# Initial Design Report

## Table of Contents

1 Introduction .................................................................................................................. 1  
   1.1 Project Definition and Goals .............................................................................. 1  
   1.2 Purpose of Document .......................................................................................... 2  
2 Design Constraints ..................................................................................................... 2  
   2.1 Resource Constraints .......................................................................................... 2  
   2.2 Time Constraints ................................................................................................ 2  
   2.3 Ergonomic Constraints ....................................................................................... 3  
   2.4 Experience and Skills of Group Members Constraints ....................................... 3  
   2.5 Software Constraints ........................................................................................ 3  
   2.6 Quality of Service Constraints ......................................................................... 4  
3 Detailed Design ........................................................................................................ 6  
   3.1 Use Case Diagrams ............................................................................................. 6  
      3.1.1 Caller Use Case Diagrams ........................................................................... 6  
      3.1.2 Callee Use Case Diagrams ......................................................................... 6  
   3.2 Activity Diagrams ............................................................................................... 7  
      3.2.1. Calling Activity .......................................................................................... 7  
      3.2.2. Getting Called Activity ............................................................................ 8  
      3.2.3. Chatting Activity ...................................................................................... 9  
   3.3 Class Diagram ..................................................................................................... 11  
4 Architectural Design .................................................................................................. 12  
   4.1. Security Model .................................................................................................. 12  
      4.1.1. Server side applications .......................................................................... 12  
      4.1.2. Asterisk side applications ....................................................................... 13  
      4.1.3. Board side applications .......................................................................... 14  
      4.1.4 VoIP threat assessment model .................................................................... 17  
   4.2 Network Module .................................................................................................. 19  
      4.2.1 SIP (Session Initiation Protocol) ................................................................. 19  
      4.2.2 RTP (Real-Time Transport Protocol) ......................................................... 22  
      4.2.3 UDP (User Datagram Protocol) ................................................................. 23  
5 Hardware Modules .................................................................................................... 25  
6 Gantt chart ................................................................................................................. 26  
7 References .................................................................................................................. 27
1 Introduction

1.1 Project Definition and Goals

The project aims to design an embedded handset which provides end-to-end secure voice over internet protocol (VoIP) communication. The main concern is to achieve the prototype of the handset. It is another issue for us to finalize the product for home or office use. This seems a design issue of the handset itself. But of course to end up with an embedded handset which is connected to a microphone and a speaker and offering telephony experience, audio transmission in technically speaking, in secure is crucial.

We will be working on a FPGA Embedded board. A microphone and a speaker is going to be connected to the board. This board will be connected to a server, which has Asterisk PBX, via Ethernet and the server machine will be connected to the internet or another network. The conversations will be carried on to end points like computers, telephones or cell phones. Whether the channel which carries the conversation is secure or not, the data should be protected by encryption. Thus no man-in-the-middle could follow the conversation.
1.2 Purpose of Document

The purpose of this report is to show our initial design concepts about project. In this report, we gave details of the project made some additions according to requirements explained in the requirement analysis report. This report implies initial design of the following parts of the project.

- Software components (Encryption, VoIP, SIP, Network)
- Hardware components (FPGA Embedded Board)

2 Design Constraints

2.1 Resource Constraints

We need datasheets of the board that we’ll setup embedded Linux on and manual of the software development environment that we will use for coding. We expect these documents to be supplied by our teaching assistant and also we will use the resources online whenever we need extra information. Also guides for using Asterisk in an efficient way is compulsory but there are enough e-books on internet for this subject.

Sometimes it may be difficult to find the resources that are really useful for us, because most of the board-related things are owned by companies and can not be seen without paying. This limits our development for now a little.

2.2 Time Constraints

At the beginning of this semester, project named “Secure Channel over VoIP” is given to our group. It is surprising that, only our group has taken second project choice. The deadline of our project is June and also we should provide a prototype at the end of this semester. Therefore, especially for an embedded project, this is the most important constraint. To use our time efficiently we must follow our schedule strictly.
We have started this project after the meeting with Atilla Özgit from Invicta which is the sponsor company of this project. This meeting is the first milestone of our project. For the first semester we are supposed to accord some reports’ deadlines which are Project Proposal, Requirement Analysis Report, Initial Design Report and Detailed Design Report.

