Travis Dennis

CS 480

Lab 3

Winter 2019

What to do:

Part I: Read this post and then answer the questions: how can we prevent deriving of the corresponding private key from a public key? (2 points)

https://stackoverflow.com/questions/50053884/use-rsa-public-key-to-generate-private-key-in-openssl

Response:

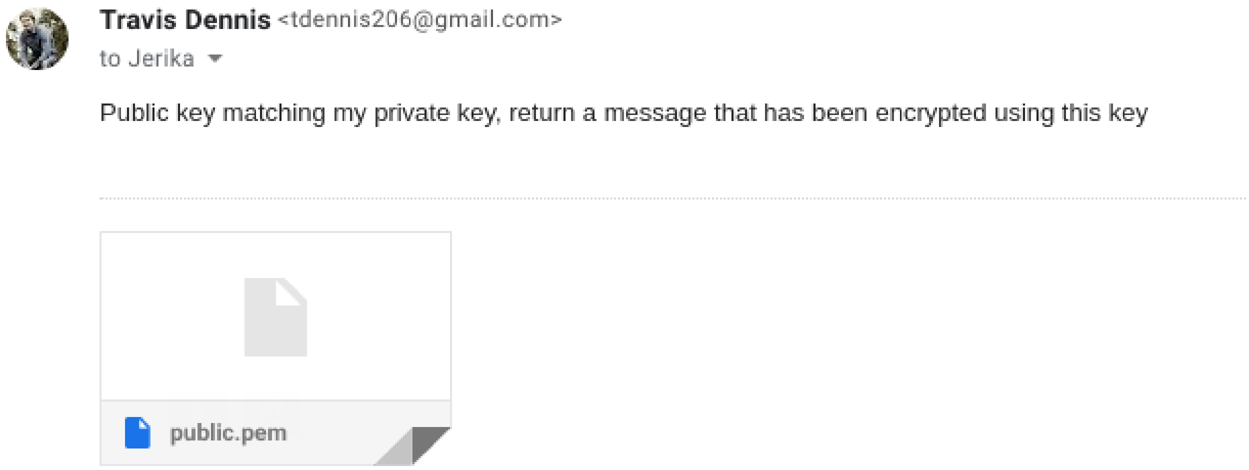
The first problem I noticed was that even though they are using 1024 bit integers for the RSA encryption, the first of the primary numbers for the encryption was relatively small. The prime number here was only 66 bits long in base two, making it very easy to brute force. From there it is a matter of using the right formulas to discover the rest of the operation. Finding the modulus from the hex provided and then the exponent would be much harder to factor given a more closer matched prime couple.

Part II: Public and private key lab (5 points)

There are a lot of ways to generate a pair of public key/private key pair as long as your do not want to register them for public use. For example, you can open an account for Bitcoin to receive your public key and private key pair (I assume, have not done that myself). **Putty** is something I used in the past for this but you have to convert the public key to a format readable by openssl rsautl. This URL has that https://superuser.com/questions/576506/how-to-use-ssh-rsa-public-key-to-encrypt-a-text. This discussion uses Python https://stackoverflow.com/questions/30056762/rsa-encryption-and-decryption-in-python (this is the simplest, seems to me)

Doing things in the following steps:

1. Find a team member from the class. We have 26 students, all need to a member. That gives us 13 groups of two students each.
2. Each student needs to generate a public key/private key pair; pass the public key to your team member; you team member encrypts a message, say. **CS 480 is the best class ever**; pass the encrypted message to you; you decrypt the message.

