Providing Mobile Users' Anonymity in Hybrid Networks*

C.A. Ardagna[1], S. Jajodia[2], P. Samarati[1], and A. Stavrou[2]

[1] DTI - Università degli Studi di Milano, 26013 Crema, Italia
{claudio.ardagna,pierangela.samarati}@unimi.it
[2] CSIS - George Mason University, Fairfax, VA 22030-4444, USA
{jajodia,astavrou}@gmu.edu

Abstract. We present a novel hybrid communication protocol that guarantees mobile users' k-anonymity against a wide-range of adversaries by exploiting the capability of handheld devices to connect to both WiFi and cellular networks. Unlike existing anonymity schemes, we consider all parties that can intercept communications between the mobile user and a server as potential privacy threats. We formally quantify the privacy exposure and the protection of our system in the presence of malicious neighboring peers, global WiFi eavesdroppers, and omniscient mobile network operators. We show how our system provides an automatic incentive for users to collaborate, since by forwarding packets for other peers users gain anonymity for their own traffic.

1 Introduction

We live in a globally interconnected society characterized by pervasive ubiquitous devices and communication technologies. The wide diffusion of the Internet, cellular networks, WiFi, low cost mobile devices, and the high availability of online services enable today's e-citizens to carry out tasks, access services, and stay connected virtually anywhere anytime. Unfortunately, the price we pay for this usability and convenience is an increased exposure of users' information and online activities. Organizations and individuals are slowly becoming aware of the privacy risks to which they are exposed. This scenario has sparked a renewed interest in the problem of providing privacy guarantees to users when operating in this brave new electronic world. Previous research has addressed different angles of the privacy problem. With respect to users' privacy, approaches like Mix-net [5], Onion Routing [8], and Crowds [19] were geared towards protecting the network anonymity of the users, preventing an adversary from linking

* This work was supported in part by the EU within the 7FP project "PrimeLife" under grant agreement 216483; by the Italian Ministry of Research within the PRIN 2008 project "PEPPER" (2008SY2PH4); by the National Science Foundation under grants CT-20013A, CT-0716567, CT-0716323, CT-0627493, and CCF-1037987; by the Air Force Office of Scientific Research under grants FA9550-07-1-0527, FA9550-09-1-0421, and FA9550-08-1-0157; and by the Army Research Office DURIP award W911NF-09-01-0352.

the user to a service request. All these solutions were designed with traditional wired networks in mind, and shared the implicit assumptions on the stability of the routing configuration and network topology. In addition, many of them use the same path to route both user requests and responses. Existing solutions for wired networks are then not applicable in mobile networks where users can move fast in a short period of time and therefore cannot maintain a static communication path involving the same nodes between requests and responses. Approaches that have addressed the privacy problem in mobile ad-hoc networks (e.g., [14, 17]) have been mostly aimed at providing anonymous routing protocols and have not considered protection of users' anonymity against the network operators; also, they typically rely on expensive multiparty computation and are therefore not suitable for mobile scenarios. On the other hand, privacy proposals for mobile networks (e.g., [7]) have addressed the problem of protecting users' anonymity against the services they access. These proposals, however, assume the mobile network operator to be in a privileged position and able to observe all the communications of the users.

In this paper, we study the privacy problem in hybrid networks where users, in addition to accessing online services via the cellular network, can communicate among each other over a WiFi network. Our goal is then to enable users to access online services using a cellular network in a privacy preserving way. To this end, we introduce a protocol that relies on the hybrid nature of mobile devices to create a local WiFi network which is impervious against global eavesdroppers that operate in the cellular network (e.g., mobile network operators). Our approach bases on the cooperation among peers in the WiFi network. An interesting aspect is that by collaborating in providing anonymity to others, peers gain themselves an immediate benefit on the anonymity of their communication. There is therefore an automatic incentive for peers to cooperate in the protocol.

Our approach represents an important paradigm shift, departing from the usual assumption of the mobile network operator as a trusted powerful entity able to know and observe all the traffic in the network. The mobile operator, while considered trustworthy with respect to the availability and working of the network, is restricted in terms of the view and traffic it can reconstruct. Addressing a novel threat and problem, our work is therefore complementary to existing solutions for privacy protection and could be applied in conjunction with them. Furthermore, we offer two important advantages over previous approaches. First, we do not rely on expensive communication or cryptographic operations including the use of multiparty computation, and we do not employ public key cryptography to convey the information to the server, beyond the connection establishment phase. Instead, we introduce a new fast packet filtering that leverages pseudo-random number generation to guarantee communication integrity. This aspect is particularly important to ensure applicability in a mobile environment, where low computation overhead and limited battery consumption are important requirements. Second, while guaranteeing privacy, we provide protection of the system against possible abuses of anonymity by maintaining the ability to block malicious traffic.

2 Problem definition

Our reference model is a distributed and mobile infrastructure which forms a hybrid network, integrating *wireless*, *cellular*, and *wired* connections. The participating entities are: *1) mobile users*, that carry mobile devices supporting both GSM/3G and WiFi protocols for communication; *2) mobile network operators*, that manage radio cells of the cellular networks to provide wired network access to mobile users; and *3) servers*, that provide online services over the cellular network or the Internet. Mobile users can establish ad-hoc (WiFi) point-to-point connections with other mobile users in the network, resulting in several Mobile Ad-hoc NETworks (MANETs). Each mobile user, receiving signals from radio cells, is also registered with a given mobile network operator to access cellular functionalities. The cellular network acts as a gateway establishing a point-to-point connection with the user and the server. *Communication* is a bidirectional exchange of messages that involves a *user* u and a *server* s. Our goal is to provide a means for users to communicate with servers without giving the operator the ability to observe the communication profiles, that is, the pairs $\langle$*user,server*$\rangle$ describing service accesses. Protection is enacted by involving, in the communication with the mobile operator, other peers (users) with whom the user communicates via the WiFi network. Our approach guarantees that also participating peers will not be able to reconstruct the communication profile. We define the degree of anonymity protection enjoyed by a communication by modeling the uncertainty over the user and the server involved in it as follows.

