MokA: Mobile k-Anonymity
Sender Anonymity Over Untrusted Mobile Network Operators

Rhandi Martin, Rahul Murmuria, Angelos Stavrou and Nelson Nazzicari
Center for Secure Information Systems
George Mason University
Fairfax, VA 22030
{rmartinl,rmurmuri,astavrou,nnazzica}@gmu.edu

***Abstract*—The increasing demand for intelligent mobile devices has bolstered the sales of smart phones for which user-centric applications and services are developed. These devices access remote services via Bluetooth, WiFi and, mostly, mobile network operators. Unfortunately, users assume the trustworthiness of network operators in their retention and preservation of private user information; however, the sensitive information and extrapolated, usage trends of clients is valuable information to businesses. We assume that a network operator is not malicious, but curious about user information, for this reason. It, therefore, becomes imperative that we preserve the privacy of users, and safeguard user information from the network operator. Moreover, we must maintain the usability of the system. In previous work, researchers leveraged multipath communication and k-anonymity to create sender anonymity in a WiFi, mobile, ad-hoc environment. We implement this concept of network k-anonymity for Android-based smart phones. Our architecture provides sender anonymity, and data integrity. We measure the overhead of our system and evaluate user perception of web services over our system. Our limited implementation centers HTTP traffic, but does not consider accountability. Preliminary results show neglible latency introduced by our system.**

***Keywords*-internet privacy, k-Anonymity, Android, multipaths.**

I. Introduction

The proliferation of Bluetooth, WiFi and cellular networks, has given rise to a number of wireless devices and techologies which leverage one or more of these networks. These technological advances have changed user attitudes and focus. They have moved traditionally desktop-oriented, everyday tasks to increasingly smaller, mobile devices. With this change comes an expectation of high connectivity- to be Internet-accessible regardless of their physical location. And, to this end, a burgeoning number of web-dependent and location-aware applications have been developed for different mobile architectures and operating systems.

For mobile phone users, the majority of connections to service providers are made through their cellular networks.s Unfortunately, by default, such technologies assume or fail to verify the trustworthiness of the players in the data path, as well as the neighbouring devices which can directly sense the broadcasted transmissions. From the security point of view, in the worst-case scenario, not even the service provider is considered a trustworthy party. In addition, both the neigbouring nodes and the network operator can invade user privacy by recording and analyzing user information. Particularly, the network operator carries a mix of Internet and voice traffic and posseses the user data, a great number of uniquely identifying characteristics, and location-sensing technology. For these reasons, concern about privacy is justified. Even if the network operators are obvious, we must consider the "Big Brother" scenario in which, investigating parties, monitor and intercept communications [15], [11], [14].

Privacy systems center around providing confidentiality from listening devices, or protecting users from untrusted service providers. They assume that mobile network operators are not malicious; however, recent research [2] may treat the network operator as honest, but curious players, so that we must protect users from the watchfulness

of the network. Ardagna, et al, extend network k-anonymity to a mobile infrastructure. They postulate its application to a WiFi-based mobile ad-hoc network (MANET) for which the devices have the ability to forward packets to another peer or network, e.g. cellular network, wired LAN, etc. In so doing, the mobile network operator associates a packet flow as originating from no fewer than k peers, thus not being able to reconstruct the exact origin of the traffic.

In choosing a platform for our k-anonymity implementation, we considered an architecture which allows the flexibility of low-level design and modification, as well as user-space applications. Android is not simply an open source project; it is built on Linux, has a rich application programming interface (API) and third party support, and provides users with almost 40,000 applications. It also represents the greatest Web traffic share among operating systems on mobile devices.[18] Furthermore, as more future buyers of cell phones declared their intent to purchase Android devices in the coming months[4], an Android-based device represents the best option available.

