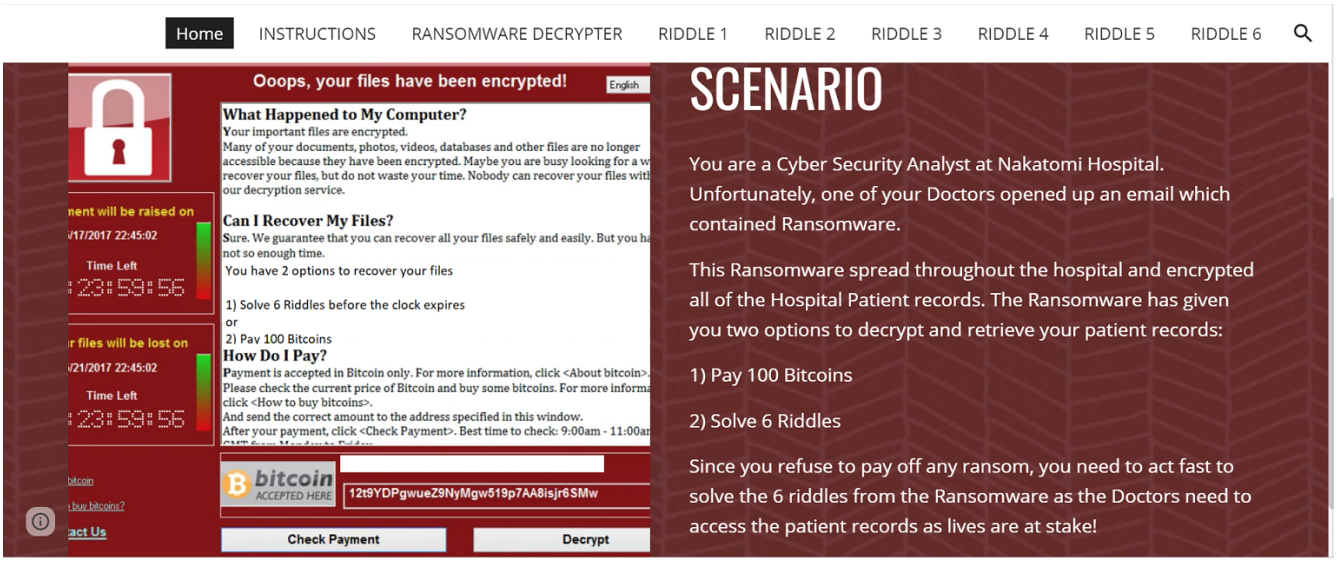
HOME PAGE



Instructions:

RIDDLE 1:

Roses are Red Violets are Blue,
Caesar would be 8 is your first clue.

Decrypt ozcjmz and enter it below,
and maybe a key then might just show.

Answer: gruber

This was a fun one to start off with. It's a caesar cypher with a shift of 8. Doing it by hand is one way or using a decoder to get the answer.

Riddle 1

Congrats, you have solved the first riddle, Your first key is: 6skd8s

RIDDLE 2:

Humpty Dumpty Sat on the Wall,
Humpty Dumpty had a great Fall,

All the king's Horses and all the
Kings Men couldn't decode this
message for him:

01000111 01100101 01101110
01101110 01100101 01110010
01101111

Looks like we have some binary here, let's plug this into a binary decoder.

```
01000111 01100101 01101110 01101110 01100101 01110010 01101111
```

Output time: 0ms length: 7 lines: 1

```
Gennero
```

Gennero, let's plug this in into the key recorder.

RIDDLE 2

Congrats for solving the second riddle, the key is: cy8snd2

I'm a little Cipher,
short and sweet.

Here is my vector,
and also my key

When I get all steamed up,
hear me shout!

RIDDLE 3:

Cipher Text:

4qMOIvwEGXzvkMvRE2bNbg

Key:

5284A3B154D99487D9D8D8508461A478C7BEB67081A64AD9A15147906E8E8564

IV (Initialization Vector):

1907C5E255F7FC9A6B47B0E789847AED

OpenSSL Options:

