# Providing Security and Privacy in Smart House Through Mobile Cloud Computing

Rafael Ribeiro[1], Altair Santin[1], Vilmar Abreu[1], João Marynowski[1,2], Eduardo Viegas[1]
[1]Graduate Program in Computer Science / Pontifical Catholic University of Parana / Curitiba, Parana, Brazil
[2]Analysis and Development Systems / Federal University of Parana / Curitiba, Parana, Brazil
{rcr_raf, santin, vilmar.abreu, jeugenio, eduardo.viegas}@ppgia.pucpr.br

***Abstract*—Smart house allows residents to access appliances and smart grid data to reconfigure its power consumption. This paper aims to provide security and privacy by design in a smart house, isolating appliances from direct Internet access, while allowing external entities to reach power consumption features. Data acquisition, processing, and consumer profiles updates are only possible inside a smart house, using a Mobile Cloud Computing (MCC). MCC enables to reconfigure consumption profile of smart houses, for instance, to avoid a potential blackout. In addition, the proposal improves security and privacy for smart houses, preventing an attacker to remotely control appliances or violating the privacy of residents by monitoring their energy consumption habits. Prototype tests showed the feasibility of our proposal, isolating the appliances and MCC mobiles, besides collecting, processing, and storing 1200 appliances attributes in 15.25 seconds while using only three mobile devices in an MCC.**

***Keywords—Security; Privacy; Smart house; Mobile Cloud Computing; SCADA***

## I. INTRODUCTION

Smart devices (appliances) are household items with computational processing power, connected through a wireless network and generally accessible from the Internet [3]. A smart house has appliances to provide several benefits to residents, such as comfort, controllability, and energy efficiency. Advances in the smart house scenario become more relevant when combined with a smart grid. Smart grid aims to improve performance, reliability, and capacity of interaction between the electric power company (EPC) and its customers, helping to make decisions about energy consumption by customers (residents) and energy supply by EPC [2].

Supervisory Control and Data Acquisition (SCADA) system enables the control and data acquisition of components from various infrastructure services [18]. For example, SCADA can facilitate the power load adjustment of an electric grid, requiring customers to reconfigure their consumption profiles to reduce the blackouts occurrence. Then, smart grid uses the benefits of smart house automation to perform possible energy consumption reconfiguration. Moreover, users can change energy consumption profiles to adapt to seasonal electricity rates for reducing costs, for instance.

Some approaches enable residents to change energy consumption profiles through embedded appliances systems, which are accessible directly from the Internet [1, 15, 22]. Proposals allow reconfiguring personalized energy consumption profiles but they require an additional cost to maintain embedded systems for that. Moreover, exposing appliances to the Internet implies risks to security and privacy of residents.

Appliance vulnerabilities may be exploited, having resident data exposed and even controlled by an attacker [16]. Proofpoint Inc., a leading security-as-a-service provider, reported a global attack involving more than 750,000 malicious e-mails coming from more than 100,000 appliances [4]. Additionally, mobile devices also have been carelessly used to control appliances remotely, either inside or outside smart houses, increasing the risk of threats [16].

In our proposal, mobile devices compose a Mobile Cloud Computing (MCC). Our hypothesis is that it is possible to provide security and privacy by design in a smart house, isolating appliances from the Internet while providing energy consumption functionalities, which are only accessible through an internal MCC. The MCC avoids the need for extra devices, unifying the control and operation of embedded appliances and enabling residents to manage consumption profiles. In the absence of residents, profiles are used to perform a required energy consumption reconfiguration.

In the following, Section II addresses smart house, MCC, and referred security and privacy. Section III discusses related work. Section IV presents the proposal of this work. Section V presents a use case scenario. Section VI presents the prototype and evaluation while Section VII presents conclusions.

## II. FUNDAMENTS

A smart house has a set of appliances, also known as things on the Internet of Things (IoT) [3]. Appliances interact with its users and the environment to optimize its operations, improving the effectiveness of its functions. Typically, appliances have limited computational and communication resources and provide their features remotely, using a wireless network and web services. An appliance provides generic or specific attributes [3]. Generic attributes are common to all appliances, such as instant, hourly or daily consumption. Specific attributes are dependent on the appliances, such as component status or the need for a firmware update.