2.3 Ergonomic Constraints
Since we will be using a simple board computer for embedded systems we may have some problems with integrating it with the linux installed pc’s.

2.4 Experience and Skills of Group Members Constraints
All of the group members do not have enough knowledge about project management. None of us has taken a role in a big project. Thus, especially documentation of project takes lots of time.

We have decided to program our board with object-oriented programming language C++. We have not designed a big project with C++ so far. But we have used C for programming our embedded device, PIC, on CENG336 course.

While analyzing the source codes of the example VoIP projects, we have encountered some difficulties. Understanding someone’s source code may be more difficult than coding a new program. However, in order to understand the important parts of VoIP technologies, using open source Asterisk properties totally, we should learn how to analyze the source codes.

2.5 Software Constraints
We need some software and tools for designing our project which will help us at drawing, documentation and implementation parts of this project.

Some software requirements;
- Embedded C++ Compiler (may be Green Hills C++ or Intel C++) or Java Compiler (Janino or GCJ)
Initial Design Report

- Microsoft Visio (for Windows) or Dia (for Linux)
- Microsoft Office (for Windows) or Open Office (for Linux)
- Asteriks
- Embedded Linux Kernel 2.6
- Adobe Reader
- U-Boot

2.6 Quality of Service Constraints

In the field of telephony, quality of service was defined in the ITU standard X.902 as "A set of quality requirements on the collective behavior of one or more objects". Quality of Service comprises requirements on all the aspects of a connection, such as service response time, loss, signal-to-noise ratio, cross-talk, echo, interrupts, frequency response, loudness levels, and so on.

Quality of Service is a major issue in VOIP implementations. The issue is how to guarantee that packet traffic for a voice or other media connection will not be delayed or dropped due interference from other lower priority traffic.

QoS is sometimes used as a quality measure, with many alternative definitions, rather than referring to the ability to reserve resources. Quality of service sometimes refers to the level of quality of service, i.e. the guaranteed service quality. High QoS is often confused with a high level of performance or achieved service quality, for example high bit rate, low latency and low bit error probability.
Initial Design Report

The basic architecture introduces the three fundamental pieces for QoS implementation:

- QoS identification and marking techniques for coordinating QoS from end to end between network elements
- QoS within a single network element (for example, queuing, scheduling, and traffic shaping tools)
- QoS policy, management, and accounting functions to control and administer end-to-end traffic across a network

<table>
<thead>
<tr>
<th>Priority Level</th>
<th>Traffic Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (lowest)</td>
<td>Best Effort</td>
</tr>
<tr>
<td>1</td>
<td>Background</td>
</tr>
<tr>
<td>2</td>
<td>Standard (Spare)</td>
</tr>
<tr>
<td>3</td>
<td>Excellent Load (Business Critical)</td>
</tr>
<tr>
<td>4</td>
<td>Controlled Load (Streaming Multimedia)</td>
</tr>
<tr>
<td>5</td>
<td>Voice and Video (Interactive Media and Voice)</td>
</tr>
<tr>
<td>6</td>
<td>Layer 3 Network Control Reserved Traffic</td>
</tr>
<tr>
<td>7 (highest)</td>
<td>Layer 2 Network Control Reserved Traffic</td>
</tr>
</tbody>
</table>
3 Detailed Design

3.1 Use Case Diagrams

3.1.1 Caller Use Case Diagrams
This diagram explains what a caller may do using the VoIP to call someone.

**Dial:** The caller may dial the number easily, so that the server routs the number to the IP directly.

**Hung up:** User may directly end the call when he/she does not want to talk anymore.

3.1.2 Callee Use Case Diagrams
This diagram explains what a user may do using the VoIP when he/she get a call.

**Answer:** Callee may answer the phone so that call time starts.

**Busy:** While another call is started during the callee is doing a call, phone gives busy tone the caller. Asterisk has several methods like saving the call, leaving message and so on.

