Department of Computer Science & Engineering
The University of Texas at Arlington

Architectural Design Specification
CSE 4316: Senior Design I
Fall 2023

ESMS
Encrypted SMS

Jacob Holz
Nam Huynh
Gilbert Lavin
Landon Moon
Parker Steach

Revision History

Revision	Date	Author(s)	Description
0.1	10.23.2023	LM	Document created
0.2	11.3.2023	ALL	First complete draft

CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

The Encrypted SMS Messaging System (ESMS) introduces a secure way of messaging peers using SMS as its main communication method. This strategic choice eliminates the need for proprietary relay servers, mitigating the risk of unauthorized access to private messages and back-doors. ESMS allows users to customize the security level of their communications, providing transparency and control over the encryption methods employed.

ESMS is designed to enable secure text messaging between users, providing an interface and experience comparable to modern-day messaging applications. The system is unique due to its reliance on SMS being used as an insecure channel for transmitting encrypted data. This ensures that even if the SMS messages are intercepted, the content remains secure and confidential. The application allows users to select their preferred level of encryption, with a secure industry standard protocol set as the default.

The scope of ESMS includes the development of a user-friendly application that enables secure messaging, contact management, and conversation navigation. It encompasses the creation of robust encryption systems, an intuitive user interface, and seamless integration with the phone's SMS service and contact information. The application is intended to serve as a standalone communication tool, accessible and valuable to anyone interested in protecting their communication privacy but also as a learning tool into the concepts of encryption and data security.

1.1 KEY REQUIREMENTS

- ESMS must have the capacity to encrypt text messages to varying security levels based on user settings before transmitting via SMS.
- ESMS should offer a user-friendly interface, allowing easy navigation between conversations, access to contact information, and adjustment of security settings.
- ESMS should educate users on communication security concepts and clearly communicate the nature of the encryption methods in use.
- ESMS must extract the phone's contact information, providing a seamless experience for the user.
- ESMS must utilize the phone's messaging history, in order to allow users to access past conversations within the application.
- ESMS must allow users to customize application settings, including encryption methods, on both a global and per-conversation basis.
- ESMS must store parameters between sessions.

2 System Overview

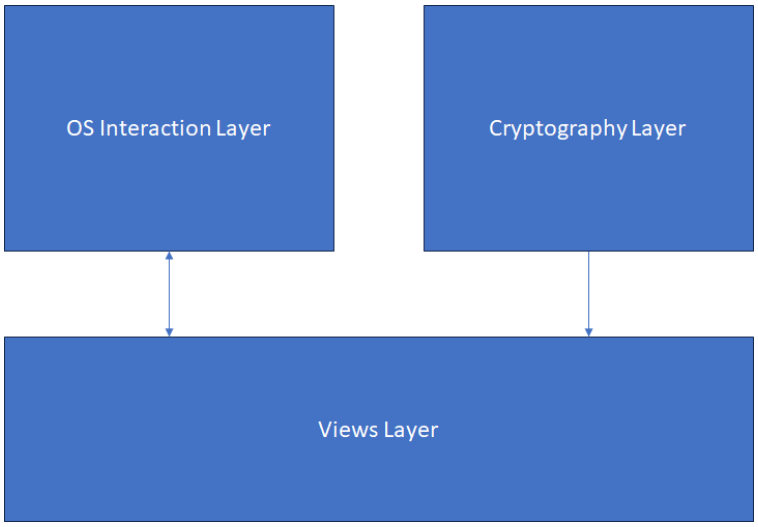

Figure 1: A simple architectural layer diagram

2.1 OS Interface Layer Description

The OS Interface Layer will handle all system related functionality of the app. The SMS system is where messages will be sent and received. The file storage system will store all settings the users sets, so they will stay the same over different sessions with the app. The contacts system is where all the users contacts will be stored on the device and accessed.

2.2 Cryptography Layer Description

The Cryptography Layer handles all encryption in the app. The Engine generator will take in the parameters that define an engine and return a cryptography engine for use else were. We will utilize predefined encryption libraries and custom encoding algorithms for our cryptography engines. The engines will handle encrypting and decrypting text. For example, any time the user receives or sends a message, it will use an engine to manipulate the text. Additionally the parameters view will allow the user to select the method of encryption for messages sent. The custom engines subsystem will be our own implementation of encryption, while these will not be as secure as the predefined libraries, they will allow us to inform and entertain the end user.