Smart grid involves aspects related to the automation of a power grid, establishing a bidirectional flow of electricity and information about the grid [2]. The company can obtain the grid information, such as device status (e.g. normal and alarm) and individual consumptions of residents, in real-time. As well as the resident, can get scheduled interruptions and seasonal electricity rates from electric power companies. A smart grid consists of devices (smart or not) connected in a complex system of networks: (i) internal network of a single user (Home-Area Network, HAN); (ii) regional network that comprises several HAN (Neighborhood-Area Network, NAN); and a global network that comprises multiple NAN (Wide-Area Network, WAN) [2].

SCADA and Metering Data Management (MDM) systems manage the production and distribution of energy in a smart grid, controlling devices and allowing interaction with smart houses [18, 21]. Smart meter is the HAN interface, with two bidirectional flows, one for the needs of power grid management and other to exchange information between smart house and smart grid [2].

Mobile Cloud Computing (MCC) defines cloud computing using mobile devices, aiming to increase the processing power, storage, and energy efficiency of mobile devices [19]. Mobile devices compose an MCC to provide services locally.

Risks to privacy and security on smart house involve threats, both internal as from the smart grid. Moreover, as appliances and smart meter can be operated from the Internet, they are also exposed to threats from the Internet [5, 21]. Privacy is also considered critical because the information stored and communicated inside an HAN may expose behaviors and habits of smart house residents. Some activities and appliances can be identified from the electrical consumption patterns, e.g. someone can infer whether residents are at home or traveling, watching television or in the shower, or if the electric fence is turned on or off [21].

## III. Related Works

Some works proposed the local management of appliances using a wireless network and a server accessible locally or from the Internet [12, 13]. Korkmaz et al. [22] proposed changing a local specific server by a cloud server to manage more than one smart house simultaneously. Perera et al. [24] used smartphones to collect and process appliances data to be stored on a local server or a cloud server. Note that exposing appliances to the Internet, either directly or using smartphones, exposes the smart house to threats. In addition, using a server implies extra energy consumption and hardware either cost, besides inserting a single point of failure, local or in the cloud – because the communication is network dependent.

In order to reuse available resources, Kristensen [10] proposed a system that distributes tasks between mobile devices (smartphones, tablets, and notebooks) composing a local MCC. Satyanarayanan et al. [14] proposed to ensure smartphones performance using virtual machines allocated on a specific server. Cuervo et al. [9] proposed an MCC that considers the communication type (3G or WiFi) and application processing features to reduce energy consumption. Other works proposed to use smartphones to execute distributed and shared tasks, as an approach that considered an MCC in the crowdsourcing context, but they used a specific server architectures [7, 23].

Demiris et al. [11] and Cook [8] highlighted resident concerns about identifying activities in a house, e.g. infer the resident's routine. Potts et al. [17] and Ramlee et al. proposed the appliances access only by Bluetooth, to limit the communication range and to provide security with a low-cost of deployment.

Chakravorty et al. [5] proposed a privacy scheme to the smart house sensors data, using an anonymization system at the first step and re-identification of the data at a second step. Witkovski et al. [6] presented an authentication method based on keys and Identity Management (IdM) to provide Single Sign-On (SSO) in IoT, managing securely the appliance's specific attributes. Although mentioned works allow secure access to appliances, they do not address the power management capability, neither the security and privacy provided by design to a smart house in the local context.

Several proposals aimed to the benefits the smart house can bring, but mentioned approaches present security and privacy weaknesses because the Internet intrusion is possible when there is a vulnerability in an appliance. In addition, the EPC interaction aiming at the rational use of energy, either to save energy or to reduce the cost to the residents, was not addressed by any work of literature. In a scenario where the demand for security and privacy solutions becomes increasingly required, it is necessary that security and privacy be designed in an architectural perspective [21].

## IV. Proposal

This section describes the proposed smart house security scheme that provides privacy as a by-product. The proposal implements a local MCC and considers the scenario of energy consumption management. The aiming is to deal with the requirements of electric power companies and residents, without exposing the house appliances to the Internet.

The proposal involves three main components (Figure 1): WSOutside, WSInside, and the MCC. For maintaining the isolation and integrity of smart meter, WSOutside and WSInside are services running on a smart meter peripheral device. WSOutside enables the communication between SCADA/MDM and the smart house, and maintains a request queue, for instance, a request from the electric power company (EPC) to reduce the consumption. WSInside is accessible only in the HAN, is responsible for storing MCC information and consumption profiles, and it reconfigures the consumption based on the profiles, in the case of MCC absence.

