FYP Interim Presentation

Jacky Chan Hai Ten
3035070107

DESIGN AND IMPLEMENTATION OF CRYPTOGRAPHIC ALGORITHMS ON PC AND MOBILE PHONE

Coverage

- Motivation & Objectives
- Progress & Demo
- Problems Encountered
- Future Actions
- Q&A

Motivation & Objectives

- Mobile security development
- 3 Objectives
  - 1 - AES, SHA-2/3 (hash) & RSA on Windows PC
  - 2 - Above algorithms on Android Phone (ICS+)
  - 3 – El'gamal Encryption on android and incorporation with ABE

Progress

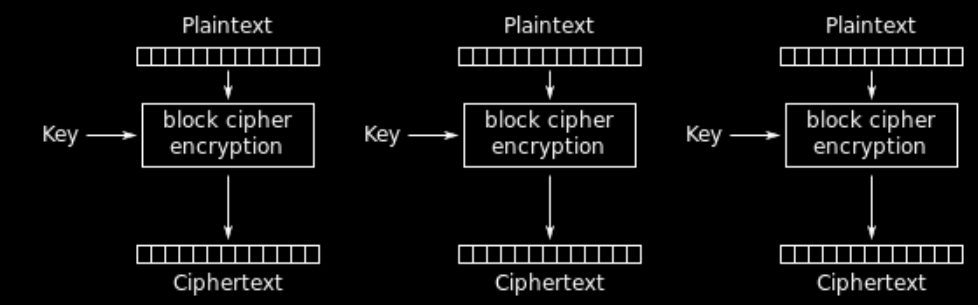
- Objective 1 – fully completed
- Objective 2 – largely completed
- Objective 3 – scheduled on 2nd semester

Objective 1

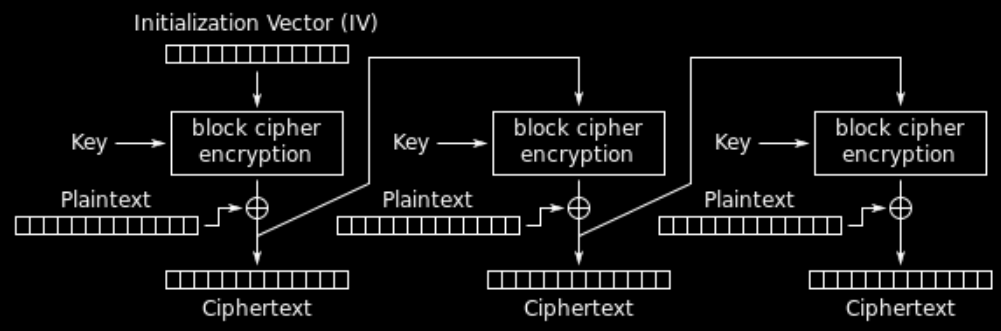
- Developed with C & OpenSSL library

Obj 1.1 - AES

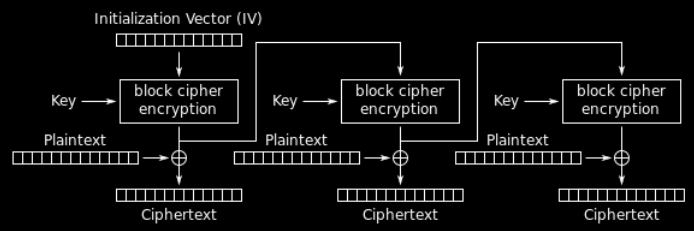
- Symmetric encryption
- Support 4 modes of operations
  - Electronic Codebook (ECB)
  - Cipher Block Chaining (CBC)
  - Cipher Feedback (CFB)
  - Output Feedback (OFB)
- 3 different key length
  - 128, 192 and 256 bits