2.3 View Layer Description

The View layer is for displaying screens to the user. This layer will receive all of the users input, this includes changing settings, sending messages to contacts, and viewing contacts and conversations. We will separate these uses into 3 different pages: The Parameters View - manage local and global settings, Contacts View - view and alter all contacts and enter conversations with individual contacts, and Conversation View - view and send messages with a specific contact.

3 Subsystem Definitions & Data Flow

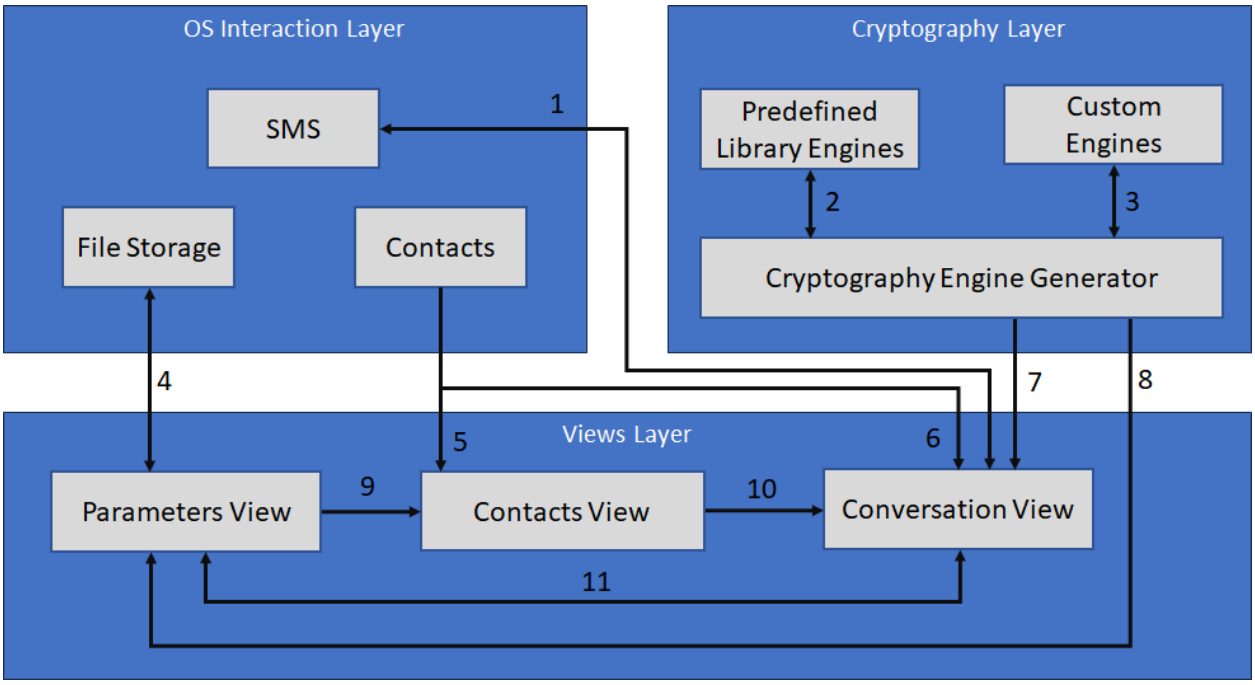

Figure 2: A simple data flow diagram

Table 2: SMS Subsystem

ID	Description	Inputs	Outputs
#1	Conversation View requests that an SMS message be sent containing the given message to the given user and receives confirmation	Message, Recipient	Confirmation Object
#1	Conversation View requests and receives a list of messages involving a given user	User Identifier (Phone Number)	Array of Message Objects
#4	Parameters View requests and receives the stored parameters string	N/A	Parameters String
#4	Parameters View requests that a given parameters string is stored to the device file system and receives confirmation	Parameters String	Confirmation Object
#5	Contacts View requests and receives the contacts list from the Android Contacts service	N/A	List of Contact Objects