We use two distinct services (WSOutside and WSInside) to mitigate attacks from the Internet, ensuring security and privacy for the smart house. SCADA/MDM can only set requests to WSOutside. Periodically, WSInside reads the WSOutside request queue. There is no direct access from the Internet to the smart house. However, EPC can still interfere in the smart house consumption.

The proposed MCC is composed of mobile devices and has three types of members: resident, header, and visitor. An MCC has one header, several residents, and several visitors. A resident is represented by a mobile device that collects and processes appliances attributes, such as consumption and status. Header coordinates the MCC and consolidates residents' data, creating and updating smart house consumption profiles. A visitor is a mobile device held by a trusted visitor that only collects appliances consumption attributes, avoiding violating the privacy of house residents. Section B details the MCC management, including the header election and the trusting assignment process for visitors.

The proposed MCC implies several advantages from the security perspective. MCC does not have a single point of compromising because various mobile devices compose it, and its coordinator (header) is defined at random – hindering specify the target of an attack. Smart house data stay distributed and encrypted on the MCC components, improving security and privacy for the residents. The communication between mobiles occurs using a short-range wireless (e.g. Bluetooth and

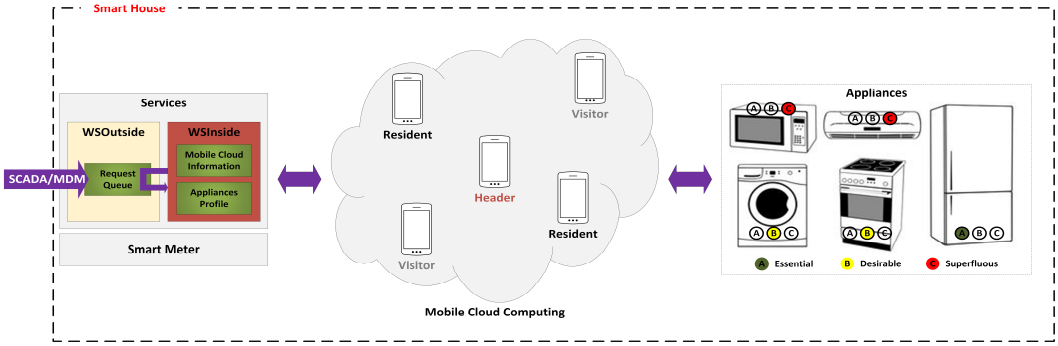


**Figure 1. Overview of the proposed smart house security scheme.**

WiFi), ensuring access only to devices physically nearby to the smart house. Furthermore, we assume that services running on the smart meter do no reach the appliances, i.e. appliances are accessible only by mobile devices.

MCC collects and processes appliances data periodically. Data are generic and specific appliances' attributes. This work addresses generic attributes (e.g. instantaneous and hourly consumption). Specific attributes were addressed by Witkovski et al. [6]. MCC also consolidates consumer energy profiles that are used to meet an energy consumption reconfiguration requirement. Thus, we have a decentralized mechanism to get and process data, using residents' mobile devices, without needing a specific server for managing the smart house.

The smart house data processing enables to determine electricity consumption patterns of each appliance during a day, week, month, and year. Consumption profiles can be created combining appliances consumption patterns and residents' needs. For instance, we present three profiles that classify appliances on essential, desirable, and superfluous (Figure 1). This classification will be used in the process of energy consumption reduction, discussed in section V.

MCC begins with the registration of a resident mobile device in the smart meter installation process. An EPC technician registers the mobile MAC address, the phone number, as well as a password to access the MCC. By this way, other smart house users can register other mobiles on the MCC.

