Pass a Note, Peek a Note - An AES Encrypted Messaging and Hacking Simulation System

Brandon Litwin
Department of Computer Science, School of Computer Science and Mathematics
Marist College, 3399 North Road, Poughkeepsie, New York 12601

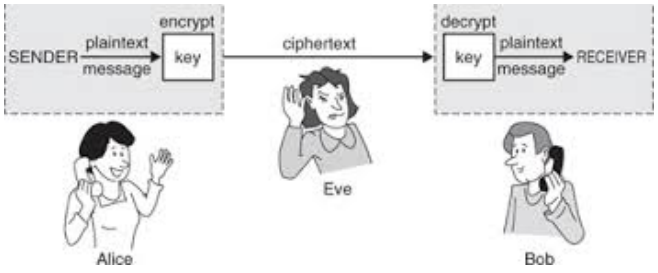

Fig. 1: Alice wants to pass a note to Bob, but Eve is trying to peek!

Abstract—*AES is one of the best and most popular symmetric encryption algorithms designed to protect data. In this paper, I discuss the design of my Flask web page that implements AES in order to allow users to send encrypted messages to each other. In addition, encrypted messages will be sent to a third-party who selects the recipient to hack. It is the hacker's job to read the user's profile and extract their password from their personal information, which acts as an example of why users should complicate their passwords.*

Keywords: AES, symmetric encryption, cybersecurity, password cracking

1. Introduction

Advanced Encryption Standard (AES) is a symmetric encryption algorithm that became a U.S. federal government standard in 2002. It has been praised for its efficiency and ease of implementation. Of course, a chain is only as strong as its weakest link. It does not matter how many bits an encryption key is or how many rounds an algorithm uses to generate its keys if a system has another vulnerability that is easily exploitable.

My motivation comes from the simple idea of Alice and Bob, the cryptography couple, and Eve, their nosy neighbor. An example of this is depicted in Fig. 1. The idea of two people wanting to pass a message to each other and an unwanted third-party secretly trying to take a peek is the basis of all encryption algorithms, and is why my project is simply called Pass a Note, Peek a Note, or PaNPaN.

In this paper, I will discuss my idea of an AES encrypted messaging system that is vulnerable to hacking. By demonstrating this system, I believe it will be clear how important defense in depth is and why we should use passwords that are not easy to crack from publicly available information.

2. Background and Other Work

People have a history of using incredibly simple passwords. Passwords created by people are typically components of dictionary words and numbers that have a certain personal connection to them [1]. Most account creation systems encourage password complication by requiring a minimum length as well as a special character and number. People also follow simple rules to try and complicate them including, in order of frequency, concatenation, replacement, spelling mistake, and insertion [1]. If we can understand the process by which people create passwords, we can more easily determine how to crack them.

3. Methodologies

My system does not have people create passwords to protect their messages. Instead, it generates a password based on words found in the user's profile page. The generated password is a simple lowercase concatenation of these words. This is done to simulate an incredibly simple password that someone may create based on their personal information.

The general use case of PaNPaN is as follows:

1) User creates an account.
2) User writes some personal information in the "About Me" section of their profile.
3) User sends a message to another user, which is AES encrypted.
4) User chooses another user to hack and becomes a Hacker.
5) Hacker receives all the messages from the user they chose to hack.
6) Hacker attempts to brute-force the decryption password by using that user's public profile information.
7) Upon successful hack, the victim is notified and their name is displayed on the Hall of Shame.
8) The victim must edit their profile and generate a new password.

By gamifying the hacking process, hackers are encouraged to get creative in their password cracking. The system is built in such a way that it is very difficult for a user to not get hacked because they have very little interference in the way their passwords are generated. The point is to demonstrate that no matter how great an encryption algorithm like AES is, it means very little if there is another weak point in the system.

4. Milestone Progress

As of the milestone deadline, I have created most of the basic features of the website. This includes the account creation, profile editing, message sending with AES encryption, and hackers receiving their victim's messages and entering a password to decrypt it. However, I have not yet developed the password generation system based on the user's information, so clicking "Generate Password" will always generate the password "testpass".

5. Future Improvements

The final version will have a simple password generation system that works as I have described in the Methodologies section. In addition, the Hall of Shame will be added with a running count of how many times someone has been hacked. Also, the user does not yet get notified when the hacking attempt is successful.

6. Conclusions

The Python code for this project has been made available at:

`github.com/brandonlitwin/mscs630litwin.git`

References

[1] Jakobsson, Markus, and Mayank Dhiman. *"The benefits of understanding passwords."* Mobile Authentication. Springer, New York, NY, 2013. 5-24.