Definition 1 ((k,h)-anonymity). *A communication is said to be (k,h)-anonymous against an adversary v, if v cannot relate the communication to less that k users and h servers.*

A communication is (k,h)-anonymous against an adversary, if the probability for the adversary of associating any u as the originating user is at most $\frac{1}{k}$ and the probability of associating any s as the server is at most $\frac{1}{h}$. A $*$ in place of a specific value for k (h, resp.) denotes that no inference can be drawn on the user (server, resp.) of a communication, which can therefore be any user (server, resp.) of the network. The degree of anonymity of a communication depends on the adversary. For each communication, user and server are known to each other, so their communications are $(1,1)$-anonymous to them. We assume the server of a communication to be always known to the mobile operator. With respect to a mobile operator, all communications will therefore be $(k,1)$-anonymous, where k defines the degree of k-anonymity [6, 23] set by the user and provided by our protocol. Since the focus of our work is the protection of user's relations with servers against the mobile operator, our goal is to guarantee the k defined by the user. The reason for considering communication anonymity as a pair taking into consideration also the uncertainty on the server, is to model also the view of peers in the network (which do not know the servers to whom packets are being delivered). A communication between a user and a server is said to be completely exposed to an adversary if it is $(1,1)$-anonymous to her. It is considered protected if it is (k,h)-anonymous with $max(k,h) > 1$.

3 Rationale and basics of our approach

The core idea of our approach is to empower users to anonymously involve other peers in sending a message to the server via a mobile operator using the WiFi network. Each message is split in k different packets and randomly distributed to k distinct peers in the WiFi network for their forwarding to the mobile network operator, that will then receive k indistinguishable packets from k different senders. Before introducing our communication protocol, we illustrate the basic knowledge that peers, operators, and servers participating in the network maintain or share.

Before any anonymous communication can be established, the user has to register and agree upon a secret key with the server. This pre-established secret key is used as a seed by the user to generate pseudo-random numbers to be associated with packets. All the servers, based on the seeds agreed with the users, jointly create a global table LEGITIMATE. LEGITIMATE consists of pairs (R^1, R^2) of pseudo-random numbers. This table can be either hosted by an external server accessible by the mobile network operators or alternatively stored by the operators themselves. Upon a packet arrival, the mobile network operator retrieves the pseudo random number attached to the packet and performs a lookup to the LEGITIMATE table to verify the validity of a packet. The cost of maintaining the LEGITIMATE table is manageable. For instance, assuming 128 pairs (R^1, R^2) of 64 bits of pseudo-random numbers to be used for packet verification and 1000 servers with 1 million users each, the storage requirements are approximately 1 TB which can be maintained by today off-the-shelf disks. The use of an external repository can then eliminate the need for a pre-storage of the random numbers since this repository can act as an intermediary between the individual servers and the mobile network operators. The size of R^1 and R^2 is chosen to be only 32 bits because each number is used only once and then discarded to avoid correlation attacks. The LEGITIMATE table acts as a blind firewall filter, allowing only packets tagged with an existing pseudo-random number (R^1) and having a valid encrypted message body to pass through.

Each peer p maintains the following tables: SENT_p contains the identifiers of the communications that the peer has helped distributing (including those originated by the peer) by forwarding a packet to the mobile operator; $\text{MYPRN}_{p,seed}$ contains the set of pseudo-random numbers prn_i=(R_i^1, R_i^2) generated by p using the *seed* shared with the corresponding server. $\text{MYPRN}_{p,seed}$ contains the same *prn* generated by the server and is then a subset of the LEGITIMATE table. Each server s has a public/secret key pair $\langle P_s,S_s\rangle$. P_s is used by users when requesting connection establishment to encrypt the body of their message. This body includes a shared session key SK to be used by the user and the server for all further message exchanges in the session. Finally, each server s locally maintains a table ORIG_{sid} for each session *sid*, which stores the original set of peers (including user u) involved in the connection establishment.

To enforce integrity verification, we employ the UMAC [3] algorithm with R^2 as the key and the first 64 bits of the encrypted body of the message as a nonce for message authentication control. UMAC is designed to be very fast to

compute in software on contemporary uniprocessors with measured speeds as low as one cycle per byte [15] . In addition, the analysis of UMAC shows this scheme to have provable security, in the sense of modern cryptography, by way of tight reductions. Once a packet is forwarded to the server, the pseudo-random pair is removed from the table. Packets with invalid (i.e., *non-existing*) R^1 or invalid UMACs are discarded. The use of random numbers enables the protection of the servers against flooding attacks (mobile operators will discard packets that are found to be not genuine) preventing Denial-of-Service (DoS) attacks to servers.

4 Protocol

We present the working of the communication protocol distinguishing management of requests and responses. We use standard notation $E^s_K()$ and $D^s_K()$ to denote symmetric encryption and decryption operation with key K whereas $E^p_K()$ and $D^p_K()$ denote public key operations *used only for connection establishment*. Also, we will use $\mathcal{P}$, $\mathcal{O}$, and $\mathcal{S}$ to denote respectively the set of peers, mobile network operators, and servers in the hybrid network, and id_p and id_s as the identifiers of a peer and a server. Note that, to access a server, a user has first to establish a connection. In our protocol, connection requests and service access requests are indistinguishable to parties different from the initiating user and the server; all these parties (participating peers and mobile network operators) will simply observe packets without knowing whether they relate to a connection establishment or to a service access. The protocol and the behavior of the involved parties are the same for the two cases; the only differences are: *i)* in the set of peers selected, which contains user u, in the case of connection request; *ii)* within the content of the message, which contains the key for the session, in the case of connection request, and the id of the session, in the case of service request. Also, the body of the connection request packet is encrypted with the server's public key while the body of the service request packet is encrypted with the session key to which the request refers. Finally, for each service request, the response is also returned to peers in $\textsc{Orig}_{sid}$.

Figure 1 illustrates the protocol operations at the different participating parties. Figure 2 illustrates the distribution of packets among parties illustrating also how the content of the packets changes. Arcs with double lines refer to communications over the WiFi network (among peers), arcs with a single line refer to communications over the cellular network (between peers and mobile operators), arcs with a bold line refer to communications that can be carried on either over the wired or the cellular network (between the mobile operators and the servers), and arcs with a dotted line represent internal computations. Encrypted content is reported as a box with the encryption key appearing in the lower right corner of the box. Packets in Figure 2 refer to service access.

