Preserving Smartphone Users' Anonymity in Cloudy Days

Claudio A. Ardagna*, Mauro Conti†, Mario Leone† and Julinda Stefa‡
* Università degli Studi di Milano, Crema - Italy - Email: claudio.ardagna@unimi.it
† Università di Padova, Padova - Italy - Email: conti@math.unipd.it, mario.leone.23@gmail.com
‡ Sapienza Università di Roma, Roma - Italy - Email: stefa@di.uniroma.it

***Abstract*—The mobile cloud computing paradigm involves communications between smartphones and their virtual (software) clones in the cloud. It offers both backup/recovery solutions as well as offload of mobile computations, increasing the communication and computation capabilities of smartphones and making their limited batteries last longer. Unfortunately, in this scenario, the privacy of the users is at stake. The cellular network operator knows how often users contact the cloud, and the cloud provider knows how often users' clones contact each other. We address this privacy problem by providing an anonymous communication protocol, leveraging properties of social networks and ad-hoc wireless networks. Our solution provides anonymous end-to-end communication between two users in the network, and in turn between a user and her clone in the cloud. The proposal copes with an adversary model, where each party observing a portion of the communication (including the cloud provider and the cellular network operator) possibly colludes with others to uncover the identity of communicating users.**

I. Introduction

Recent advances of mobile technologies have turned our smartphones into small, powerful devices that we use not only to call and text, but also to play, check our emails, watch movies, wherever and whenever we are. Current smartphones in fact come with built-in 3G/WiFi and Bluetooth technologies, and their processors outperform those of the desktop computers of ten years ago. Although their high computational power and networking capabilities, smartphones are still limited by their battery and cannot cope with the energy required by the multitude of apps that we like to use everyday. To solve this problem, researchers are focusing on optimizing energy consumption [1], by offloading mobile computation to software clones of real devices in the cloud [2], [3], [4], [5], [6]. Platforms like C2C [7], [4], [8], where clones are peer-to-peer connected to each other in the cloud, allow for communication offloading, in addition to computation offloading. As an example, a device can rely on its clone to transfer a file to other devices, which in turn rely on their clone to receive it.

In the above scenario, many entities are involved: the devices, the clones, the cloud provider, and the cellular network operator. In particular, the last two handle all communications, and can monitor in their respective network: who is communicating with whom, how often, and which is the amount of data exchanged between them. If they collude, all this information can be easily inferred for end-to-end communications, thus threatening the privacy of all users in the system.

In this work, we address the above privacy problem by proposing a communication protocol that allows smartphone users to communicate in an anonymous way. To this aim, the sender of a communication involves other smartphones, clones, and the cellular operator in the communication towards the receiver. We assume communications that complete in few seconds, since they characterize many of everyday mobile communication sessions [9]. Our solution relies on opportunistic ad-hoc communications between smartphones and on social-network properties to provide an anonymous communication channel. We preserve end-to-end anonymity by supporting: *(a) user-to-clone anonymity* when a user communicates with her clone in the cloud; *(b) user-to-user anonymity* when two users communicate through a channel partially handled by their clones in the cloud. We design our scheme assuming all parties eavesdropping on the communication as potential adversaries that possibly collude among them.

We underline that our solution offers features that would not be available using Tor [10]. First, the authors of Tor [10] clearly states that providing resiliency against end-to-end attack (e.g., end-to-end time and size correlations) is a non-goal for Tor, while it is mandatory in our scenario. Furthermore, Tor assumes an adversary observing only a fraction of the network traffic. In our scenario, where communication patterns always involve the clones of the sender and receiver devices, this assumption does not hold. As an example, the cloud provider, which eavesdrops on all communications in its domain, can act as a global attacker in the communication between the clones of the sender and receiver devices. Moreover, our solution relies on a multi-hop WiFi local communication that hides the initiator of the communication to the cellular network operator, that is, the sender using our anonymous communication protocol. This feature would not be available if the sender device directly asks the cellular network operator to contact a Tor Proxy. Finally, Tor assumes a significant amount of traffic being present in the network to provide anonymity, while this is not needed in our solution.

