DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON

PROJECT CHARTER
CSE 4316: SENIOR DESIGN I
FALL 2023

ESMS
ENCRYPTED SMS

Gilbert Lavin
Jacob Holz
Landon Moon
Nam Huynh
Parker Steach

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	9.13.2021	LM	document creation
0.2	9.15.2021	ALL	complete draft

CONTENTS

List of Figures

1 Problem Statement

In the modern age, people and businesses use messaging apps or text messages in order to communicate with each other. Messaging apps claim to be secure but use internet protocols to send the data through centralized servers. These servers expose a possible security risk to your private messages. Additionally, newer texting protocols such as the ones used by Android and Apple are unable to communicate with each other so SMS is used as a default. The SMS messaging protocol is insecure due to information being unencrypted. We believe people deserve an independent and secure way to message others without the possibility of a third party viewing your message and regardless of the type of phone they use.

-Landon

2 Methodology

We will build an app that allows users to message each other with a truly private service. When a user sends a message the app will encrypt the contents with a key, it is then directly sent to the recipient, the recipient will decrypt the contents using the same key that is securely shared between both users.

- Parker

3 Value Proposition

Any entity whether that is a single user or a multimillion-dollar business, has the need to send information securely. Additionally, in the modern age, there is a greater risk of hackers getting access to your data from the services that you use. These services might save messages on an internet server that has the possibility of being attacked. By using a secure messaging solution, an entry point into the company's private data is no longer possible. By investing in ESMS, businesses can securely send messages to co-workers with company secrets without the risk of a 3rd party saving their messages.

-Landon

4 Development Milestones

- Project Charter first draft - September 2023
- System Requirements Specification - October 2023
- Architectural Design Specification - November 2023
- Demonstration of <feature or implementation milestone> - December 2023
- Detailed Design Specification - February 2024
- Demonstration of <feature or implementation milestone> - February 2024 - sprint 1
- Demonstration of <feature or implementation milestone> - March 2024 - sprint 2
- Demonstration of <feature or implementation milestone> - March 2024 - sprint 2
- CoE Innovation Day poster presentation - Month Year
- Demonstration of <feature or implementation milestone> - April 2024 - sprint 3
- Demonstration of <feature or implementation milestone> - May 2024 - sprint 4
- Final Project Demonstration - May 2024

- Parker

5 Background

Messaging apps send data one of two ways: through SMS which is unencrypted, or through the internet which uses a server to relay messages. Nowadays phones use modern protocols that are encrypted but are sent through the internet. Apple uses a protocol called APN and Android uses a protocol called RCS. Because the two big phone OS systems can't agree on a single protocol, messages sent between the two default to SMS which as stated before is insecure. There are also apps such as Whatsapp and Facebook Messenger that are available on both platforms but utilize a server to relay information between phones. This exposes two risks to anybody who uses these services. The first is that user information has the possibility of being accessed through intentional means by the service. This could be a backdoor that the company uses to moderate your messages, or for police to see your 'encrypted' messages. Secondly, since your information might be saved on a server, attackers don't need to attack you directly to get your messaging history. Users depend that their messages are either not saved or have no security vulnerabilities. ESMS avoids these risks by using the SMS protocol which can be found on any phone and is a peer-to-peer protocol. Encrypting SMS messages doesn't require an internet server to handle requests and keeps all of a user's data on the user's phone. This gives security guarantees to the messages that people and businesses send every day.

Businesses, like any other users, need to communicate and send messages. Businesses have a lot of internal communication with each other that can contain company secrets. Most companies depend on service providers similar to the ones mentioned above which suffer from the security risk of those providers. Companies that are focused on security consider all of their attack surfaces which include their internal service communication. As long as the data is stored outside the company network, ESMS removes an attack surface by eliminating the risk of the external server from being hacked or back-doored.

ESMS has the opportunity to provide a single solution to securely send messages between any two phones. This provides a security guarantee for both individual users and internal communication at a large company.

-Landon

6 Related Work

has this ever happened to you? Say you're texting your significant other and you want things to get a little spicy so you send a picture of your left pinky toe. you hit send and realize it was to your grandma. or another scenario where you want to have a private conversation with your parents about how you "accidentally" ran over your bully's dog and you don't want authorities or a middleman looking at your convoys easily. With our state-of-the-art the art app your grandma would have received gibberish and the authorities wouldn't be knocking on your front door with our fancy encryption functionality. there are a few apps such as ours like Telegram.(www.worsethenours.com)

ProTip: Consider using a citation manager such as Mendeley, Zotero, or EndNote to generate your *.bib* file and maintain documentation references throughout the life cycle of the project.

nam huynh(work in progress (i think thats how you spell it))

7 System Overview

Explain, at a high level, how you will implement a solution to the problem. Include a diagram of the major components of the system (not a full architectural design, but a high-level overview of the major system components and how a user or external system might interface). Avoid specific implementation details (operating system, programming languages, etc.). This section should occupy at least 1 full page.

