 hosts in the same square, each host moving at a maximum speed of 10m/s. We increase the number of concurrent flows from 5 to 30. For each experiment, we choose 10 random initial configurations, each with a different set of hosts concurrently supporting flows, and perform each experiment

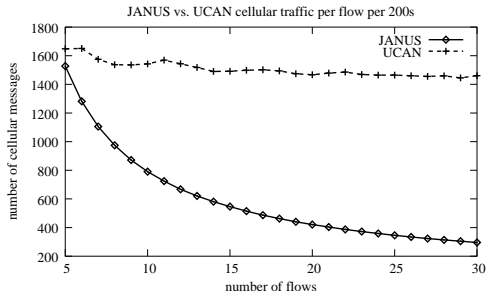


(a)

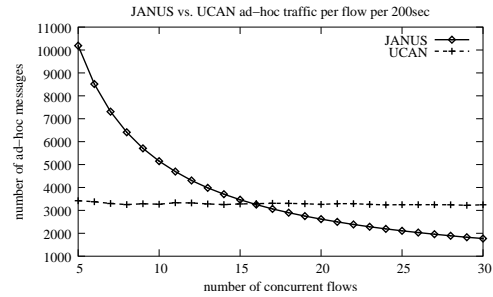


(b)

**Figure 3: Cellular and ad-hoc message overhead of JANUS and UCAN, for up to 500 mobile hosts. For each network, 20 hosts have a flow from the base station. We show the total number of messages required per flow maintenance, for the 200 seconds of hosts movements, averaged over 10 random initial network configurations.**



(a)



(b)

**Figure 4: Message overhead of JANUS and UCAN, for 250 hosts, with up to 30 concurrent flows. We show the total number of messages required to maintain all the flows, for the 200 seconds of hosts movements, averaged over 10 random initial network configurations.**

for 200 seconds. Figure 4 shows the number of cellular and ad-hoc messages required for the per-flow maintenance in JANUS and UCAN. Since most of the traffic overhead of JANUS is generated by the proactive part of the protocol, the number of messages experiences only a small increase with the number of concurrent flows, thus the number of per-flow messages decreases abruptly with the number of flows. For UCAN however, the growth is significant, proportional to the number of flows, which explains the almost constant number of messages required for a flow. In terms of the cellular overhead, JANUS is constantly more efficient than UCAN, whereas in terms of the ad-hoc overhead, JANUS starts paying off when there are more than 15 concurrent flows. However, since the main functionality of hybrid cellular and ad-hoc networks consists in accessing the cellular base station, it is reasonable to assume that the number of concurrent flows will be high.

## 7. CONCLUDING REMARKS

In this paper we have presented JANUS, a secure routing algorithm for dual interface, hybrid wireless networks. We have described several attacks that malicious hosts can perform against such networks, and we have explained the defenses provided by JANUS. We have measured experimentally the overhead incurred by JANUS and have shown that it outperforms UCAN by a large margin, without compromising the security of the network. Part of the future work will consist in extending JANUS to support alternative methods for estimating the aggregate throughput of a path, as a function of the intermediate links. Also, while we have focused on hybrid networks consisting of a cellular and a wireless interface,

we believe that the network model proposed is general enough to accommodate any network whose hosts communicate through fast but ephemeral links with each other, but with a slow and reliable link to a central access point. Future work in the same lines concentrates on extending the techniques presented here for overlay networks, which demonstrate a related set of security threats and they could be modeled in a similar fashion as hybrid networks.

## 8. REFERENCES

- [1] GPRS security threats and solutions. White Paper by NetScreen Technologies Inc., March 2002.
- [2] ANAGNOSTAKIS, K. G., AND GREENWALD, M. B. Exchange-based incentive mechanisms for peer-to-peer file sharing. In *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)* (2004), IEEE Computer Society, pp. 524–533.
- [3] AWERBUCH, B., HOLMER, D., NITA-ROTARU, C., AND RUBENS, H. An on-demand secure routing protocol resilient to byzantine failures. In *ACM Workshop on Wireless Security (WiSe)* (September 2002).
- [4] AWERBUCH, B., HOLMER, D., AND RUBENS, H. High throughput route selection in multi-rate ad hoc wireless networks. Tech. rep., March 2003.
- [5] BAL, H. E., BHOEDJANG, R., HOFMAN, R., JACOBS, C., LANGENDOEN, K., RUHL, T., AND KAASHOEK, M. F. Performance evaluation of the orca shared-object system. *ACM Trans. Comput. Syst.* 16, 1 (1998), 1–40.

