AES Encryption and AccessApp

Brandi Coon

Abstract - With the usage amount of technology and social media being at an all time high, there is always a need for usernames and passwords (i.e. login credentials/“virtual keys”). The Advanced Encryption Standard (AES) is a symmetric block cipher chosen by the U.S. government to protect classified information and is implemented in software and hardware throughout the world to encrypt sensitive data [1]. That being said, AES gives us a nice, secure way to keep login credential information extremely safe, private, and therefore, harder to steal or tamper with. In this paper, I describe the way in which AES encryption will be implemented in the application “AccessApp”. AccessApp is a nice way in which all of ones Entertainment and Social Media apps can be stored in one place, along with all necessary credential information to login and use these apps from one singular place.

1. Introduction

In our current generation, our society essentially revolves around technology and social media. Whether you are in your teens, 20s, 30s, or even older, the chance that you use some piece of technology or have some kind of social media profile is very high, it’s no surprise. Not only is it expected that a person has at least one social media profile, but if they have at least one profile, then it is quite likely that they have many.

Think about all of the different entertainment and social media apps you use on your cellphones: Instagram, Facebook, Snapchat, Twitter, Netflix, Hulu, Spotify, etc. Now think about the hassle of having to open each application separately and type in your username and password to log in to each app separately. Why not avoid this hassle and make a safe and easy way to access and use these apps all in one place?

AccessApp serves as this "Entertainment/ Social Media" consolidation app. This one application basically "interfaces" with all of those other previously mentioned apps, serving as a single point where you can log in and get to ANY of those other entertainment/ social media apps mentioned, without having to separately open and close a bunch of different ones.

This application is also a safe one in terms of data privacy. AccessApp essentially keeps a "credentials" manager for all of your accounts in the listed apps, so that you can just open up/access the intended app from the main consolidation app. In other words, it keeps your log-in credentials saved in one place (database) so AccessApp can log you in to whatever app it is you want to use. Now that we have all of this private and sensitive information saved in one vulnerable place, how is it that we keep it safe? This is where AES comes into play.

2. Background and Plan

When I was initially given this project, I was very excited. I knew it would be a lot of work, but I was very excited to be able to write an app of my own because I have never

written an application before and the topic has always interested me very much. That being said, when I began the project, I almost did not even know where to start. I started off by doing a *lot* of research. Research regarding what languages are the best to write an app in, what platforms/IDEs to use, and how to even get *started*.

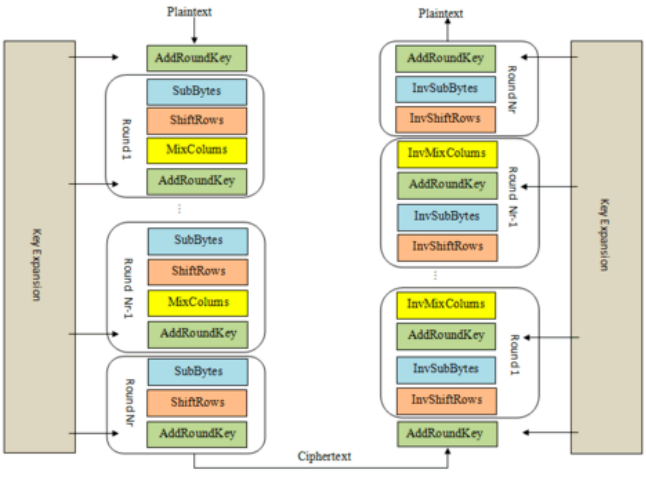
After I covered the basics of the topic and started researching IDEs and platforms to work with, I came across something called "Codename One". Codename One seemed perfect for a beginner in writing apps, like me, because really all I had to do was create the code inside of the application that would encrypt and decrypt the information from the user and do basic UI design dealing with images, buttons, and text to be displayed, without having to worry about all of the backend code that needs to be created and managed to make an application. In addition, Codename One allowed for creating apps across different platforms, such as Android *and* iOS. However, after working with Codename One for a while, I decided to not continue with it because using/accessing databases with Codename One was not an easy thing to do, and using a database within AccessApp is a *huge* part of the project. After trying to make a Postgres connection within the code and it was not working, I did some more research and, as per my knowledge, I found out that the only database that I could use within the app was SQLite. Then, after doing some research about SQLite and trying to use that, even *that* wasn't working correctly, so I started trying to work with a different IDE. Therefore, all of my work in designing the UI in the Codename One code went to waste.

Next, I tried opening NetBeans and seeing how I liked using that. I wasn't a huge fan of the UI development environment in the IDE and when I looked online at what other people had to say about it, it didn't seem like the best environment to me. Not only that, but I would have had to develop all of the backend code to create an application myself and I am not an expert in doing that since this is the first time I have ever tried to create an application, so I didn't think it would be best for me because I wanted it to look as nice as I could.

