EHN 410

Practical 2 Lecture

Tapfuma Chanaiwa

University of Pretoria

March 6, 2020

Outline

1. Cipher Modes of Operation
   - Cipher block Chaining
   - Cipher feedback mode
2. Padding
3. AES
   - AES rounds
4. Practical 2
5. Deliverables

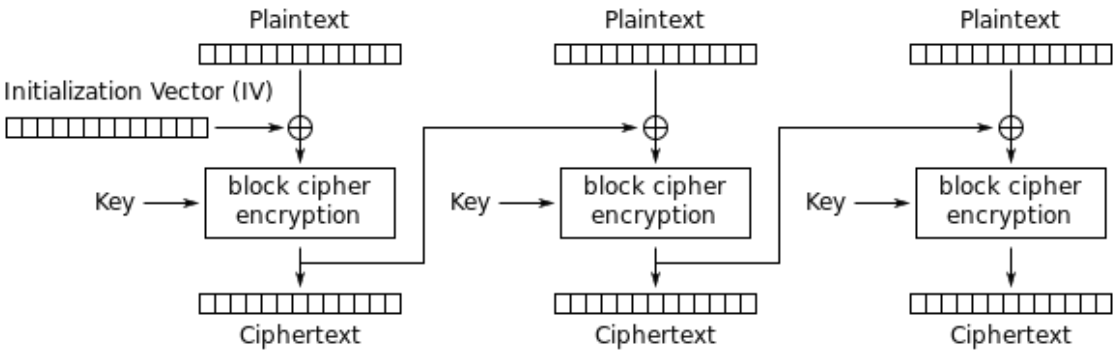
Cipher Modes of Operation

- Cipher block chaining (CBC)
- Electronic Codebook (ECB)
- Cipher Feedback Mode (CFB)
- Output Feedback Mode (OFB)
- Counter Mode (CTR)

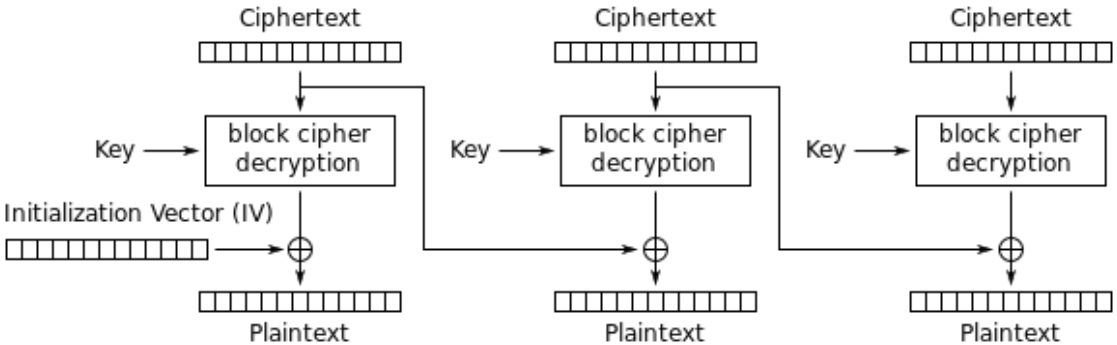
Cipher block Chaining

- Cipher-text is a function of the initialisation vector, plain-text and the key.
- The message is divided into blocks
- After each encrypted block the cipher becomes the new initialisation vector.
- Each block is encrypted concurrently.
- The implication is that identical plain-text blocks will not provide identical cipher-text blocks if the initialisation vector differs.