**Hung up:** User may directly end the call when he/she does not want to answer or wants stop the talk.

**No answer:** Other than hung up, callee may choose not to do anything so phone rings until the caller gives up trying.
3.2 Activity Diagrams

To show the operational workflow step-by-step clearly the activity diagrams are divided to three parts. These diagrams describe three separate processes of VoIP. Activities are “Calling Activity”, “Getting Called Activity” and “Chatting Activity”. Since the processes are distinct and separable, classifying the activities makes diagrams easier to read and understand.

3.2.1. Calling Activity

For successful calls, calling activity diagram shows the process from dialing to the moment that the communication of sender and receiver had accomplished. For unsuccessful calls, it shows the process until the “error” command.
3.2.2. Getting Called Activity

This diagram takes the process from the first signal that has come from the sender until the communication of both sides start.
3.2.3. Chatting Activity

Chatting activity’s initial point is the moment that media streaming starts. It is a two sided process so it should be shown with swimlane model to group the activities in a single thread. Chatting activity lasts until the communication stops.
3.3 Class Diagram

```
SecurityOptions
- publicKey : object::PublicKey
- privateKey : object::PrivateKey
- encryptionMethod : string
- dataStream : object::Data
+getPublicKey() : object::PublicKey
+getEncryptionMethod() : string
+setEncryptionMethod(in encryptionMethod : string)
+encryptDataStream(in data : object::Data)
+decryptDataStream(in data : object::Data)
```

```
Person
- name : string
- id : int
- ip : string
+getName() : string
+getPersonId() : int
+setPersonIp(in ip : string)
```

```
Connection
- io : object::SoundCard
- callee : object::Person
- security : object::SecurityOptions
- compressionMethod : string
- willBeSent : object::Data
- received : object::Data
+setPerson(in person : object::Person)
+setPublicKey(in publicKey : object::PublicKey)
+setCompressionMethod(in compressionMethod : string)
+sendInitialConnectionProperties(in publicKey : object::PublicKey)
+sendEncryptedData(in data : string)
+receiveInitialConnectionProperties() : object::PublicKey
```

```
Data
- data : string
- compressionMethod : string
- isEncrypted : bool
- isCompressed : bool
- security : SecurityOptions
+getData() : string
+encryptDataStream()
+decryptDataStream()
+compressDataStream()
+decompressDataStream()
```

```
SoundCard
- input : Data
- output : Data
+getDataFromCard()
+outputDataStream()
```
4 Architectural Design

4.1 Security Model

With the continuing pressure to reduce fixed costs within business, enterprises and small- and medium-sized businesses (SMBs) are looking at Voice over IP (VoIP) as an opportunity for cost savings.

Soft clients, powerful multi-function handheld devices, IP-enabled wireless networks within an enterprise, SIP-enabled handsets, and IP PBXs are becoming more pervasive in enterprise networks. Network managers are being asked to implement these new networks to provide top-quality services, without compromising network integrity. But with the introduction of any new IP device into the local network, there are security vulnerabilities that organizations must not only be aware of, but well prepared for.

In our SIP using, IPX based (Asterisk) VoIP system our main aim is to offer a secure communication between the users. In the security model of our software, there are several steps for maintaining the communication secure and pretending the possible threats. We may divide these processes to Asterisk side, board side, server side applications.

4.1.1 Server side applications

As VoIP is always considered an application on top of the operating system, and not really a part of the operating system, hackers are more likely to go after VoIP on the application level than from a pure network level.

I would say that the main area of attack will be the VoIP protocol itself, trying to inject malformed/malicious messages for one purpose or another; or trying to hack into
the application to gain access to user related information. No alien network packet has any right to be on our internal network without being molested and thoroughly strip searched beforehand.

Defense should start before data reaches the server. This means a DMZ in some form with an outer firewall. There is no point in allowing spurious data hitting the PBX, so they may as well be stopped here.

