Fall Design Report

Skullduggery

Greg Baker - Robert Koeninger

Advisor: Prof. Wilsey

Dec 4, 2009

Table of Contents

Encrypted Cell Phone Communication

Background:

Sensitive information is frequently transmitted over cellular communication networks as voice data. As voice recognition software improves, it will become more important for information such as social security numbers and credit card information to stay private between the parties. Encrypting voice communication more securely will provide more protection against eavesdroppers looking for information.

Problem Statement:

The current standard of secure communication, GSM, has been proven to be insecure. Current cellular phone systems do not provide sufficient privacy for sensitive information to be confidently transmitted over cellular networks. Our cellular encryption system will provide additional security for sensitive information to be transmitted during a call, while preventing an eavesdropper from deciphering the communication.

Team Members:

Greg Baker **bakergo@mail.uc.edu**

Robert Koeninger **koeninrc@mail.uc.edu**

Faculty Advisor:

Dr. Philip A. Wilsey

Goal:

Our goal is to modify a smart phone to encrypt voice packets before being sent, and decrypt them on the other end of the voice communication. If the receiving phone is incapable of decryption, the paranoid party will be alerted that the communication is insecure.

Subgoals:

Explore encryption techniques that are feasible for use in cellular communication.

1. Develop software on the cellular phone to encrypt and decrypt voice communication during a call.
2. Analyze the power and speed of various techniques to this goal.

Helpful Skills:

- Knowledge of encryption methods
- Some familiarity with host system (Google Android)
- Java and embedded programming skills

Task List

Setup Source Control – November 8 – November 10
Source control for project documentation and source code should be set up ASAP

Setup Development Environment – November 8 – November 10
Development environments for relevant SDKs should be set up before we can investigate how to use them and what capabilities they have.

Research Phone SDK – November 13 – November 16
We need to research the SDK of the possible phones that we will use. Candidates are the iPhone, Android SDK, and Windows Mobile platforms.

Write User Requirements – November 8 – November 16
We need to write user requirements between November 8 and November 16. This is partly due to class assignments but also for direction during the project.

Write Use Cases/ Activity Diagrams - November 8 – November 20
Use Cases and Activity Diagrams will be written outlining what we need to do to have a successful project at the end of the year. These will be written after the User requirements have been hammered out.

Write Design Specification – November 20 – November 30
A formal design specification should be written after we write user requirements and activity diagrams. We will write a more formal design specification to show what progress we have and what we need to get done.

Research Encryption Methods – November 23 – December 1
We should know about which encryption methods we want to use before we start implementing the

Research Phone Communication Process – November 23 – December 1
We should know about how the cellular phones communicate before we can modify how they communicate.

Investigate Phone OS Source Code – December 13 – January 6
Knowing the specifics of how to use phone communication methods is vital to the success of the project.

Find/Develop Encryption Library – December 13 – January 6
Once we have selected an encryption method and know what aspects of the phone we're messing with, we can find or develop an encryption library that meets our requirements.

Write Principal Code – January 3 – March 7
The majority of the code will be written between January 3 and March 7.

Design Complete – January 18
This milestone shows the end of most of our design activities.

Acquire Phones – March 8
This milestone shows the latest date that we should have our cellular phones by.

Bug Fixing – March 8 – April 5
Bug fixing will take place after the majority of the code is written.

Code Writing Completed – March 8
The majority of the code should be finalized by March 8 in order to catch all the small bugs that might plague us during presentations.

Write Test Plan – Feb 14 – Feb 28
A more formal testing plan should be written before we test in emulated software.

Test in Emulation – Feb 28 – March 15
We should test in an emulated environment before the end of coding, and before we acquire phones and test on hardware.

Test on Hardware – March 3 – April 5
Hardware testing will take a long time, as it may be annoying and difficult to use.

