CloneDoc: Exploiting the Cloud to Leverage Secure Group Collaboration Mechanisms for Smartphones

Sokol Kosta*, Vasile Claudiu Perta*, Julinda Stefa*, Pan Hui†, and Alessandro Mei*
*Department of Computer Science, Sapienza University of Rome, Italy,
†Deutsche Telekom Innovation Laboratories T-Labs, Berlin, Germany,
Email: *Email: surname@di.uniroma1.it, †pan.hui@telekom.de.

***Abstract*—The limited battery of smartphones makes it hard for users to fully exploit their devices. Recent research considers offloading mobile computation to the cloud to improve efficiency and to extend battery life. We design Clone2Clone (C2C), a distributed p2p platform for software cloud clones of smartphones. Upon C2C we implement CloneDoc, a secure real-time collaboration system for smartphone users. We measure the performance of CloneDoc on a testbed of six Android smartphones and clones hosted on the Amazon cloud service and show that C2C based protocols can significantly improve battery-life compared to protocols that do not make use of C2C.**

I. Introduction

The enormous boost of sales in the smartphone market (more than 900 M shipped around the world) has attracted the attention of many developers and researchers. Current smartphones allow people not only to call and text, but also to use “apps” to send emails, tweet, play video-games, shop, watch videos, take pictures, edit, index, and upload them to their favorite social network, etc. However, all of this comes at a price—battery life. Indeed, there is a huge gap between the battery capacity and the energy necessary to use current mobile devices at full power. From the other side, cloud computing has attracted the attention of the major IT multinationals (Google, Amazon, Microsoft) to the point to make them invest substantially in cloud services, and it is seen as a perfect candidate to help offload computation of mobile applications. Recent works [1], [2], [3], [4], [5] show how method or application partitioning-based offloading techniques to the cloud drastically improve computation efficiency and prolong battery life of mobiles.

In this demo we show how to push the smartphone-cloud paradigm to a further level: We demonstrate the capabilities of Clone2Clone (C2C), a distributed platform for cloud clones of smartphones, and CloneDoc, a real-time collaboration system built on top of C2C. C2C associates a software clone on the cloud to every smartphone and interconnects the clones in a p2p fashion exploiting the networking service within the cloud. In this way, C2C enables innovative services such as content sharing, search, and distributed execution among the users, and

This work has been technically supported and partially funded by Telecom Italia within the Working Capital project.

This work has been performed in the framework of the FP7 project TROPIC IST-318784 STP, which is funded by the European Community. The Authors would like to acknowledge the contributions of their colleagues from TROPIC Consortium (http://www.ict-tropic.eu).

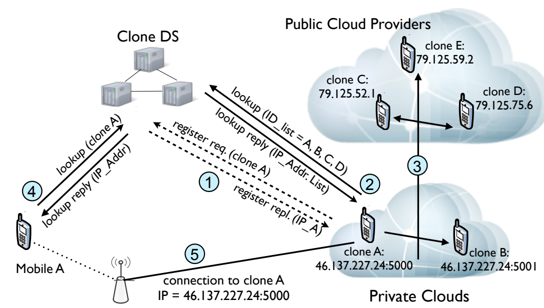

Fig. 1. The C2C architecture and networking. The numbered steps represent what happens in the system after the generation of a new clone.

it makes possible to build p2p-based protocols for smartphones without the need of relying on a continuous connection between real devices. Figure 1 highlights the baseline architecture of C2C. It includes the CloneDS, a *directory service* that takes care of mapping users to clones and clones to IPs. It is always up and its IP is known (made public by e.g. the C2C platform builder). All the entities in the system—users, cloud providers, and the CloneDS—have a private/public key pair and they can securely verify the authenticity of public keys.

On top of C2C we implement CloneDoc, a real-time collaboration system for smartphone users working simultaneously on the same document. We use CloneDoc as a paradigmatic p2p-like application that makes use of heavy crypto primitives and communication among peers. Our experiments show that by making use of C2C with the CloneDoc system we enable energy saving to the real devices by offloading computation of heavy tasks to the cloud. Simultaneously, by offloading also communication, the C2C platform allows for cellular bandwidth savings. So, we demonstrate that C2C makes possible to run a whole new class of distributed applications on mobile devices.

II. Secure Group Collaboration on Smartphones

Secure group collaboration and file access among multiple clients can be efficiently deployed by making use of external servers running on the cloud [6]. These applications need p2p networking among the clients (that do not run on the cloud) and heavy cryptography to guarantee crucial security and system properties. This makes even SPORC [6], the state of the art of real-time collaboration systems, unfit for present-day smartphones. However, with the C2C platform the tables are turned. C2C delivers efficient p2p networking for smartphones by moving computation and, most importantly, communication on the cloud. We exploit these features of C2C and modify SPORC's architecture to build CloneDoc—The first energy-efficient and real-time collaboration system for battery constrained smartphones.