We’ll configure Linux IPTables Firewall for Asterisk to get rid of outer server threats as follows.

```
# SIP on UDP port 5060. Other SIP servers may need TCP port 5060 as well
iptables -A INPUT -p udp -m udp --dport 5060 -j ACCEPT

# IAX2 - the IAX protocol
iptables -A INPUT -p udp -m udp --dport 4569 -j ACCEPT

# IAX - most have switched to IAX v2, or ought to
iptables -A INPUT -p udp -m udp --dport 5036 -j ACCEPT

# RTP - the media stream
iptables -A INPUT -p udp -m udp --dport 10000:20000 -j ACCEPT

# MGCP - if you use media gateway control protocol in your configuration
iptables -A INPUT -p udp -m udp --dport 2727 -j ACCEPT
```

4.1.2. Asterisk side applications

Asterisk is software running a PBX system that is freely available to anyone under GNU General Public License (GPL). Asterisk can be installed and run on a variety of different operating system (OS) platforms, such as Linux, BSD and Mac OS X. It has its
own security settings, by improving some of these settings and using them well, VoIP communication may be safer.

Not accepting SIP authentication requests from all IP addresses. Even if we accept inbound calls, we will not let those users reach authenticated elements.

Setting “alwaysauthreject=no” in sip.conf file prevents extension information leakage. By setting it “no”, Asterisk reject bad authentication requests on valid usernames with the same rejection information as with invalid usernames, denying remote attackers the ability to detect existing extensions with brute-force guessing attacks.

Using “STRONG” passwords for SIP entities is probably the most important step. Using symbols, numbers and a mix of upper and lowercase letters at least 12 digits long is secure.

Blocking AMI manager ports, allowing only one or two calls at a time per SIP entity and making SIP usernames different than the extensions help as well.

4.1.3. Board side applications

This part is mostly about cryptography. We’ll use a relatively new cryptographic approach, “public key cryptography”. The Public Key Infrastructure (PKI) is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates. There are several solutions (EJBCA, Entrust, Linux etc.) for deploying PKI.
Public-key encryption (also called asymmetric encryption) involves a pair of keys---a public key and a private key---associated with an entity that needs to authenticate its identity electronically or to sign or encrypt data. Each public key is published, and the corresponding private key is kept secret.

Compared with symmetric-key encryption, public-key encryption requires more computation and is therefore not always appropriate for large amounts of data. However, it is possible to use public-key encryption to send a symmetric key, which can then be used to encrypt additional data. This is the approach used by the SSL protocol.
Initial Design Report

As an algorithm, we are planning to use RSA for public-key cryptography. It is an algorithm which is used for two basic steps of this cryptography; generating the pairs and encrypting-decrypting the data.

**RSA setting**: to generate a public/private key pair.

1. Bob generates two large primes, \( p \) and \( q \).
2. He computes \( n = pq \) and \( \phi(n) = (p-1)(q-1) \). \( n \) will be used as the modulus.
3. He chooses a random \( e \) \((0 < e < \phi(n))\) such that \( e \) is relatively prime to \( \phi(n) \). \( e \) used as the public exponent.
4. He computes \( d \) as the inverse of \( e \) modulo \( \phi(n) \) (i.e., \( d \) such that \( ed \equiv 1 \ (\text{mod} \ \phi(n)) \)).
   \( d \) will be used as the private exponent.

He publishes \((n,e)\) as his public key and keeps \((n,d)\) as his private key. The primes \( p \) and \( q \) must be kept secret or destroyed.

**RSA algorithm**: to encrypt and decrypt data.

INSTANCE: group \( \mathbb{Z}_n \) and the set \((n,e,d)\): \( n = pq \), \( p \) and \( q \) primes. \( ed \equiv 1 \ (\text{mod} \ \phi(n)) \).

ASSUMPTION: Alice knows Bob's public key \((n,e)\), but does not know Bob's private key \((n,d)\).