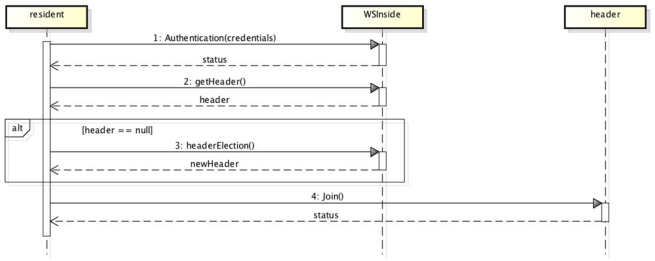


**Figure 2. Sequence diagram for a resident to join an MCC.**

Once a registered resident mobile reaches the HAN wireless coverage, the MCC application starts a process to join the MCC (Figure 2). The resident sends an authentication request to the WSInside (Event 1), providing its credentials that were set up in the register process. WSInside verifies the credentials and returns the authentication status. If credentials are valid, WSInside informs header information (Event 2), if header exists. If the header does not exist, resident requests the header election process to the WSInside (Event 3). MCC information is reported to the resident, which sent a message to join the MCC (Event 4).

The header election process occurs in the MCC starting and in the unavailability of the MCC header. A header can become frequently unavailable due to diverse reasons, such as it can leave the HAN area or be turned off. A new header is selected randomly, choosing one from the resident list and meaning that any resident can become an MCC header. The random choice has as main benefit the difficulty of predicting which will be the next header. Thus, a possible attacker cannot target a specific resident, aiming to explore a known vulnerability and control the MCC. It is worth emphasizing that visitors cannot become an MCC header.

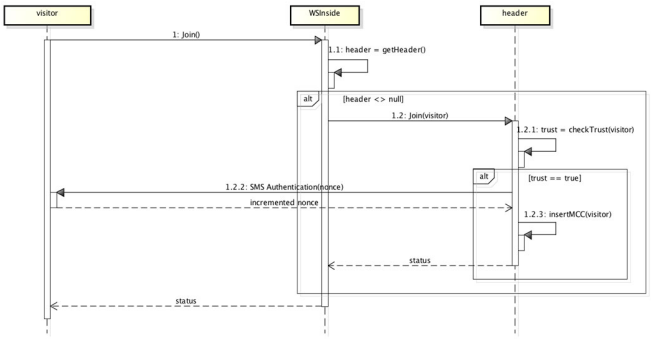


**Figure 3. Sequence diagram for a visitor to join an MCC.**

Each visitor participation on an MCC has a timeout (e.g. 24 hours). In the case of a visitor timeout is reached or it leaves the MCC, the trust validation process must be done again.

The process for a visitor to join an MCC is shown in Figure 3. Visitor requests to join the MCC to the WSInside (Event 1). If MCC does not exist, WSInside replies an error message, otherwise, if MCC exists, WSInside requests to the header to add the visitor (Event 1.2). Header evaluates the visitor trustworthiness and shows in the user GUI (Graphic User Interface) a trust level for the visitor (Event 1.2.1). The trust level of a visitor is get through the historical of messages and calls to any resident using a specified period (e.g. last 30 days).

In the case of the visitor is considered trusty by the MCC header, reaching a threshold (e.g. 3 occurrences), header sends an SMS (Event 1.2.2) with a nonce (random numbers) to the visitor. Visitor replies the header message incrementing the nonce by 1, to prevent replay attacks and to prove its authenticity, since he/she received the SMS message in his/her cell phone. After confirming the visitor identification, header registers the visitor in the MCC (Event 1.2.3).

## V. SMART GRID SCENARIO

Our proposal assumes that the EPC has a system (e.g. SCADA/MDM) integrated with smart houses and can get consolidated energy consumption from each smart house (HAN) and a region (NAN). Therefore, it is possible to obtain the instant total power consumption of a city (WAN). Monitoring the power consumption enables to compare with the power supply and predict a blackout, for instance.

A traditional way to avoid a blackout is to activate other power sources, however it is very expensive because requires a spare infrastructure (backup) to be used when a blackout was detected in advance, for example. Our proposal enables to monitor the power consumption, identifying that it is reaching critical values, and enables SCADA/MDM to require users to reduce energy consumption, avoiding the blackout. Therefore, instead of spending money on a backup infrastructure, it is possible to mitigate that situation and have an online smart house integration system, allowing deploy other facilities, e.g. fraud detection, power quality, and system monitoring.

Figure 4 represents a use case scenario considered in this proposal. SCADA/MDM is connected with several gateways (MDC - *Metering Data Collector*), representing NANs. A gateway mediates the connection between the SCADA/MDM and several HANs (smart houses). The gateway consolidates power consumption information in a NAN. Thus, SCADA/MDM does not directly communicate with the smart houses, increasing the granularity of the collected information and the privacy of residents. The gateway, however, is connected to the smart houses via WSOutside.