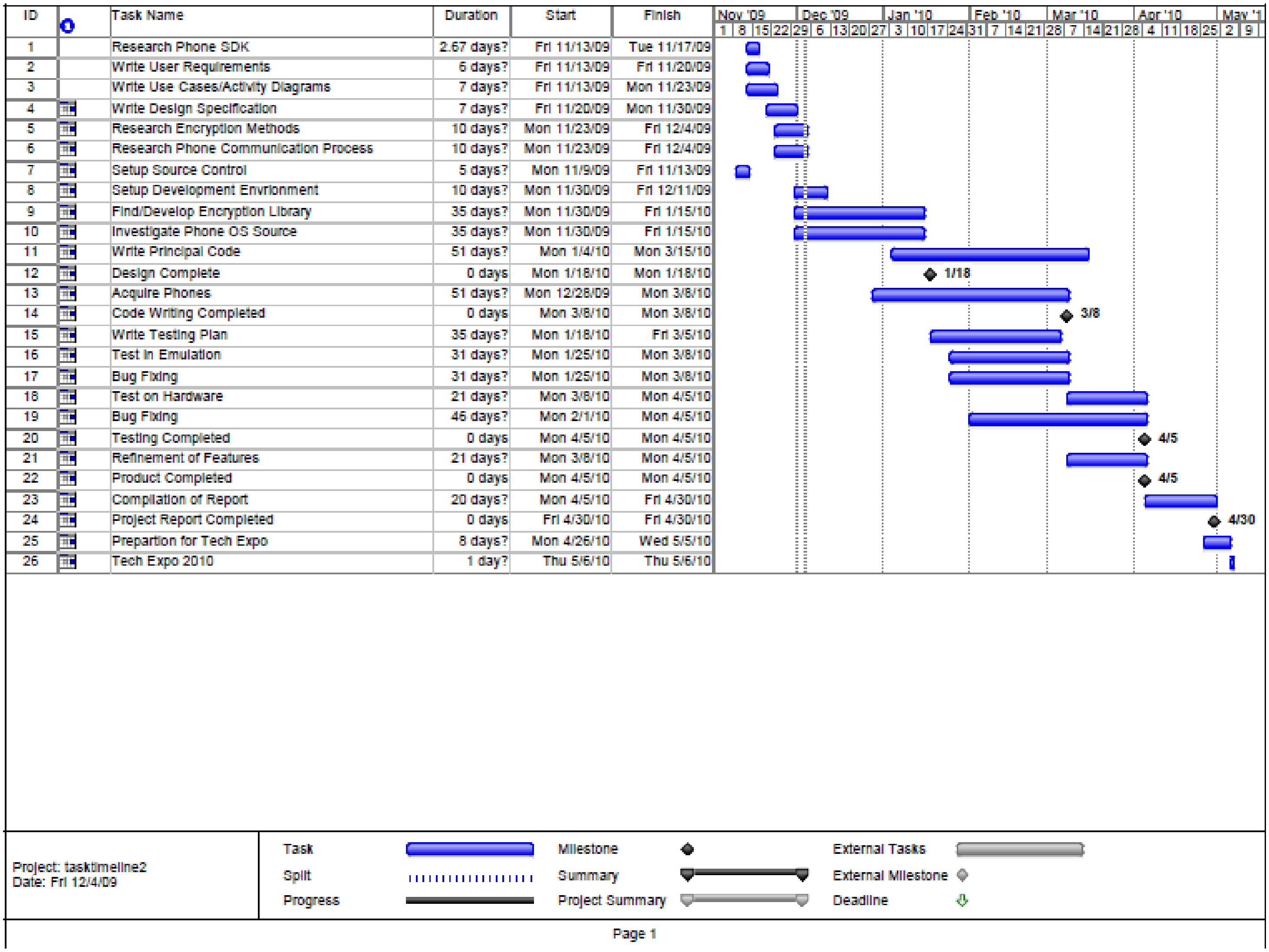
Testing Completed – April 5
This milestone shows when we should have completed testing. It is ok to miss this deadline, but the bugs should start becoming features afterwards.

Product Completed – April 5
This milestone shows when we should be completed in order to have a “buffer” and to complete other assignments and properly complete the project report.

Project Report Completed – April 30
The final project report should be done by the end of April. If it isn't we will have time issues.

Prepare for Tech Expo – May 1 – May 5
We are preparing for the tech expo up to a week in advance. This may be extended into April.

Tech Expo – May 6 - May 7
Present our project at the Tech Expo. These dates are estimates as we could not find them before.