To the best of our knowledge, this is the first work that considers anonymity protection in mobile-cloud integrated infras-

Mauro Conti is supported by a Marie Curie Fellowship funded by the European Commission for the PRISM-CODE project (Privacy and Security for Mobile Cooperative Devices) under the agreement n. PCIG11-GA-2012-321980. This work has been partially supported by the TENACE PRIN Project 20103P34XC funded by the Italian MIUR and by the EU-funded project CUMULUS (contract n. FP7-318580).

978-1-4673-5775-3/13/$31.00 ©2013 IEEE

tructures. The contribution of the proposed solution is therefore threefold: *i)* it provides a complete end-to-end anonymous channel, allowing users to anonymize their communication profiles against adversaries including the cellular operator and the cloud provider; *ii)* it supports anonymous communications between a smartphone user and her clone, thus protecting personal information on the interests and behaviors of the users; *iii)* it allows low-cost and battery-preserving communications.

II. Related Work

Achieving anonymous communications is an important and well studied issue in both wired and wireless systems. So far, the most applicable schemes are based on the concept of mixing [11], where messages are sent along a chain of proxy nodes (called mixes) that accumulate and forward source-encrypted messages in batches. Tor [10], perhaps the most popular deployed mix network in wired systems, achieves mixing by layer-encrypting a message at the source, and decrypting it once at each hop of a source-selected chain of proxy relays (called also Tor nodes). The last Tor node of the chain sends the unencrypted message to the destination specified by the client. The enormous popularity of social network platforms has given raise to new anonymity schemes that rely on friendship relations among users. By assuming that friend nodes trust each other, schemes like [12], [13], among others, show how anonymity mechanisms based on social-trust offer the same anonymity properties of Tor, yet lowering the delay of the communication and alleviating the computation burden of the source.

Early approaches proposing anonymous communication schemes for wireless systems are mostly inspired by Tor-like networks [14], [15], [16]. They either rely on source-routing, or assume a reliable network where full connectivity among peers is available. These requirements make them inapplicable in highly dynamic mobile wireless networks like the Pocket Switched Networks (PSNs). In PSNs, users carry around devices communicating with short-range technology (Bluetooth or WiFi), and the communication links appear and disappear over time, as the device holders get in physical proximity. To the best of our knowledge, the solutions in [17], [18], [19] are the only approaches that cope with the intermittent nature of PSNs. The ALAR protocol [19], inspired by [12], [13], makes use of social-communities in the network to protect data-sources from being localized, while maximizing the destination probability to receive the message. However, ALAR does not protect the identity of the destination of the message. The work in [18], instead, combines threshold-based cryptography along with randomly selected pivot nodes to provide resistance to traffic analysis, source anonymity, and sender-receiver unlinkability. All these schemes envision a scenario in which nodes communicate in multiple hops, without a fixed networking infrastructure. In addition, they are all based on the social-trust: “friends” rely on their “friends” in the network to achieve the anonymity required in the system.

More recently, the paper in [20] presents an anonymous communication protocol aimed to preserve (α,β)-anonymity in mobile hybrid networks, involving cellular and WiFi communication links. Although this approach provides communication anonymity in a scenario where mobile users move and form networks of arbitrary topology, it is not applicable to our settings, where the smartphones constantly communicate with their software clones in the cloud to either offload computation [2], [3], [4], [6] or backup/store data for reliability reasons [5]. We assume a system as the one presented in [4], where each smartphone is associated with its clone in the cloud, and clones of “friend” users (according to an on line social network, e.g. Facebook) are inter-connected by P2P links between them. We also assume an attack model where entities intercepting part of the communication are considered as adversaries. In this context, the cellular network operator and the cloud eavesdrop on every communication going on in their own network, and collude to uncover source and destination pairs involved in the communication. Under our attack scenario, we design a protocol that provides anonymous communications between a smartphone user and her clone-device, as well as between two smartphone users as end-points of a communication involving their respective clones.