After SCADA/MDM predicts that available electricity is not enough to cover the demand, it starts the process of requesting the energy consumption reconfiguration. The process consists of defining the NAN that will be the target of reduction, and getting the address of the linked gateways. Next, SCADA/MDM informs the percentage of energy consumption reduction that each gateway must forward to each HAN.

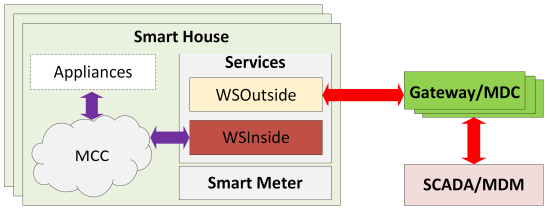


**Figure 4. Overview of smart houses and smart grid integration.**

When WSInside gets a request from EPC, WSInside checks if the header is reachable. Therefore, MCC reads the queue of requests from WSOutside (SCADA/MDM). WSInside accesses WSOutside to improve security and privacy protection. If there is any vulnerability in MCC, it cannot be exploited from the Internet because the access is not direct and the process is non-interactive, therefore avoiding data disclosure and improving privacy. Of course, we assume there are not apps in the mobiles performing reverse NAT (Network Address Translation), allowing someone from the Internet to access directly a mobile in MCC.

From an EPC request, MCC analyzes the consumption profiles and determines which appliances need to be reprogrammed or even turned off, considering the appliances categories. As an obvious policy, MCC prioritizes the reduction/shutdown of appliances considered superfluous and after desirable (e.g. in Figure 1).

If there is no MCC active in the smart house, the consumption profile stored in WSInside will be used to turn off an electric circuit with appliances in the group of superfluous first, and desirable after. Note that, in this case, does not occur interaction with MCC. So, the reprogramming or selective appliance shutdown, mediated by a mobile device, will not happen. Instead, the appliances in an electric circuit will be indiscriminately turned off.

When expires the time to execute an action, the gateway checks if the consumption reduction requests were achieved. This process consists of collecting instantaneous consumption of smart houses. If the requested reduction value has not been achieved, the gateway can act in a traditional way, cutting off the power of the smart houses. Nevertheless, such action depends on each EPC policy.

## VI. PROTOTYPE AND TEST

The prototype implementation involves the WSInside and MCC. WSInside is a web service implemented in Java (www.docs.oracle.com/javase/8), using the JAX-RS API (www.jaxrs-spec.java.net). MCC are integrated Android mobile applications. To mimic the behavior of appliances, we developed a Java application that responses consumer requests performed by MCC mobiles.

The applications that compose the MCC were developed using the Android Jelly Beans (developer.android.com/about/versions/), which is required by NSDiscovery (www.developer.android.com/reference/android/net/nsd/). NSDiscovery applies broadcast for services announcement and discovery on a LAN. MCC announces collecting and processing services information using NSDiscovery in the appliances.

The MCC header node does not announce the service, just discovers mobiles, which wishes to join the MCC through NSDiscovery. Thus, it is not possible to find out who is the header node in an MCC. Only mobiles that are already active in the MCC have the knowledge of who is the header (Figure 2, event 2). An MCC mobile that is registered in WSInside is considered resident. The registration process stores the phone number, IMEI (if exists), MAC address, and a password – defined by each mobile user. This information is required and validated at every joining of a resident in the MCC.

When a mobile request to the WSInside to join the MCC and it is not registered, it is considered visitor and WSInside forwards the request (with the phone number) to the header. If the visitor is considered trusted by the trust evaluation process, the header sends a validation SMS message (Figure 3, event 1.2.2), using the phone number informed by the visitor, through the Android telephony library. Every application installed on the mobile has an SQLite database, responsible for storing the data collected from the appliances that are under its responsibility and within a reachable range.

The application processes the data collected from appliances and synchronizes the information with the residents in the MCC. If a mobile becomes unavailable, the information collected and processed is not lost. After the collection, synchronization, and processing, the header updates the consumer's profile in WSInside with the consolidated

information. Such information is based on a calculus of the energy consumption of each appliance category. The profile update time is set to 5 minutes, for instance, to provide accurate consumption when EPC requests and MCC is not active.