Our app will be built for Samsung phones first. The idea is that when thinking of messing systems it has many layers from servers at the bottom and SMS at the top. Our group plans to add another top

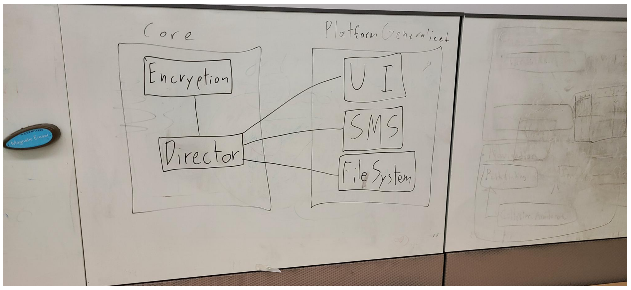

Figure 1: Enter Caption

layer above SMS to add security. SMS does not need to run through a server so with the implementation of an additional layer it's nearly impossible to unencrypt our messages. With that said we will have a GUI for users to see messages of specific users and a screen to show total contacts. If time allows the screen that shows the chat between users may have multiple chats at once. Underneath those screens will be a file system containing all contacts and user info. These will all be mostly front-end user-friendly things that may be accessed. The core of the code will be more of the back-end uses. The job of the app is to have the user send a message and have their message encrypted and sent to a fellow app user showing only them the text inside the app. Users without the app will only see Giberish in their actual SMS app with a link inviting them to download and join. When the app is downloaded the user will have an icon of our logo show up on their home screen and when clicked the app will open to a Similar feeling SMS app that everyone with a smartphone is already familiar with. A screen showing an empty contact book that the user may add contacts to will appear and won't stay empty for long. A plus symbol will be on the screen to add users having their numbers and names displayed in rows on the contact screen. Then once added the user may click their name to open a new screen that shows a typing bar that lets the user text in the app. When texting the user will see the message appear above the text bar and the contacts messages as well. A settings screen will also be present to store the users and their contact information. Using the app should be just like any other messaging app on the market with the benefit of knowing that your texts are secure. The app will be using an encrypting language in the back end. -Gilbert Landon(I have no clue on how to encrypt answer the rest thank you)

8 Roles & Responsibilities

The world is our stakeholder because anyone can download the app and benefit from the security it can give. Counting they have a smartphone to download the app. Even if most of the public don't download the app or don't have a smartphone they are still being affected by the lack of knowledge that has been decripted thanks to our app. Every day people will be our source of contacts along with our sponsor. The team members are Jacob Holz, Landon Moon, Gilbert Lavin, Nam Huynh, and Parker Steach. The product owner is Jacob with co-ownership being owned by the group. The scrub master will be Gilbert for the entirety of the project. The project will be split into two teams, the front end and the back end. Gilbert, Nam, and Parker are all working on the front end. The front end is focused on the GUI consisting of the three screens to keep up with the user's chats and people the user knows. Jacob and Landon are on the backend working on the description and the actual SMS structure to connect our users. All team members will be product testers and documenters. With the help of some friends, family, or fellow students they will be the first product testers as well to get outside opinions for our app. -Gilbert half a page neeeded

9 Cost Proposal

ESMS will need an Apple Developer Program membership in order to publish our application to the Apple App Store. If we are unable to get the membership, we will only be able to do it on Android devices.

9.1 Preliminary Budget

Item	Amount	Sponsor/Company
Department Provided Funding	$800	CSE department
Apple Developer Program Membership	-$100/yr	Apple
Android Developer Account	-$25	Google

Table 1: Overview of highest exposure project risks

9.2 Current & Pending Support

We have access to $800 from the CSE Department. We are unlikely to pursue or receive any other funding at this time.

