



HAL
open science

SECURITY OF MOBILE CLOUD APPLICATIONS

Daniela Popa

► **To cite this version:**

Daniela Popa. SECURITY OF MOBILE CLOUD APPLICATIONS. Computer Science [cs]. Université de Cluj-Napoca (TUCN), 2013. English. NNT: . tel-01349129

HAL Id: tel-01349129

<https://hal.science/tel-01349129>

Submitted on 5 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECȚIEI SOCIALE ȘI
PERSOANELOR VÂRSTNICE
AMPOSDRU



Fondul Social European
POS DRU 2007-2013



Instrumente Structurale
2007-2013



MINISTERUL
EDUCAȚIEI
NAȚIONALE
OIPOSDRU



Investește în oameni !

FONDUL SOCIAL EUROPEAN

Proiect cofinanțat din Fondul Social European prin Programul Operațional Sectorial pentru Dezvoltarea Resurselor Umane 2007 – 2013

Axa prioritară 1: „Educația și formarea profesională în sprijinul creșterii economice și dezvoltării societății bazate pe cunoaștere”

Domeniul major de intervenție 1.5 "Programe doctorale și post-doctorale în sprijinul cercetării"

Titlul proiectului: „Q-DOC- Creșterea calității studiilor doctorale în științe inginerești pentru sprijinirea dezvoltării societății bazate pe cunoaștere”

Contract : POSDRU/107/1.5/S/78534

Beneficiar: Universitatea Tehnică din Cluj-Napoca

FACULTATEA DE ELECTRONICĂ, TELECOMUNICAȚII ȘI TEHNOLOGIA INFORMAȚIEI

Ing. Daniela POPA

TEZĂ DE DOCTORAT

SECURITATEA APLICAȚIILOR MOBILE CLOUD

SECURITY OF MOBILE CLOUD APPLICATIONS

Conducător științific,

Prof.dr.ing. Monica BORDA

Comisia de evaluare a tezei de doctorat:

PREȘEDINTE: - Prof.dr.ing. *Virgil Dobrotă* - Departamentul de Comunicații, Facultatea de Electronică, Telecomunicații și Tehnologia Informației, Universitatea Tehnică din Cluj-Napoca

MEMBRI: - Prof.dr.ing. *Monica Borda* - conducător științific, Universitatea Tehnică din Cluj-Napoca;
- Prof.dr.ing. *Ioan Naforniță* - referent, Universitatea „Politehnica” din Timișoara;
- Prof.dr.ing. *Alexandru Isar* - referent, Universitatea „Politehnica” din Timișoara;
- Conf.dr.ing. *Marcel Cremene* - referent, Universitatea Tehnică din Cluj-Napoca;
- Conf.dr.ing. *Karima Boudaoud* - referent, Universitatea Nice Sophia Antipolis, Franța.

Data Susținerii: 06-12-2013.



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI ȘI
PROTECȚIEI SOCIALE
AMPOSDRU



Fondul Social European
POS DRU 2007-2013



Instrumente Structurale
2007-2013



MINISTERUL
EDUCAȚIEI
CERCETĂRII
TINERETULUI
ȘI SPORTULUI
OIPOSDRU



Investește în oameni !

FONDUL SOCIAL EUROPEAN

Proiect cofinanțat din Fondul Social European prin Programul Operațional Sectorial pentru Dezvoltarea Resurselor Umane 2007 – 2013

Axa prioritară 1: „Educația și formarea profesională în sprijinul creșterii economice și dezvoltării societății bazate pe cunoaștere”

Domeniul major de intervenție 1.5 "Programe doctorale și post-doctorale în sprijinul cercetării"

Titlul proiectului: „Q-DOC- Creșterea calității studiilor doctorale în științe inginerești pentru sprijinirea dezvoltării societății bazate pe cunoaștere”

Contract : POSDRU/107/1.5/S/78534

Beneficiar: Universitatea Tehnică din Cluj-Napoca

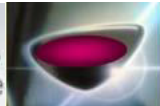
PhD Thesis

Security of Mobile Cloud Applications

Technical University of Cluj-Napoca

Eng. Daniela POPA

**PhD Advisor:
Prof.Dr.Eng. Monica BORDA**



Joint Guidance PhD Thesis

Security of Mobile Cloud Applications

Eng. Daniela POPA

Advisor:

Prof.Dr.Eng. Monica BORDA

*Data Processing and Security Research Centre of Technical
University of Cluj-Napoca (TUCN)*

Co-Advisors:

Conf.Dr.Eng. Karima BOUDAUD

*Laboratoire d'Informatique, Signaux et Systemes de Sophia-
Antipolis (I3S - UNSA)*

Conf.Dr.Eng. Marcel CREMENE

Adaptive Systems Laboratory (TUCN)

Acknowledgements

I would like to start by extending my sincerest thanks and gratitude to my thesis adviser Prof. Monica Borda. Only because of her constant support and encouragements this thesis has now a final form.

My gratitude and thanks also go to Prof. Karima Boudaoud and to Prof. Marcel Cremene, for being co-advisers to this thesis; for their continuous support, for their innovative ideas and for their guidance.

My appreciation and thanks also go to my colleagues and professors from the Data Processing and Security Research Centre for their professional advices, their friendship and kindness.

I would like to extend my sincerest thanks and gratitude to all my friends and family for believing in me, for their constant encouragements and for being always by my side.

I would like to conclude by adding that this thesis would not have been possible without the funding received from the project "Improvement of the doctoral studies quality in engineering science for development of the knowledge based society-QDOC" contract no. POSDRU/107/1.5/S/78534, project co-funded by the European Social Fund through the Sectorial Operational Program Human Resources 2007-2013.

Abstract

Mobile Cloud Computing is a new concept, which offers Cloud Computing resources and services for mobile devices.

Cloud Computing is a new technology that provides, to the Internet users, data, resources, platforms, and applications as services.

In the last years mobile phones have greatly developed, and because of the small size and also because they can be moved easily, they become indispensable to users.

Using Mobile Cloud Computing advantages, new application models were developed for mobile devices. These applications models try to use both the mobile device resources and the Cloud services to provide a more reach and a more varied functionality in order to increase the mobile device popularity and use.

From a security point of view, Mobile Cloud Computing, increases the security risks and privacy invasion due to the fact that it combines mobile devices with cloud services and also because there is not a well-defined application model.

The security issues are treated independently and the existing security solutions are supplied separately by various providers (e.g. cloud services providers or mobile platforms providers). Thereby, in the case of mobile cloud applications, it is needed to combine different solutions in order to secure them. Also, very few solution proposed to secure the data or the applications take into account the mobile device energy constraints, users constraints or data sensitivity constraints.

Rezumat

Mobile Cloud Computing este un concept nou, ce pune la dispoziția telefoanelor mobile resurse și servicii oferite de Cloud Computing.

Cloud Computing este o tehnologie nouă, ce furnizează date, resurse, platforme și aplicații sub formă de servicii, fiind dezvoltată pentru utilizatorii de Internet.

În ultimii ani telefoanele mobile, s-au dezvoltat foarte mult; și datorită dimensiunilor reduse și a ușurinței cu care pot fi deplasate au devenit indispensabile pentru utilizatori.

Folosind avantajele oferite de Mobile Cloud Computing s-au dezvoltat, pentru telefoanele mobile, diferite noi modele de aplicații. Aceste modele încearcă să se folosească de resursele telefonului și de serviciile din Cloud pentru a furniza o funcționalitate variată care să ducă la o creștere mai mare a popularității și a utilizării telefoanelor mobile.

Din punctul de vedere al securității, Mobile Cloud Computing, crește riscurile de securitate și de privacy, deoarece se combină dispozitivele mobile cu servicii Cloud și, de asemenea, pentru că nu există încă un model bine definit de aplicație.

Problemele de securitate sunt tratate independent și soluțiile de securitate existente sunt furnizate separat de către diverși furnizori (furnizori de servicii Cloud sau furnizori de platforme mobile). Astfel, în cazul aplicațiilor de tip mobil-cloud, este necesar să se combine soluții diferite, în scopul de a le asigura securitatea. De asemenea, foarte puține dintre soluțiile propuse pentru a asigura securitatea datelor sau a aplicațiile iau în considerare constrângerile energetice ale telefoanelor mobile, dorințele utilizatorilor sau sensibilitatea datelor.

Table of Contents

List of Figures	xv
List of Tables.....	xix
List of Abbreviations.....	xx
1. Introduction.....	1
1.1 Motivation and Objectives of the Thesis.....	1
1.2 Thesis Structure	4
2. Cloud Computing	7
2.1 What is Cloud Computing?	8
2.1.1 Various Definitions.....	8
2.1.2 Cloud Service Models	11
2.1.3 Cloud Deployment Models.....	14
2.1.4 Cloud Characteristics	16
2.1.5 Technologies Behind	17
2.2 Cloud Benefits.....	19
2.3 Security Challenges.....	21
2.4 Conclusions	26
2.4.1 Contributions.....	26
2.4.2 Publications	26
3. Mobile Cloud Computing.....	27
3.1 What is Mobile Cloud Computing?.....	28
3.1.1 Definitions.....	29

3.1.2 Mobile Cloud Applications Models.....	31
3.1.2.1 Client Model Class.....	31
3.1.2.2 Client/Cloud Model Class.....	32
3.1.2.3 Cloud Model Class.....	34
3.2 Mobile Cloud Computing – Security.....	34
3.2.1 Security Issues of Mobile Cloud Computing.....	35
3.2.1.1 Mobile Threats.....	35
3.2.1.2 Cloud Threats.....	36
3.2.1.3 Technological Threats.....	37
3.2.1.4 Mashup Security Issues.....	38
3.2.2 Existing Mobile Cloud Security Solutions.....	39
3.2.2.1 Mobile Security Solutions.....	39
3.2.2.2 Mobile Cloud Communications Security Solutions.....	41
3.2.2.3 Mobile Cloud Security Solutions.....	41
3.3 Conclusions.....	46
3.3.1 Contributions.....	47
3.3.2 Publications.....	47
4. Proposed Secure Mobile-Cloud Framework.....	49
4.1 Criteria and Objectives.....	50
4.1.1 Criteria.....	51
4.1.1.1 Human constraints.....	51
4.1.1.2 Data sensitivity constraints.....	51
4.1.1.3 Technical constraints.....	52
4.1.2 Objectives.....	54

4.2 The Framework Basis.....	55
4.2.1 The Integrity Component	56
4.2.2 The Authenticity Component	56
4.2.3 The Confidentiality Component.....	57
4.2.4 The Non-Repudiation Component	58
4.3 The Framework Design.....	58
4.3.1 The Security Managers.....	60
4.3.2 The Auxiliary Managers	62
4.3.3 Mobile Manager	64
4.4 Conclusions	71
4.4.1 Contributions.....	71
4.4.2 Publications	72
5. Secure Mobile-Cloud Framework Implementation.....	73
5.1 The Security Managers	74
5.2 The Auxiliary Managers.....	79
5.3 The Mobile Manager	82
5.4 Databases Implementation.....	85
5.5 The User Interface	90
5.6 Unit Tests.....	93
5.7 The Software Tools Used.....	95
5.7.1 The Android Environment.....	95
5.7.2 Eclipse development environment	96
5.8 Conclusions	97
5.8.1 Contributions.....	97

5.8.2 Publications	98
6. The integration of Secure Mobile-Cloud Framework with a mobile cloud application.....	99
6.1 Mobile-Cloud Applications Examples	100
6.2 Mobile Cloud Application Scenario	102
6.2.1 Description.....	102
6.2.2 Development	103
6.3 Secure Mobile-Cloud Framework integration with the application scenario	107
6.3.1 Theoretical Approach.....	108
6.3.2 Technical Approach	113
6.4 Conclusions	118
6.4.1 Contributions.....	118
6.4.2 Publications	119
7. Overall Conclusions	121
7.1 Contributions Overview.....	122
7.2 Future Works	123
Bibliography	125
List of Publications	137
Appendix.....	139

List of Figures

Figure 2.1 NIST Definition of Cloud Computing [NIST]	11
Figure 2.2 Infrastructure as a Service	12
Figure 2.3 Platform as a Service	12
Figure 2.4 Software as a Service.....	13
Figure 2.5 Hybrid Cloud	13
Figure 2.6 Deployment Models - Public, Private and Community.....	14
Figure 2.7 Jericho's Model [JERICHO].....	15
Figure 2.8 Technologies behind Cloud Computing.....	19
Figure 2.9 Cloud Computing benefits.....	20
Figure 2.10 Reasons for enterprises not to adopt Cloud Computing [Gens09]	22
Figure 2.11 Terms of concerns.....	24
Figure 3.1 Mobile Cloud Computing	29
Figure 3.2 The Client Model	32
Figure 3.3 The Client-Cloud Model.....	33
Figure 3.4 The Cloud Model.....	34
Figure 4.1 Component-based applications.....	50
Figure 4.2 Different security levels.....	52
Figure 4.3 Integrity Component [KBR+11]	56
Figure 4.4 Authenticity Component [KBR+11].....	57
Figure 4.5 Confidentiality Component [KBR+11]	57
Figure 4.6 Non-Repudiation Component[KBR+11].....	58

Figure 4.7 The Secure Mobile-Cloud Framework	60
Figure 4.8 The Security Parts.....	61
Figure 4.9 The Steps to secure data	62
Figure 4.10 The Auxiliary Managers.....	62
Figure 4.11 Communication Policy Manager and Mobile/Cloud Security Manager	63
Figure 4.12 The communication between State Manager and Mobile Manager	63
Figure 4.13 The Mobile Manager	64
Figure 4.14 Security Framework – answers needed	65
Figure 4.15 Capture User Choices example	69
Figure 4.16 The Security Components Execution Part	69
Figure 5.1 Mobile Security Manager – Class Diagram	75
Figure 5.2 Auxiliary Managers – Class Diagram.....	80
Figure 5.3. Eight character string (basic code decoding).....	80
Figure 5.4 Operating example for discoverComponent() method	81
Figure 5.5 Twenty two character string (advanced code decoding)	81
Figure 5.6 Operating example for discoverAdvancedComponent() method	81
Figure 5.7 Mobile Manager Class Diagram.....	83
Figure 5.8 Admin Database	86
Figure 5.9 Applications Database	87
Figure 5.10 The User Level Table.....	88
Figure 5.11 Classes Diagram for Database Implementation.....	89
Figure 5.12 Class diagram for the Admin database.....	89
Figure 5.13 XML coding.....	90
Figure 5.14 Java coding	90

Figure 5.15 UI functionality diagram.....	91
Figure 5.16 User profile set.....	91
Figure 5.17 Security options set (part a)	92
Figure 5.18 Security options set (part b)	92
Figure 5.19 Results - Unite test first scenario	93
Figure 5.20 Results - Unite test second scenario (a)	94
Figure 5.21 Results - Unite test second scenario (b)	94
Figure 6.1 Application Scenario functionality	103
Figure 6.2 Application Scenario Components	104
Figure 6.3 Log in action.....	105
Figure 6.4 Register action.....	105
Figure 6.5 Insert Data action.....	105
Figure 6.6 User Type action.....	106
Figure 6.7 Regime Type action.....	106

List of Tables

Table 2.1 Characteristics of Cloud Computing.....	17
Table 3.1 Key characteristics of Mobile Cloud Computing	30
Table 3.2 Mobile security solutions [LMS11]	40
Table 3.3 Overview of the security features provided by data security solutions	44
Table 4.1 Example of Security Combinations	59
Table 4.2 The Algorithms.....	68
Table 4.3 The Components Combinations	68
Table 6.1 Mobile commerce applications [SGG+11]	100

List of Abbreviations

AJAX - Asynchronous JavaScript and XML
CPU - Central Processor Unit
CSA - Cloud Security Alliance
CSS - Cascading Style Sheets
DOM - Document Object Model
EF - Encrypted File
EK - Encryption Key
FN - File Name
FS - File Size
HTML - Hyper Text Markup Language
IaaS - Infrastructure as a Service
IK - Integrity Key
IT - Information Technology
JSON - JavaScript Object Notation
LECCSAM - Low-Energy Consuming and User-centric Security Management
Architecture Adapted to Mobile Environments
MAC - Message Authentication Code
NIST - National Institute of Standards and Technology
OS - Operating System
PaaS - Platform as a Service
PWD - password
RAM - Random Access Memory
SaaS - Software as a Service
SMC - Secure Mobile-Cloud
SOA - Service-Oriented Architecture
XML - Extensible Markup Language

An investigator starts research in a new field with faith, a foggy idea, and a few wild experiments. Eventually the interplay of negative and positive results guides the work. By the time the research is completed, he or she knows how it should have been started and conducted.

Donald Cram

1. Introduction

Contents in Brief

1.1	Motivation and objectives of the Thesis	1
1.2	Thesis Structure	4

1.1 Motivation and Objectives of the Thesis

Back in 1961 the American computer scientist John McCarty stated that one day the computation will be organized and sold as a public utility as there are the water or the electricity. This idea of computation utility was very popular in the 60's [BYV08], but the hardware, the software and the telecommunication technologies were not ready yet to bring it to life. Only in 2007 the technology reached a development level that enabled the resumption and the implementation of this innovative idea.

Every creative idea needs a memorable name, so this new but also old idea was named Cloud Computing; and since 2007 till present Cloud Computing became a very fast-growing segment of the information technology industry. [STL10]

Security of Mobile Cloud Applications

Several characteristics have highlighted Cloud Computing and made it to be so famous. One of the characteristics is the fact that Cloud Computing supports the idea of resources separation. It suggests detaching the applications from the operating systems and from the hardware. Thus, before the Cloud Computing, if a company needed a software application, this meant that the company needed: the hardware (with the CPU, RAM and power supply), the operating system and then the software application. However, if something happened with the operating system or with the hardware (for example a power failure) then the application could not continue working.

Another characteristic is the fact that Cloud Computing focuses on fulfilling the needs of an Internet user (company, person, etc.). These needs can be various and may change over time or according to the user type. The Internet users can enjoy new applications with rich functionalities without having to deal with the applications management (e.g. install or upgrade). Thanks to the Cloud, the applications management is done automatically in the Cloud. The organizations may choose to not own and manage the IT services and use the Cloud services to support their IT requirements. In this way, they can reduce the cost of hardware maintenance and human resources.

Furthermore, Cloud Computing makes available its resources to various devices.

These characteristics of the Cloud Computing have triggered several benefits to the Internet users. One of the most frequently mentioned benefit of Cloud Computing is that it provides cost reduction. Cloud Computing lowers the cost for using IT services and the cost of everything that has to do with IT maintenance. Another benefit rises from the fact that Cloud Computing makes available its services to a wide range of devices. Thus it provides its resources to devices with various types of constraints (e.g. computational constraints), as a solution to their flaws.

One of the greatest opportunities that every person wants to enjoy is 'mobility'. Furthermore, each person has a small amount of curiosity, supplemented by a strong need for communication and knowledge. The mobile devices seem to be the devices that are able to link the mobility property with human emotional needs and information technology. All this is done using the Internet.

In order to capture people's attention towards mobile devices, powerful applications were developed for these devices. The applications allow mobile users to perform tasks like: managing personal health, games, editing, making reservations and paying tickets. As it is generally known, mobile devices are characterized by lack of resources. Thus, in order to run this new kind of applications, mobile hardware and network have known several improvements; but it wasn't enough. Thereby, Cloud

Security of Mobile Cloud Applications

Computing offers a solution to the mobile device challenges; and this solution is called Mobile Cloud Computing.

Mobile Cloud Computing is a novel concept, emerged immediately after Cloud Computing was made known. As its name implies, Mobile Cloud Computing, is the combination between mobile devices (Mobile) and Cloud Computing; it can be described as the availability of Cloud Computing resources to mobile devices.

Considering that Mobile Cloud Computing is a novel concept, which emerged from another novel concept, Cloud Computing, one of the objectives of this thesis was to do a state of the art on Cloud Computing. The purpose of this state of the art was to lay the knowledge foundation on: 1) what is Cloud Computing, 2) what are the benefits it brings, and 3) what are the major security concerns it raise. Also, the purpose of the state of the art on Cloud Computing was to provide a better understanding on what Mobile Cloud Computing is.

Another objective of this thesis was to do a state of the art on Mobile Cloud Computing and on the mobile cloud applications and to find out which are the security issues and the existing solutions for this kind of applications. Also the state of the art had to point out whether the existing solutions are enough to solve or cover the existing security issues. It was found that from a security point of view, Mobile Cloud Computing introduces many security issues due to the fact that it combines mobile devices with Cloud services and also because there were developed several new application models. The security issues were grouped in four categories: mobile threats, cloud threats, mashup threats and technological threats. In order to solve the highlighted security issues both, mobile platforms and Cloud providers, offered various solutions.

The mobile platforms have implemented five types of security strategies: traditional access control, application provenance, encryption, isolation and permission-based access control. These strategies may be used to protect the data and the applications on the mobile device but they do not resolve the security problems that arise while data are sent in Cloud to be stored or to be computed.

In order to protect data and applications, in the Cloud, several solutions were proposed. But some of the solutions are specific to a particular type of applications models while others do not take into account the lack of resources of the mobile devices. Furthermore, none of the existing solutions for mobile cloud applications do not allow the users to express their opinion regarding the security level to be applied to their private data.

Security of Mobile Cloud Applications

Some of the security solutions proposed to solve the security issues of data communication between mobile device and the Cloud do not take into account the mobile devices energy or performance constraints, or the users' opinions regarding the security level applied to their data. While, the security solutions that take into account these constraints are not adapted to mobile cloud applications.

This thesis focuses on mobile cloud applications which consist of components running on the mobile device or in Cloud. This thesis approaches the problem of data communication between the same application components. The proposed security solution has to take into account several constraints like: mobile device energy or users choices on the security level applied to their private data. Furthermore, data sensitivity constraint should be considered when providing the security solution. Data sensitivity constraint can be explained by the fact that some private data requires special care and handling, especially when their lost may cause harm to the users.

Another point approached in this thesis is the integration of the security solution into an existing mobile cloud application. The integration is static, made at the source code level by the mobile cloud application developer.

1.2 Thesis Structure

This thesis is organized in seven chapters. The first chapter is an introduction to the thesis subject and objectives. The second and third chapters present each a state of the art. The next three chapters present the thesis contributions: design, implementation and integration of the proposed security solution. In the final chapter there are presented some conclusions along with an overview of the thesis contributions and the future works.

Chapter 2 presents the state of the art on the Cloud Computing. The state of the art includes Cloud Computing various definitions, benefits and security concerns.

Chapter 3 presents the state of the art on the Mobile Cloud Computing. The state of the art includes Mobile Cloud Computing definition, mobile cloud applications models, the main security threats targeting mobile cloud applications and the approaches proposed to address these issues.

Chapter 4 presents the main contribution of this thesis, the design of the Secure Mobile-Cloud Framework. The goal of the framework is to secure the data used by a

Security of Mobile Cloud Applications

mobile cloud application. The framework features are: 1) it allows applying various security properties to different kinds of data (according to the data sensibility level) and not the same properties to all the data processed by the application, 2) the user preferences, regarding the security level applied to their private data, are taken into consideration and 3) the mobile device energy consumption constraint is also taken into account. The chapter begins with the presentation of the general criteria that should be taken into consideration when designing a security solution for a mobile cloud application. Then, the framework objectives, the framework basic notions and the framework design are presented.

Chapter 5 describes the implementation on the mobile device for Secure Mobile-Cloud Framework presented in Chapter 4. The chapter begins with the implementation of the proposed framework. Then there is presented the design and the implementation of the databases used for storing the user options followed by the description and implementation of the user interface. Finally there are presented some unit tests for this part of the implementation and the description of the used software tools.

Chapter 6 presents an application scenario. This application scenario was design as components based mobile cloud application. Then, there is presented, a solution for the integration of the application scenario along with the Secure Mobile-Cloud Framework. The integration refers to the introduction of the security framework into the application scenario in order to obtain a secured application scenario. The integration solution is made at the source code level. Two approaches are described for the integration solution: the theoretical approach and the technical approach. The chapter begins with several examples of the existing mobile cloud applications.

Chapter 7 presents the overall conclusions, the overview of contributions is also presented here. Parts of the work that can be continued are presented in future works. This chapter is followed by the bibliography, the list of publications of this work and the most relevant papers.

Security of Mobile Cloud Applications

Security of Mobile Cloud Applications

“We’ve redefined Cloud Computing to include everything that we already do. I can’t think of anything that isn’t Cloud Computing

The computer industry is the only industry that is more fashion-driven than women’s fashion.”

Larry Ellison (CEO of Oracle), 2007s Analysts Conference

2. Cloud Computing

Contents in Brief

2.1 What is Cloud Computing?	8
2.2 Cloud Benefits.....	19
2.3 Security Challenges.....	21
2.4 Conclusions	26

Chapter Overview

The goal of this chapter is to present the state of the art on the Cloud Computing and the main security issues related to this technology

The chapter comprises three main parts. The first part (Section 2.1) presents the various definitions given to the Cloud Computing, the standard definition proposed by NIST that was adopted in various works, and the technologies used by the Cloud Computing. The second part (Section 2.2) describes the benefits of the Cloud Computing. The third part (Section 2.3) focus mainly on the Cloud Computing security concerns. Finally several conclusions are presented in Section 2.4.

2.1 What is Cloud Computing?

Cloud Computing is the term everyone seem to be interested in. It has been blogged about, written about, and talked about in conferences, workshops and magazines. The high interest on this topic it is also showed by the large number of searches for the word pair 'cloud computing' undertaken with the Google search engine. Between 2010 and 2012, the search for the 'cloud computing' term has exceeded the search for terms like 'outsourcing', 'virtualization' or 'grid computing' [GoogleTrends].

Like any other thing brought into the spotlights, Cloud Computing was criticized, applauded, embraced, admired or slandered. It was considered a trap for users and strongly condemned: "It is stupidity. It is worse than stupidity: it's a marketing hype campaign." Richard Stallman quoted in The Guardian. Or it was regarded as an opportunity that could provide for new advantages: "a broad array of web-based services aimed at allowing users to obtain a wide range of functional capabilities on a 'pay-as-you-go' basis" Jeff Kaplan quoted in Virtualization Journal.

Thereby, what exactly is Cloud Computing and why is everyone interested in it? This Section seeks to respond to this question by: 1) showing the various definitions given to the Cloud Computing; 2) describing the Cloud Computing characteristics and 3) presenting the technologies on which Cloud Computing is based on.

2.1.1 Various Definitions

Cloud Computing was used mainly as a marketing term to express various ideas in different contexts [ZCB10]. Shortly after its emergence lot of people, experts or not, have expressed their view about Cloud Computing and have tried to define it. The work in [Geelan08] gathered over twenty different opinions on Cloud Computing. The fact that Cloud Computing did not have a standard definition caused confusion and distrust. Furthermore, the complexity of the ideas, the products and the technologies that Cloud Computing uses and relies on, also made it difficult to establish a standard definition.

The main reason of these different perceptions is that the Cloud Computing, unlike other technical terms, is not a new technology, but rather a new operations model that brings together a set of existing technologies to run businesses in a different way. Its name is a metaphor for the Internet; the "*Cloud*" term in Cloud Computing

Security of Mobile Cloud Applications

comes from the way in which the Internet was represented by the networks architects in the flowcharts and diagrams [Rose10]. It describes a mixture of items whose boundaries are not distinguished from the distance and which together look like a Cloud [VRM+09].

Until now, there is not a unique standard definition but several definitions proposed by [VRM+09], [NIST], [CISCO], [YBS08], [JERICHO] or [AFG+09]. These definitions are discussed below.

Vaquero et al. [VRM+09] gathered most of the definitions and views available for the Cloud and proposed a general definition based on the existing views.

Each definition was analyzed. Then the main common features were listed (e.g. virtualization, variety of resources, automatic adaptation, scalability). These features were put together to create a standard definition. [VRM+09] defined Cloud Computing as: *“Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs.”*

Youseff et al. [YBS08] have proposed the first comprehensive description for the Cloud Computing and its components. Their purpose was to establish a detailed ontology of the Cloud in order to have a better understanding of this technology. They stated that: *“better comprehension of the technology would enable the community to design more efficient portals and gateways for the cloud, and facilitate the adoption of this novel computing approach in scientific environments.”* Youseff et al. [YBS08] defined Cloud Computing as: *“cloud computing can be considered a new computing paradigm that allows users to temporary utilize computing infrastructure over the network, supplied as a service by the cloud-provider at possibly one or more levels of abstraction”*.

Another definition for the Cloud Computing has been given by Armbrust et al. [AFG+09]. Besides features like application, software and hardware delivered as services, Armbrust et al. have highlighted three new aspects from a hardware point of view: 1) the illusion of infinite computing, 2) the increase of resources only if needed, and 3) the payment made only for the used resources. Armbrust et al. defined Cloud Computing as: *“Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services. The services themselves have long been referred to as Software as a Service (SaaS). The datacenter hardware and software is what we will call a Cloud. When a Cloud is made available in a pay-as-you-go manner to the general public, we call it a Public Cloud; the service being sold*

Security of Mobile Cloud Applications

is Utility Computing. We use the term Private Cloud to refer to internal datacenters of a business or other organization, not made available to the general public. Thus, Cloud Computing is the sum of SaaS and Utility Computing, but does not include Private Clouds."

A more technical definition is given by Cisco [CISCO]. Cisco defines cloud computing as: *"IT resources and services that are abstracted from the underlying infrastructure and provided "on-demand" and "at scale" in a multitenant environment."*

In addition to these definitions, there is another one given and published by the NIST (U.S. National Institute of Standards and Technology). This definition is generally well accepted [AFG+09], [ENISA09], [CSA09] and is specifically tailored to the unique perspective of IT network and security professionals. NIST says that: *"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."*

Moreover, Cloud Computing has also been defined in terms of service models, deployment models and essential characteristics. These models and characteristics have been proposed by the NIST [NIST], the Jericho group [JERICHO], and the CSA [CSA09] (Cloud Security Alliance).

In the NIST definition, the Cloud is viewed as: 1) a well-structured layered architecture; and 2) a well-defined deployment model (see Figure 2.1). However, from Jericho group point of view, the Cloud is described as a cube that illustrates the many permutations available in the Cloud. It presents four criteria/dimensions that are intertwined. This model focuses on offering solutions regarding questions about what data and processes to move to the Clouds and where to operate. CSA focuses on a description of Cloud Computing that is specifically tailored to the unique perspective of IT network and security professionals. Thus, in order to define Cloud Computing, CSA adopt the model proposed by NIST. In addition to the NIST model, CSA add the multi-tenancy as essential characteristic. Also, CSA shows how the security responsibility of both the provider and the consumer is shared according to the cloud services.

Security of Mobile Cloud Applications

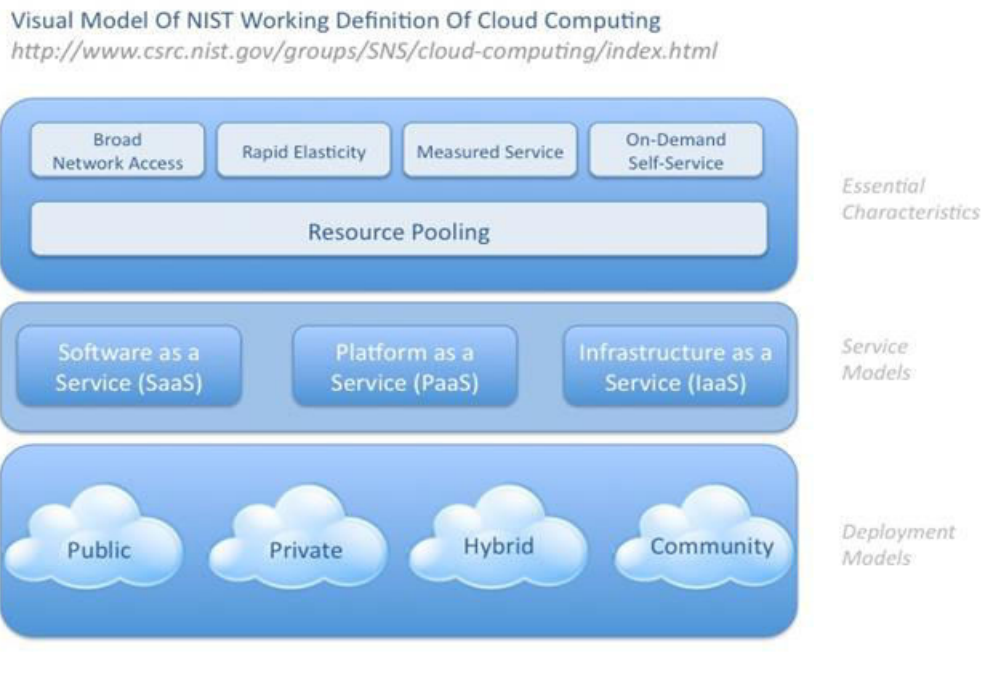


Figure 2.1 NIST Definition of Cloud Computing [NIST]

These different models and characteristics are described in the following.

2.1.2 Cloud Service Models

From the point of view of NIST, the Cloud has three cloud service models, described as overlapped levels, where each level requires the resources of the lower level (see Figure 2.1):

- *Infrastructure as a Service (IaaS)* [NIST] (Figure 2.2): It is the foundation of all cloud services and includes the entire infrastructure resources: storage, hardware, servers, network components and other computing resources. It gives to customers the opportunity to deploy and run operating systems and applications. The advantage offered is that rather than purchasing the hardware resources, a client rents them from the service provider. Examples of IaaS include: Amazon's Elastic Compute Cloud generally known as Amazon EC2 [AEC2], Google Compute Engine [GCE], Rackspace Cloud Servers [RCS] or Joyent Cloud [JCS].

Security of Mobile Cloud Applications

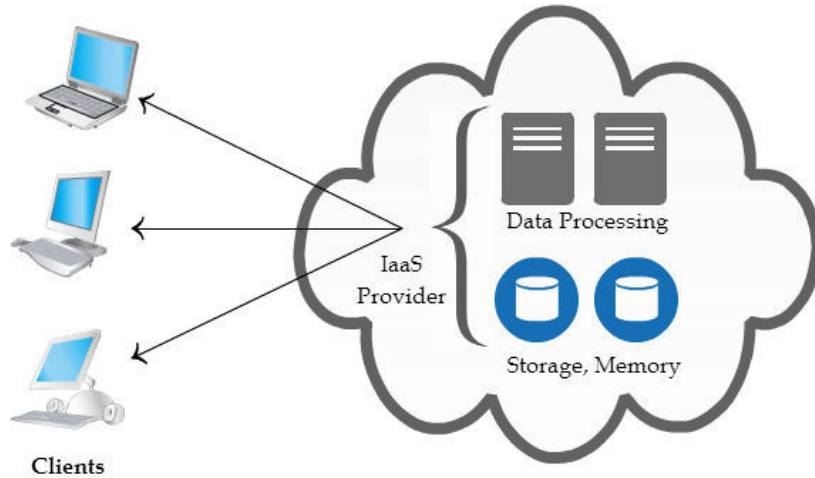


Figure 2.2 Infrastructure as a Service

- *Platform as a Service (PaaS)* [NIST] (Figure 2.3): In this layer, the cloud provider delivers integrated environment for building, testing and deploying applications. Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers. Examples of PaaS are: Google App Engine [GAP], Rollbase [RBase], MS Azure [MSA] or Collabnet [CNet].

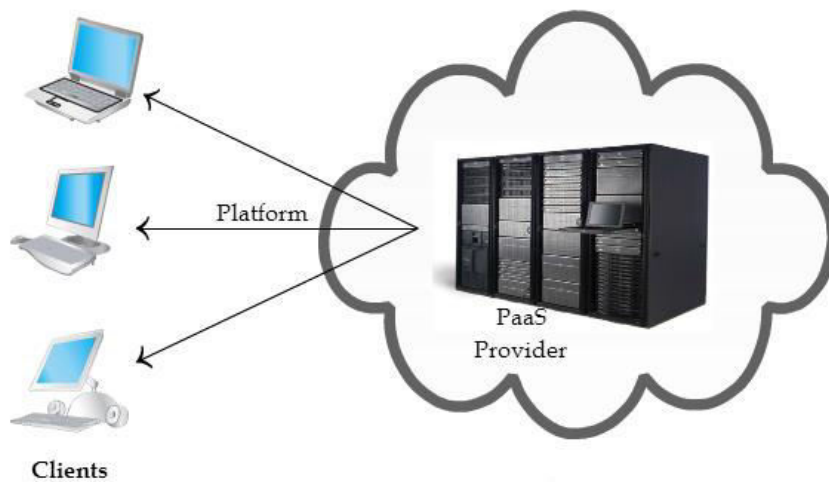


Figure 2.3 Platform as a Service

- *Software as a Service (SaaS)* [NIST] (Figure 2.4): It is built upon the underlying IaaS and PaaS layers. This layer enables the users to access applications remotely via the Internet. In this model, applications are provided as services. Examples of SaaS are Google Apps [GApps], Hyper Office [HO] or Salesforce [SForce].

Security of Mobile Cloud Applications

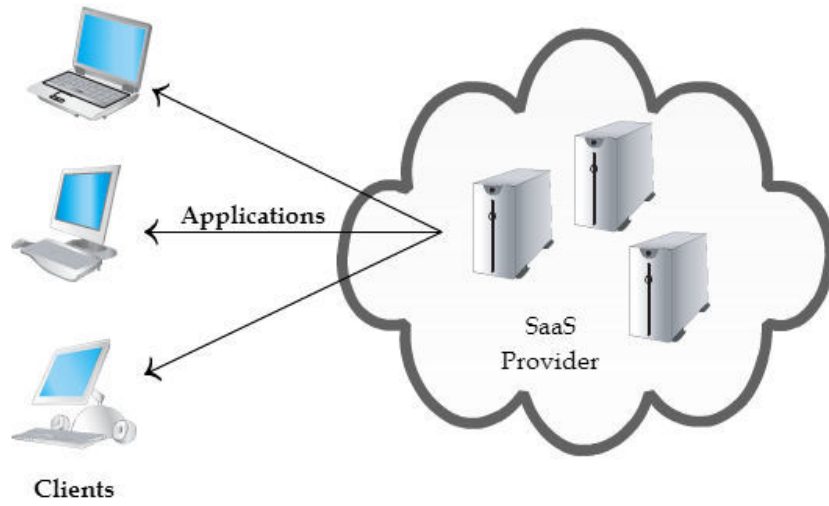


Figure 2.4 Software as a Service

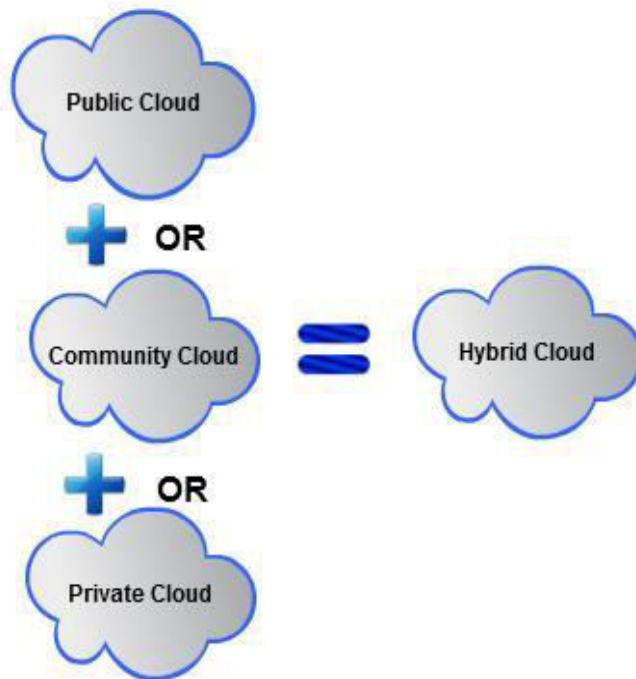


Figure 2.5 Hybrid Cloud

Security of Mobile Cloud Applications

2.1.3 Cloud Deployment Models

According to NIST, there are four cloud deployment models (Figure 2.5 and Figure 2.6):

- *Public Cloud*: The Cloud infrastructure is available to the general public. It is owned by third party organizations that sell cloud services.
- *Private Cloud*: Only an organization or a user uses the Cloud infrastructure. It can be managed by the organization or by a third party.
- *Community Cloud*: Several groups that have common interests share the Cloud infrastructure. The organizations or a third party can manage it.
- *Hybrid Cloud*: It is the composition of two or more cloud models that are bounded together (e.g. private, public or community).

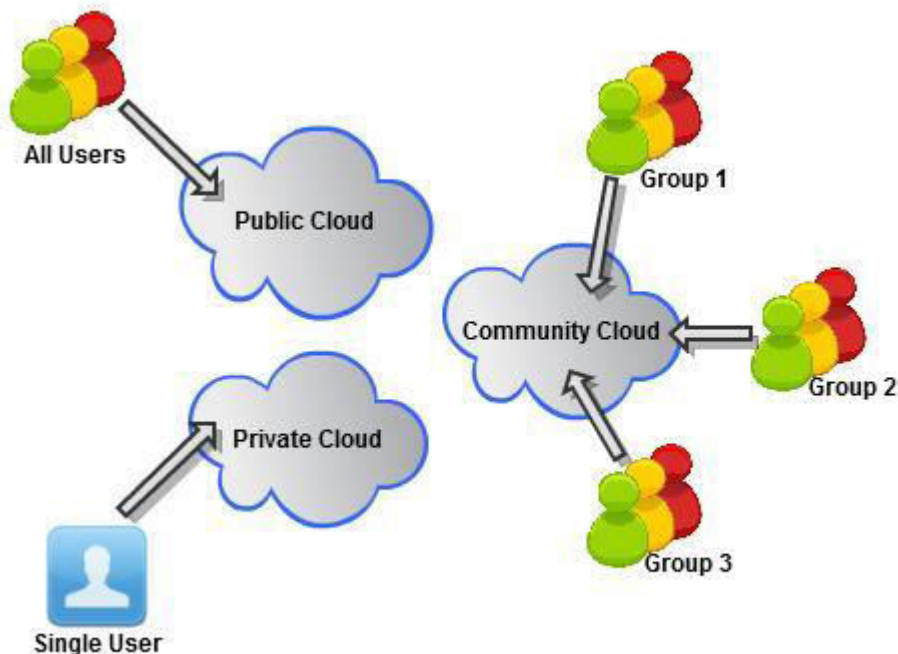


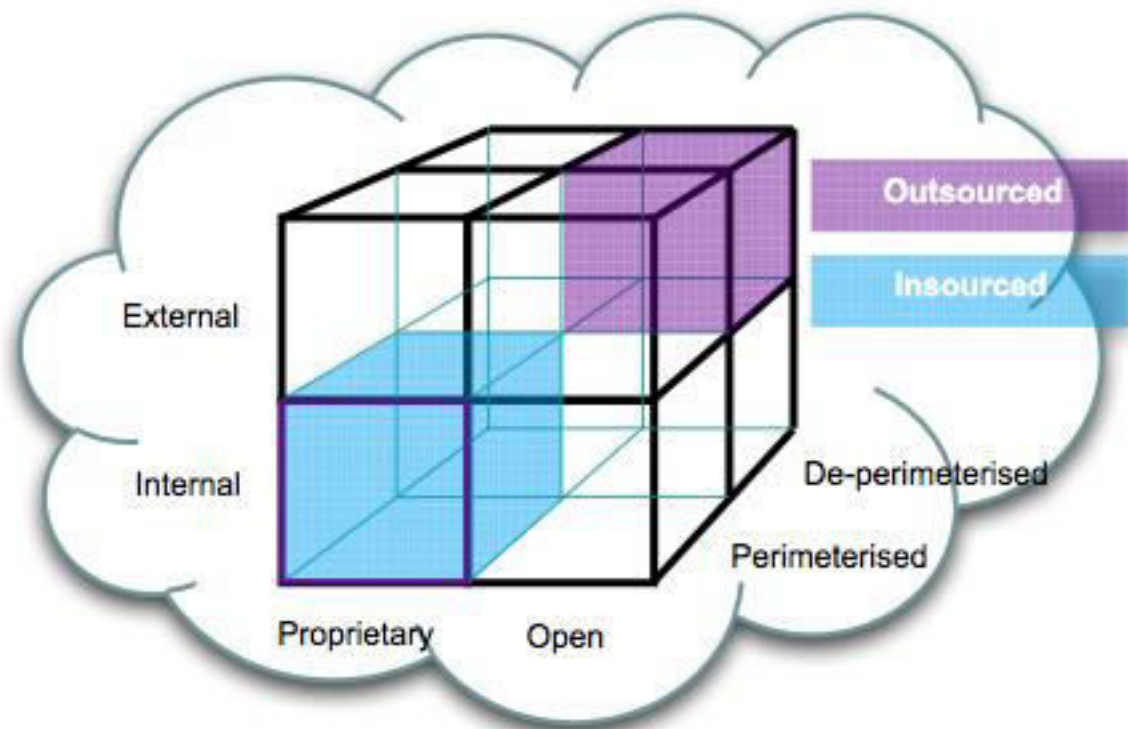
Figure 2.6 Deployment Models – Public, Private and Community

In addition to these four deployment models proposed by NIST, the Jericho Group [JERICHO] came with another approach. In this approach Cloud Computing was described as a Cloud Cube Model.

Security of Mobile Cloud Applications

The Cloud Cube Model dimensions are:

- *Internal/External*: It is the first dimension of the Cloud Cube Model. It points out the Cloud location, which can be inside or outside an organization boundary.
- *Proprietary/Open*: This dimension defines the state of ownership of the cloud services [JERICHO]. In case of a proprietary state, the organization that provides the service is keeping the means of provision under their ownership. Thereby the migration from a Cloud supplier to another requires more effort and investment. Their open state ownership does not impose so many constraints.
- *Perimeterised/De-perimeterised*: It represents the model's third dimension. It describes the operating place, inside or outside the traditional IT perimeter. For a certain computing task, an organization may expand the perimeter; then, when the task is completed, the perimeter is withdrawn to its original size [JERICHO].
- *Inourced/Outsourced*: It corresponds to the model's fourth dimension. Its purpose is to show who manages the cloud services used by a client, i.e. a third party or the client [JERICHO].



The Cloud Cube Model

Figure 2.7 Jericho's Model [JERICHO]

Security of Mobile Cloud Applications

2.1.4 Cloud Characteristics

In addition to the service and deployment models, the NIST has defined five essential characteristics that have been redefined and/or completed by [Katzan09], [WG08] [ZCB10], [BLR+11], [VRM+09] and [HB12] as follows:

- *On-demand self-service*: The Cloud services are automatically provided to a consumer at any time. There is no need for the consumer to require human interaction with service providers.
- *Broad network access*: The Cloud services are available over the network. They may be accessed by diversified thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations), as well as by traditional or cloud-specific software.
- *Resource pooling*. The provided resources (e.g. storage and processing, network bandwidth, virtual machines) are being used by multiple consumers through a multi-tenant model. Generally, a consumer does not know the exact location (e.g., country, state, or data centre) of the resources she/he uses.
- *Rapid elasticity*. The resources may be provided in a quick and flexible way. From a consumer point of view, the services provided are endless and may be accessed in any quantity and at any time.
- *Measured service*. The resource utilization can be controlled and monitored. At any time, various reports may be made to ensure transparency for both the service provider and the service consumer.
- *User-friendly*. The user does not have to change the working habits when using Cloud interfaces. The interfaces are location-independent and the software that the users have to install is light on the client side.
- *Self-organizing*. Each Cloud provider is entitled to manage the services it offers according to its own needs. Furthermore, providers may respond in a quick way to the changes that can occur in service demand.
- *Scalability*. Cloud services and computing platforms offered by Clouds could be scaled across various concerns, such as geographical locations, hardware performance, and software configurations.

The Table 2.1 lists the characteristics presented by NIST and shows how often they are described in some other works (under the same name or under a different one).

Security of Mobile Cloud Applications

TABLE 2.1 CHARACTERISTICS OF CLOUD COMPUTING

Characteristics	NIST	[BLR+11]	[VRM+09]	[WG08]	[ZCB10]	[Katzan09]	[HB12]
On-demand self-service	X	X	X	X	X	X	X
Broad network access	X	X			X	X	X
Resource pooling	X				X	X	X
Rapid elasticity	X		X	X	X	X	X
Measured service	X	X	X		X	X	X
User-friendly			X	X			X
Self-organizing			X	X	X		X
Scalability		X	X	X			X

In addition to the characteristics defined by NIST as the essential characteristics of Cloud Computing, CSA [CSA09] has identified multi-tenancy as an important element of the Cloud. Multi-tenancy describes a mode of software operation where multiple independent instances of one or multiple applications operate in a shared environment. For example, in a multi-tenancy environment, various users who do not share or see each other's data can share the same applications while running on the same operating system, using the same hardware and the same data storage mechanism.

Besides service models, deployment models and characteristics, Cloud Computing was also defined by describing several existing technologies that contributed to its development. These technologies are presented in the following.

2.1.5 Technologies Behind

Cloud Computing is not something that appeared suddenly overnight. It uses some elements from the previous existing technologies: Grid Computing [BFH03], Utility Computing [Rappa04] and Automatic Computing [JL03] (see Figure 2.8).

Security of Mobile Cloud Applications

Moreover other technologies such as: virtualization [Creasy81], SOA [NL05] and Web 2.0 [MR08], underlying Cloud Computing contributed to its development.