4.1 Request

For each session, a user can specify a privacy degree k to be guaranteed for all communications (connection and service requests related to the session) and

REQUEST ($u \rightarrow s$)
User $u \in \mathcal{P}$
u1.1 Let m be the message to be sent and *payload* its content.
Let k be the privacy preference, $(1 - P_f)$ the probability of forwarding to the operator, *cid* the communication id, UMAC_R a Universal Message Authentication Code (UMAC) using key R
u1.2 Generate a random message identifier *mid* and obtain the timestamp *tmp*
u1.3 Split *payload* in k parts $payload_i$, each with a sequence number seq_i, with i:=1...k
u1.4 **for** i:=1...k **do**
Generate $prn_i = (R_i^1, R_i^2)$ using the *seed*
$to_i := R_i^1$
if the message is a connection request
then generate session key *SK*
$body_i := E^p_{P_s}(id_u, seq_i, payload_i, SK, mid, tmp)$ /*connection establishment*/
else $body_i := E^s_{SK}(id_u, seq_i, payload_i, sid, mid, tmp)$ /*service access*/
u1.5 Wait until k peers are available
u1.6 **for** i:=2...k **do**
Choose a peer $p_i \in \mathcal{P}$
With random delay, send m_i:=[to_i,$body_i$,$\text{UMAC}_{R_i^2}\{body_i\}$,*cid*] to p_i in the WiFi network
u1.7 **if** ($cid \notin \text{SENT}_u$)
then $\text{SENT}_u := \text{SENT}_u \cup \{cid\}$
With random delay, forward [to_1,$body_1$,$\text{UMAC}_{R_1^2}\{body_1\}$] to o over the cellular network
else Send m_1:=[to_1,$body_1$,$\text{UMAC}_{R_1^2}\{body_1\}$,*cid*] to p_1 in the WiFi network

Peer $p \in \mathcal{P}$
Upon receiving a packet [*to*,*body*,$\text{UMAC}_{R^2}\{body\}$,*cid*]
p1.1 **if** ($cid \notin \text{SENT}_p$)
then With probability $(1 - P_f)$: (Forward [*to*,*body*,$\text{UMAC}_{R^2}\{body\}$] to o over the cellular network; $\text{SENT}_p := \text{SENT}_p \cup \{cid\}$; exit)
Send [*to*,*body*,$\text{UMAC}_{R^2}\{\text{body}\}$,*cid*] to a peer $p \in \mathcal{P}$

Operator $o \in \mathcal{O}$
Upon receiving [*to*,*body*,$\text{UMAC}_{R^2}\{body\}$] from peer p
o1.1 **if** (($to \in \text{LEGITIMATE}$) and ($\text{UMAC}_{R^2}\{body\}$ is valid)
then Identify s using (to, R^2), remove (to, R^2) from LEGITIMATE and forward [id_p,*to*,*body*] to s
else Drop the packet and exit

Server $s \in \mathcal{S}$
Upon receiving [id_p,*to*,*body*] from p via o
s1.1 Based on *to*, retrieve the content as $\text{D}^p_K(body) \vee \text{D}^s_K(body)$ with K:=$S_s \vee$ K:=*SK* respectively
s1.2 $\text{ORIG}_{sid} = \text{ORIG}_{sid} \cup \{[id_p, o]\}$ /*connection establishment*/
s1.3 Assemble the original message m with identifier *mid*

RESPONSE ($s \rightarrow u$)
Server $s \in \mathcal{S}$
Upon receiving all packets [id_p,*to*,*body*] for a request
s2.1 Let *payload* be the response, *sid* be the session id, and *SK* the session key
s2.2 **for** i:=1...k **do**
Let id_{p_i} and o_i be the peer id and the operator of the i-th packet of message *mid*
Generate the next random number from the seed $prn_i = (R_i^1, R_i^2)$
$body_i := E_{SK}(payload, sid, tmp)$
Send [id_{p_i},id_s,prn_i,$body_i$] to o_i
s2.3 **for each** $e_j \in \text{ORIG}_{sid}$ with j=1...k /*service access*/
Generate the next random number from the seed $prn_j = (R_j^1, R_j^2)$
$body_j := E_{SK}(payload, sid, tmp)$
Send [$e_j.id_p$,id_s,prn_j,$body_j$] to $e_j.o$

Operator $o \in \mathcal{O}$
Upon receiving [id_p,id_s,*prn*,*body*] from s
o2.1 Remove *prn* from LEGITIMATE and forward [id_s,*prn*,*body*] to p

User/Peer $p \in \mathcal{P}$
Upon receiving [id_s,*prn*,*body*]
up2.1 **if** ($prn \in \text{MYPRN}_{p,seed}$)
then retrieve response as $\text{D}^s_{SK}(body)$
else drop the packet

Fig. 1. Communication protocol

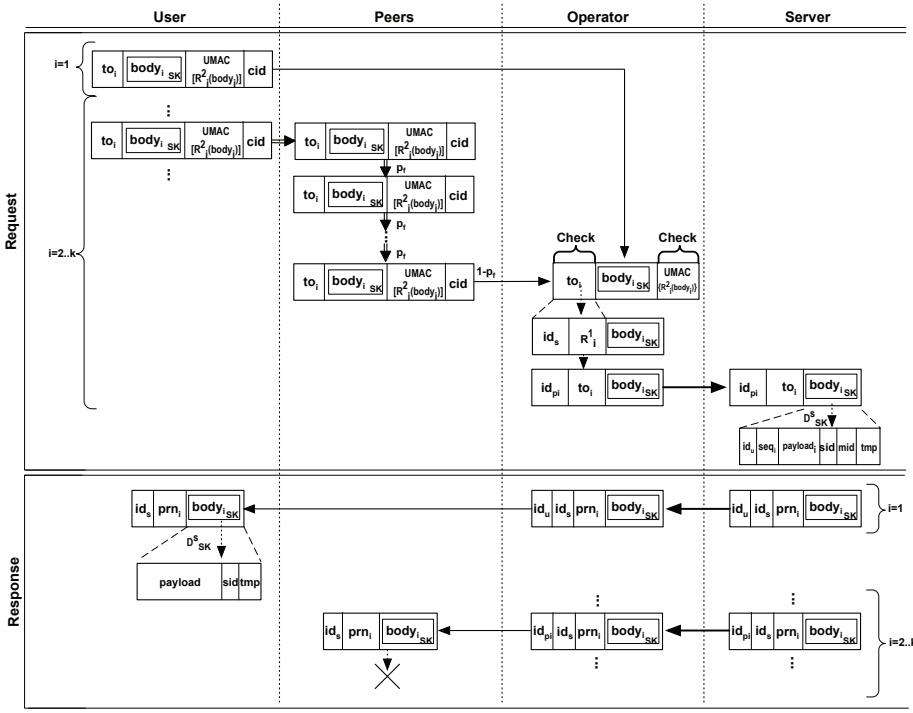

Fig. 2. Flow of packets within our protocol

the communication identifier *cid* to be used for all WiFi communications. The reason for *cid* is to limit to one the number of packets that a peer can send to the operator in each session (or in a window of time W like in Section 5).

