Providing Users' Anonymity in Mobile Hybrid Networks

CLAUDIO A. ARDAGNA, Università degli Studi di Milano
SUSHIL JAJODIA, George Mason University
PIERANGELA SAMARATI, Università degli Studi di Milano
ANGELOS STAVROU, George Mason University

We present a novel hybrid communication protocol that guarantees mobile users' 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 a 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, which possibly collude to breach user's anonymity or disrupt the communication. We also describe how a micropayment scheme that suits our mobile scenario can provide incentives for peers to collaborate in the protocol. Finally, we evaluate the network overhead and attack resiliency of our protocol using a prototype implementation deployed in Emulab and Orbit, and our probabilistic model.

Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; C.2.2 [**Computer-Communication Networks**]: Network Protocols; K.4.1 [**Computers and society**]: Public Policy Issues

General Terms: Security, Management, Reliability

Additional Key Words and Phrases: Anonymity, Mobile Hybrid Networks, Privacy

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. Governments and providers of mobile and online services in fact often collect mobile information that is not strictly needed for service release, and use it to track and profile users' activities. An improper management of location and communication information can then open the door to breaches violating the private sphere of the users (e.g., [Allan and Warden 2011; Ardagna et al. 2011b; Bettini et al. 2009; Cheng 2011]). This scenario has

---

This work was supported in part by the Italian Ministry of Research within PRIN 2010-2011 project "GenData 2020" (2010RTFWBH); by Google under the Google Research Award program; 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.
Authors' addresses: Claudio A. Ardagna and Pierangela Samarati, Dipartimento di Informatica, Università degli Studi di Milano, 26013 Crema, Italia; Sushil Jajodia and Angelos Stavrou, Center for Secure Information System, George Mason University, Fairfax, VA 22030-4444, USA.

DOI:http://dx.doi.org/10.1145/0000000.0000000

sparked a renewed interest in the problem of providing continuous network connectivity while guaranteeing privacy 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 [Chaum 1981], Onion Routing [Dingledine et al. 2004], and Crowds [Reiter and Rubin 1998] were geared towards protecting the network anonymity of the users, preventing an adversary from linking the user to a service request. All these solutions, which were designed with traditional wired networks in mind and shared the implicit assumptions on the stability of the routing configuration and network topology, are not applicable in mobile networks where users can move and change position over time. Approaches addressing the privacy problem in mobile ad-hoc networks (e.g., [Kong and Hong 2003; Lin et al. 2007]) have been mostly aimed to provide anonymous routing protocols and typically rely on expensive multiparty computation. In addition, other privacy proposals for mobile networks (e.g., [Ardagna et al. 2011a; Shin et al. 2011]) have addressed the problem of protecting location information privacy and users' anonymity against the services they access. These proposals assume a trusted mobile network operator to be in a privileged position and able to observe all the communications of the users. This scenario, however, puts the privacy of the users at high risk.

In this paper, we study the above privacy problem 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. We consider 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 capability 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, does not require additional hardware, and considers communication patterns mainly composed of short windows of communication. Peers collaborating in providing anonymity to others are anonymously rewarded using a micropayment scheme, which is adapted to the hybrid scenario we consider and minimizes the probability of fake coins and double spending. There is therefore an incentive for peers to cooperate in the protocol.

Addressing a novel threat and problem, our work is 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 communications or cryptographic operations including the use of multiparty computation and peer authentication, and we limit the use of public key cryptography. Instead, we introduce a new fast packet filtering that leverages pseudorandom 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. In [Ardagna et al. 2010] we presented an early version of our proposal that here has been extended with a refined protocol that *i)* anonymously communicates to peers their role in the anonymization protocol and *ii)* supports a communication setup that anonymously evaluates if there are enough peers willing to collaborate in the neighborhood of the requester before starting the anonymous communications. Also, the adversarial analysis

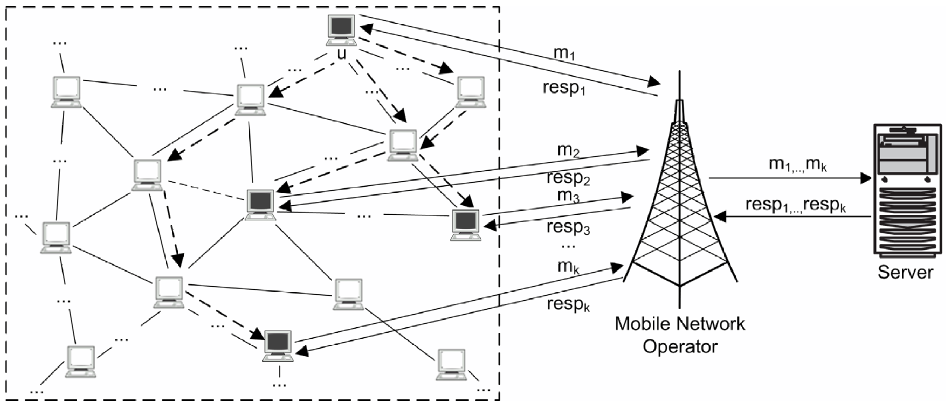


Fig. 1. Mobile hybrid network.

has been extended by considering colluding operators that exploit external knowledge and active adversaries that try to attack the system by dropping packets. We have further extended our solution with a micropayment scheme for mobile communications providing incentives for peers to participate in the anonymization protocol. Finally, the performance evaluation, originally proposed in [Ardagna et al. 2010], has been extended using our probabilistic model to analyze and evaluate the resilience of our solution against malicious attacks.

The remainder of this paper is organized as follows. Section 2 presents the scenario we consider and introduces the problem. Section 3 discusses the rationale and basics of our approach. Section 4 presents our communication protocol guaranteeing user's anonymity. Section 5 describes how users can maintain their anonymity in presence of malicious peers. Section 6 provides the analysis of our protocol against adversarial attacks aimed at compromising the privacy and functionality offered by the protocol. Section 7 shows our scheme for micropayment. Section 8 presents our experimental results. Section 9 discusses related work and, finally, Section 10 presents our conclusions.

2. PROBLEM DEFINITION AND WORKING ASSUMPTIONS

Our reference model is a distributed and mobile infrastructure which forms a hybrid network, integrating *wireless*, *cellular*, and *wired* connections. The participating entities (see Figure 1) 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 between 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, pairs $\langle$*user,server*$\rangle$ describing service accesses. We assume communications composed of short TCP sessions and few steps of requests and responses, including traditional Web browsing (e.g., Google requests), social network activities, posts in blogs, small file uploads, and the like. These communication patterns capture many of the everyday mobile interactions between users and servers [Rahmati et al. 2010], and properly model activities in critical and emergency scenarios. The consideration of short sessions permits to achieve a good balance between service usability and privacy protection, allowing users to access many services in a privacy-preserving way. Privacy protection is enacted by involving, in the communication with the mobile operator, other peers (users) with whom the user communicates via the WiFi network. Communications in the WiFi network are anonymized, meaning that all identifiable information in the communication packets is removed or not accessible to other peers in the 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 2.1 (*(k, h)-anonymity*). A communication is said to be (k, h)-anonymous against an adversary v, if v cannot relate the communication to less than k users and h servers.

A communication is (k, h)-anonymous against an adversary v, if the probability for v 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. We consider all participating entities (i.e., peers, operators) accessing a portion of the communication between users and servers as potential adversaries, which may try to break the anonymity protocol. We also assume a global WiFi eavesdropper that observes all WiFi communications and tries to subvert the anonymity of our protocol. In general, 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 [Ciriani et al. 2007; 2009; Samarati 2001] set by the user and provided by our protocol. Since the focus of our work is the protection of a 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 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 the adversary. It is considered protected if it is (k, h)-anonymous with $max(k, h) > 1$.

All adversaries might collude and combine their knowledge to build a more powerful adversary that attacks the privacy of the users. Colluding adversaries slightly change the definition of degree of anonymity as follows.

Definition 2.2 (*(k, h)-anonymity for colluding adversaries*). Consider n adversaries v_1,v_2,...,v_n. Suppose that the degree of anonymity of a communication is (k_i, h_i) against v_i, with i=1,...,n. The degree of anonymity enjoyed by a communication against a colluding adversary v=$v_1 \oplus v_2 \oplus \ldots \oplus v_n$ is *at most* (k, h)=$(\min(k_i), \min(h_i))$, with i=1,...,n.

We note that the above definition represents an upper bound to the degree of anonymity (k, h) that is preserved against v. We also note that in addition to the above eavesdropping adversaries, which aim to breach the privacy of the users, the involvement of peers in the communications between the user and the server may open the door to active attacks, which aim to disrupt the normal operation of the system. The adversary v can therefore be a peer (or a set thereof), which attacks the protocol by dropping or falsifying received packets. This scenario requires a solution allowing successful communications also when a given fraction of peers is malicious and active.

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 (where k is the anonymity degree the originating user wishes to enjoy) and randomly distributed to k distinct peers in the WiFi network for their forwarding to the mobile network operator. The distribution of the packets among peers and towards the operator is driven by a probability P_f of forwarding that introduces randomness in the process, granting heterogeneous distribution of the peers collaborating in the forwarding of a message. To preserve the anonymity of the transmission process, the distribution of the packets starts as soon as the user verifies, through the WiFi network, the availability of enough distinct peers in the communication range, such that, for each message to a service provider, the network operator will receive k indistinguishable packets from k different recipients (Figure 1). These packets are sent to the server, which originates a response for each of them, each delivered to the corresponding peer. Peers different from the original user will not be able to decipher the response content. 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 servers, based on the seeds agreed with their users, concurrently fill in a shared, global table LEGITIMATE with pairs (R^1, R^2) of pseudo-random numbers. Upon a packet arrival, the mobile network operator retrieves the pseudo random number attached to the packet and performs a lookup to table LEGITIMATE to verify the packet validity. Access to table LEGITIMATE is provisioned by means of a distributed service, which needs to support shared access by multiple operators and high performance. We note that the cloud infrastructure provides the suitable environment for implementing such a service and permits to manage table LEGITIMATE in a profitable way. In addition, the great amount of users and servers may rapidly bring to a scenario that severely stresses the storage system of the service. However, we note that, in our solution, the cost of maintaining table LEGITIMATE 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 requirement is approximately 1 TB which can be easily maintained by today off-the-shelf disks. The use of an external service can then eliminate the need for a pre-storage of the random numbers, since this service 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 and replay attacks.

Table LEGITIMATE 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. To enforce integrity verification, we employ the UMAC [Black et al. 1999] 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 [Krovetz 2006]. 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 UMAC 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), thus preventing Denial-of-Service (DoS) attacks.

Finally, 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. Also, each server s locally maintains a table ORIG_{sid} for each session *sid*, which stores the original set of peers (including u) involved in the connection establishment. Each peer p maintains the following tables: SENT_p contains the identifiers of the communications that the peer has helped distributing by forwarding a packet to the mobile operator in the connection establishment (including those originated by the peer); $\text{MYPRN}_{p,seed}$ contains the set of pseudo-random numbers prn_i=(R_i^1, R_i^2) generated by p using *seed* shared with the corresponding server. $\text{MYPRN}_{p,seed}$ contains the same prn generated by the server and is then a subset of table LEGITIMATE.

4. PROTOCOL

We present the working of the communication protocol distinguishing management of requests and responses. 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. Also, we will use standard notation $E_K^s()$ and $D_K^s()$ to denote symmetric encryption and decryption operations with key K, whereas $E_K^p()$ and $D_K^p()$ denote public key operations. Each communication is composed of a *connection establishment* phase in which the user and the server setup the communication session, and a subsequent *service access* phase in which the real communication is carried out and the user possibly gains access to the requested service. In our protocol, connection establishment 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 selected peers, which contains user u, in the case of connection establishment request; *ii)* within the content of the message, which contains the key for the session, in the case of connection establishment request, and the id of the session, in the case of service access request. Also, the body of the connection establishment request packet is encrypted with the server's public key, while the body of the service access request packet is encrypted with the session key to which the request refers. Note that public key operations are *used only for communication establishment*. Finally, for each service access request, the response is also returned to peers in ORIG_{sid}.