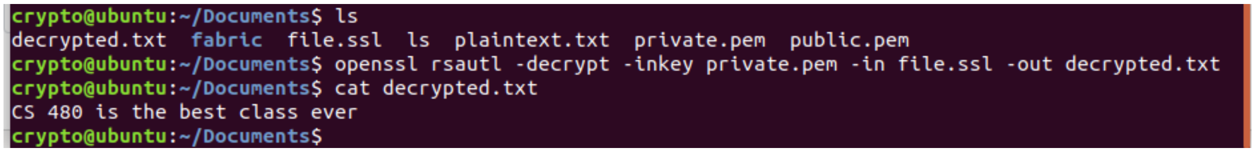


Travis Dennis <tdennis206@gmail.com>
to Jerika

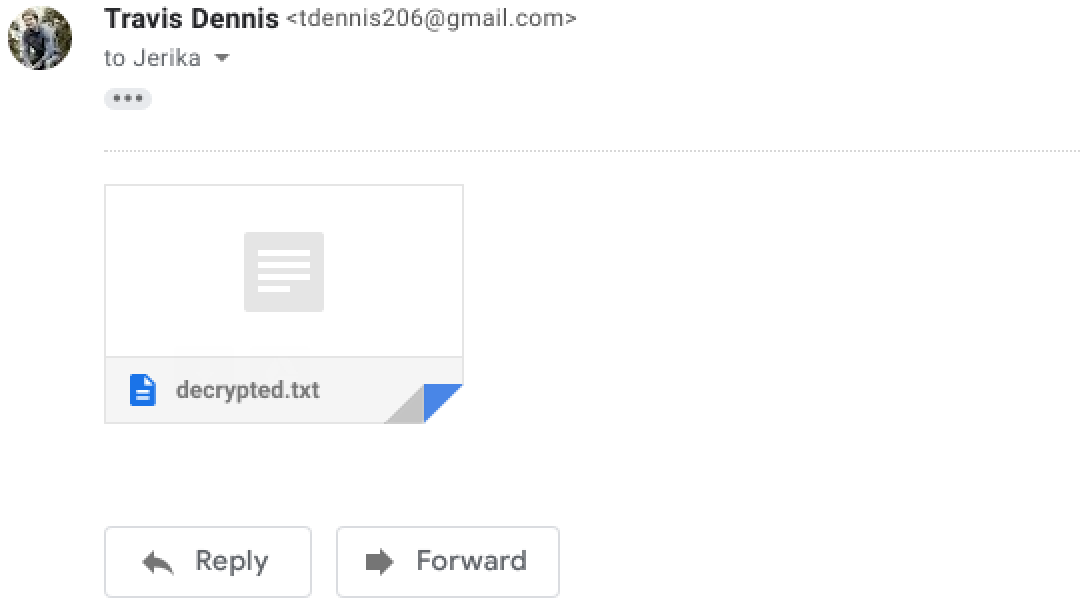
Public key matching my private key, return a message that has been encrypted using this key

public.pem

3. Pass the decrypted message back to your team member and make sure the decrypting process works.



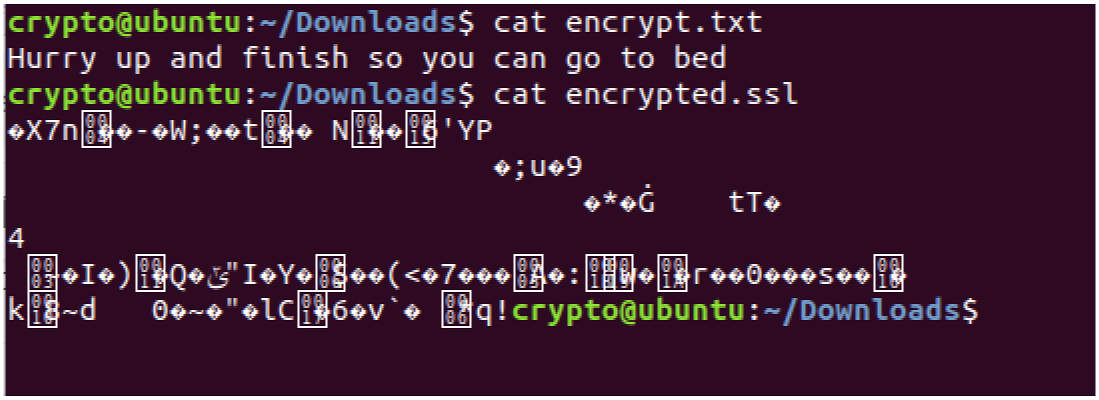
```
crypto@ubuntu:~/Documents$ ls
decrypted.txt  fabric  file.ssl  ls  plaintext.txt  private.pem  public.pem
crypto@ubuntu:~/Documents$ openssl rsautl -decrypt -inkey private.pem -in file.ssl -out decrypted.txt
crypto@ubuntu:~/Documents$ cat decrypted.txt
CS 480 is the best class ever
crypto@ubuntu:~/Documents$
```