#6	Conversation View requests and receives the contact from the Android Contacts service for the specified user	User Identifier (Phone Number)	Contact Object
#2	Cryptography Engine Generator will request and receive a Cryptography Engine with the specified type and parameters wrapping cryptography library calls	Engine Type Enum, Additional Parameters	Cryptography Engine Object
#3	Cryptography Engine Generator will request and receive a Cryptography Engine with the specified type and parameters wrapping custom made encoding functions	Engine Type Enum, Additional Parameters	Cryptography Engine Object
#7	Conversation View will request and receive a Cryptography Engine with the specified type and parameters	Engine Type Enum, Additional Parameters	Cryptography Engine Object
#8	Parameters View will request and receive a Cryptography Engine with the specified type and parameters	Engine Type Enum, Additional Parameters	Cryptography Engine Object
#9	Contacts View will request and receive the important information regarding a specified Contact	Contact Object	Saved Contact Parameters
#10	Upon entering a conversation, the Conversation View is given the Contact Object for the specified user	Contact Object	N/A
#11	Upon entering the Parameters View, the Parameters View is given the Contact Object for the specified user	Contact Object	N/A
#11	Conversation View will request and receive the important information regarding a specified Contact	Contact Object	Saved Conversation Parameters

4 OS Interface Layer Subsystems

4.1 SMS Subsystem

This subsystem wraps the Android systems for sending and receiving SMS communications. This allows for more readable and simple code.

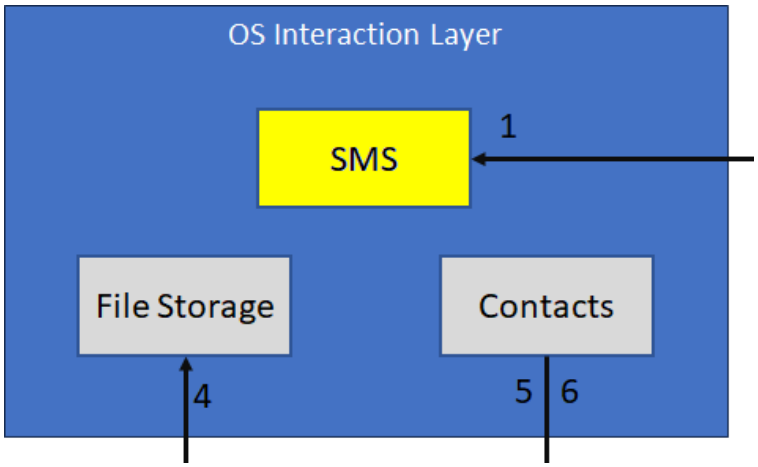

Figure 3: SMS Subsystem

4.1.1 Assumptions

- The user will grant permissions for ESMS to send and receive SMS messages.
- The device running ESMS is capable of transmitting and receiving SMS.
- SMS access will continue to be permitted on Android devices.

4.1.2 Responsibilities

- It will receive encrypted text from the Conversation View and send it to the receiving device through the SMS protocol by way of the Android SmsManager utility.
- It will handle providing SMS messages for the conversation view using the Android ContentResolver utility on the context file containing the SMS information.

4.1.3 Subsystem Interfaces

Table 3: SMS Subsystem

ID	Description	Inputs	Outputs
#1	Conversation View requests that an SMS message be sent containing the given message to the given user and receives confirmation	Message, Recipient	Confirmation Object
#1	Conversation View requests and receives a list of messages involving a given user.	User Identifier (Phone Number)	Array of Message Objects

4.2 File Storage Subsystem

This subsystem wraps the Android systems for storing information between sessions. This allows for more readable and simple code. It also allows for flexibility of implementation as new and more capable strategies become available.