Figure 2 illustrates the protocol operations at the different participating parties. Figure 3 illustrates the distribution of packets among parties illustrating also how the content of the packets changes. Big white arcs refer to communications over the WiFi network (among peers), arcs with a black line refer to communications over the cellular network (between peers and mobile operators), arcs with a black 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

REQUEST ($u \rightarrow s$)
User $u \in \mathcal{P}$
u1.1 Let m be the message to be sent and *payload* its content, k the anonymity preference,
$(1 - P_f)$ the probability of forwarding to the operator, *cid* the communication identifier,
r a random number, and UMAC$_R$ a Universal Message Authentication Code (UMAC) using key R
u1.2 Generate a random message identifier *mid* and obtain 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 *seed* and a random number r_i
to_i:=R_i^1
if the message is a connection establishment request
then generate session key *SK*
$body_i$:=$E^p_{P_s}(id_u, seq_i, payload_i, SK, mid, tmp, r_i, cid)$ /*connection establishment*/
else $body_i$:= $E^s_{SK}(id_u, seq_i, payload_i, sid, mid, tmp, r_i, cid)$ /*service access*/
u1.5 Wait until enough 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,$body_i$,UMAC$_{R_i^2}\{body_i\}$,r_i,*cid*] to p_i in the WiFi network
u1.7 **if** ((*cid*,$-$)$\notin$SENT$_u$)
then SENT$_u$:=SENT$_u \cup$(*cid*,r_1)
With random delay, forward [to_1,$body_1$,UMAC$_{R_1^2}\{body_1\}$] to o over the cellular network
else Send m_1:=[to_1,$body_1$,UMAC$_{R_1^2}\{body_1\}$,r_1,*cid*] to p_1 in the WiFi network

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

Operator $o \in \mathcal{O}$
Upon receiving [to,$body$,UMAC$_{R^2}\{body\}$] from peer p
o1.1 **if** (($to \in$LEGITIMATE) and (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 $D^p_K(body) \vee D^s_K(body)$ with K:=S_s $\vee$ K:=*SK*, respectively
s1.2 ORIG$_{sid}$=ORIG$_{sid} \cup$(id_p,o, r) /*connection establishment*/
s1.3 Assemble 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
s2.1Let *payload* be the response, *sid* be the session id, and *SK* the session key
s2.2**for each** $e_i \in$ORIG$_{sid}$ with i=1, ..., k
$body_i$:=E_{SK}(*payload*,*sid*,*tmp*)
Send [$e_i.id_p$,$body_i$,$e_i.r$] to $e_i.o$
s2.3**for** j:=1... k **do** /*service access*/
Let id_{p_j} and o_j be the peer id and the operator of the j-th packet of message *mid*
$body_j$:=$\emptyset$
Send [id_{p_j},$body_j$,$\hat{r}_j$] to o_j, with $\hat{r}_j$=*cid* $\oplus$ r_j

Operator $o \in \mathcal{O}$
Upon receiving [id_p,$body$,$\hat{r}$] from s
o2.1 Forward [$body$,$\hat{r}$] to p

User/Peer $p \in \mathcal{P}$
Upon receiving [$body$,$\hat{r}$]
up2.1 **if** (($-$,$\hat{r}$)$\notin$SENT$_p$)
then SENT$_p$=SENT$_p$ $-$ (*cid*,r), s.t. r=*cid* $\oplus$ $\hat{r}$
else if ((*cid*,$\hat{r}$) belongs to a message sent by p)
then retrieve response as $D^s_{SK}(body)$
else drop the packet

Fig. 2. Communication protocol.

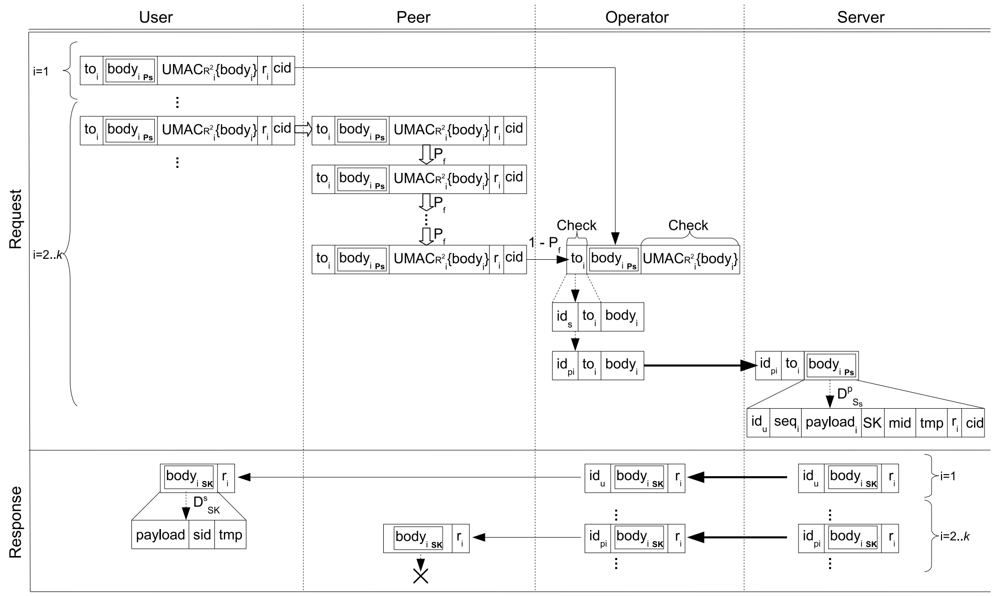


Fig. 3. Flow of packets within our protocol.

appearing in the lower right corner of the box. Packets in Figure 3 refer to connection establishment.

4.1. Request

For each session, a user can specify a privacy degree k to be guaranteed for all communications (connection establishment and service access requests related to the session) and a 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 in ORIG_{sid} can send to the operator in each communication.

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, r a random number, 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 the identifier for the message, and obtains 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 *seed* agreed with the server, a 64-bit 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 field to_i of packet m_i. Body $body_i$ of each packet to be sent, composed of user id id_u, sequence number seq_i of the packet, packet payload $payload_i$, message identifier *mid*, timestamp *tmp*, random number r_i, communication identifier *cid*, and either session key *SK* to be used for subsequent communication in the session (for connection establishment requests), or session identifier *sid* (for service access requests), is then encrypted. Encryption is performed with server's public key P_s in case of connection establishment requests and with symmetric session key *SK* in case of service requests. 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. Finally, u appends r_i and *cid* to each packet. Random number r_i and communication identifier *cid* are used to let peers know whether they are part of the connection establishment phase without exposing this information to other peers. To avoid intersection attacks, in fact, peers in ORIG_{sid} (i.e., peers involved in the connection establishment request) must not communicate with the server in subsequent requests with the same *cid*. Therefore, each packet m_i composed of [to_i,$body_i$,$\text{UMAC}_{R_i^2}\{body_i\}$,$r_i$,*cid*], with i:=$1,\ldots,k$, is sent to a different peer in the communication range. In the case of connection establishment (i.e., (*cid*,$-$)$\notin \text{SENT}_u$, where $-$ denotes any value), the first packet m_1 is managed by u herself, that adds pair (*cid*,r_1) 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. To avoid infinite loops or privacy breaches in the distribution process, the user should verify through the WiFi channel if enough collaborating peers are available in her proximity (see Section 5.2 for more details). If this is not the case, the user will not send the packet until enough peers become available.

Peer. Upon receiving a packet [to,$body$,$\text{UMAC}_{R^2}\{body\}$,$r$,*cid*], each peer p checks if it is part of ORIG_{sid} for the same communication (i.e., (*cid*,$-$)$\in\text{SENT}_p$). If it is, p sends the packet unchanged to a peer in the communication range. Otherwise, p sends packet [to,$body$, $\text{UMAC}_{R^2}\{body\}$] to its operator o with probability $(1 - P_f)$ and adds pair (*cid*,r) to SENT_p; while with probability P_f, it sends the packet unchanged to a peer in the communication range. We note that, to avoid exposing u's identity, the distribution of packets to the operator must be synchronized with the random delay introduced by u in sending the first packet of a connection establishment phase. If it is not, in the worst case, operator o can identify sender u by observing who transmits first. We will describe an approach to synchronization in Section 5.1.

Operator. Upon receiving a packet [to,$body$, $\text{UMAC}_{R^2}\{\text{body}\}$] from a peer p, the operator uses R^1 in field to to retrieve pair (R^1, R^2) in 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 s. The remote server s is identified as the one that provided pair (R^1, R^2) associated with the packet. Also, pair (R^1, R^2) is removed from 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 them to quickly determine the key to be used in body decryption.

Server. Upon receiving a packet [id_p,to,$body$] from operator o, using field to, the server determines 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,r) to its local table ORIG_{sid} and uses *cid* and r stored in the packet forwarded by id_p to notify the peer that it is part of 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.

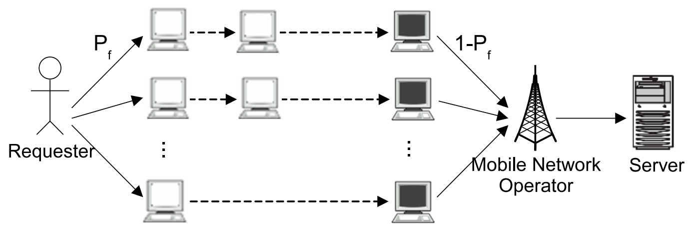


Fig. 4. Multi-path configuration.

Server. Let *payload* be the response to be sent, *sid* be the identifier of the session it refers to, and *SK* be the corresponding session key. Body $body_i$ of the response is determined by encrypting, with *SK*: *payload*, *sid*, and timestamp *tmp*. The server then sends [$e_i.id_p$,$body_i$,$e_i.r$] to each peer $e_i \in$ORIG$_{sid}$ via operator $e_i.o$. 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). For each packet related to message *mid*, received from peer p_j via operator o_j in service access communication, a response [id_{p_j},$body_j$,$\hat{r}_j$] is also sent to p_j via o_j, with $body_j$=$\emptyset$ and $\hat{r}_j$=*cid* $\oplus$ r_j.

Operator. Upon receiving a response packet [id_p,$body$,$\hat{r}$], the operator forwards [$body$,$\hat{r}$] to peer p.

User/Peer. Upon receiving a response packet [$body$,$\hat{r}$] each peer p (including u) first determine if ($-$,$\hat{r}$) belongs to SENT$_p$. If it is not, the peer was not involved in the connection establishment and deletes pair (*cid*,r), such that r=*cid* $\oplus$ $\hat{r}$, from SENT$_p$. Otherwise, if *cid* is the identifier of the message for which the peer was initiating user u, the peer determines the decryption key thus retrieving $body$ accordingly. Else, the peer drops the packet.

5. COMMUNICATION MANAGEMENT IN MALICIOUS ENVIRONMENTS

Each user establishing a communication with a server anonymously involves k peers (i.e., generates k packets) to achieve k-anonymity. In a malicious environment, k can be increased to N to provide resilience against malicious peers. As a consequence, before any communication begins, the user needs to anonymously evaluate if enough peers willing to collaborate are available in her neighborhood, that is, the area identified by the WiFi communication range and the mean number of hops taken by a packet in the WiFi network based on probability P_f of forwarding. In this section, we first discuss how a user can establish the number N of peers based on a possible adversarial environment and how to implement peer synchronization in the distribution of packets towards the cellular network (Section 5.1), we then propose a communication setup phase that allows the requester u to start the protocol only if k-anonymity can be preserved (Section 5.2), we finally discuss how the length of the communication window can influence the behavior of our protocol against position attacks (Section 5.3).