III. System setting and adversary model

Here, we define our system (Section III-A), anonymity (Section III-B), and adversary (Section III-C) models.

A. System Model

Our system includes four different parties: *i)* the smartphone users, carrying mobile devices for communication and managing a clone of their smartphone in the cloud; *ii)* the cellular network operator, managing the cellular infrastructures and allowing smartphone users to access its services; *iii)* a set of proxies, mediating requests from a smartphone user to a clone in the cloud; and *iv)* the cloud provider, managing the cloud infrastructure and its computing resources.

Figure 1 presents our system model, where communications among the entities are denoted with black arrows. The smartphone users communicate with each other through both the cellular network infrastructure (mid-layer in Figure 1) and short-range ad-hoc wireless communication links (bottom layer in Figure 1). We denote with D the set of mobile devices (and with C the set of clones). Each device $d_i \in D$ is associated (dotted lines in Figure 1) with a software clone $c_i \in C$ in the cloud platform, for offloading computations and communications, and for backup purposes. Slightly abusing the notation, in the following we refer to d_i and c_i also as the *user* of d_i and of c_i. In addition, o and cp denote respectively the cellular network operator and the cloud provider. We assume a single cellular network operator o receiving all cellular communications, and acting as a gateway between mobile peers and the cloud infrastructure. Also, we assume the presence of a single cloud provider cp.

The clones are inter-networked to any other clone through P2P links in the cloud. In particular, some of these links (dashed lines in Figure 1) reflect the friendship relations among the respective users, and define well connected social-communities of clones in the cloud. These links can be either declared by the users, bootstrapped by existing online social networks, or a mix of the two. We denote with S_{d_i} the set

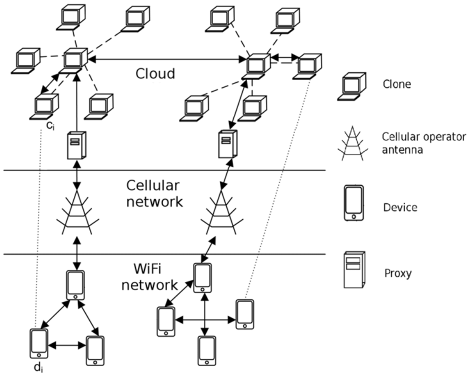

Fig. 1. System components and relative communication links.

of devices whose users are friends with the user of device d_i. Similarly, S_{c_i} is the set of clones that are friends with c_i. Intuitively, S_{d_i} and S_{c_i} are of the same size, and a user might belong to more, and possibly, overlapping sets of friends (e.g., those defined by her gym friends and her school friends). As in [12], [13], we assume that $|S_{d_i}|>1$ and that there is a trust relationship among friend-users. The same relationship holds also among the respective clones. We assume that the friendship information of users in the system is public and accessible through a centralized friendship database (e.g., via Facebook). In addition, we assume that each device d_i shares a secret key SK_{d_i} with its clone c_i (which is used to preserve confidentiality in the communication between them), and each clone c_i is provided with a public/private key pair ($K^p_{c_i}$,$K^s_{c_i}$).

Our system architecture also includes a set of n proxies $pr_1, \ldots, pr_n$, that mediate all communication channels between the devices and the clones in the cloud. Proxies handle the user registration to the system and keep a map between clone ID and clone public IP. The role of the proxies is to decouple cellular and cloud identities, further complicating attacks to the anonymity of our protocol by colluding o and cp. Each proxy pr_i is provided with a public/private key pair ($K^p_{pr_i}$,$K^s_{pr_i}$) used when communicating with user's devices and clones. A set of proxies instead of a single proxy is assumed to cope with the traffic load triggered by a potentially high number of users in the system. The load balancing is totally transparent to the users that perceive a single proxy entity. For simplicity, in the following, we consider a single proxy pr.

Finally, we consider each communication among devices as a (bi-directional) exchange of a set of messages (black arrows in Figure 1) between the communication originator $s\in D$ and the communication receiver $r\in D$.

