# Documentation for Winter2017 CSS577 Project 1

**Overview:** This program generates an output stream containing the cipher text and the sufficient amount of information to decrypt that cipher text. The cipher text is generated from either 3DES, AES128, or AES256. The Kmaster, Khmac, and the Kenc is generated from KDF specifically PBKDF2 with an iteration count of 100,000. The randomly generated IV and the cipher text will pass to the HMAC with the Khmac to generate an authentication code. More detailed information about each steps are given in the following sections.

## 1. Output stream:

The output stream structure specifies which hashing algorithm is used, which symmetric encryption algorithm is used, iteration count, three salts that is being used with KDF, HMAC, IV and the cipher text.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 10~27 | 28~42 | 43~58 | … | … | … |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | | C | | | | | | D | E | F | G | H | I |

### A. Hashing algorithm identifier (1 bit)

| identifier | Algorithm |
|---|---|
| 0 | SHA256 |
| 1 | SHA512 |

### B. Symmetric encryption algorithm (2 bits)

| identifier | Algorithm |
|---|---|
| 00 | 3DES |
| 01 | AES128 |
| 10 | AES256 |

### C. Iteration Count (6 bits)

The iteration count is set to **100,000**

### D. Salt for Kmaster (18 characters)

The salt for Kmaster is set to **thisismymastersalt**

### E. Salt for Kenc (15 characters)

The salt for Kenc is set to **thisismyencsalt**

### F. Salt for Khmac (16 characters)

The salt for Khmac is set to **thisismyhmacsalt**

G. **HMAC output**

The HMAC output is the authentication code generates from **IV+ciphertext** with **Khmac.** The size of the HMAC output may be varying depend on the given hashing algorithm.

H. **Initialization vector**

The Initialization vector (**IV**) is randomly generated in Java. The IV is represented in *Hexadecimal*. IV size could be varying, 3DES would have an IV with the size of 8 bytes while AES128 and AES256 have the IV with the size of 16 bytes.

I. **Cipher text**

The cipher text is the output from the encryption algorithm specify in B with the given **IV**, plaintext, and the **Kenc**.

## 2. Key derivation function:

The KDF that is being used within this project is **PBKDF2** with the iteration count of **100,000**. The given password will pass PBKDF2 to generate the **Kmaster**. The default password is

**password**

Then **Khmac** and **Kenc** will be generated by passing the **Kmaster** to PBKDF2 with different salts. The hashing algorithm used in the KDF is the same as the given hashing algorithm. The length of **Kenc** depends on the encryption algorithm and the length of the **Khmac** depends on the given hashing algorithm.

- Generating the **Kmaster**:
  - Pass the **password** with iteration count **100,000** and **thisismymastersalt** to PBKDF2
  - Generate **256 bits of Kmaster** if hashing algorithm is SHA256
  - Generate **512 bits of Kmaster** if hashing algorithm is SHA512

- Generating the **Kenc**:
  - Pass the **Kmaster** with iteration count **100,000** and **thisismyencsalt** to PBKDF2
  - Generate **256 bits of Kenc** if encryption algorithm is AES256
  - Generate **512 bits of Kenc** if encryption algorithm is AES128 or 3DES

- Generating the **Khmac**:
  - Pass the **Kmaster** with iteration count **100,000** and **thisismyhmacsalt** to PBKDF2
  - Generate **256 bits of Khmac** if hashing algorithm is SHA256
  - Generate **512 bits of Khmac** if hashing algorithm is SHA512

## 3. Encryption:

The encryption algorithms that are being used within this project are 3DES, AES128, and AES256. The 3DES is in **CBC** mode with the sequence of **Encrypt-Decrypt-Encrypt** and padded in

**PKCS5**. The AES128 is in **CBC** mode and padded in **PKCS5**. The AES256 is in **CBC** mode and is padded in **PKCS7.** The cipher text will be encoded in ***BASE64***.

