# SVGS Computer Security, Cryptology Quiz

My motive in writing this quiz is simple, to get students to review the slides. This test is open book. The questions are in the same order that the slides are in.

1. Why did WEP fail?
2. What is wrong with this statement? “I used base64 encryption to hide my data.”
3. How is the modular inverse of a number usually calculated?
4. What is the modular inverse of 5 mod 13 ? You can use Python or a calculator
5. What is the most important thing to remember about an IV or nonce?
6. What is wrong with AES-ECB mode?
7. What advantage does symmetric key encryption have over asymmetric key encryption? What disadvantage?
8. What is non-repudiation?
9. Why is RSA slow when you use it to encrypt a large file?
10. With RSA encryption, if Alice sends a message to Bob, she encrypts the message with…
11. How do computers get the large prime numbers they use in encryption? How do they know they are really prime?
12. Why is a good PRNG important in cryptography?
13. What is the most popular Key exchange method for HTTPS? For SSH? (Diffie-Hellman slides may be helpful.)
14. Is Elliptic Curve Cryptography more closely related to Diffie-Hellman or to RSA cryptography?
15. What are the public portions of a key exchange using Elliptic Curve Cryptography?
16. What is the primary difference between a digital signature and a MAC (Message Authentication Code, not MAC address)?
17. What are the two properties a cryptographic hash is supposed to have?
18. A Digital Certificate is used to distribute a …
19. Why is it important to know where a public key really came from?
20. When a browser uses HTTPS, how does it know it is talking to the correct server and not an imposter?
21. A browser and a server agree on a cipher suite instead of just saying ‘TLS’. What are the different protocols in the suite used for?
22. In TLS the network traffic is encrypted, and the school’s/company’s IPS cannot inspect the data to see if it is malware. What information is available (at least in TLS v1.2 and below) that can give some clues to network security personnel?
23. What is a key exchange protocol that may be used for ‘Perfect’ Forward Secrecy? What protocol may not be used?

# **Practical**

## **Simple AES Problem**

Yasmin has intercepted email traffic between the Master and one of their agents.  **See if you can decrypt the message**.

Master:  Agent 4, execute instructions contained in the attached message [MastersOrder1563.b64](https://learn.vccs.edu/courses/314236/files/92777907/preview).  As usual, the message is encrypted with AES in ECB mode and then Base64 encoded (The Master is too busy to deal with nonces and tags.)  Use the 42nd entry in your book of keys.

Master:  Agent 4, you have not executed my order.  Why?

Agent 4:  Uh, I can’t find my book of keys.

Master:  You idiot!  Look harder!

Agent 4:  Sir, the book of keys was accidentally burned.

Master:  Accidentally!?  By whom!?  Idiot!  I’ll get you the key.

Master:  Dual hearts rule

Instructor note:  In case you can’t find the file [MastersOrder1563.b64](https://learn.vccs.edu/courses/314236/files/92777907/preview), here’s the content

TGFj6ocA5LRj7lFX23zZLTJVZwIJ9ecPDeJ3zt98m8gZGlf0gzuHFRYXrqwOD+TdLbbx5GsUhl3m\nLLQBosBVwXGTOMi5PPaj6QkjLgukWAQcKAFRlFTlj00m7dgB2r+KCLF6i6PWrRU5jvHynL5vy1ut\nWCOtu70kG0wRpXIP4DxdJf2vOkrkJzJ5gq+5dim4Lungicy9LhN03v2WNHiz+q8zaEawXBktXQp+\ndw5nYgX3vLBDbKJ4H6bkSzCBvKScYXxBwNVQJFJABnRsBWtArA==\n

Note:  Hints are base64 encoded instead of ROT13 as I said in class.  ROT13 didn’t handle some of the characters I used.  Easy place to decode base64:  
[https://www.base64decode.org/ (Links to an external site.)](https://www.base64decode.org/)

Hint 1:   
VGhpcyBzaG91bGQgYmUgYWxtb3N0IGlkZW50aWNhbCB0byB0aGUgQUVTLk1PREVfRUNCIGV4YW1wbGUgaW4gQ3J5cHRvIEhvbWV3b3JrIDQu

Hint 2:  
SWYgeW91IGFyZSBwYXN0aW5nIHRoZSBiYXNlNjQgaW50byB5b3VyIFB5dGhvbiwgbWFrZSBzdXJlIGl0IGlzIG9uZSBsb25nIHN0cmluZyB3aXRoIG5vIGNhcnJpYWdlIHJldHVybnMvbGluZSBmZWVkcy4=

Hint 3:  
Y29kZWNzIGFuZCBBRVMgYm90aCB3YW50IGlucHV0IG9mIHR5cGUgYnl0ZXMuICBVc2Ugc29tZXN0cmluZy5lbmNvZGUoKSB0byBjaGFuZ2Ugc3RyaW5nIHRvIGJ5dGVzLiAgVXNlIHNvbWVieXRlcy5kZWNvZGUoKSB0byBjaGFuZ2UgYnl0ZXMgdG8gc3RyaW5nLg==

## **Hash Problem**

Whilst on the Cyberman spaceship, Graham comes across a terminal.  A sticky note near the terminal says, “The credentials for the next account are in file.”  Graham lists the files in the terminal and sees 100 files, file0.txt through file99.txt.  He looks back at the note to see which file he needs, but the sticky note was torn off just after “file”.