5.1. Anonymity preference setup and peer synchronization

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. 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 (see Figure 4), each peer

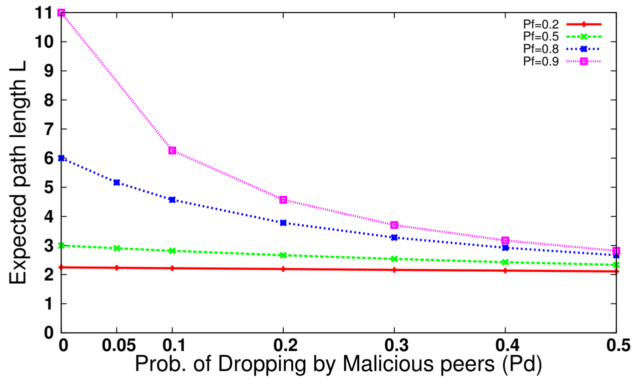


Fig. 5. Expected path length in terms of forwarding probability (P_f) and probability of malicious peers (P_d).

receiving a packet either forwards it to a random next-hop peer with probability P_f of forwarding (white PCs) or to the network operator with probability $(1 - P_f)$ (black PCs). Different packets of the same message follow different paths (that can be partially overlapped). Thus, like in [Reiter and Rubin 1998], we can derive the expected path length in a 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 [Douceur 2002] where a malicious peer can assume multiple false identities by pretending to have multiple WiFi physical occurrences. We assume that some 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 packets can then be calculated as: L=$\frac{(1-P_d)P_f}{1-(1-P_d)P_f} + 2$. Expected path length L in a malicious environment changes with P_d and P_f as shown in Figure 5. L increases as P_f increases, whereas it decreases as P_d increases.

Probability P_d of malicious peers also affects the number of packets that u must distribute to achieve a successful and anonymous communication. In principle, at each communication round, u must distribute at least N packets such that k=$N \cdot (1 - P_d)^L$, where $N \cdot (1 - P_d)^L$ indicates the expected number of successfully forwarded packets to the operator even in the presence of a fraction P_d of malicious peers. We note that, in case of a non-malicious environment (i.e., P_d=0), N=k. We also note that the value of N must be an integer. As a consequence, to ensure communication anonymity in each context, the value of N is finally generated by rounding up the real number calculated using the equation k=$N \cdot (1 - P_d)^L$.

Based on path length L and communication overhead in Section 8.1, we can define an approach that allows the users and their supporting peers to synchronize the distribution of their packets to the operator, thus avoiding the scenario in which the operator identifies a sender by simply observing who transmits a packet first. To this aim, all peers p are pre-configured with the same time t_0, which represents the beginning of all activities. Given path length L and an average latency overhead $\overline{ov}$, the time axis is divided in intervals (t_i,t_{i+1}) of length $L \cdot \overline{ov}$. Suppose know that u starts the commu-

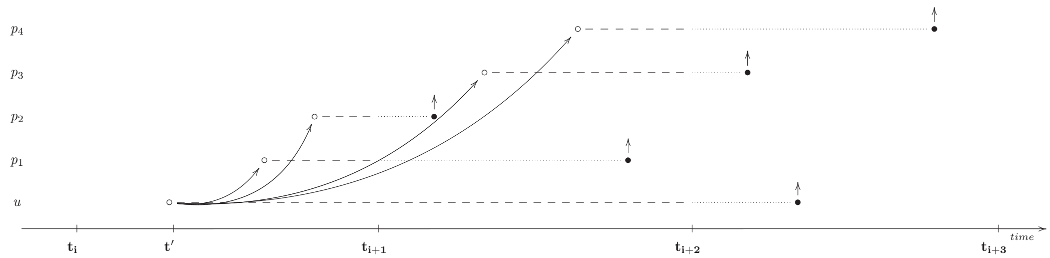


Fig. 6. Packet forwarding synchronization.

nication at time $t' \in (t_i, t_{i+1})$; based on L and $\overline{ov}$, all packets are received and accepted to be sent to o by supporting peers in (t_i, t_{i+2}). Each supporting peer p_j waits until the next interval begins and sends the packet with a random delay chosen in $[0, L \cdot \overline{ov})$; similarly, the sender u randomly selects (t_{i+1}, t_{i+2}) or (t_{i+2}, t_{i+3}) as her sending interval, and forwards the packet with a random delay again chosen in $[0, L \cdot \overline{ov})$. Operator o is therefore not able to infer the identity of u by observing the ordering of the received packets. Figure 6 shows an example in which user u defines k=5 and starts sending four packets of her message in the interval (t_i, t_{i+1}). For simplicity, in Figure 6, we assume that each peer receiving a packet, sends it directly to the operator. We denote with curved black arrows the distribution of packets in the WiFi network and with vertical black arrows the distribution of packets in the cellular network. All four packets are accepted, and ready to be sent to o, by peers p_1, p_2, p_3, and p_4 (white dots) in (t_i, t_{i+2}). Peers p_1 and p_2 (black dots), which accepted to send a packet in (t_i, t_{i+1}), wait until the next interval begins (dashed line) and forward their packet with a random delay (dotted line) in (t_{i+1}, t_{i+2}). Similarly, peers p_3 and p_4 (black dots), which accepted to send a packet in (t_{i+1}, t_{i+2}), wait until the next interval begins (dashed line) and forward their packet with a random delay (dotted line) in (t_{i+2}, t_{i+3}). User u randomly picks up the interval (t_{i+2}, t_{i+3}) (dashed line), and forwards her packet in (t_{i+2}, t_{i+3}) with a random delay (dotted line).

5.2. Communication setup

Complex communications composed of several message exchanges between requester u and server s open the door to possible intersection attacks by which an observer can exploit the fact that a given requester appears in different messages directed to a server. To counteract intersection attacks, our protocol ensures that requester u as well as peers in ORIG_{sid} participate only in the delivery of one request message, while they will receive all the responses in the communication (i.e., the server will send each reply to the original senders in ORIG_{sid}). Peers not in ORIG_{sid} instead can be involved in any subsequent request. In other words, to protect the anonymity of the entire communication, u must involve at least $2k$-1 peers, that is, before starting a communication with a server s, it needs to evaluate, based on probability P_d, if at least $2k$-1 peers are willing to forward packets to operator o. This phase, called *communication setup*, must not reveal pair $\langle u,s \rangle$, must be resistant to intersection attacks, and must be integrated within the protocol discussed in Section 4.

We present the working of the communication setup that is composed of a single round of request-response. We will use s_s to denote the *setup server* that, given as input a set of packets and anonymity preference k, evaluates if there are enough collaborating peers to start an anonymous communication. In case enough peers are available, s_s randomly selects the peers that will be part of ORIG_{sid} and notifies them using the protocol in Section 4. As for each server, the user has to register and agree upon a secret key with s_s. This pre-established secret key is again used as a seed by the user to generate pseudo-random numbers to be associated with packets.

Request. Using the protocol in Section 4, requester u distributes N packets of the form [to_i,$body_i$,UMAC$_{R_i^2}$\{$body_i$\},r_i,*cid*] such that $2k$-1=$N \cdot (1-P_d)^L$, and sends an additional packet itself to s_s. The goal of u is to evaluate using s_s how many collaborating peers are in her neighborhood.

Response. Upon receiving a packet by a peer p, s_s uses field to to decrypt $body$ and to access *mid*. For each successful decryption having the same *mid*, s_s adds *id*$_p$, r, and *cid* to ID$_p$. If $|$ID$_p| \leq 2k$-1, s_s stops the protocol and no inference can be done on u's communications; otherwise, s_s populates ORIG$_{sid}$ by randomly selecting peers (including u) in ID$_p$ and, as in the original protocol, let them know in the response that they are part of ORIG$_{sid}$. The response packet returned by s_s to each collaborating peer will also contain the set ORIG$_{sid}$ encrypted with the public key of the communication server s (i.e., $E^p_{P_s}$(ORIG$_{sid}$)).

Among all peers receiving the response, only u can decrypt it using the secret key *SK* agreed with s_s; based on the response, u either stops the communication or starts the real communication with s using the original protocol modified as follow.

(1) In the connection establishment, u adds $E^p_{P_s}$(ORIG$_{sid}$) to the body of each packet and, similarly to a service access phase, does not involve herself and peers in ORIG$_{sid}$ in the forwarding to o.
(2) Upon receiving a packet, s decrypts $body$ and ORIG$_{sid}$. Since collaborating peers in ORIG$_{sid}$ have been selected by s_s, they never send a packet to s, but they always receive the response.

In summary, the communication setup allows requester u to anonymously evaluate if there exist enough peers to start an anonymous communication with a server s, stopping the protocol if there are not at least $2k$-1 peers available. In addition, since requests in the communication setup are indistinguishable by connection establishment and service access requests in the original protocol, malicious peers cannot behave correctly only when requests in the communication setup are observed. Moreover, no intersection attacks are possible by operators observing two consecutive requests to s_s and s, because peers in ORIG$_{sid}$ are selected by s_s among peers in ID$_p$. We note that the above approach can be also adopted to account for eavesdroppers that notify the operators that they are part of ORIG$_{sid}$ while they are not the requester. In this case, probability P_d of malicious peers should also considers the possibility of peers acting as eavesdroppers. This results in a set ORIG$_{sid}$ that contains more that k users.[1]

The price we pay for an increased resilience against malicious dropping and scenarios with few peers participating in our protocol is low. In fact, the communication setup is performed once at the beginning of the session, and only requires N additional packets and an additional step of public key encryption.

5.3. Window of communication

Above in this section, we discussed how the proposed anonymization protocol has been designed to be robust against intersection attacks and to preserve anonymity even if the number of peers in the neighborhood of the requester is not sufficient to provide k-anonymity.

There is however an additional subtlety that we need to consider when we evaluate our protocol. Each communication is characterized by a communication window (i.e.,

[1]In the following, for the sake of clarity, we consider P_d as the probability of dropping.

a session) that can be defined as the time interval in which a communication is completed. This window starts as soon as the connection between the user and the server is established, and finishes when the connection is closed or ends unexpectedly. During this communication window, a requester u moves while involving different sets of peers in her neighborhood (one set for each message to be sent in the communication). These sets, which differ depending on the mobility pattern of the peers, have a single common characteristic: all selected peers are located within an area around u, which is bounded by the WiFi communication range and expected path length L. For large windows of communication, there is the risk that a mobile operator can re-identify the requester. The operator in fact has available both information about the movements of each peer and all communication patterns. Assuming a single communication (worst case), the mobile operator can infer with a good approximation requester u, since u is likely to be the only peer that will be located around the peers involved in each communication step (i.e., position attack).

In general, although the window length may seem a critical factor affecting the robustness of our solution, this is not the case for many communication scenarios. As discussed in Section 2, short windows of communication (few seconds) characterize many of everyday mobile communication sessions [Rahmati et al. 2010], and correctly represent the communication behavior in emergency and critical scenarios. Therefore, considering short windows of communication that involve few rounds of request-response, the extent to which peers can move during the communication does not expose the anonymity of our protocol to position attacks. Hence, our solution balances the need to communicate of the users and the need to preserve the privacy of the involved parties. In the following of this paper, we evaluate our protocol in an adversarial environment assuming short communication windows. We let the consideration of complex and large windows to our future work.

6. ADVERSARIAL ANALYSIS

To evaluate our approach, we analyze the impact of potential attacks against (k, h)-anonymity from adversaries with different capabilities, type, and access to information. Adversaries try to breach the communication (k, h)-anonymity or attack the availability of the overall system by attempting to corrupt the anonymity protocol. In addition, we discuss more sophisticated attacks that involve active disruption or collusion between different entities participating in the protocol.

In the following, we present an analysis on the anonymity of our protocol against attacks by *individuals* or *colluding adversaries* eavesdropping on the communication (Section 6.1), *active adversaries* trying to disrupt the communication (Section 6.2), as well as against *timing* and *predecessor attacks* (Section 6.3). We do not consider servers as part of our threat model, because the network identity of the users is assumed to be known to the servers.

6.1. Communication eavesdropping

We assume that all participating entities in our system can play the role of adversary that eavesdrops the communication. We then evaluate the anonymity provided by our protocol against such adversaries.