Nam Huynh

10 Facilities & Equipment

The app can be created anywhere but for the majority of the time it will be worked on in the senior design lab, ERB, UTA library, WH, UC, and at the given group-mates houses, mainly Gilbert's 55 million dollar Mansion since it will have the most space and entertainment such as a massive saltwater fish tank bigger than most people houses. Nam's box counting, he doesn't have a house. Jacobs virtual reality simulator where he plugs himself in to study and rest. The locations for working on this project are very open and free counting 89 percent of the project is just digital work on a laptop or computer. In retrospect, if there is wifi available we can use that area to work but preferably our group will stick to regular buildings in UTA to keep things simple and connected. When testing does occur our team plans to download it among our sleeves and use it among us. Once the app has been tested for a while we will open it to the class, then UTA, then the Dallas/Fort Worth area, then the world. To accomplish this task We will need laptops and computers to run and write the code, maybe some specific libraries to make it easier would help too. Then once the code is functional our group will need phones to text each other. If our group has time we would like to port the app to Apple. To do this we need an Apple license and access to Apple hardware. to attain these items our group already has laptops and home computers(Apple included). Along with phones (Apple and Samsung) to help test the app. Our team members will also need multiple cellular contracts to actually send each other messages. Along with many willing people to be test subjects near the end of our project. -gilbert Lavin half a page

11 Assumptions

The following list contains critical assumptions related to the implementation and testing of the project.

- Cryptographically secure encryption algorithms should not be implemented manually, so trusted implementations of any algorithms to be used must already be available for our chosen platform
- SMS is a very insecure data transfer method
- Distribution of a End-to-End encrypted communication application is legal

- All users have unlimited texts, so inflated character counts will not be an issue
- Modern SMS capable devices are sufficiently computationally powerful to encrypt and decrypt messages quickly

Jacob Holz

12 Constraints

The following list contains key constraints related to the implementation and testing of the project.

- Final prototype demonstration must be completed by May 1st, 2023
- Mobile devices are less computationally powerful than many static devices, secure cryptography will introduce noticeable lag time
- Development for Apple devices requires use of an Apple computer and publication requires a $99 Apple Developer Program membership
- Total development costs must not exceed $800
- Any stored data must be able to be encrypted at the user's discretion

Jacob Holz

13 Risks

Risk description	Probability	Loss (days)	Exposure (days)
Apple development access delayed	0.50	20	10
Unforeseen complexity in detecting receipt of SMS	0.50	10	5
Environment configuration errors	0.90	3	2.7
Vital features lacking on some platforms	0.33	3	1
Vehicular malfunction	0.50	0.2	0.1

Table 2: Overview of highest exposure project risks

Jacob Holz

14 Documentation & Reporting

- Parker

14.1 Major Documentation Deliverables

These deliverables are major grade components of the course. Completing these documents should generally be the sprint goal during the applicable sprint period. Refer to current and previous course syllabi and schedules to estimate the due dates of these items. Remove this explanatory paragraph from your draft, but leave the heading.

(Parker) - I imagine intial completions of dates will need to change, however i just wanted to get formatting completed. then we could discuss actual dates in a meeting

- only unfinished sections were "System prototype", "source code", "source code documentation"

- delete

14.1.1 Project Charter

At the end of every sprint we will review our work and confirm we are still following the charter. If any changes have happened, we will discuss and update the charter as needed (decide who will do it, either scrum master, or have assigned sections for each person to update). The initial version of the project charter will be completed by September 18 . We will complete the charter before we begin development on September 25.

14.1.2 System Requirements Specification

After the document is created, we will revisit the document towards the end of each sprint and if any changes have happened, we will discuss and update the charter as needed. The initial version of the SRS will be completed October sixth. We will have a final version of the SRS by October 16.

14.1.3 Architectural Design Specification

After the document is created, we will revisit the document towards the end of each sprint and if any changes have happened, we will discuss and update the charter as needed. The initial version of the ADS will be completed October 30. We will have a final version of the ADS by November sixth.

14.1.4 Detailed Design Specification

After the document is created, we will revisit the document towards the end of each sprint and if any changes have happened, we will discuss and update the charter as needed. The initial version of the DDS will be completed February sixth. We will have a final version of the DDS by February 13.

14.2 Recurring Sprint Items

The following items will be documented and maintained during each individual sprint. As above, remove this paragraph from your draft, but leave the heading.

- delete

14.2.1 Product Backlog

Once the SRS has been finalized, the team will take the SRS and turn them into product items. The team will take a group vote to prioritize which items to complete in the sprints. Trello will be used to maintain the product backlog.

14.2.2 Sprint Planning

Sprints will be planned by the scum master with input from the rest of the team. There will be a total of eight sprints from September 11 to April 24.

14.2.3 Sprint Goal

Sprint goals will be determined by our class syllabus. More development focused sprints will be determined by the scrum master. Three to five days before our sprint begins we will email/meet with our customer and inform them of our sprint goal.

14.2.4 Sprint Backlog

The sprint backlog will be determined by the scrum master, as they plan out each sprint. Similar to the product backlog, our spring backlog will be maintained using Trello.