Help Graham find the file he needs.  A simple loop that takes the hash of each file is one way to do it.  It does not matter which form of hash (MD5, SHA-1, etc.) you use, as we do not expect malicious files that create hash collisions.  (There are other ways to solve this if you do not like hashes.)

The files are in [**hashfiles.zip**](https://learn.vccs.edu/courses/314236/files/92777927/preview).  There are 100 files, so put the zip file in its own subdirectory before you unzip.  It could make a mess.  Just sayin’.

**What is the content of the file that contains the credentials?**

Easy way to decode hints:  [https://www.base64decode.org/ (Links to an external site.)](https://www.base64decode.org/)

Python Hint 1:  It is easy to do hashes in Python.  I need to include that in Lab 8 next year, but for now:  
import hashlib  
hashlib.md5(b'this is what I want to hash').hexdigest()  
Note:  the input must be type bytes, **b**’somestring’ or string\_variable.encode()  
Note:  b'this is what I want to hash' is the content of the file, not the file name  
Note:  99 of the hashes should be identical, 1 should be

Python Hint 2  
aW1wb3J0IGhhc2hsaWIKCndpdGggb3BlbignZmlsZTAudHh0JykgYXMgZmg6CiAgICBjb250ZW50ID0gZmgucmVhZCgpLmVuY29kZSgpCm15aGFzaCA9IGhhc2hsaWIubWQ1KGNvbnRlbnQpLmhleGRpZ2VzdCgpCnByaW50KG15aGFzaCk=

Another way Hint:  
RmluZCBhIHdvcmQgaW4gdGhlIGZpcnN0IGZpbGUgdGhhdCB5b3UgZ3Vlc3MgbWlnaHQgbm90IGJlIGluIHRoZSBmaWxlIHRoYXTigJlzIGRpZmZlcmVudApVc2UgZ3JlcCB0byBmaW5kIGEgZmlsZSB0aGF0IGRvZXNu4oCZdCBoYXZlIHRoZSB3b3JkIHlvdSBjaG9zZQpUaGVyZSBpcyBhIGZsYWcgaW4gZ3JlcCBmb3Igbm90IG1hdGNoClVzZSB0aGUgLXIgZmxhZyBmb3IgcmVjdXJzaXZl

## **Cybermen Elliptic Problem**

The Doctor gets on Graham’s terminal on the Cybermen spaceship and immediately fires up Wireshark to see what traffic she can capture.  After examining several TCP streams, she finds an interesting one.

Cybermen Master:  Prepare to receive encrypted message.  Use [Cybermen Elliptic Curve Procedure](https://learn.vccs.edu/courses/314236/files/92777920/preview)[Preview the document](https://learn.vccs.edu/courses/314236/files/92777920/preview).  My public key is 17, 23.

Cyberman15:  Acknowledge.  My public key is 17, 88

Cybermen Master:  Message follows.

'+7aYffNYPs+qMlBmPs/9x1FJ+7KhmbudPronEdIWtSUPfvLBwIeEDv4QuHqjMaSWRJ+5OP6n9Ich\nOEiM6CoRD84oF8KppHAulvMNrQTsyco=\n'

[Cybermen Elliptic Curve Procedure is here](https://learn.vccs.edu/courses/314236/files/92777920/preview)[Preview the document](https://learn.vccs.edu/courses/314236/files/92777920/preview)  
[Message as a file is here](https://learn.vccs.edu/courses/314236/files/93176489/preview)

Your job is to decrypt the message (duh 😊).

Hint 1:  VGhlIEN5YmVybWVuIGhhdmUgY2hvc2VuIGEgbW9kdWx1cyAoMjExKSBhbmQgY3VydmUgdGhhdCByZXN1bHRzIGluIDIzMiBwb2ludHMuICBJIGd1ZXNzIHlvdSBjb3VsZCBnZXQgYWxsIDIzMiBwb2ludHMgZnJvbSB0aGUgY2FsY3VsYXRvciBhbmQgdGhlbiBicmVhayBvbmUgb2YgdGhlIHB1YmxpYyBrZXlzIHRvIGdldCBhIHByaXZhdGUga2V5LiAgVGhhdCB3b3VsZCBiZSBhIHJlYWwgcGFpbiwgdGhvdWdoLiAgVGhlcmUgaGFzIGdvdCB0byBiZSBhbiBlYXNpZXIgd2F5Lg==