Operator. A single operator o can only observe the communications involving peers that use o to forward their messages over the cellular network. Our system is designed to prevent o from identifying user u of a request below the k-anonymity threshold that the user selects. Since u may not be subscribed to o, o is not able to identify the packets sent by u. In addition, the communication data are not revealed to o because it does not have access to the cryptographic keys required to decrypt the packet/flow payload. 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, h)-anonymity [Choia et al. 2007]. To this aim, it follows the rule that the more the packets sent by a peer p in the WiFi network, the more is p's probability of being the originating user.[2] 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 and the network identity of the peers is hidden since no identifiable information is added in the WiFi communication. As a consequence, the WiFi eavesdropper cannot identify originating user u. Even in case we assume the identity to be disclosed, a WiFi eavesdropper, short of breaking the cryptographic keys, cannot extract any information regarding o and s. In fact, it neither receives the responses from the server (which are communicated via the cellular network) nor knows the identity of server s. Hence, $(1, *)$-anonymity is preserved in the worst case.

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.

Neighboring peers. Another avenue of attack is to simulate a global WiFi eavesdropper employing "shadowing" neighboring peers that surround the victim. This attack is a special case of a global WiFi eavesdropper with the addition of the capability to receive the cellular message reply, which is sent to all peers sending a packet of the communication to server s. However, on their own, these nodes do not have cryptographic access to message content both in setup, connection establishment, and service access sub-protocols. Also, due to the broadcast nature of the wireless communications, every WiFi peer overhearing on the communications cannot assume that each packet forwarded by u, is originated by u herself, due to physical barriers that might prevent a clear transmission to be overheard, in addition to the hidden terminal problem that exists in all IEEE 802.11 communications [Bianchi 2000]. 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. Moreover, given that in our protocol packets need not to be manipulated by intermediate peers, there is no need to add identity or identifiable information to the packets in clear [Choia et al. 2007]. Thus, the adversary is not able to infer who is the peer sending a packet, unless there is a single peer in the communication range that is also physically visible by the adversary. Finally, our solution based on communication identifier *cid*, random number r_i, and pseudo-random numbers (R_i^1, R_i^2) prevents peers by inferring information on the server identity and the composition of ORIG_{sid}. $(k, *)$-anonymity is therefore preserved against single peers.

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 anonymity from such o, and thus, o can observe all packets header information for a given time interval. Therefore, for each communication, o receives a set of packets

[2]We note that the broadcast nature of WiFi communications can confuse the WiFi eavesdropper, making its guess probabilistic when different communications are in place and overlap.

M=$\{m_{p,s,t}\}$, where p denotes the peer forwarding the packet, s the server to which the packet has been delivered, and t the o's packet timestamp. Operator o can place the observed packets in two main sets, one for the requests $M_{\mathcal{P},\mathcal{S}}$ and one for the responses $M_{\mathcal{S},\mathcal{P}}$.

Considering $M_{\mathcal{P},\mathcal{S}}$, operator o can group request packets having the same p, the same s, 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. Based on these sets o can extract two metrics for inference: *i)* the number of packets transmitted by *unique* peers to a server 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 forwarding peers assuming that o receives all the packets from all the mobile peers and knows setup server s_s. The second metric cannot be used for inference since peers not in ORIG_{sid} are potentially involved in many forwards of the same communication.

Similarly to the case of $M_{\mathcal{P},\mathcal{S}}$, if o observes $M_{\mathcal{S},\mathcal{P}}$, it can only infer a set of peers that receive replies by a server without any inference on pair $\langle$*user,server*$\rangle$.

When the operator mixes information from $M_{\mathcal{P},\mathcal{S}}$ and $M_{\mathcal{S},\mathcal{P}}$, it can identify in the worst case of a single communication exactly the peers in the original set ORIG_{sid}. In fact o can observe peers that receive responses although they did not send requests. However, o cannot reduce the anonymity set to less that $|\text{ORIG}_{sid}|$, and then $(k, 1)$-anonymity is preserved.

An omniscient operator can also exploit external knowledge to breach the user's anonymity. As an example, o can use the position of each peer joining the cellular network in its attempt to reduce the anonymity of the users (i.e., position-based attack). While it is difficult for o to know the exact positions of the peers, o can estimate them with high accuracy using data already available in the network. Several works described and discussed location technologies and the best accuracy that can be achieved [Gustafsson and Gunnarsson 2005; Sun et al. 2005]. These studies and more recent solutions (e.g., [Anisetti et al. 2011]) have proven that, the positioning process can achieve reliable location accuracy (e.g., with a mean error of $50m$ and less), where the accuracy can be modeled as the radius of the circular area containing the location of the user [Ardagna et al. 2011a]. An omniscient operator o can then observe the traffic in the cellular network and exploit the physical positions of the peers to breach the anonymity of u. The forwarding peers involved in each round of the communication are in fact more likely to be located around the real requester u than peers in ORIG_{sid}. The mobility of the peers may then affect the anonymity of requester u. However, as discussed in Section 5.3, the current status of mobile technologies and devices, allowing communication at a high rate, and of the mobile communication profiles, assuming short windows of communication, result in a scenario where communications complete in few seconds [Rahmati et al. 2010]. Having communication windows of few seconds make position attacks ineffective, because peers move only for few meters. As a consequence, peers in ORIG_{sid} (including u) remain near to each other and therefore near to the forwarding peers selected in each round of request-response. Hence, $(k, 1)$-anonymity is still preserved.

Colluding global WiFi eavesdropper and neighboring peers. This adversarial model results in a powerful eavesdropper that integrates the knowledge of a global WiFi eavesdropper with the one of a set of "shadowing" neighboring peers. Although the knowledge of the neighboring peers on WiFi communications is a subset of the information observed by the global WiFi eavesdropper, this knowledge can be used by the global WiFi eavesdropper to reduce the uncertainty of its guess made on originating user u. In addition, as discussed above, supporting neighboring peers receive and observe the cellular replies; this additional information, however, cannot be used to expose the anonymity of the server since peers do not have access to the message content. As a consequence, when "shadowing" neighboring peers collude with a WiFi eavesdropper, $(1, *)$-anonymity is provided, because the identity of the server is not exposed by the protocol.

Colluding operators and global WiFi eavesdropper. This is the worst case scenario in which all infrastructure parties are assumed to be malicious and colluding. In this case, we cannot provide any protection: all communications are monitored and 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 eavesdropping. Also, it requires the eavesdropper to be able to associate the identity of the peers to their messages.

Although not infeasible, such sophisticated attacks are highly unlikely to occur in practice for the large investments of resources they would require. Practically speaking, an omniscient operator would employ a WiFi antenna to observe both the cellular and WiFi channels in a given 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 difference in range between cellular and WiFi transmissions. To be successful, an adversary in the form of an omniscient operator has then to install WiFi antennas in strategic points for *all* areas of interest and utilize them solely for the purpose of eavesdropping on *all* the available channels (each non-overlapping channel requires yet another antenna). This constitutes a significant investment of resources making it a very expensive targeted attack with uncertain outcomes due to the user's mobility, the unknown user's identity, and static or moving physical objects.

Colluding operators and neighboring peers. This is similar to the case of the colluding operators with WiFi eavesdropper, but it is easier to deploy assuming that the operators can place enough peering nodes around the victim to guarantee complete coverage. This is rather difficult in densely populated areas where the number of neighboring nodes is large and there is no clear identification of the signal of individual WiFi antennas. In addition, this attack assumes the fraction of malicious neighboring peers that surround the user to follow her in each movement. Finally, it assumes neighboring peers to be able to associate the device sending a message to the identity of the user carrying it; we recall that no identifiable information is added in the message by our protocol. For these reasons, although possible in practice, this attack is very unlikely to happen in real scenarios and, in general, $(k, 1)$-anonymity is achieved as for colluding operators.

6.2. Active adversaries

Active malicious peers try to attack the system by either dropping packets, jamming the WiFi or cellular communications, or otherwise attempt to cause a service disruption by preventing the participating nodes from successfully transmitting information to the servers. We do not consider the case in which a peer is not willing to participate in the protocol as an attack, because such peer will not appear as a node in the WiFi network, and it will not be selected to forward packets by another mobile user. We cannot defend against attacks that can completely and uniformly prevent the WiFi or the cellular communication signals. We believe that such attacks are beyond the scope of this work because they target the communication layer at a much lower level and then cannot be mitigated by our framework. On the other hand, protocol level attacks like the silent drop of packets can be mitigated or even alleviated by introducing error correction and packet replication.

To address adversaries that silently drop packets, we extend our multi-path approach by adopting a source coding scheme used as Forward Error Correction (FEC) for attack resilience. We use a simple XOR-based FEC as source coding scheme, with the note that more sophisticated source coding mechanisms can be also employed to achieve different degrees of loss resilience. Given the number N of generated packets and the number k, with $N \geq k$, of successfully transmitted packets that are required to reconstruct the original message, the requester splits the message into k chunks (packets) and creates N data packets, by using XOR operations, to be sent over the random paths. Assuming at least k different paths always exist between a message originator and the network operator (guaranteed by the communication setup in Section 5.2), each encoded packet is sent along a path. Upon arrival of the packets, the receiver needs at least k out of the N packets to recover the message. For example, let us assume that N=3 packets are needed to successfully transmit k=2 packets, where k is the user's preference. The original message needs to be split into two packets, m_1 and m_2 by the given redundancy. The third packet can be encoded as $m_1 \oplus m_2$ and then each packet is sent along a different path. If any one packet is lost, a receiver still can construct the original message by recovering the lost packet (i.e., XOR'ing the received two packets).

To analyze the effectiveness of our multi-path approach, we first develop our probabilistic threat model in multi-path communications. Let P_s denote the probability of a packet successfully forwarded to the destination over a path (i.e., path success) and $(1 - P_s)$ be the probability of path failure. Let $P(Success)$ be the success probability of an entire communication between the source and the destination, that is, the original message can be successfully reconstructed by the destination. $P_s(k)$ is the probability that k out of the N generated packets successfully reach the destination:

$$P_s(k) = \binom{N}{k} P_s^k (1 - P_s)^{N-k}$$

Thus, for multi-path configuration with N generated packets and at least one of the packets reaches the server for a successful communication, the overall probability is:

$$P(Success) = P_s(1) + P_s(2) + P_s(3) + \ldots + P_s(N)$$

where, $P_s(1)$ denotes the probability of only one of the packets being successfully forwarded to the destination and $P_s(N)$ is the probability of all of N generated packets accordingly (i.e., all paths lead to valid transmissions). The above equation can be generalized in terms of k and N by the binomial distribution. We can derive the overall probability of success with at least k legitimate packets reaching the server among N generated packets as follows:

$$P(Success) = \sum_{i=k}^{N} P_s(i) = \sum_{i=k}^{N} \binom{N}{i} P_s^i (1 - P_s)^{N-i}$$

In the above formula, we can substitute P_s with $(1 - P_d)^L$, as the path can be successful only when there is no malicious node that drops a packet on the path. Therefore, the probability of success for our multi-path communications in terms of P_d and L is:

$$P(Success) = \sum_{i=k}^{N} \binom{N}{i} ((1 - P_d)^L)^i (1 - (1 - P_d)^L)^{N-i}$$

We studied the impact of k, P_f, and P_d values to observe the probability of success by using the above equation. A complete analysis of our results is presented in Section 8. Note that attackers that appear to be non-deterministic in nature by selectively forwarding a subset of their packets offer a communication value that is proportional to the percentage of the packets they forward. Therefore, they pose less of threat than the silently but always dropping neighboring peers. In our system, we do not attempt to identify malicious peers. That would have been very tedious and potentially infeasible for short-lived anonymous communications.

6.3. 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* [Levine et al. 2004] and *predecessor attacks* [Wright et al. 2004].