-pbkdf2

-nosalt

-aes-256-cbc

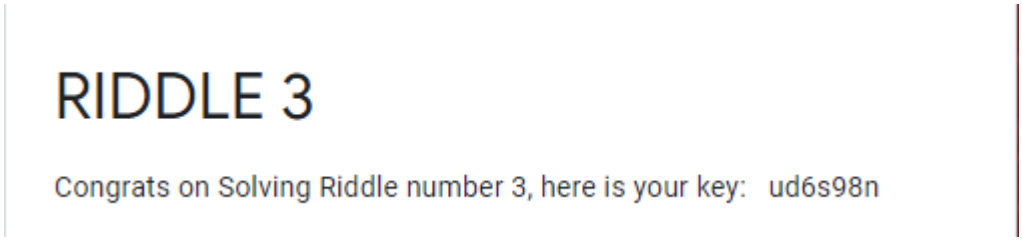
base64

Yuck, let's stick the information we need into a nice clean .txt file so that we can decode it. We'll call the .txt file "rid3.txt" for riddle 3 and place our given cipher text of 4qMOIvwEGXzvkMvRE2bNbg **in there.**

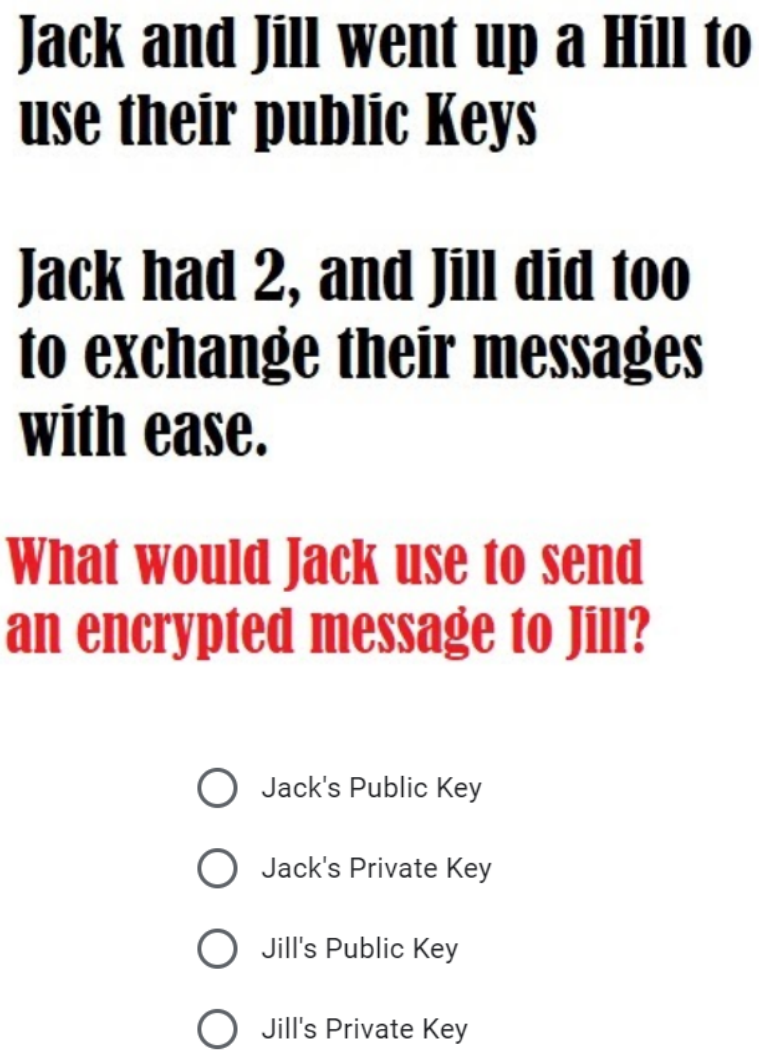
`rid3.txt` **Here it is with our cipher all loaded in. Now to use openSSL to decode it.**

```
┌──(kali㉿kali)-[~/…/ucsd-sd-virt-cyber-pt-09-2021-u-c/2-Homework/10-Cryptography/resources]
└─$ openssl enc -pbkdf2 -nosalt -aes-256-cbc -d -in rid3.txt -base64 -K 5284A3B154D99487D9D8D85
08461A478C7BEB67081A64AD9A15147906E8E8564 -iv 1907C5E255F7FC9A6B47B0E789847AED
takagi
```

Here's our line with the necessary decoding parameters, "-d" for decode, our rid3.txt and our Base64, our key with the "-K" flag and our initialization vector with "-iv". The result is "takagi" which gets us our next key!



RIDDLE 4:



So Jack would use Jill's public key to encrypt the message so that Jill can use her private key to decrypt the message. So the third answer from the top.

What would Jill use to to decrypt Jacks message? *

- ◯ Jack's Public Key
- ◯ Jack's Private Key
- ◯ Jill's Public Key
- ◯ Jill's Private Key

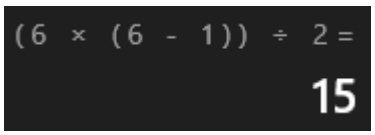
As mentioned above, Jill will use her private key to decrypt the message. Fourth answer from the top.