- **3DES Encryption:**
  - Randomly generated **IV** with the size of 8 bytes
  - Pass in the **Kmaster(128 bits)** and the **plain text**
  - Encrypted in **CBC** mode and padded in **PKCS5**
  - Cipher text is encoded to ***BASE64***

- **AES128 Encryption:**
  - Randomly generated **IV** with the size of 16 bytes
  - Pass in the **Kmaster(128 bits)** and the **plain text**
  - Encrypted in **CBC** mode and padded in **PKCS5**
  - Cipher text is encoded to ***BASE64***

- **AES256 Encryption:**
  - Randomly generated **IV** with the size of 16 bytes
  - Pass in the **Kmaster(256 bits)** and the **plain text**
  - Encrypted in **CBC** mode and padded in **PKCS7**
  - Cipher text is encoded to ***BASE64***

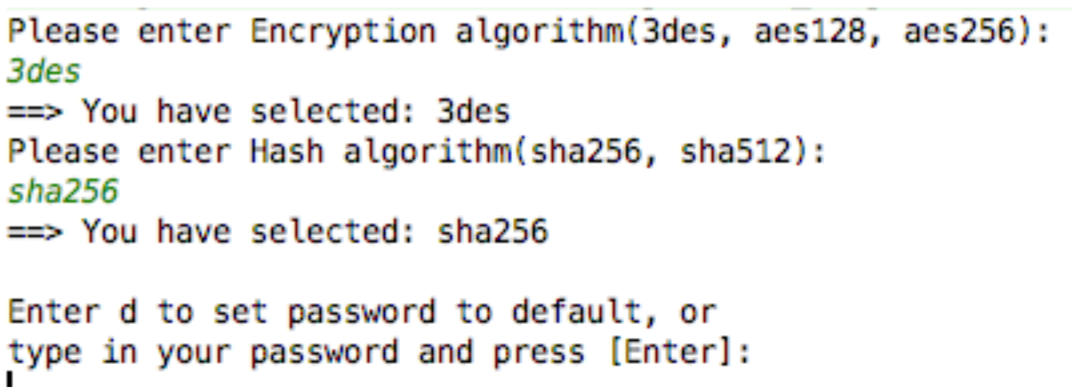
## 4. HMAC:

The HMAC is being used to generate authentication code for **IV+ciphertext. IV+ciphertext** will be passing with the **Khmac** to the HMAC and the output will be displayed in *Hexadecimal*. **Khmac** will be vary depend on the algorithm and the output size will be varying as well.

- **SHA256:**
  - Pass in the **Khmac(256 bits)** with **IV+ciphertext**
  - Generate the output with SHA256
  - Output will be displayed in *Hexadecimal*

- **SHA512:**
  - Pass in the **Khmac(512 bits)** with **IV+ciphertext**
  - Generate the output with SHA512
  - Output will be displayed in *Hexadecimal*

### 5. Output Example:

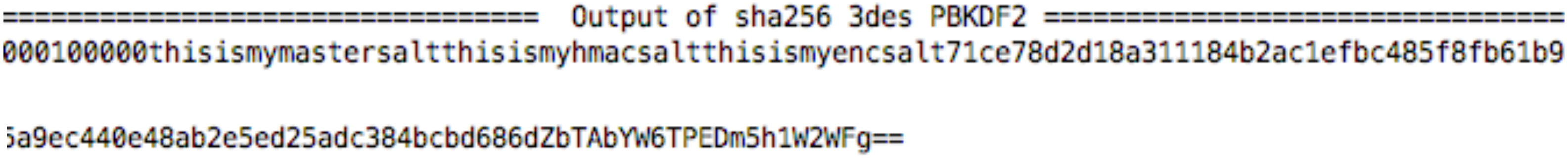


```
Please enter Encryption algorithm(3des, aes128, aes256):
3des
==> You have selected: 3des
Please enter Hash algorithm(sha256, sha512):
sha256
==> You have selected: sha256

Enter d to set password to default, or
type in your password and press [Enter]:
```

The user will enter their specific encryption and hashing algorithm for the program. The default password will be set to “password” if the user press d.

**The output:**



```
===============================  Output of sha256 3des PBKDF2 ===============================
000100000thisismymastersaltthisismyhmacsaltthisismyencsalt71ce78d2d18a311184b2ac1efbc485f8fb61b9

5a9ec440e48ab2e5ed25adc384bcbd686dZbTAbYW6TPEDm5h1W2WFg==
```

The first bit indicates that the user uses sha256 and the following two bits indicates the user uses 3des encryption. Iteration count 100,000 is followed and three of the salts.

- The HMAC output is:

71ce78d2d18a311184b2ac1efbc485f8fb61b9a5a9ec440e48ab2e5ed25adc38

- The random IV is:

4bcbd686

- The cipher text that is encoded in Base 64 is:

dZbTAbYW6TPEDm5h1W2WFg==