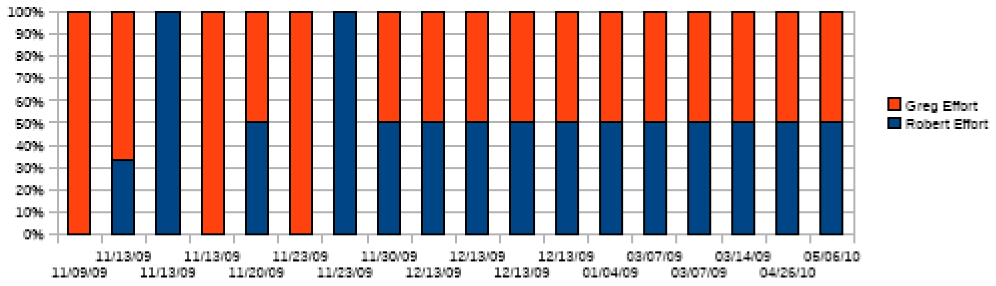


ID	ⓘ	Task Name	Duration	Start	Finish
1		Research Phone SDK	2.67 days?	Fri 11/13/09	Tue 11/17/09
2		Write User Requirements	6 days?	Fri 11/13/09	Fri 11/20/09
3		Write Use Cases/Activity Diagrams	7 days?	Fri 11/13/09	Mon 11/23/09
4	▦	Write Design Specification	7 days?	Fri 11/20/09	Mon 11/30/09
5	▦	Research Encryption Methods	10 days?	Mon 11/23/09	Fri 12/4/09
6	▦	Research Phone Communication Process	10 days?	Mon 11/23/09	Fri 12/4/09
7	▦	Setup Source Control	5 days?	Mon 11/9/09	Fri 11/13/09
8	▦	Setup Development Envrionment	10 days?	Mon 11/30/09	Fri 12/11/09
9	▦	Find/Develop Encryption Library	35 days?	Mon 11/30/09	Fri 1/15/10
10	▦	Investigate Phone OS Source	35 days?	Mon 11/30/09	Fri 1/15/10
11	▦	Write Principal Code	51 days?	Mon 1/4/10	Mon 3/15/10
12	▦	Design Complete	0 days	Mon 1/18/10	Mon 1/18/10
13	▦	Acquire Phones	51 days?	Mon 12/28/09	Mon 3/8/10
14	▦	Code Writing Completed	0 days	Mon 3/8/10	Mon 3/8/10
15	▦	Write Testing Plan	35 days?	Mon 1/18/10	Fri 3/5/10
16	▦	Test in Emulation	31 days?	Mon 1/25/10	Mon 3/8/10
17	▦	Bug Fixing	31 days?	Mon 1/25/10	Mon 3/8/10
18	▦	Test on Hardware	21 days?	Mon 3/8/10	Mon 4/5/10
19	▦	Bug Fixing	46 days?	Mon 2/1/10	Mon 4/5/10
20	▦	Testing Completed	0 days	Mon 4/5/10	Mon 4/5/10
21	▦	Refinement of Features	21 days?	Mon 3/8/10	Mon 4/5/10
22	▦	Product Completed	0 days	Mon 4/5/10	Mon 4/5/10
23	▦	Compilation of Report	20 days?	Mon 4/5/10	Fri 4/30/10
24	▦	Project Report Completed	0 days	Fri 4/30/10	Fri 4/30/10
25	▦	Prepartion for Tech Expo	8 days?	Mon 4/26/10	Wed 5/5/10
26	▦	Tech Expo 2010	1 day?	Thu 5/6/10	Thu 5/6/10

Project: tasktimeline2
Date: Fri 12/4/09

Task	Start Date	Greg Hours	Robert Hours	Total Hours	Greg Effort	Robert Effort
Setup Source Control	11/09/09	1.0		1.0	100%	0%
Research Phone SDK	11/13/09	8.0	4.0	12.0	67%	33%
Write Use Cases/Activity Diagrams	11/13/09		4.0	4.0	0%	100%
Write User Requirements	11/13/09	4.0		4.0	100%	0%
Write Design Specification	11/20/09	1.0	1.0	2.0	50%	50%
Research Encryption Methods	11/23/09	4.0		4.0	100%	0%
Research Phone Communication Process	11/23/09		4.0	4.0	0%	100%
Setup Development Environment	11/30/09	2.0	2.0	4.0	50%	50%
Acquire Phones	12/13/09	3.0	3.0	6.0	50%	50%
Find/Develop Encryption Library	12/13/09	3.0	3.0	6.0	50%	50%
Investigate Phone OS Source	12/13/09	6.0	6.0	12.0	50%	50%
Write Testing Plan	12/13/09	5.0	5.0	10.0	50%	50%
Write Principal Code	01/04/09	40.0	40.0	80.0	50%	50%
Bug Fixing	03/07/09	40.0	40.0	80.0	50%	50%
Test in Emulation	03/07/09	10.0	10.0	20.0	50%	50%
Test on Hardware	03/14/09	10.0	10.0	20.0	50%	50%
Preparation for Tech Expo	04/26/10	16	16	32	50%	50%
Tech Expo 2010	05/06/10	8	8	16	50%	50%

Effort vs. Time
By week date



Design Diagrams

Level 0 shows the simple branch where the phone determines if the call is encrypted or not. If the call is encrypted, the diagram branches to the non-descript task of communicating with encryption.

Level 1 is expanded to show the encrypted/unencrypted conversion branch on both the sending and the receiving phone. Also included is a loop to illustrate the encryption process that occurs on both phones while the conversion is taking place.

