

Information Assurance: Homework 4 – Answers/comments

Due October 6, 2005

Submit electronic files via compass. See newsgroups for details of submitting via compass.

1. A stream of encrypted traffic has been captured, and you are tasked with breaking it. We have reason to believe the stream is encrypted using a vignere cipher, but we don't know the period or the key. We do know that the text is in English. You are each assigned a portion of the text.

Last name starts with	Attack file
A, L	http://www.cs.uiuc.edu/class/fa06/cs498sh/hw4/vig1-enc.txt
B, M	http://www.cs.uiuc.edu/class/fa06/cs498sh/hw4/vig2-enc.txt
C, N	http://www.cs.uiuc.edu/class/fa06/cs498sh/hw4/vig3-enc.txt
D, O	http://www.cs.uiuc.edu/class/fa06/cs498sh/hw4/vig4-enc.txt
E, P	http://www.cs.uiuc.edu/class/fa06/cs498sh/hw4/vig5-enc.txt
F, R	http://www.cs.uiuc.edu/class/fa06/cs498sh/hw4/vig6-enc.txt
G, S	http://www.cs.uiuc.edu/class/fa06/cs498sh/hw4/vig7-enc.txt
H, T	http://www.cs.uiuc.edu/class/fa06/cs498sh/hw4/vig8-enc.txt
J, V	http://www.cs.uiuc.edu/class/fa06/cs498sh/hw4/vig9-enc.txt
K, W	http://www.cs.uiuc.edu/class/fa06/cs498sh/hw4/vig10-enc.txt

- a. For your text, calculate the index of coincidence (IC) and look for repetitions to make an initial guess of the period length. Submit your initial guess and this evidence.
- b. Go ahead and use the techniques discussed in class to find the key characters and decipher the message. You may use automated tools such as the java applet shown in class <http://math.ucsd.edu/~crypto/java/EARLYCIPHERS/Vigenere.html>. Submit the deciphered text and the key.

Most of you used the applet to break the code, which I expected. Some people lost points for not looking for repetitions or computing the IC. Some people also forgot to list the key and lost a couple points.

2. This portion of the homework involves working with AES and DES encryption. Pull the AES reference library from <http://www.cs.uiuc.edu/class/fa06/cs498sh/hw4/aes-files.zip>. This library includes a test AES program which you can augment as necessary. It also includes a makeKey program which creates a key of the specified

length using the rand() pseudo-random function.

Compile it for your target system. I have tested this program on Windows XP last year and have tested it on Linux this year. Makefiles and project files are included. If you don't have access to a Linux or Windows system with a C compiler, let us now as soon as possible.

- a. Fetch an encrypted file and a key file from <http://www.cs.uiuc.edu/class/fa06/cs498sh/hw4/hw4-enc.bin> and <http://www.cs.uiuc.edu/class/fa06/cs498sh/hw3/key06-128.bin>. Decrypt the file using ECB mode. It should result in a plain English file. Submit the resulting plaintext.
- b. Select a file to encrypt using a key in CBC mode. Submit the encrypted file, key file, and initialization vector file.
- c. Try encrypting your file in ECB mode using different key lengths using 128 bit, 192 bit, and 256 bit keys. AES operations are very fast. You will want to use a high resolution timer such as the gettimeofday or clock_gettime system calls. In addition, you will probably need to perform your target measured operation multiple times to have something that can be measured by the clock granularity. Submit a table of the key length and the associated average encrypt time.
- d. Measure the encryption using DES. On Linux the function ebc_crypt and des_parity should do the trick. On Windows the base provider for the Crypto API should provide a DES encryption operation. I will verify that over the weekend and post details in the newsgroup. Add a row to your table from part c showing the average time for the DES encryption operation.

This was another one that people either got or didn't. There was a good deal of variance in the measurements. People who avoided measuring file I/O did get measurements that nicely grew with the key length as one would expect. For some people DES measured much faster. I'm guessing in that case, they were measuring file I/O for AES but not for DES. The times for DES should have been a little slower than AES. But that will vary with the particular implementation being measured.