The MCC participant's applications exchange IAA (I Am Alive) messages with each other to find out who is active in the MCC, and especially to see if the header is active. When the application detects that the header is inactive, WSInside is invoked to start a header election (Section IV.B).

### *A. Privacy and Security Evaluation*

To evaluate the security and privacy, we considered a scenario where there was a vulnerability in one of the MCC mobiles (visitor or resident). The vulnerable mobile is the target of attacks aiming to remotely exploit the vulnerability. We compared our proposal with two other approaches (Figure 5).

Scenario A shows a general approach, commonly found in commercial solutions (www.carriots.com). In such a case, an attacker (user) has direct access to the mobile cloud from the Internet (Figure 5, event i). After successfully exploit a vulnerability, she can remotely access and handled it.

Scenario B represents the typical literature approach [22, 6], which is more robust in terms of security than the previous one. Scenario B relies on an Identity Management System (IdM), which is responsible for authenticating the user and granting access authorization (Figure 5, event i), but gateway parses requests from the Internet to the MCC (Figure 5, events ii and v). MCC is not directly exposed but is still accessible for attackers to exploit vulnerabilities, because, after the authentication and access authorization procedures, the attacker has direct access to the mobile.

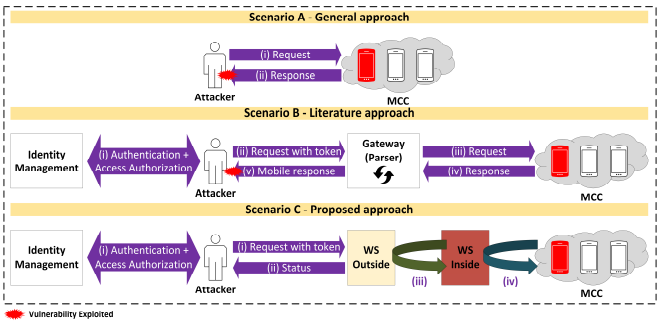


**Figure 5. Evaluation scenarios.**

Finally, scenario C represents the proposed approach, in which is not possible for the attacker to exploit a vulnerability. The reason is there is no request parser on scenario C, different from scenario B. In other words, the attacker only put a request on the WSOutside request-queue and must close the connection, after receiving a request ack (Figure 5, event i and ii). As explained in Section 4, WSInside searches for requests in the WSOutside request-queue (Figure 5, event iii), gets and processes it, putting the response back in the WSOutside request-queue. Thus, as the attacker cannot directly access the mobile, she is unable to handle the vulnerable mobile, providing privacy and security.

To evaluate the scenarios A, B, and C, we exploited the CVE-2013-4710 vulnerability present in Android 4.1 and previous. This vulnerability involves the Android Webview, the component responsible for accessing web pages. This vulnerability allows the attacker access remotely the mobile, obtaining private information and access the MCC.

To exploit this vulnerability, we used the smartphone Samsung S3 mini running Android 4.1. The attacker ran Kali operating system and used the Metasploit framework (www.metasploit.com). We created a malicious application that requests a URL available in the Metasploit. The application accesses a malicious URL that exploits the Android vulnerability and allows the attacker to access the mobile's command line. Figure 6 exhibits obtained results in scenarios A and B.


**Figure 6. Exploited vulnerability.**

The attacker displays his/her address on the network (*ifconfig* command) and then provides the malicious page on the network (exploit). Next, the app on the victim's device accesses the URL, and at this moment, the session is broken. Therefore, the attacker has access to the victim's command line, exhibiting the machine address again (by *ifconfig*) and showing the attack was successful.

In scenario C (Figure 5, event iii) the attack failed, because even with a vulnerable victim, there is not session between the client and the attacker. So, as mentioned above, there is no one who can take advantage of it.

### *B. Performance Evaluation*

In order to execute a performance evaluation of the prototype, a scenario with four (4) mobiles were built and their configurations are shown in Table 1.

**Table 1. Configuration of used mobiles.**

| *Mobile* | *Processor (GHz)* | *RAM (GB)* | *Android* |
|---|---|---|---|
| Motorola G2 | Quad-core 1.2 | 1.0 | 5.02 |
| Samsung S3 mini | Dual-core 1.2 | 1.0 | 4.1 |
| Asus ZenFone 5 | Dual-core 1.2 | 2.0 | 4.3 |
| Nexus 7 | Quad-core 1.9 | 1.0 | 4.4.2 |

The tests aim to evaluate the MCC behavior and the overhead increasing the number of mobiles. In these tests, we developed an environment where the mobiles collect, process, synchronize, and store the appliances consumption data. The same header (Motorola G2) was used on all experiments, aiming do not interfere in the measurements due to different hardware configurations.