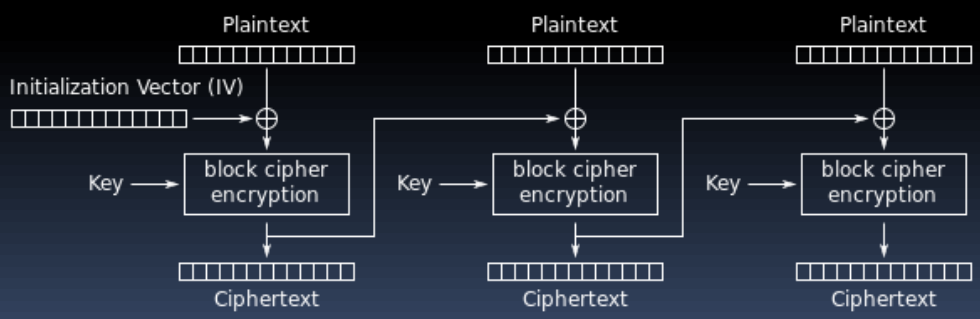
Electronic Codebook (ECB) mode encryption



Cipher Feedback (CFB) mode encryption



Output Feedback (OFB) mode encryption



Cipher Block Chaining (CBC) mode encryption

Obj 1.1 – Code Snippet

aes.h

```
typedef enum { UNDEFINED, ECB, CBC, CFB, OFB } aes_mode;

class AES {
private:

    /*
    data variables...
    */

    int init(int keyLength, aes_mode mode);
    int checkSetup();

public:
    AES(int keyLength, aes_mode mode);
    ~AES();

    int encrypt(const unsigned char *msg, size_t msgLen, unsigned char **encMsg);

    int decrypt(unsigned char *encMsg, size_t encMsgLen, unsigned char **decMsg);

    /*
    getters & setters...
    */
};
```

aes.c(partial)

```c
int AES::encrypt(const unsigned char *msg, size_t msgLen, unsigned char **encMsg) {

    int err = checkSetup();
    if (err == 0) {

        // Setup encryption
        size_t blockLen = 0;
        size_t encMsgLen = 0;
        const EVP_CIPHER* cipher = NULL;

        *encMsg = (unsigned char*)malloc(msgLen + block_size);
        if (encMsg == NULL) return FAILURE;

        switch (option_keyLength) { ... }

        if (!EVP_EncryptInit_ex(encryptCtx, cipher, NULL, key, IV)) {
            return FAILURE;
        }

        // Do the encryption
        if (!EVP_EncryptUpdate(encryptCtx, *encMsg, (int*)&blockLen, (unsigned char*)msg, msgLen)) {
            return FAILURE;
        }
        encMsgLen += blockLen;

        // Finalize
        if (!EVP_EncryptFinal_ex(encryptCtx, *encMsg + encMsgLen, (int*)&blockLen)) {
            return FAILURE;
        }

        EVP_CIPHER_CTX_cleanup(encryptCtx);
        return encMsgLen + blockLen;
    }
    else
        return err;
}
```

Obj 1.2 – Hashing

- One way mapping function
- Support 4 algorithms from SHA-2 family
  - SHA-224
  - SHA-256
  - SHA-384
  - SHA-512

hash.c(partial)

```
Hash::Hash(sha_mode mode) {

    option_mode = mode;
    if ((mdctx = EVP_MD_CTX_create()) == NULL) {
        printf("init failed\n");
    }
}

Hash::~Hash() { ... }

int Hash::digest(unsigned char *msg, int msgLen, unsigned char **digest, unsigned int *digestLen) {

    const EVP_MD* mode = NULL;

    switch (option_mode) { ... }

    if (!EVP_DigestInit_ex(mdctx, mode, NULL))
        return FAILURE;

    if (!EVP_DigestUpdate(mdctx, msg, msgLen))
        return FAILURE;

    if ((*digest = (unsigned char *)OPENSSL_malloc(EVP_MD_size(mode))) == NULL)
        return FAILURE;

    if (!EVP_DigestFinal_ex(mdctx, *digest, digestLen))
        return FAILURE;

    return SUCCESS;
```