User. Let m be a message with content *payload* to be sent by user u to server s. Let k be the privacy degree to be enforced, P_f and $(1 - P_f)$ the probability of forwarding to a peer in the communication range and to the operator, respectively, *cid* the communication identifier, and UMAC_R a Universal Message Authentication Code (UMAC) using key R. First, the user generates a random number *mid* that will be used as identifier for the message, and obtains the timestamp *tmp*. Then, the *payload* of the message is split into k different parts, $payload_1$,...,$payload_k$, each identified with its sequence number seq_i, to be sent via k different packets, composed as follows. For each packet m_i to be sent, to prove that the packet originates from a genuine user, the user generates, using the *seed* agreed with the server, a 64 bits pseudo-random number and splits it into two parts (i.e., $prn_i = (R_i^1, R_i^2)$). It then uses R_i^1 as the to_i field of packet m_i. The body $body_i$ of each packet to be sent, composed of the user id (id_u), sequence number of the packet (seq_i), packet payload ($payload_i$), message identifier *mid*, timestamp *tmp*, and either session key *SK* to be used for subsequent communication in the session (for connection requests), or session identifier *sid*

(for service access requests), is then encrypted. Encryption is performed with the server's public key P_s in case of connection requests and with the symmetric session key SK in case of service requests. In addition, a UMAC with R_i^2 as the key is used to produce the signature of the first 64 bits of the encrypted body, $\text{UMAC}_{R_i^2}\{\text{body}\}$, that is then appended at the end of the packet. Therefore, each packet m_i composed of $[to_i, body_i, \text{UMAC}_{R_i^2}\{body_i\}, cid]$, with $i := 1, \ldots, k$, is sent with a random delay to a different peer in the communication range. To avoid infinite loops in the distribution process, the user verifies through the WiFi channel if at least k peers (including u itself) are available in her proximity. If this is not the case, the user will not send the packet until enough peers become available. In the case of connection establishment (i.e., $cid \notin \text{SENT}_u$), the first packet m_1 is managed by u herself, that adds cid to SENT_u, keeping track of communications for which a packet has been forwarded to the operator; moreover, with a random delay, u forwards $m_1 = [to_1, body_1, \text{UMAC}_{R_1^2}\{body_1\}]$ to her operator o.

Peer. Upon receiving a packet $[to, body, \text{UMAC}_{R^2}\{body\}, cid]$, each peer p checks if it has already sent to the mobile operator any packet for the same communication (i.e., $cid \in \text{SENT}_p$). If it has not, the peer p sends the packet to its operator o with probability $(1 - P_f)$ and adds cid to SENT_p; while with probability P_f, it sends the packet unchanged to a peer in the communication range.

Operator. Upon receiving a packet $[to, body, \text{UMAC}_{R^2}\{\text{body}\}]$ from a peer p, the operator uses R^1 in the to field to retrieve the pair (R^1, R^2) in the global table LEGITIMATE, and checks the validity of $\text{UMAC}_{R^2}\{\text{body}\}$. If R^1 is a legitimate number (i.e., belongs to global table LEGITIMATE) and $\text{UMAC}_{R^2}\{\text{body}\}$ is a valid signature, the packet is genuine and the operator sends a message $[id_p, to, body]$ to server id_s. The remote server id_s is identified as the one that provided the pair associated with the packet. Also, the pair (R^1, R^2) is removed from the global table LEGITIMATE to ensure one-time use. If either R^1 is not in the table or the UMAC value of the body using R^2 is invalid, the packet is considered not genuine and dropped. Note that, the reason for including R^1 in each message to the servers, is to allow servers to quickly determine the key to be used in body decryption.

Server. Upon receiving a packet $[id_p, to, body]$ from operator o, using the field to, the server determines the encryption key K with which body was encrypted (server's public key P_s or session key SK), and decrypts $body$ accordingly (with server's private key S_s or session key SK, respectively). It then assembles the original message by merging the payloads in the bodies of the different packets. If the original message cannot be reconstructed, the communication is dropped and no response is returned to the user. In the case of connection establishment, for each received packet, the server adds $\{[id_p, o]\}$ to her local table ORIG_{sid}.

4.2 Response

Upon completion of the reception of all packets for the same message, the server determines the responses to be sent to different peers.

Server. Let *payload* be the response to be sent, *sid* be the session it refers to, and *SK* be the corresponding session key. For each packet related to message *mid*, received from peer p_i via operator o_i, the server generates $prn_i = (R_i^1, R_i^2)$ based on the *seed* shared with u. The body $body_i$ of the response is determined by encrypting, with session key *SK*: *payload* of the response, session identifier *sid*, and timestamp *tmp*. The server then sends $[id_{p_i}, id_s, prn_i, body_i]$ to operator o_i. Note that to make the body of responses referred to the same message different and indistinguishable from one another, the same body is encrypted i different times, by using a symmetric key encryption algorithm (e.g., 3DES, AES). In service access communication, a response $[e_j.id_p, id_s, prn_j, body_j]$ is also sent to each peer $e_j \in \text{ORIG}_{sid}$. As above, j different *prn* are used and j different *body* are generated by encrypting j individual times the plain message.

Operator. Upon receiving a response packet $[id_p, id_s, prn, body]$, the operator removes prn_i from table LEGITIMATE and forwards $[id_s, prn, body]$ to peer p.

User/Peer. Upon receiving a response packet $[id_s, prn, body]$ each peer p determines if *prn* belongs to one of her sets of pseudo random numbers ($prn \in \text{MYPRN}_{p,seed}$). If so, the peer was the initiating user u of the message to which the response refers, and can determine the decryption key thus retrieving *body* accordingly. Else, the peer drops the packet.

5 Assessing *k*-anonymity

In our approach, a user establishing a connection needs to specify the number of peers whose cooperation it requests for achieving k-anonymity. In absence of previous history and in a non malicious environment, k-anonymity can be achieved by requesting cooperation of exactly k peers (as assumed in Section 4). However, the necessary number N of peers to involve to reach k-anonymity can decrease leveraging on previous communications in which the requester was involved, either as requester or participant on behalf of others, which introduce entropy. By contrast, it can increase in the presence of malicious peers and the consequent need to introduce redundancy in the system to provide resilience against them. In this section we discuss how a user can establish the number N of peers to involve in the protocol based on past communications and on a possible adversarial environment.

To prevent potential attacks from adversaries who try to subvert anonymity by using traffic analysis, we use a probabilistic path length and a multi-path approach. The expected path length L between a mobile user and the network operator (i.e., the number of hops taken by a packet in its path from a source to a destination) is randomly and exponentially distributed. In our multi-path configuration, a message originator or one of the peers forward each packet to a random next-hop peer with the same probability of forwarding P_f. Different packets of the same message follow different paths (that can be partially overlapped). The last peer on each path that has received a packet sends it to the network operator directly with the probability $(1 - P_f)$. Thus, like

in [19], we can derive the expected path length in non-malicious environment as: $L = (1 - P_f) \sum_{k=0}^{\infty} (k + 2) {P_f}^k = \frac{P_f}{(1-P_f)} + 2$.

Unfortunately, not all forwarded packets can be considered legitimate and not all neighboring peers are honest. To account for this, we define a threshold probability P_d of peers who misbehave. This probability includes peers moving out of the transmission range, dropping out of the network, acting maliciously by dropping or falsifying the packets they receive, or, in general, attempting to disrupt the normal operation of the system. Moreover, this probability threshold accounts for Sybil attacks [10] where a malicious peer can assume multiple false identities by pretending to have multiple WiFi physical occurrences. We assume that some fraction of peers in the WiFi network are malicious but the message originator is not. The expected path length in the presence of malicious peers that drop a packet can then be calculated as: $L = \frac{(1-P_d)P_f}{1-(1-P_d)P_f} + 2$.

In the remainder of this section, we analyze how the user can determine the number of peers to involve in the protocol to guarantee k-anonymity in case of communications involving a single request-response (Subsection 5.1) and multiple requests-responses (Subsection 5.2).

5.1 Single request-response

Each mobile user maintains the number M_s of packets forwarded for others to server s within a window of time W. In the protocol, to allow peers to calculate M_s, the server identifier id_s is declared in the response. The reason for declaring the server in the response, rather than in the request, is that the response traveling over the cellular network is not visible to WiFi eavesdroppers (see Section 6.1). While in Section 4 we assumed W to be equal to the session window, in the following, window W can be as large or as small as the mobile user prefers and is taken as a reference to evaluate the degree of anonymity. The value of W is not critical for single requests but becomes significant in case of consecutive requests to the same server. We envision that a typical value of W can range from a few seconds to several minutes. Assuming no malicious neighbors, the number of peers N, that user u needs to involve in a communication to achieve the required k-anonymity, is $N = k - M_s$, where M_s represents the number of packets the user has forwarded for other mobile peers to server s. In fact, if the user has participated in M_s previous communications, there must exist at least one peer that also participated in each of them.

Assuming the probability of malicious peers is at most P_d, to achieve the required k-anonymity for a request to server s in the window of time W, u must select at least N peers to satisfy the following formula:

$$k = \underbrace{\sum_{i}^{N} (1 - P_d)^L}_{N_f} + \underbrace{M_s \cdot (1 - P_d)}_{N_m} \tag{1}$$

Equation (1) has two contributing factors: $N_f = \sum_i^N (1 - P_d)^L$ indicates the expected number of successfully forwarded packets to the operator even in the presence of a fraction of P_d malicious peers; $N_m = M_s \cdot (1 - P_d)$ accounts for the anonymity the user has gained by virtue of forwarding M_s packets for other mobile peers to server s. Of course, the mobile user will not be able to know the size UP of the set of unique peers that have communicated with the server but she can estimate that: $M_s \cdot (1 - P_d) \leq UP \leq \sum_{i=1}^{M_s} l_i \cdot (1 - P_d)$, where l_i is the number of peers that participate in the i-th communication. We consider the worst case scenario of having each l_i=1, and thus UP=$M_s \cdot (1-P_d)$. Therefore, by forwarding packets for other mobile peers, a mobile user builds the necessary communication history that allows her to gain k-anonymity for her own traffic. In general, to determine the number N of necessary peers to involve in a communication, there are two extremes (see Equation 1): if u does not have any packet history within the window W, she needs to select a set of N peers that will successfully forward k packets to the server s, even if there is a fraction of P_d malicious peers. On the other hand, if u has already forwarded M_s packets to server s for other peers, if $N_m \geq k$, u can still enjoy k-anonymity without using any neighboring mobile peer, even assuming that P_d of them are malicious. A combination of the two extremes is also possible. In addition, based on the discussion in Section 4, a safe distribution process starts if and only if k peers (possibly including u) are available in users' proximity (i.e., path lenght L=1). Therefore, if we consider Equation 1 with probability of malicious users P_d, a user can safely start the communication if the number of available peers N satisfies $k = (N + M_s) \cdot (1 - P_d)$.

5.2 Multiple requests-responses

The analysis in Subsection 5.1 assumed that the communication between a user and server entails at most one message exchange. In practical applications, service access may require several messages between the involved user and the server. This opens the door to possible intersection attacks by which an observer can exploit the fact that a given user appears in different messages directed to a server. To counteract intersection attacks we ensure that both the requester as well as any other peer participate at most in the delivery of one message to the server in each given window W. The requester participates only in the first message exchange, but will receive all the responses since the server will send all responses to the original senders $\textsc{Orig}_{sid}$ (step s2.3). Also, peers participate in delivering a message only if they have not yet delivered any message for that communication (step p1.1). The important parameter is therefore the length of window W after which *cid* and $\textsc{Orig}_{sid}$ should be reset. Large sizes of W increase the potential level of anonymity but can decrease the ability of successfully concluding the communication. In fact peers that have participated in previous message delivery within a window become not usable anymore for forwarding packets to the operator and can then be modeled, with the formalization provided in Subsection 5.1, as malicious peers. The probability of packet dropping would then become $P_d = P_d + i\frac{k}{|\mathcal{P}|}$, where i is the number of request-response

steps in the communication, $|\mathcal{P}|$ is the total number of peers in the network, and k is the preference of the requester. The probability P_d is then proportional to the number of steps i; correspondingly, the probability of finding enough peers around u (i.e., N such that $k = (N + M_s) \cdot (1 - P_d)$ holds) decreases. By contrast, if W is small the probability of identifying the requester by means of intersection attacks increases. Each requester is in fact involved many times in a single communication and is more likely to be identified by an adversary. One limit case is when W is close to 0. In this case, the requester is involved in the packet forwarding of each request and thus she can be exposed with high probability.

