



UNSOLVABLE SOLUTIONS

Client: CSIR DSSR

SOFTWARE DESIGN DOCUMENTATION

Eavesdropping Protection in Conclave

Members:

Edwin Fullard
Jaco Bezuidenhoudt
Jandre Coetzee
Maret Stoffberg
Ryno Pierce

Student Number:

12048675
11013878
10693077
11071762
12003922

Contents

1	Introduction	2
1.1	Purpose	2
1.2	Scope	2
1.3	Overview	2
1.4	Definitions and Acronyms	2
2	System Overview	3
2.1	EPIC	3
2.1.1	Eavesdropping Malware Software	3
3	System Architecture	4
3.1	Architecture Design	4
3.2	Decomposition Description	4
3.3	Design Rationale	4
4	Data Design	5
4.1	Data Description	5
4.2	Data Dictionary	5
5	Component Design	6
6	Human Interface Design	7
6.1	Overview of User Interface	7
6.2	Screen Images	7
6.3	Screen Objects and Actions	7
7	Requirements Matrix	8

1 Introduction

1.1 Purpose

The purpose of this Software Design documentation is to introduce and explain to the reader the architecture and designs used in the EPIC project.

1.2 Scope

This EPIC (Eavesdropping Protection in Conclave) project consist of a server, an Android application and a gateway device. The android device is held over the gateway node and, using NFC, a request to enter the meeting is send to the server via the gateway device. The server then respond with permission or denial. If permission is granted, the android device will proceed into protection mode and the meeting log is updated. After the meeting has been held, the device will then be held over the gateway node again to deactivate the protection mode. A user may query the log of a meeting.

The eavesdropping malware use a server and an application on the android device. Ideally the application will be hidden behind another application, but for the sake of this project it will be a visible application. A user will send an eavesdropping request from the server to a specified android device. If this device have the malware installed, it will start streaming the recording to the user. The user may then send a request to stop the recording and store the recording.

1.3 Overview

This document provides a look at the design of the EPIC project, by the Unsolvble Solutions Team.

1.4 Definitions and Acronyms

- **EPIC** Eavesdropping Protection in Conclave

2 System Overview

This system consist of two separate parts: the EPIC components and the Eavesdropping Malware software.

2.1 EPIC

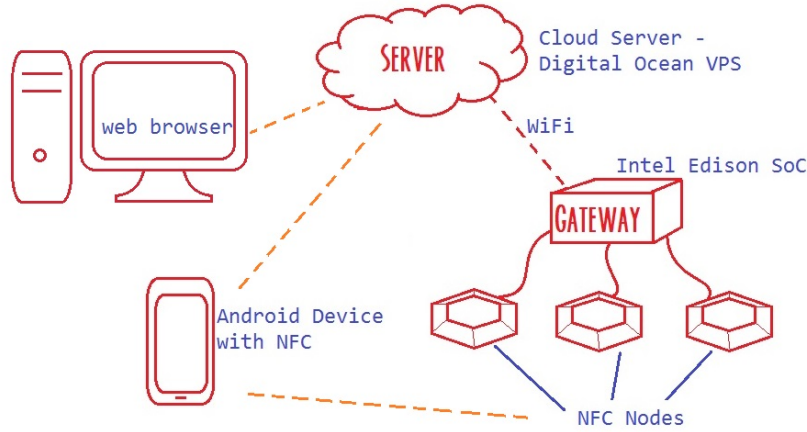


Figure 1: Basic representation of the components

Each user is registered with a unique identifier and password on the cloud server. The user may either register via the web browser or when the application is installed. When a meeting is attended, the user hold his Android NFC device over a NFC Node. The device then sends the identifier to the Node via NFC. The Node then sends the users identifier to the gateway to check if the user may enter the meeting. Each gateway is programmed for a specific meeting, and this meeting may only be attended by specified users. If the gateway give the user permission to enter the meeting, the users device is set on silent and the GSM and the Wifi are turned off. The gateway also sends the users identifier and timestamp of entrace to the server. The server keeps a log of all the entries to a meeting. This log may later be viewed via a web browser. The scheduling, updating and deletion of meetings are also done via a browser. After a person is done with the meeting, they scan their device over the NFC Node again, this time the phone is restored to its previous state.

2.1.1 Eavesdropping Malware Software

3 System Architecture

3.1 Architecture Design

- **Model-View-Controller** This architectural design pattern is used with the server.
- **Singleton** The singleton design pattern is used with the Malware. Only one instance of a recoding may happen at a time.
- **Singleton** The gateway make use of the Singleton design pattern. There may only be one meeting per gateway.

3.2 Decomposition Description

3.3 Design Rationale

4 Data Design

4.1 Data Description

All the user and meeting data are stored in a MongoDB data structure.

4.2 Data Dictionary

5 Component Design

6 Human Interface Design

6.1 Overview of User Interface

6.2 Screen Images

6.3 Screen Objects and Actions

7 Requirements Matrix