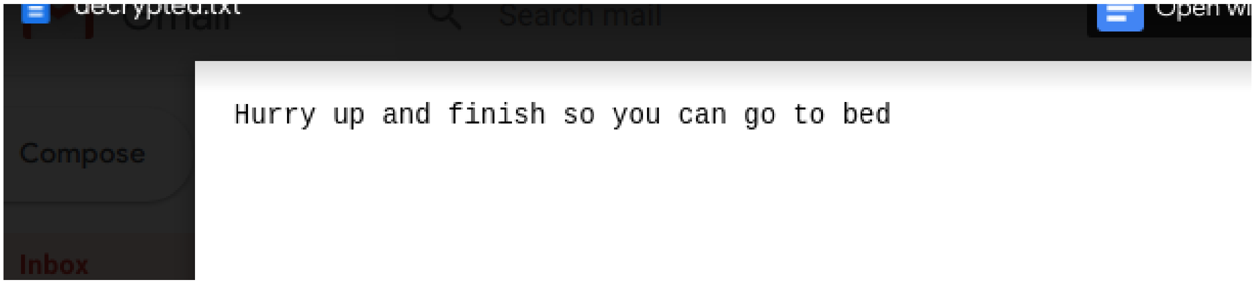
Travis Dennis <tdennis206@gmail.com>
to Jerika

decrypted.txt

4. Ask your teammate to send you an email either confirms or denies that the decrypting process works.
5. Reverse the process with your teammate for steps 2~4.



```
crypto@ubuntu:~/Downloads$ cat encrypt.txt
Hurry up and finish so you can go to bed
crypto@ubuntu:~/Downloads$ cat encrypted.ssl
�X7n��-�W;��t�� N���'YP
                           �;u�9
                                �*�Ġ    tT�
4
 ��I�)�Q�ڪ"I�Y���(<�7����:���r��0����s���
k�~d   0�~�"�lC�6�v`�  �q!crypto@ubuntu:~/Downloads$
```



```
Hurry up and finish so you can go to bed
```

https://www.devco.net/archives/2006/02/13/public_-_private_key_encryption_using_openssl.php

Part III: Hash (3 points)

I will provide a text file. You need to find an implementation of SHA-256 (SHA 2 implementation) and hash the content of the text file. You are allowed to work with the team member of Part II to make sure your hash function generates the same output as the correct answer. There are a lot of great sources for SHA-256. I know SQL Server can do that. Python can also do that.

Raw File:

SHA256: `8feda9c008725bee8fdc08a028251706bdc0b54c586a6b1cef791057edd44c4c`

`Line endings (whitespace) removed`

`SHA256: 24dddb5cd3b3981d39fe532e663fe4f033907dc61c7ecba4f1211db534fb7800`

https://defuse.ca/checksums.htm#checksums lists all of the checksums for a given file up to 5mb

What to turn in:

Part I – your answer to my question. Make it short.

Part II – a description of the process and the mail from your teammate that either confirms or denies that the decrypting process works. I may call a few team to give a demo, so please be prepared for that.

Part III – a description of the where you find the function and the result of the hash.

When to turn everything in:

3/11/2019

Incentives of doing this:

I'll add the points to your quiz scores. Even better, if you turn your lab in by 3/8/2019, I'll add 1 point if you score 8 or more points – meaning you know you are done.