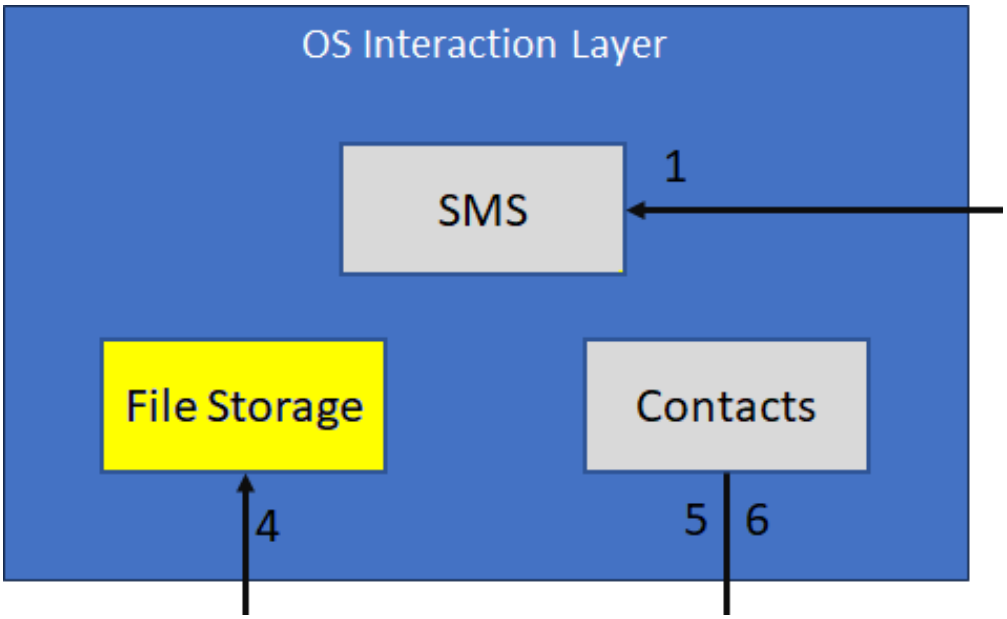

Figure 4: File Storage Subsystem

4.2.1 Assumptions

- The device has sufficient free space.

4.2.2 Responsibilities

- It will allow the user to store and retrieve encrypted parameters on their local device for continuity of state between sessions.

4.2.3 Subsystem Interfaces

Table 4: File Storage Subsystem

ID	Description	Inputs	Outputs
#4	Parameters View requests and receives the stored parameters string	N/A	Parameters String
#4	Parameters View requests that a given parameters string is stored to the device file system and receives confirmation	Parameters String	Confirmation Object

4.3 Contacts Subsystem

This subsystem will wrap the Android systems for accessing the contacts stored on the user's device. This allows for more readable and simple code.

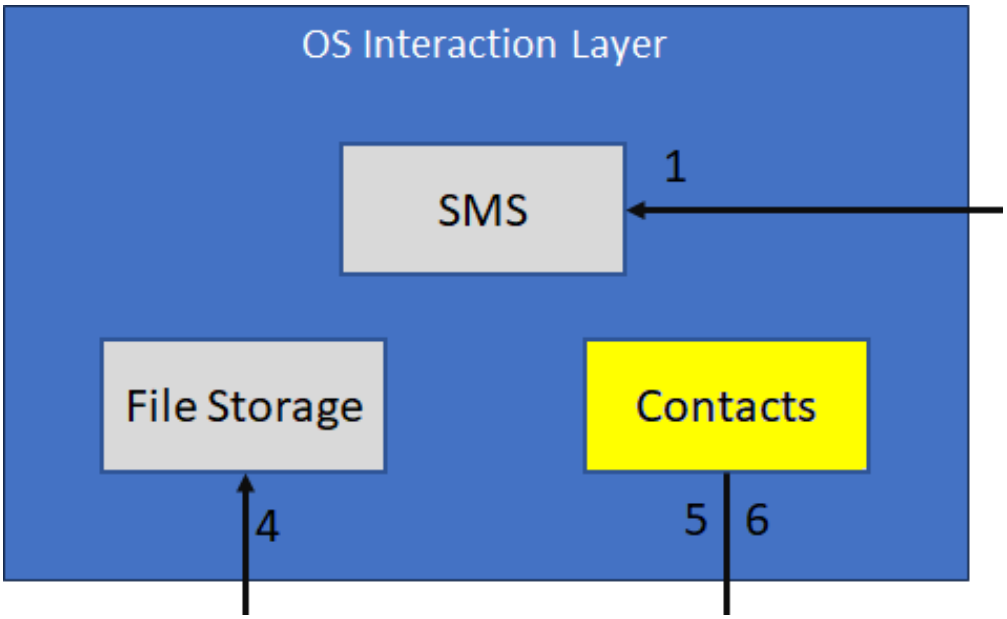

Figure 5: Contacts Subsystem

4.3.1 Assumptions

- The user will grant permissions for ESMS to access their contacts.