The remainder of the paper is organized as follows. In Section 2, we discuss related technologies and approaches in concept and components of the system. The architecture for Android, and the design and implementation of k-Anonymity for this platform, are discussed in Section 3. Section 4 addresses open problems and future directions of our application that we call Mobile k-anonymity for Android- MokA. Finally, in Section 5, we discuss the conclusions drawn from our experiment.

II. Architecture

A. Overview

Our reference model is a hybrid, mesh network of mobile devices consisting of cellular phones and laptops with wired and wireless access [12], [19]. The mobile users access both phone and Internet services through the cellular provider, such that the GSM/3G (cellular) network is the next hop in the path to the service provider.

In our scenario, the mobile devices use WiFi to establish adhoc (point-to-point) connections to other mobile peers to create a mobile ad-hoc network (MANET). Simultaneously, users remain in contact with - send and receive to - cellular, radio towers.

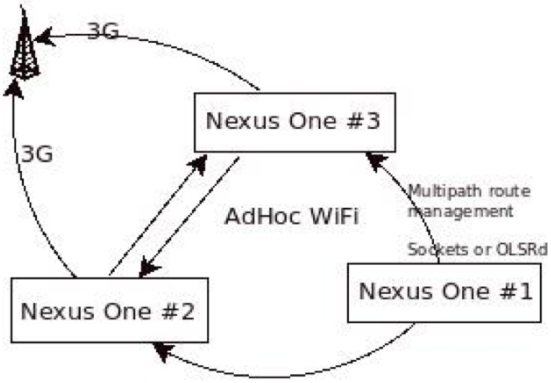

Figure 1. MANET of Nexus One (Android) Phones

In such hybrid networks, user privacy is of great concern. Most privacy solutions for mobile users place the responsibility on the service provider to provide anonymity; however, as service providers wish to hold clients responsible for their actions, they have no incentive to provide this. Additionally, they assume that the path to the service provider, for which the mobile network provider is the gateway, is trusted. We implement a work that challenges the implicit trust of network providers as they maintain the unique cellular and Internet identifiers, and continuously monitor and track their customers.

In designing a comprehensive privacy solution for wireless devices, we must consider the ability of devices in the immediate vicinity to monitor transmissions. We assume that local mobile devices are honest; however, if curious, they may eavesdrop on wirelessly broadcasted traffic, but will not maliciously modify packets. Additionally, in a multihop scenario, intermediate nodes will not drop forwarded packets. More importantly, they will not cooperate with queries about location and other user data, i.e. they provide no more information than is immediately available to the network operator.

1) Functionality: Recalling the fundamental requirements discussed in previous sections, we need to implement certain functionality and protocols on our Android platform, that are not immediately implemented on commerically availables devices. In our solution, for each cellular device, we must:

- Enable ad-hoc mode through WiFi driver configuration
- Obtain signal strength and other network metrics of neighboring nodes
- Enable IP forwarding and network address translation, and route management
- Route the packet flow from sending node via different paths (multipaths) on a per-packet basis
- Re-weight routes based on the network metrics
- Measure results for usability of this infrastructure

B. Platform

For its extensibility, we chose an Android base for which we required developer options and root access to the Android kernel. Specifically, our implementation uses the following Android base:

1) 3 Nexus One phones, unlocked to developer mode
2) Modified kernel 2.6.33 by Cyanogenmod provides root access
3) Google Android 2.1 update 1
4) 1 Apple MacBook Pro, running Mac OS X 10.5.x, in ad-hoc mode

C. Limitations

Our project was divided into two phases. The first stage was to establish the capabilities and limitations of our Android devices, and to assess the efficacy and compatibility of related technologies. We endeavoured to identify and implement each of the components necessary for our system. We segmented the project into: routing, link quality assesment, wireless adhoc connections and topology discovery. Each one of these components possessed their own challenge.

