# CS530
# HW2

## Bill Cheng

*http://merlot.usc.edu/cs530-s10*

# Base64

- You must use functions in OpenSSL to implement these functionalities
  - small buffer requirement applies
- Base64 encoding and decoding
  - `BIO_f_base64()`
  - `BIO_set_callback()`
    - to examine input before conversion
- OpenSSL does not have the greatest documentation on the web
  - man pages installed in `~csci551b/openssl/ssl/man`
  - you need to setup your environment properly

# DES

- Block cipher
  - encrypts 8 bytes at a time
    - must pad input file with zeroes
  - output file size is always a multiple of 8 bytes
  - during decryption, how can you tell how many bytes to keep for the last block?
    - must store size of last 8-byte block in encrypted file
    - OpenSSL does it differently

# Encrypted File Format

- **First 3 bytes must be "DES"**
- **Next byte is a number between 1 and 8 (inclusive)**
  - **number of bytes in the last block of the original file**
- **Next 20 bytes is the SHA-1 hash of the original file**
- **Ex: "Hello World\n" (12 bytes)**
  - `000000: 48 65 6c 6c 6f 20 57 6f  72 6c 64 0a -- -- -- --  Hello World.`
  - **SHA-1:** `648a6a6ffffdaa0badb23b8baf90b6168dd16b3a`
  - **encrypt this file with passphrase "yesnomaybe":**

```
000000: 44 45 53 04 64 8a 6a 6f  ff fd aa 0b ad b2 3b 8b  DES.d~jo~~~.~~;~
000010: af 90 b6 16 8d d1 6b 3a  fb c6 6a d7 c7 9a 35 cd  ~~~.~~k:~~j~~~5~
000020: ac ca da 2d 04 82 cd 70  -- -- -- -- -- -- -- --  ~~~-.~~p
```

# Encryption Secret Key

- Prompt the user for a passphrase
  - use `des_read_pw()`
    - Ex: `"yesnomaybe"`
  - calculate SHA-1 of passphrase
    - Ex: `fec42bbb66560a9d32a14207fb6d3de3e93bbdbe`
  - leading 8 bytes used as secret key
    - Ex: `fec42bbb66560a9d`
    - adjust for odd parity: `fec42aba67570b9d`
    - need to check for weak and semi-weak keys
  - next 8 bytes used as IV
    - Ex: `32a14207fb6d3de3`
  - encrypt with `DES_ncbc_encrypt()`