4.3.2 Responsibilities

- It will provide data to the Contracts View allowing it to display the user's contacts visually.
- It will provide data to the Conversation View allowing it to display who the user's conversation is with.

4.3.3 Subsystem Interfaces

Table 5: Contacts Subsystem

ID	Description	Inputs	Outputs
#5	Contacts View requests and receives the contacts list from the Android Contacts service	N/A	List of Contact Objects
#6	Conversation View requests and receives the contact from the Android Contacts service for the specified user	User Identifier (Phone Number)	Contact Object

5 Encryption Layer Subsystems

5.1 Cryptography Engine Generator Subsystem

This subsystem takes the minimum details for a cryptography engine and constructs an instance of that engine. This allows greater consistency and efficiency of storage of non-active cryptography engines. This is also the central location where new engines will be registered for later use.

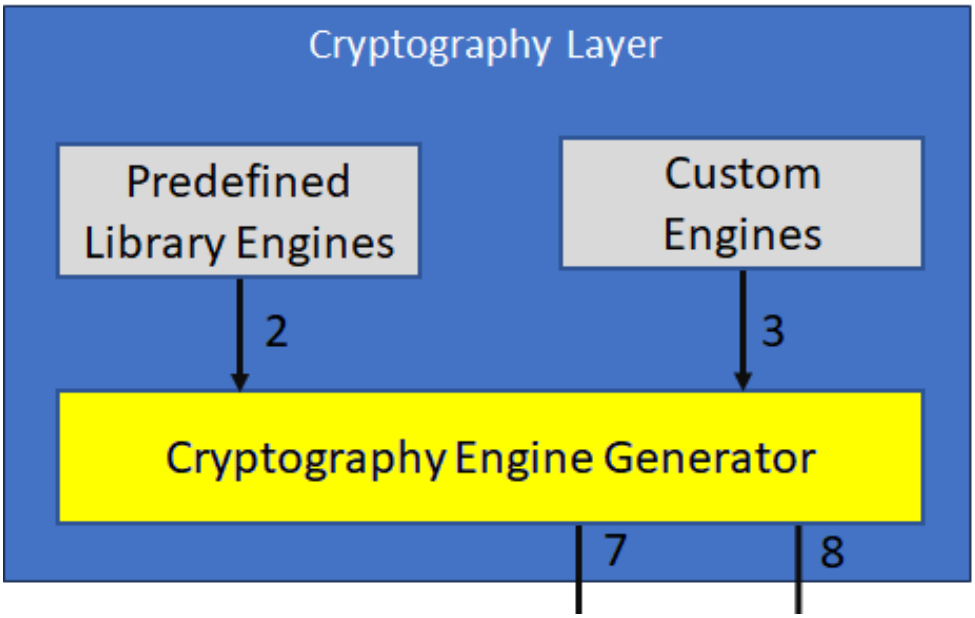

Figure 6: Cryptography Engine Generator Subsystem

5.1.1 Assumptions

- All predefined and created encryption engines comply with an interface that allows inputting a plain text string and receiving an encrypted string.
- All predefined and created encryption engines comply with an interface that allows inputting an encrypted string and receiving a plain text string.
- All predefined and created encryption engines comply with an interface that allows them to be constructed from a string.

5.1.2 Responsibilities

- It will take an enum specifying the encryption engine to be used along with the engine parameters and return the specified engine.