- [6] BARKAN, E., BIHAM, E., AND KELLER, N. Instant ciphertext-only cryptanalysis of gsm encrypted communication. In *Proceedings of CRYPTO2003* (2003), LNCS, pp. 600–616.
- [7] BELLARDO, J., AND SAVAGE, S. 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. In *In USENIX 2003* (2003).
- [8] CAPKUN, S., HUBAUX, J. P., AND JAKOBSSON, M. Secure and privacy-preserving communication in hybrid ad hoc networks. Tech. Rep. IC/2004/10, EPFL-IC, January 2004.
- [9] CHEN, K., AND NAHRSTEDT, K. ipass: An incentive compatible auction scheme to enable packet forwarding service in manet. In *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)* (2004), IEEE Computer Society, pp. 534–542.
- [10] DANA, P. H. Global positioning system overview. The Geographer's Craft Project, Dept. of Geography, Univ. TX Austin, 1998.
- [11] DE COUTO, D. S. J., AGUAYO, D., BICKET, J., AND MORRIS, R. A high-throughput path metric for multi-hop wireless routing. In *Proceedings of the 9th annual international conference on Mobile computing and networking* (2003), ACM Press, pp. 134–146.
- [12] FREDERICKSON, G. N. Ambivalent data structures for dynamic 2-edge-connectivity and k smallest spanning trees. *SIAM J. of Comp.* 26(2) (1997), 484–538.
- [13] FUJIWARA, T., IIDA, N., AND WATANABE, T. An ad hoc routing protocol in hybrid wireless networks for emergency communications. In *IEEE ICDCS2004 (WWAN2004)* (March 2004), pp. 748–754.
- [14] GABBER, E., AND WOOL, A. On location-restricted services. *IEEE Network* 13 (Nov/Dec 1999), 44–52.
- [15] HU, Y.-C., JOHNSON, D. B., AND PERRIG, A. SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks. In *The 4th IEEE Workshop on Mobile Computing Systems and Applications* (June 2002).
- [16] HU, Y.-C., PERRIG, A., AND JOHNSON, D. B. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *The 8th ACM International Conference on Mobile Computing and Networking* (September 2002).
- [17] HU, Y.-C., PERRIG, A., AND JOHNSON, D. B. Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. In *Proceedings of the 22<sup>nd</sup> Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)* (April 2003).
- [18] HU, Y.-C., PERRIG, A., AND JOHNSON, D. B. Rushing attacks and defense in wireless ad hoc network routing protocols. In *ACM Workshop on Wireless Security (WiSe)* (2003).
- [19] JAIN, N., DAS, S., AND NASIPURI, A. A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks. In *In Proceedings of IEEE IC3N, Phoenix, October 2001.* (2001).
- [20] JOHNSON, D. B., AND MALTZ, D. A. *Dynamic Source Routing in Ad Hoc Wireless Networks*, vol. 353 of *Mobile Computing*. Kluwer Academic Publishers, 1996.
- [21] JOHNSON, D. B., MALTZ, D. A., AND BROCH, J. *DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks*. in *Ad Hoc Networking*. Addison-Wesley, 2001, ch. 5, pp. 139–172.
- [22] KYASANUR, P., AND VAIDYA, N. Detection and handling of MAC layer misbehavior in wireless networks. In *International Conference on Dependable Systems and Networks (DSN'03)* (2003).
- [23] LUO, H., RAMJEE, R., SINHA, P., LI, L. E., AND LU, S. Ucan: a unified cellular and ad-hoc network architecture. In *Proceedings of the 9th annual international conference on Mobile computing and networking* (2003), ACM Press, pp. 353–367.
- [24] MA, R. T. B., LEE, S. C. M., LUI, J. C. S., AND YAU, D. K. Y. An incentive mechanism for p2p networks. In *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)* (2004), IEEE Computer Society, pp. 516–523.
- [25] MARTI, S., GIULI, T., LAI, K., AND BAKER, M. Mitigating routing misbehavior in mobile ad hoc networks. In *The 6th ACM International Conference on Mobile Computing and Networking* (August 2000).
- [26] NASIPURI, A., AND DAS, S. Multichannel CSMA with signal power base channel selection for multihop wireless networks. In *In Proceedings of IEEE Vehicular Technology Conference (VTC)* (2000).
- [27] NIEMI, V., AND NYBERG, K. *UMTS Security*. Wiley Publishers, December 2003.
- [28] PAPADIMITRATOS, P., AND HAAS, Z. Secure routing for mobile ad hoc networks. In *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference* (January 2002), pp. 27–31.
- [29] PERKINS, C. E., AND BHAGWAT, P. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications* (1994).
- [30] PERKINS, C. E., AND ROYER, E. M. *Ad hoc Networking*. Addison-Wesley, 2000, ch. Ad hoc On-Demand Distance Vector Routing.
- [31] SALEM, N. B., BUTTYAN, L., HUBAUX, J.-P., AND JAKOBSSON, M. A charging and rewarding scheme for packet forwarding in multi-hop cellular networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing* (2003), ACM Press, pp. 13–24.
- [32] SANZGIRI, K., DAHILL, B., LEVINE, B. N., SHIELDS, C., AND BELDING-ROYER, E. A secure routing protocol for ad hoc networks. In *10th IEEE International Conference on Network Protocols (ICNP'02)* (November 2002).
- [33] SO, J., AND VAIDYA, N. Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver. In *In Proceedings of MobiHoc* (2004).
- [34] SUN, Q., AND GARCIA-MOLINA, H. Slic: A selfish link-based incentive mechanism for unstructured peer-to-peer networks. In *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)* (2004), IEEE Computer Society, pp. 506–515.
- [35] WU, S.-L., LIN, C.-Y., TSENG, Y.-C., AND SHEU, J.-P. A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks. In *Proceedings of the 2000 International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '00)* (2000), IEEE Computer Society, p. 232.
- [36] YOON, J., LIU, M., AND NOBLE, B. Random waypoint considered harmful. In *INFOCOM* (2003).