

**Discussion:** Hopefully we all now understand how vulnerable network traffic is to eavesdropping. Even if data is not being sent over a network, laptop computers get lost or stolen, and then a hacker has all the time in the world to examine the digital contents. So we need to have a way to keep data secret.

Encryption solves our problem, mostly. It's very hard to create a completely unbreakable cipher that is "unconditionally secure," but the good news is that we often just need something that is "computationally secure." Be sure to do the reading assignments in Erickson so that you understand this terminology. The bottom line for engineering is this: encryption is a trade-off between the resources (and time) necessary to encrypt the data and the length of time that we need to protect the data. If we only need to keep a secret for a day and our cipher takes weeks to break under the best of circumstances, then it might be adequate. A more complicated cipher still works, but it might require too much computation.

Today we're going to have some fun with various ciphers and make sure that we understand the basic trade-offs between cipher complexity, key-length, and message security.

**Setup:** Open your personal Ubuntu Virtual Machine (VM) using either the VSphere Client or VMWare Viewer (if the network is having trouble) and be sure you are at a CLI with the "midshipman" user prompt. You will have sudo NOPASSWD privileges on your machine, so be careful!

**Reading:** Erickson pages 396 to page 398 (sections 0x714 and 0x720).



## Section 2: Symmetric Encryption

4. Go to your Ubuntu CLI and type the command `man gpg` and hit enter. This gives you an overview of the encryption tool that we're going to use now. Just read the summary. GPG is a very powerful program with lots of options. For now we'll be using it with simple options.
  - a. Create a simple text file using `nano message` and typing in a message of your choice. Be sure to save the file upon exit. Then type `hexdump -C message` and note what you observe. The left-most column is a count of the number of bytes. What is the middle column?
  - b. The right-most column is ASCII and should look familiar. Is your English message actually stored in the computer? What is stored, and how does the computer manage to show it to you as if it were English?
  - c. Now encrypt the file with this command: `gpg -c message` and type in a passphrase of your choosing when prompted (a passphrase is the same thing as a key in this case). You should now find a file called `message.gpg` in the current directory. Type `hexdump -C message.gpg` and note the output. What are your observations? Is the cipher text the same number of bytes as the plaintext message?
  - d. Do you think that your lab partner would be able to decrypt the message (assuming that they don't know the passphrase)?
  - e. If the goal is to keep this message secure for the class period, do you think you've done enough? Could someone decipher your message in ten years?
  - f. How might one go about cracking this encrypted message?

- g. Now decrypt the message using the command “`gpg message.gpg`” and follow the prompts. Did it work? **Note:** Do not overwrite the file, name the new one **message1**, so you can compare to the original.
- GPG also prints some results. Be sure to note them. One tells you the cipher algorithm actually used. The other tells you that the original message was not digitally signed (we’ll learn all about that next lab).
- h. GPG uses the algorithm **CAST-5** by default, but let’s just see what happens if we want to use **AES** instead. Type this:  
`gpg -c --cipher-algo AES message1`  
then hit enter and use the same passphrase. Then use  
`hexdump -C message1.gpg`  
to observe the output. How does it differ from the previous **CAST-5** encrypted version?
5. In this lab, we were actually using simple symmetric encryption. It’s called symmetric because both the sender and receiver use the exact same key to encrypt and to decrypt the message. The encryption ciphers are designed to work such that the exact same cipher and key will do both actions (encrypt and decrypt).
- What’s the fatal flaw in a symmetric scheme? (Assume we have designed an unconditionally secure cipher)
  - How might this flaw be mitigated?
  - What about a computationally secure cipher? What additional flaw does it have?
  - How is this flaw mitigated?

6. Now we're going to play a class game. When your instructor tells you to, you're going to try and pass your message to a classmate on the other side of the room. You will also have to try to get the secret key to them. Here's the catch: the winner is the team that manages to pass two messages without anyone else managing to intercept and decrypt it.
  - a. Create a secret message of your choice and put it in a file named with your alpha login, e.g. "**mXXXXXX**". Then encrypt that file with a passphrase (key) of your choice. Decide for yourself whether it's better to have a more complex or simple passphrase and give your rationale here:
  
  - b. Now copy your ciphertext file into your local webserver by typing  

```
cp -a mXXXXXX.gpg ~/work/webroot/
```

and compile and run **tinyweb.c**, just as you did in the last lab. Your ciphertext message is now available at **http://192.168.1.Y/mXXXXXX.gpg**, where Y is the last octet of your machine's IP address.
  
  - c. Now get your IP address, alpha code, and passphrase to your partner so they can download and decode your message. Try to be creative, and explain how you transferred this information here:
  
  - d. Fight's On! (yes, this lab was written by an aviator). Your mission is to get your message to the recipient safely and then try to crack as many of the other messages as possible. You cannot leave your seat, so you will need to get creative...

**Conclusion and Results:**

Your typed lab report will consist of two paragraphs, in the first paragraph:

- Briefly describe what you did in the lab in your own words.
- Discuss something new that you learned.

In the second paragraph, answer the questions:

- How could an adversary use this knowledge or these tools for malicious purposes?
- How could you use your new understanding to protect your systems and personnel from attack?

**Staple** the completed report to the back of your original lab and turn it in to your instructor at the beginning of the next class.