ALGORITHM:

1. Alp encrypts message \( m \) by computing \( c = m^e \mod n \).
2. Alp sends $c$ to Murat.
3. Upon receipt, Murat decrypts $c$ by computing $c^d \mod n$ and gets back $m$.

4.1.4 VoIP threat assessment model

Starting from the basic OSI Reference Model and the Department of Defense (DoD) or TCP/IP reference model, the SIP-based VoIP network can be analyzed by a layered approach. Threats and therefore countermeasures can also be mapped to the layers of the network reference models. With this layered analysis strategy, it becomes immediately apparent that each layer has different security threats.
A defense strategy can also follow this layered approach. This eases deployment and leads to the three-layer security model as follows:

- **Infrastructure security layer:** Protect and secure the network infrastructure
- **Network services security layer:** Protect and secure end-users, access and service enablers
- **Application security layer:** Protect and secure SIP-based VoIP and other network applications

Based on general network security precepts, each security layer then needs to be evaluated on the basis of the following parameters:

- **Authentication:** Confirm the identity of communicating entities, whether individuals, devices, services or applications. Authentication guards against impersonation or replay of previous communications.
- **Authorization:** Cross-checks identity for role and access. This prevents unauthorized access to services, access to stored information, toll fraud, etc.
- **Accountability/Audit:** Keeps track of usage and security services. This helps in early detection and recovery from threats and attacks.
- **Availability/Reliability:** Redundancy, perimeter protection and hardening ensure that authorized users continue to have access to network devices, services and stored information despite an ongoing attack such as a DoS attack.
- **Confidentiality:** Encryption of communication streams prevents unauthorized intercepts and eavesdropping. In addition, encryption can be coupled with access control to protect stored information.
- **Integrity:** Prevents unauthorized modifications, deletion, creation or replication of data. Typical mechanisms are based on hashing algorithms such as HMAC, MD-5 and SHA-1. This also helps in early detection of unauthorized activity.
Initial Design Report

- **Non-repudiation**: Proof that communications actually happened. Required for forensic evidence purposes.

- **Privacy/Anonymity**: Privacy tackles issues like phone number harvesting, call pattern tracking, etc. that violates the privacy of the user. Anonymity, on the other hand, allows a user to communicate without revealing their identity and is usually contrary to most security policies.

4.2 Network Module

This project mainly concerns on sending and receiving voice data over a network. This network may a Wide Area Network, Local Area Network or the Internet, etc. Compared to securing the data stream, Network issues play a critical role in the scenario too. Therefore we are going to give some brief technical and popular information about use of protocols in the market and challenges and details of them in the following sections of Network Module.

4.2.1 SIP (Session Initiation Protocol)

A SIP (Session Initiation Protocol) connection is a service offered by many ITSP (Internet Telephony Service Providers) that connects a company's PBX to the existing telephone system infrastructure (PSTN) via Internet using the SIP VoIP standard.

Using a SIP connection may simplify administration for the organization as the SIP connection typically will use the same Internet connection that is used for normal data.

This removes the need to also have a BRI/PRI installed as well, although sharing the same bearer circuit for calls and data raises its own challenges in maintaining call quality.

If the call traffic runs on the same connection with other traffic like Email or Web, voice and even signaling packets may be dropped and the voice stream can get interrupted.
To mitigate this, many companies split voice and data up into two separate internet connections to solve this problem, so that the resource conflict on the Internet access side is avoided. Other devices perform traffic shaping in order to avoid this resource conflict, but they still depend on the merit of the service provider not to drop packets from the Internet to the PBX.

Registration is required if the end user has a dynamic IP address, if the provider does not support static hostnames, or if NAT is used. In order to share several DID numbers on the same registration, the IETF has defined additional headers (for example "P-Preferred-Identity", see RFC 3325). This avoids multiple registrations from one PBX to the same provider. Using this method the PBX can indicate what identity should be presented to the Called party and what identity should be used for authenticating the call. This feature is also useful when the PBX redirects an incoming call to a PSTN number, for example a cell phone, to preserve the original Caller ID.
The increasing concerns about security of calls that run over the public Internet has made SIP encryption more popular. Because VPN is not an option for most service providers, most service providers that offer secure SIP connections use TLS and SRTP for encrypting the traffic. The keys for SRTP are exchanged using RFC 4568 (SDES).