Obj 1.3 RSA

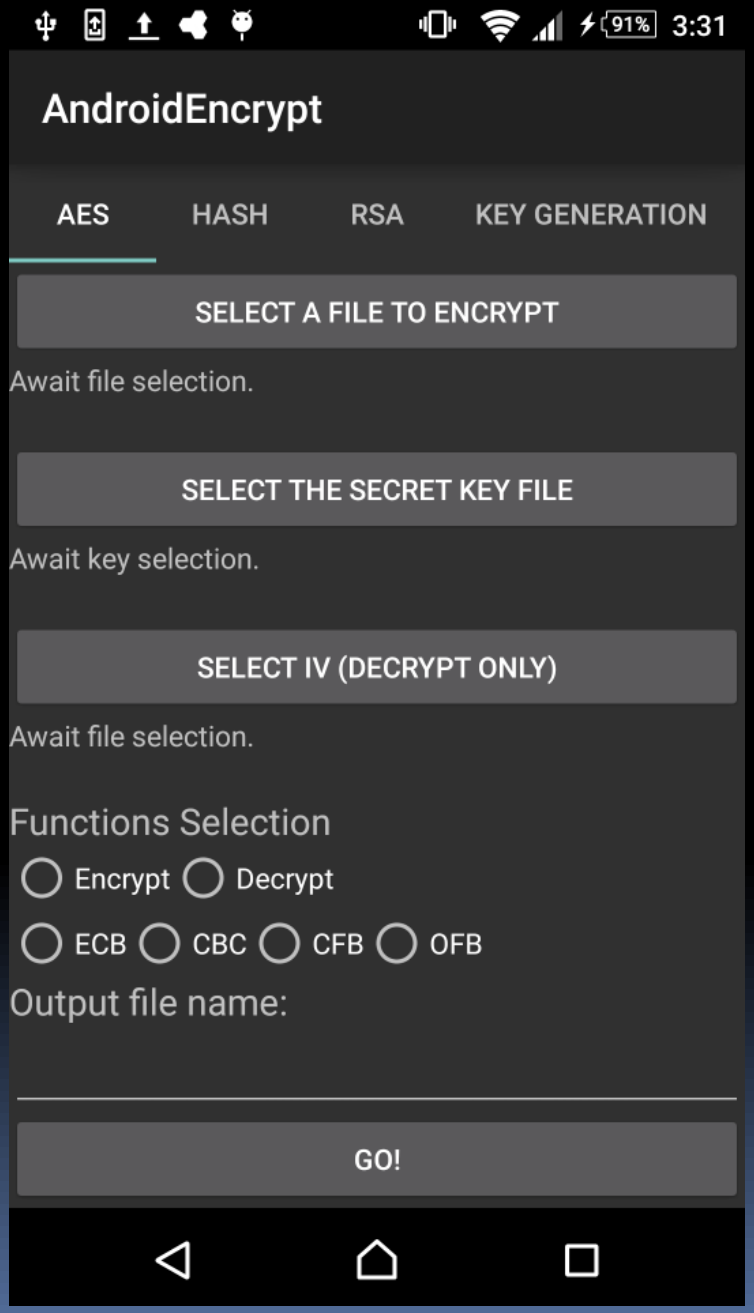
- Asymmetric encryption
  - Public & private key pairs
- Support using RSA to encrypt an one-time AES key