B. Anonymity Model

Our goal is to anonymize the end-to-end communication between a device and its clone, and between two devices communicating through the cloud infrastructure. For this second case, extending the concept of k–anonymity, we aim at hiding the originator of the communication among (at least) α possible devices, as well as hiding the receiver among (at least) β possible devices, with α and β chosen by the originator. Formally, we use the following definition.

Definition 3.1 ((α,β)–anonymity): Given an originator s and a receiver r, a communication between them is (α,β)–anonymous, if an adversary Adv cannot associate device s to less than α devices, and device r to less than β devices.

In particular, we observe that the general case is for $\alpha \neq \beta$ and α,β>1, while specific scenarios correspond to different levels of anonymity: s and r are protected at the same level ($\alpha = \beta$); s can be identified as being the originator of a communication with one among β possible receivers ($\alpha = 1$); r can be identified as being the receiver of a communication with one among α possible originators ($\beta = 1$); s and r are unambiguously identified as the originator and receiver of a communication (no anonymity) (α,$\beta = 1$). Our aim is to guarantee end-to-end (α,β)–anonymous communications against adversaries with different knowledge and capabilities that can possibly collude (Section III-C).

C. Adversary Model

Users communications, being through either WiFi, cellular, or cloud network, are exposed to attacks from adversaries operating in these networks. The adversaries in our model are honest but curious, meaning that they neither tamper with the exchanged messages nor are able to read encrypted messages. The adversaries are grouped in four categories as follows.

- *Malicious devices*. They are devices owned by malicious users in the WiFi network that intentionally attack the anonymity of mobile users involved in a communication.
- *Cellular network operator*. It does not tamper with the communication channel and with the traffic packets. Yet, it observes all cellular communications between users, and users and clones. Cellular network operator can measure the position of each device, at least by observing the cellular antenna it joins.
- *Cloud provider*. It does not tamper with the clones running on its machines and with the networking channels. Yet, it eavesdrops all the in/out traffic for every clone.
- *Malicious clones*. They refer to clones under the control of malicious devices. They intentionally attack the anonymity of mobile users involved in a communication.

All adversaries can collude among them to the aim of identifying source s and destination r of the communication. The attacker goal is to uncover identities of s and r involved in an end-to-end communication, by reducing the anonymity in Definition 3.1 to (1,1)–anonymity. We note that the proxies are trusted and do not collude with any of the adversaries, although our solution is resilient to the scenario in which they are compromised by malicious adversaries. In addition, devices (and corresponding clones) trust other devices (and corresponding clones) with which they are in a friendship relation; in other words, a device or clone does not attack or collude with an adversary to compromise the anonymity of a friend device or clone.

IV. Our Solution

The main goal of our protocol is to achieve (α, β)–anonymity for communications between a sender s and a