Users should also be aware that a SIP connection can be used as a channel for attacking the company's internal networks, similar to Web and Email attacks. Users should consider installing appropriate security mechanisms to prevent malicious attacks.
4.2.2 RTP (Real-Time Transport Protocol)

The Real-time Transport Protocol (RTP) defines a standardized packet format for delivering audio and video over the Internet. It was developed by the Audio-Video Transport Working Group of the IETF and first published in 1996 as RFC 1889, and superseded by RFC 3550 in 2003.

RTP is used extensively in communication and entertainment systems that involve streaming media, such as telephony, video teleconference applications and web-based push to talk features. For these it carries media streams controlled by H.323, MGCP, Megaco, SCCP, or Session Initiation Protocol (SIP) signaling protocols, making it one of the technical foundations of the Voice over IP industry.

RTP is usually used in conjunction with the RTP Control Protocol (RTCP). While RTP carries the media streams (e.g., audio and video) or out-of-band events signaling (DTMF in separate payload type), RTCP is used to monitor transmission statistics and quality of service (QoS) information. When both protocols are used in conjunction, RTP is usually originated and received on even port numbers, whereas RTCP uses the next higher odd port number.
The User Datagram Protocol (UDP) is one of the core members of the Internet Protocol Suite, the set of network protocols used for the Internet. With UDP, computer applications can send messages, in this case referred to as datagrams, to other hosts on an Internet Protocol (IP) network without requiring prior communications to set up special transmission channels or data paths. UDP is sometimes called the Universal Datagram Protocol. The protocol was designed by David P. Reed in 1980 and formally defined in RFC 768.[1]

UDP uses a simple transmission model without implicit hand-shaking dialogues for guaranteeing reliability, ordering, or data integrity. Thus, UDP provides an unreliable service and datagram may arrive out of order, appear duplicated, or go missing without
notice. UDP assumes that error checking and correction is either not necessary or performed in the application, avoiding the overhead of such processing at the network interface level. Time-sensitive applications often use UDP because dropping packets is preferable to waiting for delayed packets, which may not be an option in a real-time system. If error correction facilities are needed at the network interface level, an application may use the Transmission Control Protocol (TCP) or Stream Control Transmission Protocol (SCTP) which are designed for this purpose.

Network module will work as to send the encrypted voice data to the server and internet. We will use RTP protocol and SIP in the application layer, udp protocol in the transport layer.

Protocol standards are important to synchronize the transmitted data.
5 Hardware Modules

In our project, we are going to use a board which is connected to the server. We will connect a microphone and a speaker to this board and use it as a voip telephone. This FPGA board with its cables and connectors and programs will be provided by INVICTA as a FPGA Development Kit. In few days we are planning to take the board from Mr. Özgit. Our board’s requirements are;

- It has to have its own CPU and memory so that we may install Embedded Linux Kernel on it.
- It needs at least one Ethernet port for the connection between the server.
- At least one analog port for the connection of microphone. Analog-digital conversions will be made by using this port.
- We need basic microphone and speaker for communication.

For these requirements Atmel's AT18F Low Cost configuration Family can be used to configure low Cost SRAM FPGAs as well as higher density high performance FPGAs from Xilinx. The AT18F Family is offered in Densities of 1 Mb up to 8 Mb.
6 Gantt chart
7 References

1. “Cryptography Theory and Practice” R. Stinson (PKI Encryption)
2. SIP (Session Initiation Protocol) en.wikipedia.org
3. “VoIP and Skype Security” Simson L. Garfinkel
4. “RSA Encryption” Tom Davis
5. “Windows XP Quality of Service (QoS) enhancements and behavior” http://support.microsoft.com
6. “QoS (Quality of Service)” en.wikipedia.org
8. Asterisk Source Documentation and Configuration www.asterisk.org