5.1.3 Subsystem Interfaces

Table 6: Cryptography Engine Generator Subsystem

ID	Description	Inputs	Outputs
#2	Cryptography Engine Generator will request and receive a Cryptography Engine with the specified type and parameters wrapping cryptography library calls	Cryptography Engine Object	Engine Type Enum, Additional Parameters
#3	Cryptography Engine Generator will request and receive a Cryptography Engine with the specified type and parameters wrapping custom made encoding functions	Cryptography Engine Object	Engine Type Enum, Additional Parameters
#7	Conversation View will request and receive a Cryptography Engine with the specified type and parameters	Engine Type Enum, Additional Parameters	Cryptography Engine Object
#8	Parameters View will request and receive a Cryptography Engine with the specified type and parameters	Engine Type Enum, Additional Parameters	Cryptography Engine Object

5.2 Predefined Library Engines Subsystem

This subsystem will contain several class definitions for objects following an interface that allow the encryption and decryption of a given string. These classes will be wrappers for existing verified encryption library functions.

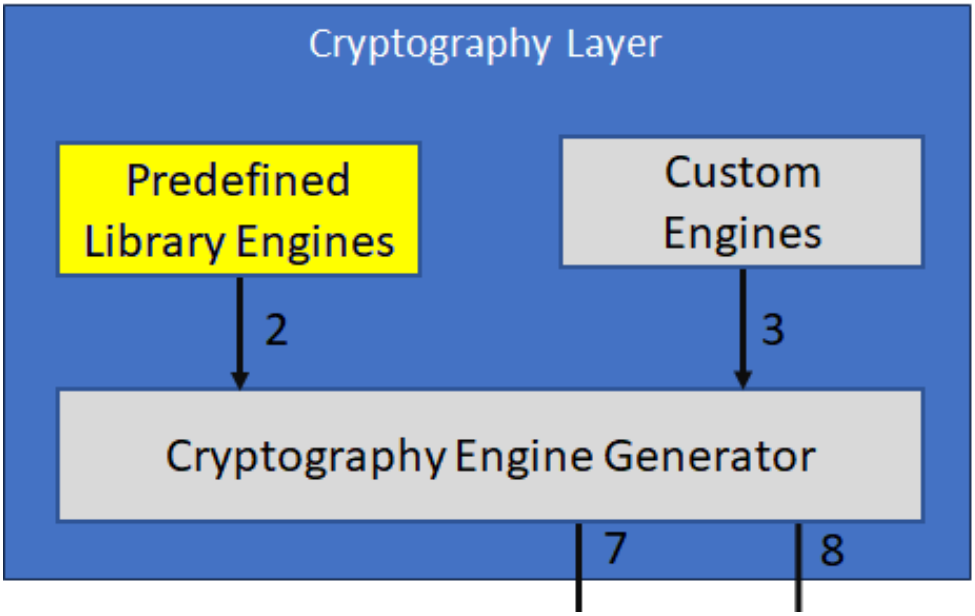

Figure 7: Predefined Library Engines Subsystem

5.2.1 Assumptions

- Assume the library has everything needed to implement the cryptography engine interface.

5.2.2 Responsibilities

- It will provide predefined encryption methods allowing other subsystems to use them such as providing the necessary encryption methods within the returned cryptography engine.

5.2.3 Subsystem Interfaces

Table 7: Predefined Library Engines Subsystem

ID	Description	Inputs	Outputs
#2	Cryptography Engine Generator will request and receive a Cryptography Engine with the specified type and parameters wrapping cryptography library calls	Engine Type Enum, Additional Parameters	Cryptography Engine Object

5.3 Custom Cryptography Engine Subsystem

This subsystem will contain several class definitions for objects following an interface that allow the encryption and decryption of a given string. These classes will be wrappers for custom encoding functions created by ESMS.