Cipher block Chaining



Cipher Block Chaining (CBC) mode encryption

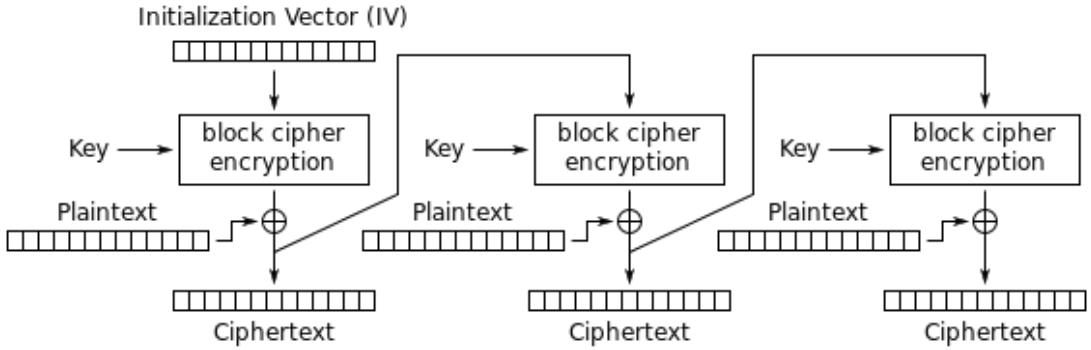


Cipher Block Chaining (CBC) mode decryption

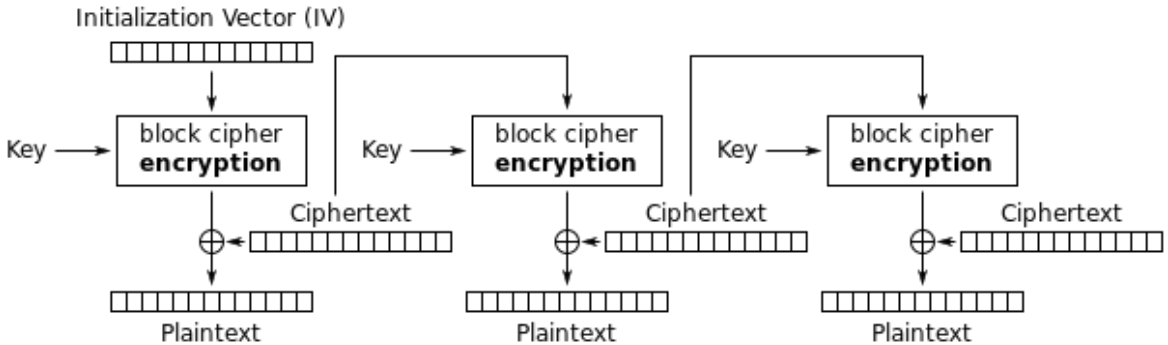
Cipher feedback mode

- The cipher-text is the same length as the plain-text
- Cipher-text is a function of the preceding plain-text and the current plain-text.
- The stream-length and block-length do not match (i.e. the encryption algorithm might use a block-length of 64-bits but the stream cipher only makes use of 8-bits)
- Input to the encryption algorithm is a shift-register set to the Initialisation vector (IV).
- MSB of the block cipher is XOR'ed with the first current plain-text.
- Contents of the shift-register is shifted left by one byte and the cipher-text is placed in the LSB of the register.

Cipher feedback mode



Cipher Feedback (CFB) mode encryption



Cipher Feedback (CFB) mode decryption

Padding information

Regularly there aren't enough data to fill up an encryption block. In such cases, these block need to be filled in for the encryption to proceed. The following lists the most commonly used padding schemes:

- Zero padding
- Ansi X.932
- ISO 10126
- PKCS7

Note: All padding schemes are based on hexadecimal/byte values.

Advanced Encryption Standard (AES)

- Created by Joan Daemon and Vincent Rijmen.
- Accepted by the National Institute of Standard Technology (NIST) in 2001.
- Also known as Rijndael
- Is based on a substitution-permutation network.
- Created as a substitute for DES.
- Unlike a Feistel network (used in DES), AES's encryption and decryption differ in procedure.

Advanced Encryption Standard (AES)

AES makes use of Rijndael's key schedule. This key schedule has three operations that it carries out:

- Rotating a 32-bit word (4 bytes)
- Exponentiation of 2, also known as the Galois Field(GF 2^8)
- Substitution boxes (S-box)

AES block sizes

- Also commonly referred to as as the state array
- It has a fixed block size of 128-bits (i.e. block size of 4×4, each containing one byte)
- The block is populated column-wise, as shown below

$byte_1$	$byte_5$	$byte_9$	$byte_{13}$
$byte_2$	$byte_6$	$byte_{10}$	$byte_{14}$
$byte_3$	$byte_7$	$byte_{11}$	$byte_{15}$
$byte_4$	$byte_8$	$byte_{12}$	$byte_{16}$

AES block sizes

- If the plain-text is not enough to fill the last block, padding is used to fill the block.
- Zero-padding is the simplest padding method where the hexadecimal value of zero (i.e. $0x00$) is used to fill the block.