- *Grid Computing*: Grid computing is a computing paradigm [FRL08] that enables sharing resources (e.g. supercomputers, data sources, storage systems, specialized devices) geographically distributed over a large area [BYV08], in order to fulfill, at the same time, a single computational objective. Grid Computing is more academia-oriented [BM09] and it has been developed to address scientific or technical problems requiring intensive computation. A well-known example is *Search for Extraterrestrial Intelligence project* [SETI]. Cloud Computing as well as Grid Computing employs distributed resources in order to improve and to increase the applications functionalities [ZCB10]. As a step forward, Cloud Computing focuses in offering real-time services; applying virtualization technologies in order to realize the resource sharing and allow resources to grow, shrink and self-heal dynamically [HB12].
- *Utility Computing*: Utility computing is a business model [Haleem12] that presents the computing resources as measurable services similar to a physical public utility, such as electricity. Its main properties are: 1) share resources fairly between consumers; 2) avoid resource starvations; and 3) maximize resource utilization [HT12]. Cloud computing adopted the utility-based model for economic reasons (e.g. it allows the use of pay-per-use mechanism).
- *Autonomic Computing*: Autonomic computing idea was introduced by IBM in 2001 and was inspired by human autonomic nervous system [ZCB10]. An autonomic system is characterized by one of the following features: 1) self-configuring; 2) self-healing; 3) self-optimizing; and 4) self-protecting [Ganek07]. The goal of autonomic computing is to overcome the management complexity of today's computer systems [ZCB10]. Cloud Computing infrastructures and platforms have been designed and built based on autonomic computing concepts to 1) reduce the complexity of resources management; 2) increase the resource availability; 3) raise resources flexibility; and 4) optimize resource employment [HMC08].
- *Virtualization*: The idea of virtualization emerged in 1965 when IBM created the first Virtual Machine Monitor (VMM) [Creasy81]. Since then, virtualization technologies have rapidly evolved. Virtualization is the technology on which Cloud Computing relies. It offers flexible and scalable hardware services; and creates an abstraction layer between computing resources and the software application that uses them [HT12]. For a user, virtualization creates the impression of applications that: 1) run simultaneously and 2) use all the available resources [FRL08].

Security of Mobile Cloud Applications

- *SOA*: SOA (Service-Oriented Architecture) is a business model offering a way of thinking about IT assets as service components. The SOA approach is based on creating stand-alone, task-specific reusable software components that function and are made available as services [Papazoglou03]. The services organization and orchestration inside Clouds could be managed in a Service- Oriented Architecture. Furthermore, the Cloud services could be used in SOA application environment, making them available on various distributed platforms to be accessed across the Internet [WG08].
- *Web 2.0*: The Web 2.0 technology provides techniques that aim to enhance information sharing, collaboration and functionality of the Web [Oreilly08]. Web 2.0 developed aiming to improve interconnectivity and interactivity of Web applications. The benefits that Web 2.0 technology offers to the users are: 1) an easier and more efficiently way to access the Web; 2) a large number of Web applications with richer functionality [MR08].

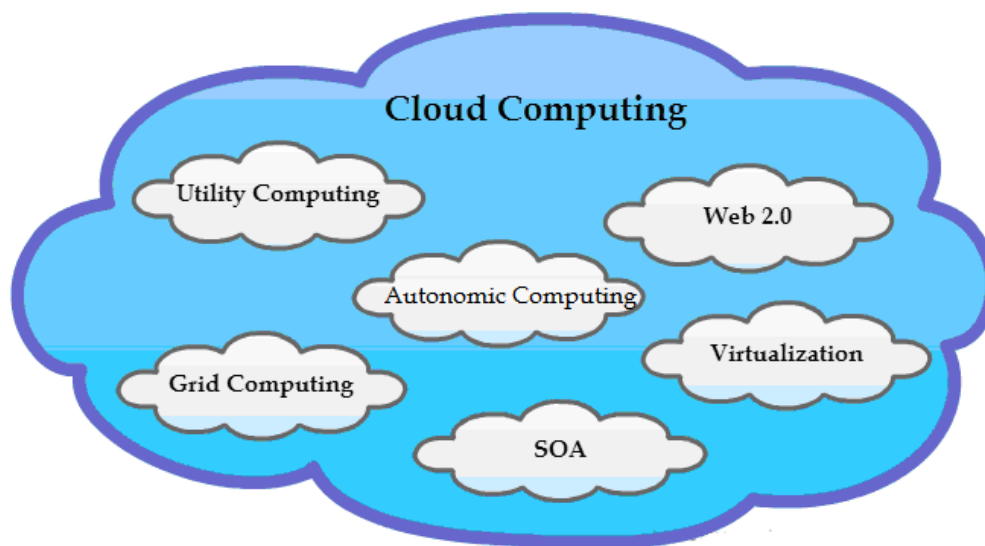


Figure 2.8 Technologies behind Cloud Computing

2.2 Cloud Benefits

The Cloud Computing popularity increases also with the benefits it provides. Some of these benefits are presented in this section.

Security of Mobile Cloud Applications

Cloud Computing brings a change in the way that IT services are delivered. These changes provide several benefits to both Internet simple users and to organizations. The Internet users can enjoy new Web applications with rich functionalities that may be accessed from various devices. In this way, the users do not need to deal with the management of the applications (e.g. install or upgrade) because, thanks to the Cloud, this is done automatically in the Cloud. The organizations may choose to not own and manage the IT services and use the Cloud services to support their IT requirements. In this way, they can reduce the cost of hardware maintenance and human resources.

More generally, the various advantages (technological, economical, etc.) of Cloud Computing have been highlighted in several works [SW12], [Agarwal11], [Sun09], [Swaminathan08]. The Figure 2.9 below gives an overview about these benefits.

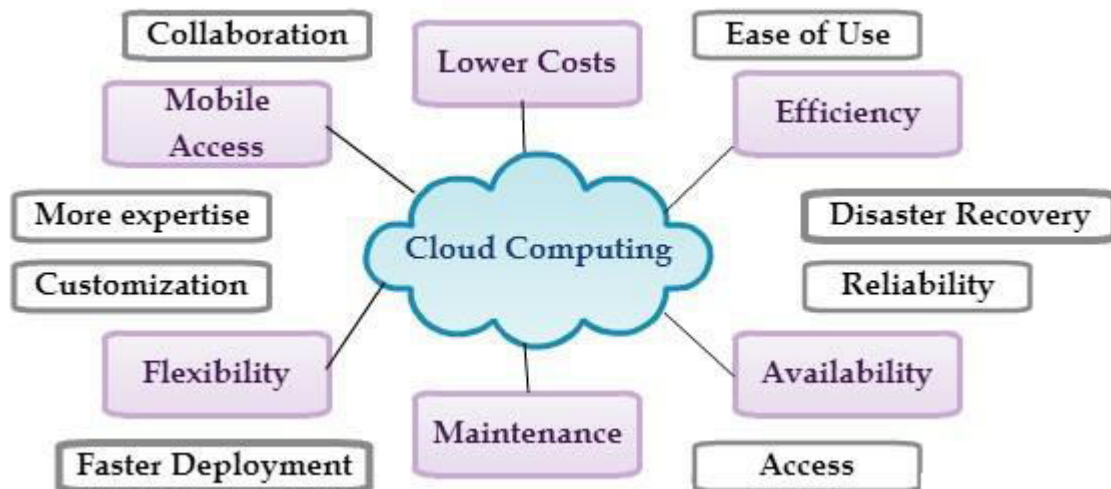


Figure 2.9 Cloud Computing benefits

The most common benefits are:

- *Savings*: It is the most frequently mentioned benefit. Cloud Computing lowers the cost for using IT services and the cost of everything that has to do with IT maintenance. It reduces the cost of acquiring and maintaining computing power and allows the organizations to buy only the computing services needed. It also reduces the purchase and updating cost of the software that an organization or an end-user needs. The Cloud services may be acquired for a specific period of time. This enables agencies to not invest in complex and expensive IT infrastructures which may be used only temporarily.

Security of Mobile Cloud Applications

- *Flexibility:* The Cloud enables users to adapt the services they use according to their needs. Companies may start with a small amount of resources and grow to a large amount of resources rapidly, and then scale back if necessary. Furthermore, the companies can adapt to the users demands over a period of time; if the users demands grow, companies may extend their resources and then release them after the demands fall.
- *Maintenance:* The maintenance is made by cloud providers. A customer (company/user) does not need to worry about the hardware, the operating system or the software maintenance. The maintenance cost is included into the cost of the service provided. This gives the customer the impression of a free maintenance.
- *Availability:* From a customer point of view, cloud services are always available. she/he does not need to worry about the management of failures that may occur while a certain service is used. The Cloud providers are required to manage any type of problem that may arise. Furthermore, Cloud Computing can be utilized as a viable disaster recovery option—especially for storage.
- *Efficiency:* The maintenance made by Cloud providers may increase the overall efficiency of the system.
- *Mobile Access:* Mobile users can benefit from the services provided by Cloud Computing. The reduced storage and processing capacities that characterized till now the mobile device were an impediment into the use of various tasks and applications. Now, through Cloud services, mobile users can take advantages of external IT services and use various applications.

2.3 Security Challenges

As presented in the previous section, Cloud Computing brings different benefits to its customers. However, even if the Cloud Computing seems so great, lot of companies are not ready to use it. Actually, a group of researchers from Berkeley University of California [AFG+09] identified ten obstacles to Cloud Computing: 1) availability of service, 2) data lock-in, 3) data confidentiality and auditability, 4) data transfer issues, 5) performance unpredictability, 6) scalable storage, 7) bugs in large distributed systems, 8) scaling quickly, 9) reputation fate sharing, and 10) software licensing. In [Ness09] three major barriers to Cloud Computing have been identified: 1)

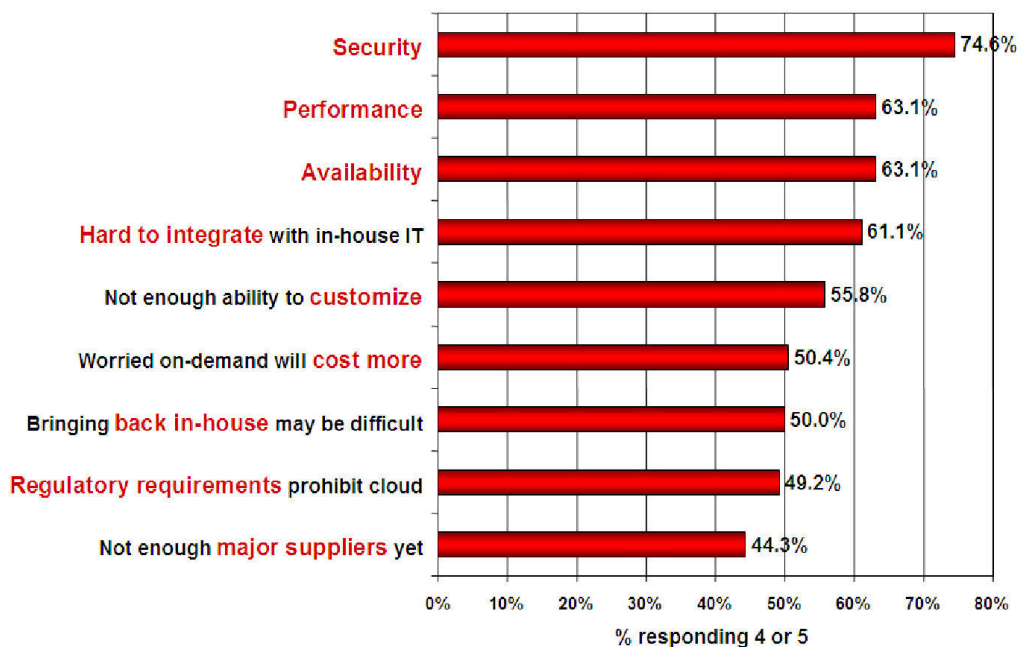
Security of Mobile Cloud Applications

Cloud depends on new approaches to security, 2) Cloud can break static networks, and 3) network automation is critical. By contrast, Leavitt [Leavitt09] argued six challenges to adopt Cloud Computing: 1) control performance, 2) latency and reliability; 3) security and privacy; 4) related bandwidth costs; 5) vendor lock-in and standards; and 6) transparency.

According to a study made by IDC Enterprise Panel [Gens09], security is the major reason why enterprises are not yet willing to adopt Cloud Computing.

Q: Rate the **challenges/issues** ascribed to the 'cloud'/on-demand model

(1=not significant, 5=very significant)



Source: IDC Enterprise Panel, August 2008 n=244

Figure 2.10 Reasons for enterprises not to adopt Cloud Computing [Gens09]

The Cloud acts as a big black box where nothing inside is visible to the clients. Therefore clients have no idea or control over what happens with their assets.

Cloud Computing is about clients transferring the control of their resources (e.g data, applications) and responsibilities to one or more third parties (cloud services providers). This brings an increased risk to which client assets are greatly exposed.

Before Cloud's emergence, generally, the companies were keeping their data inside their perimeter and protecting them from any risks caused by malicious intruders. A malicious intruder was considered to be an outside attacker or a malicious employee. Now, if a company chooses to move its assets into the cloud, it is forced to trust the Cloud provider and the security solutions it offers when provided. However, even if the cloud provider is honest, it can have malicious employees (e.g system

Security of Mobile Cloud Applications

administrators) who can tamper with the virtual machines and violate confidentiality and integrity of client's assets.

In Cloud Computing the obligations in terms of security are divided between the cloud provider and the cloud user. In the case of SaaS, this means that the provider must ensure data and application security; so service levels, security, governance, compliance, and liability expectations of the service are contractually stipulated and enforced. In the case of PaaS or IaaS the security responsibility is shared between the consumer and the provider. The responsibility of the consumer's system administrators is to effectively manage the data security. The responsibility of the provider is to secure the underlying platform and infrastructure components and to ensure the basic services of availability and security [CSA09].

Several analyses have been conducted to identify the main security issues regarding the Cloud Computing [ENISA09, CSA09, TCSA10, TCSA13, SK10, OWASP10, CDB11, RSB+09, SL10]. Following these analyses, security issues have been classified in terms of concerns: domain concerns, services concerns, threats, actors concerns and properties concerns (Figure 2.11).

The domain concerns are divided in two types: 1) governance concerns and 2) operation concerns.

Governance address strategic and policy issues within cloud computing [CSA09]. Loss of governance can generate several issues like unclear roles and responsibilities, lack of information on jurisdictions or lack of completeness and transparency [ENISA09].

Operation concerns focus on technical security constraints.

Cloud Security Alliance (CSA) released their document "Security Guidance for Critical Areas of Focus in Cloud Computing" in December 2009. In the document, there were identified twelve areas of concerns. These are: Domain 1: Governance and enterprise risk management, Domain 2: Legal and electronic discovery, Domain 3: Compliance and audit, Domain 4: Information lifecycle management, Domain 5: Portability and interoperability; Domain 6: Traditional security, business continuity, and disaster recovery, Domain 7: Data center operations, Domain 8: Incident response, notification, and remediation, Domain 9: Application security, Domain 10: Encryption and key management, Domain 11: Identity and access management, Domain 12: Virtualization [CSA09].

Security of Mobile Cloud Applications

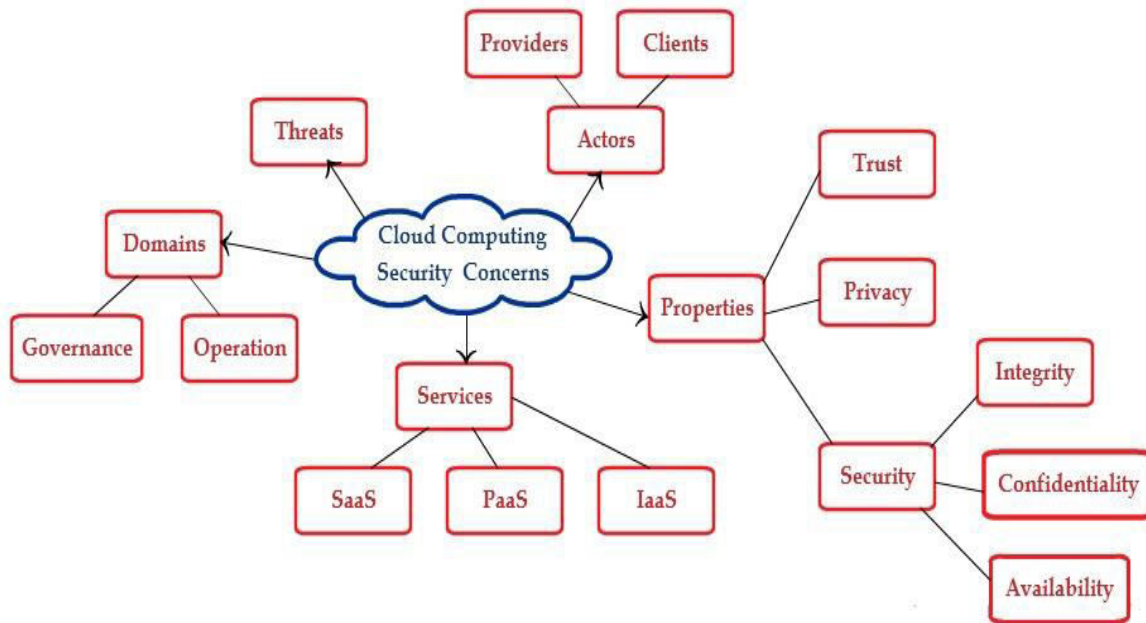


Figure 2.11 Terms of concerns

The threats were described also by CSA in several papers entitled “Top threats to Cloud Computing”. CSA had published their research findings on the top threats to cloud computing in 2010 [TCSA10] and 2013 [TCSA13]. The purpose of the research was to assist cloud providers as well as their potential customers in identifying the major risks and to help them decide whether or not to join in cloud infrastructure, and also, how to proactively protect them from these risks. In the 2013 research, top nine threats are mentioned: “Data Breaches, Data Loss, Account or Service Traffic Hijacking, Insecure Interfaces and APIs, Denial of Service, Malicious Insiders, Abuse of Cloud Services, Insufficient Due Diligence and Shared Technology Vulnerabilities”. The document shows, for each risk, the following: 1) the services models where the risk may arise; 2) the risk level (e.g. high or low); 3) the risk relevance; 4) how the risk rank modified since 2010; 5) the domains from CSA guidance where the risk can be placed.

The services concerns were presented by Subashini and Kavitha in the paper called “A survey on security issues in service delivery models of cloud computing”. The paper argues that the cloud elements security depends on the deployment model that is used. Therefore, it proposes a survey more specific to the different security issues that have emerged due to the nature of the service delivery models. The fundamental security challenges mentioned in the paper are: data storage security, data transmission security, application security and security related to third-party resources [SK10]. A list of top vulnerabilities to SaaS model is maintained by the Open Web Application

Security of Mobile Cloud Applications

Security Project (OWASP); the list is updated as the threat landscape changes [OWASP10].

Choubey et al. [CDB11] have done a short but very specific review of cloud computing security and identified the key advantages, disadvantages and trade-offs between cost and security. Rad et al. [RSB+09] have done a survey of cloud platforms that mainly focused on foundation, storage system, infrastructure service and integration. Srinivasamurthy and Liu [SL10] have done another survey on secure cloud architecture advantages and different security threats with some existing ways to minimize these threats.

The properties that bring out the security issues encountered in the Cloud are: the privacy, the security and the trust.

Privacy [ITU01] is the desire of a person to control the disclosure of personal information. Organizations dealing with personal data are required to obey to a country's legal framework that ensures appropriate privacy. The cloud presents a number of legal challenges towards privacy issues involved in data stored in multiple locations in the cloud, additionally increasing the risk of confidentiality and privacy breaches [SM13]. Instead of its data being stored on the company's servers, data is stored on the service provider's servers, which could be in Europe, Asia, or anywhere else. This tenet of Cloud Computing conflicts with various legal requirements, such as the European laws that require that an organization know where the personal data in its possession are at all times [ZL10].

Security in general, is related to the following aspects: confidentiality, integrity and availability [Danielson08].

Confidentiality refers to the prevention of the unauthorized disclosure of the information. Assets delegated to the Cloud are subject to a high risk of compromise, because they become accessible to an augmented number of parties [CPK10].

Integrity points out that an asset like data, software or hardware can only be modified by authorized parties in authorized ways. Integrity covers two important aspects: data integrity and software integrity. Both data and software integrity refers to the protection from unauthorized deletion, modification or fabrication [CWR10].

Availability refers to the property of a system being accessible and usable upon demand by an authorized entity. System availability includes a systems ability to carry on operations even when some authorities misbehave. The system must have the ability to continue operations even in the possibility of a security breach [CPK10].

Trust shows how two parties are involved in a transaction: "An entity A is considered to trust another entity B when entity A believes that entity B will behave exactly as expected and required" [ITU01].

2.4 Conclusions

Since its emergence a lot has been said about Cloud Computing. It has been defined many times in several ways and by different peoples. The fact that it did not have a standard definition from the beginning created confusion and mistrust. A standard definition was given by NIST. Beside the definition, NIST presented the following Cloud Computing features: services models, deployment models and characteristics.

Cloud Computing includes several elements of previous existing technologies: Grid Computing, Utility Computing, Automatic Computing, virtualization, SOA and Web 2.0. These elements were combined and used in order to bring various benefits to the Internet users.

Cloud Computing changes the way how IT services are delivered. This feature provides several benefits to both Internet simple users and to the organizations.

Cloud Computing raises various issues. But the most important and the most frequently mentioned are the security issues. The security issues are numerous, as presented into the existing analyses, and they were classified in various groups: domain concerns, services concerns, threats, actors concerns and properties concerns. The security issues are the major reason why enterprises are not yet willing to adopt Cloud Computing.

2.4.1 Contributions

A state of the art on Cloud Computing was made [PBC+13g]. In this state of the art were presented: 1) the various definitions given to the Cloud Computing, 2) the benefits of the Cloud Computing and 3) the Cloud Computing security concerns.

2.4.2 Publications

[PBC+13g] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "Overview on Mobile Cloud Computing Security Issues", in Scientific Bulletin of the Politechnica University of Timisoara, Transactions on Electronics and Communications, 2013 (Submitted - accepted for publication).

“... based on cloud computing service development, mobile phones will become increasingly complicated, and evolve to a portable super computer ...”

Eric Schmidt (Google CEO), 2010 interview for ITNews

3. Mobile Cloud Computing

Contents in Brief

3.1 What is Mobile Cloud Computing?.....	28
3.2 Mobile Cloud Computing - Security.....	34
3.3 Conclusions	46

Chapter Overview

This chapter presents a state of the art on Mobile Cloud Computing, mobile cloud applications models, security issues and existing security solutions for this kind of applications.

The chapter is split in two main parts. The first part (Section 3.1) focuses on mobile cloud applications models after explaining what the Mobile Cloud Computing is. The second part (Section 3.2) presents the main security threats targeting mobile cloud applications and the approaches proposed to address these issues. At the end several conclusions are presented (Section 3.3).

3.1 What is Mobile Cloud Computing?

Mobile devices have turned into mini-computers that people carry constantly with them and from which they expect to be connected to the Internet twenty four hours a day seven days a week.

Just a short time ago a user was only expecting from her/his mobile phone to allow her/him to perform activities using just the device resources (e.g. to take pictures and save them locally on the device, or to read different types of files that were saved locally).

Today, the same user wants to be able to take advantage of powerful and complex applications that manipulate not only the mobile local resources but also external resources as computation power and storage place. To obtain these types of performances several improvements have been made in the domains of mobile hardware and network [KCK11]. Even with those improvements mobile devices still have a lack of resources and energy, an unstable connectivity and introduce several security issues.

To resolve some of these issues, the concept of mobile cloud computing has been proposed as a solution where the Cloud is used as a platform to execute mobile applications. Mobile Cloud Computing as a term was born shortly after the emergence of Cloud Computing model in 2007 [Sood12]. Marketing research [Chetan10] stated that in 2015 there would be more than 240 million customers using Mobile Cloud Computing services while in 2008 there were only 42.8 million customers.

Thanks to the emergence of Mobile Cloud Computing different novel mobile applications models have been defined where the Cloud is used to overcome the limitations imposed by mobile devices such as processing power, memory capacity and display size.

Mobile devices are vulnerable to numerous security threats that aim the theft of users' data. Moreover, as seen previously, the Cloud Computing introduces several security, privacy and trust issues regarding the data stored in the Cloud. Consequently to maintain consumer's trust in mobile platforms more specifically in mobile cloud applications, it is important to secure data that will be used and processed by mobile cloud applications. In this context several security solutions have been proposed to protect mobile cloud data and applications from malicious users.

In order to answer the question raised in the section title, it will be presented in the following, several definition and characteristics of Mobile Cloud Computing along with the mobile cloud applications models.

Security of Mobile Cloud Applications

3.1.1 Definitions

Mobile Cloud Computing (Figure 3.1) is a new concept that can be described as the availability of Cloud Computing resources and services for mobile devices.

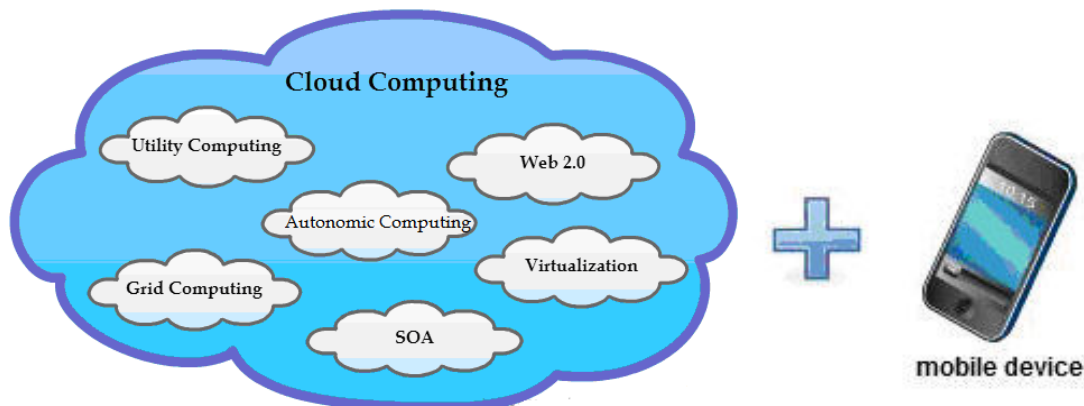


Figure 3.1 Mobile Cloud Computing

As in the case of Cloud Computing, there are several opinions on what Mobile Cloud Computing is. There is not a consensual definition for Mobile Cloud Computing.

For example:

Mobile Cloud Computing is defined in [MCCF] as follows:

“Mobile cloud computing at its simplest refers to an infrastructure where both the data storage and the data processing happen outside of the mobile device. Mobile cloud applications move the computing power and data storage away from mobile phones and into the cloud, bringing applications and mobile computing to not just smart-phone users but a much broader range of mobile subscribers.”

Another definition given in [AEPONA10]:

“Mobile cloud computing is a model for transparent elastic augmentation of mobile device capabilities via ubiquitous wireless access to cloud storage and computing resources, with context-aware dynamic adjusting of offloading in respect to change in operating conditions, while preserving available sensing and interactivity capabilities of mobile devices.”

The first definition emphasizes that Mobile Cloud Computing benefits from Cloud Computing features – storage and data processing, and also reveals a Mobile Cloud Computing characteristic – moving part of the computation and the storage away from mobile phones.

Security of Mobile Cloud Applications

The second definition is more concise. It starts by saying what is Mobile Cloud Computing – a model; it also tells the purpose of using Mobile Cloud Computing – to overcome the mobile device challenges; it tells the way – using storage and computation resources offered by Cloud Computing model; it also specifies that is appropriate to take into account the context of the mobile operating conditions.

Table 3.1 summarizes the key characteristics of Mobile Cloud Computing described in various papers.

TABLE 3.1 KEY CHARACTERISTICS OF MOBILE CLOUD COMPUTING

Characteristics/ Papers	Model	A combination	Outside device processing	Outside device storage	Elasticity	Remote access
[CL12]	X	X	X	X		
[KA11]	X		X	X		X
[KLK12]		X				X
[DLN+11]		X	X	X		
[AEPONA10]	X		X	X	X	
[MCCF]	X		X	X		
[Christensen09]	X	X				
[KMK+12]			X	X	X	X
[QG11]	X	X	X	X		X
[FLR12]	X	X	X	X	X	X

As a conclusion, we can say that Mobile Cloud Computing offers Cloud Computing resources such as storage and computations to the mobile devices with limited CPU speed, memory capacity and display size; which allows the development, deployment and execution of powerful mobile applications.

3.1.2 Mobile Cloud Applications Models

In order to benefit as much as possible from the advantages of Cloud Computing, several mobile cloud applications models have been proposed.

Beside the models already available for mobile devices, like offline and online applications models [KCK11], different novel applications models were proposed [CM09], [CBC+10], [ZSG+09], [MGL+11], [GZK+11], [HCD10] using techniques like: augmented execution, elasticity and mobility to overcome the mobile devices limitations by distributing the application execution. A comparison of these techniques and models is given in [KCK11]:

- Augmented execution is a technique proposed by researchers from Berkeley. This technique assumes creating virtual clones of smartphones execution environments on non-mobile computers and pushing task execution to these virtual devices [CM09].
- Elasticity is a property applied to an application. An elastic application is split or partitioned so that the execution occurs partially on the device and partially on the cloud [ZSG+09].
- Mobility is the ability to change any part of the infrastructure used by some parts of an application; this has to be done without interrupting the application execution.

Mobile Cloud applications are classified in three main model classes: Client model class, Client/Cloud model class, Cloud model class [KCK11].

3.1.2.1 Client Model Class

The Client model class (Figure 3.2) includes the online applications. The mobile is seen only as a more convenient way to access the Internet. The online applications, known as thin client applications, use the mobile device browser to run. They are accessible only when there is an Internet connection. The user interface, the business logic and the database are stored and run on a remote server. Examples of Client model applications are Facebook or Tweeter.

Security of Mobile Cloud Applications



Figure 3.2 The Client Model

3.1.2.2 Client/Cloud Model Class

The Client/Cloud model class (Figure 3.3) includes applications that are distributed between the mobile device and the Cloud. The applications in this class are characterized by the following features: 1) they have the user interface running on the mobile device; 2) the business logic is split between the mobile device and the Cloud; 3) the databases are deployed in Cloud; 4) the latest version of data from databases is saved on the mobile device while the application is running in order to use them when there is no internet connection; 5) if there are new data on the mobile device (created while there was no internet connection) they will be synchronized with data in the Cloud.

But now, because it could take advantages of Mobile Cloud Computing features there have been proposed new applications models; some of them are presented in the following.

Researchers from Berkeley proposed in [CM09] a mobile cloud application model, CloneCloud, based on the following idea: “let the smart-phone host its expensive, exotic applications”. This model uses the augmented execution technique that allows loading the entire application or some tasks of the application into the cloud. In the Cloud, mobile phone’s clones perform the loaded parts. When the operation is completed, the results are reintegrated in the smartphone. In this way several illusions are created for both: the applications users and the applications programmers. The applications users get the impression of a much powerful computation device; while the applications programmers get the impression that there is no need to split the applications while implementing them.

In [CBC+10] is proposed, an architecture called MAUI that also use the augmented execution idea, but it is combined with the properties of today’s managed code environments (portability, programming reflection, type safety, serialization). The

Security of Mobile Cloud Applications

goal of MAUI is to partition the application code at runtime in order to save mobile device's energy under the current networking conditions (connectivity, bandwidth and latency).

In [ZSG+09], Zhang et al. propose a framework that uses the elasticity property to design applications. Thanks to this framework, a new applications model has been developed: elastic applications. An application of this type is divided into several components called "weblets". Each component can run on the mobile side, on the cloud side or can migrate between the mobile and the cloud. The components operate independently but communicate with each other. Concerning the migration, particularly the "weblets" migration, the decision is taken by an elasticity manager according to the user's preferences, mobile computation load or network conditions. The main advantage of this model compared to the models proposed in [CM09] and [CBC+10] is that the "weblets" are not tied to a particular programming language.

Furthermore in [MGL+11], HP researchers proposed " μ Cloud", a new paradigm of rich mobile applications. Rich mobile applications are designed for mobile devices to provide rich functionalities, offline usability and portability through Cloud Computing. The proposed framework uses the mash-up concept where an application is viewed as a graph of components distributed between the mobile device and the cloud. A component is independently developed, never communicates with the others, the underlying runtime platform routes data across components. The main advantage of this framework is that it allows using/reusing a component by different applications.

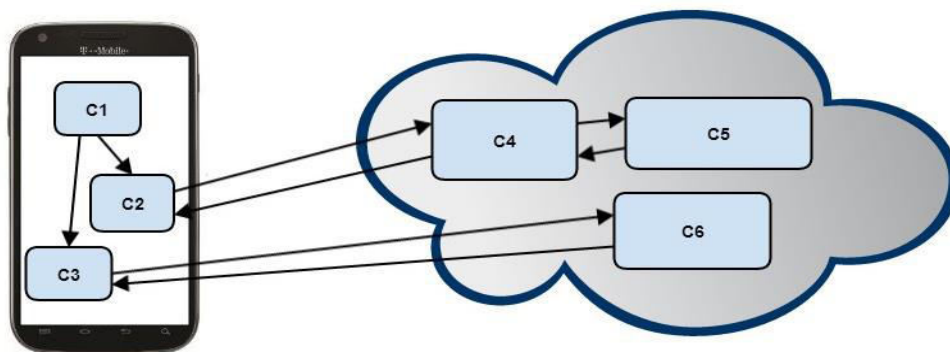


Figure 3.3 The Client-Cloud Model

Security of Mobile Cloud Applications

3.1.2.3 Cloud Model Class

In a Cloud model class (Figure 3.4), the mobile device is part of the Cloud. The goal of this model is to provide a distributed infrastructure that exploits the storage capacity and computing of several mobile devices in order to support new applications. As an example, in [HCD10] a framework is proposed to replace a traditional provider with a network of mobile devices found in user's proximity. This approach tries to resolve the issue of no connectivity to Cloud Computing.

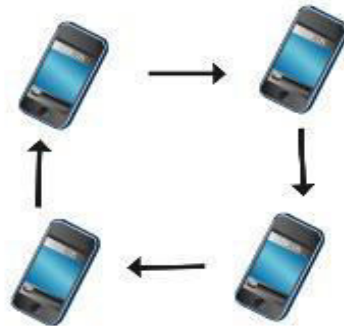


Figure 3.4 The Cloud Model

3.2 Mobile Cloud Computing - Security

Mobile Cloud Computing exposes private data of the mobile user to different security risks. User's data can be stored on the mobile side or on the Cloud side, can be accessed by applications (or application components) running on the mobile device or in Cloud, or can be transmitted between the mobile device application components and Cloud application components.

This section presents in the first part the security issues related to Mobile Cloud Computing and highlights in the second part the state of the art work proposed to address these security issues.

3.2.1 Security Issues of Mobile Cloud Computing

As we have said previously, Mobile Cloud Computing is a combination of mobile and Cloud Computing. Thus, the security issues in Mobile Cloud Computing are due to the security threats against the Cloud, the mobile devices and the applications running on these devices. These threats can be classified in four categories: mobile threats, cloud threats, mashup threats and technological threats. The main purpose of these menaces is to steal personal data (e.g. credit card numbers, passwords, contact database, calendar, location) or to exploit mobile device resources.

3.2.1.1 Mobile Threats

A little while ago the malware development for mobile devices was seen as a myth due to their limitations in terms of hardware and software. Nowadays, the increasing use and development of mobile devices (e.g. smartphones) has led to the evolution of mobile threats; from the first case of malware on mobile devices in 2004 targeting Symbian, to the code of DroidDream, DroidKungFu and Plankton discovered in 2011 in the official Android Market [Castillo11].

Recent studies [LSM11], [Nachenberg11] have classified mobile attacks in several categories such as: application based attacks, web-based attacks, network based attacks and physical based attacks.

The application based attacks concern both offline and online applications. In these kinds of attacks are included: malware, spyware and privacy threats.

- Malware is software that performs a malicious behaviour on a device without the user being aware of this behaviour (e.g. sending unsolicited messages and increasing the phone's bill or allowing an attacker to have the control over the device).
- Spyware is software designed to collect private data without the user's knowledge (e.g. phone call history, text messages, camera pictures).
- Privacy Threats are caused by applications (malicious or not), that in order to run they need more sensitive data such as location (e.g. location based applications).

The web-based attacks are specific to online application and include: phishing scams, drive-by-downloads, or browser exploits.

- Phishing scams aim stealing information like account login and password.

Security of Mobile Cloud Applications

- Drive-by-Downloads is a technique that allows the automatic download of applications when a user visits a certain web page.

In addition to these attacks, attackers use different techniques to obtain private data: repackaging, misleading disclosure and update.

- Repackaging was the most used technique in 2011 to infect applications running under Android [LSM11]. In this kind of attack, an attacker takes a healthy application; modifies it with a malicious code and then republishes it. The main difference between the healthy and modified applications is that the last ones require more access control permissions such as to access the phone contacts or to send SMS messages.
- Misleading disclosure [LSM11] is a technique used by an attacker to hide the undesirable functionality of an application, so that a user would not notice it and would agree to. The undesirable functionality is usually hidden in the applications terms and conditions. The attackers rely on the fact that usually the users do not pay attention to the applications terms and conditions while these are installed. Those applications are difficult to block or remove because they do not violate their own terms of service or any application market's user agreement.
- The update technique was recently used by malware writers as an attack method in Android Market [Nachenberg11]. Firstly, the malware writer publishes an uninfected application, then the application is updated with a malicious version. Using this technique, the attacker takes advantage of the users trust in the applications market. The number of infected devices increases; there are affected the users that only use the official market to download the applications. A consequence of this attack technique is a decrease of users' confidence in the application market. This may lower the market customers' number and therefore the market profits.

3.2.1.2 Cloud Threats

From a user point of view, the Cloud Computing must answer several questions about data security and privacy, data ownership and location, data access and integrity.

Who can see my data?

Privacy is one of the significant concerns of Mobile Cloud applications. For example, some smart phone applications use the Cloud to store user's data. The main risk in this context is that unauthorized people can access and get user's data in order to use them to get money. Another example concerns location-aware applications such as

Security of Mobile Cloud Applications

applications that finds nearby restaurants for the user; or applications that allows user's friends and family to receive updates regarding her/his location [ZM10].

Who is the owner of "my" data? Where are my data located?

Data Ownership refers to the ownership of purchased digital data. Thanks to the Cloud it is possible to store purchased media files, such as audio, video or e-books remotely rather than locally. This can lead concerns regarding the true ownership of the data. If a user purchases media using a given service and the media itself is stored remotely there is a risk of losing access to the purchased media. The service used could go out of business, for example, or could deny access to the user for some other reasons [TCSA10].

Data location raises many issues because of the compliance problem of privacy laws that are different from a country to another. For example, the laws in European Union (EU) and South America are different from the laws in United States (US) regarding data privacy [CSA09]. Under EU law [EU95] and South American law [AG], personal data can be collected only under strict conditions and for a legitimate purpose. In the US, there is no all-encompassing law regulating the collection and processing of personal data [IDD].

Who can access and modify my data?

Data access and integrity issues are mainly related to security policies provided to the users while accessing the data. If an application relies on remote data storage and Internet access in order to function then, any changes to these data can significantly affect the user. The access to data has to be done by an identified user (application); it also has to be an audit of all the application that accessed those data.

3.2.1.3 Technological Threats

The growth of the mobile computing market has been enhanced by a wide range of enabling technologies: HTML5, Hypervisor, Cloudlets and AJAX. Each of these technologies introduces some security issues and provides opportunities to malicious users to access personal data.

HTML5 [FR11] is an important step for mobile web applications. HTML is a document publishing mark-up language that provides a means of specifying web page elements such as headings, text, tables, lists, and photos. HTML5, last version of HTML, allows specification of offline support, making local storage possible and helping with connectivity interruptions. It also adds video features, enabling graphics and video without plug-ins, and provides a geolocation API. HTML5 can be the target of several

Security of Mobile Cloud Applications

potential threats: Cross-Document Messaging, Local Storage and Attribute Abuse [Eilers12].

AJAX (Asynchronous JavaScript and XML) [Zakas05] is a collection of technologies each providing robust foundations when designing and developing web applications (HTML, CSS, DOM, XML HTTP Request and JavaScript). AJAX is meant to increase interactivity, speed and usability. It is widely used in Mobile Cloud Computing applications to ensure data synchronization. The power of AJAX is seen in many applications including Google Maps and Yahoo!mail. AJAX applications are the target of various attacks like cross-site scripting, cross-site request forgery, JSON Hijacking [CNW07].

A hypervisor allows a web application to run on any smartphone without being aware of the underlying architecture. The security of the entire virtual infrastructure relies on the security of the virtualization management system that controls the hypervisor and allows the operator to start guest OSs, create new guest OS images, and perform other actions [SL12].

A cloudlet is a device that can be found nearby (e.g. coffee shop). When needed, the cloudlet downloads user data from a centralized location, permitting local access by the user and thereby reducing latency. The downloaded and processed data can then be returned to the centralized location, if necessary. This process occurs invisibly to the user, except that the user is pleased with faster response. Trust and security issues are the major factors in cloudlet deployment, because adversaries can create a fake cloudlet to steal user's information [KCK11].

3.2.1.4 Mashup Security Issues

A Mashup [Raman09] is a combination of several components to achieve a new service or a new application. This concept is used for the client/cloud applications. The components can be widgets or services. A widget, that is an innovation of Web 2.0 technology, is software that can display various information and access external resources according to its purpose. Widgets are able to communicate with each other and with the environment.

There are several types of mashups: 1) mobile mashup [Wang10], 2) personalized mobile mashup [BBK11] and 3) mashup as a Service. These categories have been defined based on: 1) the place where the application components are executed and 2) the place where the composition is done (e.g mobile device or Cloud).

In the case of mobile mashups, all the application components run on the mobile device and the composition is also done on the mobile side. For personalized mobile mashups, all the application components run on a middleware in the Cloud and the

Security of Mobile Cloud Applications

composition is also done on the middleware. Mashup as a Service allows the application components to run either on the Cloud side or on the mobile device side. The composition can be done on the mobile device or in Cloud; it depends on the mobile capabilities or context (e.g. private or public networks).

The security attacks against *mobile mashups* are mainly due to communication with a malicious component. A malicious component may: 1) favour phishing and disinformation attacks; 2) perform unauthorized access to the phone services and private data; 3) send and receive data on behalf of another component [BBK11].

In the case of *personalized mobile mashups*, the security attacks may occur on the mobile device, on the cloud or on the communication channels.

As said previously, in a *mashup as a service* an application has components that run on the mobile device and on the Cloud. To communicate with each other, the components have references to the other components. A frequent security issue in this kind of mashups is the modification of the component's reference. In this way data can be transmitted to a malicious component in order to be stolen.

3.2.2 Existing Mobile Cloud Security Solutions

The existing security solutions treat independently the different types of security problems. Some of these solutions are implemented and provided by mobile platforms and Cloud providers to secure mobile devices and communications between mobile devices and Cloud.

In the next sub-sections, we discuss the different approaches proposed to tackle the security issues.

3.2.2.1 Mobile Security Solutions

Mobile devices are constrained with processing and power limitations. Protecting them from security threats is more difficult than protecting a regular computer.

Mobile platform providers (e.g. Android, iOS) implemented several security solutions into the devices operating system. Five types of security features have been implemented in the different platforms: traditional access control, application provenance, encryption, isolation and permission-based access control [LMS11].











Security of Mobile Cloud Applications

- Traditional access control is a technique that uses passwords and idle-time screen locking to protect the mobile device.
- Application Provenance refers to the fact that each application has to be labelled with the identity of its author and also signed with a digital signature.
- Encryption is used to protect and hide the data saved on the mobile device in case of loss or theft.
- Isolation is a technique that separates each application on the device to deny access to other applications data.
- Permission-based access control is the mobile device capacity to provide an application with a certain level of access control to the device data and system. This access control is established when the application is deployed according to the application provider. This task can be entrusted to the end user.





Android and iOS (see Table 3.2) tried to make the platforms secure rather than to force the users to rely upon third-party security software. While iOS offers four of the five security solutions (traditional access control, application provenance, encryption and isolation), Android offers only three of them (traditional access control, isolation and permission-based access control) [LMS11].

Android platform is less rigorous than iOS regarding the applications provenance and consequently less secure. This weakness has been exploited by attackers in 2010 and 2011 who replaced some of the legitimate applications with applications that contained malicious code.

TABLE 3.2 MOBILE SECURITY SOLUTIONS [LMS11]

Security Feature	Apple iOS	Google Android
Access Control		
Application Provenance		
Encryption		
Isolation		
Permission-based Access Control		

Security of Mobile Cloud Applications

-  full protection
-  moderate protection
-  greater protection
-  less protection

3.2.2.2 Mobile Cloud Communications Security Solutions

The mobile cloud application providers have to secure the data exchanged between the mobile devices and the Cloud. The most used security protocol for securing data transmission between the mobile device and the Cloud is SSL/HTTPS [HZX11]. However, this protocol is on one hand high energy consuming [KBR+11] and on a second hand provides security properties (integrity, confidentiality and authenticity) as a block without taking into account the type of data transmitted or the user expectations.

To optimize the energy consumption, a security components-based architecture called LECCSAM (A low-energy consuming and user-centric security management architecture adapted to mobile environments) has been designed for securing communications between two mobile devices or between a mobile device and a server [KBR+11]. LECCSAM aims to optimize the mobile device energy consumption and consider the user's constraints and options regarding the security level applied to the transmitted data. LECCSAM is based on the principle of security properties separation. Each of these security properties is designed and implemented under the form of independent components. Thus, each property can be applied separately only when it is required and only to data whose security level demands it. Even if LECCSAM brings these advantages, it is not adapted to the mobile cloud applications.

3.2.2.3 Mobile Cloud Security Solutions

To protect mobile cloud environments against security attacks, three categories of security solutions have been proposed: solutions for data security, solutions for applications security and solutions for privacy. These solutions are presented in the following sub-sections.

Security of Mobile Cloud Applications

Data security solutions

To ensure the confidentiality and integrity of the user's data stored in Cloud, several solutions have been proposed.

Itani et al. [IKC10] proposed a solution to ensure the integrity of files stored in Cloud. Furthermore this proposed solution seeks to be an energy efficient framework for mobile devices. The concepts used to design it are: incremental cryptography [BGG94] and trust computing [BGG95]. The main entities defined are: 1) mobile client, 2) cloud service provider, and 3) trust third party. The mobile client uses the cloud service provider storage services. The trust third party holds the coprocessors responsible for Secret Key distribution to mobile client and message authentication code generation on behalf of mobile client. The authors have discussed the following operations: uploading, block insertion, block deletion and integrity verification for files in the Mobile Cloud Computing environment. When uploading files in Cloud, the mobile client generates an incremental message authentication code (MAC_f) using the Secret Key and store it. When mobile client wants to perform insert, delete or update operations on the uploaded files the following steps are performed: 1) the mobile client request a file; 2) the cloud service sends the file to the mobile client and also to the trusted third party; 3) the trusted third party reconstructs the message authentication code (MAC_{cop}) then it sends it to the mobile client; 4) the received MAC_{cop} is compared with MAC_f, if they are the same the integrity is checked. After the integrity check, the mobile client can perform insert, delete or update operations, then performs a new calculation of MAC_f.

In [LLL+12], the authors propose three schemes to ensure the confidentiality and integrity of the users' files stored on the Cloud considering three assumptions: 1) the mobile device is semi-trusted, 2) cloud servers are distrusted; and 3) the communication channel is secured. The files are created and modified on the mobile device and stored on the Cloud servers. In each scheme, the mobile device is responsible for encryption, decryption, and integrity verification.

The first scheme is called 'Encryption based scheme'. When the user wants to upload a file from a mobile device to a cloud server he has to provide a password (PWD). Then the mobile device performs the following steps: 1) it generates the Encryption Key (EK) and the Integrity Key by using a Hash function (H) on concatenation of File Name (FN), File Size (FS) and password (PWD); 2) it encrypts the file using EK in order to obtain the confidentiality; 3) it generates the message authentication code (MAC) using the file and the IK for authentication; 4) it uploads on the cloud server the encrypted file (EF), the hash of the file and the MAC; 5) it deletes the IK and the EK; 6) it stores the FN in the local file table.

Security of Mobile Cloud Applications

While downloading a file, mobile device computes the hash of the FN and transfer the hash value to the cloud server. The cloud server looks for corresponding EF and MAC with the help of the received data. If a file is found, the cloud server sends EF and MAC to the mobile device. The mobile device performs the following steps: 1) it prompts the user to enter a password for the file; 2) it regenerates the EK and the IK; 3) it decrypts the EF; 4) it regenerates the MAC and 5) it verifies the file integrity.

The second scheme is called 'Coding based scheme'. It was developed in order to reduce the computation overhead of the encryption operation. The solution was to eliminate the encryption operation. Thereby, when uploading a file on a cloud server the following steps are performed: 1) the file to protect is split in d parts of t chunks and each chunk has n bits; 2) for each chunk it is generated a coding vector (α) by applying recursive the hash function on the concatenation of PWD, FN and FS; 3) then the IK is computed by applying a hash function on the concatenation of each generated α ; 4) the α is used to produce a Secrecy Code (SC) for each part of the file; and finally the MAC is generated. On the server the mobile device uploads the SC of each chunk along with the MAC and the hash of the concatenation between FN and the chunk number.

While downloading, the mobile regenerates the hash; with this hash the server can provide the SC and MAC. The mobile device prompt the user to enter a PWD for the file in order to reproduce the α and the IK. The original file is decoded by multiplying SC with the inverse of α .