Eventually, after exhausting a couple of other options, I finally decided to choose to work with Android Studio. Although Android Studio only allows us to develop applications for an Android device, I liked the UI development environment and by the time I actually decided to work with Android Studio, there wasn't a lot more time for me to work on my project before I had to submit it, so I needed to stop wasting time, pick an environment, stick with it, and figure it out.

3. Methodology

The initial intended implementation of AES in this application was to be used to keep login credential information safe and secure. When users would initially enter their usernames and passwords into the system for



a certain social media/entertainment app, AES would be used to encrypt this information and store it safely within a Postgres database through AccessApp.

As seen in the diagram above, the program code will take in the user's username, for example, as plaintext, and perform the necessary steps following it in order to fully encrypt the sensitive information. Once the information is encrypted, it can now be saved in the database without worrying that someone can easily hack in and access and/or tamper with this information. It's hard to do anything productive (as a hacker) with gibberish login credentials; it would virtually be meaningless information to them.

Once the user's login credentials are entered and stored, my intent was to store them in the Postgres database that I had created. However, I ran out of time and was having issues in doing this, so right now all I have is a UI with an initial login screen, a screen listing the apps that you can use, and a registration page. The login screen has an email entry box, password entry box, and two buttons that either sign you into your account, or register you with a new account. The screen that it takes you to once you log in has pictures of apps that will be links to the apps themselves. Finally, the registration page has text boxes that allow you to fill in certain information to create your account.

I initially hoped I would be able to get all of this working for at least one application, however, I was not able to attain my goal in doing that, and I hope to continue working on this application this summer because I am actually very intrigued and think it could be a very cool application for people to use. So although it is not complete now for final project submission, you may be able to see the finished product some day.

4. Experiments/Progress

For final submission, I made progress in creating a UI for the application using Android Studio. As mentioned in the previous section, I created three screens for the application, which is all I planned to do for now. The one thing I still intended to add as a feature in the login page was autofill for a user's account so that they can open the app and just hit sign in to access the account. In the future, when I add more apps into the second screen, I want to make it a scrollable screen so that it *is* possible to have more applications in there.

In addition to the UI created in Android Studio, I wrote code that can be used to encrypt the user credentials which will be entered into the database, as well as the code that will decrypt those credentials when we need to take the information from the database and enter it into the application.

Then, I created a database using Postgres that stores a unique userid (number), email address, and password for the intended user. Within that, there are more tables that link a user by its userid and stores username and password information based on the application. For example, if a user entered his/her information for the Spotify app, that information would be entered into the table for Spotify accounts based on the user's userid. Then, anytime he/she wants to use Spotify, the information will be readily available, pulled out, decrypted, entered into the login page for Spotify, and he/she will be brought to his/her own Spotify page without having to deal with the hassle of logging in themselves.

5. Discussion/Analysis

Although I didn't finish implementing the application, when I finished, I planned on using a multitude of my social media accounts as well as my friends accounts to see if the information encrypted and decrypted correctly when it was entered into and pulled out of the database, I was going to check and see if when the credentials were initially entered that the data still remained accessible, I wanted to ensure that multiple accounts of the same name and password could not be created, and test to make sure that when a user logged into his or her account that it brought them to the correct account without accidentally accessing another user's information, etc. Obviously I would include a few more minor test cases, but those are some of the main and most important ones that need to be tested before this goes into full production mode.

DES vs AES

	DES	AES
Date	1976	1999
Block size	64	128
Key length	56	128, 192, 256
Number of rounds	16	9,11,13
Encryption primitives	Substitution, permutation	Substitution, shift, bit mixing
Cryptographic primitives	Confusion, diffusion	Confusion, diffusion
Design	Open	Open
Design rationale	Closed	Open
Selection process	Secret	Secret, but accept open public comment
Source	IBM, enhanced by NSA	Independent cryptographers

6. Conclusion

In this paper I described the new application AccessApp and its intentions to use AES for encrypting credentials. I further explained my total progress for this final project, including what has been done so far, and what still needs to be done to get the app fully up and running. Although the application is not complete and still has a good amount of work to be done, when it is finally finished I think it will be a very useful and cool application for people to use, especially in this new generation. When doing research, and just from my own personal knowledge about applications, I do not think there is another app that does what AccessApp will do, so it is something new that will hopefully spark peoples interest. I also believe that using AES to encrypt this type of login credential information is a safe and secure way to make sure it is not stolen or tampered with. Therefore, I think that this will make it even more enticing for avid users of multiple social media accounts to take advantage of.

7. References

[1] M. Rouse, M. Cobb, GEM100, and B. Pawliw, "Advanced Encryption Standard (AES)", https://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard