AESDroid

Paul Ippolito

Marist College

Dr. Pablo Rivas

MSCS 630L Security Protocols and Algorithms

April 12, 2020

Abstract: In this paper, I propose the idea and methodology of using the Advanced Encryption Standard to encrypt and decrypt messages on an Android application. An issue I face with doing this is how do I go about encrypting or decrypting the user's message accurately without demanding the encryption key to be provided by the user. Can I efficiently and accurately encrypt or decode the user's messages with a random key? How can I use AES to accomplish my mission with this application? This paper delves into my thoughts and ideas and how I have tested them in my Android application I call AESDroid.

1 Introduction: The purpose of this project is to use AES to either encrypt or decrypt a message given from the user. This project is meant to be a semi-simplistic app, at least to the user. The user can either choose to encrypt or decrypt a message that they will input themselves. Afterwards, the app will go through the process of the user's choice and return either the encrypted ciphertext or the decrypted message. I will be using AES-128 in order to accomplish these tasks. I am using Android Studio to develop this application, as it is the IDE of choice for developing, debugging, and maintaining Android software. My primary programming language will be Java, as I am already well versed in its usage, syntax, and debugging.

1.1 Technology: For this project, I intend to show mastery over the Java language after several years of usage. At the time of writing this paper, I have had five years of experience with coding, running, and debugging Java, and I still have much to learn. I intend to code the AES-128 encryption/decryption methods myself rather than use libraries as I feel I am up to the challenge and the result will be more rewarding.

2. Methodologies: Currently, my application can be broken down into three basic Android Activities: Main Menu, Encryption, and Decryption. The user will choose whether they want to encrypt or decrypt a message. For the case of encryption, the user will not be required to enter a key and one will be randomly generated and returned along with the encrypted message after completion, otherwise whatever key they input will be used instead. For decryption, the user either can input the key if they know it or leave the field blank. If the user leaves the key field blank, the program will randomly generate several keys and run through the decryption process, returning each possible decrypted message. My dilemma is how to get this to be ACCURATE, as using a random key to decrypt a message will most likely not return the desired result. I want to give the user the option of not using the key to decrypt a message. Chances are if a user seeks to decrypt something with software, they most likely don't have all the necessary information to do it themselves.

3. Experiments: The AES algorithm I've developed takes two parameters: the plaintext message and the encryption/decryption key. If the user inputs a key, that will be passed in, otherwise a 32-bit hexadecimal key will be randomly generated for use in the algorithm. The message will be broken into length 16 Strings and then converted to 32-bit ASCII values for use, all of which will be formatted to all uppercase. The ciphertext and key will be the encryption activity's output, and just the plaintext will be the decryption output.

4. Conclusion: I claimed this would be simple and efficient. ENCRYPTION is simple as I had previously done an assignment that involved using AES encryption. I believe efficiency is still possible as the Advanced Encryption Standard has yet to be broken (at least of current knowledge). My biggest challenges will be the decryption algorithm, although that should just be a matter of encrypting the message but in reverse and doing the decryption accurately without being given the key that was used to encrypt the message originally. It is going to be a major obstacle within the coming weeks.

5. Completed: As of writing this paper, I have the AES Encryption (with secure keys of course) written and have the layout of the app completed.

5.1. To BE Completed: I still need to implement the decryption process as well as being able to accurately return the correct message if not given a key by the user.