The third scheme is called 'Sharing based scheme'. This scheme introduces on exclusive or operation. As a result, it is argued that this scheme requires less computation power on the device side, which leads to the mobile device energy saving.

Another solution is proposed by [QMS+12] to ensure the privacy, the integrity and the confidentiality of the user's data stored in the Cloud. This solution makes two assumptions: 1) the communication channel is secure; and 2) the third party is trusted. The proposed architecture is responsible for handling encoding/decoding, encryption/decryption, signature generation, and integrity verification on behalf of the mobile user. It contains three main entities: the mobile end-user, a trusted third party, and a Cloud storage service.

A framework to secure storage services in mobile cloud computing is proposed in [HLL11]. This framework has four modules: 1) a mobile device that uses cloud services, 2) a cloud service provider, 3) a certification authority that authenticates the mobile devices and 4) a telecommunication module that generates and keeps track of mobile device passwords. In this framework, the secret key, public key and session key are securely distributed. To use the cloud services, the mobile user, has to register with the telecommunication module through the certification authority. On successful

Security of Mobile Cloud Applications

registration, the telecommunication module issues a password (PWD) for mobile device to use cloud resources. In order to a secure delivery of PWD to the mobile device, the telecommunication module encrypts it with the mobile devices public key. Before uploading a file, the mobile device encrypts it with its secret key. When downloading a file, the mobile device needs first to authenticate; on success cloud sends the encrypted file to the mobile device along with a signature. The mobile verifies first the signature and then decrypts the file.

In Table 3.3, presents an overview of the security features provided by each data security solution.

Applications security solutions

Zhang et al. [ZSG+09] designed a solution to solve the security issues of an elastic mobile cloud application. An elastic application consists of one or more weblets. The weblets work independently on the mobile device or in the Cloud and communicate with each other to perform the application tasks. They can migrate between the mobile device and the Cloud according to the changes on the mobile device. The main components of this application model are: Device Elasticity Manager (DEM) and Cloud Elasticity Service (CES).

TABLE 3.3 OVERVIEW OF THE SECURITY FEATURES PROVIDED BY DATA SECURITY SOLUTIONS

Solution	Security properties	Operations performed by mobile device	Energy saving considerations
[IKC10]	Integrity of files stored in Cloud	MAC generation; MACs comparison	yes
[LLL+12] Encryption based scheme	Confidentiality and Integrity of files stored in Cloud	EK, IK, MAC generation; Encrypt, decrypt files	no
[LLL+12] Coding based scheme	Confidentiality and Integrity of files stored in Cloud	IK generation; Additional computation	no
[LLL+12] Sharing based scheme	Confidentiality and Integrity of files stored in Cloud	exclusive or computation	yes
[QMS+12]	Privacy, Integrity and Confidentiality of users data in Cloud	No operations	yes
[HLL11]	Secure storage services	Encrypt, decrypt files	no

Security of Mobile Cloud Applications

The Device Elasticity Manager configures the application at launch time (e.g. it decides where a weblet should be launched, in Cloud or on the mobile device) and makes configuration adjustments at runtime. The Cloud Elasticity Manager ensures the execution resources for the weblets. The proposed security solution: 1) ensures the secure installation of the elastic application, 2) manages weblets authentication, 3) secures communication between weblets running concurrently on mobile device and in the Cloud nodes, 4) secures the weblets migration between the mobile device and the Cloud and 5) ensures weblets authorization to access user data.

A solution, called TrustCube, is proposed in [SML+09] to secure data access. This solution uses the authentication property to provide or deny the access of a mobile client to a web server. The authentication is made through two services: an Integrated Authentication Service and an Implicit Authentication Service. The Integrated Authentication Service receives the access requests from the web server, extracts the necessary information and sends it through a secured channel to the Implicit Authentication Service. The Implicit Authentication Service generates a report, and sends it back to the Integrated Authentication Service. After receiving the report, the Implicit Authentication Service determines whether or not the mobile client is authenticated successfully and sends the result to the web server. Based on the authentication result, the web server either provides the service or denies it. To authenticate mobile users the authors propose to use short passwords or PINs, which is not enough to ensure a high security level.

Huan et al. [HZK+10] proposed a new mobile cloud computing framework, called MobiCloud that provides conventional computation services and improves the functionality of the Mobile Adhoc Network (MANETS) in terms of risk management, trust management and secure routing. MobiCloud provides several functionalities to manage security, risks assessment, location-based services, network and status monitoring, and context aware routing. In addition, MobiCloud can 1) extend and augment the functionality of MANETs, and 2) predict the future MANET situations for decisions making by using historical data.

The MobiCloud architecture defined in [HZK+10] does not consider the privacy and security of users' data stored in the cloud. An improved version of MobiCloud is proposed in [HZX11] where a secure data processing model is added. This model consists of three domains: 1) cloud public service and storage domain, 2) cloud trusted domain and 3) cloud mobile and sensing domain. The authors assume that a trusted authority is always available to control the key distribution and manage certificate distribution and user identity.

Security of Mobile Cloud Applications

Privacy solutions

Users need to be aware of what personal information is exactly visible to the public and to have control over the personal information stored on their mobile devices. It is very important that any private data is shared only with the users consent.

A solution for ensuring privacy regarding location is proposed in [ZM10]. In this solution is defined a Location Trusted Server that uses the location cloaking to submit data (make data submitted either spatially or temporally imprecise) and send them to a Location Based Service. The Location Based Services knows only general information about the users but cannot identify them. The assumption made by the authors is that the cloud service provider is semi-trusted. The performance of the proposed solution has been evaluated with a traditional cloaking mechanism called Casper [MCA06] and proved to be efficient.

3.3 Conclusions

Mobile Cloud Computing is a model that can be described as the availability of Cloud Computing resources to mobile environments. In order to benefit as much as possible from the advantages offered by the Cloud, several mobile cloud applications models have been proposed. The most important feature of the mobile cloud applications models is that they are distributed between the mobile device and the Cloud. They have the user interface running on the mobile device, the business logics may be spitted between the mobile device and the Cloud and the databases stored in the Cloud or on the mobile device. Besides this application model, several new models were proposed; these displace the entire application execution from the mobile device to the Cloud, or run into the Cloud only a curtain part of the applications code.

From a security point of view, mobile cloud computing introduces many security issues due to the fact that it combines mobile devices with Cloud services and also because there were added several new application models. To tackle these issues, several solutions have been proposed and some of them have been implemented and provided by mobile and Cloud providers.

In order to secure users data and applications, the mobile platform providers implemented various strategies. But these strategies were not applied in the same way by all the providers. Furthermore, the application of these strategies will allow securing

Security of Mobile Cloud Applications

data on the mobile device, however, when the data will be sent and stored in the Cloud, it will become out of the user control.

The private data in mobile cloud applications have to be secured while they are exchanged between the mobiles and the Cloud. The solutions already used provide data security but increase the energy consumption of mobile devices.

To protect the data stored in the Cloud several solutions were proposed. Their purpose was to ensure the following security properties: integrity, the confidentiality and the privacy for data or files stored on the cloud servers. Several of these solutions do not take into account the mobile energy constraint; and when the energy constraint is considered, the number of security properties provided decreases. Furthermore, in one solution there was made the assumption that the communication channels are secured.

In the case of mobile cloud applications, there were proposed solutions customized to a certain type of application like in the case of elastic applications, or in the case of MobiCloud framework. Even if these solutions are innovative, they are particular to a specific type of applications.

3.3.1 Contributions

A state of the art on Mobile Cloud Computing was made in order to point out: 1) what is Mobile Cloud Computing, 2) which are Mobile Cloud Computing main features, 3) the novel mobile cloud applications models, 4) which are the security issues and the existing solution for mobile cloud applications [PBC+13g]. The aim of this state of the art was to give a research direction for this thesis [PBC+13a].

3.3.2 Publications

[PBC+13a] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "A Security Framework for Mobile Cloud Applications "Networking in Education and Research", Sinaia, Romania, ARNIEC/RoEduNet Agency, IEEE Romanian Section, ISSN-L 2068-1038, 17-19 January, 2013.

[PBC+13g] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "Overview on Mobile Cloud Computing Security Issues", in Scientific Bulletin of the Politechnica University of Timisoara, Transactions on Electronics and Communications ,2013 (Submitted - accepted for publication).

Security of Mobile Cloud Applications

Security of Mobile Cloud Applications

“Mobile technology... can expose users and organizations valuable data to unauthorized people if necessary precautions are not taken.”

Security in the age of mobility

4. Proposed Secure Mobile-Cloud Framework

Contents in Brief

4.1 Criteria and Objectives	50
4.2 The Framework Basis.....	55
4.3 The Framework Design.....	58
4.4 Conclusions	71

Chapter Overview

This chapter presents the major contribution of this thesis: to propose a framework to secure the data used by a mobile cloud application.

The general structure of this chapter is as following: 1) the general criteria that should be taken into consideration when designing a security solution for a mobile cloud application (Section 4.1), 2) the proposed framework objectives (Section 4.1), 3) the basic notions related to the framework design (Section 4.2), 4) the framework design (Section 4.3) and at the end the conclusions (Section 4.4).

4.1 Criteria and Objectives

This work focuses on components based mobile cloud applications (Figure 4.1). The term component defines a well define part of an application developed in order to fulfill a certain action. The user interface can be seen as an application component, the entire business logic also can be seen as an application component, the application business logic may be spitted in little parts each part can represent a component. The components can have different execution locations (e.g. mobile device or Cloud).

From the Chapter 3, it can be seen that the security issues related to Mobile Cloud Computing are various. The solutions proposed to resolve the mobile cloud security issues treat independently each type of security problems. Thereby, in the case of components base mobile cloud applications, is needed to combine different solutions in order to secure them. Also it can be seen that very few solution proposed to secure the data take into account the mobile device energy constraints; furthermore the security solutions proposed to secure the communications between the mobile device and the Cloud are energy consuming solution.

The lack of resources is one of the biggest disadvantages of mobile phones. Therefore, a security solution for mobile cloud applications must consider mobile device constraints. Furthermore, the mobile owners (mobile end-users), running mobile applications may have different expectations regarding the security level of their private data. The existing solutions for mobile cloud applications do not allow the users to express their opinion regarding the security level of their private data.

Several criteria that should be taken into account when designing a security solution for mobile cloud applications and more generally for today mobile applications are presented in the following subsection.

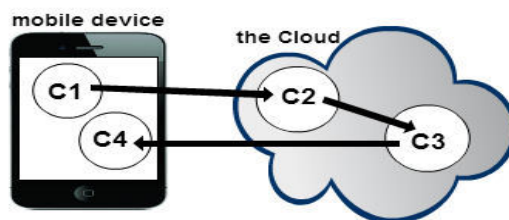


Figure 4.1 Component-based applications

Security of Mobile Cloud Applications

4.1.1 Criteria

4.1.1.1 Human constraints

Each person values its private data. However, the degree of importance for private data may vary from a person to another person. This degree can differ according to each person perceptions regarding privacy and also according to each person status (citizen, politician, actor, government employee etc.). For example: 1) a person may not consider to be private data the information on her/his birthday while for another this information is private and 2) a politician may need a higher security level for her/his data than a non-politician person.

For a well-known person (e.g. an actress or an actor) discovering even the smallest details about her/him, it may cause her/him many damages. For example: someone (a person) wants to use a mobile cloud e-health application that monitors the individual weight; or/and it monitors the types of products an individual eats each day. If that person is an actress, the loss and the publication of her data related to its weight may cause her many damages in terms of her image. If that person is not public (e.g. the author of this work) the loss and the publication of her data related to its weight will cause her very little prejudice or would not cause her any prejudice at all.

Users' requirements are not taken into consideration by various traditional security solutions. Actually, most of the time, these solutions, do not fit with users expectations. Even if this was more or less acceptable until now, today it is an important issue that cannot be ignored as end-users are more and more concerned about security of their private data. Thus, a security solution must allow an end-user to express her/his needs regarding the security level of her/his data and more generally of the applications she/he uses and run on her/his mobile.

Furthermore, the level of knowledge regarding data security differs from one person to another. A security solution has to be able to provide for each user type (non-security expert or security expert) a way to express her/his options regarding the level of security applied to her/his data. The security solution must offer very good results both if is used by a security expert user or a non-security expert user.

4.1.1.2 Data sensitivity constraints

Some information requires special care and handling, especially when inappropriate handling of the information could result in penalties, identity theft,

Security of Mobile Cloud Applications

financial loss, invasion of privacy, or unauthorized access by an individual or many individuals.

To each kind of data is assigned a level of sensitivity based on the application scenario that will use this data and also the user preferences and/or context. Moreover, in my opinion data are even more sensitive if their loss or theft causes important damages regarding the income and integrity of a user. For example the password data of a user or her/his bank account is more sensitive than her/his preferred colour.

Each sensitivity level requires an adequate security level, where each security level implies providing the right security properties (e.g. integrity, confidentiality, authenticity, etc.). Thus in a mobile cloud application, data transmitted between components may have different security levels and require different security properties (as shown in Figure 4.6).

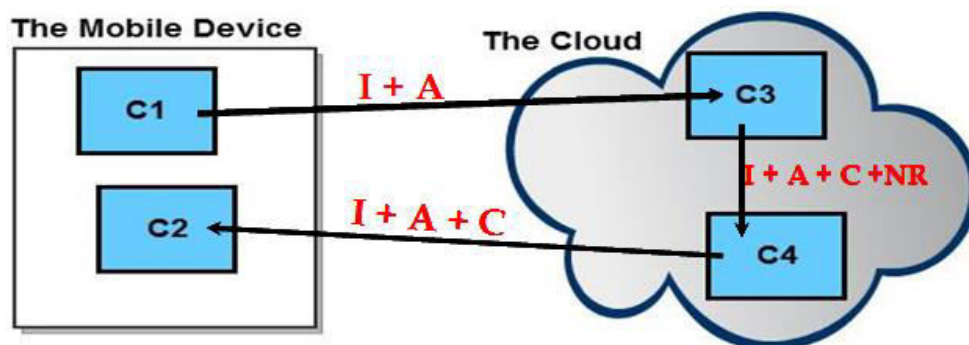


Figure 4.2 Different security levels

4.1.1.3 Technical constraints

The technical constraints regarding mobile cloud applications concern components location, user context and mobile device capabilities.

In a component-based mobile cloud application, each component runs in a specific location. It can be on the mobile device or in the Cloud. Moreover, some components can migrate between the mobile device and the Cloud. When a component changes a location, the security level may also change. It should be strengthened, if the components migrate from the mobile device to the Cloud, or it may be weakened if the migration is from the Cloud to the mobile device. Actually, these changes depend on the components that communicates between each other and on their location. For example: two components, running on the mobile device and exchanging data between them, may need a low security level (e.g. to ensure only data integrity) in order to secure the data exchanged between them. When one of them change its execution

Security of Mobile Cloud Applications

location (migrates to Cloud), the security level applied to the data exchanged between should be strengthened (e.g. to add the confidentiality property).

Concerning the user context, we refer to the area where the end-user is when executing an application: private or public area (e.g. home, office, public space such as airport, commercial centre, etc.). The user context may change very often, particularly in the future, which will influence considerably the security level applied to the data transmitted between the components of a mobile cloud application. The user context can be also classified as a human constraint because the user is the one who influence the device location. But it was chosen for this constraint to be classified as a technical constraint because at the end the security solution provided depends on the location (public or private) and not on the user options.

Regarding mobile devices constraints, for devices like mobile phones, with limited resources and energy, it is important to provide security solutions that consume fewer resources without compromising and reducing the security level of the data to secure.

The energy consumed by the cryptographic algorithms (encryption, decryption, hash functions, etc.) used to traditionally secure data (in transit and data at rest) depends on the algorithm type. Asymmetric ciphering algorithms consume more resources than symmetric ones, which in their turns are more consuming than hash functions. According to [NWD05] public key consumes more energy and also it is more expensive as compared to symmetric key. Also, it was stated in [NWD05] that public key is used in some applications for secure communications (e.g. Secure Socket Layer). In [GC01] it is state that public key consumes more energy due to great deal of computation and processing involved (e.g. a single public key operation can consume same amount of time and energy as encrypting tens of megabits using a secret key cipher).

Also, for the symmetric cipher algorithms the performances (e.g. computing resources such as CPU time, memory, or battery power) may vary depending on the type. It was presented in [EKH10] a performance evaluation of several symmetric encryption algorithms: AES, DES, Triple DES, RC6, Blowfish and RC2. It was concluded that: 1) AES has better performance than RC2, DES, and 3DES; 2) a higher key size leads to clear change in the battery and time consumption; 3) 3DES still has low performance compared to DES; and 4) Blowfish has better performance than the other common encryption algorithms used, followed by RC6. In [Hirani08] it was concluded that AES is faster and more efficient than other encryption algorithms (e.g. 3DES, DES). Also it was stated that the most of the resources are consumed for data transmission rather than symmetric key schemes computation. In [Nadeem06] were compared the

Security of Mobile Cloud Applications

following secret key algorithms: DES, 3DES, AES, and Blowfish. It was concluded that Blowfish had a very good performance compared to other algorithms. Also it showed that AES had a better performance than 3DES and DES.

4.1.2 Objectives

As mentioned above this work focuses on component based mobile cloud applications. The components have different execution locations (e.g. mobile device or Cloud). In this work it is considered that the component location during the execution is fixed (the components do not migrate). Furthermore, from the constraints presented above there were considered the following:

- the human constraints: 1) users options regarding the security level applied to data; and 2) users type (security expert or non-expert);
- the data sensitivity constraints
- the technical constraints: 1) mobile device energy consumption;

The main goal of this work was to propose a security framework in order to secure the users private data used by a component based mobile cloud application. The security framework proposed is called Secure Mobile-Cloud (SMC) and its functionalities are:

- 1) To allow the users to choose the security level they want to apply to their data.
- 2) To adapt the security level and therefore the security services to the user requirements.
- 3) To adapt the security services applied to the mobile device energy consumption.
- 4) To secure the communication between the same application components (i.e. between components running on the mobile side and those running in Cloud and between the components running only in Cloud).

4.2 The Framework Basis

In order to adapt the security solution to the various concerns (e.g. user options, device energy) the proposed security framework, SMC, is based on the principle of security properties separation. The term of security properties denote the following properties: integrity, authenticity, confidentiality and non-repudiation.

To enforce this principle, it is necessary for each security property to be designed and implemented as an independent component. A solution to this need has been proposed before in [Nobelis08], its name is LECCSAM. Thereby, the Secure Mobile-Cloud framework uses the security components provided by LECCSAM [KBR+11]. The security components are an assembly of cryptographic tools satisfying each a security property (e.g. integrity, confidentiality, authenticity, non-repudiation, access control). The main objectives of LECCSAM are: 1) Securing the transfer of data between two mobile devices or a mobile device and a server by applying the required security properties (security properties being chosen according to the sensitivity of the data to be transmitted); 2) to optimize the mobile device energy consumption by moving the security components execution out from the mobile device.

In the following sub-section, we will describe four of the fifth LECCSAM security components used by our framework: integrity, authenticity, confidentiality and non-repudiation.

Before describing the different security components, we present briefly the standard notation used: 1) to express the cryptographic elements used; and 2) to describe the cryptographic constructions.

D - users' data needed to be secured;

K - the secret key used for symmetric encryption;

PK/SK - the public and the secret keys used for asymmetric encryption;

{D}K - users' data encrypted with the secret key

{D}PK - users' data encrypted with the public key

{D}SK - users' data encrypted with the secret key

D | D' - the concatenation of two messages

h(D) - the hash of a message

4.2.1 The Integrity Component

Integrity is the security property, that checks whether a sent/received message (or document) was modified or not. The modification may be intentional or unintentional (e.g. transmission error). The scheme that ensures the integrity was proposed in [KBR+11] and is presented in Figure 4.2.

The Integrity component is designed by using two functions: a hash function providing the security operation and a concatenation function. The component receives as an input the raw data (D); and returns as result the concatenation between the received data and its hash.

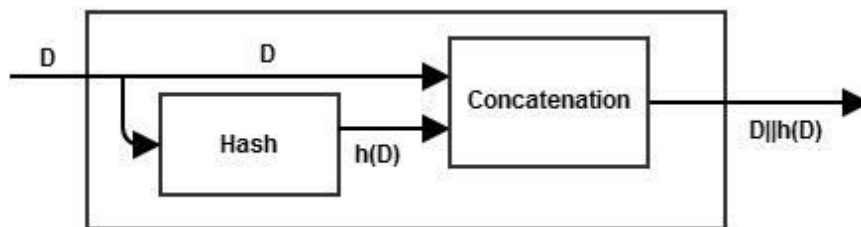


Figure 4.3 Integrity Component [KBR+11]

4.2.2 The Authenticity Component

Authenticity is the security property which helps: 1) to identify the author of a message (the author authenticity); or 2) to identify if a message/document is authentic (the message authenticity). The scheme that ensures the authenticity was proposed in [KBR+11] and is presented in Figure 4.3.

The authenticity component has the following functions: a hash function, a concatenation function and an asymmetric encryption algorithm. The component receives as an input the raw data and the secret key used for the asymmetric encryption. The core functionality is made in several steps; 1) It computes the hash of the input data; 2) the computed hash is encrypted using an asymmetric algorithm; and 3) the encrypted hash is concatenated to the input data and returned as the output value.

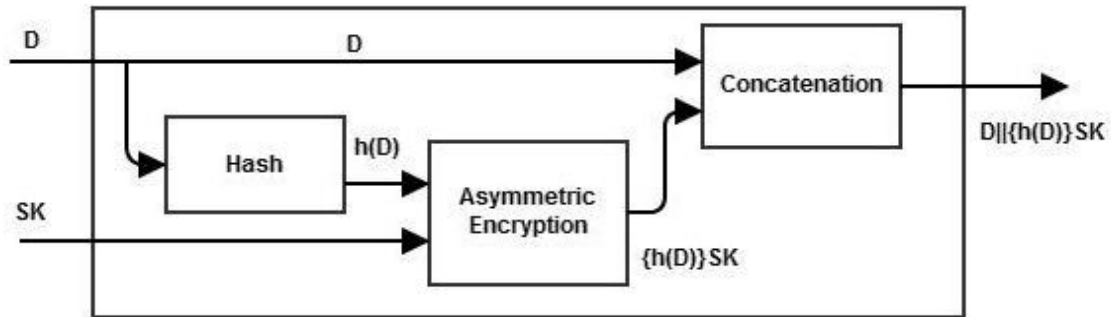


Figure 4.4 Authenticity Component [KBR+11]

4.2.3 The Confidentiality Component

Confidentiality is the security property which guarantees that data sent by a certain sender can be read only by their addressee. It transforms the plain text data in ciphered data. The scheme that ensures the confidentiality was proposed in [KBR+11] and is presented in Figure 4.4.

The confidentiality component uses: a symmetric and an asymmetric cipher to which is added the concatenation operation, more specifically: 1) the component receives as an input the raw data and the public key used for the asymmetric encryption; 2) the secret key is generated; 3) the generated key is used to encrypt the input data using a symmetric encryption algorithm; 4) the generated key is also encrypted using an asymmetric algorithm; and 5) the results of both encryptions are concatenated and returned as the output parameter.

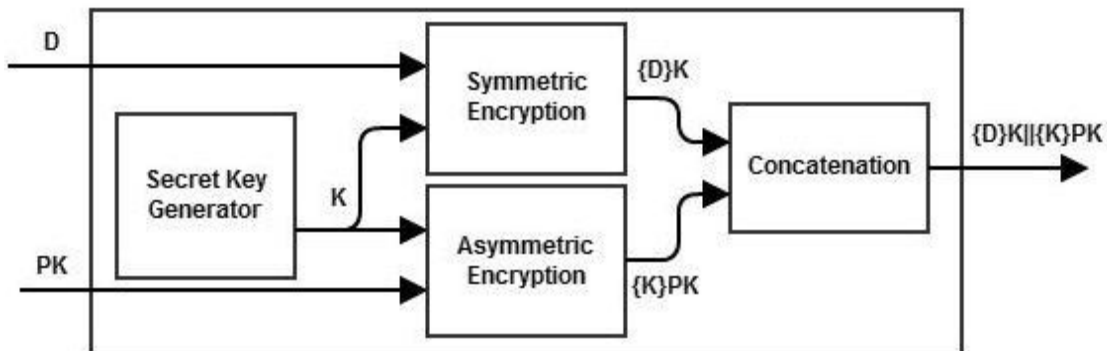


Figure 4.5 Confidentiality Component [KBR+11]

4.2.4 The Non-Repudiation Component

Non-Repudiation is the security property that aims to provide to the sender of the data a proof confirming the reception by the receiver of the sent data. The scheme that ensures the non-repudiation property was proposed in [KBR+11] and is presented in Figure 4.5.

The non-repudiation component uses: a symmetric and an asymmetric cipher to which is added a concatenation operation, more precisely: 1) a secret key K is randomly generated along with a nonce N ; 2) data D is encrypted using the private key; 3) the private key is concatenated to the nonce; 3) concatenation $K || N$ is encrypted using a asymmetric algorithm; 4) the results of both encryptions are concatenated ($\{D\}_K || \{K || N\}_{PK}$) and sent to the intended receiver .

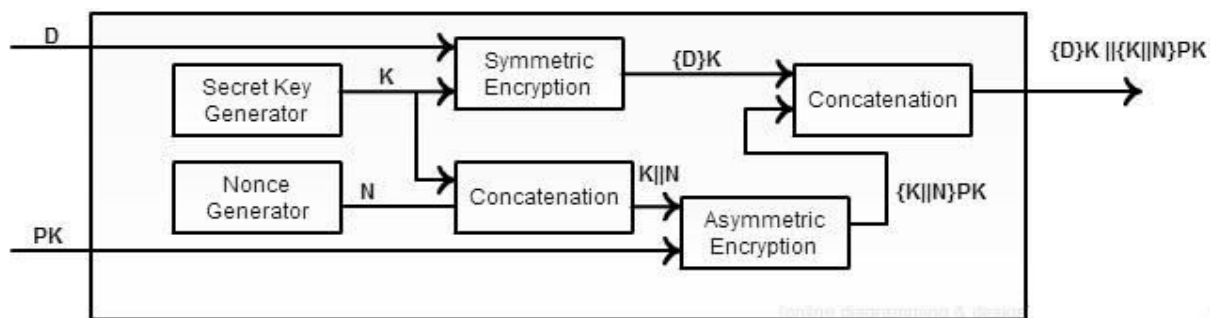


Figure 4.6 Non-Repudiation Component[KBR+11]

4.3 The Framework Design

In this section, it is present the Secure Mobile-Cloud (SMC) Framework.

The SMC Framework is composed of two types of components, as it can be seen in Figure 4.7: 1) security components (LECCSAM components) and 2) management components.

As said in section 4.2, the security components have been designed to implement the eponym security properties. These security components are deployed in both mobile device and Cloud. The join of the security components defines a security combination (example Table 4.1). Also in this work the simple security combination term indicates only one security property.

Security of Mobile Cloud Applications

TABLE 4.1 EXAMPLE OF SECURITY COMBINATIONS

Combinations	Examples
Simple Security Combination	Integrity, Authenticity, Confidentiality, Non-Repudiation
Security Combination	Integrity + Non-Repudiation, Authenticity + Non-Repudiation Integrity + Confidentiality, Authenticity + Confidentiality Integrity + Confidentiality + Non-Repudiation Authenticity + Confidentiality + Non-Repudiation

The management components have been designed to identify and apply the appropriate security properties and therefore security components to user's data. Some of these management components are deployed on the mobile device and some of them are deployed in the Cloud. These managers have been divided in three groups:

1) Security managers

- a. *Mobile Security Manager*: ensures the orchestration of the security components in the mobile device.
- b. *Cloud Security Manager*: ensures the orchestration of the security properties on the Cloud side.

2) Auxiliary managers:

- a. *State Manager*. It sends the information collected from the sensors (e.g. energy sensor) to the mobile manager. It is deployed on the mobile.
- b. *Policy Manager*. It determines which security components are required for a specific security level. It is deployed on both mobile device and Cloud.

3) Mobile Manager:

- a. *Mobile Manager*. It collects the users' data and the events that occurs on the mobile side and sends them to the appropriate manager (e.g. Cloud Security Manager or Mobile Security Manager) in order to be secured. For example, in a scenario application (e.g. healthcare application), if a component on the mobile device has to send data to another component in the Cloud, the Mobile Manager collects these data and sends them to the Mobile Security Manager or to the Cloud Security Manager in order to be secured.

Security of Mobile Cloud Applications

The role of these and their interactions are described in detail in the following sub-sections.

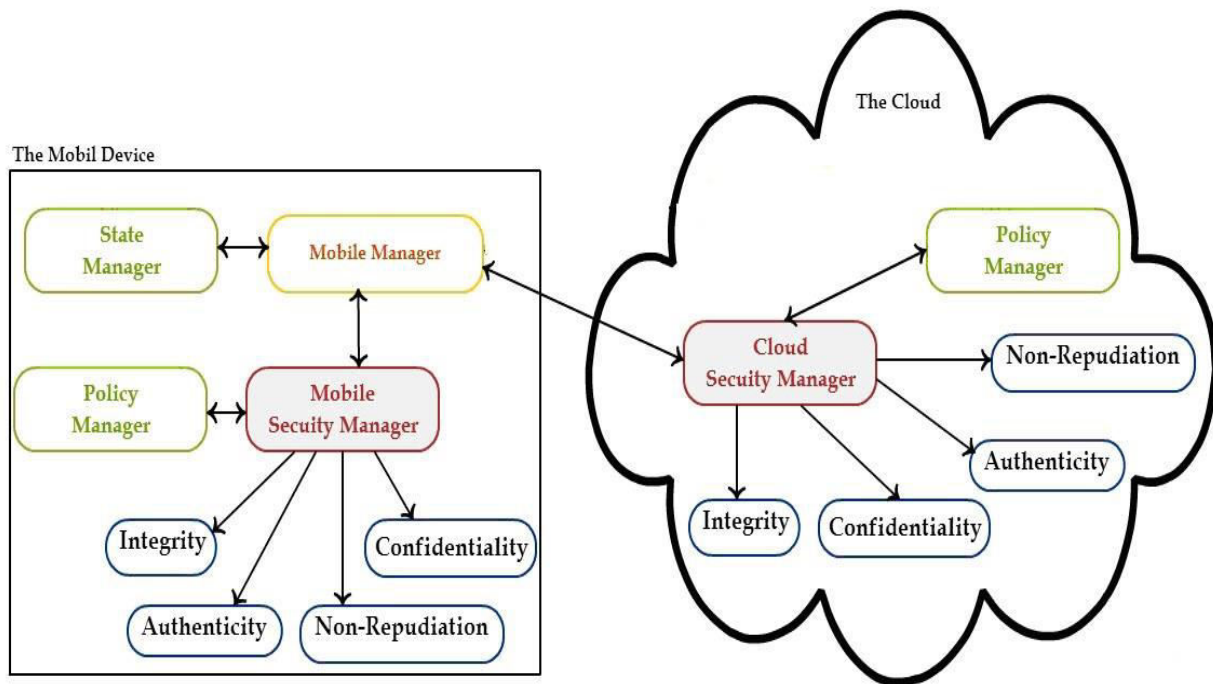


Figure 4.7 The Secure Mobile-Cloud Framework

4.3.1 The Security Managers

In the proposed framework, the security of the data transmitted between the mobile device and the Cloud is ensured at both side mobile device and Cloud (see Figure 4.8).

As it is shown in Figure 4.8, the security management of the data transmitted between the different components of the same mobile cloud application is done by the following managers: Mobile Security Manager and Cloud Security Manager. These managers receive parameters like: the security level needed and the data to secure. Each manager interacts with its Policy Manager to determine the security properties to apply according to the security level required. When the data is transmitted between the mobile device and the Cloud, the Mobile and Cloud Security Managers receive this data through the Mobile Manager. In the Cloud, the application components communicate directly with the Cloud Security Manager.

Security of Mobile Cloud Applications

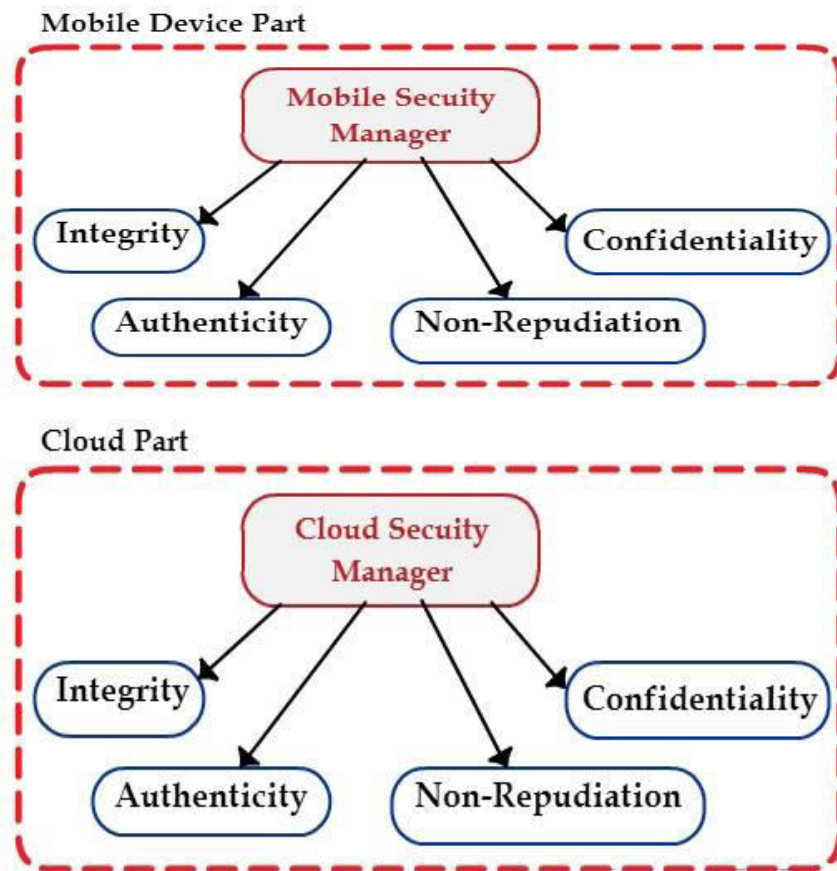


Figure 4.8 The Security Parts

Data security is ensured, by both Mobile and Cloud Security Managers, through the following steps (see Figure 4.9):

- Step1 (S1) - The Mobile/Cloud Security Manager receives data and the security level (SL). Data can be in clear (D) or encrypted $E\{D\}$.
- Step2 (S2) - The security level (SL) is sent to the Policy Manager. This latter sends the proper combination (C) of the security properties back to the Mobile/Cloud Security Manager.
- Step3 (S3) - The Mobile/Cloud Security Manager applies the needed security properties $SW(C)$ to the received data.
- Step4 (S4) - The result is sent back to the Mobile Manager.

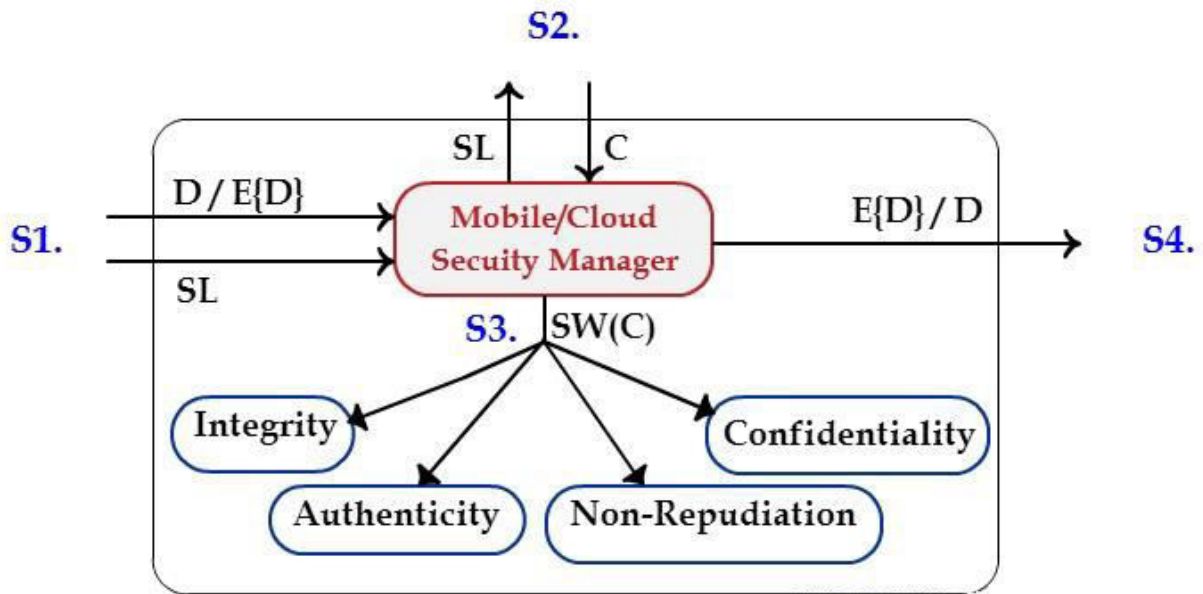


Figure 4.9 The Steps to secure data

4.3.2 The Auxiliary Managers

The role of the auxiliary managers, i.e. the Policy Managers and the State Manager is to communicate with the Security Managers or Mobile Manager in order to provide helpful additional information (see Figure 4.10) like: 1) the security properties combination corresponding to a certain security level, in the case of the Policy Manager or 2) the battery level of the mobile device, in the case of State Manager.

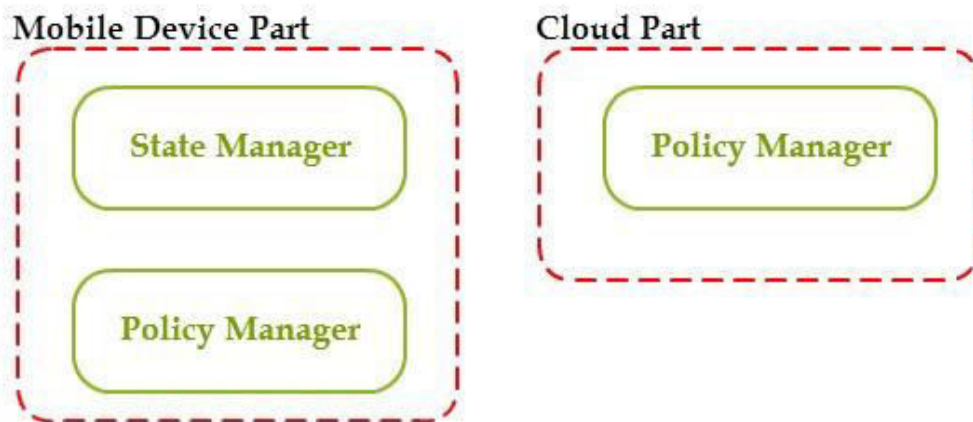


Figure 4.10 The Auxiliary Managers

Security of Mobile Cloud Applications

The Policy Managers keep the security composition rules. These rules define the security properties (components) combination specific to a certain level of security. The Mobile/Cloud Security Manger communicates with its Policy Manager in order to obtain the security properties combination corresponding to the required security level sent by the Mobile Security Manager. For example, as seen in Figure 4.11 the Mobile/Cloud Security level (SL = C8). The Policy Manager will return the list of the adequate security properties for the required security level: Authenticity (A) and Confidentiality (C).

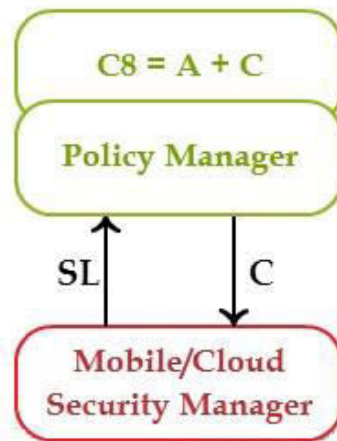


Figure 4.11 Communication Policy Manager and Mobile/Cloud Security Manager

The State Manager (Figure 4.12) monitors the battery level. It communicates with the Mobile Manager and sends to this latter the battery status whenever it is needed.

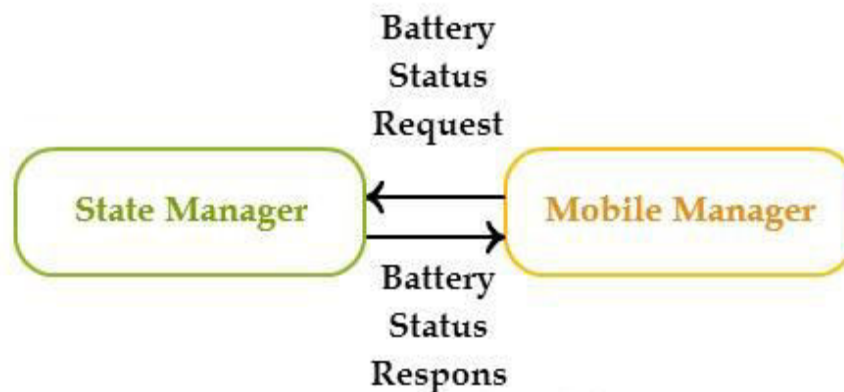


Figure 4.12 The communication between State Manager and Mobile Manager

4.3.3 Mobile Manager

The Mobile Manager (Figure 4.13) has several functionalities:

- 1) It captures the application data transmitted from the mobile device to the Cloud and vice versa. In addition, it sends the intercepted data to the appropriate managers (e.g. Mobile Security Manager or Cloud Security Manager) in order to secure it;
- 2) It operates also as an analysis system.

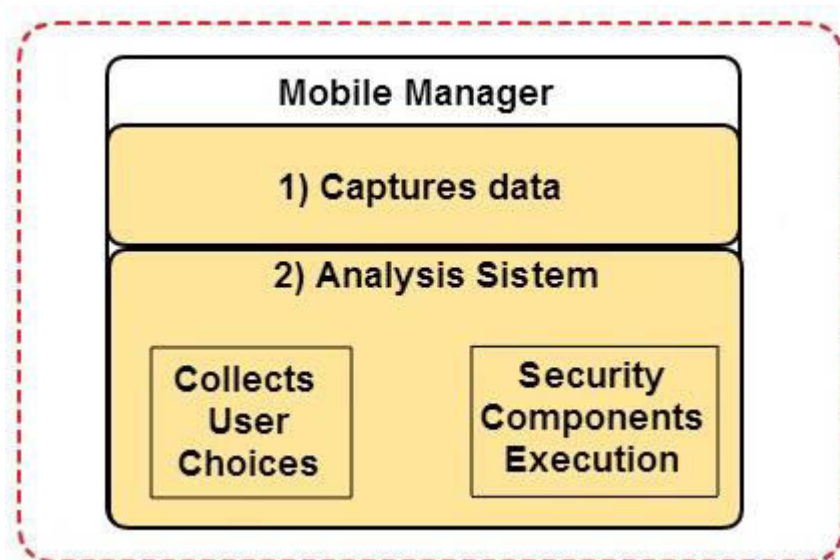


Figure 4.13 The Mobile Manager

Mobile Manager - Analysis functionality

The security solution provided by the SMC framework depends on the data sensibility constraints, on the users' choice regarding the security level applied to his/her private data and on the device capabilities (e.g. energy).

In order to provide a solution that allows achieving this characteristic, a new functionality was added for the mobile manager, the one of an analysis system. So, the mobile manager, has to offer a solution regarding the security combination needed to be applied to data (security combination = security properties + security algorithms) ; and also the location where this combination can be performed (e.g. on the mobile or in the Cloud) (see Figure 4.15).

For example, when an application component running on the mobile device wants to send data into the Cloud, the Mobile Manager performs the following actions: 1) intercepts the data that have to be sent into the Cloud; 2) read their sensitivity level;

Security of Mobile Cloud Applications

3) verifies the user choice regarding the security level he/she wants to apply to that level of sensibility; 4) verifies the mobile device energy level; 5) verifies the user choice regarding saving or not the mobile device energy while a certain application is running. According to all this data, the Mobile Manager identifies the optimal security solution that needs to be applied to the data sent into the Cloud. The optimal security solution refers to the security components (properties) combination, along with the security algorithms used by the security component and the execution location.

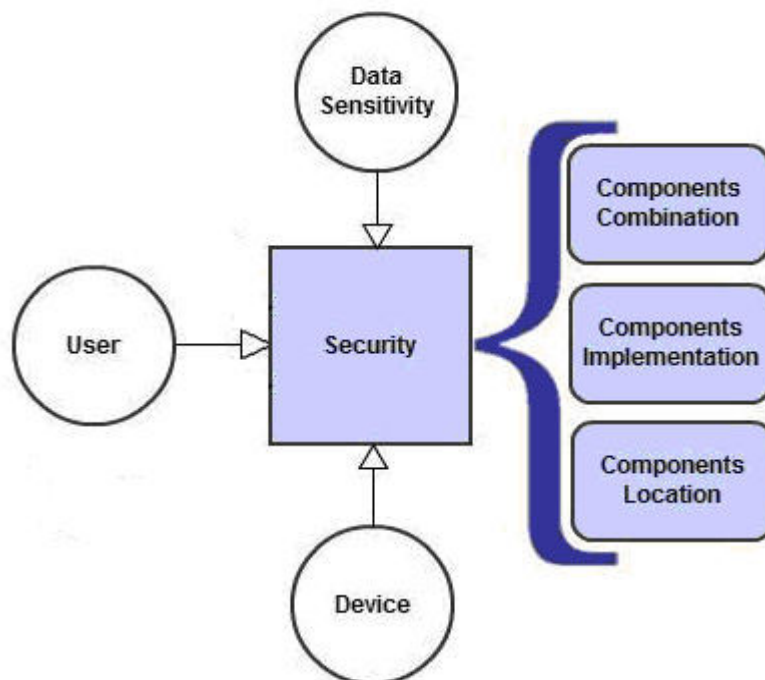


Figure 4.14 Security Framework – answers needed

The analysis system is based on the following three statements:

Statement 1: Application data can have various sensitivity levels. The sensitivity levels are defined according to the harm produced by the data loss or the data theft.

We have chosen three sensibility levels:

- *Low sensibility level.* This level can be applied to all the data whose loss or theft brings no harm to the user.
- *Medium sensibility level.* This level can be applied to all the data whose loss or theft undermines the user image (e.g. user's private data whose loss or theft do not harm their bank account or their security).

Security of Mobile Cloud Applications

- *High sensibility level.* This level can be applied to all the data whose loss or theft causes damages to the user (e.g. passwords, banking information, and identity information).

Statement 2: The user can affect the security solution applied to her/his data. She/he can have an impact on the solution according to her/his level of knowledge in the security field.

We have defined three types of users:

- *The Standard User Type.* She/he is characterized by her/his low knowledge regarding security. To this type of user should be explained some basic notions regarding data and applications security.
- *The Intermediary User Type.* She/he has some knowledge in the security area. For example she/he has basic concepts regarding the various levels of data sensitivity.
- *The Expert User Type.* She/he has high knowledge regarding data security and sensitivity. For example she/he knows security details like security algorithms, what terms like integrity, authenticity, confidentiality and non-repudiation means. She/he also knows what level of security would be suitable for a certain level of sensitivity.

Statement 3: There are several constraints that may impact data security.

The constraints considered in this work and that can affect the security solution are specified by: the application architect and the application user. These constraints are handled as predefined variables.

The list of constraints that need to be defined, by the application architect and the application user, and how these constraints affects the security solution is presented below.

Constraints specified by the application architect

The application architect can specify the following constraints:

- *Application type.* The application type can be of various categories (e.g. banking, e-health, and gaming). The application type affects the value of the default

Security of Mobile Cloud Applications

combination of the security properties. This default combination shows the lowest combination of security properties and security algorithms that may be accepted to secure a certain level of data sensitivity.

- *Sensitivity level.* The sensitivity level of the data to secure. This level can be low, medium or high.

Constraints specified by the application user

To identify the types of constraints that an application user can specify, we had to define a flexibility level that a user should have according to her/his security knowledge. We have then decided to give to the Standard User the lowest level of flexibility whereas the Expert User will have the highest level of flexibility.

The application user can specify the following constraints:

- *Security_choice.* Application data can all be secured with the same security level, or can be secured differently depending on the sensitivity level. Thanks to this constraint, the application's user will be able to take this decision. The values of this constraint are: all, each.
- *Security_level.* The security level chosen by the user for an application data. Its values are: strong, average.
- *Save_battery.* The user has to choose if she/he want to save the device energy while a certain application is running. The values of this constraint are: yes, no. This constraint will have an impact on the choice of the security components that will be used to secure the data; more precisely where the security properties will be applied (i.e. where security components will be executed) rather in the Cloud or in the mobile device.
- *Feature_importance.* This constraint shows what is more important for the user when a certain application is running. The values of this constraint are: battery or security.
- *Components_combination.* This constraint allows the user application to specify the security properties combination. In addition to the default combination of security properties (explained previously), the user may add one or more security properties. However, she/he can never remove a security property. The different values for the security components combination are shown in Table 4.3.

Security of Mobile Cloud Applications

- *Algorithms.* This constraint allows the application user to specify the symmetric, the asymmetric and the hash algorithms she/he wants to use to secure her/his data. In the context of this work, we have chosen the following algorithms: DES, AES-128, AES-256 RSA, SHA-1, SHA-256 (see Table 4.2).

TABLE 4.2 THE ALGORITHMS

Algorithm Type	Algorithm Name
Symmetric	DES, AES-128, AES-256
Asymmetric	RSA
Hash	SHA-1, SHA-256

TABLE 4.3 THE COMPONENTS COMBINATIONS

Combination ID	Combination Value
C1	I (Integrity)
C2	A (Authenticity)
C3	C (Confidentiality)
C4	I + NR (non-repudiation)
C5	A + NR
C6	C + NR
C7	I + C
C8	A + C
C9	I + C + NR
C10	A + C + NR

The Analysis System has two functions: 1) User Choices Capture, and 2) Security Components Execution.

The User Choices Capture function collect the constraints (*security_choice*, *security_level*, *save_battery*, *feature_importane*, *components_combination* and *algorithms*) specified by the application through the user interface and store it in a local database. The first four constraints (*security_choice*, *security_level*, *save_battery*, *feature_importane*) can be defined by all the users without taking into the account the user type. The fifth constraint (*components_combination*) can be defined by the

Security of Mobile Cloud Applications

intermediary and the expert users. The sixth parameter (algorithms) can be defined only by expert users.

In the local database is stored the combination ID (e.g. C1, C2...). This information is stored for each sensitivity level of an application data. For example, as presented in Figure 4.16, for the data of an application 'App1' having a *High* sensitivity level, the combination value that will be stored is 'C8'.

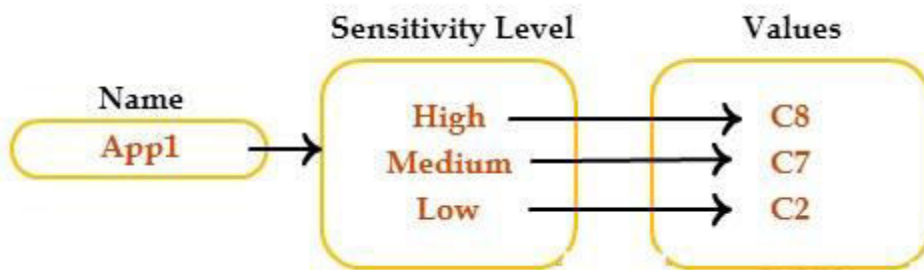


Figure 4.15 Capture User Choices example

The Security Components Execution function determines the execution location of the security components (i.e. where the security components will be executed) according to the constraints specified by all user types. In the case of the Standard and Intermediary User Type, the security algorithms will be chosen by the Mobile Manager while operating the analysis system function.

The input parameters of this function are: battery_status and user choices: {security_level_combination_ID, save_battery, feature_importance}. Basing on these parameters, the output result is obtained using IF...THEN...ELSE rules.

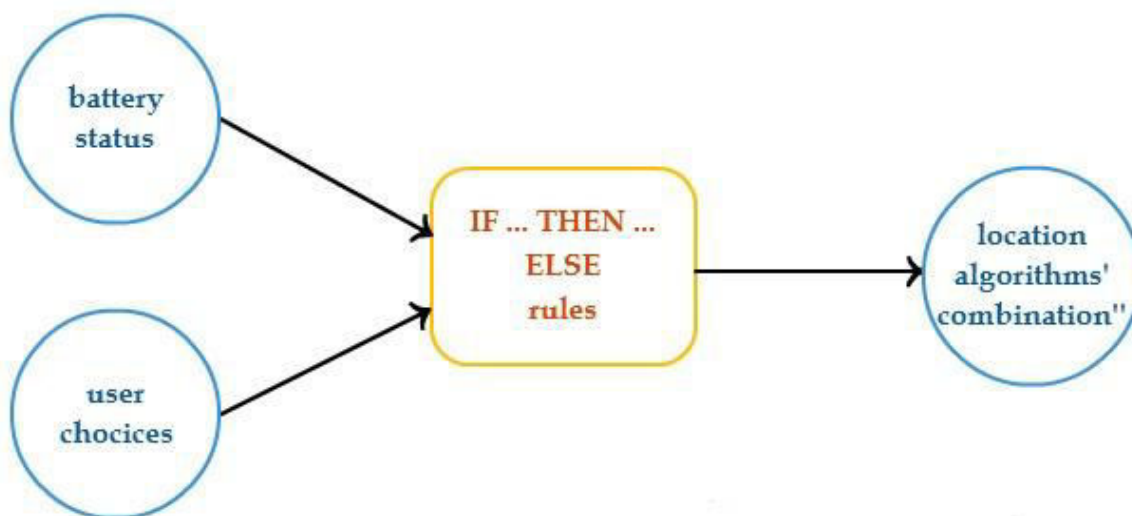


Figure 4.16 The Security Components Execution Part

Security of Mobile Cloud Applications

At the beginning the rules were written in natural language. In the following we are presenting some rules examples.

“If the user chose not to save the phone battery and the user stated that between battery and security options the most important is security then verify the mobile device battery status and the security properties combinations chose by the user.”

“If the battery level is greater than forty per cent and the security combination chose by the user includes integrity property and the confidentiality property then execution location is the mobile device and the hash security algorithms used are: SHA-256, AES-256 and RSA”

“If the battery level is less than twenty per cent and the security combination chose by the user includes only the integrity then the execution location is the mobile device and the has algorithm used is SHA-256”.

“If the user chose to save the phone battery and the user stated that between battery and security options the most important is battery then verify the mobile device battery status and the security properties combinations chose by the user.”

“If the battery level is greater than forty per cent and less than sixty per cent and the security combination chose by the user includes integrity property and the confidentiality property then execution location is the cloud.

“If the battery level is less than twenty per cent and the security combination chose by the user includes only the integrity then the execution location is the mobile device and the has algorithm used is SHA-1”.

4.4 Conclusions

Mobile Cloud Computing introduces many security issues due to the fact that it combines mobile devices with Cloud services and because there is not a well-defined application model. Generally, the existing security solutions have tackled these issues independently, thus are not enough to secure mobile cloud applications.

The solutions proposed to resolve the mobile cloud security issues treat independently each type of security problems. Thereby, in the case of components base mobile cloud applications, is needed to combine different solutions in order to secure it.

From the existing security solutions very few take into account the mobile device energy constraints; furthermore the security solutions proposed to secure the communications between the mobile device and the Cloud are energy consuming solution. The lack of resources is one of the biggest disadvantages of mobile phones. Therefore, a security solution for mobile cloud applications must consider mobile device constraints.

The mobile owners (mobile end-users), running mobile cloud applications may have different expectations regarding the security level of their private data. The existing solutions for mobile cloud applications do not allow the users to express their opinion regarding the security level to be applied to their private data. Furthermore, there are no security solutions which take into account the data sensibility constraint.

This work focuses on component based mobile cloud applications. The main goal of this work is to propose a solution to secure data communication between the same application components that can run on the mobile device and/or the Cloud. The most important characteristic of our framework is that: 1) it allows applying different security properties to different kinds of data (according to the data sensibility level) and not the same properties to all the data processed by the application, 2) the user preferences are taken into consideration and 3) the mobile device energy consumption constraint is also taken into account.

4.4.1 Contributions

A new framework, Secure Mobile-Cloud Framework, to secure data communication between mobile cloud application components was proposed [PBC+13a]. The Secure Mobile-Cloud Framework is composed of two types of components 1) security components (LECCSAM components) and 2) management components. The security components have as purpose the implementation of the

Security of Mobile Cloud Applications

eponym security properties. The management components have been designed to identify and apply the appropriate security properties and therefore security components to user's data.

While providing a security solution, the proposed framework must take into account several constraints: data sensibility level, user preferences, and the mobile device energy consumption [PBC+13b].

In order to take into account the constraints listed above, a new functionality was added for one of the management components (for the Mobile Manager), the one of an analysis system. The analysis system works based on 'if then else' rules [PBC+13c].

4.4.2 Publications

[PBC+13a] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "A Security Framework for Mobile Cloud Applications "Networking in Education and Research", Sinaia, Romania, ARNIEC/RoEduNet Agency, IEEE Romanian Section, ISSN-L 2068-1038, 17-19 January, 2013.

[PBC+13b] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "Personalized Security Mechanism for Mobile Cloud Applications", in Acta Tehnica Napocensis, Electronics and Communications, No.2 Vol. 54, pp. 24-27, 2013.

[PBC+13c] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "A System to Analyze the User's Security Options for Mobile Cloud Applications", The 6th International Conference on Security for Information Technology and Communications, June 25, 2013.

Security of Mobile Cloud Applications

“The more users’ expectations prove right, the more they will feel in control of the system and the more they will like it.”

Jakob Nielsen

5. Secure Mobile-Cloud Framework Implementation

Contents in Brief

5.1 The Security Managers	74
5.2 The Auxiliary Managers.....	79
5.3 The Mobile Manager.....	82
5.4 Databases Implementation	85
5.5 The User Interface	90
5.6 Unit Tests.....	93
5.7 The Software Tools Used	95
5.8 Conclusions.....	97

Chapter Overview

This chapter describes the implementation on the mobile device for Secure Mobile-Cloud Framework presented in the previous chapter.

Security of Mobile Cloud Applications

This chapter presents at the beginning the implementation for the Secure Mobile-Cloud Framework. The implementation of the Secure Mobile-Cloud Framework includes the implementation of the: 1) Security Managers (Section 5.1), 2) Auxiliary Managers (Section 5.2) and 3) Mobile Manager (Section 5.3). Then, the design and implementation of the databases used for storing the user options is presented (Section 5.4) followed by the description and implementation of the user interface (Section 5.5). The last parts includes some unit tests (Section 5.6), a description of the used software tools (Section 5.7) and at the end there are presented the conclusions.

5.1 The Security Managers

This section will present the implementation on the mobile device of the Security Managers described in the Chapter 4, Section 4.3.1. This part includes the Mobile Security Manager and the Security Components.

In Figure 5.1 is presented the class diagram for the Security Managers Implementation part. On this diagram, we can see the relations between the various classes. To implement this part, it have been defined seven classes: MobileSecurityManager, Integrity, Authenticity, Confidentiality, NonRepudiation, Operations and Algorithms.

The MobileSecurityManager class implements the functionality of the Mobile Security Manager presented in the Chapter 4, Section 4.3.1. This class has five methods. The most significant are: `apply_combination_toEncrypt(data, combination)` and `apply_combination_toDencrypt(data, combination)`.

The pseudo code of the `apply_combination_toEncrypt(data, combination)` method is given below. This method applies the appropriate security level to data in order to secure them, whereas the `apply_combination_toDencrypt(data, combination)` method is used to unsecure the received data.

Security of Mobile Cloud Applications

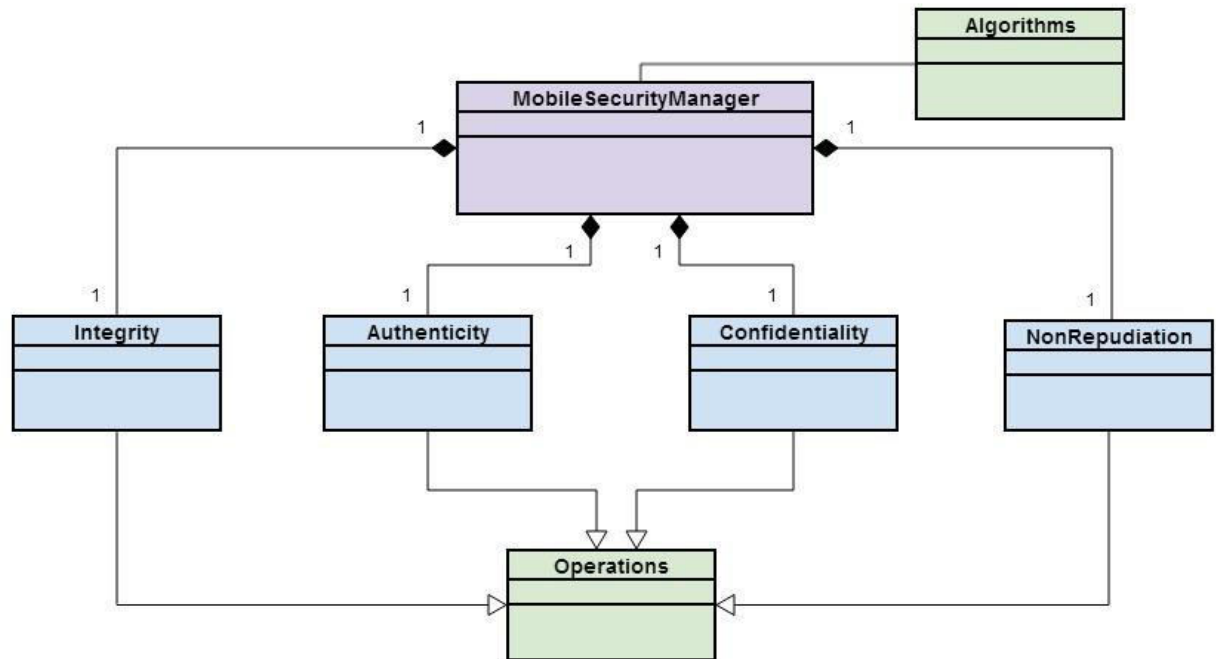


Figure 5.1 Mobile Security Manager – Class Diagram

As it can be seen in the pseudo code, given below, the method follows the following steps:

- 1) It finds the security properties combination and the corresponding algorithms for data security level provided. This is done by calling `discoverAdvancedComponents(securityLevel)`, method implemented by the Policy Manager. This method result returns a string containing the first letter form the name of the security properties plus the corresponding cryptographic algorithms.
- 2) The string returned to the previous step contains information about the security properties combination and about the security algorithms corresponding to a certain security level. Thereby this string needs to be read. While reading the string several intern parameters (e.g Integrity, Authenticity, integrity_algorithm) are initialized. For example, if in the string the letter I is found, the internal parameter Integrity is set to TRUE, and also the integrity_algorithm parameter is initialized with one of this values SHA-1 or SHA-256.
- 3) After the security properties and algorithms have been determinate, it follows their application to the users' data. This is made by calling the appropriated methods (e.g. `applyIntegrity`, `applyAuthenticity`) implemented by the Integrity, Confidentiality, Authenticity or NonRepudiation classes.

Security of Mobile Cloud Applications

Algorithm description (pseudo code):

```
apply_combination_toEncrypt(data, securityLevel){  
  
    discoverAdvancedComponents(securityLevel);  
    read_propr_algorith_string(propr_algorith_string);  
  
    if (Integrity)  
    switch(integrity_alg){  
        case '1':  
            applyIntegrity(data, alg_hash_1);  
        case '2':  
            applyIntegrity(data, alg_hash_2);  
        ...  
  
    if (Authenticity)  
        if(alg_hash_1 and alg_asim_1)  
            build_asymmetric_cipher;  
            applyAuthenticity(data, alg_hash_1, asymCipher);  
        if(alg_hash_2 and alg_asim_2){  
            build_asymmetric_cipher;  
            applyAuthenticity(data, alg_hash_1, asymCipher);  
        ...  
  
    if (Confidentiality)  
        if(alg_sym_1 and alg_asym_1)  
            build_asymmetric_cipher;  
            build_symmetric_cipher;  
            applyConfidentiality(data, symCipher_c, asymCipher_c);  
        ...  
  
    if (NonRepudiation)  
        if(alg_sym_1 and alg_asym_1)  
            build_asymmetric_cipher;  
            build_symmetric_cipher;  
            applyNonRepudiation(data, symCipher_c, asymCipher_c);  
        ...  
  
    if (IntegrityConfidentiality)  
    switch(integrity_alg)  
        case '1':  
            if(alg_sym_1 and alg_asym_1)  
                build_asymmetric_cipher;  
                build_symmetric_cipher;  
                result = applyConfidentiality(data, symCipher_c, asymCipher_c);  
                applyIntegrity(result, alg_hash_1);  
            ...  
  
    if (AuthenticityConfidentiality)  
        if(alg_sym_1 and alg_asym_1)  
            if(alg_hash_1 and alg_asim_1)  
                build_asymmetric_cipher;
```

Security of Mobile Cloud Applications

```
build_symmetric_cipher;  
result = applyConfidentiality(data, symCipher_c, asymCipher_c);  
applyAuthenticity(result, alg_hash_1, asymCipher);
```

...

The Operations class implements the methods performing symmetric and asymmetric encryption and decryption and also the hash operation. These methods are:

- *encrypt*: It use a symmetric or asymmetric algorithm to encrypt data. As an input, it has an array of byte, which corresponds to the data to encrypt, and the type of the cipher algorithm (symmetric or asymmetric). As an output, it returns a variable of bytes type representing the encrypted data;
- *decrypt*: It decrypts data encrypted with a symmetric or asymmetric algorithm. As an input, it has an array of byte, which corresponds to the encrypted data, and the type of the cipher algorithm (symmetric or asymmetric). As an output, it returns an array of bytes that corresponds to the dencrypted data;
- *hash*: It takes as input an array of byte (i.e. the data to which will be added an hash and adds an hash according to the hash algorithm given as an input parameter. It returns a variable of bytes representing the data concatenated to their hash;

In order to implement the security components presented in the Chapter 4, four classes have been implemented: Integrity, Authenticity, Confidentiality and NonRepudiation. Each of these classes extends the Operations class.

The Integrity class has four methods, two private and two public methods. It was chosen this implementation solution in order to apply the encapsulation property. The public methods are visible to the other classes (e.g. MobileSecurityManager) while the private methods are invisible to the external classes. One of the private methods is the one who implements/applies the integrity property to the input data, the other private method is the one who verifies the integrity property of the input data. These private methods can only be called by the public methods in this class.

- *applyIntegrity* and *aIntegrity*: These two methods provide the functionality of the Integrity component for plain text data. The first method is public and the second one is private. These methods are receiving as input the plain text data and the hash algorithm type. The public method calls the private method that uses the hash method implemented into the Operations class to perform the hash operation

Security of Mobile Cloud Applications

- *verifyIntegrity* and *vIntegrity*: These two methods provide the functionality of the Integrity component for the secured data. The first method (*verifyIntegrity*) is public and the second one (*vIntegrity*) is private. These methods are receiving as input the secured data and the hash algorithm type. The public method calls the private method that uses the hash method implemented into the Operations class to perform the operation. Then, the private method returns true if the integrity is ensured and false if not

The Confidentiality class has also four methods, two public and two private. This class was built on the same principle as the Integrity class.

- *applyConfidentiality* and *aConfidentiality*: These two methods (one public and one private) provide the functionality of the Confidentiality component for plain text data. These methods receive as input the plain text data and the type of the ciphering algorithm (symmetric/asymmetric cipher). The public method calls the private method which uses symmetric and asymmetric encrypt methods implemented into the Operations class to perform the encryption operations. The symmetric encrypt method is used to encrypt the input data while the asymmetric encrypt method is used to protect the key used in the symmetric encryption
- *verifyConfidentiality* and *aConfidentiality*: These two methods (one public and one private) provide the functionality of the Confidentiality component for ciphered data. These methods are receiving as input the ciphered data (dataIn - of String), a symmetric and an asymmetric cipher (symCipher - of Cipher type, asymCipher - of Cipher type). The public method calls the private method which uses symmetric and asymmetric decrypt methods implemented into the Operations class to perform the decryption operations

The other classes that achieve the authenticity and non-repudiation components are implemented following the same principles as integrity and confidentiality components.

5.2 The Auxiliary Managers

This section will present the implementation of the auxiliary managers, i.e. Policy Manager and the State Manager described in the Chapter 4 Section 4.3.2.

In Figure 5.2 is presented the class diagram of these managers. As it shown in this figure, there are two classes: the PolicyManager and StateManager.

The PolicyManager class implements the functionality of the Policy Manager presented in the Chapter 4, Section 4.3.2. It includes four methods: discoverComponents(), discoverAdvancedComponents(), discoverSecurityLevel() and discoverAdvancedSecurityLevel().

As specified in Chapter 4 Section 4.3.2, the Policy Managers keeps the security composition rules. These rules define the security properties (components) combination specific to a certain level of security.

It was also showed in Chapter 4 Section 4.3.3 that each user type can specify several constraints in order to secure her/his private data. The following constraints: security_choice, security_level, save_battery, feature_importane can be defined by all the users without taking into the account the user type. The components_combination can be defined by the intermediary and the expert users. The security algorithms can be defined only by expert users. After all this constraints are analyzed by the Mobile Manager, the returned result has to include the following details: 1) the security properties combination, 2) the security algorithms and 3) the execution location. But because the users can select a different number of constraints, Mobile Manager has to follow a different flow for standard and intermediary user types compared to advanced user type. In the case of standard and intermediary user types the Mobile Manager has to decide the security algorithms used for the security properties and the execution location. In the case of advanced user type the only choice needed to be made is the execution location. Also it has been shown in Chapter 4 Section 4.3.3 that for each security properties combination has been defined a combination identifier (e.g. for the combination identifier C8 the security properties combined are Authenticity and Confidentiality, see Table 4.3). At the implementation, this combination had to be improved to include also the security algorithms.

Therefore into the implementation we have defined two encoding types for a security level:

Security of Mobile Cloud Applications

- *The basic_encode*: it defines the security properties combination. Its form is as follow: C[n]; where C stands for combination and n is a number between 1 to 10. This code is used when the type of the user is Standard or Intermediary.
- *The advanced_encode*: it also defines the security properties combination. However, it includes the security algorithm chosen by the Advanced user. The form of this code is: AC[n]; where AC stands for advanced combination and n is a number starting from 1.

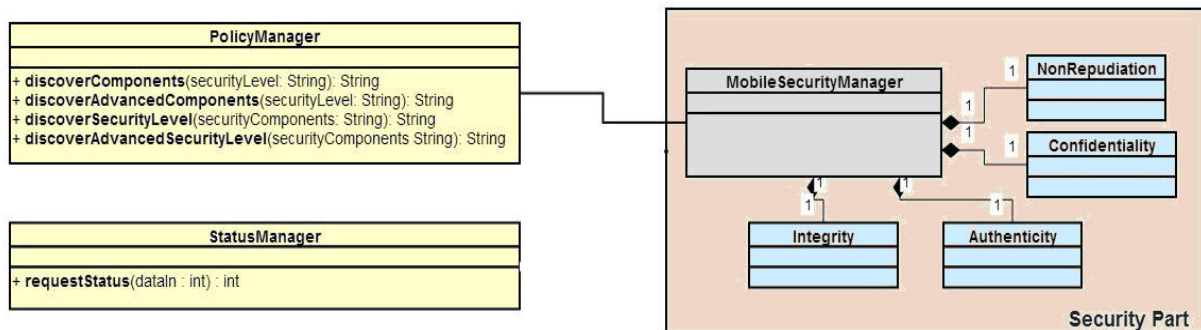


Figure 5.2 Auxiliary Managers – Class Diagram

Policy Manager is the only one who contains/knows the link between a certain encoding (basic or advanced) and the security properties combination and algorithms (Figure 5.4, Figure 5.5). This link can be determinate using the methods implemented in the PolicyManager class. These methods are described in the following:

- *discoverComponents()*: receives the *basic_encode* as input and returns a string containing the name of the security properties, more specifically the first letter of the component. The length of this string is eight characters as it can be seen in Figure 5.3. The name of the component is specified if the eponym security property is needed for the requested security level. An example is given in the Figure 5.4.



Figure 5.3. Eight character string (basic code decoding)

Security of Mobile Cloud Applications

On the current implemented version of the `StateManager` class, it does not collect the energy level of the device. It simulates the compartment of a battery level collector. Its functionality is explained into the following.

`StateManager` class contains a method that receives as input an integer value. This integer value represents a simulated moment of time. The method returns a certain number that depends on the value received as input. This number represents the simulated battery level value at a certain moment given.

5.3 The Mobile Manager

This section will present the implementation of the Mobile Manager described in the Chapter 4 Section 4.3.3; particularly the analysis functionality.

To implement the analysis functionality of the Mobile Manager, it have been defined two classes and one interface: `MobileManager` (the interface), `MobileManagerParA` and `MobileManagerPartB` (the classes). These classes are shown in the class diagram of the Figure 5.7.

The `MobileManagerPartA` class was implemented as part of the process that deals with the capture of users choices. It is the link between the user interface and the `PolicyManager` class.

The `MobileManagerPartB` class was designed to implement the analysis functionality of the Mobile Manager presented in the Chapter 4 Section 4.3.3. The method that handles the analysis functionality is called `location_executed()`. It receives as inputs the following parameters:

- *user_type_group*: identifies the user type group. We have divided the users into two groups. The first group includes both standard and intermediary users and the second group includes the advanced users. We have defined these two groups for the following reasons: 1) the flexibility a user may have in defining the available constrains on which is based the security solution applied to her/his data (which leads to the next reason); 2) the encoding mode chosen to identify a certain security level (mentioned into this chapter, Section 5.2).

Security of Mobile Cloud Applications

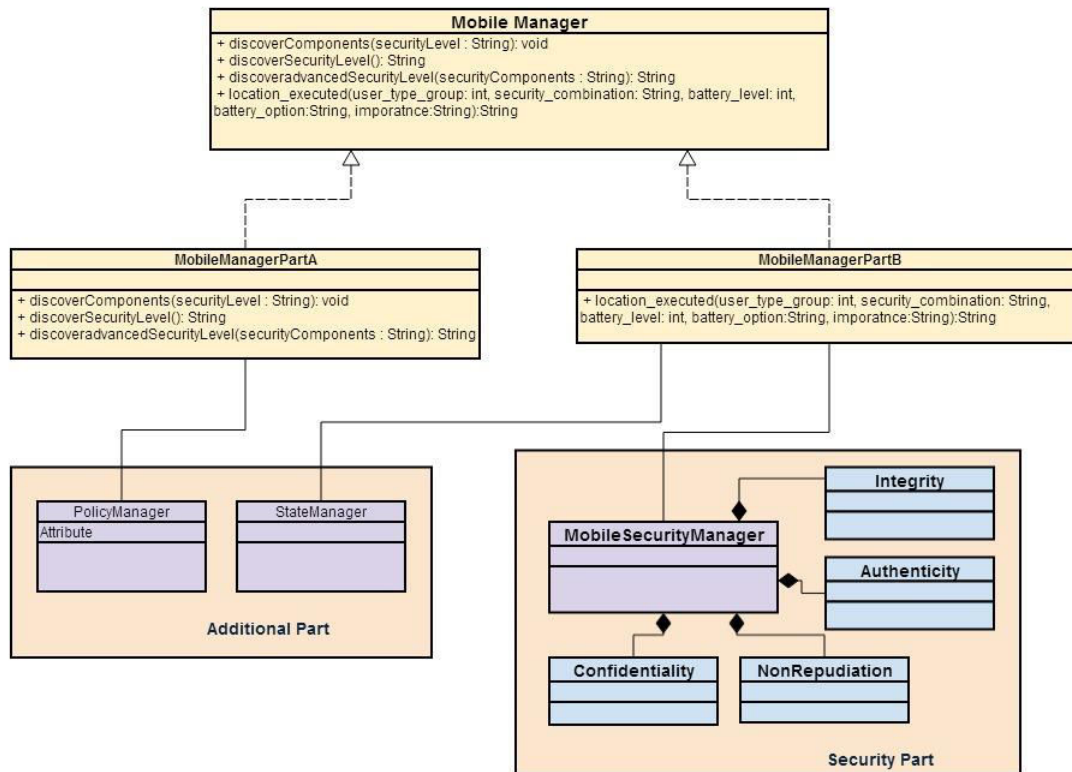


Figure 5.7 Mobile Manager Class Diagram

- *security_combination*: identifies the security level identifier. It can be in the form C[n] or AC[n], depending on the user type.
- *battery_level*: receives the result sent by the State Manager.
- *battery_options* and *importance*: represent the user options. The first parameter is the user choice for saving or not the device energy while a specific application is running. The second parameter shows what is more important for the user: battery or security.

A portion of the pseudo code of the `location_executed()` method is given below.

```

switch(user_type_group){
case '1':
    if battery_option is false and importance is security
        if battery is greater then 40
            if security_combination is C1
                result = AC2;
  
```

Security of Mobile Cloud Applications

```
if security_combination is C2
    result = AC5;
if security_combination is C3
    result = AC7;
if security_combination is C4
    result = AC11;
if security_combination is C5
    result = AC23;
if security_combination is C6
    result = AC27;
if security_combination is C7
    result = AC35;
if security_combination is C8
    result = AC43;
if security_combination is C9
    result = AC47;
if security_combination is C10
    result = AC53;
```

```
if battery is less then 40 and greater then 20
    if security_combination is C1
        result = AC2;
    if security_combination is C2
        result = AC5;
    if security_combination is C3
        result = AC9;
    if security_combination is C4
        result = AC14;
    if security_combination is C5
        result = AC25;
    ...
```

```
if battery is less then 20 and greater then 10
    if security_combination is C1
        result = AC2;
    if security_combination is C2
        result = AC3;
    if security_combination is C3
        result = AC9;
    if security_combination is C4
        result = CLOUD;
    if security_combination is C5
        result = CLOUD;
    ...
```

```
if battery is less then 10
    if security_combination is C1
        result = AC1;
    if security_combination is C2
        result = CLOUD;
    if security_combination is C3
        result = CLOUD
```

Security of Mobile Cloud Applications

```
if security_combination is C4
    result = CLOUD;
if security_combination is C5
    result = CLOUD;
```

5.4 Databases Implementation

The security framework needs to know the values chosen by the user for each constraint. Thereby the security framework uses relational databases to save these data. The databases have been implemented using SQLite, a powerful and effective library to manage databases, available to all Android applications.

5.4.1.1 The Design

The databases that we have defined for our framework are: Admin, Applications, and User (see Figure 5.8, Figure 5.9, and Figure 5.10).

The Admin database was designed to keep the default information needed by the security framework. The default information is data already predetermined and describe which security properties correspond to a certain type of applications. This default information shows the lowest combination of security properties and security algorithms that may be accepted to secure a certain level of data sensitivity. As it can be seen in the Figure 5.8 this scheme contains three tables:

- *Types_Table*: contains the a list with the types of applications. This table kepps the following information: 1) the primary key *TYPE_ID* and 2) an attribute *TYPE_NAME*. In the current version of the database, only three types of applications have been defined: *bank*, *health*, *game*.
- *CombinationEncods_Table*: contains a list of security properties combination encoding. As shown in Table 4.3 (in Chapter 4, Section 4.3.3), each combination is identified by a predetermined combination identifier (e.g combination identifier: 'C1' or 'C2'). The combination identifier is uniquely identified by the primary key *COMBINATION_ID* and has also another attribute *COMBINATION_NAME*. In the current version of the database, this table contains only the combinations specified in Table 4.3, Chapter 4, Section 4.3.3.

Security of Mobile Cloud Applications

- *SecurityCombinations_Table*: contains the links between the two tables *Types_Table* and *CombinationEncods_Table*. For each data sensibility level of each mobile cloud application type, we have defined a combination of security properties. This combination is called the default combination and it has been presented in Chapter 4, Section 4.4.3. Each default combination is uniquely identified by the primary key *DEFAULT_COMBINATION_ID* and is described by the application type, the security combination encoding, the security level and the sensibility level: *TYPE_ID*, *COMBINATION_ID*, *SECURITY* and *SENSIBILITY*.

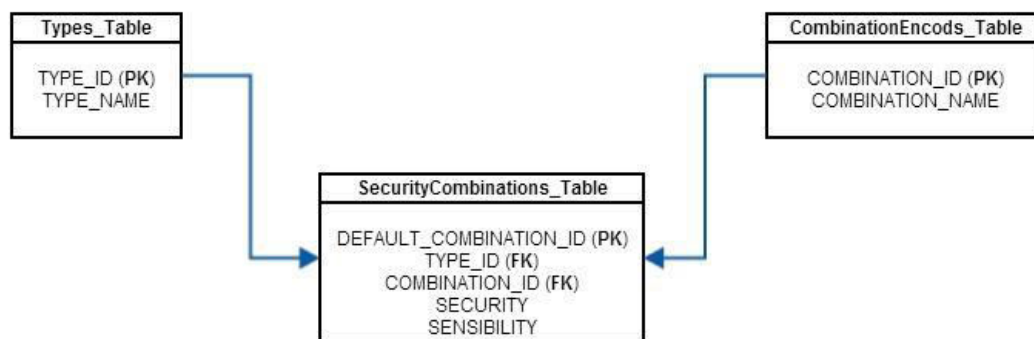


Figure 5.8 Admin Database

The Applications database was designed in order to store the user options regarding the data security level for the mobile cloud application (i.e. the security level that it chooses for each data sensibility level). As it can be seen in the schema Figure 5.9, this database has four tables:

- *Applications_Table*: contains the list of mobile cloud applications installed on the mobile device. Each record is identified by the primary key *APPLICATION_ID*. Furthermore, a record is described by the mobile cloud application name (*NAME*) and type (*TYPE*). In the current version of the database, this table contains a default record data: *e-health* and *health*.
- *Applications_Security_Table*: contains the user options regarding the data security level for a certain mobile cloud application (i.e. the security level that it chooses for each data sensibility level). This data is collected from the security framework user interface. Each option saved in this table is identified by the primary key *APPLICATION_SECURITY_ID* and by the foreign key *APPLICATION_NAME*. Furthermore an option is described by the data sensibility level

Security of Mobile Cloud Applications

(*SENSIBILITY_LEVEL*) and by the security properties combination (*SECURITY_COMBINATION*) appropriate to each data.

- *Applications_Battery_Table*: contains the user options regarding the preservation of the battery for a specific mobile cloud application. Actually, the user will have to specify if she/he would like to preserve or not the battery of her/his mobile phone when the application is running. This information is given through a user interface and stored in the column table *BATTERY*. Each record, in this table is identified by the primary key *BATTERY_ID* and by the foreign key *APPLICATION_NAME*. The user option is saved into the column table *BATTERY*.
- *Applications_Priority_Table*: contains data that specifies which of the two constraints: security or battery is more important for the user. As for the *Applications_Battery_Table*, a record is identified by a combination of a primary key, *PRIORITY_ID*, and a foreign key *APPLICATION_NAME*. The option is saved into the column table *PRIORITY*.

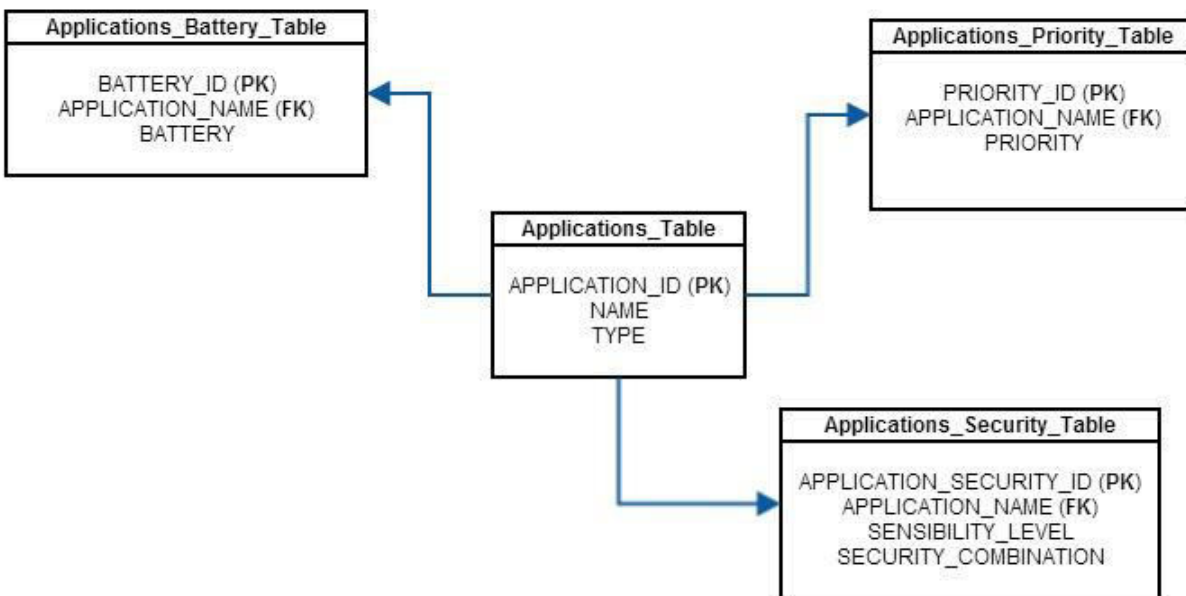


Figure 5.9 Applications Database

The User database was designed in order to store the user options regarding her/his level of knowledge in the security area. As it can be seen in the schema Figure 5.10, this database has only one table:

- *User_Level_Table*: contains the user option regarding her/his expertise in the security field. This table keeps only one record; it does not have primary or foreign keys. The

Security of Mobile Cloud Applications

record is described by the user type (*TYPE*) and the time (*DATE*) when the user option was collected .

User_Level_Table
TYPE
DATE

Figure 5.10 The User Level Table

5.4.1.2 The Implementation

To store the information into the database, we have used the SQLite database in Android applications. SQLite is an Open Source database. SQLite supports standard relational database features like SQL syntax, transactions and prepared statements.

The advantages of SQLite are: 1) the database requires limited memory at runtime; 2) SQLite is embedded into every Android device; 3) there is no need to have a setup procedure or to administrate the database; the only step that is required is to define the SQL statements for creating and updating the database.

To create and upgrade a database in an Android application, it is necessary to create a subclass of the SQLiteOpenHelper class. The classes that will create the databases presented in Section 5.4.1.1 are: AdminDatabase, ApplicationsDatabase and UserDatabase (see Figure 5.11). Each of these overrides the following SQLiteOpenHelper class methods in order to create and update the databases: *onCreate()* and *onUpgrade()*. The *onCreate()* method will create all the database tables with the default data. The *onUpgrade()* method will simply delete all existing data and re-create the tables.

The classes that will handle the databases connections, the data access and the data modification: AdminDatabaseInitialSources, ApplicationsDatabaseInitialSources and UserDatabaseInitialSources (see Figure 5.11). These classes are used as data access object (DAO) to manage the data. The methods created for these classes, implement CRUD (create, delete, update, insert) operations to manage stored data.

For each database table, we have defined a classes in order to implement the *get()* and *set()* methods. These classes contain the methods that will be used to access the data saved in the database. For the Admin database these classes are: ItemTypeApps, ItemSecurityCombination and ItemDefaultSecurityCombination (see Figure 5.12).

Security of Mobile Cloud Applications

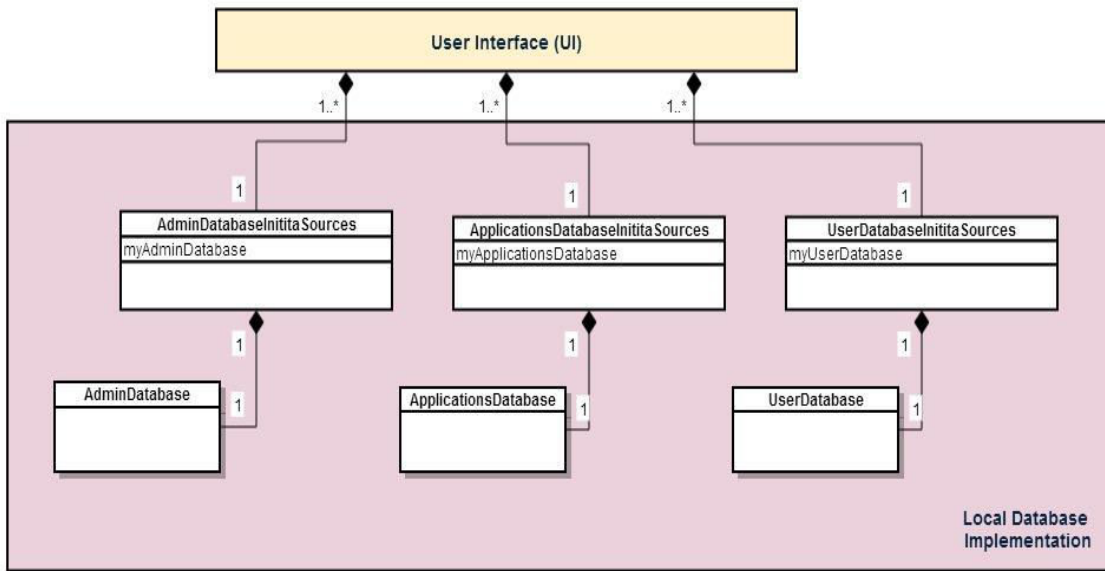


Figure 5.11 Classes Diagram for Database Implementation

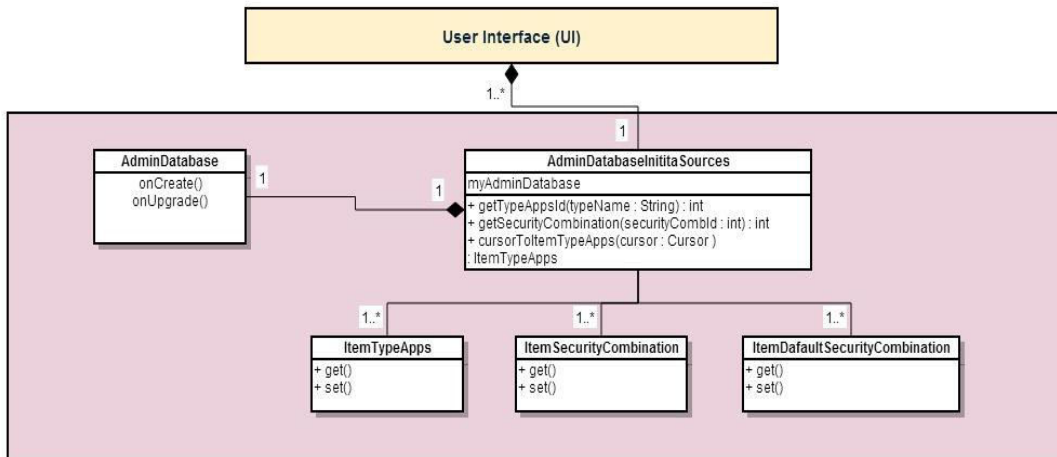


Figure 5.12 Class diagram for the Admin database

5.5 The User Interface

As said previously, a feature of this proposed framework is to allow the users to express their choices regarding the security level they want to apply to their data.

This section will present the implementation of the user interface on the mobile device. The user interface was designed in order to collect the user options regarding the security level that she/he would like for her/his data and her/his options on saving or not the device energy .

To create a user interface running on Android, there are two methods. The first one is the conventional method of Java and XML coding (see Figure 5.13 and Figure 5.14). The second method is to use the palette provided into the Eclipse development environment. This method provides the facility to add an item by drag and drop.