In the initial phase, we manually created our network. From previous projects that addressed wireless adhoc networking on Android [13], [7], we gained the ability to tether, and obtained a framework for sensing neighbours- by broadcast to populate the ARP table. While tethering provided the ability to forward traffic received through Wifi out another interface, we also needed to forward packets. Android is a more limited capability Linux operating system; however, by using a modified kernel that elevates our privilege, we configured the kernel to forward packets received from peers, manually added multiple routes using the 'route' command, and instituted routing policy, sending traffic from one peer to many, using 'iproute' utility. Also, in trying to implement OLSR, we discovered that there was no method for true per-packet multipath.

At this layer, we also could obtain good metric for adjusting the weight of the routes. OLSR only provided a failover implementation of moving to another route when one link was two busy, or to weight each equally, independent of link quality.

D. Implementation

For phase two, we designed a dynamic method of acheiving the above functionality, and implemented a new framework for per-packet multipath. In order to achieve these requirements, we needed to enable kernel-level functionality; however, access is not possible in Android devices without the subsitution of a modified kernel. And, while most functionality can be enabled at such low level, a functional solution requires user-space implementation. To this end, we employed, or implemented concept from, the following tools:

- Android Software Development Kit (SDK) and API for building our user application [6]
- Bionic library in the Android Native Development Kit (NDK) - ARM cross-compilation
- Wireless Tether for Android [13] - enables tethering (ad-hoc mode) from user-space
- WifiManager in SDK for connection information
- lipipq (netfilter) - kernel-level packet processing

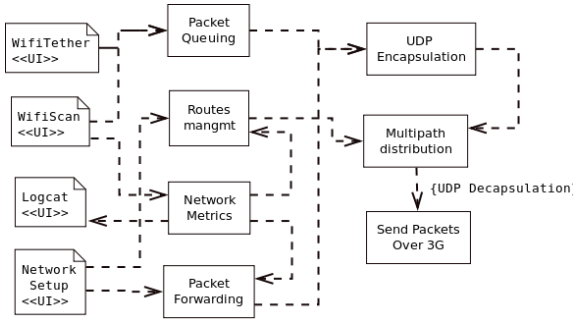

Figure 2. Components of MokA System

- Java Native Interface (JNI) - to integrate Android API (Java) with our libipq (C) code

To this end, we rebuilt a kernel with our desired functionality, and implemented it on our Nexus one devices.

With these implements we create our Nexus One tested. We create filter rules to capture and redirect web traffic to a system queue. We compile and run our low-level packet-processing application, with root permissions. From the userspace, we identify the nodes in the network and populate our routing table. Also, at this level, using Android API functions, we also acquire WiFi state information . This information includes: supplicant state, signal strength, and link speed. With these, we can determine the availability of a node, and compute a load balancing metric for packet distribution. This would allow Moka to, periodically, re-check cooperating nodes and re-weight routes.

For our limited implementation, each node in the network is known and allowed by access control mechanisms included in Wifi Tether. Only on successful application to join the ad-hoc network is a node added to the routing table. We forward the packet to the next hop in our route list. We assume that the receiving node will NAT and forward the received packet to the network operator via its 3G or LAN link.

In MokA, we test our the following functionality with:

- 3G connection on both Android devices for handing the traffic over to the network operator
- Wired connection on laptop for forwarding to LAN
- Real web traffic generated via typical browsing (Android Browser and Chrome for Mac, respectively)

III. Experimental results

Due to our inability to dump traffic from the phones, we were unable to provide metrics for user experience and latency. However, in the Phase 1 of our implementation, we confirmed by dumped traffic from the laptop, that traffic was routed through it, as well as a secondary, handheld device. We also traced connections to websites for which the nexthop varied between the routes we created.

We verified mobility, by varying distance, and thereby signal strength, between multiple adhoc nodes in the network, such that a node with decreased link quality to one nexthop, would reroute packets to a node with stronger signal. By wireless scanning, we determined at a given time to which receiving node the sending node connected, via MAC address.

We created an application which: discovers adhoc network topology; creates a routing table; recalculates and updates route weights; and, enables multipath for mobile phones.