AsymEnc.c(partial)

```
    free(encryptCtx);
    free(decryptCtx);
}

int AsymEnc::seal(const unsigned char *msg, size_t msgLen, unsigned char **encMsg,
    unsigned char **ek, size_t *ekl, unsigned char **iv, size_t *ivl) {
    size_t encMsgLen = 0;
    size_t blockLen = 0;

    *ek = (unsigned char*)malloc(EVP_PKEY_size(keypair));
    *iv = (unsigned char*)malloc(EVP_MAX_IV_LENGTH);
    if (*ek == NULL || *iv == NULL) return FAILURE;
    *ivl = EVP_MAX_IV_LENGTH;

    *encMsg = (unsigned char*)malloc(msgLen + EVP_MAX_IV_LENGTH);
    if (encMsg == NULL) return FAILURE;

    if (!EVP_SealInit(encryptCtx, EVP_aes_256_cbc(), ek, (int*)ekl, *iv, &keypair, 1)) { ... }

    if (!EVP_SealUpdate(encryptCtx, *encMsg + encMsgLen, (int*)&blockLen, (const unsigned char*)msg, (int)msgL
    encMsgLen += blockLen;

    if (!EVP_SealFinal(encryptCtx, *encMsg + encMsgLen, (int*)&blockLen)) { ... }
    encMsgLen += blockLen;

    EVP_CIPHER_CTX_cleanup(encryptCtx);

    return (int)encMsgLen;
```

Objective 2

- Developed with Java & JCE
- A Java program that run on PC is completed
- Porting to android is in progress


91%
3:31
AndroidEncrypt
AES
HASH
RSA
KEY GENERATION
SELECT A FILE TO ENCRYPT
Await file selection.
SELECT THE SECRET KEY FILE
Await key selection.
SELECT IV (DECRYPT ONLY)
Await file selection.
Functions Selection
Encrypt
Decrypt
ECB
CBC
CFB
OFB
Output file name:
GO!

aes.java

```
    if(!isValid)
        System.err.print("AES Object setting invalid\n");
    else{

        cipherString = this.getCipherString();
    }
}

public byte[] encrypt(byte[] plainText, IvWrapper iv){

    if(isValid){

        Cipher aes;
        try {

            aes = Cipher.getInstance(cipherString);

            iv.iv = Keygen.getSecureRandomBytes(BLOCK_SIZE/8);
            aes.init(Cipher.ENCRYPT_MODE, key, new IvParameterSpec(iv.iv));
            byte[] cipherText = aes.doFinal(plainText);
            return cipherText;

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    else{
```

Cross Platform Demo

- Do the encryption from PC
- Decryption from Phone, vise versa

```
F:\fyp\javaencrypt\testfiles>wencrypt.exe -keygen -sym -128 -f "AES128.key"
Generating a symmetric key of 128 bits...
Key successfully written to "AES128.key"!

F:\fyp\javaencrypt\testfiles>wencrypt.exe -aes -e -cbc -f "HKU-shield.jpg" -k "A
ES128.key"
Generating random IV for the encryption...Okay!
IV written to HKU-shield.jpg.iv.
818383 bytes to be encrypted
818384 bytes encrypted
Encrypted message written to "HKU-shield.jpg.enc"

F:\fyp\javaencrypt\testfiles>_
```

AES128.key	13/1/2016 6:04	KEY 檔案	1 KB
e&d.bat	13/1/2016 3:06	Windows 批次檔案	1 KB
HKU-shield.jpg	13/1/2016 6:06	JPEG 影像	800 KB
HKU-shield.jpg.enc	13/1/2016 6:07	ENC 檔案	800 KB
HKU-shield.jpg.iv	13/1/2016 6:07	IV 檔案	1 KB
jencrypt.jar	5/1/2016 1:53	Executable Jar File	10 KB
test.mp3	28/7/2014 15:29	MP3 格式聲音	5,835 KB
wencrypt.exe	4/1/2016 20:47	應用程式	67 KB

HKU-shield.jpg

HKU-shield.jpg.enc

Windows 相片檢視器無法開啟此圖片，因為相片檢視器不支援此檔案格式，或您並未安裝相片檢視器的最新更新。

Problems Encountered

- Compatibility of encryption algorithm on different android devices
  - Manufacturer has their own standard

Future Actions

- Continue developing objective 3
  - El Gamal encryption on android
  - Incorporate with Attribute based encryption
- Enhance the flexibility
  - Allow more parameters, setting
  - UI improvement

Q&A