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="68dp"
    android:orientation="vertical" >

    <Button
        android:id="@+id/each_low_ok.button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_marginRight="24dp"
        android:text="@string/main_b_OK" />

    <Button
        android:id="@+id/each_low_exit.button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="28dp"
        android:text="@string/main_b_Exit" />
</RelativeLayout>
```

Figure 5.13 XML coding

```
ok_button.setOnClickListener(
    new OnClickListener() {

        public void onClick(View v) {

            Intent intent = null;
            String message = "";
            String securityLevel = "";

            String appType = applications_datasource_management.getItemAppType(appName);

            switch (profil.getCheckedRadioButtonId()) {
                case R.id.each_low_strong:
                    message = "Do you want to proceed forward?";

                    securityLevel = "strong";

                    int typeIdS = admin_database_management.getTypeAppsId(appType);
                    int securityCombinationIdS = admin_database_management.getSecurityCombinationId(typeIdS, securityLevel, "low");
                    String securityCombinationS = admin_database_management.getSecurityCombination(securityCombinationIdS);

                    intent = new Intent(EachLowActivity.this,
                        EachLowPropertiesActivity.class);
                    intent.putExtra("securityCombination", securityCombinationS);
                    intent.putExtra("appName", appName);
            }
        }
    }
);
```

Figure 5.14 Java coding

Security of Mobile Cloud Applications

The goal of the user interface is to: 1) set up the user profile and 2) collect the security options selected by the user (see Figure 5.15, Figure 5.16, Figure 5.17 and Figure 5.18).

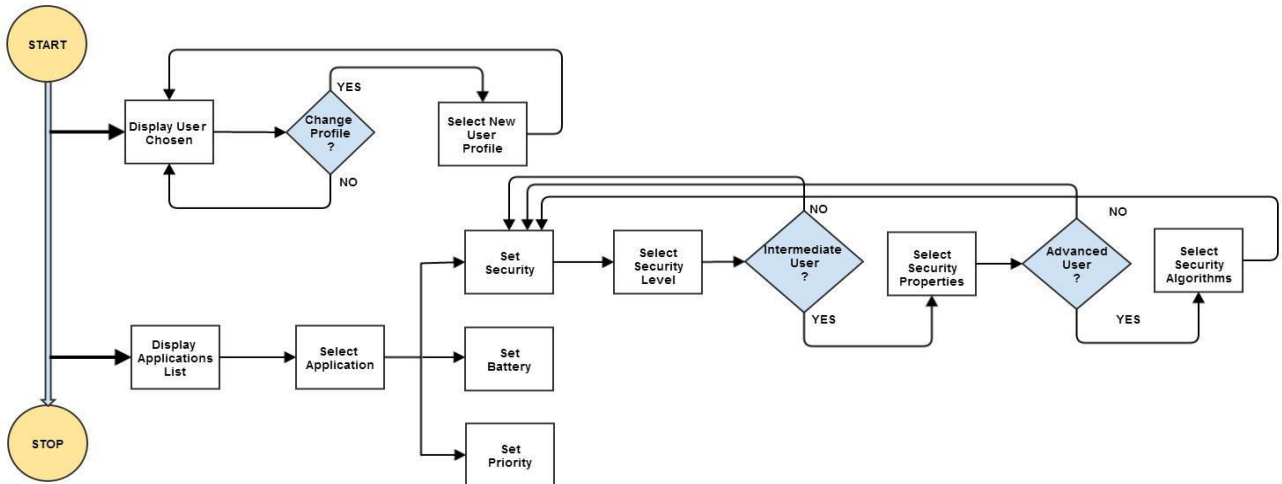


Figure 5.15 UI functionality diagram

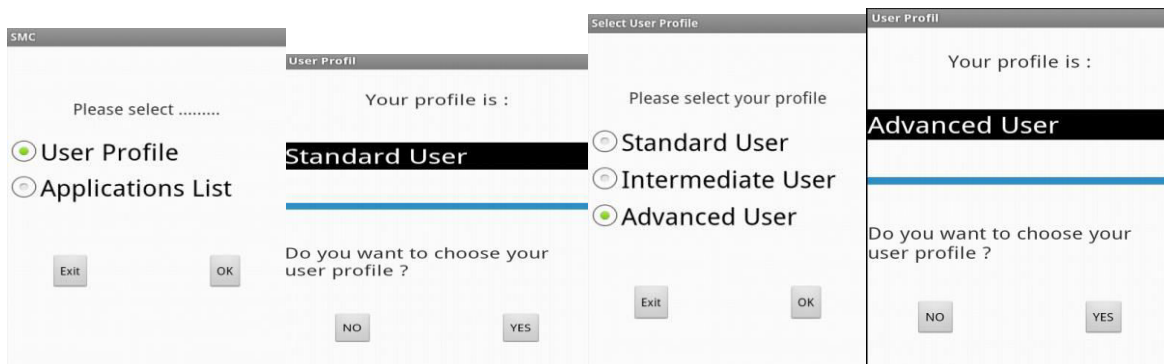


Figure 5.16 User profile set

Security of Mobile Cloud Applications

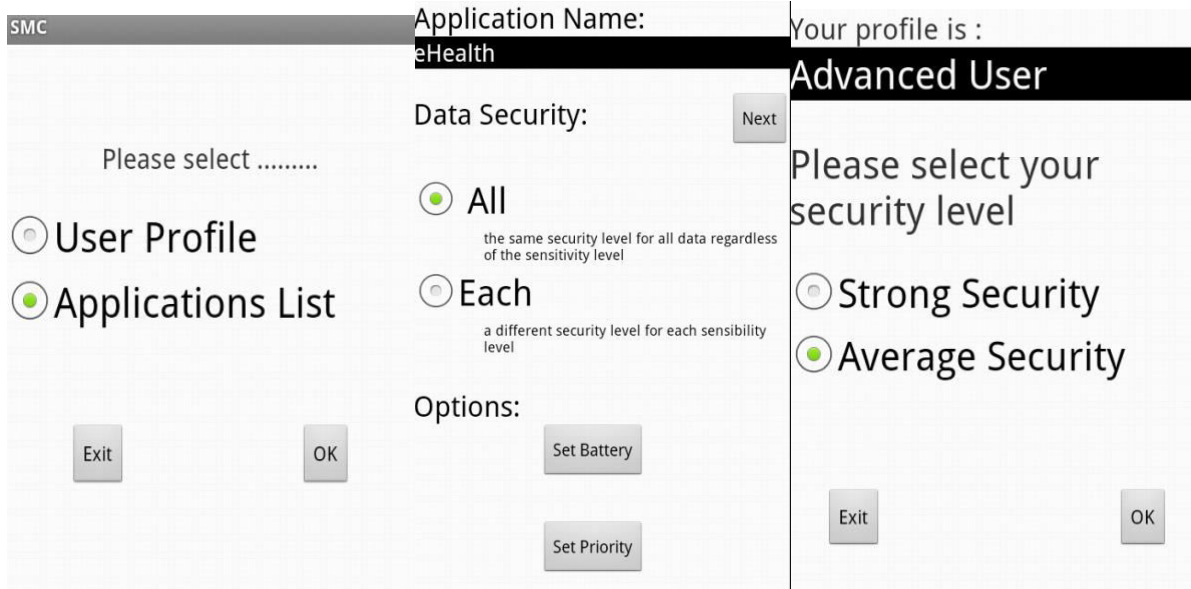


Figure 5.17 Security options set (part a)



Figure 5.18 Security options set (part b)

5.6 Unit Tests

In this section there are presented a couple of test. These tests target the security framework functionality on the mobile device.

The first scenario:

The user is of type advanced. She/he chooses the following options: 1) all data are secured equally regardless of the sensitivity level; 2) security level of type average; 3) as security properties he chooses only confidentiality; 4) as security algorithms he chooses: SHA(Secure Hash Algorithm), AES(Advanced Encryption Standard) and RSA; 5) he chooses to save battery; and 6) the priority is also the battery.

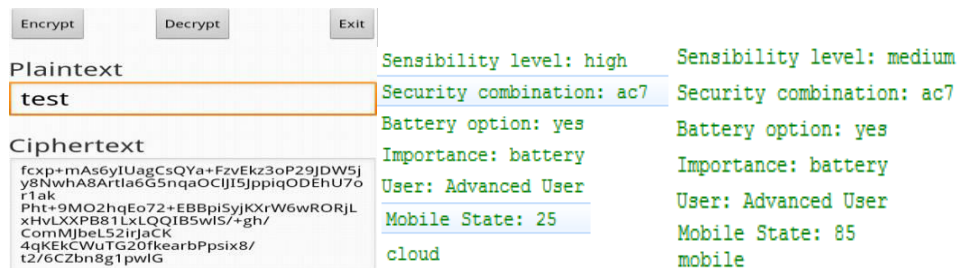


Figure 5.19 Results – Unite test first scenario

It can be seen in Figure 5.19 that, for data with different sensibility level (e.g. high and medium), there is the same security combination (e.g. ac7). Also, according to the mobile energy status (e.g. 25 or 85), one operation is executed on the mobile device (the result is also shown in Figure 5.19) and the other in Cloud.

The second scenario:

The user is of type standard. She/he chooses the following options: 1) all data are secured in a distinct way, depending on the sensitivity level; 2) security level of type average is chosen for low sensitivity data; 3) security level of type strong is chosen for medium and high sensitivity data; 4) he chooses not to save battery; and 5) the priority is the security.

Security of Mobile Cloud Applications

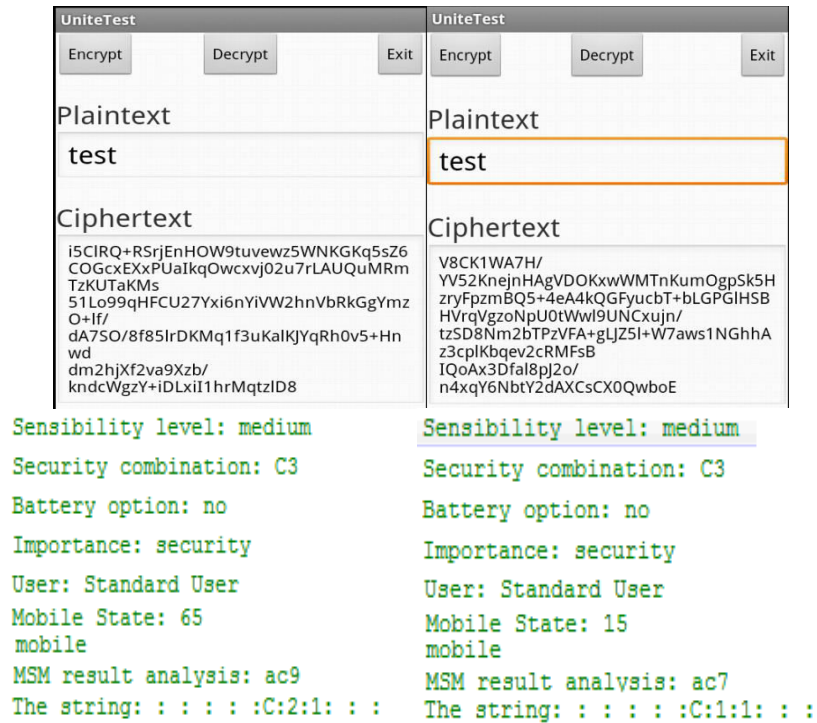


Figure 5.20 Results – Unite test second scenario (a)

It can be seen in Figure 5.21 that, for data with different sensibility level there are different security combinations (e.g. C7 [ac35] and C1 [ac2]). The mobile energy status also can influence the security algorithm (Figure 5.20, security combination ac9 or ac7).

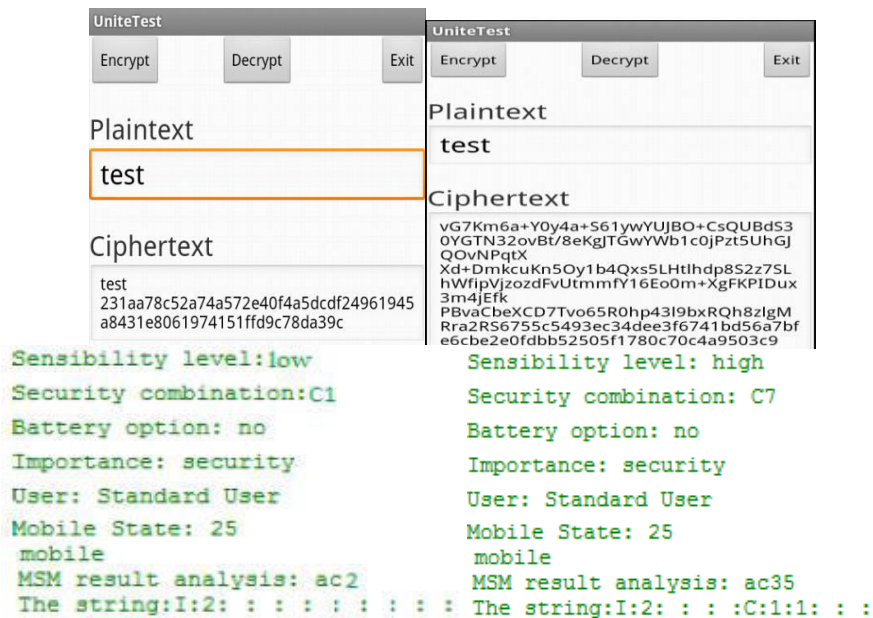


Figure 5.21 Results – Unite test second scenario (b)

5.7 *The Software Tools Used*

To implement the security framework on the mobile device, we have used Java programming language (more specifically Eclipse programming environment) and the Android mobile platform.

5.7.1 The Android Environment

Android is an operating system for mobile devices developed by Android Inc. company. In 2005, it was bought by Google, who, together with other member firms of the "Open Handset Alliance" work to the promotion and the development of the operating system. The result of this collaboration was basically a framework of libraries C/C++/Java, based on the Linux kernel.

The Android operating system has a multi-layer architecture:

- *The Linux Kernel:* Linux kernel was chosen because, when Android operating system was developed, it has been proved that it was a strong and stable core. Linux has a very good memory and processes management. It has integrated multiple functionalities such as the TCP/IP links that are necessary for an operating system (especially for data transfer)[Android1]. Moreover, the Linux kernel acts as an abstraction layer between the hardware and software [Android2].
- *The Runtime Libraries:* Android incorporates a set of libraries used by various components of the operating system and whose facilities are available to the programmer. Some of the libraries are the following: 1) System C library, 2) Media Libraries, 3) LibWebCore, 4) Scene Graph Library, 5) The 3D Libraries, 6) FreeType and 7) SQLite. Moreover, Android includes a set of core libraries that provides most of the functionalities available in the core libraries of the Java programming language [Android2].
- *The Application Framework:* By providing an open development platform, Android offers to the developers the opportunity to develop rich and innovative applications. The developers have access to the physical components of the device and to the locally stored information. They also have the possibility to run background services or to set alarms.

Security of Mobile Cloud Applications

- *The Applications:* Android comes with a set of embedded applications such as calendar, maps, browser, contacts, and others that were deemed necessary to any user. Java is the development language for the applications. Java was chosen because of its independence from hardware and software, and because Java was already available on most mobile phones [Android1]. Android has a very rich and standardized set of APIs and libraries. Developers are able to use these APIs for developing applications and Java makes these applications portable across any mobile phone with Android operating system [Android1].

5.7.2 Eclipse development environment

As previously mentioned, Android applications are developed using Java programming language. Android itself is not a language, but rather an environment where applications are running. So, theoretically, their development can use any distribution or integrated development environment (IDE). In fact, by using the command line interface, it is possible to not use any development environment.

Developers can choose from various development environments such as: JBuilder, Borland or NetBeans. Thought, Google and the Open Handset Alliance encourages the use of Eclipse development environment for the following reasons:

- Open Handset Alliance has developed a plug-in called Android development Tools (ADT) for Eclipse that provides facilities for the applications development, compilation and packaging [Dar10].
- The plug-in offers also the possibility to run and debug the applications on the emulator [Dar10].
- Eclipse, is available on both Mac and Linux, which allows the developers to create applications on any operating system [DiM08].

5.8 Conclusions

This chapter described the implementation of the Secure Mobile-Cloud Framework on the mobile device, more specifically the security components, the mobile security manager, the policy manager, the state manager and the mobile manager.

It has been pointed the way mobile security manager applies the appropriate security properties by presenting the pseudo code for one of the methods that belong to the MobileSecurityManager class.

Each security level received a code identifier; the connection between this code identifier and the security properties is known only by the policy manager. Therefore it has been described the role of each method contained by the PolicyManager class.

For the Mobile Manager it was presented the development of the analysis functionality.

This chapter has also presented the security framework user interface. The role of this user interface is to capture from the user information about: 1) the user type, 2) the security level the user wants to apply to his private data, 3) the security properties and the security algorithms chosen by the user, and 4) the user option regarding saving or not the mobile device energy.

In order to save the user choices a database was designed and implemented.

The implementation of the Secure Mobile-Cloud Framework was made using Android, Eclipse and SQLite.

At the end several unit tests were made as proof of concept for the implementation.

5.8.1 Contributions

In this chapter was proposed the implementation of the Secure Mobile-Cloud Framework on the mobile device. Also, it was proposed the framework user interface and the framework database. Finally, several unit tests were proposed in order to show the implementation functionality. These contributions were described and present in the following papers: [PBC+13c] and [PBB13e]. In the [PBC+13c] there was presented the sketch of the user interface; while in [PBB13e] was presented the whole implementation together with the unit tests.

5.8.2 Publications

[PBC+13c] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "A System to Analyze the User's Security Options for Mobile Cloud Applications", The 6th International Conference on Security for Information Technology and Communications, June 25, 2013.

[PBB13e] D. Popa, K. Boudaoud, M. Borda, "Secure Mobile-Cloud Framework - Implementation on the Mobile Device", in Acta Tehnica Napocensis, Electronics and Communications, 2013. (Submitted - accepted for publication)

Learning is the beginning of wealth. Learning is the beginning of health. Learning is the beginning of spirituality. Searching and learning is where the miracle process all begins.

Jim Rohn

6. The integration of Secure Mobile-Cloud Framework with a mobile cloud application

Contents in Brief

6.1 Mobile-Cloud Applications Examples	100
6.2 Mobile Cloud Application Scenario	102
6.3 Secure Mobile-Cloud Framework integration with the application scenario	107
6.4 Conclusions	118

Chapter Overview

This chapter presents the integration of the security framework into a mobile cloud application scenario.

The chapter is structured as follow: 1) a state of the art on the mobile cloud applications examples (Section 6.1); 2) the design of a mobile cloud application scenario (Section 6.2); 3) a solution to the integration of the security framework into the application scenario (Section 6.3). At the end, Section 6.4, the chapter conclusions are presented.

6.1 Mobile-Cloud Applications Examples

This section presents a short overview on the examples of mobile cloud applications.

Mobile Cloud Computing triggered the implementation of several applications that easily provide the users with more complex features and characteristics.

Some of the mobile cloud applications types are summarized in the following.

- *Commerce applications.* Those are business model applications that use the mobile device as a terminal to fulfill tasks like mobile transactions and payments, mobile messaging and mobile ticketing. These applications can be categorized into finance, advertising and shopping applications (Table 6.1).

TABLE 6.1 MOBILE COMMERCE APPLICATIONS [SGG+11]

Category	Examples
Mobile financial applications	Banks, brokerage firms, mobile-user fees
Mobile advertising	Sending custom made advertising according to user's physical location
Mobile shopping	Order certain products from a mobile terminal

- *Learning applications.* Developed combining electronic learning and mobile devices, those types of applications offer advantages like an improved communication between students and teachers or a quicker access to learning resources. In [Alabbadi11] is proposed mLaaS architecture. The presented architecture was implemented on iPhone as three layers architecture: the user and device layer, the services layer, and the infrastructure layer; it has the following features: transparency, collaboration, sharing of educational and learning resources, and personalized learning. In [SGG+11] is developed an education tool that allows the creation of a course about image and video processing. Using this application a learner can compile and understand algorithms that achieve face detection, de-noising, de-blurring and image enhancement.
- *Healthcare applications.* Over the last few years, were developed a very diverse range of health applications for mobile devices. The reason for such a development is the increasing use and capacity of mobile networks and devices (e.g. in some countries the mobile infrastructure is more used than fixed telecom network, also in certain countries the mobile phone is used by frontline healthcare workers [MCM11]). One

Security of Mobile Cloud Applications

of the principal objectives of using mobile technology in the health sector is to improve the access to care and resources (e.g. patient health records). The purposes of these applications can be diverse [DLN]: to monitor, to provide emergency management, to use health-aware devices (e.g. detect blood pressure), to offer access to healthcare information. In [SGG+11] is presented an application developed to provide on patients mobile phones, the prescriptions, the health records and the medical image records. In [KP] is described an application used to access and update patient's records, remotely; it facilitates the access to information for doctors and health workers and also the monitor of patients medication consumption; it was realized in collaboration with health care centers in Ghana and South Africa. In [BAD] is introduced an application that provide real-time crossing guidance to blind persons; it capture pictures of street intersections and offer real-time image processing to detect the location and the status of the immediate environment.

- *Games applications.* Mobile gaming has expanded at an impressive manner and become the fast growing sector of the gaming industry. Recent reports say: "the global market for mobile games will continue its rapid growth over the next years to eclipse 18 billion dollars in total revenue by 2016" [WD11]. Mobile gaming industry can benefit tremendously by using Cloud Computing features. In [WD11] is presented a technique that dynamically adjusts the games performances according to communication constraints and games demands. The basic idea in this paper is to reduce the number of objects in the display list because not all the objects created by the game engine are mandatory. The goal is to increase and improve the user experience given the communication and computing costs.
- *Other types of applications.* The Cloud offers the possibility to store and share photos and video clips. An application MeLog [LH10] was developed to enable mobile users to share real-time experience from traveling shopping or events. At the same time, mobile users are able to store video and images, and also to receive guidance or maps for trips. In [YCL10] is proposed a mobile location search service that allows users to shoot a short video clip with the surrounding buildings. The matching algorithm that runs in the Cloud will provide a summary of the location and services available (e.g. renting places). Google Wallet application [FK11] can carry a person's credit card, debit cards on an android phone. It allows using the phone to shop in-store or to shop online. This application has as purpose the wallet replacement; in this way is eliminated the need to take care of another object and also provide a faster way for paying.

6.2 Mobile Cloud Application Scenario

As it was presented in the previous section, the mobile cloud application that lately emerged are of various types and provide reach functionality in order to fulfill a various range of users requirements.

In the following section it is presented a proposed application scenario used in order to see how it may be integrated with the Secure Mobile-Cloud Framework.

6.2.1 Description

The scenario proposed is a healthcare application. This application aims to monitor a person type according to his/hers bodies characteristics and to propose a regime. The application initial characteristics and functionalities are presented in the following sub-section.

Application Characteristics

The application captures the users' characteristics: body size (e.g. height, weight). According to the information received a type of user is established. Depending of the user type, a certain regime is provided.

When a customer (simple user) is using the application the following actions will be performed. The application functionalities from the user point of view are (seen also Figure 6.1):

1. Create a user account if the user is using the application for the first time. In order to do this the user must provide several private data. This private data are: the name, password, e-mail.
2. If the user already has an account he/she need to login.
3. Insert the user data:
 - Personal data: day of birth and gender
 - Body measurements: height and weight
4. Obtain the user type. The user type is established by user's personal data and user's body size data. There are defined four types of users: weight-gain users, weight-loss users and normal users. The weight-gain users are too skinny and that they have to take in weight. The weight-loss users are too corporal and they have too loose weight. The normal users do not need to change their body size.

Security of Mobile Cloud Applications

5. Obtain the regime. The regime denotes the types of food the user can or cannot consume.

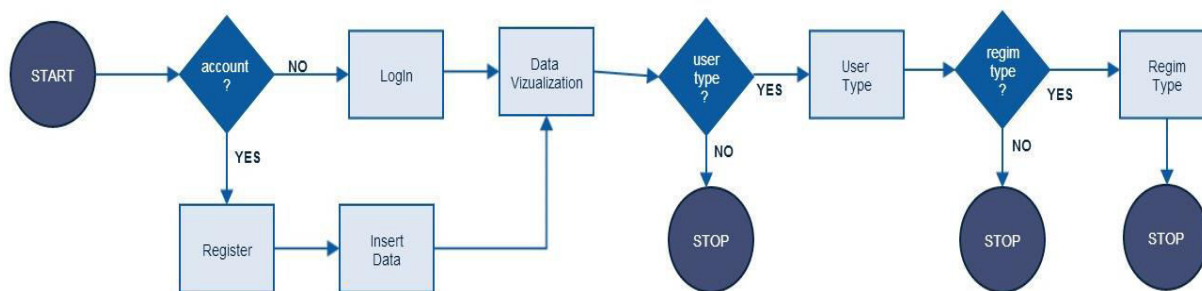


Figure 6.1 Application Scenario functionality

6.2.2 Development

This application is intended to be a Mobile Cloud Computing application, based on components running on the mobile side or in Cloud.

The application was split in three major parts. These parts are:

- *The user interface.* On the mobile device. It is designed in order to collect the users data.
- *The database.* On the Cloud side. It is designed in order to store the data.
- *The services.* On the Cloud side. They are designed in order to execute the following application functionalities: compute the user type, compute the BMI (body max index), and compute the regime. For each functionality there was designed a service.

As it can be seen in Figure 6.2 the user interface was designed to be a component; the database was also designed as another component; and each service represent a different component.

Security of Mobile Cloud Applications

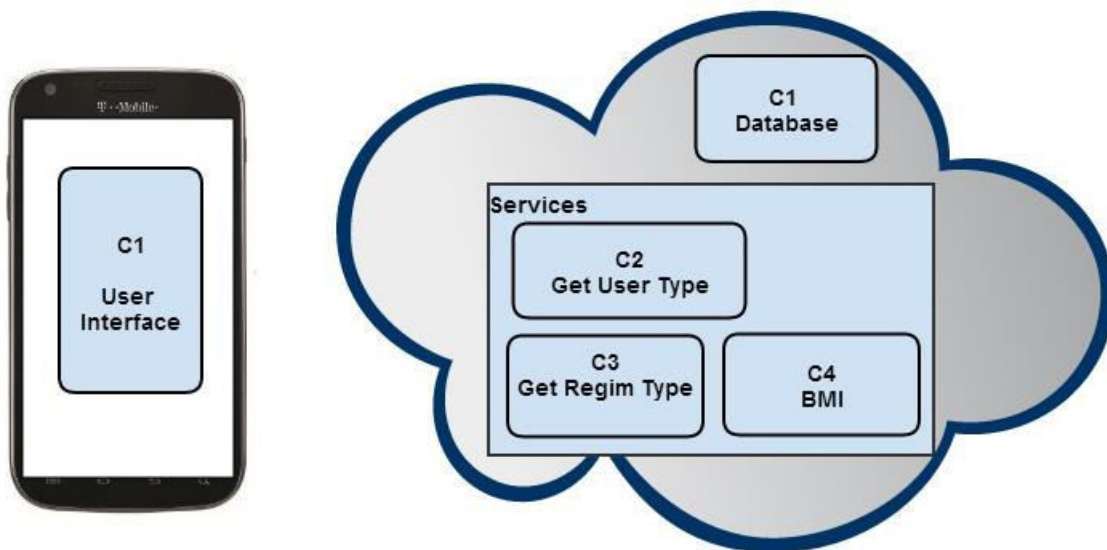


Figure 6.2 Application Scenario Components

As described in *Applications Characteristics*, the application user can perform the following actions: Login, Register, InsertData, DataVizualization, UserType, RegimeType.

In the following there are presented the interactions between the application components for each user action. Through components interaction it is understood the data exchange between the components.

For the following user actions: Login, Register, InsertData and DataVizualization the communication is made between the user interface and the database. Thus, the user interface collects user data and sends them in Cloud, in order to be saved in the database.

In the case of the LOGIN action, the user has to insert the following data: the e-mail and the password in order to login (Figure 6.3). These data are sent to the database to verify the account existence. If there is no account previous created, there will be an error message as response; and the user is redirected to perform the Register action (Figure 6.4). For the Register action, the user needs to provide the name, the e-mail and the password. A success message is received if there was made a registration in the database. The following action the user needs to perform is InsertData. This action assumes the registration in of the following data: day of birth, gender, height and weight into the database (Figure 6.5). Just as in the case of Register action a success message is received if data were registered into the database. DataVizualization action retrieves from the database the entire data particular to a certain user.

Security of Mobile Cloud Applications

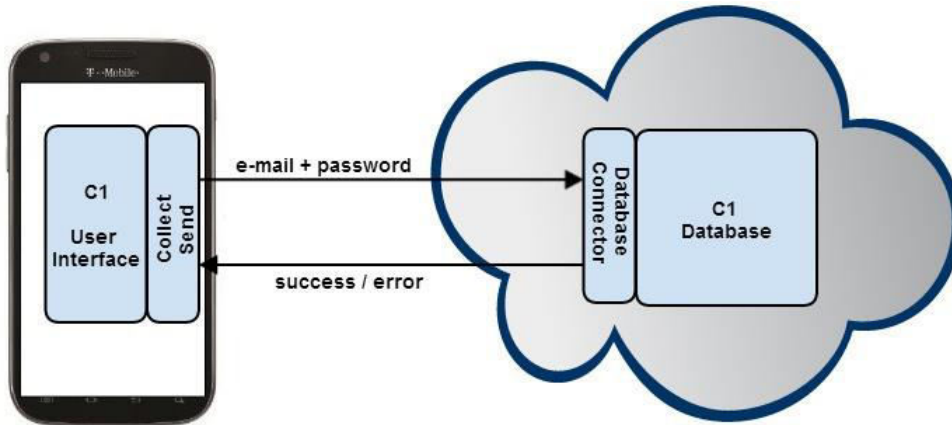


Figure 6.3 Log in action

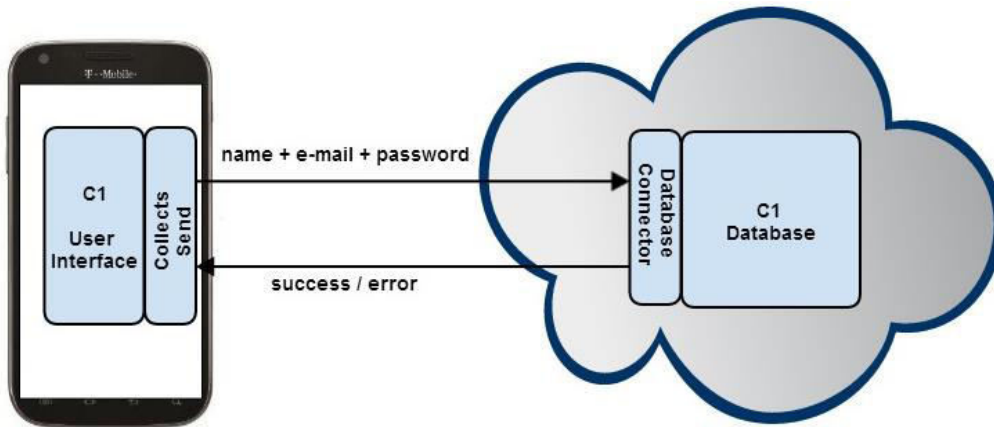


Figure 6.4 Register action

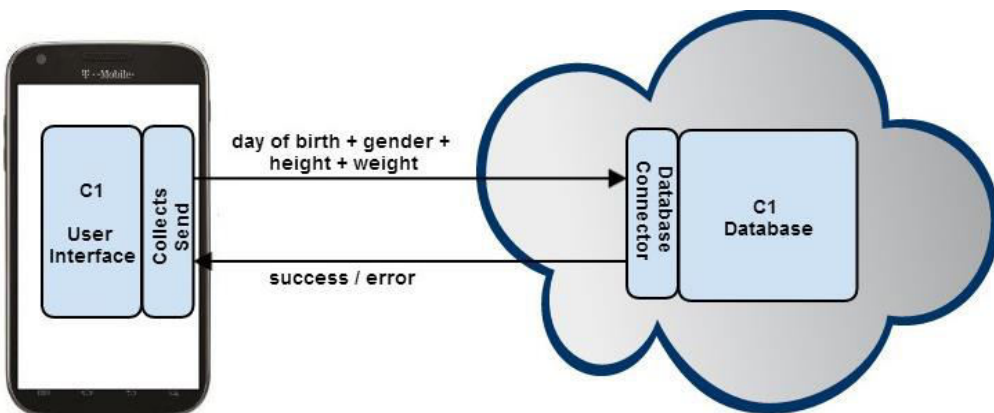


Figure 6.5 Insert Data action

Security of Mobile Cloud Applications

For the following user actions: UserType (Figure 6.6) and, RegimeType (Figure 6.7) the communication is made between the user interface component and services. Thus in the case of UserType action, GetUserType service receives from the mobile device the following data: weight and height, calls the BMI (body mass index) services and retrieves the user type information. In the case of RegimeType action, it is called by the mobile device the GetRegimType services, which, according to the user type information received retrieves information about the types of foods the user may and may not consume.

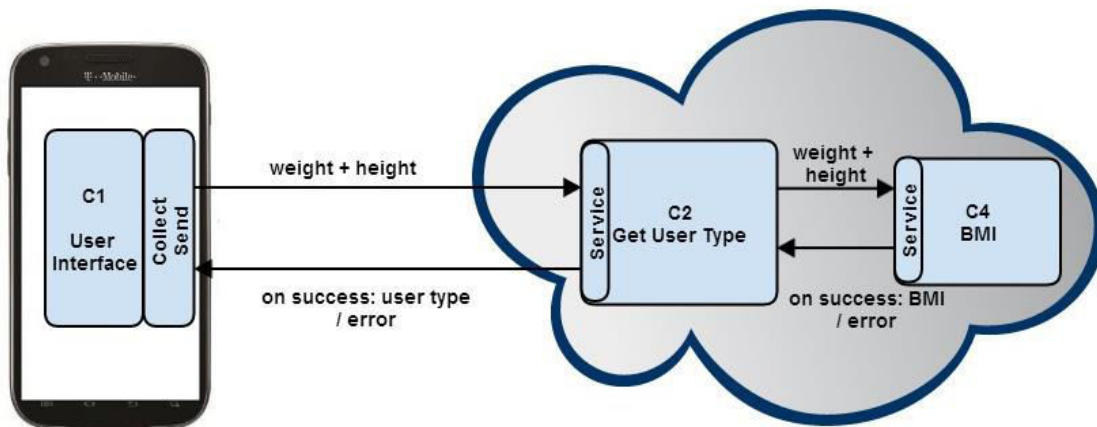


Figure 6.6 User Type action

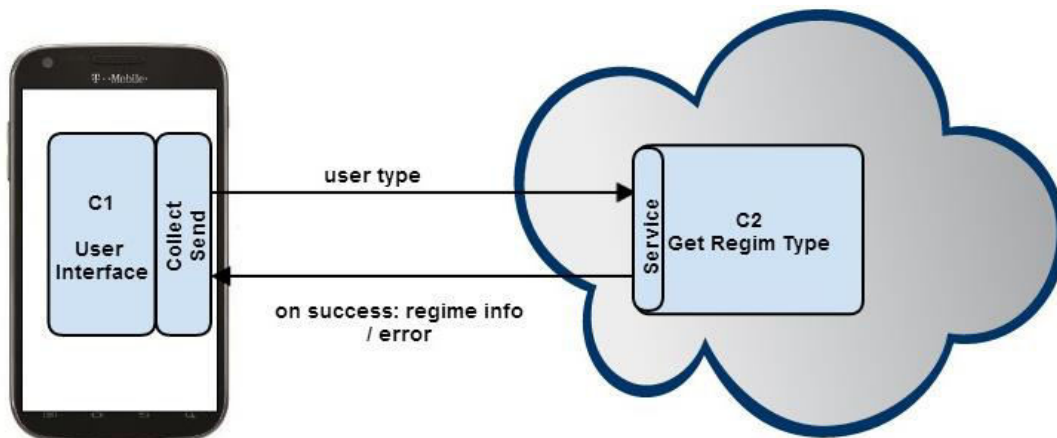


Figure 6.7 Regime Type action

Security of Mobile Cloud Applications

Regarding the mobile cloud application components design, the following observations were made:

The design of the user interface component consists of two parts: 1) a first part that is the proper interface, with the role of collecting data from the users or displaying data from the database; and 2) a second part called CollectSend, with the role of communication between the mobile device and the database or services in the Cloud.

Also, in the case of the database design there was created a part called DatabaseConnector. Its role is to respond to various requests made by the mobile device. These requests are of saving data into the database or retrieving data from the database. The DatabaseConnector response may be a simple message or a collection of data from the database.

So far, it has been described the application scenario, it has been showed the components proposed to be part of the application, it has been described how the component interact in order to fulfil an user action and it has been shown the data exchanged between the components.

Furthermore, in the previous chapters it has been presented the design of a security framework whose objectives are to secure data communication between mobile cloud application components taking into account the mobile device energy constraints, the user constraint and the data sensitivity.

The questions that rise now is how to combine these two parts together, and where in the application scenario should the security framework take action. An answer to these questions is given in the following section.

6.3 Secure Mobile-Cloud Framework integration with the application scenario

The term integration denotes the combination of the application scenario with the security framework. Furthermore, it refers to the way in which the application scenario will operate using the security framework.

For the integration solution it is assumed that there is access to the application scenario source code. The proposed solution for the integration is a static solution. A

Security of Mobile Cloud Applications

static solution assumes the integration at the source code level before the application scenario compiling and deploy. It is also assumed that the keys (public and secret) are securely distributed.

This section is split in two parts. The first part presents the theoretical approach of the integration; the second part is the technical approach.

6.3.1 Theoretical Approach

As previously said the integration assumes the combination of the application scenario with the security framework. In this way it is obtained a new application, the application scenario secured; an application that has the functionalities of the application scenario, but that it also has the data security ensured.

The problems that need to be resolved here are the following: 1) where and how the application scenario will call the security framework to secure the data; 2) which is the flow of the new application scenario secured. A solution to these problems is described in the following:

As solution to the first issue, for each component, on the mobile device (user interface) or in Cloud (services), it was designed a part, called ApplySecurity, in order to communicate with Mobile Manager, in the case of the user interface, or with Cloud Security Manager in the case of a service in Cloud.

A solution to the second issue is described below:

From the description of the scenario application in the previous section it can be seen that the application includes three types of communications: 1) the communication between the mobile device and the database; 2) the communication between the mobile device and the services; and 3) the communication between two services.

The communication between the mobile device and the database assumes the following actions: 1) saving data provided by the application user into the database (e.g. Register action or Insert Data action); and 2) requesting data from the database in order to be displayed on the application interface (e.g. Data Visualization action).

The communication between the mobile device and the services assumes calling the services and providing the proper users' private data in order to obtain a certain result; the result may be or not a users' private data.

The communications between two services refers to one service calling the other services and providing the proper data (private or not) in order to obtain certain data as result (private or not).

Security of Mobile Cloud Applications

The secured application scenario flow in the case of communication between the mobile device and the database requires several steps (Figure 6.8). The following steps are followed in the case a certain level of security is applied:

S-1. Users' private data, retrieved by the user interface, which need to be sent into the Cloud database are intercepted by the ApplySecurity part and sent to the Mobile Manager along with their sensitivity level;

S-2. The Mobile Manager, uses its analysis functionality to verify where is more suitable to apply the security, on the mobile device or in Cloud;

S-3. If the mobile device is chosen by the Mobile Manager, the data along with the security level (the combination of security properties along with the security algorithms) are sent to the Mobile Security Manager;

S-4. After discovering which security properties and security algorithms correspond to the received security level, the Mobile Security Manager orchestrates the application of the appropriate security properties (components) to the received data;

S-5. When the security operation is finished, the secured data are sent back to Mobile Manager;

S-10. The secured data along with the security level are sent to the ApplySecurity part;

S-11. The secured data are sent to the Cloud database;

S-12. A response message is received;

S-6. If the Cloud is chosen by the Mobile Manager, data along with the security level are ciphered using a secret key;

S-7. Encrypted data are sent to the Cloud Security Manager;

S-8. After decrypting the received data, Cloud Security Manager orchestrates the application of the appropriate security properties (components) to the received data;

S-9. When the security operation is finished, the secured data are sent back to Mobile Manager;

Then, the steps from S-10 to S-12 are followed.

Security of Mobile Cloud Applications

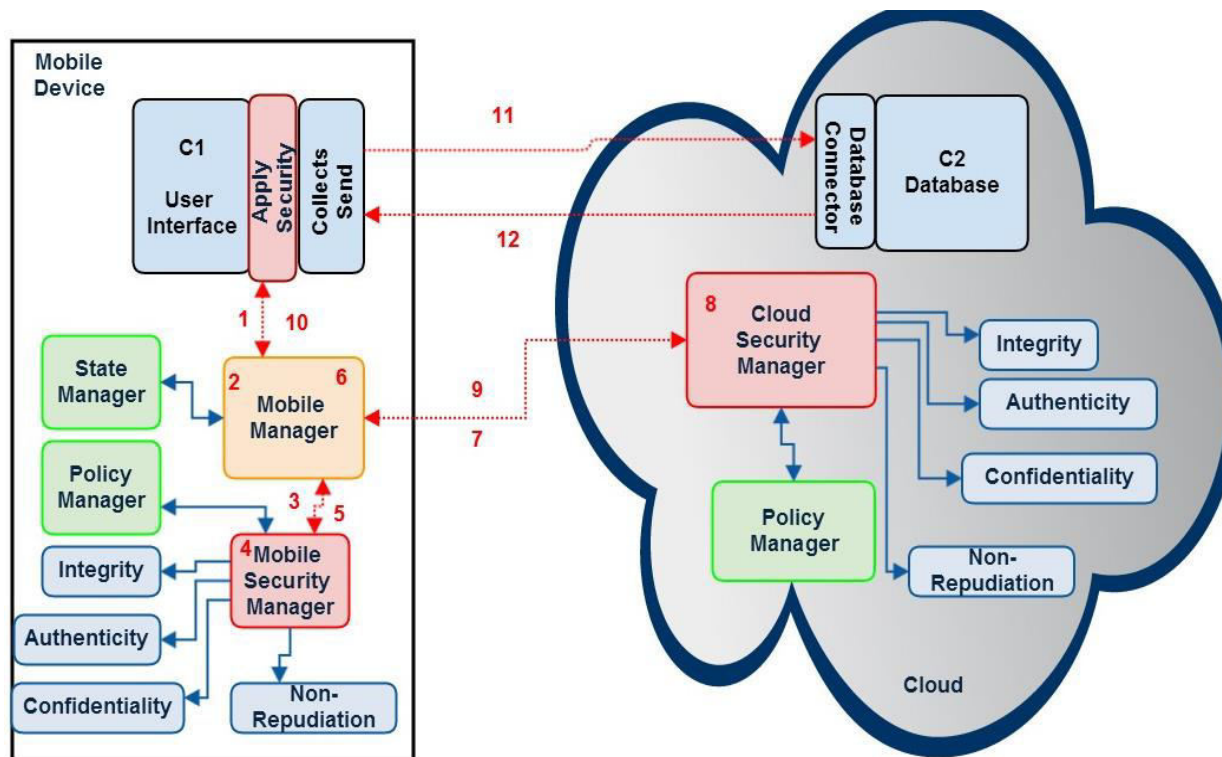


Figure 6.8 Application scenario secures Flow – Communication between mobile device and database

The secured application scenario flow in the case of communication between the mobile device and the services in the Cloud requires several steps (see Figure 6.9), as follow:

- S-1. The users' private data intercepted by the ApplySecurity part are sent to the Mobile Manager along with their sensibility level;
- S-2. The Mobile Manager, uses its analysis functionality to verify where is more suitable to apply the security, on the mobile device or in Cloud;
- S-3. If the mobile device is chosen by the Mobile Manager, the data along with the security level (the combination of security properties along with the security algorithms) are sent to the Mobile Security Manager;
- S-4. After discovering which security properties and security algorithms correspond to the received security level, the Mobile Security Manager orchestrates the application of the appropriate security properties (components) to the received data;
- S-5. When the security operation is finished, the secured data are sent back to Mobile Manager;

Security of Mobile Cloud Applications

S-10. The Mobile Manager attaches the security level valued to the received data and send them to the ApplySecurity part;

S-11. Then, the ApplySecurity send the secured data along with the security level to the components (services) in Cloud;

S-12. The ApplySecurity part on the service side intercepts the data sent from the mobile device and sends them to the Cloud Security Manager in order to obtain the actual data;

S-13. According to the data security level received, Cloud Security Manager orchestrates the security properties in order to obtain the actual data;

S-14. The data, obtained in step S-13, are encrypted by the Cloud Security Manager with the component (service) secret key and then are sent to it as response (S-15);

S-16. The component (service) decrypts the received data using its private key, and uses them to perform its functionality;

S-17. Before sending the data (results) back to the components on the mobile device, they also have to be secured. This securing process is as following:

The data along with the sensibility level are encrypted with the component (service) private key and sent to the Cloud Security Manager. The Cloud Security Manager establishes the security level to be applied to the data and orchestrates security components. The secured data along with the security level applied are sent back to the component (service), which sends them, to the component on the mobile device.

S-6. If the Cloud is chosen by the Mobile Manager, data along with the security level are ciphered whit using a secret key;

S-7. The Encrypted data are sent to the Cloud Security Manager;

S-8. After decrypting the received data, Cloud Security Manager orchestrates the application of the appropriate security properties (components) to the received data;

S-9. When the security operation is finished, the secured data are sent back to Mobile Manager;

Then, the steps from 10 to 17 are followed.

Security of Mobile Cloud Applications

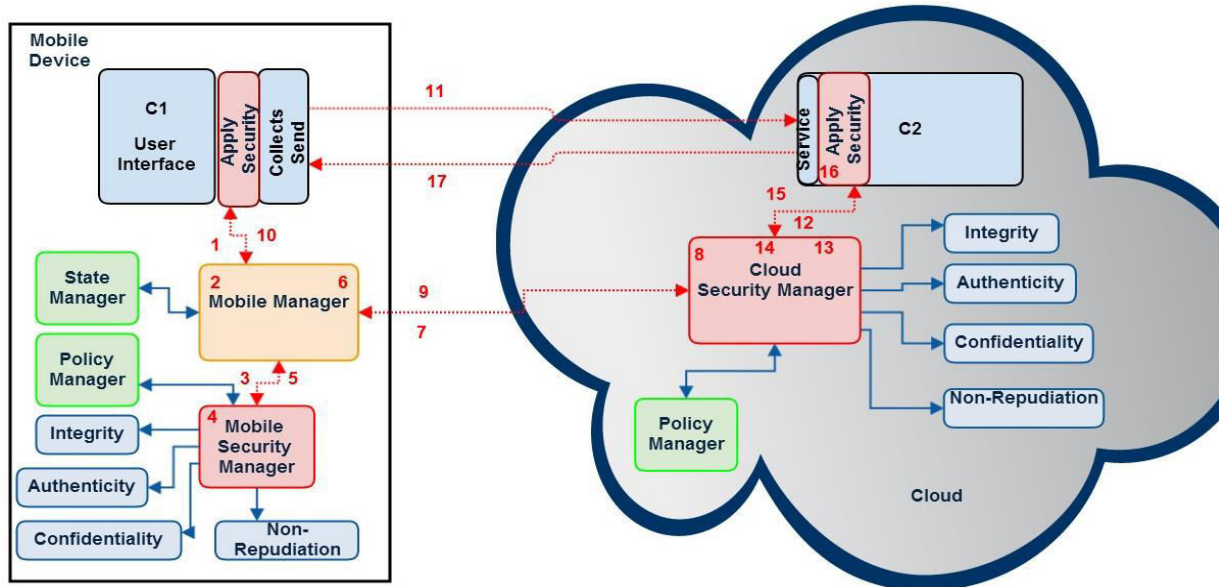


Figure 6.9 Application scenario secures Flow – Communication between mobile device and service component

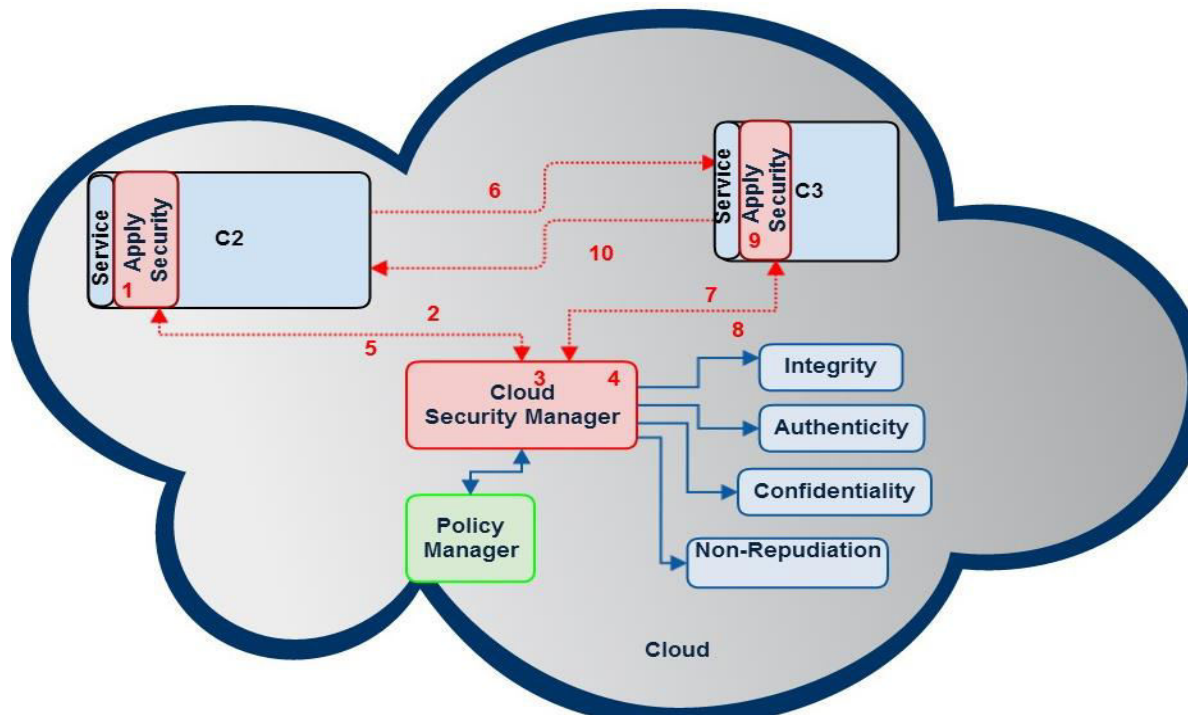


Figure 6.10 Application scenario – the secured Flow – Communication between two services components in Cloud

The secured application scenario flow in the case of communication between two components (services) in Cloud (see Figure 6.10) is as follow:

Security of Mobile Cloud Applications

The data along with the sensibility level are encrypted with the component (service) private key (S-1) and sent to the Cloud Security Manager (S-2). The Cloud Security Manager establishes the security level to be applied to the data and orchestrates security components (S-4). The secured data along with the security level applied are sent back to the component (service) (S-5), which sends them, to the second component (service) (S-6).

In order to use the received data, the second component (service) in Cloud, sends the received data to the Cloud Security Manager (S-7). According to the data security level received, Cloud Security Manager orchestrates the security properties in order to obtain the actual data (S-4). Then encrypts the data with the component private key (S-3) and then are sent to the component as response (S-8). The component (service) decrypts the received data using its private key, and uses them to perform its functionality. Before sending the data (results) back to the first component (service) they also have to be secured. In order to do that the same mechanism as in S-17 (communication between the mobile device and a service) is applied.

6.3.2 Technical Approach

This section includes the application scenario implementation and it shows from a technical point of view where in the application scenario it has intervene in order to add the security framework.

At this time from the application scenario it has been implemented only the communication between the mobile device and a database stored in Cloud. The implementation has been done in conjunction with a team of students from Politech Nice Sophia Antipolis [RP13].

Implementation overall description

Three of the user actions presented in the application scenario were implemented: LogIn, Register and InsertData.

The main languages that were used to implement the application scenario are Java, MySQL, PHP and JavaScript Object Notation (JSON).

The interface on the mobile device was implemented using Java based on Android, PHP was used in order to access and query the database and for the database

Security of Mobile Cloud Applications

implementation was used MySQL. Then, in order to make a connection between the mobile device and the server the JSON data-interchange format was used.

The implementation was made according to the design described in this chapter section 6.2. For the communication between the mobile device and a database in cloud the following the application design consists of: the user interface, the collect/send part, the connector part and the database. Thus, the user interface consists of three screens: LogIn screen, Register screen and Details screen. The collect/send part consists of two classes implemented in Java: UserFunctions and JSONParser. The connector part consists of two files: index and DBFunctions implemented in PHP. The database is implemented in MySQL.

The UserFunctions class was implemented in order to send the user data, retrieved from the user interface, to the Cloud database. It implements the following methods:

- loginUser(): it implements the login requests
- registerUser(): it register the user login details (name, e-mail and password) into the external database
- registerUserInfo(): it register the user other info like: gender, height, weight, day of birth into the external database

The JSONParser class implements the following method:

- getJSONFromUrl(String url, List<NameValuePair> params): it passes and receive the data from and to the server side implemented in PHP

The index file handles all the request coming from the mobile device. The DBFunctions include the implementation for DBFunctions class methods. The class implements two methods: storeUser and storeNewUserInfo. Their role is to implement the operations of insert into the database.

The database contains two tables:

- users: to store the register data: name, e-mail and password
- info: to store the users information like gender, height, weight and day of birth

Security of Mobile Cloud Applications

The changes into the application scenario source code

The following section will present: 1) how were defined and integrated, into the application code, the constraints that the application developer should specify; 2) in what consist the ApplySecurity part mentioned in Theoretical approach section; and 3) the modifications made to the user interface.

The constraints that the application architect needs to specify are the application type and the data sensitivity level.

For the application type constraint, the solution applied was to use in the application manifest file the attribute 'description'. This description attribute will take the application type value; for this particular application scenario the value is health. To retrieve the attribute description value, it was implemented into the main user interface page (UIMainActivity.java) a method called *getApplicationType* (see Figure 6.11). The *getApplicationType* method uses the style resource identifier *getApplicationInfo.descriptionRes*, provided by *android.content.Context* in order to identify in the application resources the attribute description value.

```
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:description="@string/app_type"
    android:theme="@style/AppTheme" >
    <activity
        android:name="com.app.sampleUI.StartApiSampleActivity"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    ...
    app_name = getApplicationName(getApplicationContext());
    app_type = getApplicationType(getApplicationContext());
    public static String getApplicationType(Context context) {
        int stringId = context.getApplicationInfo().descriptionRes;
        return context.getString(stringId);
    }
```

Figure 6.11 Application type constraint - solution

For the sensitivity level constraint, the solution applied was to define several private static variables into the class (UserFunctions) that is in charge of the data

Security of Mobile Cloud Applications

transmission. These variable are: add_security, highS, mediumS and lowS as it can be seen in the Figure 6.12.

```
/*for security*/  
private static String add_security = "add_security";  
private static String highS = "high";  
private static String mediumS = "medium";  
private static String lowS = "low";  
/*for security*/
```

Figure 6.12 Sensitivity level constraint - solution

In the theoretical approach section was mentioned the ApplySecurity part which the functionality of collecting the user private data and sending them to the Mobile Manager.

ApplySecurity part and its functionality were implemented into each UserFunction method that sends data to the database as follow:

While the list whit pairs of type <parameter,value> is build, it is inserted before each pair an additional pair of type <S_level, value> as it can be seen in the Figure 6.13. After the list is built, the apply_security_encrypt() method in MobileManagerPartB class is called. This method will return a list of type <parameter,value>, where the values are secured.

Then the data may be sent to the database.