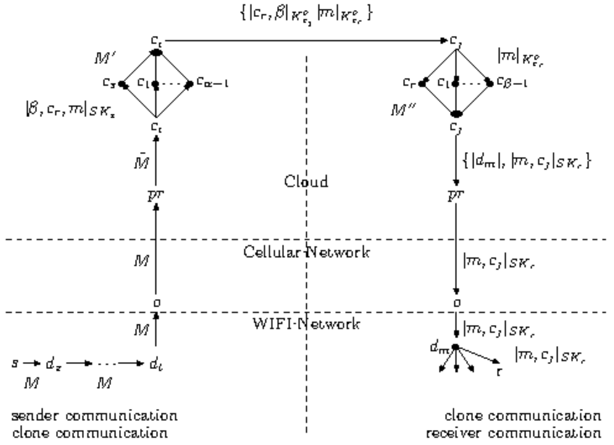

Fig. 2. Communication flow within our protocol

receiver r, with part of the communications handled by their clones c_s and c_r in the cloud. In our protocol, sender s and receiver r communicate as follows. First, s anonymously sends a message to its clone c_s (sender communication). This is achieved through a probabilistic multi–hop WiFi forward to devices in its physical proximity and a forward in the cloud through a proxy and a set of clones. Second, upon reaching c_s, the message is anonymously forwarded to clone c_r of r (clone communication). This involves a clone in the same social network with s and a clone in the same social network with r. Third, upon reaching c_r, the message is (possibly) distributed to mobile device r, via a proxy, the cellular network operator, and a device in the proximity of r (receiver communication). Finally, receiver r (or directly its clone) can reply to sender s by either re-using the same path or by building a new one (response communication). Figure 2 depicts the communication flow in our scheme illustrating the content of each message. The details of these steps are discussed in the remaining of this section. To this aim, we consider s defining anonymity preferences α and β, and sending a message with payload m to r. Before an anonymous communication can start, s must register to the system using trusted proxy pr. In particular, it sends to pr its identity, a proof of its identity, and a secret key $SK_{s,pr}$ to be used in all communications with the proxy, encrypted with public key K^p_{pr}.

Sender Communication. User s looks up in the friendship database and randomly selects a friend clone c_i with more than α social network friends ($|S_{c_i}| > \alpha$). It then sends the message to proxy pr, using a probabilistic multi–hop WiFi forward to devices in its physical proximity, that will forward the message to c_i. At this stage c_i forwards the message to α of its friends including c_s, the only one able to decrypt the message. To this aim, user s prepares a bundle M that contains *(a)* the identity of c_i encrypted by $SK_{s,pr}$ (the shared key between s and pr), *(b)* the identity of c_s and parameter α encrypted with $K^p_{c_i}$ (the public key of c_i); *(c)* parameter β, the identity of clone c_r and payload m encrypted with SK_s (the shared secret key between s and c_s) as follows: $M = \{[c_i]_{SK_{s,pr}}, [c_s, \alpha]_{K^p_{c_i}}, [\beta, c_r, m]_{SK_s}\}$. Then, s checks the number of devices in its proximity, reachable by WiFi ad-hoc networking links. This can be achieved by having devices periodically broadcast *hello* messages through the WiFi network, to announce their presence to neighbors in their proximity. User s waits till it is "surrounded" by at least α other devices, and then sends M with probability p to device d_z, randomly selected among the nearby devices, and with probability 1-p directly to pr through cellular network operator o. This step is necessary to avoid a device in user's proximity from guessing with probability higher than $1/\alpha$ the identity of s. After receiving M, device d_z chooses with probability p to forward it through WiFi to a nearby device d_t, and with probability 1-p to send it through the cellular network operator o to proxy pr. Intuitively, the probabilistic message forward aims to confuse a curious cellular network operator: in fact, the cellular network operator might leverage the user identifier (e.g., the SIM number) and information on the position of forwarding device d_t to guess the identity of sender s.

When proxy pr receives the bundle M, it retrieves the identity of c_i using the shared key $SK_{s,pr}$ and forward the rest of the bundle $\tilde{M} = \{[c_s, \alpha]_{K^p_{c_i}}, [\beta, c_r, m]_{SK_s}\}$ to c_i. We note that pr can neither access the identity of c_s nor of c_r. Upon clone c_i gets $\tilde{M}$, it retrieves the identity of c_s and parameter α by decrypting $[c_s, \alpha]_{K^p_{c_i}}$ with its private key $K^s_{c_i}$. Then, it forwards the remaining of the bundle, $[\beta, c_r, m]_{SK_s}$, to a subset of α clones in its social network including c_s. Among the α clones receiving the message, only c_s is able to decrypt $[\beta,c_r,m]_{SK_s}$ using SK_s.

Clone Communication. Each of the α clones receiving $[\beta, c_r, m]_{SK_s}$ replies, after a timespan τ, with a response to c_i. For c_s, the response is a bundle M' which will be then forwarded towards destination clone c_r. The responses of the other clones are randomly generated, have variable lengths, and are encrypted with c_i's public key. In this way, cloud provider cp cannot tell, among the α replying clones, which one is the actual source clone c_s.