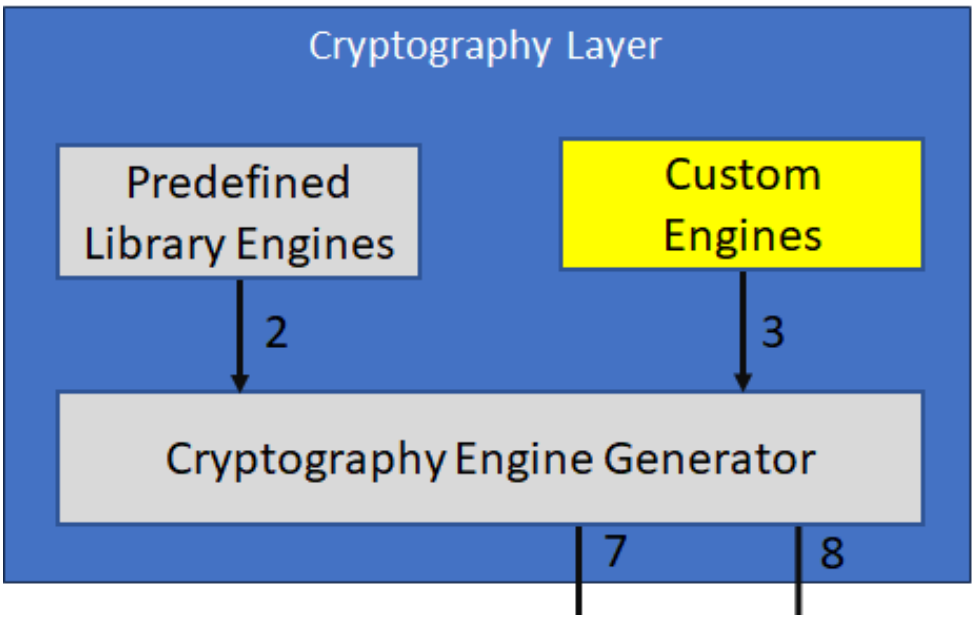

Figure 8: Custom Engines Subsystem

5.3.1 Assumptions

- The user does not expect security from any encryption algorithm we manually implement.

5.3.2 Responsibilities

- It will contain the in house encryption methods for users to use.
- These will not be as secure as the predefined libraries, so it will notify users of this if they select a custom encryption method.

5.3.3 Subsystem Interfaces

Table 8: Custom Cryptography Engine Subsystem

ID	Description	Inputs	Outputs
#3	Cryptography Engine Generator will request and receive a Cryptography Engine with the specified type and parameters wrapping custom made encoding functions	Engine Type Enum, Additional Parameters	Cryptography Engine Object

6 View Layer Subsystems

6.1 Contacts View Subsystem

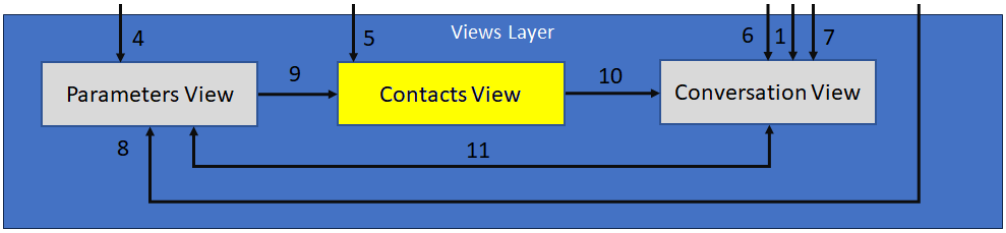

Figure 9: Contacts View Subsystem

6.1.1 Assumptions

- The screen is of a reasonable size and resolution.

6.1.2 Responsibilities

- This subsystem will display all of the user's contacts.
- Clicking on a contact will take the user to the conservation view.
- Meta-data about contacts will be used to inform the user about each contact
- This includes what encryption scheme the application is currently using with that contact along with whether or not that contact is confirmed to have access to the app.

6.1.3 Subsystem Interfaces

Table 9: Contacts View Subsystem

ID	Description	Inputs	Outputs
#5	Contacts View requests and receives the contacts list from the Android Contacts service	List of Contact Objects	N/A
#9	Contacts View will request and receive the important information regarding a specified Contact	Saved Contact Parameters	Contact Object
#10	Upon entering a conversation, the Conversation View is given the Contact Object for the specified user	N/A	Contact Object