```
params.add(new BasicNameValuePair("security", add_security));  
params.add(new BasicNameValuePair("tag", register_tag));  
params.add(new BasicNameValuePair("num_reg", num_regist1));  
params.add(new BasicNameValuePair("S_level", highS));  
params.add(new BasicNameValuePair("name", name));  
params.add(new BasicNameValuePair("S_level", highS));  
params.add(new BasicNameValuePair("email", email));  
params.add(new BasicNameValuePair("S_level", highS));  
params.add(new BasicNameValuePair("password", password));  
  
params_json = SMC_manager.apply_security_encrypt(context,db_app_name,params);  
  
JSONObject json = jsonParser.getJSONFromUrl(registerURL, params_json);
```

Figure 6.13 ApplySecurity part – solution on the mobile device

For the user interface modification; it was introduce a new main user interface from which the user may choose to set the security settings or to start the application. In the case the user won't set the security settings, it will be applied to all data without taking into consideration data sensitivity level, the default security level for a standard user type.

Security of Mobile Cloud Applications

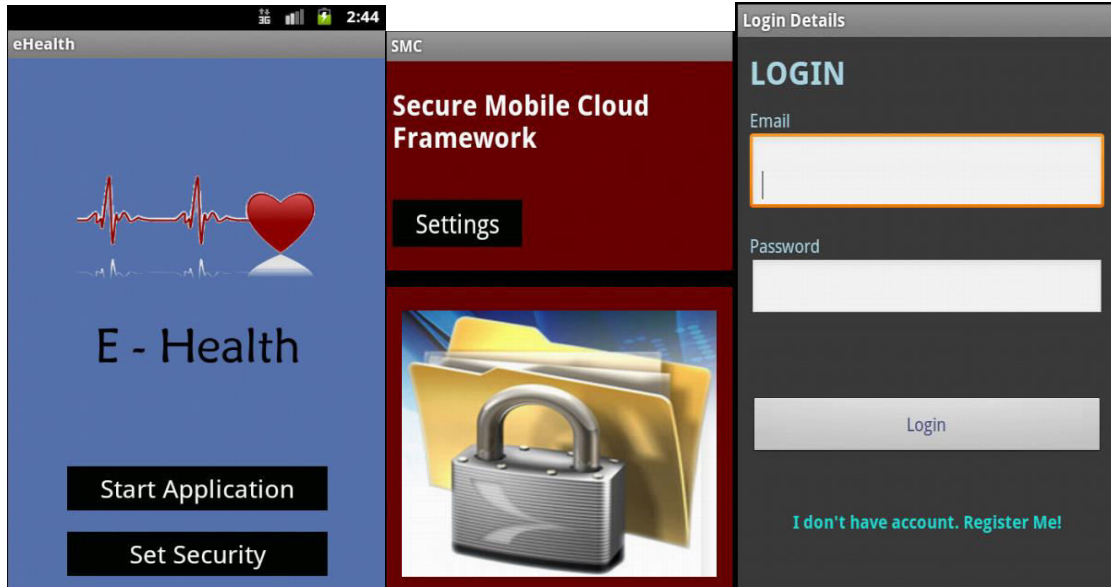


Figure 6.14 Modified user interface

After performing the Register user action data saved in the user database were secured (see Figure 6.15)

name	email	password
xNFt1zLE5EFtyoj4PqWDJVsFurXIS/	DB6MsfPaARvGuXJ1urqfK1RWiOvLtx	14x/dElahyZh2WTuNHUVYSCjxDYK4y

Figure 6.15 Private user data save into the database

6.4 Conclusions

Several mobile cloud applications have been developed lately in various domains like: commerce, learning, healthcare, or gaming.

A healthcare mobile cloud application scenario was presented in this chapter. The application aims to monitor a person type according to her/his bodies' characteristics and to propose a regime. The application was split into components. The user interface was designed as a mobile device component, while the database was designed as a Cloud component. Three services running as components in the Cloud were also designed.

In this chapter was also presented a solution to integrate the Secure Mobile-Cloud Framework into the healthcare application scenario. The intended result was to obtain only one application. The new obtained application has the healthcare application functionalities, but it also has the security of data transmitted between components. The integration is made at the source code level.

There were established three cases of communications: 1) between the mobile device and a database in Cloud; 2) between the mobile device and a service in Cloud; and 3) between two services in Cloud. A theoretical approach was proposed for each case of communication. A technical approach was proposed for the communication between the mobile device and the database in Cloud.

The theoretical approach shows where in the application scenario flow it is needed to be done the call to the Secure Mobile-Cloud Framework components. The technical approach, show the modifications brought to the application scenario source code in order to integrate the security framework.

6.4.1 Contributions

The main contributions of this part are: the overview of the existing mobile cloud application, the design of the healthcare application scenario, and the integration solution both theoretical and technical approach. The contributions are described and presented in the following papers: [PBC+13d] and [PBC+13f].

6.4.2 Publications

[PBC+13d] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "Mobile Cloud Applications and Traceability", in Proceedings ROEduNet 12 th International Conference, Constanta, 26-28 September, 2013.

[PBC+13f] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "Integration of Secure Mobile-Cloud Framework into a mobile cloud application scenario", in Scientific Bulletin of the Politechnica University of Timisoara, Transactions on Electronics and Communications, 2013 (Submitted - accepted for publication).

Security of Mobile Cloud Applications

“Conclusions are not always pleasant.”

Helen Keller

7. Overall Conclusions

Contents in Brief

7.1 Contributions Overview.....	122
7.2 Future Works	123

This work is a research on the security of component based mobile cloud applications; more accurate on the security of data transmitted between the same mobile cloud application components.

Mobile Cloud Computing is a novel model, developed as a solution to the mobile devices constraints. It emerged after the Cloud Computing model was introduced; and it is one of the benefits resulted from the way Cloud Computing delivers the IT resources.

Thereby, using Mobile Cloud Computing, new application models were developed for mobile devices. These applications models try to use both the mobile device resources and the Cloud services to provide a more reach and a more varied functionality in order to increase the mobile device popularity and use.

The development of mobile cloud application assumes the involvement of mobile devices and the Cloud resources. Because of this fact, it was shown in the state of the art presented into this work that from a security point of view a wide number of security issues arise. The security issues presented into these work state are of four categories: mobile threats, cloud threats, mashup threats and technological threats. In order to solve the security issues both mobile platforms and Cloud providers offered

Security of Mobile Cloud Applications

various solutions. One direction of the security solutions is to secure data and applications on the mobile device; another direction of the solutions is to secure the data stored in the Cloud or to secure various types of newly emerging applications; a different direction is to secure the data transmitted between the mobile device and the Cloud. The solutions proposed to resolve the security issues, treat independently each type of security problems. Thereby, in the case of components base mobile cloud applications, is needed to combine different solutions in order to secure them. Also, very few solutions take into account the mobile device energy constraints; or the users' expectations regarding the security level that should be applied to their data; or even the data sensibility.

The security solution proposed in this work focuses on component based mobile cloud applications. Its goal is to secure the communication between the application components. The feature of this solution is the fact that it tacks into consideration the following constraints: mobile device energy, data sensitivity and users' options. Thus, this solution can adapt to different kind of mobile devices capabilities; it is flexible to the mobile devices users' needs and also can adapt the security solution to the data sensibility level.

7.1 Contributions Overview

In the beginning was made a state of the art on Cloud Computing followed by a state of the art on Mobile Cloud Computing [PBC+13a], [PBC+13g]. The purpose of these studies is to point out which are the security issues and the existing solution for mobile cloud applications. The studies have also as purpose to give a research direction for this thesis.

In the following there are presented the thesis main contributions along with the related publications.

Chapter 4 presents the design of the Secure Mobile-Cloud Framework. It was designed to fulfill these features: 1) to secure the data used by a component based mobile cloud application; 2) to apply different security properties to different kinds of data (according to the data sensibility level) and not the same properties to all the data processed by the application; 3) to taken into consideration the user preferences and 4) to take into consideration the mobile device energy consumption. The framework design is presented in [PBC+13a]. The criteria that the security framework should take

Security of Mobile Cloud Applications

into consideration were presented in [PBC+13b]. The analysis functionality of Mobile Manager was presented in paper [PBC+13c].

In Chapter 5 it is described the implementation on the mobile device of the Secure Mobile-Cloud Framework presented Chapter 4. There were implemented the classes that provide the security properties functionalities. The Mobile Security Manager, Policy Manager and State Manager were also implemented. The system analysis functionality was implemented within the Mobile Manager. Furthermore the user interface windows were designed and implemented. Also the databases used to store the user choices were designed and implemented. At the end several unit tests were implemented in order to test the implementation functionality.

A part of the user interface design was presented in [PBC+13c], the full implementation on the mobile device was presented in [PBB13e].

In Chapter 6 it is presented an integration solution. The integration refers to the combination of a mobile cloud application with the Secure Mobile-Cloud Framework. There were established three cases of communications: 1) between the mobile device and a database in Cloud; 2) between the mobile device and a service in Cloud; and 3) between two services in Cloud. A theoretical approach was proposed for each case of communication. A technical approach was proposed for the communication between the mobile device and the database in Cloud. An overview of the existing mobile cloud applications was made and it was presented in the paper [PBC+13d] along with part of the application scenario design. The integration solution was presented in [PBC+13f].

7.2 Future Works

The work presented is just a beginning on the field of mobile cloud applications security. There are still many future works and improvements that can be done. One of the most important future works is to make practical experiments. To measure the mobile devices energy consumption and the secured application scenario response time, more precisely the time needed to apply the security properties and to communicate with the services in Cloud.

Then, improvements can be made to the Mobile Manager's analysis functionality as: 1) to introduce new constraints in order to be taken into account when a security solution is provided; 2) to change the '*if then else*' rules with a dynamic analysis

Security of Mobile Cloud Applications

solution, since the *'if then else'* rules need to be change every time a new constraint is added.

Other future works would be to change the way the integration solution is done. The static solution proposed into this work, solution that needs the application scenario source code, can be improved and changed into a dynamic one. A dynamic solution that does not needs the source code and which can be applied for any component based mobile cloud application.

Bibliography

- [AEC2] Amazon Elastic Compute Cloud (Amazon EC2), <http://aws.amazon.com/ec2/>
- [AEPONA10] *White Paper*, "Mobile Cloud Computing Solution Brief," AEPONA, November 2010.
- [AFG+09] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, "Above the clouds: a berkeley view of cloud computing", EECS Department, University of California, Berkeley, in *Tech. Rep*, UCB/EECS-2009-28, 2009.
- [AG] Personal Data Protection Act No. 25,326, (Arg.), *available online* www.privacyinternational.org/countries/argentina/argentine-dpa.html, Oct. 4, 2000.
- [Agarwal11] C. Agarwal, "Concepts, Challenges and Opportunities of Cloud Computing for Business Analyst", *AKGEC International Journal of Technology*, Vol. 2, No. 2, 2011.
- [Alabbadi11] M. M. Alabbadi, "Mobile Learning (mLearning) Based on Cloud Computing: mLearning as a Service (mLaaS)", in: *UBICOMM 2011: The Fifth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, 2011.
- [Android1] The Windows Club, "What is Android operating system? A beginners read!", *available online*: <http://www.thewindowsclub.com/what-is-android-operating-system-a-beginners-read>
- [Android2] Development, "Android, the world's most popular mobile platform", <http://developer.android.com/guide/basics/what-is-android.html>
- [BAD] B. Bhargava, P. Angin, L. Duan, "'A Mobile-Cloud Pedestrian Crossing Guide for the Blind"
- [BBK11] F.Batard, K.Boudaoud,M.Riveil, "Middleware for secure Mobile mashup", 2011.
- [BFH03] F. Berman, G. Fox, and A. Hey, "Grid Computing: Making the global infrastructure a reality", Wiley, 2003.
- [BGG94] M. Bellare, O. Goldreich, S. Goldwasser, "Incremental cryptography: the case of hashing and signing", in: *Proc. 14th Annual Int. Cryptology Conference on Advances in Cryptology*, Santa Barbara, California, USA, Aug. 1994.

Security of Mobile Cloud Applications

- [BGG95] M. Bellare, O. Goldreich, S. Goldwasser, "Incremental cryptography and application to virus protection", in: *Proc. 27th Annual ACM Symposium on Theory of Computing, STOC '95*, Las Vegas, NV, USA, May 1995.
- [BLR+11] M. Bohm, S. Leimeister, C. Riedl, H. Krcmar, "Cloud Computing and Computing Evolution", San Murugesan (ed.) *Cloud Computing: Technologies, Business Models, Opportunities and Challenges*, CRC Press, 2011
- [BM09] G. Briscoe, A. Marinos, "Digital Ecosystems in the Clouds: Towards Community Cloud Computing", in *3rd IEEE International Conference on Digital Ecosystems and Technologies*, New York, USA, pp. 103-108, 2009.
- [BYV08] R. Buyya, C. S. Yeo, S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities", *10th IEEE International Conference on High Performance Computing and Communications, HPCC '08*, pp. 5-13, 2008.
- [Castillo11] C.A. Castillo, "White Paper: Android Malware Past, Present and Future", Mobile Security Working Group, McAfee, *available online*
<http://www.mcafee.com/us/resources/white-papers/wp-android-malware-past-present-future.pdf>, 2011.
- [CBC+10] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, P. Bahl, "MAUI: Making Smartphones last longer with Code Offload", in *MobiSys'10*, June 15-18, San Francisco, California, USA, 2010.
- [CDB11] R. Choubey, R. Dubey, J. Bhattacharjee, "A survey on cloud computing security, challenges and threats", *International Journal on Computer Science and Engineering* Vol. 3, 2011.
- [Chetan10] S. Chetan, G. Kumar, K. Dinesh, K. Mathew and M.A. Abhimanyu "Cloud Computing for Mobile World", *available online*:
<http://chetan.ueuo.com/projects/CCMW.pdf>, 2010.
- [Christensen09] J. H. Christensen, "Using RESTful web-services and cloud computing to create next generation mobile applications," in *Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications (OOPSLA)*, pp. 627-634, October 2009.
- [CL12]. R. D. Caytiles, S. Lee, "Security Considerations for Public Mobile Cloud Computing", *International Journal of Advanced Science and Technology* Vol. 44, July, 2012

Security of Mobile Cloud Applications

- [CM09] B.G. Chun and P. Maniatis, "Augmented Smartphone Applications Through Clone Cloud Execution," in *Proceedings of the 12th Workshop on Hot Topics in Operating Systems (HotOS XII)*, Monte Verita, Switzerland: USENIX, 2009.
- [CNet] CollabNet, *available online*: <http://www.collab.net/>
- [CNW07] B. Chess, Y. T. O'Neil, J. West, "JavaScript Hijacking", Fortify Software, *available online*: http://james.padolsey.com/wp-content/uploads/javascript_hijacking.pdf , March, 2007.
- [Creasy81] R. Creasy, "The origin of the VM/370 time-sharing system", *IBM Journal of Research and Development* Vol. 25 pp. 483-490, 1981.
- [CPK10] Y. Chen, V. Paxson, R. H. Katz, "What's New About Cloud Computing Security?", Technical Report No. UCB/EECS-2010-5, *available online*: <http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-5.html>, 2010.
- [CSA09] A. Archer "Boehm, Security guidance for critical areas of focus in cloud computing", Cloud Security Alliance, 2009.
- [CWR10] V. Chang, G. Wills, D. De Roure, "A review of cloud business models and sustainability", 2010.
- [CISCO] "Cisco Cloud Computing - Data Center Strategy, Architecture, and Solutions", Point of View *White Paper* for U.S. Public Sector, 1st Edition, Cisco Systems, 2009.
- [Danielson08] K. Danielson, "Distinguishing Cloud Computing from Utility Computing", *available online*: <http://www.ebizq.net/blogs/saasweek/2008/03/distinguishing-cloud-computing/>, 2008.
- [Dar10] L. Darcey, S. Conder: "Teach yourself Android application development in 24 hours", Sams, 2010
- [DLN+11] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches", *Accepted in Wireless Communications and Mobile Computing - Wiley*, 2011, *available online*: <http://onlinelibrary.wiley.com/doi/10.1002/wcm.1203/abstract>
- [DiM08] J. DiMarzio: "Android - A programmer's guide" , McGraw Hill, 2008
- [Eilers12] C. Eilers, "HTML5 Security", *Developer.Press*, October 11, 2012.
- [EKH10] D. S. Abd Elminaam, H. M. Abdual Kader, M. M. Hadhoud, "Performance Evaluation of Symmetric Encryption Algorithms", *International Journal of Network Security*, Vol.10, No.3, pp. 216-222, May 2010.

Security of Mobile Cloud Applications

[ENISA09] D. Catteddu, G. Hogben, "Benefits, risks and recommendations for information security", European Network and Information Security Agency, 2009.

[EU95] EU Directive 95/46/EC of the European Parliament and the Council of 24 October 1995 on the Protection of Individuals with Regard to the Processing of Personal Data and the Free Movement of Such Data, 1995 O.J. L 281

[FK11]. R. Ferzli, I. Khalife, "Mobile cloud computing educational tool for image/video processing algorithms," in *Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE)*, pp. 529, March 2011.

[FLR12] N. Fernando, S. W. Loke, W. Rahayu, "Mobile cloud computing: A survey", *Future Generation Computer Systems*, Vol. 29, pp. 84–106, 2013.

[FR11] E. Freeman, E. Robson, "Head First HTML5 Programming" O'Reilly Media, ISBN 10:1-4493-9055-2, October 2011

[FRL08] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-degree compared," in *Grid Computing Environments Workshop*, pp. 1–10, 2008.

[Ganek07] A. Ganek, "Autonomic Computing: Concepts, Infrastructure, and Applications", *CRC Press*, pp. 3–18, 2007.

[GAP] Google App Engine, available online <https://developers.google.com/appengine/>

[GApps] Google Apps for Business, available online <http://www.google.com/enterprise/apps/business/>

[Geelan08] Jeremy Geelan. "Twenty one experts define cloud computing", *Electronic Magazine*, available online: <http://virtualization.sys-con.com/node/612375>, August 2008.

[GC01] J. Goodman, P. Chandrakasan, "An Energy Efficient Reconfigurable Public Key Cryptography Processor", *IEEE journal of solid state circuits*, pp. 1808-1820, November 2001.

[GCE] Google Compute Engine, available online <https://cloud.google.com/products/compute-engine>

[Gens09] F. Gens, "New IDC IT Cloud Services Survey Top Benefits and Challenges", December 2009.

[GoogleTrends] Google Trends, available online <http://www.google.com/trends/explore#cmpt=q>

[GZK+11] Y. Guo, L. Zhang, J. Kong, J. Sun, T. Feng, X. Chen, "Jupiter: Transparent Augmentation of Smartphone Capabilities through Cloud Computing", *MobiHels'11*, October 23, Cascais, Portugal, 2011.

Security of Mobile Cloud Applications

[Haleem12] T. Haleem, "Cloud Computing: Concepts and Trends", *Asian Journal of Engineering, Sciences and Technology*, Vol.2, No.1, March 2012.

[HT12] M. Hamdaqua, L. Tahvildari, "Cloud Computing Uncovered: A Research Landscape", *Advances in Computers*, Vol. 86, ISSN: 0065-2458, <http://dx.doi.org/10.1016/B978-0-12-396535-6.00002-8>, 2012.

[HB12] S. M. Hashemi, A. K. Bardsiri, "Cloud Computing Vs. Grid Computing", *ARPN Journal of Systems and Software*, ISSN 2222-9833, Vol. 2, No.5, May 2012.

[HCD10] G. Huerta-Canepa and D. Lee, "A Virtual Cloud Computing Provider for Mobile Devices", *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services Social Networks and Beyond (MCS '10)*, San Francisco, CA, USA: ACM, 2010

[HLL11] S.C. Hsueh, J.Y. Lin, M.Y. Lin, "Secure cloud storage for conventional data archive of smart phones", in: *Proc. 15th IEEE Int. Symposium on Consumer Electronics, ISCE '11*, Singapore, June 2011.

[Hirani08] S. Hirani, "Energy Consumption of Encryption Schemes in Wireless Devices", Hirani " Thesis, University of Pittsburgh, Apr. 9, 2003, Retrieved Oct. 1, 2008. (<http://portal.acm.org/citation.cfm?id=383768>)

[HMC08] M. Huebscher, J. McCann, "A survey of autonomic computing degrees, models, and applications", *ACM Computing Surveys*, Vol. 40, pp. 1-28, 2008.

[HO] HyperOffice, available online <http://www.hyperoffice.com/>

[HZK+10] D. Huan, X. Zhang, M. Kang, J. Luo, "MobiCloud: building secure cloud framework for mobile computing and communication", in: *Proc. 5th IEEE Int. Symposium on Service Oriented System Engineering, SOSE '10*, Nanjing, China, June 2010.

[HZX11] D. Huang, Z. Zhou, L. Xu, "Secure Data Processing Framework for Mobile Cloud Computing", *Workshop on Cloud Computing, INFOCOM*, June 2011.

[IDD] International Due Diligence: U.S. vs. European Privacy Laws Kroll an altegrity Company, available online: http://www.kroll.com/media/pdfs/International_Due_Diligence_US_vs_Euro_WP_040811P.pdf

[ITU01] International Telecommunication Union, X-509 | ISO/IEC 9594-8, "The directory: Public-key and attribute certificate frameworks", *ITU, X-Series*, 2001.

[IKC10]W. Itani, A. Kayssi, A. Chehab, "Energy-efficient incremental integrity for securing storage in mobile cloud computing", in: *Proc. Int. Conference on Energy Aware Computing, ICEAC '10*, Cairo, Egypt, Dec. 2010.

Security of Mobile Cloud Applications

[JCS] The Joyent Compute Service, *available online*
<http://www.joyent.com/products/compute-service>

[JERICH0] "Cloud Cube Model: Selecting Cloud Formations for Secure Collaboration", *Jericho Forum*, Position Paper, Version 1.0, April, 2009.

[JL03] X. Jin and J. Liu, "From individual based modeling to autonomy oriented computation," in *Agents and Computational Autonomy*, ser. *Lecture Notes in Computer Science*, M. N. et al., Ed., vol. 2969. Springer, pp. 151-169, 2003.

[KA11]. A. Khan and K.K. Ahirwar, "Mobile Cloud Computing as a future of mobile multimedia database" ,, *International Journal of Computer Science and Communication* Vol. 2, No. 1, January-June, pp. 219-221, 2011.

[Katzan09]H. Katzan Jr., "Computing Services in the Cloud", *S AIS 2009 Proceedings*, Paper 3,
<http://aisel.aisnet.org/sais2009/3> , 2009.

[KBR+11] M. Kamel, K. Boudaoud, S. Resondry, M. Riveill "Low-Energy Consuming and User-centric Security Management Architecture Adapted to Mobile Environments" In *Proceedings of the 12th IFIP/IEEE International Symposium on Integrated Network Management (IM'2011)*, Dublin, Ireland, May, 23 - 27, 2011.

[KCK11]. D. Kovachev, Yiwei Cao and Ralf Klamma. Mobile Cloud Computing: "A Comparison of application Models" . In *eprint arXiv: 1107.4940*, July 2011.

[KLK12]. S.K. Ko, J.H. Lee, S.W. Kim, "Mobile Cloud Computing Security Considerations", *Journal of Security Engineering*, 2012.

[KMK+12] A. N. Khan, M.L. MatKiah, S. U. Khan, S. A. Madani, "Towards secure mobile cloud computing: A survey", *Future Generation Computing Systems* 2012, doi:10.1016/j.future.2012.08.003

[KP]. A. Kumar, A. Purandare, J. Chen, A. Meacham, L. Subramanian, "ELMR: Lightweight Mobile Health Records"

[Leavitt09] N. Leavitt, "Is cloud computing really ready for prime time?", *Growth*, Vol. 27 pp.5, 2009.

[LH10]. H. Li, X.S. Hua, "Melog: mobile experience sharing through automatic multimedia blogging," in *Proceedings of the 2010 ACM multimedia workshop on Mobile cloud media computing (MCMC)*, pp. 19-24, 2010.

[LLL+12] P. Lindberg, J. Leingang, D. Lysaker, S.U. Khan, J. Li, "Comparison and analysis of eight scheduling heuristics for the optimization of energy consumption and

Security of Mobile Cloud Applications

make span in large-scale distributed systems”, *Journal of Supercomputing*, pp. 323–360, 2012.

[LMS11] Lookout Mobile Security, Lookout Mobile Threat Report, August 2011.

[MCA06] M.F. Mokbel, C. Chow, W.G. Aref, “The new casper: query processing for location services without compromising privacy”, in: *Proc. 32nd Int. Conference on Very Large Databases, VLDB '06*, Seoul, Korea, Sep. 2006.

[MCCF] Mobile cloud Computing Forum, *available online: <http://www.mobilecloudcomputingforum.com/>*

[MCM11]. “Mobile Communication for medical care – Final Report”, 21 April 2011

[MGL+11] V. March, Y. Gu, E. Leonardi, G. Goh, M. Kirchberg, B. S. Lee. “ μ Cloud: Towards a New Paradigm of Rich Mobile Applications”. In the *8th International Conference on Mobile Web Information Systems (MobiWIS)*, June 21, 2011.

[MR08] P. Morville, L. Rosenfeld, “Information Architecture for the World Wide Web”, 3rd Edition, *O'Reilly Media*, ISBN 10:0-596-15291-4, July 2008.

[MSA] Windows Azure, *available online <http://www.windowsazure.com/en-us/>*

[Nachenberg11] C. Nachenberg, “A Window Into Mobile Device Security – Examining the security approaches employed in Apple’s iOS and Google’s Android”, Symantec Security Response, *available online: http://investor.symantec.com/files/doc_news/2012/symc_mobile_device_security_june2011.pdf*, 2011

[Nadeem06] A. Nadeem, “A performance comparison of data encryption algorithms,” *IEEE Information and Communication Technologies*, pp. 84-89, 2006.

[Ness09] G. Ness, 2009, “3 Major barriers to cloud computing retrieved”, 22 May 2011, *available online <http://www.infra20.com/post.cfm/3-major-barriers-to-cloudcomputing>*.

[NIST] P. Mell, T. Grance, “The NIST Definition of Cloud Computing”, National Institute of Standards and Technology, U.S. Department of Commerce, *Special Publication 800-145*, September, 2011.

[NL05] E. Newcomer, G. Lomow, “Understanding SOA with web services.” *Addison-Wesley*, 2005.

[Nobelis08] N. Nobelis, “Une architecture pour le transfert électronique sécurisé de document”, Thèse, l’Université de Nice - Sophia Antipolis, 2008.

Security of Mobile Cloud Applications

[NWD05] P. Ning, R. Wang, W. Du, "An efficient scheme for authenticating public keys in sensor networks", *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, Chicago, IL, USA, pp. 58-67, 2005.

[OWASP10] Open web application security project (OWASP) Top 10, *available online: https://www.owasp.org/index.php/Top_10_2010-Main*, 2010.

[Oreilly08] T. Oreilly, "What is Web 2.0: Design patterns and business models for the next generation of software," O'Reilly Media, Tech. Rep., 2008. *available online: <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>*

[Papazoglou03] Mike P. Papazoglou, "Service-Oriented Computing: Concepts, Characteristics and Directions", in *Proceeding WISE '03 Proceedings of the Fourth International Conference on Web Information Systems Engineering* pp. 3, IEEE Computer Society Washington, DC, USA, 2003

[PBC+13a] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "A Security Framework for Mobile Cloud Applications "Networking in Education and Research", Sinaia, Romania, ARNIEC/RoEduNet Agency, IEEE Romanian Section, ISSN-L 2068-1038, pp. 13-16, 17-19 January, 2013.

[PBC+13b] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "Personalized Security Mechanism for Mobile Cloud Applications", in *Acta Tehnica Napocensis, Electronics and Communications*, No.2 Vol. 54, pp. 24-27, 2013.

[PBC+13c] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "A System to Analyze the User's Security Options for Mobile Cloud Applications", in *Proceedings of The 6th International Conference on Security for Information Technology and Communications, SECITC'13*, Bucharest, Romania, ISSN-L 2285-1798, pp. 289-298, June 25, 2013.

[PBC+13d] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "Mobile Cloud Applications and Traceability", in *Proceedings ROEduNet 12 th International Conference*, Constanta, 26-28 September, 2013.

[PBB13e] D. Popa, K. Boudaoud, M. Borda, "Secure Mobile-Cloud Framework - Implementation on the Mobile Device", in *Acta Tehnica Napocensis, Electronics and Communications*, 2013. (Submitted - accepted for publication).

[PBC+13f] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "Integration of Secure Mobile-Cloud Framework into a mobile cloud application scenario", in *Scientific Bulletin of the Politechnica University of Timisoara, Transactions on Electronics and Communications*, 2013 (Submitted - accepted for publication).

Security of Mobile Cloud Applications

- [PBC+13g] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "Overview on Mobile Cloud Computing Security Issues", in *Scientific Bulletin of the Politechnica University of Timisoara, Transactions on Electronics and Communications*, 2013 (Submitted - accepted for publication).
- [QG11] H. Qi, A. Gani, "Research on Mobile Cloud Computing: Review, Trend and Perspectives"
- [QMS+12] D.M. Quan, F. Mezza, D. Sannenli, R. Giafreda, "T -alloc: a practical energy efficient resource allocation algorithm for traditional data centers", *Future Generation Computer Systems*, pp. 791-800, 2012.
- [Raman09] T.V. Raman, "Toward 2W, Beyond Web 2.0", *Communications of the ACM*, pp. 52-59, 2009.
- [RCS] Rackspace Cloud Services, *available online* <http://www.rackspace.com/cloud/servers/>
- [RSB+09] M. Pastaki Rad, A. Sajedi Badashian, G. Meydanipour, M. Ashurzad Delcheh, M. Alipour, H. Afzali, "A survey of cloud platforms and their future", *Computational Science and Its Applications – ICCSA 2009*, pp. 788-796, 2009.
- [Rappa04] M. Rappa, "The utility business model and the future of computing services," *IBM Systems Journal*, Vol. 43, no. 1, pp. 32-42, 2004.
- [RBase] Rollbase, *available online*: <http://www.progress.com/products/rollbase>
- [Rose10] M. Rouse, "Cloud Computing", in *SearchCloudComputing*, *available online*: <http://searchcloudcomputing.techtarget.com/definition/cloud-computing>, 2010.
- [RP13] I. F. bin Aminuddin, bin. Kamaruzaman, "Application Android de e-sante", *Rapport de projet*, 2013.
- [SETI] Seti at Home Needs your Help, *available online*: <http://setiathome.berkeley.edu/>
- [SForce] Salesforce, *available online*: <http://www.salesforce.com/eu/>
- [SGG+11]. M. Somasundaram, S.Gitanjali, T.C.Govardhani, G. Lakshmi Priya, R. Sivakumar, "Medical Image Data Management System in Mobile Cloud Computing Environment", *IACSIT Press*, Vol. 21, Singapore, 2011.
- [SK10] L. Subashini, V. Kavitha, "A survey on security issues in service delivery models of cloud computing", *Journal of Network and Computer Applications*, 2010.
- [SL10] S. Srinivasamurthy, D. Liu, "Survey on cloud computing security", 2010.

Security of Mobile Cloud Applications

- [SL12] J. Szefer, R. B. Lee, "Architectural Support for Hypervisor-Secure Virtualization", in *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, March 2012.
- [SM13] S. Sharma, U. Mittal, "Comparative analysis of various authentications techniques in cloud computing", *International Journal of Innovative Research in Science, Engineering and Technology*, Vol. 2, Issue 4, April 2013.
- [SML+09] Z. Song, J. Molina, S. Lee, S. Kotani, and R. Masuoka. "TrustCube: An Infrastructure that Builds Trust in Client," in *Proceedings of the 1st International Conference on Future of Trust in Computing*, 2009.
- [Sood12] S.K.Sood, "A combined approach to ensure data security in cloud computing", in *S.K. Sood/Journal of Network and Computer Applications* Vol.35, pp. 1831-1838, 2012.
- [Sun09] Sun microsystems, "Introduction to Cloud Computing architecture", *White Paper 1st Edition*, June 2009.
- [STL10] L. J. Sotto, B. C. Treacy, and M. L. McLellan, "Privacy and Data Security Risks in Cloud Computing", *Electronic Commerce & Law Report*, 15 ECLR 186, pp. 1-4, 2010.
- [SW12] J. Sun, P. Wang, "Community Ecology for Innovation Concept: The Case of Cloud Computing", *Thirty Third International Conference on Information Systems*, Orlando, 2012.
- [Swaminathan08] K. S. Swaminathan, "Computing in the clouds", *The journal of high-performance business*, Accenture, May 2008.
- [TCSA10] "Top threats to cloud computing", version 1.0, Cloud Security Alliance CSA, available online: <http://www.cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf>, Retrieved March, 2010.
- [TCSA13] "The Notorious Nine: Cloud Computing Top Threats in 2013", Cloud Security Alliance CSA, Top Threats Working Group, February 2013.
- [VRM+09] L.M. Vaquero, L. Rodero-Merino, J. Caceres, M. Lindner, "A Break in the Clouds: Towards a Cloud Definition", *ACM SIGCOMM Computer Communication*, Vol.39, pp. 50-55, January 2009.
- [Wang10] Q. Wang, "Mobile Cloud Computing", *Master Thesis*, 2010.
- [WD11]. S. Wang, S. Dey, "Rendering Adaptation to Address Communication and Computation Constraints in Cloud Mobile Gaming," in *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1-6, January 2011.

Security of Mobile Cloud Applications

- [WG08] L. Wang, G. von Laszewski, "Scientific Cloud Computing: Early Definition and Experience", *10th IEEE International Conference on High Performance Computing and Communications*, pp. 825-830, doi:10.1109/hpcc.2008.38, 27 September, 2008.
- [YBS08] L. Youseff, M. Butrico, D. Da Silva, "Toward a Unified Ontology of Cloud Computing", *Grid Computing Environments Workshop*, pp. 1 - 10, 12-16 Nov, Austin, TX, 2008.
- [YCL10]. Z. Ye, X. Chen, Z. Li, "Video based mobile location search with large set of SIFT points in cloud," in *Proceedings of the 2010 ACM multimedia workshop on Mobile cloud media computing (MCMC)*, pp. 25-30, 2010.
- [Zakas05] N. C. Zakas, "Professional JavaScript for Web Developers", *Wrox*, 1 edition, 978-0764579080, April 22, 2005.
- [ZCB10] Q. Zhang, L. Cheng, R. Boutaba, "Cloud computing: state-of-the-art and research challenges", *Journal Internet Serv Appl*, Vol.1 pp.7-18, DOI 10.1007/s13174-010-0007-6, 2010.
- [ZL10] D. Zissis, D.s Lekkas, "Addressing cloud computing security issues", *Future Generation Computer Systems* 28, pp. 583-592, 2012.
- [ZM10] H. Zhangwei and X. Mingjun, "A Distributed Spatial Cloaking Protocol for Location Privacy," in *Proceedings of the 2nd International Conference on Networks Security Wireless Communications and Trusted Computing (NSWCTC)*, vol. 2, pp. 468, June 2010
- [ZSG+09] X. Zhang, J. Schiffman, S. Gibbs, A. Kunjithapatham, and S. Jeong. "Securing Elastic Applications on Mobile Devices". In *CCSW'09*, November 13, Chicago, Illinois, USA, 2009.

Security of Mobile Cloud Applications

List of Publications

- [1]. D. Popa, K. Boudaoud, M. Cremene, M. Borda, "A Security Framework for Mobile Cloud Applications "Networking in Education and Research", Sinaia, Romania, ARNIEC/RoEduNet Agency, IEEE Romanian Section, ISSN-L 2068-1038, pp. 13-16, 17-19 January, 2013.
- [2]. D. Popa, K. Boudaoud, M. Cremene, M. Borda, "Personalized Security Mechanism for Mobile Cloud Applications", in Acta Tehnica Napocensis, Electronics and Communications, No.2 Vol. 54, pp. 24-27, 2013.
- [3]. D. Popa, K. Boudaoud, M. Cremene, M. Borda, "A System to Analyze the User's Security Options for Mobile Cloud Applications", in Proceedings of The 6th International Conference on Security for Information Technology and Communications, SECITC'13, Bucharest, Romania, ISSN-L 2285-1798, pp. 289-298, June 25, 2013.
- [4]. D. Popa, K. Boudaoud, M. Cremene, M. Borda, "Mobile Cloud Applications and Traceability", in Proceedings ROEduNet 12 th International Conference, Constanta, 26-28 September, 2013.
- [5]. D. Popa, K. Boudaoud, M. Borda, "Secure Mobile-Cloud Framework - Implementation on the Mobile Device", in Acta Tehnica Napocensis, Electronics and Communications, 2013. (Submitted - accepted for publication).
- [6]. D. Popa, K. Boudaoud, M. Cremene, M. Borda, "Integration of Secure Mobile-Cloud Framework into a mobile cloud application scenario", in Scientific Bulletin of the Politechnica University of Timisoara, Transactions on Electronics and Communications, 2013 (Submitted - accepted for publication).
- [7]. D. Popa, K. Boudaoud, M. Cremene, M. Borda, "Overview on Mobile Cloud Computing Security Issues", in Scientific Bulletin of the Politechnica University of Timisoara, Transactions on Electronics and Communications, 2013 (Submitted - accepted for publication).

Security of Mobile Cloud Applications

Appendix

The most relevant papers for the thesis research activity:

- [1]. **D. Popa**, K. Boudaoud, M. Cremene, M. Borda, “**A Security Framework for Mobile Cloud Applications**”, Networking in Education and Research”, Sinaia, Romania, ARNIEC/RoEduNet Agency, IEEE Romanian Section, ISSN-L 2068-1038, pp. 13-16, 17-19 January, 2013.
- [2]. **D. Popa**, K. Boudaoud, M. Cremene, M. Borda, “**A System to Analyze the User’s Security Options for Mobile Cloud Applications**”, in Proceedings of The 6th International Conference on Security for Information Technology and Communications, SECITC’13, Bucharest, Romania, ISSN-L 2285-1798, pp. 289-298, June 25, 2013.
- [3]. **D. Popa**, K. Boudaoud, M. Borda, “**Secure Mobile-Cloud Framework - Implementation on the Mobile Device**”, in Acta Tehnica Napocensis, Electronics and Communications, 2013. (Submitted – accepted for publication).
- [4]. **D. Popa**, K. Boudaoud, M. Cremene, M. Borda, “**Integration of Secure Mobile-Cloud Framework into a mobile cloud application scenario**”, in Scientific Bulletin of the Politechnica University of Timisoara, Transactions on Electronics and Communications, 2013 (Submitted – accepted for publication).
- [5]. **D. Popa**, K. Boudaoud, M. Cremene, M. Borda, “**Overview on Mobile Cloud Computing Security Issues**”, in Scientific Bulletin of the Politechnica University of Timisoara, Transactions on Electronics and Communications, 2013 (Submitted – accepted for publication).

Security of Mobile Cloud Applications

A Security Framework for Mobile Cloud Applications

Daniela POPA, Marcel CREMENE, Monica BORDA

Communications Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
{Daniela.Popa, cremene, Monica.Borda}@com.utcluj.ro

Karima BOUDAUD

I3S-CNRS Laboratory
University of Nice Sophia Antipolis
Sophia Antipolis, France
karima@polytec.unice.fr

Abstract— Mobile Cloud Computing is a new concept, which offers Cloud resources and services for mobile devices. It also brings several advantages to mobile devices and to the applications developed for them. However, it increases the security risks and privacy invasion due to the fact that it combines mobile devices with Cloud services and because there is not a well-defined application model. The security issues are treated independently and the existing security solutions are supplied separately by various providers. In this paper, we propose a framework to secure the data transmitted between the components of the same mobile cloud application; and to ensure the integrity of the applications at the installation on the mobile device and when being updated. Our framework allows applying different security properties to different kinds of data and not the same properties to all the data processed by the application. Also our approach takes into consideration the user preferences and the mobile device performances.

Keywords—*Mobile Cloud Computing; Applications; Security;*

I. INTRODUCTION

The concept of Mobile Cloud Computing is relatively new in the research. It brings various advantages for mobile devices since it enables the use of Cloud resources and services.

The use of smart-phones has grown and continues to grow. A study made by Gartner [1] shows that in the third quarter of 2011 the sale of smart-phones increased with 42 percent. Furthermore, according to ABI Research [2], by 2015 more than 240 million business customers will use Cloud resources and services through mobile devices and this will conduct to revenues of billions of dollars. This increase is due to the fact that powerful applications were developed for mobile devices; these allow users to perform tasks like: managing personal health, games, editing, making reservations and paying tickets. To run this kind of applications, mobile hardware and network have known several improvements [3]. Even with those improvements mobile devices still provide challenges like: lack of resources and energy, security issues and unstable connectivity. A solution to the mobile device challenges is Mobile Cloud Computing, who offers Cloud Computing as a platform for powerful applications. However, this new solution

increases the security risks and privacy invasion due to data outsourcing and synchronization via Internet.

The security issues in Mobile Cloud Computing are due to the security threats against the Cloud, the mobile devices and applications running on these devices, which can be native or mobile web applications. These threats can be classified in four categories: mobile threats, cloud threats, mashup threats and technological threats. All this menaces have as a purpose to steal user private data or to exploit mobile device resources.

Our work focuses on securing private data used by a component-based mobile cloud application. There are very few studies in this area. The existing security solutions treat independently the different types of mobile cloud security problems. Solutions for security issues on mobile devices are proposed by the mobile platforms, also the services providers suggest solutions for issues in Cloud. The security issues concerning data transmission are solved by service providers using security protocols such as SSL/HTTPS. However, this kind of protocols are on one hand high energy consuming and on a second hand provide security properties as a block without taking into account the type of data transmitted or the user expectations.

In this paper we propose a framework to secure the data transmitted between the components of the same mobile cloud application; and to ensure the integrity of the applications at the installation on the mobile device and when being updated.

The paper is organized as follow. Section II provides a brief overview of Cloud Computing and Mobile Cloud Computing, including definition and application models. In Section III are presented several security issues and approaches to address the issues. In section IV, we detail our approach and architecture. Finally, we conclude this paper.

II. MOBILE CLOUD APPLICATIONS MODELS

Cloud Computing for mobile world or rather Mobile Cloud Computing is a new concept that can be described as the availability of Cloud resources and services for mobile devices.

Cloud Computing is a new internet-based paradigm that is focused on providing services to its customers, according to their needs. It offers the advantages of having on-demand computing services, paying according to the resources used and tolerance to resources alteration; moreover cloud services can

be employed by different types of client platforms (e.g., mobile phones, laptops, and PDAs). A definition that is generally accepted by several works [4], [5], [6] and that is employed by IT network and security professionals is the one published by NIST. This declares that Cloud Computing is defined by describing three Cloud service models: *Infrastructure as a Service* (e.g. servers, networks and storages), *Platform as a Service* (e.g. middleware services and operating systems) and *Software as a Service* (e.g. application programs); those are provided by cloud providers like Amazon, Google at certain prices. In addition to this service models, there were established four Cloud deployment models: *Public*, *Private*, *Community* and *Hybrid*.

The security framework that we propose in this paper fits in the SaaS layer. It provides security, using services in the Cloud (e.g. the components that provide confidentiality or integrity are deployed in Cloud as services).

Mobile Cloud Computing is a model [3] developed as a solution to overcome the mobile devices challenges by using Cloud Computing services like storage and computing resources. It also tacks into account the context of the mobile operating conditions. In order to benefit as much as possible from the advantages offered by the Cloud, there have been several studies on mobile cloud applications models.

Mobile cloud applications can be classified in three categories; a feature used in defining these three categories is the mobile device involvement in the execution of a mobile cloud application. The three categories are as follows:

a) *The Client model*: Here the mobile device is seen only as a more convenient way to access services in the Cloud.

b) *The Client/Cloud model*: It includes applications divided into components and distributed between mobile device and the Cloud. These models use techniques like: augmented execution [7], elasticity [8] and mobility [9] to overcome the mobile devices limitations by distributing the application execution.

c) *The Cloud model*: It considers the fact that the mobile device is an integral part of the Cloud [10]. The objective for the Cloud model approach is to provide a distributed infrastructure that exploits the storage and computing capacity of several mobile devices in order to support new applications.

In [2] is made a comparison between the novel applications models proposed and developed for mobile devices. The new applications models go on the idea of separating an application into components.

This separation can be made at implementation code level, like in elastic applications model proposed in [8], or at architectural level (see Fig. 1), like in rich applications model proposed in [9].

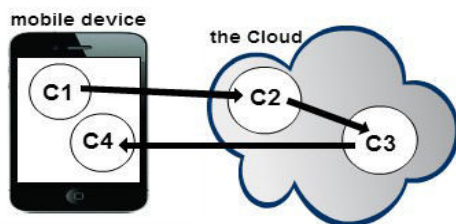


Fig. 1. Component-based application

III. SECURITY ISSUES AND EXISTING SOLUTIONS

Mobile cloud applications expose user private data to different security risks. User data can be stored on the mobile side or on the Cloud side, can be accessed by applications (or application components) that run on the mobile device or in Cloud, or can be transmitted between mobile device application components and Cloud application components.

A. Security Issues Related to Mobile Cloud Applications

As we have said previously, Mobile Cloud Computing is a combination of mobile and Cloud Computing. Thus, the security risks are caused by the security attacks on the mobile side, the security issues on the Cloud side and also by the security attacks against the communication channels.

The last studies [11][12] on mobile security issues have revealed the following categories of mobile attacks: application based attacks, web-based attacks, network based attacks and physical based attacks. These attacks affect the integrity and the confidentiality of mobile data and applications. As a result, data may be corrupted, modified or deleted and the application functionality can be altered. Repackaging was the most used technique in 2011 to infect applications running under Android [11]. An attacker takes a healthy application; changes it so that this contains malicious code and after republishes it. Technologies such as HTML5 and AJAX enabled the growth of the mobile computing market, but as a drawback these technologies introduce some security issues and provide opportunities to the malicious users to obtain user’s private data.

Mobile Cloud Computing provides the advantage of storing a large amount of data outside the mobile device, i.e. in the Cloud. However, the Cloud can be the target of various attacks [13][4] concerning about data privacy, data ownership and location, data access and integrity. Moreover, in mobile cloud models the various components of an application may communicate or communicate with other web services and the used communication channel can be the target of network attacks such as man in the middle when the attacker connects with the victims and takes controls over their communication.

B. Existing Solutions

To secure user data and applications, the mobile platform providers (e.g. Android, iOS) implemented several security solutions. These solutions were included into the operating system of the devices. Five types of security features have been implemented by the different platforms: traditional access control, application provenance, encryption, isolation and permission-based access control [11]. Each mobile platform implements a different strategy to secure data. Thereby, the service providers have to adapt the applications to the strategy already adopted. Applying a high level of protection implies a limitation of performance, and also a high-energy consumption of the mobile. The application of these strategies will allow securing data on the mobile device, however, when the data will be sent and stored in the Cloud, it will become out of the user control.

For the Cloud side, different solutions [14][15] have been proposed to secure the data access. For example, to secure elastic applications a solution has been developed in [8], but this solution is provided for a specific application model, i.e. elastic applications with code migration. Beside the elastic application model different novel application models have been proposed [9]. However, the solution presented in [8] cannot be applied to these new models, i.e. component-based applications.

The mobile cloud application providers have to secure the data exchanged between the mobiles and Cloud. A commonly used solution is the SSL protocol, but as it has been said previously and proved in [16] using SSL increases the energy consumption of mobile devices. Another solution proposed for securing the communication between mobile devices is LECCSAM [16]. LECCSAM is an architecture based on security components that aims to optimize the mobile device energy consumption. Even if LECCSAM brings several advantages in securing the communication between mobile devices, it is not adapted to the mobile cloud applications.

IV. SECURITY FRAMEWORK

We focused on component-based mobile cloud application models with different execution locations, and with no security solutions provided for data transmitted between components. In our work, we assume that there is no need to apply the same security level for all data transmitted between the components. Moreover, we want to allow the users to choose the security level they want to apply to their data and to adapt the security level applied according to the mobile device energy consumption.

Thereby the framework that we propose called Secure Mobile-Cloud (SMC) has to fulfill the following features: to ensure the integrity of an application at setup and to secure the communication between the same application components (i.e. between components running on the mobile side and those running in Cloud and between the components running only in Cloud). Our architecture has to be able to adapt the security services according to the user needs, device characteristics and user context.

SMC framework (see Fig. 2) has several components running in the Cloud and on the mobile: 1) five kinds of

managers where each manager has a well-defined functionality (see Table I) and 2) the security components deployed in both Cloud and mobile device. Each security component satisfies one security property (e.g. integrity or confidentiality).

TABLE I. DESCRIPTION OF THE MANAGERS

Manager	Description
Mobile Manager	It collects data and events that occurs on the mobile side and sends them to the appropriate manager to be analyzed.
Mobile Security Manager Cloud Security Manager	Both provide the composition of the security properties. The Mobile Security Manager ensures security composition on the mobile side and the Cloud Security Manager ensures the composition on the Cloud side.
Optimization Manager	It sends the information collected from sensors (e.g. network sensor, energy sensor) to the mobile manager.
Application Manager	It checks the application integrity at setup.
Policy Manager	It determinates which security components are required for a specific security level.

A. Application Integrity at Setup

Application integrity has to be verified at installation and update. In this way it can be avoid the use of malicious application and the loos of private data. For this integrity check the framework proposed has to accomplish the following verifications: 1) if the application exists, 2) the application signature and 3) the application access described into the “manifest” file. To verify if an application exists, its name is searched in an official application store (e.g. Amazon, Apple). Then the signature is verified. The signature is compared whit the application signature found in the application store. If, it is different, it means that the application is malicious; otherwise it means that it is reliable, on condition that applications store has not been compromised. The next step is to verify the “manifest” file. Traditionally, applications running on mobile devices require user’s authorization to access some information or to perform certain operations. Unfortunately, very often, user’s give the authorization without thinking. The framework provides: a function whose feature is to analyze the different access levels

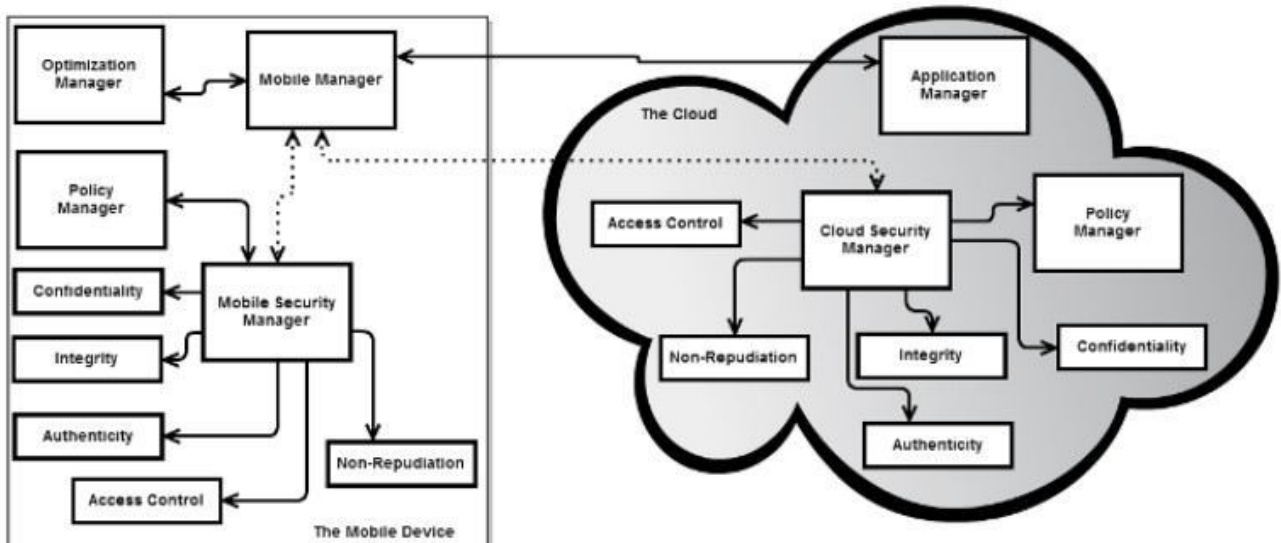


Fig. 2. Secure Mobile-Cloud framework

recorded in “manifest” file and to evaluate the risks for the access authorized by user; and, a function of comparison between manifest files. Events that signal an application setup are gathered by Mobile Manager. For the install operation this manager sends the application name, signature and ‘manifest’ file to the Application Manager where they are verified. For the update operation the new manifest file and the old manifest file are sent to be compared and verified. The results are sent back to Mobile Manager.

B. Secure the Communication

To secure the exchange of data between parts of an application running on the mobile device and in Cloud, we based our reasoning on LECCSAM because it offers a solution very flexible for the security management. Our solution consists essentially in extending the security components on the mobile device so that data security to be able to perform either in Cloud or on the mobile device.

Security of data transmitted is done by the following managers: Mobile Security Manager and Cloud Security Manager. They receive parameters like: the security level needed and the data to secure. Each manager uses its Policy Manager to determine the security properties for the security level to be applied. In the case of data transmitted between the mobile device and the Cloud, the security managers receive data from Mobile Manager. In Cloud, application components communicate directly with the Cloud Security Manager.

Mobile Manager keeps the information about user options regarding the data security level for the applications installed. It receives the information regarding user context, mobile battery level, and network availability from the Optimization Manager. It also intercepts the data that has to be transmitted to Cloud; for each data transmitted is defined a sensibility level. With all this information it establishes the security level applied to data and the mobile manager that will applied it.

Optimization Manager monitors the user context, battery level and network availability. These parameters may change frequently, so whenever there is a change the information is sent to the Mobile Manager.

The framework implementation is still in progress. However, the implementation of the security components on the mobile side is completed. The programming language used is Java and the mobile platform is Android.

V. CONCLUSIONS

Mobile Cloud Computing introduces many security issues due to the fact that it combines mobile devices with Cloud services and because there is not a well-defined application model. The security issues are treated independently and the existing security solutions are supplied separately by various providers. The framework that we propose aims to secure data communication between the same application components. The most important characteristics of our framework is that: 1) it allows applying different security properties to different kinds of data and not the same properties to all the data processed by the application, 2) the user preferences are taken into consideration and 3) the mobile device performances (e.g.

energy consumption) are also taken into account. The framework provides also a solution to verify the integrity of an application. At the moment we are working to an approach to secure data transmitted between the Cloud components.

ACKNOWLEDGMENT

This paper was supported by the project: Improvement of the doctoral studies quality in engineering science for development of the knowledge based society-QDOC" contract no. POSDRU/107/1.5/S/78534, project co-funded by the European Social Fund through the Sectorial Operational Prog. HR 2007-2013.

REFERENCES

- [1] Gartner, Inc., “Gartner Says Sales of Mobile Devices Grew 5.6 Percent in Third Quarter of 2011; Smartphone Sales Increased 42 Percent,” November, 2011, <http://www.gartner.com/it/page.jsp?id=1848514>
- [2] ABI Research., 2010, <http://www.abiresearch.com/>.
- [3] D. Kovachev, Y. Cao and R. Klamma, “Mobile Cloud Computing: A Comparison of application Models”, in eprint arXiv: 1107.4940, July 2011.
- [4] Cloud Security Alliance, “Security Guidance for Critical Areas of Focus in Cloud Computing V2”, December 2009.
- [5] M. Armbrust, et al., „Above the Clouds: A Berkeley View of Cloud Computing”, February, 2009.
- [6] ENISA, “Cloud Computing Benefits, risks and recommendations for information security”, November, 2009,
- [7] B.G. Chun and P. Maniatis, “Augmented Smartphone Applications Through Clone Cloud Execution,” in Proceedings of the 12th Workshop on Hot Topics in Operating Systems (HotOS XII), USENIX, 2009.
- [8] X. Zhang, J. Schiffman, S. Gibbs, A. Kunjithapatham, and S. Jeong, “Securing Elastic Applications on Mobile Devices”, In CCSW’09, November, 2009, Chicago, Illinois, USA.
- [9] V. March, Y. Gu, E. Leonardi, G. Goh, M. Kirchberg, B. S. Lee, “μCloud: Towards a New Paradigm of Rich Mobile Applications”, in the 8th International Conference on Mobile Web Information Systems (MobiWIS), June, 2011.
- [10] G. Huerta-Canepa and D. Lee, “A Virtual Cloud Computing Provider for Mobile Devices”, in Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services Social Networks and Beyond (MCS ’10) San Francisco, CA, USA: ACM, 2010.
- [11] Lookout Mobile Security, “Lookout Mobile Threat Report”, August 2011.
- [12] C. Nachenberg, “A Window Into Mobile Device Security – Examining the security approaches employed in Apple’s iOS and Google’s Android”, Symantec Security Response.
- [13] Cloud Security Alliance, “Top Threats to Cloud Computing V 1.0”, March 2010.
- [14] Z. Song, J. Molina, S. Lee, S. Kotani, and R. Masuoka, “TrustCube: An Infrastructure that Builds Trust in Client”, in Proceedings of the 1st International Conference on Future of Trust in Computing, 2009.
- [15] H. Zhangwei and X. Mingjun, “A Distributed Spatial Cloaking Protocol for Location Privacy,” in Proceedings of the 2nd International Conference on Networks Security Wireless Communications and Trusted Computing (NSWCTC), vol. 2, pp. 468, June, 2010.
- [16] M. Kamel, K. Boudaoud, S. Resondry and M. Riveill, “Low-Energy Consuming and User-centric Security Management Architecture Adapted to Mobile Environments”, in Proceedings of the 12th IFIP/IEEE International Symposium on Integrated Network Management (IM’2011), Dublin, Ireland, May, 2011.

A System to Analyze the User's Security Options for Mobile Cloud Applications

D. POPA¹ K. BOUDAUD² M. CREMENE¹ M. BORDA¹

¹Communications Department, ²I3S-CNRS Laboratory

¹Technical University of Cluj-Napoca, ²University of Nice Sophia Antipolis

¹Str. Dorobantilor. 71-73 CP. 400609, ²930 Route des Colles - BP 145- 06903

¹ROMANIA, ²FRANCE

¹{Daniela.Popa, cremene, Monica.Borda}@com.utcluj.ro ,

²karima.boudaoud@unice.fr

Abstract: Mobile Cloud Computing connects the mobile devices to services into the Cloud. It has brought several advantages and thereby it has received an increasing attention from mobile devices users and entrepreneurs. It has also raises a wide range of issues. In our work we emphasize on the security issues. We focus on securing data used and processed by component-based mobile cloud applications. We want to adapt the security according to end-user needs and mobile devices constraints.

Key-Words: Mobile Cloud Computing, Applications Types, Security Issues

1 Introduction

Mobile devices have turned into mini-computers that people carry constantly with them and from which they expect to be connected to the Internet 24 hours a day. But, mobile devices are characterized by the following flaws: lack of resources and energy, security issues and unstable connectivity.