Let us get back to M' contained in the response of c_s sent to clone c_i. To generate it, clone c_s looks up in the friendship database and picks, from c_r's friend set, a clone c_j with more than β friends. Then, it prepares a bundle M' containing *(a)* the identity of c_j, *(b)* the identity of clone c_r and parameter β encrypted with $K^p_{c_j}$ (the public key of c_j), *(c)* payload m encrypted with $K^p_{c_r}$ (the public key of c_r). The resulting message is then encrypted with the public key of clone c_i and looks as follows: $M' = \{c_j, [c_r, \beta]_{K^p_{c_j}}, [m]_{K^p_{c_r}}\}_{K^p_{c_j}}$. M' is then sent back to c_i, after timespan τ elapses.

Upon receiving M' from c_s, c_i decrypts it, retrieves c_j, and forwards message $\{[c_r, \beta]_{K^p_{c_j}}, [m]_{K^p_{c_r}}\}$ to c_j. The responses of the other clones are simply dropped. Upon receiving $\{[c_r, \beta]_{K^p_{c_j}}, [m]_{K^p_{c_r}}\}$, c_j decrypts $[c_r, \beta]_{K^p_{c_j}}$ using its private key $K^s_{c_j}$. Then, c_j forwards $[m]_{K^p_{c_r}}$ to a subset of β clones in its social network including c_r. Among clones receiving the message, only c_r is able to decrypt $[m]_{K^p_{c_r}}$ using its key $K^s_{c_r}$.

We note that clone c_r might be the actual destination of the message. In this case, the protocol ends; otherwise, the protocol proceeds with the *receiver communication* step.

Receiver Communication. If the destination of the communication is not clone c_r, but the actual device r, a further communication step is needed. We propose a solution that follows a *push* approach. First, similarly to the previous phase, all β clones that received a message from c_j, after a timespan τ, reply to it with a response. The response of c_r is a bundle M'' that includes the message to be sent to the real device r. M'' contains *(a)* the identity of a device d_m in the proximity of r (reachable through WiFi), *(b)* the payload m and the identity of c_j encrypted with SK_r (the secret key of r), as follows: $M'' = \{[d_m], [m, c_j]_{SK_r}\}_{K^p_{c_j}}$. M'' is encrypted with the public key of clone c_j and sent to it as a response, after timespan τ elapses. As in the previous case, since β responses are sent to c_j from β different clones, cp cannot tell which is the one generated by the destination clone, and containing the actual message for the real device.

Upon receiving M'', c_j sends $\{[d_m], [m, c_j]_{SK_r}\}$ to pr, which in turn retrieves the identity of d_m and forwards $[m, c_j]_{SK_r}$ to it. Then, d_m broadcasts the received message to nearby devices. Among other devices, r receives the message, decrypts it with SK_r, and reads the payload.

We note that, for this to work, c_r must know which devices are currently in physical proximity of r. To this aim, r periodically notifies c_r of neighboring devices around it, that is, the ones from which it receives an *hello* message, with more than β neighboring devices. Neighboring devices with less than β devices in their proximity would in fact expose the anonymity of r, if selected as destination d_m. This communication between a device and its clone is the only one happening outside our protocol.

Response Communication. If the communication between s and r is bi-directional, there are two possible approaches as follows: *i)* the message from r to s follows the same path used by the message received by r; *ii)* the protocol is repeated as a one-way communication switching s with r. For the first to be possible, each clone c_i and c_j involved in the communication must keep track for each session of the α and β clones, respectively, that they have used to anonymize the communication. Also, r must select c_j as the supporting clone for message forward. To this aim, message M'' contains the identity of c_j.

It can be proven that our protocol guarantees (α, β) anonymity against the following attack scenarios: single adversaries, colluding adversaries belonging to the same category (e.g., colluding clones), and colluding adversaries belonging to different categories (e.g., cloud provider colluding with cellular operator). We omit the proof due to space restrictions.

V. Conclusions

In this paper we addressed the anonymity concerns of end-to-end communicating smartphone users, where clones of the mobile devices handle part of the communication. Our solution provides anonymous end-to-end communication between two users in the network, and in turn between a user and her clone in the cloud, by leveraging on properties of social networks and ad-hoc wireless networks. We considered each party observing a communication as honest but curious, and possibly colluding with others to uncover the identity of communicating users. Our protocol provides the required anonymity in each of these scenarios.