Jack and Jill invited Bob, Alice, Tim and Peter along to exchange some messages. How many keys would they all need for asymmetric vs

Tim just sent an encrypted message to one of his friends, which of the following keys did he likely use to encrypt the message *

- ◯ Tim's Public Key
- ◯ Alice's Public Key
- ◯ Peter's Private Key
- ◯ Tim's Private Key
- ◯ Bob's Private Key

It looks likes there was an error copying the third question over, but it looks like the number of asymmetric and symmetric keys are needed for 6 people in total. Symmetric keys are calculated as (N x [N-1]) \ 2 with "N" being the number of people keys are needed for.

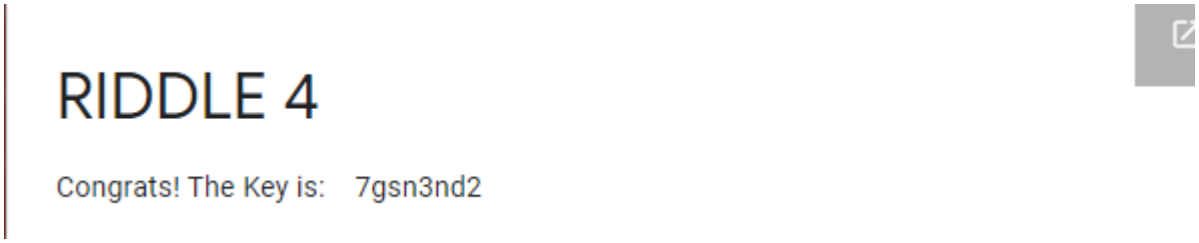


So 15 symmetric keys will be needed.

Asymmetric is easier, it's just N x 2 with "N" also being the number of people. So 6x2 = 12. 12 Asymmetric keys are needed.

Finally, Tim's sending an encrypted message to someone so he needs to encrypt using *that* person's public key. So Alice's public key is the only option since Tim cannot use his own or the other person won't be able to decrypt the message!

Anyhoo, here's the next key:



RIDDLE 5:

Hey diddle diddle,
the cat and the fiddle,
The cow jumped over the moon.

The little dog laughed
when it found this MD5 hash,

Hash:

3b75cdd826a16f5bba0076690f644dc7

An MD5 hash can be run through hashcat very simply. We first need to get the hash into a usable text file which we'll call "mash.txt" for

```
┌──(kali㉿kali)-[~/…/ucsd-sd-virt
└─$ cat mash.txt
3b75cdd826a16f5bba0076690f644dc7
```

There it is. Now to run it through hashcat.

```
┌──(kali㉿kali)-[~/…/ucsd-sd-virt-cyber-pt-09-2021-u-c/2-Homework/10-Cryptogr
└─$ hashcat -m 0 -a 0 -o mash2.txt mash.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.1.1) starting ...
```

So we run the "hashcat" command with flag "-m" for mode which is MD5 specified as "0", "-a" for attack method which we specify as "0" for a dictionary attack, "-o" for output the new filename of the result which we call "mash2.txt", and finally the list of passwords from rockyou.txt to check against.

```
└─$ cat mash2.txt
3b75cdd826a16f5bba0076690f644dc7:argyle
```

If we read the completed file, we find argyle as our next passphrase.

RIDDLE 5

Congrats on solving Riddle number 5, Here is your key: ajy39d2

There's our next key.

RIDDLE 6:

Mary had a secret code,
Hidden in a photo,
And everywhere that photo went,
The code was sure to go

She wrote the passphrase on the
book, to access the code
You just need to use some stego
tricks and the secret will be showed.

Image Link:
https://drive.google.com/file/d/1m9ykscnTGzgkkVet9wmiBCYsbhzbrKR9/view

This is where we do a little steghide command action.

```
└─$ steghide extract -sf mary-lamb.jpg
Enter passphrase:
wrote extracted data to "code_is_inside_this_file.txt".
```

Our passphrase is written in the book as “ABC” . The result is a txt file we can read.

```
└─$ cat code_is_inside_this_file.txt
mcclane
```

Which we type in to get

RIDDLE 6

Congrats on solving Riddle number 6, they key is: 7skahd6. Now go and enter in all of your keys into the Ransomware decrypter!!

RANSOMWARE DECRYPTER

Congratulations! You have decrypted the Ransomware! All the Nakatomi Hospital Records are now Decrypted! Please take a screenshot of this message and submit as your homework!

That’s everything!