The smart house has 12 appliances, which provides 10 attributes for each consumption request. Therefore, we added other mobiles to evaluate the impact on the MCC performance (Figure 7). As expected, by increasing the number of mobiles in the MCC, the performance is proportionally augmented due to the distribution of the collection, processing and storing tasks.

The second test checks the overhead for the number of participants in the MCC, regarding the collected data synchronization. In such a case, it was not considered the collection and processing time but only evaluated the information synchronization time, considering the header as a reference. Similarly, to the previous test, the header yielded same throughout. The average synchronization time for the attributes when using two mobiles was 12.29 ms. When using three mobiles in the MCC, the spent time was 14.3 ms, and for the four mobiles were 15.86 ms. Thus, it is possible to observe that there is a mobiles synchronization overhead from 10% to 15% for each mobile added in the MCC.

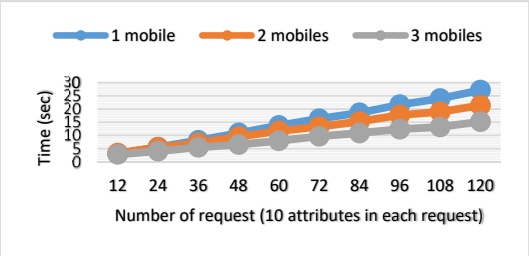


**Figure 7. Evaluation of the number of mobiles in MCC.**

## VII. CONCLUSION

We presented a mechanism of energy consumption management for smart houses with security and privacy conceived by design. Unlike the literature, our approach makes the house accessible from the Internet without exposing the appliances. Moreover, we use ad hoc Mobile Cloud Computing (MCC), a decentralized mechanism that takes advantage of the computing power of the mobiles of each resident, without requiring additional costs to buy a dedicated equipment.

Our proposal has an architecture designed to avoid a single point of failure and attack. Changes in the consumer profiles and the MCC member registration are possible only if the mobile is reachable in the smart house LAN; this prevents attacks and privacy violations that come from the Internet. This security and privacy level is possible because the MCC is ad hoc formed and is not possible direct access to appliances from the Internet. The proposed scheme isolates the MCC from the Internet, making it accessible only inside the smart house, as shown in the security evaluation scenarios.

MCC responds to the EPC requests through a service that mediates communication. Thus, when the EPC needs an energy consumption reduction, e.g. to avoid a possible blackout, it communicates with a service that works exclusively to receive external requests. For security reasons and privacy protection, another service detects the EPC requests to start a process of energy consumption reduction in a smart house.

Our prototype isolated the appliances and MCC from the Internet and presented a controlled overhead when increased the number of MCC mobiles. Vulnerabilities in the MCC mobiles could not be directly explored. Increasing the number of mobiles in the MCC, proportionally increases its performance, although there is a controlled overhead in the mobiles management. Regarding the synchronization of collected data, the prototype presented an overhead between 10% and 15% for each mobile added.

## ACKNOWLEDGMENTS

This work was partially sponsored by the Brazilian National Council for Scientific and Technological Development (CNPq), grants 310671/2012-4 and 404963/2013-7. R. Ribeiro, E. Viegas and V. Abreu wishes to thanks CNPq and J. E. Marynowski the Coordination for the Improvement of Higher Level Personnel (CAPES) for the scholarships granting.

## REFERENCES

[1] A. R. Al-Ali and M. AL-Rousan, “Java-based home automation system,” IEEE Trans. Consum. Electron., vol. 50, no. 2, pp. 498–504, 2004.

[2] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, “A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges,” IEEE Com. Surv. Tutorials, vol. 15, no. 1, pp. 5–20, 2013.

[3] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” Comput. Networks, vol. 54, no. 15, pp. 2787–2805, 2010.

[4] Proofpoint. “Proofpoint Uncovers Internet of Things (IoT) Cyberattack.” http://investors.proofpoint.com/releasedetail.cfm?ReleaseID=819799.