Our ability to provide a wireless ad-hoc network will be demonstrated, for Android, visually.

IV. Previous Work

Previous work of *Ardagna, Stavrou, et al.* on multipath communication and k-anonymity examines a technique to disassociate packet data traffic and the real-world identity of the sender. Anonymity towards the Network Operator is provided by using this k-Anonymity approach where we confuse the Network Operator from pinpointing the sender, reducing the indentification information on packets to mere geographical region only. *Hopper et. al* discuss in their paper how k-anonymity serves the goal to provide anonymity by unobservability [8]. Whereas, unlinkability is the other form of anonymity where a “mix” server is used to unlink messages delivered by the system, from the messages input to the system during that

time period. *Danezis and Wittneben* observed that resistance to surveillance attacks on social networks, where identity is well-known, is ineffective using unlinkable connections mechanism, like the mix servers. That is because it is not hard to send probes to the other side of the "mix" server.

Although most of current research in anonymity discuss techniques to "unlink" traffic with the help of intermediate servers, there have been several designs for unobservable networks, like SPM [17]. The idea is to distribute data while ensuring that one's private contribution to the global computation is not revealed [5]. The origin for such algorithms can be traced back to the well-known Dining Cryptographers Problem. These research assume that no parties are trusted in the network, which adds significant complexity to the design. There is no commercially available implementation of these techniques.

Recent work by *Johnson et. al* explores performing route-selections based on arbitrary levels of trust assigned to the nodes taking part in the onion-routing network [10]. Anonymous Onion Routing (AOR) is a technique designed to provide anonymous communication infrastructure over the Internet [1][16]. An "onion" is a data structure composed by a layer of encryption for each router in the path, in a server-client communication. TOR is a second generation onion routing-based solution that provides anonymity by preventing adversaries from following packets from sender to receiver and vice versa [**?**]. These works prove that assuming some level of trust exists among your peer nodes, who take part in the k-Anonymity network, anonymity can be effectively protected against any node outside of the k-anonymity network.

V. Related Projects

For our platform, Google Android operating system, several open source projects had to be leveraged to provide a unified solution and create a base for us to perform our tests.

A. AdHoc Networks

MANETs are composed by mobile hosts that form networks of arbitrary topology, by means of wireless communications, and use ad-hoc routing protocols to communicate among them. In order to modify our devices to listen to Wi-Fi in Ad-Hoc mode, two existing projects were used.

- WiFi Tether for Android : This program enables tethering (via wifi and bluetooth) for "rooted" handsets running android. As a result other nodes can connect to handsets running this application over Wi-Fi Adhoc mode, and have the tethering handset forward traffic either to the Network Operator over 3G or to other peer nodes in the k-Anonymity network, depending on route selections performed by the routing service.
- Adhoc Client for Android : This project will address discovery of phones in communication range, ad-hoc data communications between phones, and exchange of location information to facilitate social networking applications. This may include displaying relative locations and exchanging text messages or other multimedia content.

B. Network Quality Measurements

- ETX - a packet loss measurement (estimation)
- TFRC-SP - a congestion control protocol which produces smooth sending rate
- LMDR - protocol senses the mobility of nodes and adjusts TCP rates

C. Route Management

Load Balancer Routing:

- Zebra/Quagga - a routing daemon for many protocols, e.g. BGP, RIP, OSPF, etc.
- OLSRd - built on Zebra, implements OLSR[9] for older Android models

VI. Future Work

There are many interesting areas of research into forwarding protocols, location obfuscation and gateway load balancing that may be investiagated to optimize or augment the efficient and relibility of MokA. Additionally, as we do not address a more comprehensive threat model that considers malicious peers, we envision a more comprehensive version of MokA that accounts for malice. Finally,

we would like augment our decision forwarding algorithm to factor a peer's gateway connection type – whether a connection is unlimited or limited data usage – to distinguish between uncoorporative peers and restricted ones.

In a more complete version of MokA, we would add the following capabilities.