Timing attacks [Levine et al. 2004] 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 their propagation on the victim's path. This usually requires at least one malicious node in the victim's path. In our scheme, timing attacks may happen either in the WiFi network or in the cellular network, where the malicious node is a peer or the mobile network operator, respectively. Focusing on WiFi network and communications, there is no recurrent path due to the mobility of the users and therefore, timing attacks are not effective against our protocol. 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 in the WiFi network. Moreover, the latency of each hop is intrinsically noisy: wireless communication performance can change due to weather conditions, interference by other devices, and physical obstacles. Focusing on cellular network and communications, an adversary (i.e., the mobile network operator) observes the timing t of the k packets $m_{p,s,t}$ forwarded by peers p to server s using our protocol, to the aim of identifying user u. However, this is not possible in our protocol for two main reasons as follows. First, u is involved only once in each communication towards s. Based on our extended protocol in Section 5.2, u never sends a packet directly to s, while it sends a single packet to s_s during the communication setup. Second, also in the worst case scenario where there is a single communication, and the adversary is able to identify s_s and observe the sender of the first packet, our solution to packet distribution synchronization in Section 5.1 does not allow the adversary to deterministically bind the sender of the first packet to user u.

The predecessor attack [Wright et al. 2004] builds on the idea that by monitoring the communication for a given number of rounds, 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). In our scenario, the predecessor attack can be exploited both in the WiFi and in the cellular networks. In the WiFi network, this attack is based on the assumption that peers in the neighborhood of requester u will observe many packets from u. However, our solution is not vulnerable to the predecessor attack since, by design of our protocol, the surrounding peers are not able to expose the identity of u. The broadcasted packets in fact do not contain identifiable information [Choia et al. 2007]. 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 in the cellular network, we need to consider the operator view of the cellular traffic. Contrary to Crowds [Reiter and Rubin 1998] where "path reformation" (definition of $\textsc{Orig}_{sid}$ in our settings) happen each time a peer joins or leaves the set of available peers, in our protocol the definition of $\textsc{Orig}_{sid}$ happens at the beginning of each communication only. Although operator o is able to identify peers in $\textsc{Orig}_{sid}$, no inference can be drawn on the identity of u.

6.4. Discussion

In this section, we presented an analysis on the anonymity provided by our protocol against different adversaries. In the following, we briefly discuss the anonymity and robustness of our protocol.

Anonymity. We evaluated adversaries aiming to breach the anonymity of our protocol and achieve $(1, 1)$-anonymity, where the identity of both u and s are exposed. More in detail, we can distinguish between three degrees of protection. The first, $(k, *)$-anonymity, is provided against neighboring peers, which do not have information on both endpoints of the communication. The second, $(1, *)$-anonymity, is preserved against a global WiFi eavesdropper, which possibly colludes with neighboring peers, assuming the worst case scenario of a single communication. The fact that k equals to one means that the adversary is able to retrieve the identity of the originating user. This attack is very challenging since it assumes a single WiFi entity that owns an infrastructure allowing the complete monitoring of the WiFi communications. The third, including $(k, 1)$-anonymity and $(*, 1)$-anonymity, is achieved against colluding operators which possibly colludes with neighboring peers ($(k, 1)$-anonymity) and single operators ($(*, 1)$-anonymity). By protocol definition, the identity of the server is exposed to the (omniscient) operator (i.e., h equals to one) because we want to reduce the impact our solution would have on the existing cellular infrastructure. The anonymity of the communication is exposed, $(1, 1)$-anonymity, when the global WiFi eavesdropper and the omniscient mobile operator collude. In this case, all communication patterns are observed and no protection is guaranteed in the worst case of a single communication.

Moreover, we evaluated timing and predecessor attacks against anonymity. Timing attacks are difficult to implement due to the fact that we are considering mobile communications on noisy channels with probabilistic paths. In addition, differently from existing solutions for wired networks (e.g., [Dingledine et al. 2004]), our protocol relies on multiple paths that are changed at run-time and implements an algorithm for packet distribution synchronization, making inferences on timing of the packets hard to achieve in practice. Predecessor attacks, where an attacker identifies a source as the one that sends a significantly higher number of packets, are not possible due to the fact that no identifiable information is in the message. The only case in which this class of attacks is successful in when a global WiFi eavesdropper observing all communications is considered. However, as already discussed, the identity of the server is not exposed in this case.

Robustness. We analyzed the impact of active adversaries (i.e., WiFi malicious dropping adversaries) on our protocol. Following the effort done in wireless sensor networks (e.g., [Li et al. 2009; Rios and Lopez 2011]), an active adversary would try to maximize its success by discovering the position of the message destination in the WiFi network and by distributing its nodes around it. This attack is not successful against our protocol since packets are first *probabilistically* distributed by the source in the WiFi network and then delivered to the destination in the cellular network. We therefore assumed active peers to be uniformly distributed in the WiFi network and, in the worst case, to surround the originating user. The main goal of active adversaries is then to drop as much packets as possible to either *i)* expose the anonymity of the user or *ii)* break the communication protocol. To counteract the attack in case *i)*, we implemented a sub-protocol to evaluate the number of honest peers in the proximity of the user (see Section 5.2); for case *ii)*, we implemented a multi-path communication with error correction and packet replication (see Section 6.2). By mixing these techniques, our protocol is able to provide a robust communication, still preserving the anonymity of the users.

7. INCENTIVES FOR COMMUNICATIONS: MICROPAYMENT

A critical aspect affecting the sustainability of anonymity-based solutions, including the one in this paper, is the lack of incentives for users to participate in the anonymization protocols. Peers' participation in fact is an essential pre-requisite to guarantee the privacy of the users. For instance, in our mobile environment, peers have costs in terms of communication overhead, computational overhead, and battery consumption, which are far more greater than the advantages gained by forwarding packets for other users.

To foster participation in our protocol, we propose to integrate a micropayment scheme within it to provide the necessary incentives for peers to collaborate. Differently from existing works that usually discuss micropayment in wired networks [Androulaki et al. 2008; Micali and Rivest 2002], we consider mobile micropayments at two levels: *i) WiFi network level,* traffic between peers in the WiFi network; and *ii) cellular network level,* traffic between peers and servers in the cellular network.

The proposed micropayment scheme rewards peers for their forwards in the network still maintaining the anonymity of the communication. The scheme relies on anonymous coins, that is, coins that are paid independently by the identity of the peers, and considers a mobile scenario with no fixed paths, mobility of the peers, and no direct communication between requester u and the peers in the path. The bank, which is responsible for checking the micropayments and rewarding the peers, is assumed to be trusted meaning that it does not collude with others to uncover the user. The bank has a public/secret key pair $\langle P_{\mathcal{B}},S_{\mathcal{B}}\rangle$ and shares a secret key $SK_{\mathcal{B},p}$ with each peer p. $P_{\mathcal{B}}$ and $SK_{\mathcal{B},p}$ are used by the peers when requesting a payment.

Before any anonymous communication can be established, the user has to buy a set of coins at the bank. Each packet in Section 4, to be sent through the anonymizing network, is extended with *i)* a coin that is cryptographically bound to it and *ii)* the secret key shared with the bank encrypted using the public key of the bank. Note that, each coin can be added to a single packet only and contains the payment for all peers in the path towards the server. Each peer receiving a packet signs the packet and the coin, and adds the secret key shared with the bank encrypted with the public key of the bank. Finally, upon receiving the packet, the server forwards it to the bank that checks the correctness of the payment and notifies the result to the server. Each collaborating peer has to show its signature of the packet and its secret key to be paid by the bank. Note that, existing solutions for payment aggregation (e.g., [Jakobsson et al. 2003; Micali and Rivest 2002]) can be used to reduce the communication overhead between the peers and the bank, also reducing the probability for an adversary to trace back the originator of a message.

Focusing on the security of the payment system, first, our scheme should be resistant to malicious peers that drop packets. Note that, the user already considers the extra cost caused by malicious dropping in their communications (see Section 6.2). In the case of malicious dropping, the peers can still be paid by communicating offline the packet to the bank. Then, the adoption of anonymous coins makes the identification of fake coins or double spending by peers difficult. To limit the distribution of malicious packets, the peers use a probability $(1 - P_b)$ as the probability of checking the packet and its payment with the bank before forwarding them to the next hop.

In the following of this section, we present the working of our payment scheme based on anonymous coins and designed for mobile environments. We will denote $\mathcal{B}$ as the bank, $\langle P_{\mathcal{B}},S_{\mathcal{B}}\rangle$ as the public and private keys of the bank, $\textsc{Sig}_K(\cdot)$ as a signature with key K, $SK_{\mathcal{B},p}$ as the secret key shared between bank $\mathcal{B}$ and a peer p, and C as the coin. The payment protocol is composed of three phases as follows.

7.1. Bootstrap

Each peer p subscribes to $\mathcal{B}$ to join the anonymizing network. Each p has to show a valid credit card linked to a valid identity. After verification of the credentials, $\mathcal{B}$ releases a secret key $SK_{\mathcal{B},p}$ that is used by the peer to deposit coins and by the bank to verify coins. To counteract peers that misbehave by presenting multiple fake identities, at most one key $SK_{\mathcal{B},p}$ is released for each identity. In this phase, p also buys a set of coins each one generated as the signature of the bank over a big pseudorandom number.

7.2. Payment

Each peer p is paid with a fixed amount of money. This means that peers are paid "per number of forwards" and different forwards are paid with the same amount known to the bank. Differently from most of the existing solutions, requester u does not need to insert L coins (where L is the length of the path) in each packet. This is extremely important in our mobile scenario since u does not know the path in advance, and has no knowledge about the number of hops required for a packet forwarding to s.

User. User u first selects a *coin* C that is then cryptographically bound to a single packet m_j. To this aim, m_j, C, and timestamp t are signed with secret key $SK_{\mathcal{B},u}$ (i.e., $\textsc{Sig}_{SK_{\mathcal{B},u}}(\{m_j, C, t\})$). This avoid peers reusing the received coin for their communications. Before sending the packet, u adds the signature and secret key $SK_{\mathcal{B},u}$ encrypted with the public key of the bank (i.e., $P_{\mathcal{B}}(SK_{\mathcal{B},u})$) to the packet. Then, u sends $[m_j, C, t, \textit{sig}=\textsc{Sig}_{SK_{\mathcal{B},u}}(\{m_j, C, t\}), \textit{sk}=P_{\mathcal{B}}(SK_{\mathcal{B},u})]$ to a peer p following the protocol in Section 4.

Peer. Upon receiving a packet $m'=[m, C, t, \textit{sig}, \textit{sk}]$, peer p checks with probability $(1-P_b)$ the packet and related payment correctness at the bank. Note that to limit intersection attacks, before sending the packet to $\mathcal{B}$, p encrypts m' with $SK_{\mathcal{B},p}$ that becomes the new *body* of the message, and generates a new pseudo random number $prn=(R^1, R^2)$ that is used to produce field *to* and the UMAC of the message. If the packet and its payment are correct, p signs m, C, and t (i.e., $\textit{sig}_p=\textsc{Sig}_{SK_{\mathcal{B},p}}(\{\text{m,C,t}\})$) and encrypts *sig*, *sk*, and secret key $SK_{\mathcal{B},p}$ (i.e., $sk_p=P_{\mathcal{B}}(\textit{sig},\textit{sk},SK_{\mathcal{B},p})$). Packet $[m, C, t, \textit{sig}_p, sk_p]$ is sent to the operator with probability $(1-P_f)$, or to a peer in the communication range otherwise.

Operator. The operator behaves as in the original protocol in Section 4.

Server. Upon receiving the packet, the server first sends it to the bank that checks for payment correctness and integrity. The server starts the response protocol in Section 4, if and only if the packets with correct payments are enough to reconstruct the original message. Otherwise, the message is dropped.

Bank. Upon receiving a packet of the form $[m, C, t, \textit{sig}, \textit{sk}]$ from the server, the bank first decrypts all *sk* with its private key $S_{\mathcal{B}}$, and retrieves signatures $\textit{sig}_p$ and shared secret keys $SK_{\mathcal{B},p}$ of all peers (including u) involved in the process. Then, it verifies the signature of each peer using the relevant secret key and notifies the server of the result. Note that, in case the bank receives a message from a peer p, it retrieves the real packet to be verified by first decrypting the message with $SK_{\mathcal{B},p}$.

7.3. Deposit

The deposit process involves either a *push process* or a *pull process*.

