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

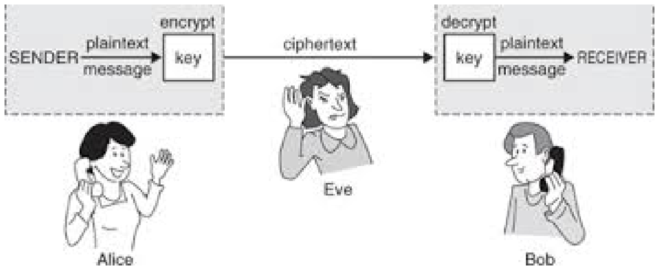


Fig. 1: Alice wants to pass a note to Bob, but Eve is trying to peek!

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.*

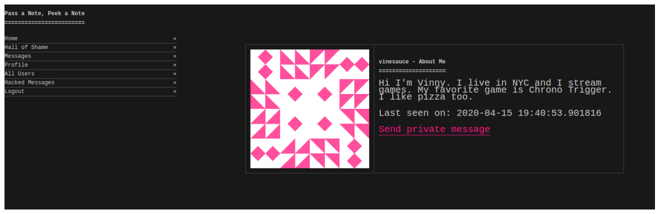
Keywords: AES, symmetric encryption, cybersecurity, password cracking

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.



vinesauce - About Me

Hi I'm Vinny. I live in NYC and I stream games. My favorite game is Chrono Trigger. I like pizza too.

Last seen on: 2020-04-15 19:40:53.901816

Fig. 2: A sample user profile.

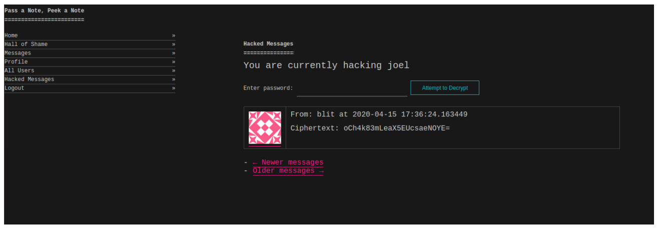
2. Background and Other Work

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. Methodology

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. A sample user profile is shown in Fig 2. The user is told that their profile must contain at least three words that are at least four characters long. This is to ensure that the password is at least twelve characters long so it is not incredibly trivial to guess. The generated password is a simple lowercase concatenation of a random selection of these words. This is done to simulate an incredibly simple password that someone may create based on their personal information.

During a hack, a hacker user will read through their chosen victim's profile. Since the rules of password generation are made public, the hacker understands that the password will be a combination of three words in that profile that are at least four characters long. When the hacker inputs the correct



Hacked Messages

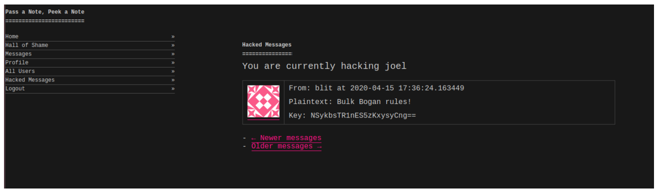
You are currently hacking joel

Enter password:

From: blit at 2020-04-15 17:36:24.163449

Ciphertext: oCh4k83mLeaX5EUcsaeNOYE=

Fig. 3: Hacker's view before the hack.



Hacked Messages

You are currently hacking joel

From: blit at 2020-04-15 17:36:24.163449

Plaintext: Bulk Bogan rules!

Key: NSykbsTR1nES5zKxysyCng==

Fig. 4: Hacker's view after the hack.

password, the user's message is displayed in plaintext, as well as the key that was used to decrypt the message. The hacker's view of the messages before a successful hack is shown in Fig. 3 and the view after a successful hack is shown in Fig. 4.

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 clicks the password generation button to create a password to protect their messages.
4) User sends a message to another user, which is AES encrypted.
5) User chooses another user to hack and becomes a Hacker.
6) Hacker receives all the messages from the user they chose to hack.
7) Hacker attempts to brute-force the decryption password by using that user's public profile information.
8) Upon successful hack, the victim is notified and their name is displayed on the Hall of Shame.
9) 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 influence 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. AES Implementation

The actual AES encryption is handled by the Python library Pycryptodome. When a user's message is sent, the message text is passed into an instance of an AES cipher from Pycryptodome's library. The AES class is able to take in a parameter to determine the mode of cipher to use. For this project, I chose to use CFB (Cipher FeedBack). As described in Pycryptodome's documentation, CFB mode performs an XOR operation from each byte of plaintext to a byte of the keystream. This mode was chosen because it makes it easy to pass in a message of any length without having to manually pad it to become a modulo of 16 bytes as a block cipher would require. The key for encryption is a randomly generated string of 16 bytes. An IV is generated through the Pycryptodome AES class's default iv method, which is later used for decryption. Next, a decryption cipher is created using the default decrypt method with the same key that was used for encryption and the IV from the previous step. The ciphertext is then passed into the decryption cipher to revert it back to plaintext. Both the ciphertext and plaintext of the message are stored into the database. This is done to save loading time of the web page during the actual hacking process. When a user's hack is successful, the page simply loads the plaintext message from the database and overwrites the encrypted text on the page.

5. Results

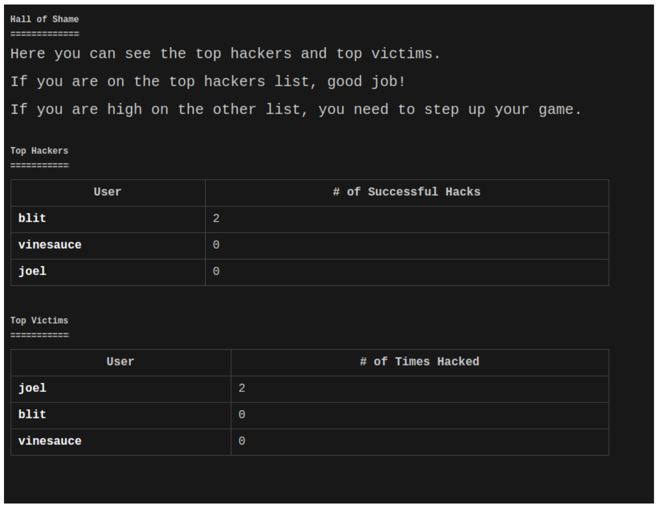
I was able to achieve all of the goals I set for this project. I have created a system that allows users to send AES encrypted messages to each other. The system allows each user to select another user that they wish to hack. The hacking works as described in the previous section, and the victim of the hack is notified to update their profile and generate a new password. Finally, I implemented the Hall of Shame page which displays the most successful hackers and the most attacked victims. The Hall of Shame is shown in Fig. 5.

6. Conclusion

In this paper I discussed my motivation for Pass a Note, Peek a Note, the methodology by which I would create the system, and the results of the final implementation. I believe my system can be a fun and engaging way for those who are unfamiliar with hacking methods to get an introductory experience to basic password cracking. It also serves as a lesson in defense-in-depth. The well known cybersecurity principle, that a system is only as strong as its weakest length, is demonstrated here.

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

`github.com/brandonlitwin/mscs630litwin.git`



```
Hall of Shame
=============
Here you can see the top hackers and top victims.
If you are on the top hackers list, good job!
If you are high on the other list, you need to step up your game.

Top Hackers
===========
```

User	# of Successful Hacks
blit	2
vinesauce	0
joel	0

```
Top Victims
===========
```

User	# of Times Hacked
joel	2
blit	0
vinesauce	0

Fig. 5: You want to be among the top hackers, not the top victims.

References

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