6 Adversarial analysis

Here, we present an analysis of our protocol against attacks by individual or colluding adversaries eavesdropping the communication as well as against timing and predecessor attacks. In addition, we point out the differences between wireless and wired networks and between full and k-anonymity.

6.1 Adversaries eavesdropping the communication

We assume that all participating parties in our system can play the role of adversary eavesdropping the communication and possibly collude.

Operator. A single o can only observe the communications involving peers that use o to forward their communications over the cellular network. Our system is designed to prevent o from identifying the originator u of a request below the k-anonymity threshold that the user selects. In fact, the originator u may not be subscribed to o, and then o is not able to observe the packets sent by u. Therefore, although o can relate the request to server s, it cannot deduct any information regarding u; hence, $(*, 1)$-anonymity is preserved.

Global WiFi eavesdropper. A Global WiFi eavesdropper can collect and analyze all WiFi traffic. Therefore, it can identify packets originating from mobile peers and potentially breach the requester's k-anonymity. However, a WiFi eavesdropper is not capable of identifying packets of the same message (i.e., with the same *mid*) in a short time interval. Moreover, it does not receive the responses from the server (which are communicated via the cellular network) and then it does not know the identity of server s. Therefore, a WiFi eavesdropper, short of breaking the cryptographic keys, cannot extract any information regarding o and s; hence $(1, *)$-anonymity is preserved.

But how easy is to create a WiFi eavesdropper? In WiFi communications peers establish point-to-point WiFi connections on ad-hoc channels. Hence, traditional WiFi providers are not able to simply use their access points to observe *all* WiFi communications. Rather, they need to employ ad-hoc antennas to cover *all* the area of interest and overhear on *all* point-to-point communications. Thus, the global WiFi eavesdropper scenario is possible in principle but difficult in practice. Another avenue of attack is to simulate a global WiFi eavesdropper

employing "shadowing" neighbor peers that follow the victim. This attack is a special case of global WiFi eavesdropper. On their own, these nodes do not have access to message content both in connection establishment and service access sub-protocols. In addition, every WiFi peer overhearing on the communications cannot assume that each packet forwarded by u, is originated by u himself, due to the hidden terminal problem that exists in all IEEE 802.11 communications [2]. This is a serious limitation and assumes that the WiFi nodes shadowing the victim will have to calculate and compensate for channel fading and signal loss due to physical objects. Finally, since packets need not to be manipulated by intermediate peers, there is no need to add identity or identifiable information to the packets in clear. Thus, the adversary is not able to infer who is the peer broadcasting a packet, unless there is a single peer in the communication range that is also physically visible by the adversary. $(k, *)$-anonymity is therefore preserved.

Colluding operators. This adversarial model results in an omniscient operator o that can observe all the traffic in the cellular network generated by mobile users using o to route their packets to the server. Our system does not attempt to protect the server's anonymity from such o and thus, o can observe all packets header information for a given time interval. Therefore, for each window of time W, o receives a set $M = \{m_{p,s,t}\}$ of packets from the cellular network where p denotes a mobile user, s is a server, and t is o's packet timestamp. Also, $\mathcal{P}$ denotes the set of mobile peers, $\mathcal{S}$ the set of servers, with $|\mathcal{S}| \leq |\mathcal{P}|$, and $K = \{k_1, \ldots, k_{|\mathcal{P}|}\}$ the set of peers' preferences. Operator o can place the observed packets in different sets, grouping packets having the same s, the same p, or the same pair (p, s). Given a server s, $M_{*,s}$=$\{m_{p',s',t'} \in M | s' = s\}$ is the set of all packets sent to the same server s, and $M_{p,s}$=$\{m_{p',s',t'} \in M | p' = p, s' = s\}$ is the set of packets sent from a peer p to a server s.

But, what can o extract from these sets that can be used for inference? There are two important metrics in each window of time W: *i)* the number of packets transmitted by unique peers to a server s, that is, $|\hat{M}_{*,s}|$, and *ii)* the maximum number of packet repetitions from a specific mobile peer p towards a specific server s. The first metric can be used to bound the maximum number of mobile peers that have *potentially* communicated with server s *assuming that o receives all the packets from all the mobile peers*. Let $max\{k_u\}$ be the greatest among all preferences of requesters u_i. If $max\{k_u\} \leq |\hat{M}_{*,s}|$, k-anonymity is preserved. This holds because if there are more than one requester communicating with the same server, then o will receive packets from all the peers involved in the communications. Our analysis in Section 5 (Equation 1) proves that $max\{k_u\}$ is the lower bound that we guarantee to all requesters. The second metric can be used to infer the lower bound on the number of communications that a server s received, that is, the maximum number of packet repetitions from a single peer. Although o may infer a lower bound to the number of communications, it will not be able to infer if a given user was the requester or a mere facilitator of the communication. Hence, $(k, 1)$-anonymity is preserved. In case that an omniscient operator employs a WiFi antenna, it could be able to observe both the cellular and WiFi channels in a given area, thus breaching the k-anonymity of

the users in that area. However, the omniscient operator has to solve a much more complex problem. This involves all challenges discussed in the Global WiFi eavesdropper scenario, including the hidden terminal problem and the difference in range between cellular and WiFi transmissions. To be successful, an adversary in the form of an omniscient operator has to employ many resources which make some of the attacks difficult to implement in practice: install WiFi antennas in strategic points for *all* areas of interest and utilize them solely for the purpose of eavedropping on *all* the available channels (each non-ovelapping channels requires yet another antenna). This constitues a significant investment of resources making the global WiFi eavesdropper a very expensive targeted attack with uncertain outcomes due to the user's mobility, the hidden terminal problem, and static or moving physical objects.