[5] A. Chakravorty, T. Wlodarczyk, and Chunming Rong, “Privacy Preserving Data Analytics for Smart Homes,” in IEEE Security and Privacy Workshops, pp. 23–27, 2013.

[6] A. Witkovski, A. Santin, V. Abreu, and J. Marynowski, “An IdM and Key-based Authentication Method for providing Single Sign-On in IoT.”, in IEEE Global Communication Conf. (GLOBECOM), pp. 1-6, 2015.

[7] G. Chatzimilioudis, A. Konstantinidis, C. Laoudias, and D. Zeinalipour-Yazti, “Crowdsourcing with Smartphones,” IEEE Internet Comput., vol. 16, no. 5, pp. 36–44, 2012.

[8] D. J. Cook, “How Smart Is Your Home?,” Science, vol. 335, no. 6076, pp. 1579–1581, 2012.

[9] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, “MAUI: Making Smartphones Last Longer with Code Offload” in MobiSys, pp. 49-62, 2010,

[10] M. Daro Kristensen, “Scavenger: Transparent development of efficient cyber foraging applications,” in IEEE PerCom, pp. 217–226, 2010.

[11] G. Demiris, B. K. Hensel, M. Skubic, and M. Rantz, “Senior residents’ perceived need of and preferences for ‘smart home’ sensor technologies.,” Int. J. Technol. Assess. Health Care, vol. 24, no. 1, pp. 120–124, 2008.

[12] A. Rajabzadeh, A. Manashty, and Z. Jahromi, “A Mobile Application for Smart House Remote Control System,” in ICWCMC - Int. Conf. Wirel. Commun. Mob. Comput., pp. 80–86, 2010.

[13] R. A. Ramlee, M. A. Othman, M. H. Leong, M. M. Ismail, and S. S. S. Ranjit, “Smart home system using android application,” in Intl. Conf. of Information and Communication Technology, pp. 277–280 , 2013

[14] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The Case for VM-Based Cloudlets in Mobile Computing,” IEEE Pervasive Comput., vol. 8, no. 4, pp. 14–23, 2009.

[15] M. G. Golzar and H. Tajozzakerin, “A New Intelligent Remote Control System for Home Automation and Reduce Energy Consumption,” in Asia Int. Conf. on Mathematical/Analytical Modelling and Computer Simulation, pp. 174–180, 2010.

[16] Haowen Chan and A. Perrig, “Security and privacy in sensor networks,” Computer (Long. Beach. Calif)., vol. 36, no. 10, pp. 103–105, 2003.

[17] J. Potts and S. Sukittanon, “Exploiting Bluetooth on Android mobile devices for home security application,” In IEEE Southeastcon, pp. 1–4. 2012.

[18] S. C. Patel and P. Sanyal, “Securing SCADA systems,” Inf. Manag. Comput. Secur., vol. 16, no. 4, pp. 398–414, 2008.

[19] A. N. Khan, M. L. Mat Kiah, S. U. Khan, and S. a. Madani, “Towards secure mobile cloud computing: A survey,” Futur. Gener. Comput. Syst., vol. 29, no. 5, pp. 1278–1299, 2013.

[20] K. Y. Lian, S. J. Hsiao, and W. T. Sung, “Intelligent multi-sensor control system based on innovative technology integration via ZigBee and Wi-Fi networks,” J. Netw. Comput. Appl., vol. 36, no. 2, pp. 756–767, 2013.

[21] H. Khurana, M. Hadley, Ning Lu, and D. A. Frincke, “Smart-grid security issues,” IEEE Secur. Priv. Mag., vol. 8, no. 1, pp. 81–85, 2010.

[22] I. Korkmaz, S. K. Metin, A. Gurek, C. Gur, C. Gurakin, and M. Akdeniz, “A cloud based and Android supported scalable home automation system,” Comput. Electr. Eng., vol. 43, pp. 112–128, 2015.

[23] V. Della Mea, E. Maddalena, and S. Mizzaro, “Crowdsourcing to Mobile Users: A Study of the Role of Platforms and Tasks,” in DBCrowd-VLDB wks on DBs and Crowdsourcing, pp. 14–19, 2013.

[24] C. Perera, P. P. Jayaraman, A. Zaslavsky, D. Georgakopoulos, and P. Christen, “Sensor discovery and configuration framework for the Internet of Things paradigm,” in IEEE World Forum Internet Things, pp. 94–99, 2014.