References

[1] M. Conti, D. Diodati, C. M. Pinotti, and B. Crispo, "Optimal solutions for pairing services on smartphones: a strategy to minimize energy consumption," in *Proc. of IEEE CPSCom '12*, 2012.

[2] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "Maui: making smartphones last longer with code offload," in *Proc. of MobiSys '10*, 2010.

[3] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading." in *Proc. of IEEE INFOCOM '12*, 2012.

[4] S. Kosta, C. Perta, J. Stefa, P. Hui, and A. Mei, "Clone2Clone (C2C): Peer-to-Peer Networking of Smartphones on the Cloud," in *Proc. of USENIX HotCloud '13*, 2013.

[5] M. V. Barbera, S. Kosta, A. Mei, and J. Stefa, "To Offload or Not to Offload? The Bandwidth and Energy Costs of Mobile Cloud Computing," in *Proc. of IEEE INFOCOM '13*, 2013.

[6] G. Portokalidis, P. Homburg, K. Anagnostakis, and H.Bos, "Paranoid android: versatile protection for smartphones," in *Proc. of ACSAC '10*, 2010.

[7] M. V. Barbera, S. Kosta, J. Stefa, P. Hui, and A. Mei, "CloudShield: Efficient anti-malware smartphone patching with a P2P network on the cloud," in *Proc. of IEEE P2P '12*, 2012.

[8] S. Kosta, V. C. Perta, J. Stefa, P. Hui, and A. Mei, "CloneDoc: Exploiting the Cloud to Leverage Secure Group Collaboration Mechanisms for Smartphones," in *Proc. of IEEE INFOCOM 2013*, 2013.

[9] A. Rahmati, C. Shepard, A. Nicoara, L. Zhong, and P. Singh, "Mobile tcp usage characteristics and the feasibility of network migration without infrastructure support," in *Proc. of ACM MobiCom '10*, 2010.

[10] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second–Generation Onion Router," in *Proc. of USENIX Security '04*, 2004.

[11] L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Commun. ACM*, vol. 24, pp. 84–90, 1981.

[12] K. Puttaswamy, A. Sala, O. Egecioglu, and B. Zhao, "Rome: Performance and anonymity using route meshes," in *Proc. of IEEE INFOCOM '09*, 2009.

[13] K. Puttaswamy, A. Sala, and B. Zhao, "Starclique: guaranteeing user privacy in social networks against intersection attacks," in *Proc. of CoNEXT '09*, 2009.

[14] A. Boukerche, K. El-Khatib, L. Xu, and L. Korba, "Sdar: A secure distributed anonymous routing protocol for wireless and mobile ad hoc networks," in *Proc. of IEEE LCN '04*, 2004.

[15] S. Seys and B. Preneel, "ARM: anonymous routing protocol for mobile ad hoc networks," *Int. J. Wire. Mob. Comput.*, vol. 3, pp. 145–155, 2009.

[16] R. Song, L. Korba, and G. Yee, "Anondsr: efficient anonymous dynamic source routing for mobile ad-hoc networks," in *Proc. of ACM SASN '05*, 2005.

[17] A. Davoli, A. Mei, and J. Stefa, "Fan: friendship based anonymous routing in wireless social mobile networks of malicious communities," in *Proc. of ExtremeCom '11*, 2011.

[18] R. Jansen and R. Beverly, "Toward Anonymity in Delay Tolerant Networks: Threshold Pivot Scheme," in *Proc. of MILCOM '10*, 2010.

[19] X. Lu, P. Hui, D. Towsley, J. Pu, and Z. Xiong, "Anti-localization anonymous routing for delay tolerant network," *Computer Networks*, vol. 54, no. 11, pp. 1899 – 1910, 2010.

[20] C. Ardagna, S. Jajodia, P. Samarati, and A. Stavrou, "Providing mobile users' anonymity in hybrid networks," in *Proc. of ESORICS '10*, 2010.