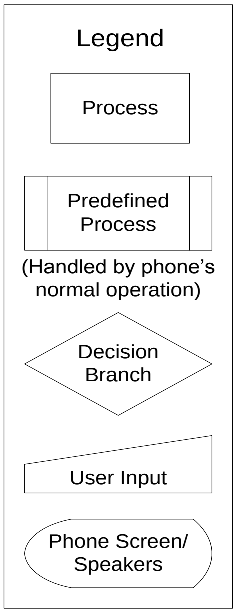
Level 2 elaborates on the encryption key exchange process and the conversation encryption process. This diagram also mentions the on-screen indicator to the user warning them that the call will not be encrypted. The names of the processes in the "Both Phones" section describe the conversation process in more detail, stating each step between when the voice data is picked up by the microphone and sent to the antenna.

Conventions Used in Diagrams

In Design Level 1 and Level 2 Diagrams, the phone operations are divided into three groups: "Sending Phone", "Receiving Phone" and "Both Phones" in order to separate activities that are unique to sender or receiver. The "Both Phones" section contains the process that encrypts the conversation as this is the same for both phones once the conversation has been established.

Several processes are enclosed in Predefined Process nodes in order to keep the phone's normal operation out of the diagram. The "Continue Call Normally" process, for instance, is there as a place holder for the normal operation of a phone call, without leaving that step in the overall sequence vague.

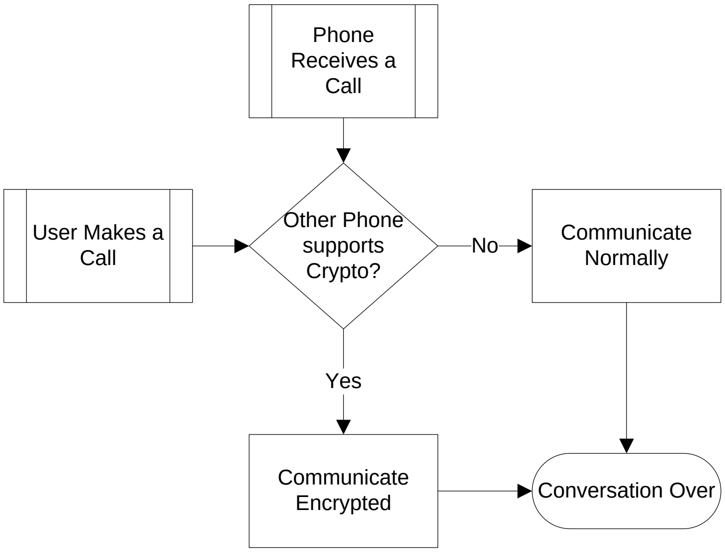
(Refer to the legend for the meaning of node shapes)