Colluding operators and WiFi eavesdroppers. This is the worst case scenario in which all parties are assumed to be malicious and colluding. In this case, we cannot provide any protection: all communications are monitored. Therefore information about both the cellular and the WiFi networks can be exposed. However, to be successful, this attack would require a malicious WiFi access point with enough range and capability of spectrum eaveasdropping or a fraction of malicious neighboring peers that shadow the user's every move. Although not infeasible, such sophisticated adversaries are highly unlikely to occur in practice for the large investements of resources they would require. Lastly, the higher the number of legitimate or non-cooperating neighboring peers, the more difficult it is for malicious peers to reach the required number of nodes to successfully breach k-anonymity.

6.2 Traditional attacks

Our anonymity scheme can be further evaluated against attacks that have been primarily defined for wired networks. Two classes of such attacks are *timing attacks* [16] and *predecessor attacks* [24]. Timing attacks [16] focus on the analysis of the timing of network messages as they propagate through the system with the intent to link them back to the real user. This class of attacks has been successful in mix-based anonymity schemes for wired networks. They require the capability to manipulate the timing of packets and monitor its propagation on the victim's path. This usually requires at least one malicous node in the victim's path. In our scheme, there is no recurrent path due the definition of our protocol to the mobility of the users. Therefore, timing attacks are not effective against our approach. Indeed, the path and its length are generated probabilistically and change at each request. This makes practically infeasible for adversaries to setup a timing attack. Finally, the latency of each hop is intrinsically noisy: wireless communication performance can change due to weather conditions, interference by other devices, and physical obstacles. The predecessor attack [24] builds on the idea that by monitoring the communication for a given number of rounds (i.e., windows in this paper), a set of colluding attackers will receive messages with a higher rate from the real requesters. This is also based on the assumption

that the real requesters communicate multiple times with the server and that are part of anonymity groups (more or less stable). Our solution is not vulnerable to the predecessor attack since, by design, our protocol does not consider groups and assumes mobile users with ephemeral connectivity. A requester u that communicates on the mobile ad-hoc network moves fast and randomly during the communication. This makes it difficult for a set of adversaries to infer information about the requester by tracking her and intercepting her traffic. Moreover, the set of neighborhood peers around u may change at two consecutive time instants, and u may be involved in several other anonymous communications. Finally, to be successful in our settings, predecessor attacks must require the availability of a great fraction of corrupted peers, which follow the requester in her every move. This scenario is equivalent to the global WiFi eavesdropper discussed in Section 6.1. Note that, also in case of a static requester u, the sorrounding peers are not able to expose the identity of u, since the broadcasted packets do not contain identifiable information. Nevertheless, communication anonymity is preserved since peers do not know the server with whom u is communicating. If we change our view by considering a predecessor attack brought by an omniscient operator o, we can counteract this attack by tuning the length of the communication window. Contrary to Crowds [19], the "path reformation" (*i.e., the definition of a set of forwarding peers including* u) does not happen each time a peer joins or leaves the set of available peers but only at the end of the window. Moreover, while in Crowds the system is aware of peers joining or leaving, in our solution no involved parties (i.e., peers, mobile network operators) have knowledge of the length of the used window at any time. This leaves the adversary with guessing as the only option and increases dramatically the effort required to identify the window expiration time and protocol re-initialization.

7 Performance evaluation

We have performed experiments to evaluate the performance of our protocol in terms of latency overhead imposed on the communication among parties. We measured the systems' performance using the Emulab (`http://www.emulab.net/`) and Orbit (`http://www.wirelessorbit.com/`) testbeds. In all of our experiments, we used devices equipped with standard IEEE 802.11 wireless network communication cards. All the results represent the average of multiple measurements (> 50) repeated over different periods of time to avoid wireless interference and transient effects from the wireless equipments. We varied the Signal-to-Noise Ratio (SNR) of the wireless link for single hop, peer-to-peer wireless connections, between 14 and 64, and we measured its impact on the link latency. As expected, our results show that the latency varies between 1ms and 52ms when the SNR is greater than 16. To characterize the behavior of a multi-hop wireless ad-hoc network, we employed the Random Waypoint [22] and the Orbit Mobility Framework [12] using city models for pedestrians. These models take into consideration mobility and interference which can degrade the signal quality. Then, we employed node mobility scenarios consisting of tens of

nodes $(5-30)$. For mobility scenarios, we varied the SNR between 24 and 64 using a timed event script, and we measured the impact of our anonymity protocol on the end-to-end latency. The overhead trend is linear with the number of hops and ranges between 28.9ms for 2-hops and 127.9 for 6-hops in average. This overhead includes both the *communication and the computational costs*. The worst case scenario, in terms of overhead we observed, was for a 6-hop network. The maximum increase in latency overhead was approximately 150ms, which is acceptable for the majority of time-sensitive streaming applications. The latency impact when selecting a 3-hop or 4-hop network is relatively low (about 50ms and 70ms, respectively). The latency results indicate that our solution does not incur prohibitive overhead or packet losses.

8 Related work

Past research addressing communication privacy in mobile networks [4, 17, 20] has been inspired by works in wired networks. Traditional solutions like TOR [8] for route anonymity and Crowds [19] for Web-communication anonymity usually assume a known network topology to create meaningful routes and use the path generated by the sender for both the request and the response. In addition, they often rely on trusted third parties (e.g., mix, onion router, blender) and on heavy multiparty computation. Other systems including I2P [13], MorphMix [21] take a different approach and provide P2P-based solutions for network anonymity. I2P [13] is an anonymizing network for secure communication that relies on tunnels and garlic routing to route data anonymously. I2P does not rely on centralized resources and does not use the same path for both the request and the response. MorphMix [21] is a P2P system for Internet-based anonymous communications, where each node is also a mix and can contribute to the anonymization process. Both I2P and MorphMix are based on heavy multiparty computation, consider wired networks, and are not able to manage mobility of the users. In general, all the above solutions are not applicable in a mobile scenario, where users move fast, form networks of arbitrary topology, and use devices with limited capabilities. Some solutions using mixes however have been designed for protecting privacy in mobile scenarios with constrained devices [11, 18]. They focused on location management and protection, rather than on identity protection, and assume the existence of trusted parties.