- Auto-configuration - improved IP address management [3].
- Handoff - fast handover/handoff in MANET
- Accountability/Billing - percentage usage limit, attack origin.
- Resource Exhaustion - battery usage per flow, and per packet.

In addition, we would measure the per-flow and per-packet latency introduced by MokA, and categorize the perceived user experience.

VII. Conclusions

In this paper, we present an implementation of network k-anonymity for preserving the privacy of mobile users. We leverage multi-path in a mobile ad-hoc network, on commercial mobile devices, to demonstrate the efficability of this solution. We assert that ours is a reliable method for safeguarding users' privacy from honest but curious network operators, and suggest that future applications may be made scalable. In addition to obfuscating sender identity, MokA confuses the location-tracking ability of network operators.

VIII. Acknowledgements

We acknowledge the Center for Secure Information Systems at George Mason University for access to tools and data. We thank our peer researchers - in particular, Quan Jia and Sharath Hiremagalore - for invaluable suggestions and feedback. Also, we thank Dr. Robert Simon for his guidance on this project.

References

[1] Ontion router, Apr. 2010.

[2] C. A. Ardagna, A. Stavrou, S. Jajodia, P. Samarati, and R. Martin. A multi-path approach for k-anonymity in mobile hybrid networks. In *PiLBA*, 2008.

[3] C. Bernardos, M. Calderon, and H. Moustafa. Survey of IP address autoconfiguration mechanisms for MANETs, Nov. 2008.

[4] changewaveresearch.com. Google android continues to transform smart phone market, Mar. 2010.

[5] R. Dong and R. Kresman. Indirect disclosures in data mining. In *Frontier of Computer Science and Technology, 2009. FCST '09. Fourth International Conference on*, pages 346 –350, dec. 2009.

[6] Google. Android developers, 2010.

[7] D. Gurecki. Ad-hoc social networking for the google android platform, Apr. 2009.

[8] N. Hopper and E. Y. Vasserman. On the effectiveness of k-anonymity against traffic analysis and surveillance. In *WPES '06: Proceedings of the 5th ACM workshop on Privacy in electronic society*, pages 9–18, New York, NY, USA, 2006. ACM.

[9] S. Jazayeri. Mobility management with olsr protocol for fourth generation (4g) mobile networks. In *NGMAST '08: Proceedings of the 2008 The Second International Conference on Next Generation Mobile Applications, Services, and Technologies*, pages 598–603, Washington, DC, USA, 2008. IEEE Computer Society.

[10] A. Johnson and P. Syverson. More anonymous onion routing through trust. In *Computer Security Foundations Symposium, 2009. CSF '09. 22nd IEEE*, pages 3 –12, july 2009.

[11] A. Kornblum. Foreign intelligence surveillance act, 1978.

[12] P. Li, C. Zhang, and Y. Fang. Capacity and delay of hybrid wireless broadband access networks. *Selected Areas in Communications, IEEE Journal on*, 27(2):117 –125, february 2009.

[13] H. Mue, Ulfada, and B. Buxton. android-wifi-tether, Apr. 2010.

[14] J. Murray. Regulation of investigatory powers act, 2000.

[15] U. D. of Justice. Omnibus crime control and safe streets act, jun 1968.

[16] M. Reed, P. Syverson, and D. Goldschlag. Anonymous connections and onion routing. *Selected Areas in Communications, IEEE Journal on*, 16(4):482 –494, may 1998.

[17] J. Ren, Y. Li, and T. Li. Spm: source privacy for mobile ad hoc networks. *EURASIP J. Wirel. Commun. Netw.*, 2010:5–5, 2010.

[18] seekingalpha.com. Android passes iphone web traffic in the u.s., Apr. 2010.

[19] C. Tchepnda, H. Moustafa, and H. Labiod. Hybrid wireless networks: Applications, architectures and new perspectives. In *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, volume 3, pages 848 –853, sept. 2006.