Allow for encrypted
voice conversation
between two enabled
smart phones

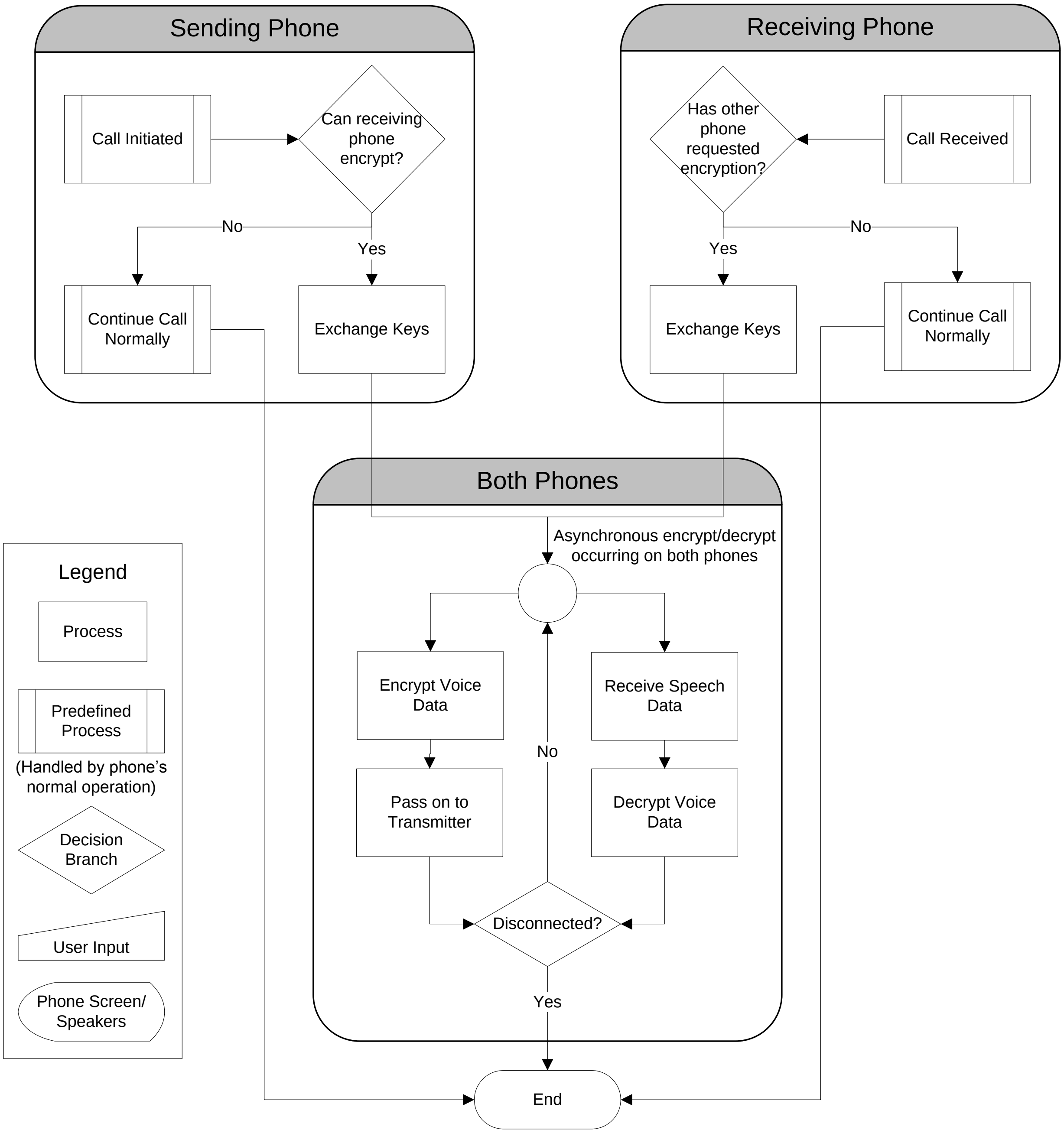
Encrypted Cellphone Communication

Level 0 Diagram



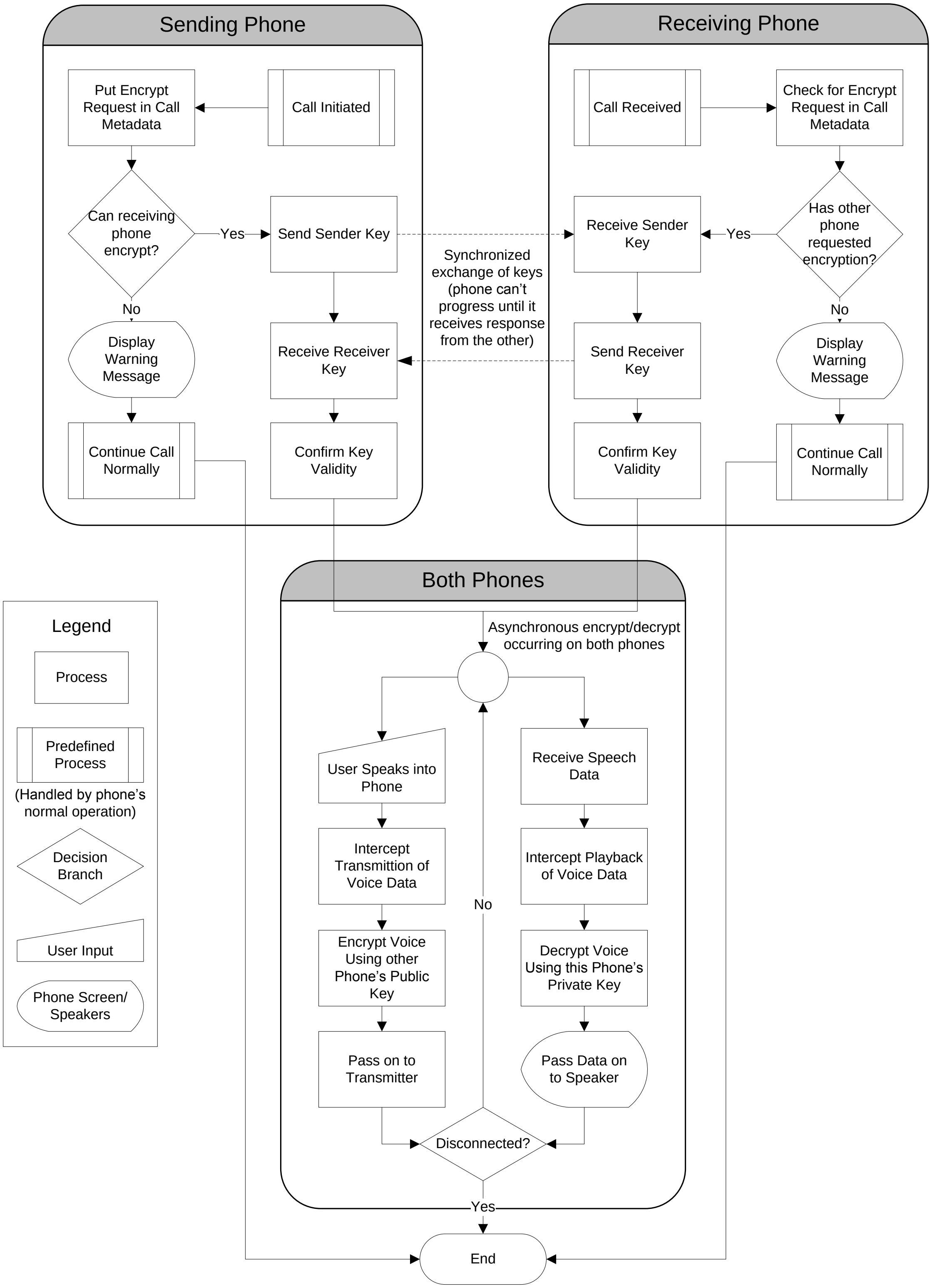
Encrypted Cellphone Communication

Level 1 Diagram



Encrypted Cellphone Communication

Level 2 Diagram


Sending Phone
Put Encrypt Request in Call Metadata
Call Initiated
Can receiving phone encrypt?
Yes
Send Sender Key
No
Display Warning Message
Receive Receiver Key
Continue Call Normally
Confirm Key Validity
Synchronized exchange of keys (phone can't progress until it receives response from the other)
Receiving Phone
Call Received
Check for Encrypt Request in Call Metadata
Receive Sender Key
Yes
Has other phone requested encryption?
No
Send Receiver Key
Display Warning Message
Confirm Key Validity
Continue Call Normally
Both Phones
Asynchronous encrypt/decrypt occurring on both phones
User Speaks into Phone
Receive Speech Data
Intercept Transmittion of Voice Data
Intercept Playback of Voice Data
No
Encrypt Voice Using other Phone's Public Key
Decrypt Voice Using this Phone's Private Key
Pass on to Transmitter
Pass Data on to Speaker
Disconnected?
Yes
End
Legend
Process
Predefined Process
(Handled by phone's normal operation)
Decision Branch
User Input
Phone Screen/ Speakers

Statement of ABET Concerns

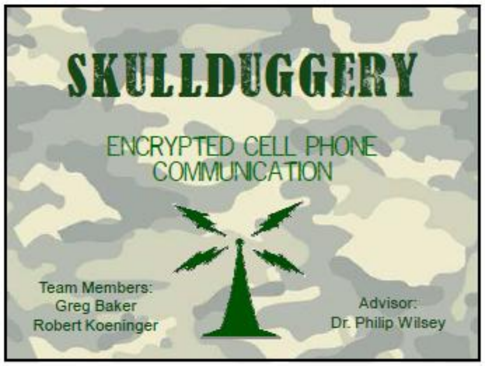
Economic constraints on the Skullduggery project are numerous. Since we are modifying the operating system in a smart phone, we need to consider costs between two phones. Each phone is expensive, between $200 and $425. Each phone requires a data plan with at minimum a $20/month contract, frequently with a set-length lock-in. The total cost of this project may be as high as $700 per person, or $1400. The software and development environment is the cheapest part of the project, each are free. We have no grants or sponsors for the project.