$0x4e$	$0x5a$	$0xb9$	$0xb3$
$0x34$	$0x8d$	$0x3e$	$0x00$
$0x65$	$0x1a$	$0xe3$	$0x00$
$0x3f$	$0xc8$	$0x6f$	$0x00$

AES key sizes

- 128-bits
- 192-bits
- 256-bit

Difference between the key sizes is the amount of cycles of repetition that the keys go through (i.e. 10 cycles for 128-bit keys).

AES rounds

Once the Rijndael key schedule has been created the algorithm goes through a number of rounds:

- Initial round
- Repetition rounds
- Final round

Each of these rounds contain one or more of the following procedures:

- A Subbytes step
- Row shifting step
- Mixing columns steps
- A round-key step

*(See the practical guide for more details)

Practical 2

The following needs to be implemented for practical 2:

- A 128-, 196- and 256-bit AES encryption/decryption algorithm.
- CBC cipher running on your AES implementation.
- A CFB stream cipher using a stream-size of 64-bits with your AES implementation.

Deliverables

Demo

- Date: 9 April
- I will upload files 1 hour before the demo that you will need to decrypt.
- Your implementation should be able to decrypt every combination of the encryption scheme (AES 128, 196 and 256) with the cipher mode (CBC and CFB).
- Ensure that you have a way to access (and display in hexadecimal) each individual function's results on the day of the demo

Example

```
kobie@Madness:~/Dropbox/AL/EHN410/2015/Practicals/Prac2/Code$
kobie@Madness:~/Dropbox/AL/EHN410/2015/Practicals/Prac2/Code$ ./aes_cbc -m "test functionality"
running test
test functionality
sz: 18
Input string:

        74 20 63 6e
        65 66 74 61
        73 75 69 6c
        74 6e 6f 69
Block setup (hex):
        74 20 63 6e
        65 66 74 61
        73 75 69 6c
        74 6e 6f 69
Subbyte step:
        92 b7 fb 9f
        4d 33 92 ef
        8f 9d f9 50
        92 9f a8 f9
Shiftrows step:
        74 20 63 6e
        66 74 61 65
        69 6c 73 75
        69 74 6e 6f
MixColumns step:
        74 20 63 6e
        65 66 74 61
        73 75 69 6c
        74 6e 6f 69
KeyExpansions step:expanded key:
Creating expansion keys
74      65      73      74      20      66      75      6e      63      74      69      6f      6e      61      6c      69
9a      35      8a      eb      ba      53      ff      85      d9      27      96      ea      b7      46      fa      83
c2      18      66      42      78      4b      99      c7      a1      6c      f       2d      16      2a      f5      ae
23      fe      82      5       5b      b5      1b      c2      fa      d9      14      ef      ec      f3      e1      41
26      6       1       cb      7d      b3      1a      9       87      6a      e       e6      6b      99      ef      a7
d8      d9      5d      b4      a5      6a      47      bd      22      0       49      5b      49      99      a6      fc
16      fd      ed      8f      b3      97      aa      32      91      97      e3      69      d8      e       45      95
fd      93      c7      ee      4e      4       6d      dc      df      93      8e      b5      7       9d      cb      20
23      8c      70      2b      6d      88      1d      f7      b2      1b      93      42      b5      86      58      62
7c      e6      da      fe      11      6e      c7      9       a3      75      54      4b      16      f3      c       29
47      18      7f      b9      56      76      b8      b0      f5      3       ec      fb      e3      f0      e0      d2
kobie@Madness:~/Dropbox/AL/EHN410/2015/Practicals/Prac2/Code$
```

Deliverables

Submission

- Doxygen documentation (as a PDF).
- Code and makefile
- Code appended into a single pdf file.
- "ReadMe" file containing usage instructions.
- Upload a ZIP file where the file-name should be:
  EHN410-G*XX*-P2
  where:
  - GXX your group number.
  - P2 refers to practical 2.