Existing research in the context of mobile networks mainly focused on protecting privacy in mobile and vehicular ad-hoc networks [9, 17, 20, 25], and mobile hybrid networks [1, 4]. Dong et al. [9] propose an anonymous protocol for mobile ad-hoc networks that does not rely on topological information to protect identities and the locations of the nodes. Data packets are forwarded in real and fake routes to assure random route transmission and confuse adversaries, at a price of an increased communication overhead. GSIS [17] presents a protocol, based on Group Signature and Identity-based Signature techniques, used to protect security and privacy in vehicular networks. Capkun et al. [4] provide a scheme for secure and privacy-preserving communications in hybrid ad-hoc networks

based on pseudonyms and cryptographic keys. Differently from the above works, our solution does not rely on multiparty computation, preserves the privacy of the requester also from the mobile network operators, and provides an anonymous mechanism to verify the legitimacy of the traffic produced by mobile users. Ardagna et al. [1] present a multi-path approach for k-anonymity in mobile hybrid networks. The system in this paper considerably extends and improves the work presented in [1] by: *i)* removing public key encryption except for connection establishment; *ii)* extending the communication protocol to make it resistant to intersection attacks and suitable for multiple rounds of requests-responses, *iii)* allowing the requester to assess if there is enough entropy in the system to build communication anonymity, before start sending the message, and *iv)* providing a deep analysis and evaluation of the attacker model. Similarly to our approach, the work by Ren and Lou [20] is aimed at providing a privacy yet accountable security framework. This solution, however, is based on multiparty computation and groups of users established a priori, and assumes a semi-trusted group manager and network operator.

9 Conclusions

We proposed a protocol for protecting users' privacy that harness the availability of both mobile and WiFi connectivity in current phones creating a hybrid network. Differently from traditional solutions that offer privacy protection against servers and other peers only, we assumed mobile network operators as a potential source of privacy threats. The intuition behind our approach is that while users can trust the mobile operators to properly provide network accessibility, they want at the same time to be maintained free to act in the network without feeling their activities are constantly monitored. Therefore, our solution protects the privacy of the requester from all parties involved in a communication.

References

1. C. Ardagna, S. Jajodia, P. Samarati, and A. Stavrou. Privacy preservation over untrusted mobile networks. In Bettini, S. Jajodia, P. Samarati, and S. Wang, editors, *Privacy in Location Based Applications*. Springer, 2009.
2. G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on selected areas in communications*, 18(3):535–547, 2000.
3. J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway. UMAC: Fast and secure message authentication. In *Proc. of CRYPTO 1999*, Santa Barbara, CA, USA, August 1999.
4. S. Capkun, J.-P. Hubaux, and M. Jakobsson. *Secure and Privacy-Preserving Communication in Hybrid Ad Hoc Networks*, January 2004. Tech. Rep. IC/2004/10, EPFL-IC, Lausanne, Switzerland.
5. D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.
6. V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. k-Anonymity. In T. Yu and S. Jajodia, editors, *Secure Data Management in Decentralized Systems*. Springer-Verlag, 2007.

7. C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos. Anonysense: privacy-aware people-centric sensing. In *Proc. of MobiSys 2008*, Breckenridge, CO, USA, June 2008.
8. R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proc. of the* 13^{th} *USENIX Security Symposium*, San Diego, CA, USA, August 2004.
9. Y. Dong, T. Chim, V. Li, S. Yiu, and C. Hui. ARMR: Anonymous routing protocol with multiple routes for communications in mobile ad hoc networks. *Ad Hoc Networks*, 7(8):1536–1550, 2009.
10. J. Douceur. The sybil attack. In *Proc. of IPTPS 2002*, Cambridge, MA, USA, March 2002.
11. H. Federrath, A. Jerichow, and A. Pfitzmann. Mixes in mobile communication systems: Location management with privacy. In *Proc. of the First International Workshop on Information Hiding*, Cambridge, U.K., May-June 1996.
12. X. Hong, T. Kwon, M. Gerla, D. Gu, and G. Pei. A mobility framework for ad hoc wireless networks. In *Proc. of MDM 2001*, Hong Kong, China, January 2001.
13. *I2P Anonymous Network*. `http://www.i2p2.de/`.
14. J. Kong and X. Hong. ANODR: Anonymous on demand routing with untraceable routes for mobile ad-hoc networks. In *Proc. of MobiHoc 2003*, Annapolis, MD, USA, June 2003.
15. T. Krovetz. UMAC: Message authentication code using universal hashing. RFC 4418 (Informational), March 2006.
16. B. Levine, M. Reiter, C. Wang, and M. Wright. Timing attacks in low-latency mix systems (extended abstract). In *Proc. of FC 2004*, Key West, FL, USA, February 2004.
17. X. Lin, X. Sun, P.-H. Ho, and X. Shen. GSIS: A secure and privacy preserving protocol for vehicular communications. *IEEE Transaction on Vehicular Technology*, 56(6):3442–3456, November 2007.
18. M. Reed, P. Syverson, and D. Goldschlag. Protocols using anonymous connections: Mobile applications. In *Proc. of the 5th International Workshop on Security Protocols*, Paris, France, April 1997.
19. M. Reiter and A. Rubin. Crowds: Anonymity for web transactions. *ACM TISSEC*, 1(1):66–92, 1998.
20. K. Ren and W. Lou. A sophisticated privacy-enhanced yet accountable security framework for metropolitan wireless mesh networks. In *Proc. of ICDCS 2008*, Beijing, China, June 2008.
21. M. Rennhard and B. Plattner. Introducing MorphMix: peer-to-peer based anonymous internet usage with collusion detection. In *Proc. of WPES 2002*, Washington, DC, USA, November 2002.
22. A. Saha and D. Johnson. Modeling mobility for vehicular ad-hoc networks. In *Proc. of VANET 2004*, Philadelphia, PA, USA, October 2004.
23. P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.
24. M. Wright, M. Adler, B. N. Levine, and C. Shields. The predecessor attack: An analysis of a threat to anonymous communications systems. *ACM TISSEC*, 7(4):489–522, 2004.
25. Y. Zhang, W. Liu, W. Lou, and Y. Fang. Mask: Anonymous on-demand routing in mobile ad hoc networks. *IEEE Transaction on Wireless Communications*, 5(9):2376–2385, September 2006.