Ethical concerns of the Skullduggery project are common with any encryption-based project. The project may be used to conceal illegal activities. Testing the efficacy of the phone encryption may be difficult and illegal, and would require tapping into the phone conversation. Depending on the contract with the cell-phone provider, modification of the phone may cause a breach of contract. The goal of the project is to protect the privacy of the user. A successful project will be ethically sound by designing testing methods that do not breach laws or privacy of cellular users in the area and do not violate signed contracts.

Sustainability past the project scope is relatively bleak. The Skullduggery project is currently open source, which will assist in the long-term maintainability. It may be that the initial implementation of the project is not compatible between phones, or between different versions of the same phone. We do not plan on maintaining the project beyond the end-of-year deadline. However, the project will ideally be written in a manner that allows for fast maintenance and extensibility later. A successful project will be able to be extended to several different smart-phones with minimal changes to the code.

From a manufacturing standpoint, the Skullduggery project aims to be an implementation of an encryption system. A successful project will not necessarily be a stand-alone product that can be installed on any phone, but we should have a demonstrable encrypted conversation between two phones. There should be no barriers to turning the project into a product, but it will require additional work. As the project is not targeted to become a marketable product, there is no requirement to be portable across different phone systems. As a demonstrable project, there are requirements for an installation process and configuration.