6.2 Conversation View Subsystem

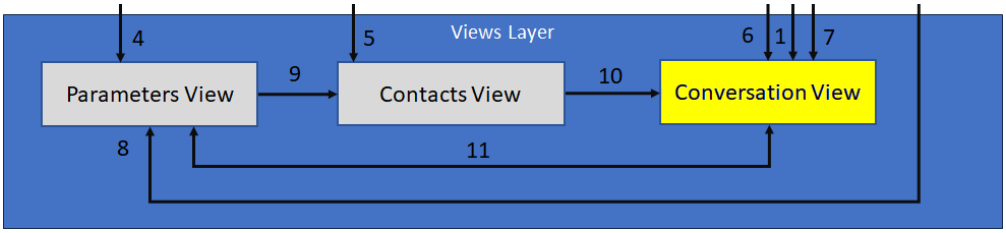

Figure 10: Conversation View Subsystem

6.2.1 Assumptions

- The user has a contact to have a conversation with.
- The screen is of a reasonable size and resolution.

6.2.2 Responsibilities

- It will coordinate the systems responsible for sending and receiving encrypted messages.
- It will display information to the user in a UI consistent with that of other messaging applications.
- It will display past messages.
- It will allow users to send messages.
- It will display the name and image of the contact involved in the communication.
- It will display navigation buttons to access the Contacts View and the Parameters View for the specific conversation.

6.2.3 SUBSYSTEM INTERFACES

Table 10: Conversation View Subsystem

ID	Description	Inputs	Outputs
#1	Conversation View requests that an SMS message be sent containing the given message to the given user and receives confirmation	Confirmation Object	Message, Recipient
#1	Conversation View requests and receives a list of messages involving a given user	Array of Message Objects	User Identifier (Phone Number)
#6	Conversation View requests and receives the contact from the Android Contacts service for the specified user	Contact Object	User Identifier (Phone Number)
#7	Conversation View will request and receive a Cryptography Engine with the specified type and parameters	Cryptography Engine Object	Engine Type Enum, Additional Parameters
#10	Upon entering a conversation, the Conversation View is given the Contact Object for the specified user	Contact Object	N/A
#11	Upon entering the Parameters View, the Parameters View is given the Contact Object for the specified user	N/A	Contact Object
#11	Conversation View will request and receive the important information regarding a specified Contact	Saved Conversation Parameters	Contact Object

6.3 Parameters View Subsystem

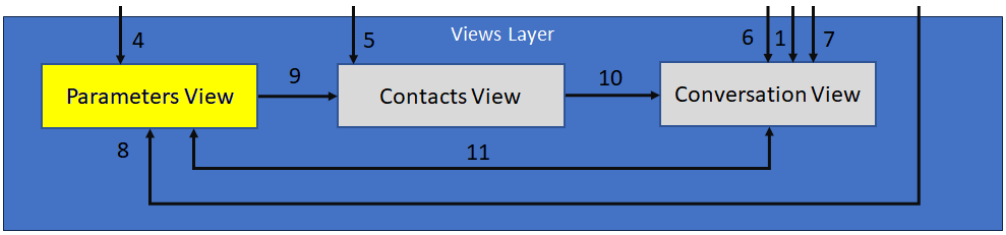

Figure 11: Parameters View Subsystem

6.3.1 Assumptions

- The screen is of a reasonable size and resolution.

6.3.2 Responsibilities

- It will handle all preferences and settings within the app.
- It will allow the user to change those within the GUI.
- It will allow the user to change the encryption method and other settings such as color theme.
- It will display information bubbles allowing the user to learn more about the system.
- It will allow the user to return to the contacts or conversation page from whence they came.
- It will acquiring and storing application data that is not local to any one view.

6.3.3 SUBSYSTEM INTERFACES

Table 11: Parameters View Subsystem

ID	Description	Inputs	Outputs
#4	Parameters View requests and receives the stored parameters string	N/A	Parameters String
#4	Parameters View requests that a given parameters string is stored to the device file system and receives confirmation	Parameters String	Confirmation Object
#8	Parameters View will request and receive a Cryptography Engine with the specified type and parameters	Cryptography Engine Object	Engine Type Enum, Additional Parameters
#9	Contacts View will request and receive the important information regarding a specified Contact	Contact Object	Saved Contact Parameters
#11	Upon entering the Parameters View, the Parameters View is given the Contact Object for the specified user	Contact Object	N/A
#11	Conversation View will request and receive the important information regarding a specified Contact	Contact Object	Saved Conversation Parameters

REFERENCES