Mobile Cloud Computing is a solution for mobile devices challenges that wants to meet the mobile devices user's expectations. This solution offers Cloud Computing as a platform for powerful applications. A powerful application is an application that in order to operate needs to use the device local resources and also external resources (e.g. computing power, storage place). To model these types of applications new techniques like augmented execution, elasticity and mobility [1] have been developed. The idea behind the new application models is to separate an application into components. A component may run in the Cloud, on the mobile device

or it can migrate.

Because of its various advantages Mobile Cloud Computing has received increasing attention from mobile devices users and entrepreneurs [2] even since its emergence. Marketing research [3] stated that in 2015 it will be more than 240 million customers using Mobile Cloud Computing services while in 2008 there were only 42.8 million customers.

The advantages offered by Mobile Cloud Computing are offset by the wide range of issues it raises. In [4] N. Fernando et al, submitted the key issues in Mobile Cloud Computing in the way they are approached in academia. The following categories of issues are discussed: 1) operational issues, 2) end-user issues, 3) service and application issues, 4) privacy, security and trust issues, 5) context-awareness issues and 6) data management issues. The security and privacy issues are the issues most of the users and services provides give attention to [3].

The definition of Mobile Cloud Computing means to connect mobile devices to

services in the Cloud. Thereby the security issues are various and fall into one of these three categories: mobile threats [5], Cloud threats [6] and threats at the communication channels level.

In our work we are focusing on the security of user's private data transmitted between the components of the same mobile cloud application. Also, we focus on the adaptation of security according to end-user needs and mobile devices constraints.

In this paper we discuss some criteria that may affect the security properties applied to component-based mobile cloud application data. Then, we present an overview of the security solution that we propose to secure mobile cloud applications, i.e. to secure data transmitted between the components (running on a mobile or in a Cloud) of a same application. Afterwards, it is showed the way we have integrate the user's requirements into the security framework.

This paper is organized as it follows. Section 2 describes the criteria that influence the security properties applied to data. In Section 3, we detail our approach. Section 4, covers a briefly description of several solutions which target the security challenges posted by Mobile Cloud Computing. Finally, we conclude the paper and highlight some future work.

2 Criteria

The lack of resources is one of the biggest disadvantages of mobile phones. Therefore, a security solution must consider mobile device constraints. In addition, as said previously, mobile owners (mobile end-users), running mobile applications have different expectations regarding the security of their private data. In this section we discuss several criteria that must be taken into account when designing a security solution for mobile cloud applications and more generally for today mobile applications.

2.1 Human Constraints

For each individual, personal data are important. The degree of importance may vary from a person to another person. This degree can differ according to each person perceptions regarding privacy and also according to each person status (citizen, politician, actor, government employee etc.). For example, a politician may need a higher security level for her/his data than a non-politician person.

User's requirements are not taken into consideration by various traditional security solutions. Actually, most of the time, these solutions, do not fit with the users expectations. Even if this was more or less acceptable until now, today it is an important issue that cannot be ignored as end-users are more and more concerned about security of their private data. Thus, a security solution must allow an end-user to express her/his needs regarding the security level of her/his data and more generally of the applications she/he uses and run on her/his mobile and also regarding the device energy consumption (i.e. an end-user may requires for a security solution that does not consume all the battery of her/his mobile when using a specific mobile cloud application). Furthermore a security solution has to be adapted to the user profile (non-security expert or security expert).

2.2 Technical Constraints

In our work, we consider different kinds of technical constraints: components location, user context and mobile device capabilities. In a component-based mobile cloud application, each component runs in a specific location. It can be on the mobile device or in Cloud. Moreover, some components can migrate between the mobile device and Cloud. When a component changes a location, the security level may also change. It should be strengthened, if the components migrate from the mobile device to the Cloud, or it

may be weakened if the migration is from the Cloud to the mobile device. Actually, these changes depend on the components that communicate between each other and on their location.

Concerning the user context, we refer to the area where the end-user is when executing an application: private or public area (e.g. home, office, public space such as airport, commercial center, etc.). The user context may change very often, particularly in the future, which will influence considerably the security level applied to the data transmitted between the components of a mobile cloud application.

Regarding mobile devices constraints, for devices like mobile phones, with limited resources and energy, it is important to provide security solutions that consume fewer resources without compromising and reducing the security level of data to secure. The energy consumed by the cryptographic algorithms (encryption, decryption, hash functions, etc.) used to traditionally secure data (in transit and data at rest) depends on the algorithm type. Asymmetric ciphering algorithms consume more resources than symmetric ones, which in their turns are more consuming than hash functions.

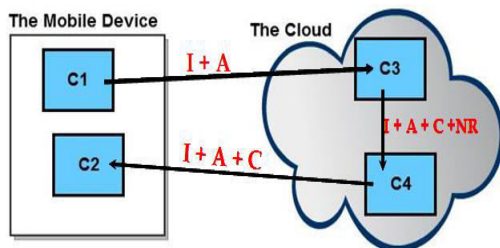


Figure 1 Different security levels

2.3 Data Sensitivity Constraints

Data may have different levels of sensitivity depending on user needs or user context. Moreover, in our opinion data are even more sensitive if their loss or theft causes important damages regarding the income and integrity of a user. For example the password data of a user or her/his bank

account is more sensitive than her/his preferred color.

Each sensitivity level requires an adequate security level, where each security level implies providing the right security properties (integrity, confidentiality, authenticity, etc.). Thus in a mobile cloud application, data transmitted between components may have different security levels and require different security properties (as shown in Figure 1).

3 The Proposed Security Approach

Through Secure Mobile-Cloud framework we want to propose a solution for securing data transition between the components of a mobile cloud application.

We started from an unsecured application and we assumed that there is no need to apply the same security levels to all data. We rely on the fact that data can have different levels of sensitivity, as we showed in Section 2. Also, we want our solution to give the user more flexibility in expressing his criteria for the security levels he needs/wants for his private data. Furthermore, we want our solution to take into account mobile device's technical constraint, particularly the energy consumption.

3.1 The Bases

Our security framework is based on the principle of the security properties separation. Each security property is designed and implemented under the form of independent components. Such a solution has been chosen to take in consideration: 1) the user's involvement in choosing the security levels for his data and 2) mobile device characteristics regarding saving energy.

In our framework the foundation for the security components design is LECCSAM [7]. LECCSAM is an architecture based on security components. The components are an assembly of cryptographic tools

satisfying a security property (e.g. integrity, confidentiality, authenticity, non-repudiation, access control). LECCSAM objectives are: 1) to secure the communication between mobile devices by applying the required security properties and 2) to optimize the mobile device energy consumption by moving the security components execution out from the mobile device.

3.2 The Security Framework

Secure Mobile-Cloud framework consists of two types of components: security components and management components. The security components are designed to implement the security properties. Each security component uses combinations of symmetric or asymmetric algorithms and some processing (e.g. concatenation). The management components are designed to discover and to apply the appropriate security components to user's private data. A briefly description of the management components is given in Table I. A detailed description of the architecture has been made in [8]

Secure Mobile-Cloud framework has some of the management components deployed on the mobile device side and some of them in Cloud. For the security components there are deployed versions for both Cloud and mobile device.

3.3 The User's requirements analyze

As we previous said we want to allow the users to express their choices regarding the security level they want to apply to their data.

In order to provide a solution that allows achieving this characteristic we have to design an analysis system. This system is integrated into the security framework. And it has to provide to the framework the security combination needed to be applied to data (security combination = security properties + security algorithms); and also the location where this combination can be performed (e.g. on the mobile or in Cloud).

Manager	Description
Mobile Manager	It collects data and events that occurs on the mobile side and sends them to the appropriate manager to be analyzed.
Mobile Security Manager Cloud Security Manager	Both provide the composition of the security properties. The Mobile Security Manager ensures security composition on the mobile side and the Cloud Security Manager ensures the composition on the Cloud side.
Optimization Manager	It sends the information collected from sensors (e.g. network sensor, energy sensor) to the mobile manager.
Application Manager	It checks the application integrity at setup.
Policy Manager	It determines which security components are required for a specific security level.

Table 1. The Managers Description

The analysis system is based on the following three statements:

Statement 1: Application data can have various sensitivity levels. The sensitivity levels are defined according to the harm produced by the data loss or the data theft. We have chosen three levels of sensibility

for data:

- 1) *Low sensibility level*: for all the data whose loss/theft brings no harm to the user.
- 2) *Medium sensibility level*: for all the data whose loss/theft undermines the user image (e.g. user's private data whose loss/theft do not harm their bank account or their security).
- 3) *High sensibility level*: for all the data whose loss/theft causes damages to the user (e.g. passwords, banking information, and identity information).

Statement 2: The user can affect the security solution applied to his data. He can affect the solution according to his level of knowledge in the security field.

We have defined three types of users:

- 1) *The Standard User Type*. He is characterized by his low knowledge regarding the security. To this type of user should be explained some basic notions regarding data and applications security.
- 2) *The Intermediary User Type*. He has some knowledge in the security area. For example he has basic concepts regarding the various levels of data sensitivity.
- 3) *The Expert User Type*. He has high knowledge regarding data security and sensitivity. For example he knows security details like security algorithms, what terms like integrity, authenticity, confidentiality and non-repudiation means. He also knows what level of security would be suitable for a certain level of sensitivity.

Statement 3: There are several constraints that may impact data security. The constraints considered by us and that can affect our security solution are specified by two entities: application architect and application user. The constraints are handled as predefined variables.

In the following it is presented the list of constraints that each entity needs to define. Also it is showed how each constraint affects the security solution.

Constraints specified by the application architect:

1) *application_type*. The application type can have a large area of values; for this reason we chose to group the applications into categories (e.g. banking, e-health, and gaming). The *applicatipn_type* values affect the value of the *default_combination* parameter. This *defoult_combination* parameter shows the lowest combination of security properties and security algorithms that may be accepted to secure a certain level of data sensibility.

2) *sensibility_level*. The data sensibility level, as has been showed previously. Its values are: {low, medium, high}.

3) *the_need_for_non_repudiation*. Non-repudiation is the intent to provide the sender of the data with a proof of receipt from the addressee. Its values are: {yes, no}.

In the case of the application users it comes to following question: *How much flexibility an user shall have?* Here, through flexibility we understand the number of constrains the user can define. We decided for the flexibility to vary according to the user type. The Standard User Type is provided with the lowest flexibility and the Expert User Type has the greater flexibility.

Constrains specified by the application users:

1) *security_choice*. Data from an application can be secured all at the same level, or can be secured differently depending on the sensitivity level. We want to allow the application's user to make this decision; this constraint allows that. Its values are: {all, each}.

2) *security_level*. The security level chosen by the user for an application data. Its values are: {strong, average}

3) *save_battery*. The user has to choose if he want to save the device energy while a

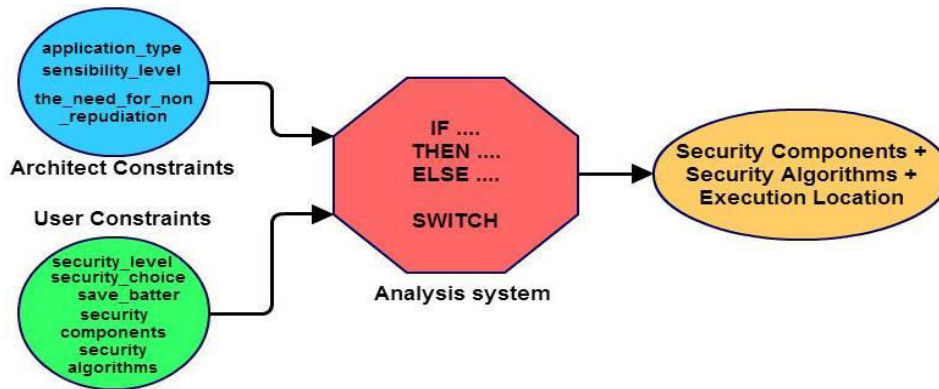


Figure 2. The System Requirement Analysis

certain application is running. Its values are: {yes, no}.

This constraint will affect the location where data will be secured; more precisely it will influence which of the security components will be used, those from the mobile device or those from Cloud.

4) *components_combination*. This constraint shows the combination chosen by the user for the security properties. To the default combination of properties (indicated by the default_combination parameter) the user may add one or more security properties but it can never removes any security properties. The values for the components combination are showed in Table 2.

5) *algorithms*. This constraint shows the symmetric, the asymmetric and the hash algorithms an user wants to use in securing

his data. In our implementation we used only the algorithms presented into Table 3.

The first three constraints can be defined by all the users without taking into the account the user type. The fourth constraint can be defined by the intermediary and the expert users. The fifth parameter can be defined only by the expert users.

The system for analyzing the constraints already presented (see Figure 2) was designed to rely on **IF...THEN...ELSE** rules and **SWITCH** rules. It has as **input parameters** the values of the architect constraints and the values of the user constraints. As **output parameters** the system must bring the combination for the security components and the algorithms they use.

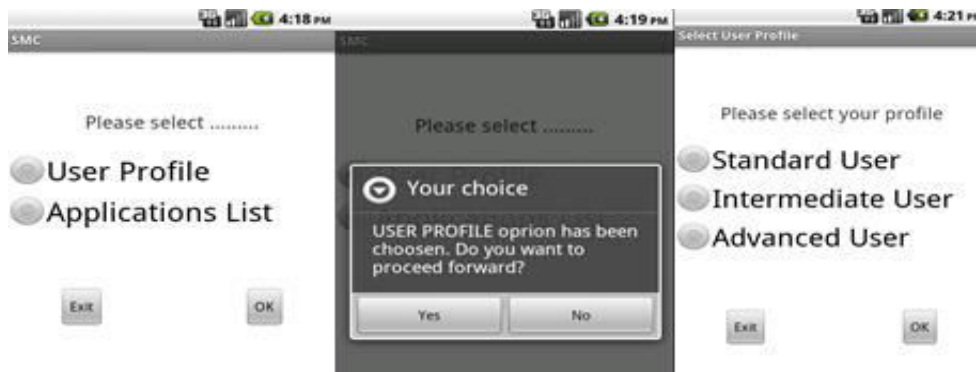


Figure 3. User Interface

In addition as output it is also needed the location for the security component execution.

At the beginning the rules were written in natural language and then they were transformed into rules that use IF...THEN...ELSE and SWITCH. In the following we are presenting some rules examples.

1) Rules for SWITCH:

"In the case we have a <standard user> the it is considered the following..."

2) Rules for IF...THEN:

If the running application type is a banking application and the security level chosen by user is average then the <security_combination> is 'I+C'.

If the running application type is a banking application and the security level chosen by user is average then the <security_combination> is 'A+C'.

2) Rules for IF...THEN...ELSE:

If the user wants to save battery while a certain application is running then the <execution_location> is the Cloud, otherwise the <execution_location> is the mobile device.

Combination ID	Combination Value
C_1	I (Integrity)
C_2	A (Authenticity)
C_3	C (Confidentiality)
C_4	I + NR (non-repudiation)
C_5	A + NR
C_6	C + NR
C_7	I + C
C_8	A + C
C_9	I + C + NR
C_10	A + C + NR

Table 2. The Components Combinations

Algorithm Type	Algorithm Name
Symmetric	DES, AES
Asymmetric	RSA
Hash	MD5, SHA-1

Table 3. Security Algorithms

The functionality of this analysis system is integrated into the Mobile Manager.

Regarding the implementation:

We designed a user friendly interface (see Figure 3). This interface is designed to gathered information from the user. The user will be able to set its type; then for each mobile cloud application he is able to set the security considerations (security_@{choice, level}, save_battery, components_combination, algorithms) he wants or it is allowed to.

The default_combination parameter it is already defined in to the SM-C framework.

After collecting the user's data those are saved into a SM-C database to which only the Mobile Manager has access. The database is implemented on the mobile device using SQLite.

Also, we have implemented in Java all the security components and the Security Manager on the mobile (Android) and the Cloud side. We are still working on the implementation of the other SM-C framework components.

4 The Existing Security Solutions

The security issues related to Mobile Cloud Computing are highlighted in various works. The existing security solutions treat independently the different types of mobile cloud security problems. They can be classified in solutions for: security issues on mobile devices [5], security issues in Cloud and security issues concerning data transmission [7]. Furthermore, some works [3] classify the existing security solutions in two categories: data security frameworks and application security frameworks. In [9] and [10] are presented two data security frameworks that ensure the confidentiality and integrity of the user's data stored in. In [11] it is presented a solution that ensures the security, integrity and authentication of mobile user data. Data are encrypted using traditional asymmetric algorithms. The

encrypted data are stored on cloud servers along with user's credentials (e.g. username, signature and password). The encryption and the decryption is made entirely on the mobile device; there are ignored the device limitation. In [12] is proposed a solution to secure an elastic mobile application. This solution is provided for a specific application model, and it covers the secure installation of elastic application, authentication, secure migration, and authorization of wablets.

5 Conclusions

In this paper, we don't want to criticize the existing security solutions for mobile cloud applications. Our objective is to propose an alternative solution to apply different security properties according to data sensitivity, human requirements and technical constraints.

As future work we plan to integrate the key management into the framework. We also want to evaluate the performances of our solution regarding adaptation of the security composition.

Acknowledgment:

This paper was supported by the project: Improvement of the doctoral studies quality in engineering science for development of the knowledge based society-QDOC" contract no. POSDRU/107/1.5/S/78534, project co-funded by the European Social Fund through the Sectorial Operational Prog. HR 2007-2013.

References:

[1] D. Kovachev, Y. Cao and R. Klamma, Mobile Cloud Computing: A Comparison of application Models, *in eprint arXiv: 1107.4940*, July 2011.
 [2] H. Liang, D. Huang and D. Peng, On Economic Mobile Cloud Computing Model, *in Mobile Computing, Applications, and Services Lecture Notes of the Institute for Computer Sciences, Social Informatics and*

Telecommunications Engineering V. 76, 2012, pp. 329-341.

[3] A. N. Khana, M.L. Mat Kiah, S.U. Khanb and S. A. Madanic, Towards secure mobile cloud computing: A survey, *doi: 10.1016/j.future.2012.08.003*

[4] N. Fernando, S.W. Loke and W. Rahayu, Mobile cloud computing: A survey, *Future in Generation Computer Systems* V. 29, 2013 pp. 84–106

[5] Lookout Mobile Security, Lookout Mobile Threat Report, Aug. 2011.

[6] Cloud Security Alliance, Top Threats to Cloud Computing V 1.0, March 2010.

[7] M. Kamel, K. Boudaoud, S. Resondry and M. Riveill, Low-Energy Consuming and User-centric Security Management Architecture Adapted to Mobile Environments, *in Proceedings of the 12th IFIP/IEEE International Symposium on Integrated Network Management (IM'2011)*, Dublin, Ireland, May, 2011.

[8] D. Popa, K. Boudaoud, M. Cremene, M. Borda, A Security Framework for Mobile Cloud Applications, *in Proceedings ROEduNet 11 th International Conference*, Sinaia, January 17-19, 2013.

[9] W. Ren, L. Yu, R. Gao, F. Xiong, Lightweight and compromise resilient storage outsourcing with distributed secure accessibility in mobile cloud computing, *in Journal of Tsinghua Science and Technology* V. 16 pp. 520–528, 2011.

[10] J. Yang, H. Wang, J. Wang, C. Tan and D. Yu1, Provable data possession of resource constrained mobile devices in cloud computing, *in Journal of Networks* V. 6 pp. 1033–1040, 2011

[11] S.C. Hsueh, J.Y. Lin, M.Y. Lin, Secure cloud storage for conventional data archive of smart phones, *in Proc. 15th IEEE Int. Symposium on Consumer Electronics, ISCE '11*, Singapore, June 2011.

[12] X. Zhang, J. Schiffman, S. Gibbs, A. Kunjithapatham, and S. Jeong, Securing Elastic Applications on Mobile Devices, *In CCSW'09*, November, 2009, Chicago, Illinois, USA.

SECURE MOBILE-CLOUD FRAMEWORK – IMPLEMENTATION ON THE MOBILE DEVICE

D. POPA¹ K. BOUDAUD² M. BORDA¹

¹Communications Department, Technical University of Cluj-Napoca, Romania
Str. Dorobantilor, 71-73, Tel/Fax: +40(0)264401575, {Daniela.Popa, Monica.Borda}@com.utcluj.ro

²IS-CNRS Laboratory, University of Nice Sophia Antipolis, France
930 Route des Colles - BP 145- 06903, Tel(Fax): +33(0)492965172(55), karima.boudaoud@unice.fr

Abstract: Secure Mobile-Cloud is a framework proposed to secure the data transmitted between the components of a mobile cloud application. In addition, the framework, takes into account the following aspects: 1) the users options regarding the security level required for private data and 2) the device energy consumption. The framework includes several distributed components. Some of these components are deployed on the mobile device and some of them in Cloud. This paper is focused on the implementation of the Secure Mobile-Cloud framework components on the mobile device. A proof of concept Android prototype is proposed.

Keywords: Mobile Cloud Computing, Applications, Security

I. INTRODUCTION

One of the greatest opportunities that every person wants to enjoy is mobility. Furthermore, each person has a small amount of curiosity, supplemented by a strong need for communication and knowledge. The mobile devices seem to be the devices that are able to link the mobility property with human emotional needs and information technology. All this is done using the Internet.

In order to capture people's attention towards mobile devices, powerful applications were developed for these devices. The applications allow mobile users to perform tasks like: managing personal health, games, editing, making reservations and paying tickets. As it is generally known, mobile devices are characterized by lack of resources. Thus, in order to run this new kind of applications, mobile hardware and network have known several improvements; but it wasn't enough. A solution to the mobile device challenges is Mobile Cloud Computing.

Mobile Cloud Computing (MCC) [1] is a new concept that can be described as the availability of Cloud Computing resources and services on the mobile device. This fact brings several advantages for the mobile devices (saving device energy, new storage place, additional computing power, etc.) and enables new powerful applications developed for them (e.g. a wide ranges of features) [2].

However, Mobile Cloud Computing increases the security risks and privacy invasion due to data outsourcing and synchronization via Internet. The security issues are various and fall into one of these three categories: mobile threats [3], Cloud threats [4] and threats at the communication channels level. Personal data (e.g. credit card numbers, passwords, contact database, calendar, location) is one of the main target of the hackers.

We are particularly interested in the security of data transmission, more specifically, the security of private data transmitted between the components of the same mobile cloud application. In our work, we focus on the security protocol adaptation according to end-user needs and mobile devices constraints. Furthermore, we assume that there is no need to apply the same security level (i.e. same security

properties) for all data transmitted between the mobile cloud application's components.

We proposed a framework in [5] called Secure Mobile-Cloud (SMC). This framework has to secure the communication between the same mobile cloud application components. Also, it has to be able to adapt the security services according to the user needs and device (particularly the energy constraints). The framework includes two kinds of components: components deployed on the mobile device and components deployed in the Cloud.

In this paper we discuss in detail the implementation of the Secure Mobile-Cloud framework components on the mobile device side. In addition, we describe the design and implementation of the databases used for storing the user options. The user interface it is also presented.

This paper is organized as follows: section II describes the security framework. This section is divided in three parts: the first part is a short overview of the Secure Mobile-Cloud Framework described in more details in [5] and [6]; the second part presents the framework implementation on the mobile device; and in the last part are shown some unit tests. Section III presents the conclusions.

II. SECURITY FRAMEWORK

This section presents the Secure Mobile-Cloud framework design, short overview, and implementation.

A. Framework design – short overview

The Secure Mobile Cloud (SMC) framework is composed of two types of components, as presented in [6]: 1) security components and 2) management components. The security components have been designed in [7] for the LECCSAM architecture. The security components implement the eponym security properties: integrity, authenticity, confidentiality and non-repudiation. These security components are deployed in both, mobile device and in the Cloud. The management components have been designed to identify and apply the appropriate security properties to user's data. Some of these management

components are deployed on the mobile device and some of them are deployed in the Cloud.

The users are able to express their choices regarding the security level they want to apply to their data. In order to provide a solution that allows achieving this characteristic an analysis system was designed. This system is integrated into the security framework. Its function is to provide to the framework the security combination needed to be applied to data (security combination = security properties + security algorithms); and also the location where this combination can be performed (e.g. on the mobile or in Cloud).

The components of the SMC framework that are designed for the mobile device are presented in Figure 1. A short description for each component is given in Table I.

TABLE I. COMPONENTS DESCRIPTION

Component Name	Description
Mobile Security Manager	Manager, whose role is to ensure the composition of the security components on the mobile side.
Integrity	Security component, which applies the integrity property to data.
Authenticity	Security component, which applies the authenticity property to data.
Confidentiality	Security component, which applies the authenticity property to data.
Non-Repudiation	Security component, which applies the non-repudiation property to data.
Policy Manager	Manager, whose role is to determine which security components are required for a specific security level.
State Manager	Manager, whose role is to send the information regarding mobile device energy state to the Mobile Manager.
Mobile Manager	Manager, whose role is to collect data and events on the mobile device; it also includes the functionality of the analysis system.

B. Framework implementation

This section is divided in five subsections: 1) The Security Part, 2) The Auxiliary Part, 3) The Analysis System, 4) The Databases and 5) The User Interface.

The Security Part

This section presents the implementation of the Mobile Security Manager and the Security Components.

The class diagram is presented in Figure 2. The diagram depicts the connections between the various classes. The diagram consists of seven classes:

- *MobileSecurityManager* class: implements the functionality of the Mobile Security. It includes several methods, between which the most significant are the following two methods: *apply_combination_toEncrypt* and *apply_combination_toDecrypt*.

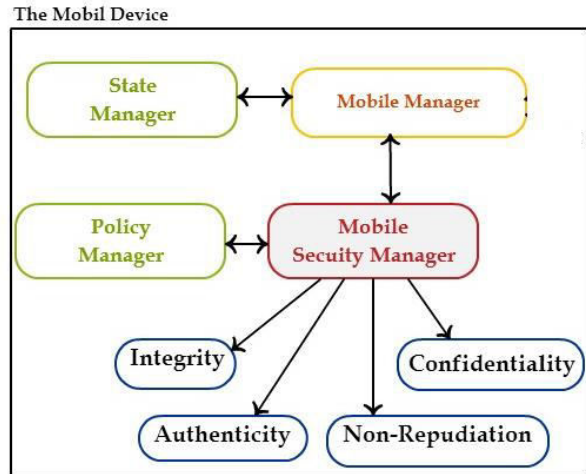


Figure 1. SMC framework – mobile device side

The *apply_combination_toEncrypt* method applies the appropriate security level to data in order to encrypt them (as the *apply_combination_toDecrypt* method is used when needed to decrypt the received data). The method has the following steps:

- 1) It finds that security properties combinations and corresponding algorithms for data security level provided. This is done by calling a method implemented by the Policy Manager; this method result returns a string with information.
- 2) It reads the string returned at the previous step; it sets the internal parameters (e.g Integrity, Authenticity) with the information read from the string.
- 3) It applies the corresponding security properties, by calling the appropriated methods.

- *Integrity, Authenticity, Confidentiality, NonRepudiation* classes: implement the security components functionality. Each of them comprises four methods, two private and two public methods. In the following there are described only the Integrity methods:

- *applyIntegrity* and *aIntegrity*: This two methods, the first one public and the second one private, are designed to provide the functionality of the Integrity component for plain text data. The public method calls the private method which uses a hash method implemented into the Operations class to perform the operation.

- *verifyIntegrity* and *vIntegrity*: This two methods, the first one public and the second one private, are designed to provide the functionality of the Integrity component for ciphered data.

- *Operations* class: implements methods that perform symmetric and asymmetric encryption and decryption and also the hash operation. These methods are as it follows:

- *encrypt*: which is symmetric or asymmetric; it takes as input an array of byte and a cipher; it performs a symmetric or an asymmetric encryption and returning an array of bytes.

- *decrypt*: which is symmetric or asymmetric; it takes as input an array of byte and a cipher; it performs a symmetric or an asymmetric decryption and returning an array of bytes.

- *hash*: it takes as input an array of byte and an hash algorithm performing an hash and returning an array of bytes.

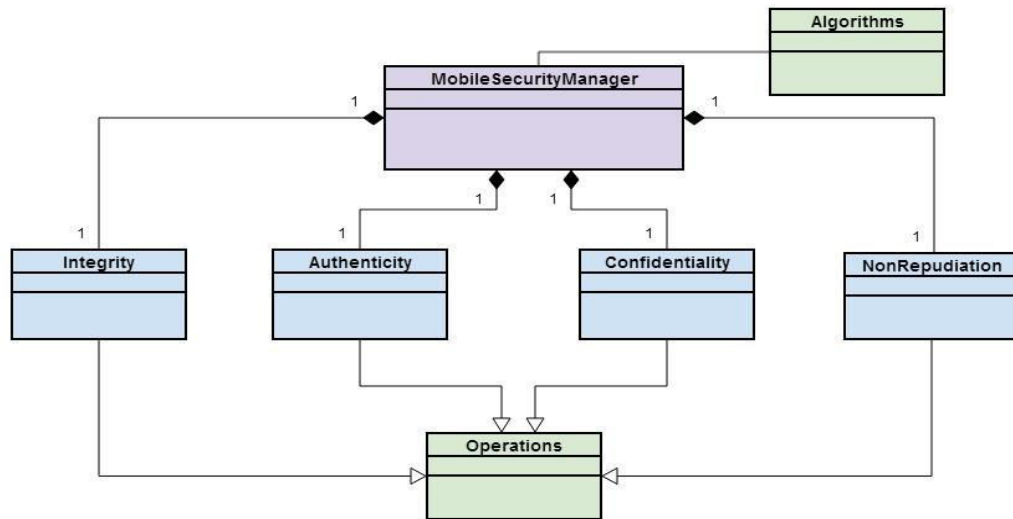


Figure 2. Security Part – Class diagram

The Auxiliary Part

In this section is presented the implementation of the Policy Manager and the State Manager. The implementation consists of two classes: PolicyManager and StatusManager as it can be seen in Figure 3.

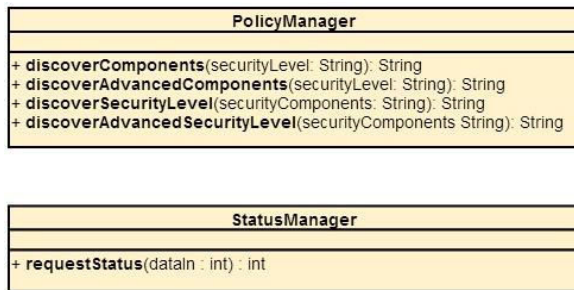


Figure 3. Auxiliary Part – Class diagram

The Policy Manager manages the security composition rules. These rules define the security properties (components) combination specific to a certain level of security.

Into the implementation there two types of encoding for a security level :

- **Basic code:** it defines the combinations of the security properties. Its form is as follow: C[n]; where C stands for combination and n it is a number (e.g. C7). This code is used when the user is of type standard or intermediary.
- **Advanced code:** it also defines the combinations of the security properties; but also includes the security algorithm chose by the user. Its form is as follow: AC[n]; where AC stands for advanced combination and n it is a number (e.g. AC33). This code is used when the user is of type advanced.

The PolicyManager includes four methods as it can be seen in Figure 3:

- **discoverComponents():** receives a basic code of security level as input and returns a string with the corresponding security properties. This string contains the first letter of each security component name, if that component corresponds to the security level; the letters are separated by a colon. An example is presented in Figure 4.

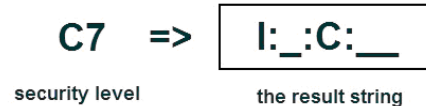


Figure 4. Operating example for discoverComponent() method

- **discoverAdvancedComponents():** receives an advanced code of security level as input and returns the string of corresponding security properties together with the corresponding algorithms. This string contains the first letter of each security component name, if that component corresponds to the security level, and a number which represent the security algorithm the user has chosen. All the information is separated by a colon. An example is presented in Figure 5.

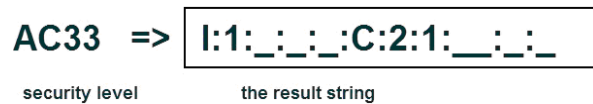


Figure 5. Operating example for discoverAdvancedComponent() method

- **discoverSecurityLevel()** and **discoverAdvancedSecurityLevel():** are the reverse operations of the methods presented above.

The StateManager class implements the functionality of

the State Manager. On the current implemented version the StateManager class does not collect the energy level of the device. It contains a method that returns a certain number according to the value received as input.

The Analysis System

This section will present the implementation of the Analysis System. The Analysis System is integrated in the Mobile Manager. The class diagram of the Analysis System is shown in Figure 6; it can be seen here the connection between the various classes. The implementation for the Analysis System consists of two classes and one interface: MobileManager (the interface), MobileManagerParA and MobileManagerPartB (the classes).

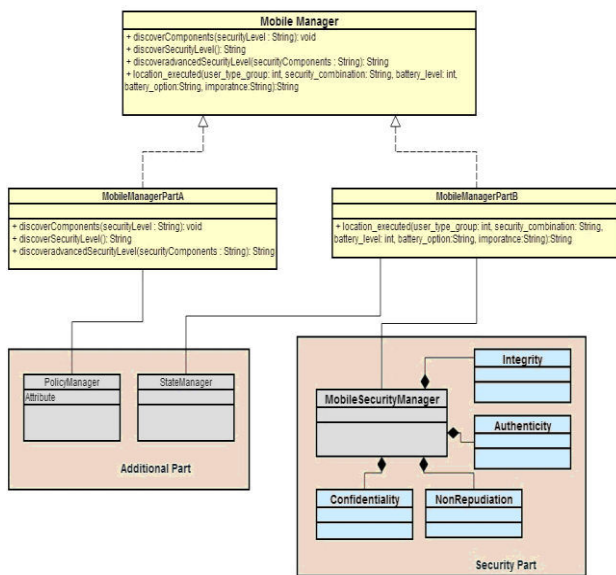


Figure 6. Analysis System - Class diagram

- *MobileManagerPartA class*: was implemented as part of the process that deals with the capture of users choices. It is the link between the user interface and the PolicyManager class. The methods implemented here are designed to use the PolicyManager methods in order to discover the adequate security level for a certain security properties combinations (or for the reverse operation).
- *MobileManagerPartB class*: was designed in order to implements the Analysis System The method that handles the Analysis System functionality is called location_executed(). This method receives as input the users constrains chosen through the user interface. It returns the security combination, the security algorithms and the execution location.

The Databases

The databases used by the framework are the following: Admin, Applications, and User.

The Admin database was designed to keep the default information needed by the security framework. The default information is data already predetermined; and refers to the general type of applications and the security properties combination encoding. This scheme contains three tables, described in Table II.

TABLE II. ADMIN DATABASE – TABLES DESCRIPTION

Table Name	Role Description
Types Table	It contains the types of applications.
Combination Encodes Table	It contains the security properties combination encoding.
Security Combinations Table	It contains the links between the two tables previously defined.

The Applications database was designed in order to keep the user options regarding the data security level of a certain mobile cloud application. This scheme contains four tables, described in Table III.

TABLE III. APPLICATIONS DATABASE – TABLE DESCRIPTION

Table Name	Role Description
Applications Table	It contains the list of the mobile cloud applications installed on the mobile device.
Applications Security Table	It contains the user options regarding the data security level for a certain mobile cloud application.
Applications Battery Table	It contains the user options regarding of to preserve or not the battery while a certain mobile cloud application is running.
Applications Priority Table	It contains data that specifies which of the two constraints: security or battery is more important for the user.

The User database was designed in order to keep the user options regarding to his level of knowledge in the security field. This database has only one table: “User Level Table”.

To store information into the database, it has been used the SQLite database in Android applications. SQLite is an Open Source database. SQLite supports standard relational database features like SQL syntax, transactions and prepared statements. The advantages for using SQLite are: 1) the database requires limited memory at runtime; 2) SQLite is embedded into every Android device; 3) it is not required a setup procedure or the database administration; it is only necessary to define the SQL statements for creating and updating the database.

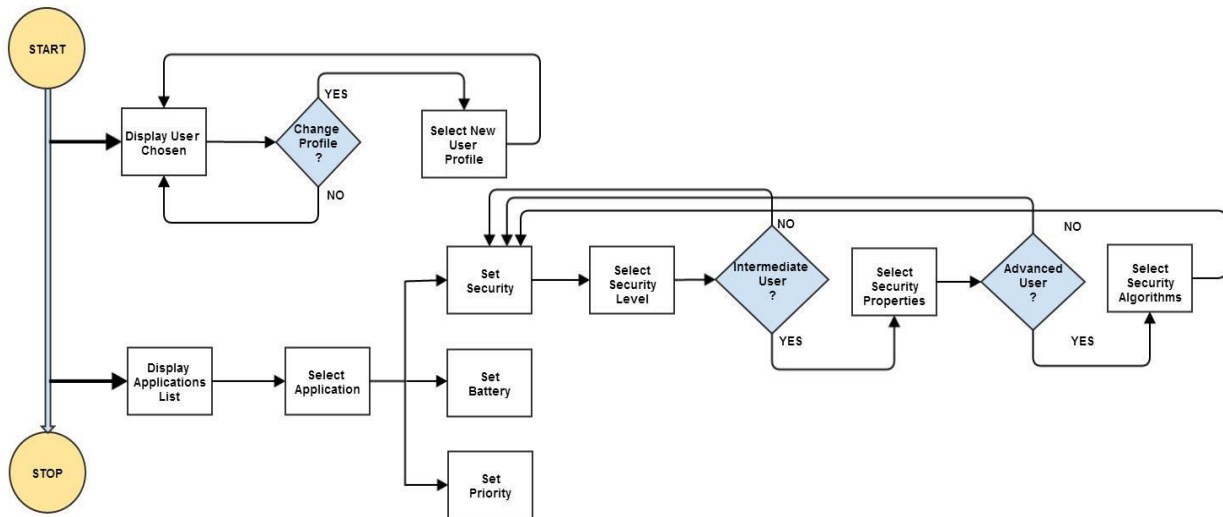


Figure 7. User Interface - Functionality diagram

The User Interface

The user interface was designed in order to capture the user option regarding the security level of his data and also regarding the device energy consumption. The user interface functionality is presented in Figure 7. Its functionality is divided in two phases: 1) setting the user profile and 2) setting the security.

The first phase allows the user to select the group to which it belongs according to his level of knowledge in the security field. As it can be seen in the Figure 8, there were defined three types of users: 1) Standard User Type, 2) Intermediary User Type and 3) Advanced User Type.

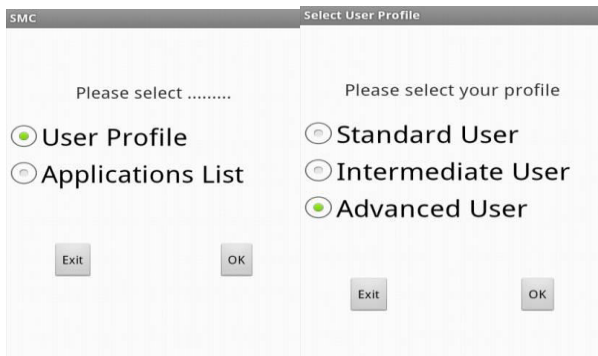


Figure 8. User Interface – The user type

The second phase, as its name suggest, allows the user to select data security level (Set Security step in Figure 7). In the case of the application users it comes to following question: How much flexibility a user shall have? Through flexibility it is understand the number of constrains (e.g. security level, security properties combination, security algorithms) the user can define. We decided for the flexibility to vary according to the user type. The Standard User Type is provided with the lowest flexibility and the Expert User Type has the greater flexibility. The lowest flexibility includes only the security level (e.g. strong,

average). The grater flexibility includes, besides the security level, the security properties combinations and the security algorithms (see Figure 9). Also in this phase, all the users are allowed to chose if they want to save or not the mobile device energy (Set Battery step in Figure 7). In addition, all the users have to specify which of these two constraints: 1)security and 2)battery is more important for them (Set Priority in Figure 7).



Figure 9. User Interface – The security properties and algorithms

C. Unit tests

In this section there are presented a couple of test. These tests target the security framework functionality.

The first scenario:

The user is of type advanced. He chooses the following options: 1) all data are secured equally regardless of the sensitivity level; 2) security level of type average; 3) as security properties he chooses only confidentiality; 4) as security algorithms he chooses: SHA(Secure Hash Algorithm), AES(Advanced Encryption Standard) and RSA; 5) he chooses to save battery; and 6) the priority is also the battery.

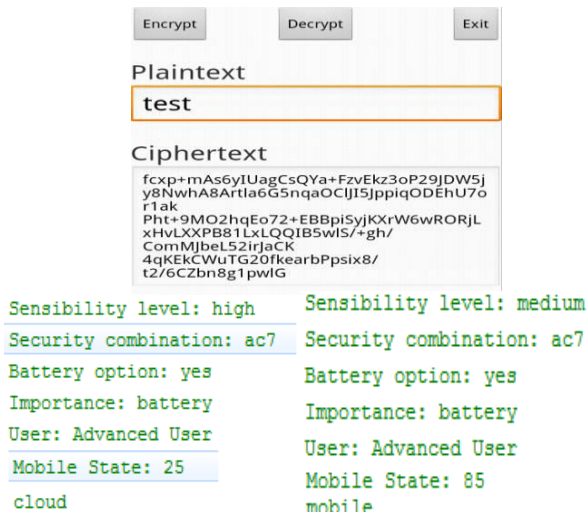


Figure 10. Results – Unit test first scenario

It can be seen in Figure 10 that, for data with different sensibility level (e.g. high and medium), there is the same security combination (e.g. ac7). Also, according to the mobile energy status (e.g. 25 or 85), one operation is executed on the mobile device (the result is also shown in Figure 10) and the other in Cloud.

The second scenario:

The user is of type standard. He chooses the following options: 1) all data are secured according to the sensitivity level; 2) security level of type average is chosen for low sensitivity data; 3) security level of type strong is chosen for medium and high sensitivity data; 4) he chooses not to save battery; and 5) the priority is the security.

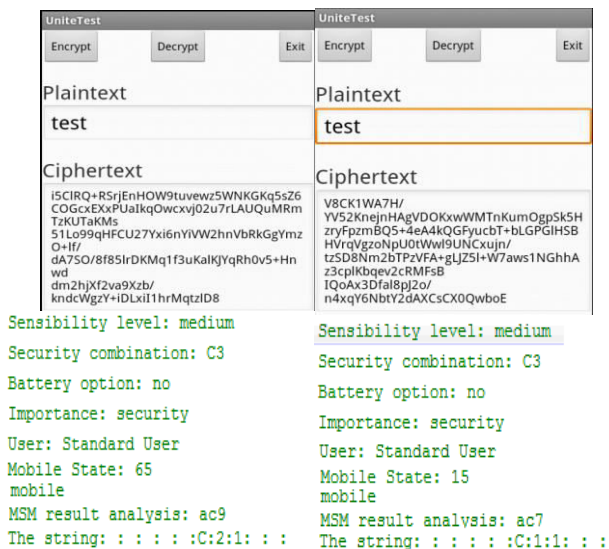


Figure 11. Results – Unit test second scenario

It can be seen in Figure 12 that, for data with different sensibility level there are different security combinations (e.g. C7[ac35] and C1[ac2]). The mobile energy status also can influence the security algorithm (Figure 11, security combination ac9 or ac7).

The security framework implementation on the mobile device was made using Java programming language and the

Android [8] mobile platform. The programming environment used was Eclipse.

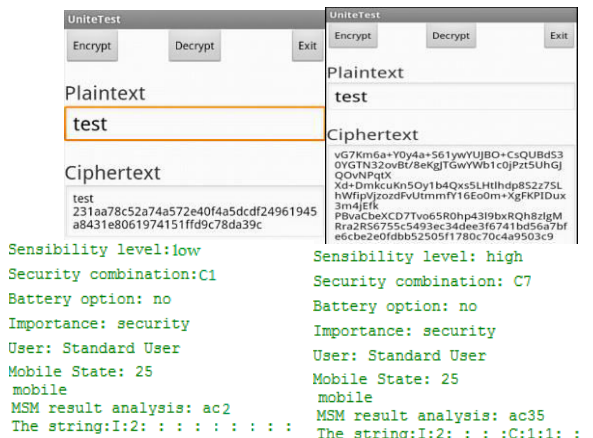


Figure 12. Results – Unit test second scenario

III. CONCLUSIONS

This paper describes the Secure Mobile-Cloud Framework implementation on the mobile device and the implementations details about the security components, the mobile security manager, the policy manager, the state manager and the analysis system. In order to allow the user to express his/her requirements regarding the security level to be applied to his/her data, a user interface was implemented. The information collected from users and the information from the analysis system are stored in a local SQLite database. Several unit tests were implemented in order to verify the security framework functionality. As future development we intend to integrate the proposed security framework into a mobile cloud application.

ACKNOWLEDGMENT

This paper was supported by the project: Improvement of the doctoral studies quality in engineering science for development of the knowledge based society-QDOC" contract no. POSDRU/107/1.5/S/78534, project co-funded by the European Social Fund through the Sectorial Operational Prog. HR 2007-2013.

REFERENCES

- [1] S. Gautam Kumar, K. Dinesh, Mathew K. and Abhimanyu M.A. "Cloud Computing for Mobile World".
- [2] D. Kovachev, Y. Cao and R. Klamma, "Mobile Cloud Computing: A Comparison of application Models", in eprint arXiv: 1107.4940, July 2011.
- [3] Lookout Mobile Security, Lookout Mobile Threat Report, Aug. 2011.
- [4] Cloud Security Alliance, Top Threats to Cloud Computing V 1.0, March 2010.
- [5] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "A Security Framework for Mobile Cloud Applications", in Proceedings ROEduNet 11 th International Conference, Sinaia, 2013.
- [6] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "A System to Analyze the User's Security Options for Mobile Cloud Applications", The 6th International Conference on Security for Information Technology and Communications, June 25, 2013.
- [7] M. Kamel, K. Boudaoud, S. Resondry and M. Riveill, Low-Energy Consuming and User-centric Security Management Architecture Adapted to Mobile Environments, in Proceedings of the 12th IFIP/IEEE, Dublin, Ireland, May, 2011.
- [8] R. Rodger, "Beginning Mobile Application Development in the Cloud", WROX Programmer to Programmer.

Integration of Secure Mobile-Cloud Framework into a mobile cloud application scenario

D. Popa¹ K. Boudaoud² M. Cremene¹ M. Borda¹

Abstract – Mobile Cloud Computing triggered the implementation of several applications that easily provide the users with more complex features and characteristics. Secure Mobile-Cloud Framework is designed in order to secure users private data transmitted between the same mobile cloud applications. This work will present a solution to integrate the Secure Mobile-Cloud Framework into a healthcare application scenario.

Keywords: Mobile Cloud Computing, Security, Mobile Cloud Applications

I. INTRODUCTION

Mobile Cloud Computing [1] is a new concept, which offers Cloud Computing resources and services for mobile devices.

Cloud Computing [2] is a new technology that provides, to the Internet users, data, resources, platforms, and applications as services.

In the last years mobile phones have greatly developed, and because of the small size and also because they can be moved easily, they become indispensable to users.

Using Mobile Cloud Computing advantages, new application models were developed for mobile devices. These applications models try to use both the mobile device resources and the Cloud services to provide a more reach and a more varied functionality in order to increase the mobile device popularity and use.

From a security point of view, Mobile Cloud Computing, increases the security risks and privacy invasion due to the fact that it combines mobile devices with Cloud services and also because there is not a well-defined application model [3].

The security issues are treated independently and the existing security solutions are supplied separately by various providers (Cloud providers or mobile platforms). Thereby, in the case of mobile cloud applications, it is needed to combine different solutions in order to secure them. Also, very few solution proposed to secure the data or the applications take into account the mobile device

energy constraints, users constraints or data sensitivity constraints.

The Secure Mobile-Cloud Framework proposed in [4] focuses on component based mobile cloud applications. Its goal is to secure the communication between the application components. The feature of this solution is the fact that it tacks into consideration the following constraints: mobile device energy, data sensitivity and users' options. The security framework is composed of components deployed on the mobile device and components deployed in Cloud.

In this paper we discuss the integration of the security framework into a mobile cloud application scenario.

The paper is organized as follow. Section II briefly describes the security framework. In Section III, it is presented the integration solution. This section also describes the design of a mobile cloud application scenario. Finally, in section IV, several conclusions are presented.

II. SECURE MOBILE-CLOUD FRAMEWORK SHORT OVERVIEW

The Secure Mobile Cloud (SMC) framework is composed of two types of components, as presented in [5]: 1) security components and 2) management components.

The security components have been designed in [6] for the LECCSAM architecture. Their role is to implement the eponym security properties (e.g. integrity, authenticity, confidentiality, non-repudiation). These security components are deployed in both mobile device and Cloud. The management components have been designed to identify and apply the appropriate security properties and therefore security components to users' data. Some of these management components are deployed on the mobile device and some of them are deployed in the Cloud.

We want to allow the users to express their choices regarding the security level they want to apply to their data. In order to provide a solution that allows achieving this characteristic there was designed an analysis system. This system is part of the security

¹ Technical University of Cluj-Napoca, Communications Department,
Str. Dorobantilor. 71-73 CP. 400609 Cluj-Napoca, Romania, Daniela.Popa @com.utcluj.ro

²University of Nice Sophia Antipolis,
930 Route des Colles - BP 145- 06903 Sophia Antipolis

framework. And it has to provide the security combination needed to be applied to data (security combination = security properties + security algorithms); and also the location where this combination can be performed (e.g. on the mobile or in Cloud).

A briefly description of the management components is given in Table 1.

Table 1. Description of Managers

Manager	Description
Mobile Manager	It collects data and events that occurs on the mobile side and sends them to the appropriate manager to be analyzed.
Mobile Security Manager Cloud Security Manager	Both provide the composition of the security properties. The Mobile Security Manager ensures security composition on the mobile side and the Cloud Security Manager ensures the composition on the Cloud side.
Optimizati on Manager	It sends the information collected from sensors (e.g. network sensor, energy sensor) to the mobile manager.
Applicatio n Manager	It checks the application integrity at setup.
Policy Manager	It determinates which security components are required for a specific security level.

III. THE INTEGRATION

The term integration denotes the combination of the mobile cloud application scenario with the security framework. Furthermore, it refers to the way in which the application scenario will operate using the security framework.

For the integration solution it is assumed that there is access to the application scenario source code. The proposed solution for the integration is a static solution. A static solution assumes the integration at the code level before the application compiling and deploying.

It is also assumed that the keys (public and secret) are securely distributed. Also because the integration is done at the code level the authentication of the application components is not considered.

This section is split in two parts. The first part

presents the application scenario design and the second part describes the integration solution.

A. The mobile cloud application scenario

The proposed application scenario is a healthcare application. This application aims to monitor a person type according to his/hers bodies characteristics and to propose a regime. The application initial characteristics and functionalities are presented in the following.

The application captures the users' characteristics: body size (e.g. height, weight). According to the information received, a type of user is established. Depending of the user type, a certain regime is provided.

The application functionalities from the user point of view are (see also Fig.1):

1. Create an account, when a user is using the application for the first time. In order to do this the user must provide several private data. This private data are: the name, password, e-mail.
2. If the user already has an account he/she need to login.
3. Insert the user data, which may be of two types:
Personal data: day of birth and gender
Body measurements: height and weight
4. Obtain the user type. The user type is established by users' personal data and users' body size data. There are defined four types of users: weight-gain users, weight-loss users and normal users. The weight-gain users are too skinny and that they have to take in weight. The weight-loss users are too corporal and they have too loose weight. The normal users do not need to change their body size.
5. Obtain the regime. The regime denotes the types of food the user can or cannot consume.

This application scenario is intended to be a Mobile Cloud Computing application, based on components running on the mobile side or in Cloud.

The application was split in three major parts. These parts are:

- *The user interface.* On the mobile device. It is designed in order to collect the users data.
- *The database.* On the Cloud side. It is designed in order to store the data.
- *The services.* On the Cloud side. There are designed in order to execute the following application functionalities: compute the user type, compute the BMI (body max index), and compute the regime. For

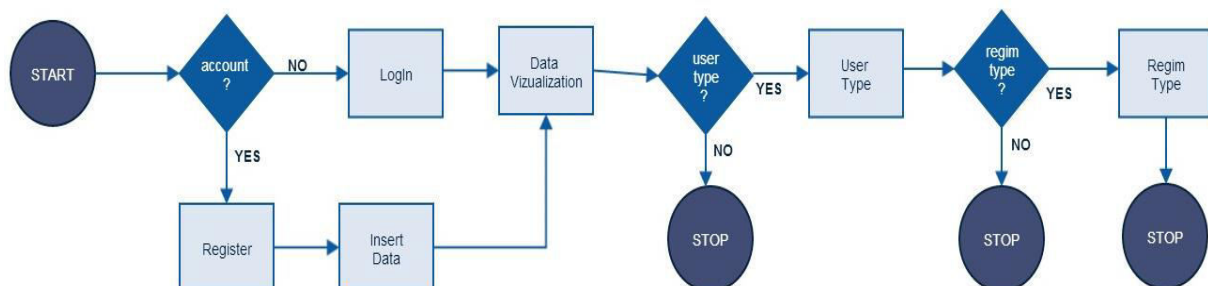


Fig.1 Application Scenario functionality

each functionality there was designed a service.

As it can be seen in Fig.2 the user interface was designed to be a component; the database was also designed as another component; and each service represent a different component.

B. The integration solution – theoretical approach

As previously said the integration assumes the combination of the application scenario with the security framework. In this way it is obtained a new application, the application scenario secured; an application that has the functionalities of the application scenario, but that it also has the data security ensured.

The problems that need to be resolved here are the following: 1) where and how the application scenario will call the security framework to secure the data; 2) which is the flow of the new application scenario secured.

As solution to the first issue, for each component, on the mobile device (user interface) or in Cloud (services), it was designed a part, called ApplySecurity, in order to communicate with Mobile Manager, in the case of the user interface, or with Cloud Security Manager in the case of a service in Cloud.

A solution to the second issue is described below:

From the description of the scenario application in the previous section it can be seen that the application includes three types of communications: 1) the communication between the mobile device and the database; 2) the communication between the mobile device and the services; and 3) the communication between two services.

The communication between the mobile device and the database assumes the following actions: 1) saving data provided by the application user into the database; and 2) requesting data from the database in order to be displayed on the application interface.

The communication between the mobile device and the services assumes calling the services and providing the proper users' private data in order to obtain a certain result; the result may be or not a users' private data.

The communications between two services refers to one service calling the other services and providing the proper data (private or not) in order to obtain certain data as result (private or not).

In this work it will be presented the secured flow for the communication between the mobile device and the database.

The secured application scenario flow in the case of communication between the mobile device and the database requires several steps (see Fig. 3). This steps are:

S-1. Users' private data, retrieved by the user interface, who need to be sent into the Cloud database are intercepted by the ApplySecurity part and sent to the Mobile Manager along with their sensibility level;
S-2. The Mobile Manager, uses its analysis functionality to verify where is more suitable to apply the security, on the mobile device or in Cloud;

S-3. If the mobile device is chosen by the Mobile Manager, the data along with the security level (the combination of security properties along with the security algorithms) are sent to the Mobile Security Manager;

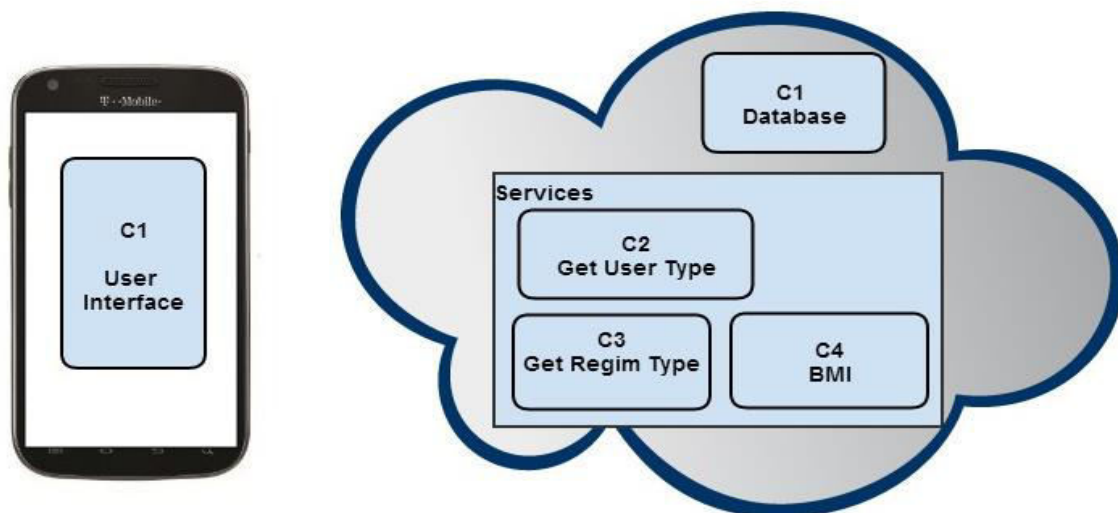


Fig.2 Application Scenario Components

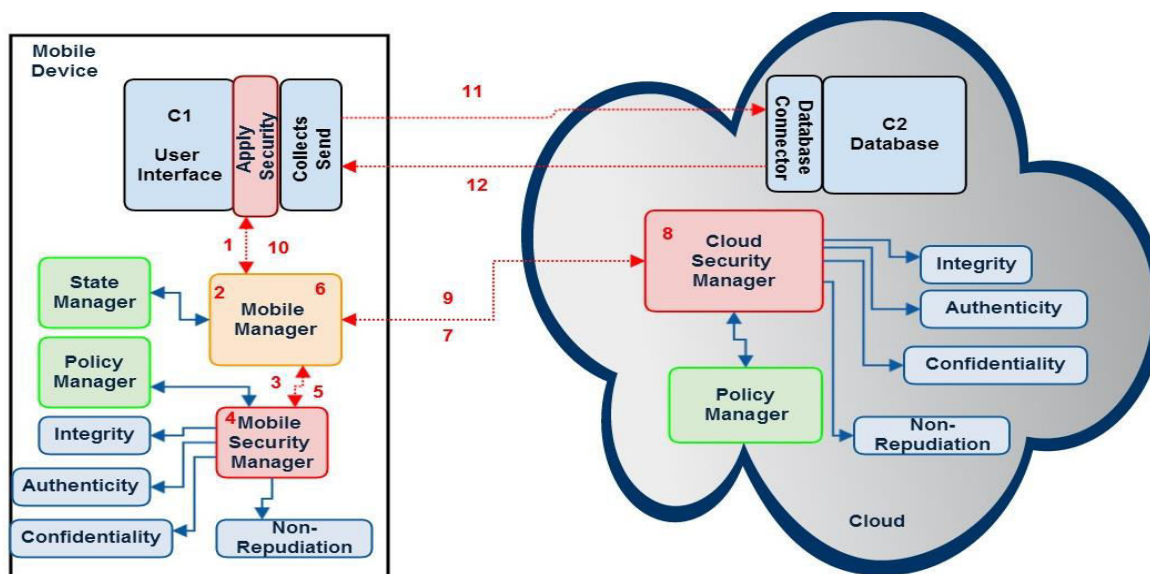


Fig.3 Secured Application Scenario Flow

S-4. After discovering which security properties and security algorithms correspond to the received security level, the Mobile Security Manager orchestrates the application of the appropriate security properties (components) to the received data;
 S-5. When the security operation is finished, the secured data are sent back to Mobile Manager;
 S-10. The secured data along with the security level are sent to the ApplySecurity part;
 S-11. The secured data are sent to the Cloud database;
 S-12. A response message of success or of error is received;

S-6. If the Cloud is chosen by the Mobile Manager, data along with the security level are ciphered using a secret key;
 S-7. Encrypted data are sent to the Cloud Security Manager;
 S-8. After decrypting the received data, Cloud Security Manager orchestrates the application of the appropriate security properties (components) to the received data;
 S-9. When the security operation is finished, the secured data are sent back to Mobile Manager;
 Then, the steps from S-10 to S-12 are followed.

C. The integration solution – technical approach

This section includes the application scenario implementation and it shows from a technical point of view where in the application scenario it must intervene in order to add the security framework. At this time from the application scenario it has been implemented only the communication between the mobile device and a database stored in Cloud. The main languages that were used to implement the application scenario are Java, MySQL, PHP and JavaScript Object Notation (JSON). The interface on the mobile device was implemented using Java based on Android, PHP was used in order to access and query the database and for the database

implementation was used MySQL. Then, in order to make a connection between the mobile device and the server the JSON data-interchange format was used. For the communication between the mobile device and a database in Cloud the application design consists of: the user interface, the collect/send part, the connector part and the database. Thus, the user interface consists of three screens: LogIn screen, Register screen and Details screen. The collect/send part consists of two classes implemented in Java: UserFunctions and JSONParser. The connector part consists of two files: index and DBFunctions implemented in PHP. The database is implemented in MySQL.

The UserFunctions class was implemented in order to send the user data, retrieved from the user interface, to the Cloud database. It implements the following methods:

- loginUser(): it implements the login requests
- registerUser(): it register the user login details (name, e-mail and password) into the external database
- registerUserInfo(): it register the user other info like: gender, height, weight, day of birth into the external database

The JSONParser class implements the following method:

- getJSONFromUrl(String url, List<NameValuePair> params): it passes and receive the data from and to the server side implemented in PHP.

The index file handles all the request coming from the mobile device. The DBFunctions include the implementation for DBFunctions class methods. The class implements two methods: storeUser and storeNewUserInfo. Their role is to implement the operations of insert into the database.

The database contains two tables:

- users: to store the register data: name, e-mail and password

- info: to store the users information like gender, height, weight and day of birth

The following section will present: 1) how there were defined and integrated, into the application code, the constraints that the application designer should specify; 2) in what consist the ApplySecurity part mentioned in Theoretical approach section; and 3) the modifications made to the user interface.

The constraints that the application architect needs to specify are the application type and the data sensitivity level.

For the application type constraint, the solution applied was to use in the application manifest file the attribute 'description' (see Fig. 4). This description attribute will take the application type value; for this particular application scenario the value is health. To retrieve the attribute description value, it was implemented into the main user interface page (UIMainActivity.java) a method called getApplicationType (see Fig. 4). The getApplicationType method uses the style resource identifier getApplicationInfo.descriptionRes, provided by android.content.Context in order to identify in the application resources the attribute description value.