Oral Report Slideshow

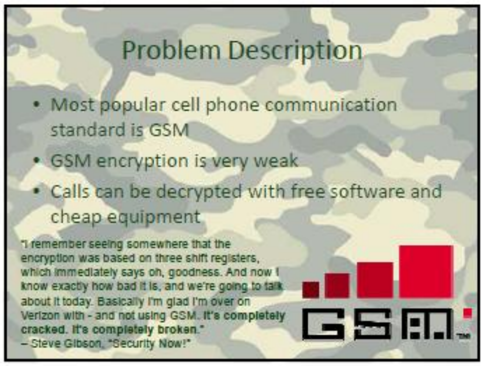


SKULLDUGGERY

ENCRYPTED CELL PHONE COMMUNICATION

Team Members:
Greg Baker
Robert Koeninger

Advisor:
Dr. Philip Wilsey

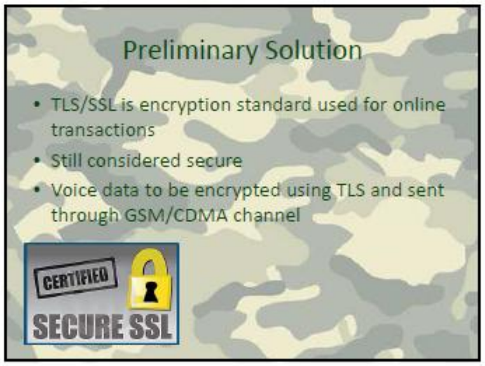


Problem Description

- Most popular cell phone communication standard is GSM
- GSM encryption is very weak
- Calls can be decrypted with free software and cheap equipment

"I remember seeing somewhere that the encryption was based on three shift registers, which immediately says oh, goodness. And now I know exactly how bad it is, and we're going to talk about it today. Basically I'm glad I'm over on Verizon with - and not using GSM. **It's completely cracked. It's completely broken.**"
– Steve Gibson, "Security Now!"

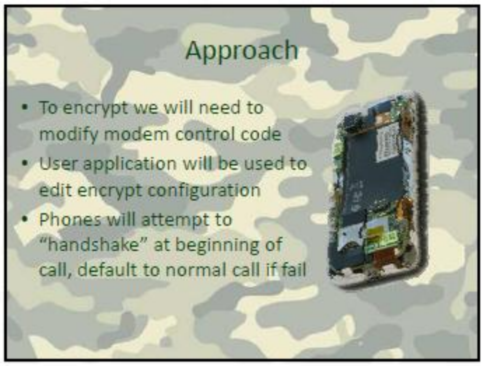
GSM™



Preliminary Solution

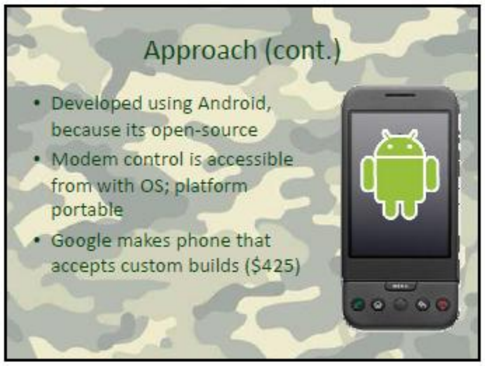
- TLS/SSL is encryption standard used for online transactions
- Still considered secure
- Voice data to be encrypted using TLS and sent through GSM/CDMA channel

CERTIFIED
SECURE SSL



Approach

- To encrypt we will need to modify modem control code
- User application will be used to edit encrypt configuration
- Phones will attempt to "handshake" at beginning of call, default to normal call if fail



Approach (cont.)

- Developed using Android, because its open-source
- Modem control is accessible from with OS; platform portable
- Google makes phone that accepts custom builds ($425)

Gregory Baker bakergo@mail.uc.edu

co-op or other experience and responsibilities

- Software Development Co-op, WhatIfSports.com (1 quarter): Participated in a 5 person multidisciplinary team making the Football Survivor game for FoxSports.com. Added new features to the game simulation engine to track player-specific statistics on a game-by-game basis. Assisted with conversion of forums to a new, upgraded system.
- Web Development Co-op, UCit Web Development (4 quarters): Designed and implemented web applications with administrative functionality for various departments at UC. Maintained and debugged applications. Helped design and write new tools to reduce the cost and development time of writing new applications. Helped deploy new testing environments for UCit.

skills/expertise areas

- Familiar with C#, VB.NET, VB6, Python, PHP, C, C++, Java.
- Familiar with Linux, Windows and OS X
- Software design, testing, time estimation
- Familiar with usage of SQL databases, Source control and regular expressions.

areas of interest

- Programming
- Embedded systems
- Video Games

type of project sought

- I am interested in working on any of the L-3 projects.
- Programming on an embedded device or non-standard environment (such as a robot or low-power system).
- Programming with a unique application, analysis of music or visual systems.
- I'm very good at web development, but it is not preferred.