A. *CloneDoc: Secure group collaboration on smartphones*

CloneDoc's purpose is the same as SPORC's [6]: a system for group collaboration with an untrusted server whose role is to force a global order on the concurrent users' operations. The server is potentially malicious—its goal may be to partition the clients in disjoint groups with different views of the document. CloneDoc encouples Operational Transformation (OT) [7], with Fork* [8] consistency checks that (1)prevent the distortion of clients' document view from the server; (2) allow the execution of lock-free concurrent operations that preserve casual consistency and make the clients converge to a common shared state.

In CloneDoc the clone on the cloud receives operations from the mobile device, handles as many tasks as possible on the device's behalf and keeps the device up-to-date on the modifications submitted by other users. More specifically: (a) it submits to the server operations that he gets from the user's real-device; (b) appropriately transforms the operations of the other users received from the server so that its view of the document is coherent to that of other clones; (c) handles all the security checks including signature verification for the server and other users, RSA encryption of operations submitted by the user, Fork* consistency checks to detect server misbehavior. Last, but not least, the clone makes sure that the user's view is coherent to that of other users in the system.

Inheriting from SPORC [6], CloneDoc is a typical p2p application and induces not-so-light computations due to cryptography. Nonetheless, by building it on top of the C2C platform, we show how CloneDoc leverages on the cloud to reduce battery consumption of real devices as well as diminish cellular traffic by offloading communication to the clones. Indeed, our preliminary experiments show that 99% of the signatures, 99% of the signature verifications, and 93.75% of the RSA encryption operations are offloaded to the clone. These operations are a fundamental and crucial part of the system since they recur in many tasks such as addition and removal of a user, client-server communication, user-user communication, update of a user's state after her reconnection. In particular, C2C makes it possible to implement an advanced peer-to-peer service as CloneDoc in a network of mobile smartphones saving 99%, 80%, and 30% of the battery for respectively security checks, user status update and document editing. In addition, as the smartphones execute less heavy tasks, also the responsiveness of the system improves. We compare the responsiveness of CloneDoc (which exploits C2C) to that of SPORC in two cases: (1) a temporarily disconnected user reconnects to the system; (2) during an editing phase. It is crucial that a user gets the operations generated by the other users as fast as possible and that a temporarily disconnected user gets its status updated quickly. Note that temporary disconnection of a smartphone can be very frequent for many reasons (e.g. low coverage, overload of the cellular network, etc.). With CloneDoc the execution of these processes is much faster than with SPORC—around 5 times faster for the status update and 1.8 times faster to apply all users' operations to the document (see Figure 2).

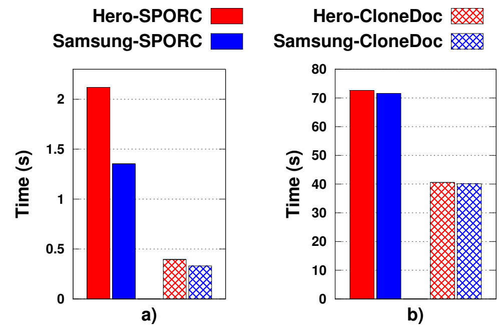

Fig. 2. Time to (a) update the state of a temporarily disconnected user; (b) apply all users' operations during the editing. The results are shown for a Samsung Galaxy Plus and for a HTC Hero smartphone.

III. Demo setup

To setup our demonstration we use six real devices running CloneDoc and an equal number of clones deployed on the Amazon EC2 cloud platform. We also have an additional device running the original protocol SPORC, that does not make use of the cloud. All seven smartphones are Samsung Galaxy S Plus and use WiFi/3G to communicate with both the untrusted server and the clones. To avoid biased results due to other possible applications running simultaneously, we disable all unnecessary phone services and applications.

During the demonstration the users (demo presenters and possible volunteers) use the system at their own preference: Perform text editing (both adds and deletions) and group dynamics (adding and removing a randomly chosen user in the system). In addition, the users will set-up their preferred rate of scheduled security checks that protect from possible server misbehavior. While the system is up and running, we show, in real time, the energy savings and the enhanced response time of the CloneDoc system that exploits C2C as opposed to the SPORC system that runs entirely on the smartphone.

References

[1] B. G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in *Proc. of EuroSys '11*, 2011.

[2] 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 2012*, 2012.

[3] 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 2012*, 2012.

[4] 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 2013*, 2013.

[5] M. V. Barbera, S. Kosta, A. Mei, V. C. Perta, and J. Stefa, "CDroid: Towards a Cloud-Integrated Mobile Operating System," in *Proc. of IEEE INFOCOM 2013*, 2013.

[6] A. J. Feldman, W. P. Zeller, M. J. Freedman, and E. W. Felten, "Sporc: Group collaboration using untrusted cloud resources," in *Proc. of OSDI '10*, 2010.

[7] C. A. Ellis and S. J. Gibbs, "Concurrency control in groupware systems," *SIGMOD Rec.*, vol. 18, pp. 399–407, June 1989.

[8] D. Nichols, P. Curtis, M. Dixon, and J. Lamping, "High-latency, low-bandwidth windowing in the jupiter collaboration system," in *Proc. of ACM UIST '95*, 1995.