```
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:description="@string/app_type"
    android:theme="@style/AppTheme" >
    <activity
        android:name="com.app.sampleUI.StartApiSampleActivity"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    ..
    app_name = getApplicationName(getApplicationContext());
    app_type = getApplicationType(getApplicationContext());
    public static String getApplicationType(Context context) {
        int stringId = context.getApplicationInfo().descriptionRes;
        return context.getString(stringId);
    }
}
```

Fig.4 The application type constraint

For the sensitivity level constraint, the solution applied was to define several private static variables into the class (UserFunctions) that is in charge of the data transmission. These variable are: add_security, highS, mediumS and lowS as it can be seen in the Fig.5.

```
/*for security*/
private static String add_security = "add_security";
private static String highS = "high";
private static String mediumS = "medium";
private static String lowS = "low";
/*for security*/
```

Fig.5 Sensitivity level constraint - solution

In the theoretical approach section was mentioned the ApplySecurity part whit the functionality of collecting

the user private data and sending them to the Mobile Manager.

ApplySecurity part and its functionality were implemented into each UserFunction method that sends data to the database as follow:

While the list whit pairs of type <parameter,value> is build, it is inserted before each pair an additional pair of type <S_level, value> as it can be seen in the Fig. 6. After the list is built, the apply_security_encrypt() method in MobileManagerPartB class is called. This method will return a list of type <parameter,value>, where the values are secured.

Then the data may be sent to the database.

```
params.add(new BasicNameValuePair("security", add_security));
params.add(new BasicNameValuePair("tag", register_tag));
params.add(new BasicNameValuePair("num_reg", num_regist1));
params.add(new BasicNameValuePair("S_level", highS));
params.add(new BasicNameValuePair("name", name));
params.add(new BasicNameValuePair("S_level", highS));
params.add(new BasicNameValuePair("email", email));
params.add(new BasicNameValuePair("S_level", highS));
params.add(new BasicNameValuePair("password", password));

params_json = SMC_manager.apply_security_encrypt(context,db_app_name,params);

JSONObject json = jsonParser.getJSONFromUrl(registerURL, params_json);
```

Fig.6 ApplySecurity Part

For the user interface modification (see Fig. 7); it was introduce a new main user interface from which the user may choose to set the security settings or to start the application. In the case the user won't set the security settings, it will be applied to all data without taking into consideration the data sensitivity level, the default security level for a standard user type.

V. CONCLUSIONS

A healthcare mobile cloud application scenario was presented in this paper. The application aims to monitor a person type according to her/his bodies characteristics, and then to propose a regime. The application was split into components. The user interface was designed as a mobile device component, while the database was designed as a Cloud component. Three services running as components in the Cloud were also designed.

In this work was also presented a solution to integrate the Secure Mobile-Cloud Framework into the healthcare application scenario. The intended result was to obtain only one application. The new obtained application has the healthcare application functionalities, but it also has the security of data transmitted between components. The integration is mad at the source code level.

For the integration solution, two approached are discussed, the theoretical approach and the technical approach. The theoretical approach shows where in the application scenario flow it is needed to be done the call to the Secure Mobile-Cloud Framework components. The technical approach, show the modifications brought to the application scenario

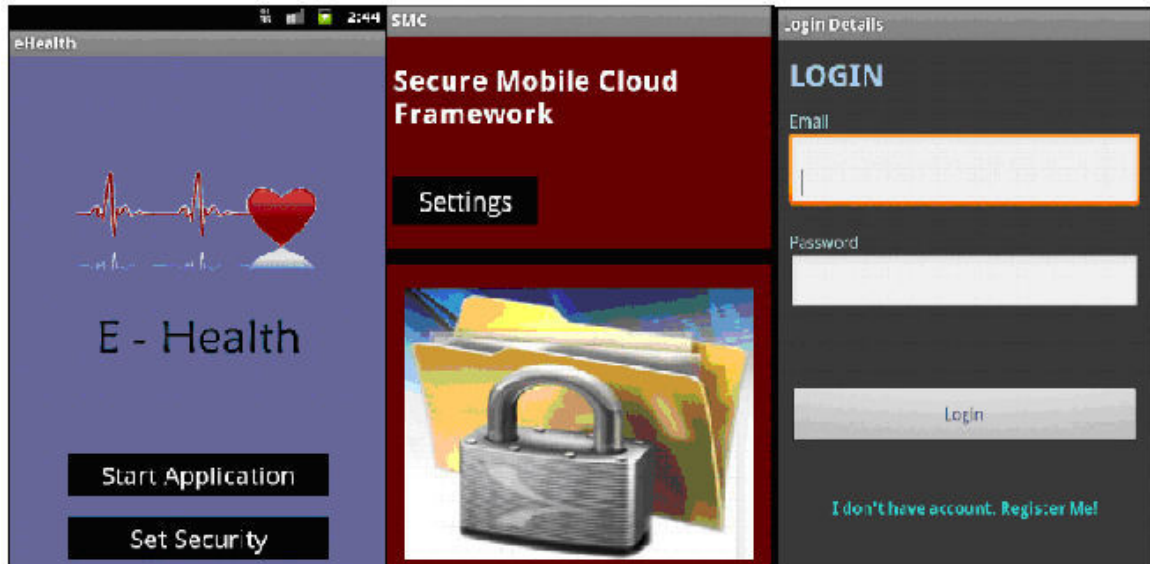


Fig.7 User Interface

source code in order to integrate the security framework.

ACKNOWLEDGMENT

This paper was supported by the project: Improvement of the doctoral studies quality in engineering science for development of the knowledge based society-QDOC" contract no. POSDRU/107/1.5/S/78534, project co-funded by the European Social Fund through the Sectorial Operational Prog. HR 2007-2013.

REFERENCES

- [1] B.G. Chun and P. Maniatis, "Augmented Smartphone Applications Through Clone Cloud Execution," in Proceedings of the 12th Workshop on Hot Topics in Operating Systems (HotOS XII), USENIX, 2009.
- [2] Jeremy Geelan. "Twenty one experts define cloud computing", Electronic Magazine, available online: <http://virtualization.sys-con.com/node/612375>, August 2008.
- [3] C. Nachenberg, "A Window Into Mobile Device Security – Examining the security approaches employed in Apple's iOS and Google's Android", Symantec Security Response.
- [4] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "A Security Framework for Mobile Cloud Applications", in Proceedings ROEduNet 11 th International Conference, Sinaia, 2013.
- [5] D. Popa, K. Boudaoud, M. Cremene, M. Borda, "A System to Analyze the User's Security Options for Mobile Cloud Applications", The 6th International Conference on Security for Information Technology and Communications, June 25, 2013.
- [6] M. Kamel, K. Boudaoud, S. Resondry and M. Riveill, Low-Energy Consuming and User-centric Security Management Architecture Adapted to Mobile Environments, in Proceedings of the 12th IFIP/IEEE, Dublin, Ireland, May, 2011.

Overview on Mobile Cloud Computing Security Issues

D. Popa¹ K. Boudaoud² M. Cremene¹ M. Borda¹

Abstract – Mobile Cloud Computing, the combination of mobile devices with Cloud Computing services. It brings several advantages to the devices with low resources; advantages that lead to the development of rich functionality applications. The security issues in Mobile Cloud Computing can be classified as follows: mobile threats and cloud threats. The main purpose of these menaces is to steal personal data (e.g. credit card numbers, passwords, contact database, calendar, location) or to exploit mobile device resources. This paper is an overview on Mobile Cloud Computing security issues.

Keywords: Mobile Cloud Computing, Security Issues

I. INTRODUCTION

Just a short time ago a user was only expecting from her/his mobile phone to allow her/him to perform activities using just the device resources (e.g. to take pictures and save them locally on the device, or to read different types of files that were saved locally).

Today, the same user wants to be able to take advantage of powerful and complex applications that manipulate not only the mobile local resources but also external resources as computation power and storage place. To obtain these types of performances several improvements have been made in the domains of mobile hardware and network [1]. Even with those improvements mobile devices still have a lack of resources and energy, an unstable connectivity and introduce several security issues.

To resolve some of these issues, the concept of Mobile Cloud Computing has been proposed as a solution where the Cloud is used as a platform to execute mobile applications. Mobile Cloud Computing as a term was born shortly after the emergence of Cloud Computing model in 2007 [2]. Marketing research [3] stated that in 2015 there would be more than 240 million customers using Mobile Cloud Computing services while in 2008 there were only 42.8 million customers.

Thanks to the emergence of Mobile Cloud Computing different novel mobile applications models have been defined where the Cloud is used to overcome the limitations imposed by mobile devices such as processing power, memory capacity and display size.

Mobile devices are vulnerable to numerous security threats that aim the theft of users' data. Moreover Cloud Computing introduces several security, privacy and trust issues regarding the data stored in the Cloud. Consequently to maintain consumer's trust in mobile platforms more specifically in mobile cloud applications, it is important to secure data that will be used and processed by mobile cloud applications.

In this paper we present an overview of Mobile Cloud Computing security issues. The paper is organized as follow. Section II presents what Mobile Cloud Computing is, by showing several main characteristics described in various papers. In Section III, there is presented the overview on Mobile Cloud Computing security issues, namely the mobile threats and the Cloud threats. Finally, in section IV, several conclusions are presented.

II. WHAT IS MOBILE CLOUD COMPUTING?

In order to answer the question raised in the section title, it will be presented in the following, several definition and characteristics of Mobile Cloud Computing along with the mobile cloud applications models.

A. Definitions

Mobile Cloud Computing (Fig.1) is a new concept that can be described as the availability of Cloud Computing resources and services for mobile devices. As in the case of Cloud Computing, several definitions were proposed to define Mobile Cloud Computing.

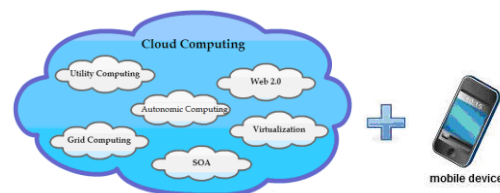


Fig.1 Mobile Cloud Computing

¹ Technical University of Cluj-Napoca, Communications Department,
Str. Dorobantilor, 71-73 CP. 400609 Cluj-Napoca, Romania, Daniela.Popa @com.utcluj.ro

²University of Nice Sophia Antipolis,
930 Route des Colles - BP 145- 06903 Sophia Antipolis

As in the case of Cloud Computing, there are several opinions on what Mobile Cloud Computing is. There is not a consensual definition for Mobile Cloud Computing.

Mobile Cloud Computing is defined in [4] as follows: *“Mobile cloud computing at its simplest refers to an infrastructure where both the data storage and the data processing happen outside of the mobile device. Mobile cloud applications move the computing power and data storage away from mobile phones and into the cloud, bringing applications and mobile computing to not just smart-phone users but a much broader range of mobile subscribers.”*

Another definition given in [5]: *“Mobile cloud computing is a model for transparent elastic augmentation of mobile device capabilities via ubiquitous wireless access to cloud storage and computing resources, with context-aware dynamic adjusting of offloading in respect to change in operating conditions, while preserving available sensing and interactivity capabilities of mobile devices.”*

The first definition emphasizes that Mobile Cloud Computing benefits from Cloud Computing features – storage and data processing, and also reveals a Mobile Cloud Computing characteristic – moving part of the computation and the storage away from mobile phones.

The second definition is more concise. It starts by saying what is Mobile Cloud Computing – a model; it also tells the purpose of using Mobile Cloud Computing – to overcome the mobile device challenges; it tells the way – using storage and computation resources offered by Cloud Computing model; it also specifies that is appropriate to take into account the context of the mobile operating conditions.

As a conclusion, we can say that Mobile Cloud Computing offers Cloud Computing resources such as storage and computations to the mobile devices with limited CPU speed, memory capacity and display size which allows the development, deployment and execution of powerful mobile applications.

III. MOBILE CLOUD COMPUTING - SECURITY

Mobile Cloud Computing exposes private data of the mobile user to different security risks. User’s data can be stored on the mobile side or on the Cloud side, can be accessed by applications (or application components) running on the mobile device or in Cloud, or can be transmitted between the mobile device application components and Cloud application components.

This section presents in the first part the security issues related to Mobile Cloud Computing and highlights in the second part the state of the art work proposed to address these security issues.

As we have said previously, Mobile Cloud Computing is a combination of mobile and Cloud Computing. Thus, the security issues in Mobile Cloud Computing are due to the security threats against the Cloud, the mobile devices and the applications running on these devices. These threats can be classified as follows: mobile threats and cloud threats. The main purpose of these menaces is to steal personal data (e.g. credit card numbers, passwords, contact database, calendar, location) or to exploit mobile device resources.

A. Mobile Threats

A little while ago the malware development for mobile devices was seen as a myth due to their limitations in terms of hardware and software.

Table 1. Key Characteristics of Mobile Cloud Computing

Characteristics/ Papers	Model	Combination	Outside device processing	Outside device storage	Elasticity	Remote access
R.D.Caytiles [6]	X	X	X	X		
A. Khan [7]	X		X	X		X
S.K.Ko [8]		X				X
H.T.Dinh [9]		X	X	X		
AEPONA [5]	X		X	X	X	
MCCForum [4]	X		X	X		
J.H.Christensen [10]	X	X				
A.N.Khan [11]			X	X	X	X
H.Qi [12]	X	X	X	X		X
N.Fernando [13]	X	X	X	X	X	X

Nowadays, the increasing use and development of mobile devices (e.g. smartphones) has led to the evolution of mobile threats; from the first case of malware on mobile devices in 2004 targeting Symbian, to the code of DroidDream, DroidKungFu and Plankton discovered in 2011 in the official Android Market [14].

Recent studies [15], [16] have classified mobile attacks in several categories such as: application based attacks, web-based attacks, network based attacks and physical based attacks.

The application based attacks concern both offline and online applications. In these kinds of attacks are included: malware, spyware and privacy threats.

- Malware is software that performs a malicious behavior on a device without the user being aware of this behavior (e.g. sending unsolicited messages and increasing the phone's bill or allowing an attacker to have the control over the device).
- Spyware is software designed to collect private data without the user's knowledge (e.g. phone call history, text messages, camera pictures).
- Privacy Threats are caused by applications (malicious or not), that in order to run they need more sensitive data such as location (e.g. location based applications).

The web-based attacks are specific to online application and include: phishing scams, drive-by-downloads, or browser exploits.

- Phishing scams aim stealing information like account login and password.
- Drive-by-Downloads is a technique that allows the automatic download of applications when a user visits a certain web page.

In addition to these attacks, attackers use different techniques to obtain private data: repackaging, misleading disclosure and update.

- Repackaging was the most used technique in 2011 to infect applications running under Android [15]. In this kind of attack, an attacker takes a healthy application; modifies it with a malicious code and then republishes it. The main difference between the healthy and modified applications is that the last ones require more access control permissions such as to access the phone contacts or to send SMS messages.
- Misleading disclosure [15] is a technique used by an attacker to hide the undesirable functionality of an application, so that a user would not notice it and would agree to. The undesirable functionality is usually hidden in the applications terms and conditions. The attackers rely on the fact that usually the users do not pay attention to the applications terms and conditions while these are installed. Those applications are difficult to block or remove because they do not violate their own terms of service or any application market's user agreement.

- The update technique was recently used by malware writers as an attack method in Android Market [16]. Firstly, the malware writer publishes an uninfected application, than the application is updated with a malicious version. Using this technique, the attacker takes advantage of the users trust in the applications market. The number of infected devices increases; there are affected the users that only use the official market to download the applications. A consequence of this attack technique is a decrease of users' confidence in the application market. This may lower the market customers' number and therefore the market profits.

B. Cloud Threats

The Cloud acts as a big black box where nothing inside is visible to the clients. Therefore clients have no idea or control over what happens with their assets. Cloud Computing is about clients transferring the control of their resources (e.g data, applications) and responsibilities to one or more third parties (cloud services providers). This brings an increased risk to which client assets are greatly exposed.

Before Cloud's emergence, generally, the companies where keeping their data inside their perimeter and protecting them from any risks caused by malicious intruders. A malicious intruder was considered to be an outside attacker or a malicious employee. Now, if a company chooses to move its assets into the cloud, it is forced to trust the Cloud provider and the security solutions it offers when provided. However, even if the cloud provider is honest, it can have malicious employees (e.g system administrators) who can tamper with the virtual machines and violate confidentiality and integrity of client's assets.

In Cloud Computing the obligations in terms of security are divided between the cloud provider and the cloud user. In the case of SaaS, this means that the provider must ensure data and application security; so service levels, security, governance, compliance, and liability expectations of the service are contractually stipulated and enforced. In the case of PaaS or IaaS the security responsibility is shared between the consumer and the provider. The responsibility of the consumer's system administrators is to effectively manage the data security. The responsibility of the provider is to secure the underlying platform and infrastructure components and to ensure the basic services of availability and security [18].

Several analyses have been conducted to identify the main security issues regarding the Cloud Computing [17, 18, 19, 20, 21, 22, 23, 24, and 25]. Following these analyses, security issues have been classified in terms of concerns: domain concerns, services concerns, threats, actors concerns and properties concerns (Fig.2).

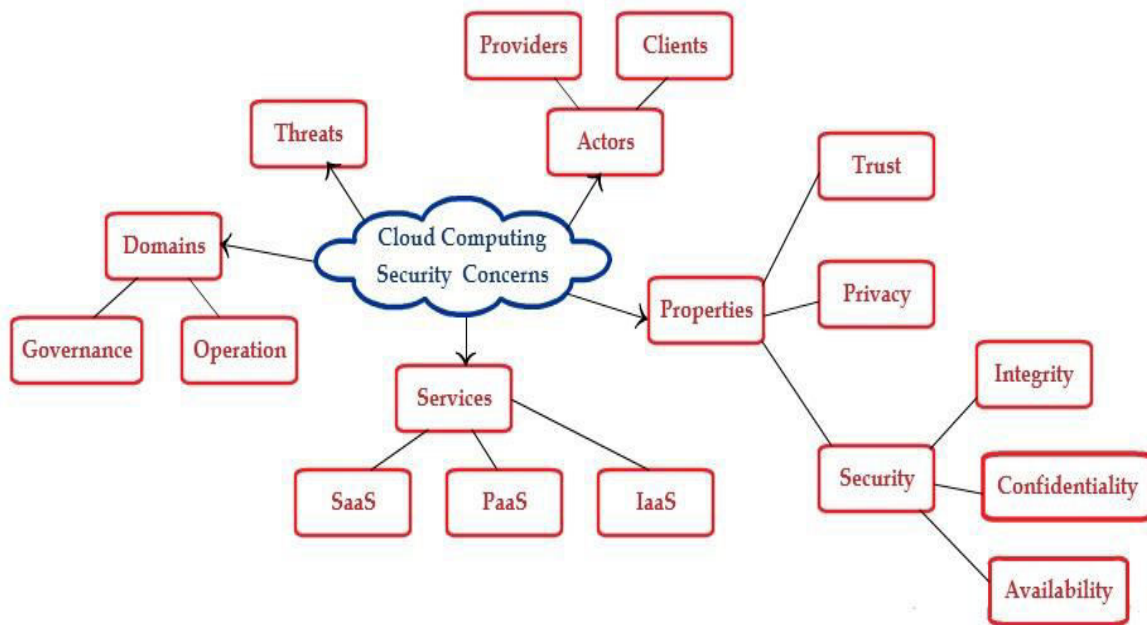


Fig.2 Cloud Computing terms of concerns

The domain concerns are divided in two types: 1) governance concerns and 2) operation concerns.

Governance addresses strategic and policy security issues within Cloud Computing [19]. The highlighted issues are: data ownership and data location. Data Ownership refers to the ownership of purchased digital data. Thanks to the Cloud it is possible to store purchased media files, such as audio, video or e-books remotely rather than locally. This can lead concerns regarding the true ownership of the data. If a user purchases media using a given service and the media itself is stored remotely there is a risk of losing access to the purchased media. The service used could go out of business, for example, or could deny access to the user for some other reasons [19]. Data location raises many issues because of the compliance problem of privacy laws that are different from a country to another. For example, the laws in European Union (EU) and South America are different from the laws in United States (US) regarding data privacy [18]. Under EU law [26] and South American law [27], personal data can be collected only under strict conditions and for a legitimate purpose. In the US, there is no all-encompassing law regulating the collection and processing of personal data [28].

Operation addresses technical security issues within Cloud Computing [19]; issues as: 1) the security of data stored into the Cloud, 2) the security of data transmitted between the Cloud services, 3) the security of data transmitted between the Cloud services and a mobile platform or 4) data access and integrity. If an application relies on remote data storage and Internet access in order to function then, any changes to these data can significantly affect the user.

Threats class identifies the main security issues an organization may face when it wants to move its assets into the Cloud. The main concerns mentioned are: data loss, unsecured applications interfaces, denial of services or malicious insider.

Actor class identifies the main security issues that may be caused by the Cloud provider, by the Cloud clients or by an outsider. Thereby, a Cloud provider may be affected by the malicious Cloud client's activities. The malicious Cloud clients can target honesty clients' data; they can legitimately be in the same physical machine as the target and they can gather information about the target. A Cloud client may be affected by the malicious Cloud provider. The malicious provider may log the client communication and read the unencrypted data; also it may peek into the virtual machines or make copies of the virtual machines assigned to run client assets. In this way a Cloud provider gain information about client data or behavior and sell the information or even use it itself. An outsider can affect a Cloud client. The outsider may listen to the network traffic or it may insert malicious traffic and launch the denial of service attack.

Services class lists the security issues that may occur while using any of the Cloud provided services: SaaS, PaaS or IaaS. The fundamental security challenges are: data storage security, data transmission security, application security and security related to third-party resources [21].

The properties that bring out the security issues encountered in the Cloud are: the privacy, the security and the trust. Security in general, is related to the following aspects: data confidentiality, data integrity and data availability. Privacy is one of the significant concerns in Mobile Cloud Computing. For example,

some smart phone applications use the Cloud to store user's data. The main risk in this context is that unauthorized people can access and get user's data. Another example concerns location-aware applications such as applications that finds nearby restaurants for the user; or applications that allows user's friends and family to receive updates regarding her/his location [29].

V. CONCLUSIONS

Mobile Cloud Computing is a model that can be described as the availability of Cloud Computing resources to mobile environments. From a security point of view, Mobile Cloud Computing introduces many security issues due to the fact that it combines mobile devices with Cloud services.

In this paper were presented the security issues that can jeopardize the Mobile Cloud users' private data or applications. The issues were divided in two types: mobile threats and Cloud threats. For each threats type were presented the security issues that may affect the data, the applications, the device (in the case of mobile threats) and the users' privacy. Also the paper presented an overview of the main Mobile Cloud Computing characteristics. Characteristics used to provide a definition for Mobile Cloud Computing.

ACKNOWLEDGMENT

This paper was supported by the project: Improvement of the doctoral studies quality in engineering science for development of the knowledge based society-QDOC" contract no. POSDRU/107/1.5/S/78534, project co-funded by the European Social Fund through the Sectorial Operational Prog. HR 2007-2013.

REFERENCES

- [1] D. Kovachev, Yiwei Cao and Ralf Klamma. Mobile Cloud Computing: "A Comparison of application Models". In eprint arXiv: 1107.4940, July 2011.
- [2] S.K.Sood, "A combined approach to ensure data security in cloud computing", in S.K. Sood/Journal of Network and Computer Applications Vol.35, pp. 1831–1838, 2012.
- [3] S. Chetan, G. Kumar, K. Dinesh, K. Mathew and M.A. Abhimanyu "Cloud Computing for Mobile World", available online: <http://chetan.ueuo.com/projects/CCMW.pdf>, 2010.
- [4] Mobile Cloud Computing Forum, available online: <http://www.mobilecloudcomputingforum.com>
- [5] White Paper, "Mobile Cloud Computing Solution Brief," AEPONA, November 2010.
- [6] R. D. Caytiles, S. Lee, "Security Considerations for Public Mobile Cloud Computing", International Journal of Advanced Science and Technology Vol. 44, July, 2012.
- [7] A. Khan, K.K. Ahrwar, "Mobile Cloud Computing as a future of mobile multimedia database", International Journal of Computer Science and Communication Vol. 2, No. 1, January-June, pp. 219-221, 2011.
- [8] S.K. Ko, J.H. Lee, S.W. Kim, "Mobile Cloud Computing Security Considerations", Journal of Security Engineering, 2012.
- [9] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches", Accepted in Wireless Communications and Mobile Computing – Wiley, 2011, available online: <http://onlinelibrary.wiley.com/doi/10.1002/wcm.1203/abstract>
- [10] J. H. Christensen, "Using RESTful web-services and cloud computing to create next generation mobile applications," in Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications (OOPSLA), pp. 627-634, October 2009.
- [11] A. N. Khan, M.L. MatKiah, S. U. Khan, S. A. Madani, "Towards secure mobile cloud computing: A survey", Future Generation Computing Systems 2012, doi:10.1016/j.future.2012.08.003.
- [12] H. Qi, A. Gani, "Research on Mobile Cloud Computing: Review, Trend and Perspectives".
- [13] N. Fernando, S. W. Loke, W. Rahayu, "Mobile cloud computing: A survey", Future Generation Computer Systems, Vol. 29, pp. 84–106, 2013.
- [14] C.A. Castillo, "White Paper: Android Malware Past, Present and Future", Mobile Security Working Group, McAfee, available online <http://www.mcafee.com/us/resources/white-papers/wp-android-malware-past-present-future.pdf>, 2011.
- [15] Lookout Mobile Security, Lookout Mobile Threat Report, August 2011.
- [16] C. Nachenberg, "A Window Into Mobile Device Security – Examining the security approaches employed in Apple's iOS and Google's Android", Symantec Security Response, available online: http://investor.symantec.com/files/doc_news/2012/symc_mobile_device_security_june2011.pdf, 2011.
- [17] D. Catteddu, G. Hogben, "Benefits, risks and recommendations for information security", European Network and Information Security Agency, 2009.
- [18] A. Archer "Boehm, Security guidance for critical areas of focus in cloud computing", Cloud Security Alliance, 2009.
- [19] "Top threats to cloud computing", version 1.0, Cloud Security Alliance CSA, available online: <http://www.cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf>, Retrieved March, 2010.
- [20] "The Notorious Nine: Cloud Computing Top Threats in 2013", Cloud Security Alliance CSA, Top Threats Working Group, February 2013.
- [21] L. Subashini, V. Kavitha, "A survey on security issues in service delivery models of cloud computing", Journal of Network and Computer Applications, 2010.
- [22] Open web application security project (OWASP) Top 10, available online: https://www.owasp.org/index.php/Top_10_2010-Main, 2010.
- [23] R. Choubey, R. Dubey, J. Bhattacharjee, "A survey on cloud computing security, challenges and threats", International Journal on Computer Science and Engineering Vol. 3, 2011.
- [24] M. Pastaki Rad, A. Sajedi Badashian, G. Meydanipour, M. Ashurzad Delcheg, M. Alipour, H. Afzali, "A survey of cloud platforms and their future", Computational Science and Its Applications—ICCSA 2009, pp. 788–796, 2009.
- [25] S. Srinivasamurthy, D. Liu, "Survey on cloud computing security", 2010.
- [26] EU Directive 95/46/EC of the European Parliament and the Council of 24 October 1995 on the Protection of Individuals with Regard to the Processing of Personal Data and the Free Movement of Such Data, 1995 O.J. L 281
- [27] Personal Data Protection Act No. 25,326, (Arg.), available online www.privacyinternational.org/countries/argentina/argentine-dpa.html, Oct. 4, 2000.
- [28] International Due Diligence: U.S. vs. European Privacy Laws Kroll an altegrity Company, available online: http://www.kroll.com/media/pdfs/International_Due_Diligence_US_vs_Euro_WP_040811P.pdf
- [29] H. Zhangwei and X. Mingjun, "A Distributed Spatial Cloaking Protocol for Location Privacy," in Proceedings of the 2nd International Conference on Networks Security Wireless Communications and Trusted Computing (NSWCTC), vol. 2, pp. 468, June 2010