In the *push process*, the bank after verifying a packet sent by a server may decide to pay the peers involved in the forwarding process without an explicit request. The push process however would expose peers in the communication path. The bank can then use aggregation schemes (e.g., [Jakobsson et al. 2003; Micali and Rivest 2002]) and evaluate itself when a peer should be paid.

In the *pull process*, it is the peer that explicitly requires a payment to the bank. In particular, the peer sends its signature $\textsc{Sig}_{SK_{\mathcal{B},p}}(\{m,C,t\})$ over message m, coin C, and timestamp t, and shared key $SK_{\mathcal{B},p}$ to $\mathcal{B}$. Also in this case an aggregation scheme can be used by the peer, thus reducing the communication overhead and the probability of exposing u's identity due to greedy peers that would immediately ask for payments.

Moreover, an *hybrid process* can be implemented in the following two cases. The first case considers a peer p that forwards a packet to the bank with probability $(1 - P_b)$ for verification of correctness. If the packet verification is successful, the bank first pays p (*pull process*) and then, based on the aggregation scheme, all or part of the peers in the path (*push process*). The second case considers the scenario in which the bank does not pay or refuse a payment for a packet, due to malicious users that drop or modify the packet. In this scenario, a collaborating peer can send the whole packet to the bank to be paid (*pull process*) and, similarly to the previous case, the bank can also pay all or part of the peers in the path whose activity is correctly verified (*push process*).

Finally, there is a subtlety to consider when the last peer in the path or a peer checking the validity of a payment are involved. The traffic made by these peers towards the cellular network is subject to a fee. To avoid selfish peers forwarding all the packets they receive in the WiFi network to increase their income, every forwarding activity, both in the WiFi and cellular networks, are paid with the same net amount of money. The payments for communications in the cellular network in fact refund the peer of the additional cost of the cellular communication.

7.4. Discussion

In the following of this section we evaluate the correctness, robustness, and anonymity of the proposed solution, and we discuss possible security attacks that may target the payment scheme.

Correctness. When all peers act honestly and strictly follow the protocol, the payment scheme behaves correctly. In this context, all packets are forwarded to the server, all peers get rewarded for their activities, and the anonymity of the message originator is preserved. Peers are paid per number of forwards, meaning that they have the same income for each forward in the cellular and wireless networks. This solution counteracts greedy peers that are likely to change the protocol to maximize their income.

Robustness. Robustness refers to the probability of a successful communication in case of a probability P_d of malicious peers. Although we already discussed a solution to maximize the number of successful communications based on packet replication and a source coding scheme (see Sections 5.1 and 6.2), malicious peers have also an impact on the costs perceived by u. The higher is P_d, the higher is the number N of packets to be forwarded by u and the money to be paid for the communication. Given N and expected path length L based on P_d and P_f, u will need a coin C for each of the N packets and will pay $N \cdot L$ forwards (the payment $N \cdot L$ does not consider the fixed cost of the setup phase). The extra cost in terms of extra packets can be calculated as $(N - k) \cdot L$, where k is the original user's preference. Also, the user has an extra cost in terms of number of times the peers check with the bank the correctness of the payment. This cost can be calculated as $N \cdot L \cdot (1 - P_b)$, thus resulting in a total extra cost of $(N - k) \cdot L + N \cdot L \cdot (1 - P_b)$.

Security Attacks. The mobility of the users and the randomness of the path generation make the problem of malicious peers critical. The peers in fact may attack the scheme by having multiple fake identities, by double spending, by creating a clique of malicious peers, or by dropping packets.

(1) Our payment scheme binds each identity to a credit card, and then each credit card to a secret key between the peer and the bank. This counters sybil attacks and in general peers pretending to have multiple fake identities, unless a user is able to create a fake WiFi peer, a fake identity document, and a fake credit card. The bank receiving requests for payment has to verify the validity of the identity by checking the secret key shared with the peer.
(2) Double spending can happen at two stages. The first case involves the user that double spends its own coins (e.g., uses the same coin received by the bank for two packets). Since the coin is cryptographically bound to each packet and signed by the user with its secret key, the bank can identify the double spending (at least at deposit time) and force the user to pay for the additional traffic. The second and more critical case is when a peer receiving a packet extracts and re-uses the coin for its own communications. In this scenario, the peers in the path cannot identify the fake coins unless they directly communicate with the bank. To mitigate this attack, similarly to the discussion in Section 5.1 about the expected path length, we introduce probability $(1 - P_b)$ of payment verification with the bank. Clearly, if peers in the path towards s employ small P_b, better robustness against double spending is provided. By contrast, less anonymity is preserved, since many peers in the path will check the payment with the bank thus exposing part of the path and possibly the sender identity.
(3) Malicious peers may collude to form a *clique*, that is, a group of peers that maliciously participate in the protocol to maximize their income. Peers in a clique do not strictly follow the anonymization protocol, but rather they manipulate probability P_f of forwarding and the next hop selection. Upon receiving a packet, a peer in a clique forwards it to another peer in the clique with high P_f. As a consequence, peers in a clique forward more packets than peers following the anonymization protocol, and therefore increase their income. However, since the entire path is known to the bank when it receives a packet, the bank can analyze the path and identify colluding peers.
(4) A malicious peer may drop a packet and pretend to be paid. To this aim, it first signs the packet and then drops it. Afterwards, it sends the whole packet to the bank to get paid. Since the signature is correct the bank will pay the peer though it never forwarded the packet. Peers that repetitively sign and drop packets can be easily uncovered by the bank that observes the same peers at the end of many broken paths.

Anonymity. The payment scheme introduces the need for peers to communicate with the bank, over the cellular network, to get paid and to limit security attacks to the payment scheme. The mobile network operators can then observe an increasing amount of traffic, that can be exploited to re-identify the originator of a given communication. The proposed solution tries to minimize the potential given to the mobile operator by using: *i) payment aggregation* protocols and *ii) a probabilistic payment verification* driven by probability P_b. *Payment aggregation* avoids cases in which the mobile peers require the payments immediately after packet forwarding, thus uncovering the whole path. Payments are required by the peers (pull process) or performed by the bank (push process) only when the selection rate for payments is satisfied [Micali and Rivest 2002]. This approach results in scenarios where the payments can be requested/received after the communications have been completed. *Probabilistic payment verification* requires a careful selection of P_b such that, given an expected path length (see Figure 5), only

Table I. Latency vs Signal-to-Noise (SNR) Ratio

SNR	Minimum(ms)	Maximum(ms)	Loss(%)
14	-	-	100
16	3	52	0
24	1	28	0
32	1	20	0
48	1	10	0
64	1	8	0

few checks are performed during the forwarding path giving no information about the message originator.

8. PERFORMANCE EVALUATION

There are two primary concerns in terms of system performance when introducing a new anonymizing mechanism: the impact on the end-to-end latency and its robustness in the presence of adversaries. In this section, we first provide extensive measurements of the end-to-end latency using our own mobile devices in addition to experiments conducted in the Emulab [Emulab] and Orbit [Orbit] testbeds. The measurements presented in this paper are not based on simulations, but rather real-world experiments using well-known mobility models and the hybrid network configuration in our protocol. In all of our experiments, we used devices equipped with standard IEEE 802.11 [Networks] wireless network communication cards. To account for mobility, we measured the Signal-to-Noise ratio for neighboring nodes over a period of time. 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 equipment. Then, we use our probabilistic model to measure the robustness and attack resilience of our protocol.

8.1. Latency Overhead

The end-to-end latency overhead is an important characteristic of an anonymity system because it can adversely impact the usability of the proposed scheme. We implemented a prototype of our approach using WiFi-enabled devices and measured the latency overhead when we forward packets to neighbors. We setup an ad-hoc wireless network of multiple nodes and we measured the effect of the quality of wireless connectivity between two peers to the link latency for regular TCP packets. Unlike wired communications, wireless communications are affected by physical obstacles, mobility, and interference from other wireless devices. For the mobility, we used the Random Waypoint [Saha and Johnson 2004], that is the most popular entity-based mobility model in literature, and also the Orbit Mobility Framework [Hong et al. 2001], using city models for pedestrians. To model the link communication quality, we varied the Signal-to-Noise Ratio (SNR) of the wireless link and we measured its impact on the link latency. We employed NetStumbler [NetStumbler.com] to estimate SNR, and Wireshark [Wireshark] to calculate the link latency. In Table I, we present our finding for different SNR values for single hop, peer-to-peer wireless connections. Our results indicate that there is no significant latency overhead when SNR is within acceptable bounds.

The single-hop experiments are not enough to characterize the behavior of a multi-hop wireless ad-hoc network where interferences from transmissions by other wireless devices can degrade the signal quality. Therefore, we employed node mobility scenarios consisting of tens of nodes ($5 - 30$), where we varied SNR between 24 and 64 and we quantified the impact of our protocol on the end-to-end latency. For the mobility scenarios, we used a timed event script that was varying SNR of the link based on

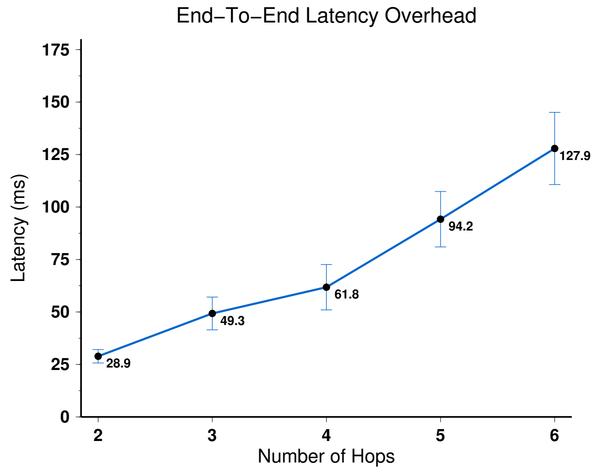


Fig. 7. End-to-End latency overhead for a multi-hop ad-hoc WiFi network running our anonymity protocol. The vertical bars represent the confidence interval for the average.

the position of the nodes. The overhead shown in Figure 7 includes the communication cost of the basic protocol in Section 4. The overhead trend is approximately linear with the number of hops. The vertical bars depict the confidence interval for each measurement. The worst case scenario, in terms of overhead, was for a 6-hop network. The increase 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).

8.2. Attack Resilience

We analyze and evaluate attack resilience in our multi-path approach by applying our probabilistic model (see Section 6.2). Figure 8 shows the probability of path success P_s with a fraction of malicious nodes dropping packets for different expected probability of forwarding, P_f=0.2, P_f=0.5, and P_f=0.8. As the figure shows, P_s rapidly decreases as the fraction of malicious nodes increases. The probability of path success also decreases as P_f increases (i.e., the expected path length increases).