14.2.5 Task Breakdown

At the start of each sprint we will let individual members claim any specific tasks they want to work on. If there are any disputes or leftover tasks, the scrum master will distribute those tasks.

–decide how to track time - trello? - excel sheet, shared on git

14.2.6 Sprint Burn Down Charts

We will use Excel/Sheets to generate our sprint burn down chart, this file will be in the project Github, therefore anyone could generate the chart for each sprint. (look at previous section to answer, how will we access the total amount of effort (time) for each member). (describe burn down chart)

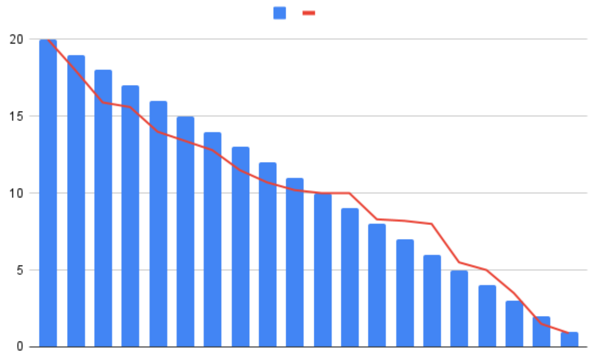

Figure 2: Example sprint burn down chart

14.2.7 Sprint Retrospective

On the last Friday before a sprint ends we will have our sprint retrospective. In these meetings our members will discuss aspects that went well, what could've been improved, and what needs to be changed. Any thing notable, that fits into the three aforementioned categories, will be written down in the group document. Any personal feedback will be recorded by that individual (possibly recorded in group doc)

14.2.8 Individual Status Reports

We will have group meetings three times a week Mon/Wed/Fri, in these meetings each member will give a status update. This will include progress on current tasks for this sprint, how much estimated man-hours is left before each task is completed, and if they believe assistance is required to complete their tasks.

- any other important info to deliver? - written?

14.2.9 Engineering Notebooks

As of Fall 2023, we will not be required to keep or maintain an engineering notebook

14.3 Closeout Materials

The following materials, in addition to major documentation deliverables, will be provided to the customer upon project closeout. Remove this paragraph from your draft, but leave the heading.

- delete

14.3.1 System Prototype

What will be included in the final system prototype? How and when will this be demonstrated? Will there be a Prototype Acceptance Test (PAT) with your customer? Will anything be demonstrated off-site? If so, will there be a Field Acceptance Test (FAT)?

The final system prototype will be a set of software that will allow people to message each other, securely, through SMS. This will be demonstrated at the end of the Spring 2024 semester, we will provide phones/quick download for those interested (idk yet). We will do a Prototype Acceptance Test with our customer

– unfinished

14.3.2 Project Poster

Our poster will include information about the security issues with modern day messaging services and explain how our app is different and secure. Dimensions (tbd). Our poster will be delivered with out project demonstration at the end of April/ early May.

14.3.3 Web Page

Our web page will contain our goal for this project, explain the issue with modern messaging systems and visual of how our app is different. This will delivered with our project at the end, with out project poster, April 30 (guess).

14.3.4 Demo Video

What will be shown in the demo video(s)? Will you include a B-reel footage for future video cuts? Approximately how long will the video(s) be, and what topics will be covered?

The demo video will provide a video (ideally some form of animation, maybe like a Kurzgesagt type video) this will explain how our app works at a very surface level so anyone can understand how our app works.

14.3.5 Source Code

How will your source code be maintained? What version control system will you adopt? Will source code be provided to the customer, or binaries only? If source code is provided, how will it be turned over to the customer? Will the project be open sourced to the general public? If so, what are the license terms (GNU, GPL, MIT, etc.). Where will the license terms be listed (in each source file, in a single readme file, etc.).

–unknown, talk with group We will use git to maintain our code

14.3.6 Source Code Documentation

What documentation standards will be employed? Will you use tools to generate the documentation (Doxygen, Javadocs, etc.). In what format will the final documentation be provided (PDF, browsable HTML, etc.)?

– unknown, talk with group

14.3.7 Hardware Schematics

This project is a software focused and does not require any hardware schematics.

14.3.8 CAD files

This project is a software focused and does not require any CAD files.

14.3.9 Installation Scripts

Our app will be on the android store and apple store (ideally) so the user will just need to download the app, and update it as we put out updates.

14.3.10 User Manual

Our apps interface will mimic modern messaging apps, so it will be assumed that the user knows how to text someone. Any encryption related information will be handled in the background, abstracted away from the user.

REFERENCES