Self Assessment Essay
Greg Baker

My project involves modifying a cellular phone to more securely encrypt its voice communication.

Classes I have taken that help prepare me for this project include Computer Science II, Data Structures, Software Engineering, Network Communications and Algorithms I & II. The basics of public key encryption were talked about in Algorithms Design II, and many of the earlier classes used C++ to implement coding, which is the language used in the phone OS. The Android phones use Java, which I learned during High School Computer Science, and that prior experience will be useful if we use that.

Much of the code will be written in Java, which is similar syntactically to the C# I used during my co-ops. I am used to working with large, unfamiliar code bases due to my experience, so after a few days of browsing through it I will hopefully become proficient enough to throw something that works together. Much of the code will be written in Java, with a portion of the code written in C and C++.

Robert Koeninger koeninrc@email.uc.edu

co-op or other experience and responsibilities

- Web Developer and Tester, ITI Transcendata, Milford (3 quarters): Installed and maintained networked application on variety of machines and platforms, Performed regression testing and interface testing, developed web component to upload files
- Control Software Developer and Tester, Siemens Energy and Automation, Mason (2 quarters): Developed software packinge in C++ to store program state (all members in an object hierarchy), performed unit and regression testing
- Webmaster, UC Center for Community Engagement, Cincinnati (1 quarter): Maintained website content, worked with CMS, investigated web design

skills/expertise areas

- Software languages and paradigms: C/C++, Assembler, Java, OOP
- Software tools, Visual Studio, Eclipse, Clearcase, Previously maintained CVS on home server
- Web languages and technologies: HTML/CSS/JavaScript/AJAX, PHP, Java Servlets, SQL
- Web tools and software: Tomcat, Apache, MYSQL, Linux as a server, Macromedia MX Studio
- Operating Systems: Windows 4x and NT, various distrobutions of Linux, training with Mac OS X, Worked with other UNIX-style systems like Solaris, HP-UX and AIX

areas of interest

- Software development
- Multimedia - audio, video, image processing
- Custom web development
- Network programming

type of project sought

- Project involving manipulation or recognition of images, video, or audio
- Projects involving network communication

Self-Assessment Essay

Robert Koeninger

My project involves modifying the functionality of a commercial cell phone to encrypt it's voice communication.

Classes I have taken to prepare me for this include Computer Science, Software Engineering, Network Communications and Algorithm Design. Some of the code will be written in C/C++, which I became fluent in while working at Siemens Energy and Automation. Other parts will be modules written in Java, which I became familiar with while working at ITI Transcendata and on my own in high school. Also during co-op, I had experience in reading other developer's code, something that is a bit of a weakness for me. On the non-technical side, co-op and Software Engineering Lab taught me how to describe ideas to others as that is another weak point of mine. Most of the skills I will need to do this project I learned from classes and co-op experience.

Team Budget To-Date

To-Date, there have been no expenses and there were no expected expenses.

Billable Hours

Greg Baker	Robert Koeninger
Week of October 19	**Week of October 19**
Meetings: 2	Meetings: 2
Independent work: 1	Independent work: 0
Week of October 26	**Week of October 26**
Meetings: 2	Meetings: 2
Independent work: 2	Independent work: 1
Week of November 2	**Week of November 2**
Meetings: 5	Meetings: 5
Independent work: 2	Independent work: 2
Week of November 9	**Week of November 9**
Meetings: 1.5	Meetings: 1.5
Independent work: 0	Independent work: 1
Week of November 16	**Week of November 16**
Meetings: 2.25	Meetings: 2.25
Independent work: 0	Independent work: 2
Week of November 23: 0	**Week of November 23: 1**
Week of November 30:	**Week of November 30:**
Meetings: 5	Meetings: 5
Independent work: 0	Independent work: 2
Total: 15.75, @ $75.00/hr = $1181.25	Total: 19.75, @ $75.00/hr = $1481.25

Skullduggery Team Contract

In order to ensure good communication we will set meeting times two days in advance. An agenda will be made of topics for discussion before the meeting.

To ensure accountability, milestones will be set at a given date. Demonstrations of functionality should be given at meetings.

To give direction while a decision is being made, a deadline will be set for making overall design and functionality decisions. In case of a deadlock, the simpler design decision will be made. Designs will favor simplicity and speed over features and robustness.

A research log will be kept, documenting sources of all resources viewed and a brief summary of any potentially relevant sources.

A meeting log will be kept, addressing decisions made during the course of the meeting as well as any unplanned topics that came up during the meeting.

Team Members will keep an individual worklog documenting the hours spent working toward milestones.

Team members signing this document agree to contribute to the project in accordance with the above guidelines.

Gregory L. Baker ______________________________ Date ____________

Robert C. Koeninger ______________________________ Date ____________