Before evaluating the robustness of our approach, we present some results on the total number of packets that need to be sent in an adversarial environment by u for a successful message delivery to o. Based on the equation k=$N \cdot (1 - P_d)^L$ described in Section 5, we evaluate N for different values of anonymity preference k=2, k=3, k=5, and k=10, and different values of probability of forwarding, P_f=0.2, P_f=0.5, and P_f=0.8. As presented in Figure 9, the more probability P_f of forwarding, the more the generated packets N. Intuitively, higher P_f results in higher path length L, thus increasing the probability that a malicious peer is in the path towards operator o. As an example, we note that with a probability of malicious peers P_d=0.1 and a probability of forwarding P_f=0.2, the additional packets to be generated are 3 in the worst case (i.e., k=10). The number of additional packets decreases to 2 for k=5, and to 1 for k=2 and k=3. As another example, if we consider P_d=0.2 and P_f=0.2, we have 6 additional packets for k=10, 3 for k=5, 2 for k=3, and 1 for k=2. Moreover, if we consider P_d=0.1 and P_f=0.5, we have 4 additional packets for k=10, 2 for k=5, and 1 for k=3 and k=2. For combination of high probabilities $P_d \geq$0.3 and $P_f \geq$0.5, the number of generated

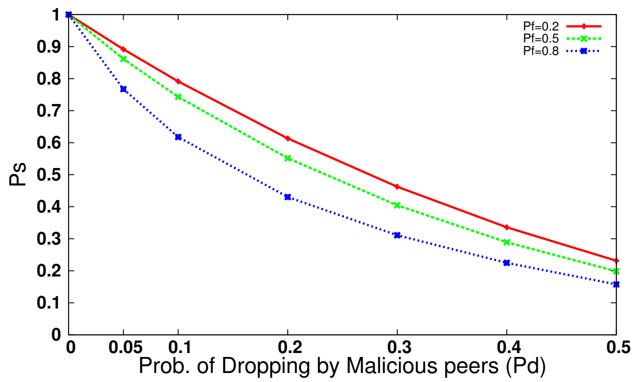
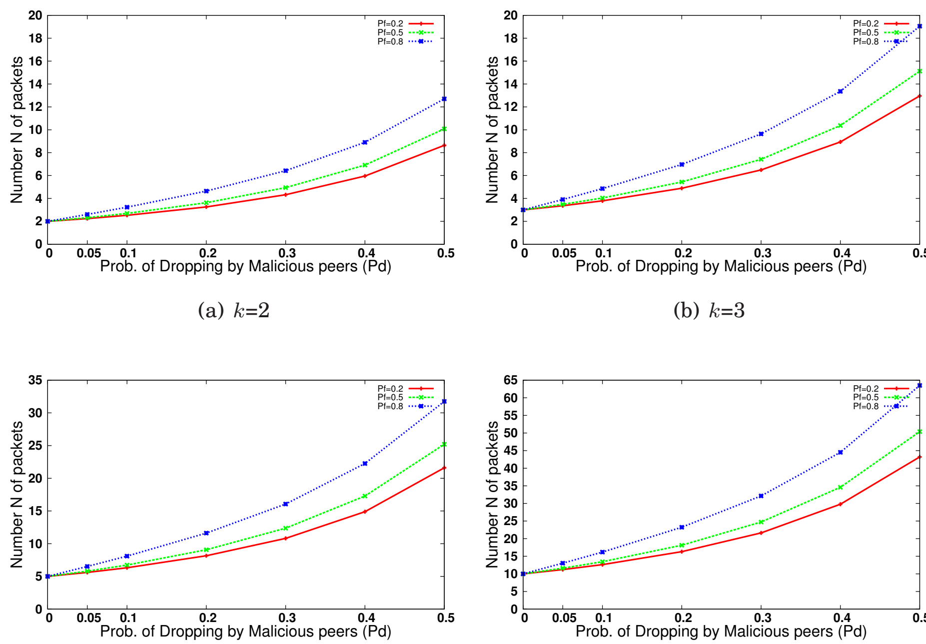


Fig. 8. The probability of path success varying P_d and P_f.



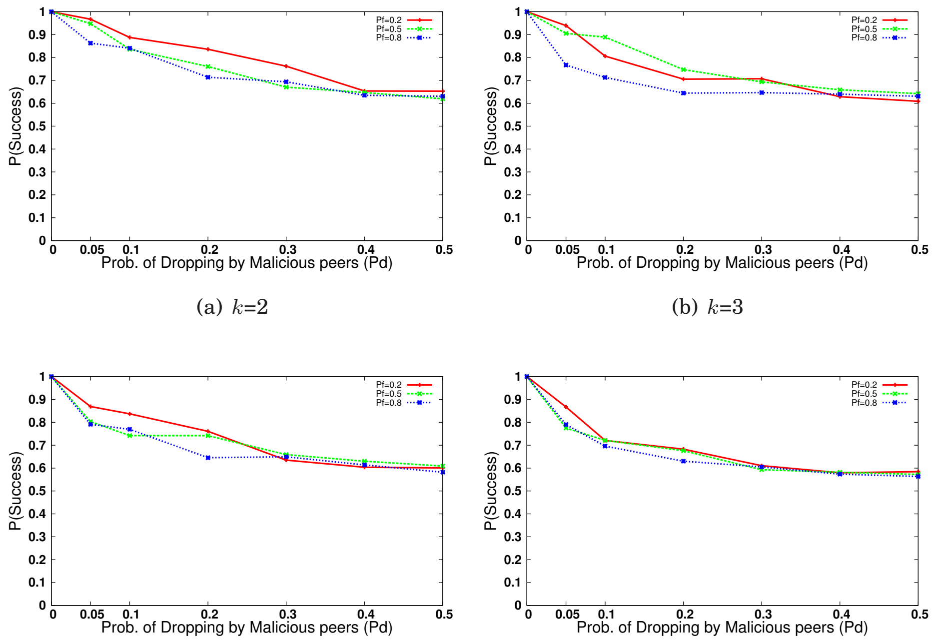
(a) k=2 (b) k=3

(c) k=5 (d) k=10

Fig. 9. Number N of generated packets varying anonymity preference k.

packets rapidly growth. The reason is that high P_f substantially increases the path length thus increasing the probability of dropped packets for high P_d. In general, for a realistic fraction of malicious users (P_d<0.3), the cost in terms of additional packets is manageable by our solution.

To quantify the network robustness of our multi-path approach, we compare the single-path (i.e., single-message sent using a single packet) versus the multi-path (i.e.,



(a) k=2 (b) k=3

(c) k=5 (d) k=10

Fig. 10. Probability of success varying anonymity preference k.

single message split in multiple packets using source coding redundancy) transmissions. Note that, for the single-path case, the k-anonymity guarantees cannot be satisfied. In our approach all packets are transmitted over distinct paths (possibly partially overlapped), and the number of packets and generated paths are equal in number.

Figure 8 shows the probability of success when sending a single message over one path, since for k=N=1, $P(Success)$=P_s. Approximately, when more than 25% of peers are malicious, $P(Success)$ becomes less than 50%, which means that the communication fails more often than succeeds. Figure 10 shows the probability of success when sending a message over multiple paths. For the analysis of our multi-path approach, we vary anonymity preference k=2, k=3, k=5, and k=10, and probability of forwarding P_f=0.2, P_f=0.5, and P_f=0.8. As depicted in Figure 10, the probability of success has a relevant boost using multiple paths independently by the chosen k and P_f. $P(Success)$ is in fact always greater than 50% for $P_d \leq 0.5$. Also, we note that the trend of $P(Success)$ is decreasing as P_d increases and, unless some noise that is introduced by the need of rounding up the value of N, as P_f increases. This is again due to the fact that a higher P_f results in a higher path length. Finally, the higher anonymity preference k of the requester, the lower the probability of success. This result shows that malicious peers are more effective in case the user wants to preserve a higher k-anonymity. For high k, in fact, more packets need to be delivered to the operator, thus increasing the number N of generated packets and the probability that a malicious peer catches a packet. Clearly this introduces a conflicting scenario where better anonymity protection reduces the probability of success. There is therefore the need to define anonymity preference k balancing the preserved k-anonymity and the probability of success.

In summary, based on our results, we can verify that multi-path offers attack resilience even when a significant fraction of the paths are compromised. In addition, the multi-path approach achieves better results than the single-path approach in terms of probability of success, although single-path approach does not consider anonymity protection. To conclude, our multi-path approach paired with the source-coding scheme can improve attack resilience significantly by adding redundancy to the local WiFi network.

9. RELATED WORK

Past research addressing communication privacy in mobile networks [Capkun et al. 2004; Lin et al. 2007; Ren and Lou 2008] has been inspired by works focusing on wired networks. Traditional solutions like TOR [Dingledine et al. 2004] for route anonymity and Crowds [Reiter and Rubin 1998] 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 [Network], MorphMix [Rennhard and Plattner 2002] take a different approach and provide P2P-based solutions for network anonymity. I2P [Network] is an anonymizing network for secure communications 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 [Rennhard and Plattner 2002] 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, form networks of arbitrary topology, and use devices with limited capabilities.

Existing research in the context of mobile networks mainly focused on protecting communication anonymity and privacy in mobile ad-hoc networks [Aiache et al. 2008; Chen and Wu 2010; Dong et al. 2009; Kong and Hong 2003; Takahashi et al. 2010; Zhang et al. 2006], vehicular ad-hoc networks [Lin et al. 2007; Sampigethaya et al. 2007], and mobile hybrid networks [Ardagna et al. 2010; Capkun et al. 2004]. ANODR [Kong and Hong 2003] provides an untraceable and intrusion tolerant routing protocol, based on the paradigm of "broadcast with trapdoor information". It provides communication anonymity, by preventing adversaries from following packets in the network, and location privacy, by preventing adversaries to discover the real position of local transmitters (which could disclose also their identity). MASK [Zhang et al. 2006] proposes an anonymous routing protocol, which provides both MAC-layer and network-layer communications without the need of using the real identities of the participating nodes. MASK provides communication anonymity, in addition to node location anonymity and untraceability, and end-to-end flow untraceability. MASK relies on the use of dynamic pseudonyms, rather than static MAC and network addresses, and on pairing-based cryptography to establish an anonymous neighborhood authentication between nodes and an anonymous network-layer communication. Dong et al. [Dong et al. 2009] propose an anonymous protocol for mobile ad-hoc networks that does not rely on topological information to protect identity and 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. Chen and Wu [Chen and Wu 2010] provide an anonymous routing protocol for mobile ad-hoc networks. Similarly to our solution, the authors use a multipath approach where different packets of the same message are sent into different paths and only a subset of the packets is needed to reconstruct the message at the receiver side. Aiache et al. [Aiache et al. 2008] propose a solution aimed to provide security and anonymity in mobile ad-hoc networks. The proposed approach is based on a multipath routing protocol, message splitting, and asymmetric cryptography. Also, dummy traffic is added to the communication to guarantee anonymity. GSIS [Lin et al. 2007] presents a protocol, based on Group Signature and Identity-based Signature techniques, used to protect security and privacy in vehicular networks. Sampigethaya et al. [Sampigethaya et al. 2007] present AMOEBA, a robust location privacy scheme for VANET. AMOEBA focuses on protecting users' privacy against malicious parties aiming at tracking vehicles and building a profile of Location-Based Services (LBSs) they access. To these aims, AMOEBA relies on vehicular groups and random silent periods. Finally, Capkun et al. [Capkun et al. 2004] provide a scheme for secure and privacy-preserving communications in hybrid ad-hoc networks based on pseudonyms and cryptographic keys. Differently from the above approaches, our solution does not rely on heavy 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 thus protecting the servers against DoS.

Other work on privacy protection has addressed the problem of preserving the anonymity and the location privacy of requesters that interact with LBSs (e.g., [Ardagna et al. 2011a; Chow et al. 2011; Gedik and Liu 2008; Gruteser and Grunwald 2003]). LBSs are considered untrusted parties that can exploit location information of users to breach their privacy. The main goal of most of the current solutions is to guarantee anonymity, by preventing adversaries to use location information for re-identifying the users. In this scenario, each location measurement is manipulated to keep users' identity hidden, still preserving the best accuracy possible. Differently by the solution in this paper, these approaches only provide anonymity and location privacy at application level, while they do not consider communication privacy. Furthermore, they assume trusted mobile network operators.

Ren and Lou [Ren and Lou 2008] and Magkos et al. [Magkos et al. 2010] present two approaches similar to the one in this paper. The work by Ren and Lou [Ren and Lou 2008] is aimed at providing a privacy yet accountable security framework. The proposed solution, however, is based on multiparty computation and groups of users established a priori, and assumes a semi-trusted group manager and network operator. Magkos et al. [Magkos et al. 2010] consider the problem of providing privacy-preserving location-based queries in mobile hybrid networks. Their solution assumes no trusted third parties (including the mobile network operator) and is aimed at traffic untraceability. The proposed solution is based on multiparty computation, and does not consider malicious users and the need of incentives for collaborating peers.

This paper considerably extends the works in [Ardagna et al. 2008; Ardagna et al. 2009; 2010] by providing an enhanced protocol that includes a communication setup phase to anonymously evaluate if there are enough peers willing to collaborate in the neighborhood of the requester. Also, it proposes an adversarial analysis that considers colluding operators with external knowledge and active adversaries that try to attack the system by dropping packets. Furthermore, it presents an incentive for peers to participate in the protocol based on a micropayment scheme for mobile communications. Finally, it proposes an extended protocol evaluation that considers resilience against malicious attacks.

10. CONCLUSIONS

We proposed a protocol for protecting users' privacy that harnesses the availability of both mobile and WiFi connectivity in current phones creating a hybrid network. Differently from traditional solutions that mostly focused on protecting the privacy of the users by the prying eyes of 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. We also formally quantified the privacy protection provided by our protocol in the presence of malicious adversaries, which can collude to breach user's anonymity or disrupt the communication. Moreover we proposed micropayment-based incentives for users to collaborate in the protocol. Finally we evaluated the network overhead and attack resiliency of our solution.

REFERENCES

H. Aiache, F. Haettel, L. Lebrun, and C. Tavernier. 2008. Improving security and performance of an Ad Hoc network through a multipath routing strategy. *Journal in Computer Virology* 4, 4 (2008), 267–278.

A. Allan and P. Warden. 2011. *Got an iPhone or 3G iPad? Apple is recording your moves*. http://radar.oreilly.com/2011/04/apple-location-tracking.html accessed in February 2013.

E. Androulaki, M. Raykova, S. Srivatsan, A. Stavrou, and S.M. Bellovin. 2008. PAR: Payment for Anonymous Routing. In *Proc. of the 8th Privacy Enhancing Technologies Symposium (PET 2008)*. Leuven, Belgium.

M. Anisetti, C.A. Ardagna, V. Bellandi, E. Damiani, and S. Reale. 2011. Map-Based Location and Tracking in Multipath Outdoor Mobile Networks. *IEEE Transactions on Wireless Communications* 10, 3 (March 2011), 814–824.

C.A. Ardagna, M. Cremonini, S. De Capitani di Vimercati, and P. Samarati. 2011a. An Obfuscation-based Approach for Protecting Location Privacy. *IEEE Transactions on Dependable and Secure Computing* 8, 1 (January-February 2011), 13–27.

C.A. Ardagna, S. De Capitani di Vimercati, and P. Samarati. 2011b. Personal Privacy in Mobile Networks. In *Mobile Technologies for Conflict Management: Online Dispute Resolution, Governance, Participation*, M. Poblet (Ed.). Springer Science+Business Media B.V.

C.A. Ardagna, S. Jajodia, P. Samarati, and A. Stavrou. 2009. Privacy Preservation over Untrusted Mobile Networks. In *Privacy in Location-Based Applications: Research Issues and Emerging Trends*, C. Bettini, S. Jajodia, P. Samarati, and S. Wang (Eds.). Lecture Notes of Computer Science, Springer.

C.A. Ardagna, S. Jajodia, P. Samarati, and A. Stavrou. 2010. Providing mobile users' anonymity in hybrid networks. In *Proc. of the 15th European Symposium on Research in Computer Security (ESORICS 2010)*. Athens, Greece.

C.A. Ardagna, A. Stavrou, S. Jajodia, P. Samarati, and R. Martin. 2008. A Multi-Path Approach for k-Anonymity in Mobile Hybrid Networks. In *Proc. of the International Workshop on Privacy in Location-Based Applications (PILBA 2008)*. Malaga, Spain.

C. Bettini, S. Jajodia, P. Samarati, and S. Wang (Eds.). 2009. *Privacy in Location-Based Applications: Research Issues and Emerging Trends*. Vol. 5599. Lecture Notes of Computer Science, Springer.

G. Bianchi. 2000. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on selected areas in communications* 18, 3 (2000), 535–547.

J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway. 1999. UMAC: Fast and Secure Message Authentication. In *Proc. of the 19th Annual International Cryptology Conference (CRYPTO 1999)*. Santa Barbara, CA, USA.

S. Capkun, J.-P. Hubaux, and M. Jakobsson. 2004. *Secure and Privacy-Preserving Communication in Hybrid Ad Hoc Networks*. Tech. Rep. IC/2004/10, EPFL-IC, Lausanne, Switzerland.

D. Chaum. 1981. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Commun. ACM* 24, 2 (1981), 84–88.

S. Chen and M. Wu. 2010. Anonymous Multipath Routing Protocol Based on Secret Sharing in Mobile Ad Hoc Networks. In *Proc. of the International Conference on Measuring Technology and Mechatronics Automation (ICMTMA 2010)*. Changsha, China.

J. Cheng. 2011. *Pandora sends user GPS, sex, birthdate, other data to ad servers*. http://arstechnica.com/gadgets/news/2011/04/pandora-transmits-gps-gender-birthdate-other-data-to-ad-servers.ars accessed in February 2013.

H. Choia, P. McDaniel, and T.F. La Porta. 2007. Privacy Preserving Communication in MANETs. In *Proc. of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2007)*. San Diego, CA, USA.

C.-Y. Chow, M.F. Mokbel, and X. Liu. 2011. Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments. *Geoinformatica* 15 (2011), 351–380.

V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. 2007. k-Anonymity. In *Secure Data Management in Decentralized Systems*, T. Yu and S. Jajodia (Eds.). Springer-Verlag.

V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. 2009. Theory of Privacy and Anonymity. In *Algorithms and Theory of Computation Handbook (2nd edition)*, M. Atallah and M. Blanton (Eds.). CRC Press.

R. Dingledine, N. Mathewson, and P. Syverson. 2004. Tor: The Second-Generation Onion Router. In *Proc. of the* 13^{th} *USENIX Security Symposium*. San Diego, CA, USA.

Y. Dong, T.W Chim, V.O.K. Li, S.M. Yiu, and C.K. Hui. 2009. ARMR: Anonymous routing protocol with multiple routes for communications in mobile ad hoc networks. *Ad Hoc Networks* 7, 8 (2009), 1536–1550.

J.R. Douceur. 2002. The Sybil Attack. In *Proc. of the First International Workshop on Peer-to-Peer Systems (IPTPS 2002)*. Cambridge, MA, USA.

Emulab. *Network Emulation Testbed Home*. http://www.emulab.net/.

B. Gedik and L. Liu. 2008. Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms. *IEEE Transactions on Mobile Computing* 7, 1 (January 2008), 1–18.

M. Gruteser and D. Grunwald. 2003. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proc. of the 1st International Conference on Mobile Systems, Applications, and Services (MobiSys 2003)*. San Francisco, CA, USA.

F. Gustafsson and F. Gunnarsson. 2005. Mobile Positioning Using Wireless Networks: Possibilities and fundamental limitations based on available wireless network measurements. *IEEE Signal Processing Magazine* (July 2005), 41–53.

X. Hong, T.J. Kwon, M. Gerla, D.L. Gu, and G. Pei. 2001. A Mobility Framework for Ad Hoc Wireless Networks. In *Proc. of the Second International Conference On Mobile Data Management (MDM 2001)*. Hong Kong, China.

M. Jakobsson, J.-P. Hubaux, and L. Buttyán. 2003. A Micro-Payment Scheme Encouraging Collaboration in Multi-Hop Cellular Networks. In *Proc. of the 7th International Financial Cryptography Conference (FC 2003)*. Gosier, Guadeloupe, FWI.

J. Kong and X. Hong. 2003. ANODR: ANonymous On Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks. In *Proc. of the Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003)*. Annapolis, MD, USA.

T. Krovetz. 2006. UMAC: Message Authentication Code using Universal Hashing. RFC 4418 (Informational). (March 2006). http://www.ietf.org/rfc/rfc4418.txt

B.N. Levine, M.K. Reiter, C. Wang, and M. Wright. 2004. Timing Attacks in Low-Latency Mix Systems (Extended Abstract). In *Proc. of the 8th International Financial Cryptography Conference (FC 2004)*. Key West, FL, USA.

X. Li, X. Wang, N. Zheng, Z. Wan, and M. Gu. 2009. Enhanced Location Privacy Protection of Base Station in Wireless Sensor Networks. In *Proc. of the 5th International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2009)*. Wu Yi Mountain, China.

X. Lin, X. Sun, P.-H. Ho, and X. Shen. 2007. GSIS: A Secure and Privacy Preserving Protocol for Vehicular Communications. *IEEE Transactions on Vehicular Technology* 56, 6 (November 2007), 3442–3456.

E. Magkos, P. Kotzanilolaou, S. Sioutas, and K. Oikonomou. 2010. A Distributed Privacy-Preserving Scheme for Location-based Queries. In *Proc. of the Fourth IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC 2010)*. Montreal, Canada.

S. Micali and R.L. Rivest. 2002. Micropayments Revisited. In *Proc. of the The Cryptographer's Track at the RSA Conference on Topics in Cryptology (CT-RSA 2002)*. San Jose, CA, USA.

NetStumbler.com. http://www.netstumbler.com/.

I2P Anonymous Network. http://www.i2p2.de/.

IEEE 802.11^{TM} Wireless Local Area Networks. http://www.ieee802.org/11/.

Wireless Orbit. http://www.wirelessorbit.com/.

A. Rahmati, C. Shepard, A. Nicoara, L. Zhong, and P.J. Singh. 2010. Mobile TCP Usage Characteristics and the Feasibility of Network Migration without Infrastructure Support. In *Proc. of the 16th Annual International Conference on Mobile Computing and Networking (MobiCom 2010)*. Chicago, IL, USA.

M.K. Reiter and A.D. Rubin. 1998. Crowds: Anonymity for Web Transactions. *ACM Transactions on Information and System Security* 1, 1 (1998), 66–92.

K. Ren and W. Lou. 2008. A Sophisticated Privacy-Enhanced Yet Accountable Security Framework for Metropolitan Wireless Mesh Networks. In *Proc. of the 28th IEEE International Conference on Distributed Computing Systems (ICDCS 2008)*. Beijing, China.

M. Rennhard and B. Plattner. 2002. Introducing MorphMix: peer-to-peer based anonymous Internet usage with collusion detection. In *Proc. of the Workshop on Privacy in the Electronic Society (WPES 2002)*. Washington, DC, USA.

R. Rios and J. Lopez. 2011. Exploiting Context-Awareness to Enhance Source-Location Privacy in Wireless Sensor Networks. *Comput. J.* 54, 11 (2011), 1603–1615.

A.K. Saha and D.B. Johnson. 2004. Modeling mobility for vehicular ad-hoc networks. In *Proc. of the First ACM Workshop on Vehicular Ad Hoc Networks (VANET 2004)*. Philadelphia, PA, USA.

P. Samarati. 2001. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering* 13, 6 (2001), 1010–1027.

K. Sampigethaya, M. Li, L. Huang, and R. Poovendran. 2007. AMOEBA: Robust Location Privacy Scheme for VANET. *IEEE Journal on Selected Areas in Communications* 25, 8 (October 2007), 1569–1589.

M. Shin, C. Cornelius, D. Peebles, A. Kapadia, D. Kotz, and N. Triandopoulos. 2011. AnonySense: A system for anonymous opportunistic sensing. *Pervasive and Mobile Computing* 7, 1 (February 2011), 16–30.

G. Sun, J. Chen, W. Guo, and K.J. Ray Liu. 2005. Signal Processing Techniques in Network-Aided Positioning: A survey of state-of-the-art positioning designs. *IEEE Signal Processing Magazine* (July 2005), 12–23.

D. Takahashi, X. Hong, and Y. Xiao. 2010. On-Demand Anonymous Routing with Distance Vector Protecting Traffic Privacy in Wireless Multi-hop Networks. In *Proc. of the 4th International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2008)*. Wuhan, China.

Wireshark. http://www.wireshark.org/.

M. Wright, M. Adler, B. Neil Levine, and C. Shields. 2004. The predecessor attack: An analysis of a threat to anonymous communications systems. *ACM Transactions on Information and System Security* 7, 4 (2004), 489–522.

Y. Zhang, W. Liu, W. Lou, and Y. Fang. 2006. MASK: Anonymous On-Demand Routing in Mobile Ad Hoc Networks. *IEEE Transactions on Wireless Communications* 5, 9 (September 2006), 2376–2385.